IDtrust 2010
9th Symposium on Identity and Trust on the Internet
Program

Notes

Transportation

There will be a shuttle bus leaving the Gaithersburg Holiday Inn at 8:00 a.m. Tuesday and Wednesday morning to travel to NIST. The shuttle will return to the hotel at the end of the sessions on Tuesday and Wednesday. There will not be a shuttle bus on Thursday - please car pool or use the hotel shuttle.

Wireless

802.11b Wireless access points will be available for at least SSH, IPSEC, HTTP, DNS, FTP, POP, IMAP, and SMTP connectivity.

Proceedings at ACM Digital Library

The proceedings are also available in the ACM International Conference Proceeding Series archive: Proceedings of the 9th Symposium on Identity and Trust on the Internet (ISBN:978-1-60558-895-7).

Blogging

Participants and observers are encouraged to use the tag "idtrust2010" when blogging and tweeting about the symposium.

Program

Tuesday, April 13, 2010 - Full Day

8:00 Bus Departs from Gaithersburg Holiday Inn for NIST
8:30 - 9:00 Registration and Continental Breakfast
9:00 - 9:10 Welcome and Opening Remarks
Program Chair: Carl Ellison, *Independent* (Slides: pptx)

9:10 - 10:00 Keynote Talk I
The Central Role of Identity Authentication in Immigration Reform
Bruce Morrison, *Morrison Public Affairs Group*

10:00 - 10:15 Break
10:15 - 11:40 Session 2: Technical Papers - Identity Providers and Federation
Session Chair: Peter Alterman, *National Institutes of Health*

Federated Login to TeraGrid
(Presentation slides: pptx pdf)

Jim Basney, *National Center for Supercomputing Applications*
Terry Fleury, *National Center for Supercomputing Applications*
Von Welch, *National Center for Supercomputing Applications*
An Identity Provider to manage Reliable Digital Identities for SOA and the Web
(Presentation slides: pdf)

Ivonne Thomas, *Hasso-Plattner-Institute*
Christoph Meinel, *Hasso-Plattner-Institute*
CardSpace-Liberty Integration for CardSpace Users
(Presentation slides: ppt)

Haitham Al-Sinani, *Royal Holloway, University of London*
Waleed Alrodhan, *Royal Holloway, University of London*
Chris Mitchell, *Royal Holloway, University of London*

11:40 - 12:00 Break
12:00 - 1:00 Session 3: Technical Papers - Policy Conflict Resolution
Session Chair: David Chadwick, *University of Kent*

An Attribute-based Authorization Policy Framework with Dynamic Conflict Resolution
(Presentation slides: ppt)

Apurva Mohan, *Georgia Tech*
Douglas Blough, *Georgia Tech*
Computational Techniques for Increasing PKI Policy Comprehension by Human Analysts
(Presentation slides: ppt)

Gabriel Weaver, *Dartmouth College*
Sean Smith, *Dartmouth College*
Scott Rea, *Dartmouth College*

1:00 - 2:00 Lunch
2:00 - 3:30 Session 4: Panel - Identity Proofing

Panel Moderator: Elaine Newton, *National Institute of Standards and Technology*

Darrell Williams, *Department of Homeland Security*

Jim McCabe, *American National Standards Institute (ANSI)* (Slides: ppt pdf)

Brian Zimmer, *Coalition for a Secure Driver's License*

3:30 - 4:00 Break
4:00 - 5:30 Session 5: Panel - Four Bridges Forum: How Federated Identity Trust Hubs Improve Identity Management

Panel Moderator: Peter Alterman, *National Institutes of Health*

Tim Pinegar, *Federal PKI Architecture* (Slides: pptx pdf)

Mollie Shields-Uehling, *SAFE-BioPharma Association* (Slides: ppt)

Scott Rea, *HEBCA Operating Authority* (Slides: ppt)

Jeff Nigriny, *CertiPath* (Slides: ppt)

5:30 Bus Departs for Gaithersburg Holiday Inn
6:00 Social Gathering and Dinner Buffet - Gaithersburg Holiday Inn

---

Wednesday, April 14, 2010 - Full Day

8:00 Bus Departs from Gaithersburg Holiday Inn for NIST
8:30 - 9:00 Registration and Continental Breakfast
9:00 - 9:50 Keynote Talk II

Internet Voting: Threat or Menace?

(Presentation slides: ppt pdf)

Jeremy Epstein, *SRI International*

9:50 - 10:10 Break
10:10 - 11:10 Session 7: Panel - End-to-End and Internet Voting

Panel Moderator: Neal McBurnett, *Internet2*

Poorvi Vora, *George Washington University* (Slides: ppt)

Jeremy Epstein, *SRI International*

11:10 - 11:40 Invited Talk - Using the DNS as a Trust Infrastructure with DNSSEC

Scott Rose, *NIST* (Slides: pptx)

11:40 - 12:00 Break
12:00 - 1:00 Session 8: Technical Papers - Privacy

Session Chair: Stephen Whitlock, *Boeing*

Efficient and Privacy-Preserving Enforcement of Attribute-Based Access Control
(Presentation slides: pdf)

Ning Shang, *Purdue University*
Federica Paci, *University of Trento*
Elisa Bertino, *Purdue University*

Privacy-Preserving DRM
(Presentation slides: ppt)

Radia Perlman, *Intel Labs*
Charlie Kaufman, *Microsoft*
Ray Perlner, *NIST*

1:00 - 2:00 Lunch

2:00 - 2:30 Session 9: Invited Talk - Hash Competition
Introduction: Carl Ellison, *Independent*

Bill Burr, *National Institute of Standards and Technology* (Slides: pdf ppt)

2:30 - 3:00 Session 10: Technical Papers - Biometrics
Session Chair: David Chadwick, *University of Kent*

Biometrics-Based Identifiers for Digital Identity Management
(Presentation slides: pdf)

Abhilasha Bhargav-Spantzel, *Intel Corporation*
Anna Squicciarini, *Pennsylvania State University*
Elisa Bertino, *Purdue University*
Xiangwei Kong, *Dalian University of Technology*
Weike Zhang, *Dalian University of Technology*

3:00 - 3:30 Break

3:30 - 4:00 Session 11: Invited Talk: Personal Identity Platforms
Bill MacGregor, *National Institute of Standards and Technology* (Slides: ppt)

4:00 - 5:30 Session 12: Panel - The Path to Citizen Identity Federation Worldwide: How Kantara Initiative Programs are enabling Citizen Identity Federation
Panel Moderator: Roger Martin, *Kantara Initiative*

Vikas Mahajan, *AARP*
Jim Zok, *Computer Sciences Corporation*
Elaine Newton, *National Institute of Standards and Technology*
Jack Leipold, *Social Security Administration*
Bill Young, *Department of Internal Affairs*

5:30 Bus Departs for Gaithersburg Holiday Inn
Dinner (on your own)

Thursday April 15, 2010 - Half Day

8:00 No Bus - please share rides to NIST
8:30 - 9:00 Registration and Continental Breakfast
9:00 - 10:00 Session 13: Technical Papers - Infrastructure

Session Chair: Peter Alterman, *National Institutes of Health*

Practical and Secure Trust Anchor Management and Usage

(Presentation slides: ppt pdf)

Carl Wallace, *Cygnacom Solutions*

Geoff Beier, *Cygnacom Solutions*

A Proposal for Collaborative Internet-scale trust infrastructures deployment: the Public Key System (PKS)

(Presentation slides: pdf)

Massimiliano Pala, *Dartmouth College*

10:00 - 10:30 Break
10:30 - 11:30 Session 14: Panel – Levels of Assurance for Attributes

Panel Moderator: Carl Ellison, *Independent* (Slides: pptx)

David Chadwick, *University of Kent* (Slides: ppt)

Ken Klingenstein, *Internet2* (Slides: ppt)

Chris Louden, *Protiviti* (Slides: ppt)

Peter Alterman, *National Institutes of Health* (Slides: ppt)

11:30 - 12:15 Session 15: RUMP Session (Work in Progress)

Session Chair: Neal McBurnett, *Internet2*

Deployment Experience for the PKI Resource Query Protocol for Grids and FBPKI

(Presentation slides: odp pdf)

Massimiliano Pala, *Dartmouth College*

Preferred model for multiple federations: 'Interfederation' ala the Internet or 'Superposition of federations' ala credit card industry?

Bill MacGregor's question, *NIST*

12:15 - 12:30 Wrap up

See Also

This workshop is part of the IDtrust Symposium Series

- 2011: 10th Symposium on Identity and Trust on the Internet (IDtrust 2011)
- 2010: 9th Symposium on Identity and Trust on the Internet (IDtrust 2010)
- 2009: 8th Symposium on Identity and Trust on the Internet (IDtrust 2009)
- 2008: 7th Symposium on Identity and Trust on the Internet (IDtrust 2008)
- 2007: 6th Annual PKI R&D Workshop
- 2006: 5th Annual PKI R&D Workshop
- 2005: 4th Annual PKI R&D Workshop
- 2004: 3rd Annual PKI R&D Workshop
- 2003: 2nd Annual PKI Research Workshop
- 2002: 1st Annual PKI Research Workshop

IDtrust 2010

13 April 2010

Sponsors

- NIST
- Internet2
- Federal Public Key Infrastructure Policy Authority (FPKIPA)
- OASIS IDtrust Member Section

Special Thanks

- To the program committee – who did a great job evaluating and selecting papers
- To Neal McBurnett – who carried a heavy load and provided stability through this process
- To Radia Perlman – for coordinating the selection of panels

- And most especially, to Sara Caswell – who made this symposium happen

Brief History of IDtrust

- 2001: PKI Labs
- 2002: the 1st PKI Research Workshop
- 2004: name change: PKI R&D Workshop
- 2008: IDtrust

Longer History

- 1976: *New Directions in Cryptography*
- 1978: RSA
- 1978: Loren Kohnfelder
- 1980s: X.500 and X.509
- 1990s: commercial CAs
- Where's the beef?
  - Bridges, Federation, Applications, SSO
  - Real security policies

The Real Job: Making a Security Decision

- Deciding what policy to enforce; and
- Translating that policy into a form computers and other humans can understand.

- Our end customers have real security problems – not just a desire to deploy PKI.
  - PKI, bridges, federation, smart-cards, etc., are tools available to address the real problems and are no longer ends in themselves.

Today's Keynote

- Bruce Morrison:
  - The Central Role of Identity Authentication in Immigration Reform

Federated Login to TeraGrid

Jim Basney
jbasney@illinois.edu

Terry Fleury
tfleury@illinois.edu

Von Welch
vwelch@illinois.edu

National Center for Supercomputing Applications
University of Illinois
1205 West Clark Street
Urbana, Illinois 61801

ABSTRACT

We present a new federated login capability for the TeraGrid, currently the world's largest and most comprehensive distributed cyberinfrastructure for open scientific research. Federated login enables TeraGrid users to authenticate using their home organization credentials for secure access to TeraGrid high performance computers, data resources, and high-end experimental facilities. Our novel system design links TeraGrid identities with campus identities and bridges from SAML to PKI credentials to meet the requirements of the TeraGrid environment.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Security

Keywords

PKI, SAML, identity federation, grid computing, TeraGrid, MyProxy, GridShib, Shibboleth

1. INTRODUCTION

TeraGrid[1] is an open scientific discovery infrastructure combining leadership class resources at eleven partner sites to create an integrated, persistent computational resource. TeraGrid serves over 4,500 researchers from over 300 colleges, universities, and research institutions in the United States. TeraGrid resources are allocated to researchers by peer review. Researchers must authenticate to TeraGrid resource providers and charge their usage to project accounts. TeraGrid supports authentication via passwords, SSH public keys, and X.509 certificates.

[1] `http://www.teragrid.org`

IDtrust '10, April 13-15, 2010, Gaithersburg, MD

In this article, we present the design and implementation of a new system that enables researchers to use the authentication method of their home organization for access to TeraGrid. Participating in the InCommon Federation[2] enables TeraGrid to accept authentication assertions from U.S. institutions of higher education, so researchers can use their existing campus login to authenticate to TeraGrid resources.

This federated login capability brings multiple benefits:

- It mitigates the need for researchers to manage authentication credentials specific to TeraGrid in addition to their existing campus credentials. Simplifying researchers' access to TeraGrid helps them to better focus on doing science.

- Reducing or eliminating the need for a TeraGrid password eases the burden on TeraGrid staff, by reducing the number of helpdesk calls requesting password resets and avoiding the need to distribute passwords to researchers in the first place.

- Using the campus login to access TeraGrid helps to integrate campus computing resources with TeraGrid resources. Researchers should be able to easily combine resources on campus with resources from TeraGrid and other national cyberinfrastructure. Harmonizing security interfaces across the infrastructure is a positive step towards this goal.

- Federated login enables the provisioning of TeraGrid resources according to campus-based identity vetting and authorization. TeraGrid resources could be allocated to a university class or department, and TeraGrid could rely on the university to determine who on their campus is authorized to use the resource allocation (e.g., who is enrolled in the class or who is a department member), thereby eliminating the need for per-user accounting by TeraGrid staff and giving the campus greater flexibility and control in managing the TeraGrid allocation.

Federated login is being applied in many environments to simplify authenticated access to resources and services. In this article, we focus on the unique challenges we faced in implementing federated login for TeraGrid. A primary technical challenge was the need to support multiple usage models, from interactive browser and command-line access

[2] `http://www.incommonfederation.org`

to multi-stage, unattended batch workflows. Another challenge was the need to establish trust among campuses, TeraGrid members, and peer grids (such as Open Science Grid[3] and the Enabling Grids for E-sciencE[4]) in the mechanisms and procedures underlying the federated login capability. In the remainder of the article, we discuss these and other challenges and present our solution in detail.

2. BACKGROUND

Before presenting the federated login capability we developed for TeraGrid, we first provide background information about the previously existing TeraGrid authentication architecture and the InCommon Federation.

2.1 TeraGrid Authentication Architecture

The TeraGrid allocations process provisions TeraGrid user accounts and assigns TeraGrid-wide usernames and passwords, which grant single sign-on access to TeraGrid resources. Our work, which we describe subsequently, leverages this existing architecture without modifying it in order not to disrupt access for existing users.

2.1.1 TeraGrid Allocations

As described in the Introduction, TeraGrid resources are allocated to researchers by peer review. Principal Investigators (PIs) submit proposals for resource allocations to a resource allocations committee, which consists of volunteers selected from the faculty and staff of U.S. universities, laboratories, and other research institutions. All members serve a term of 2–5 years and have expertise in computational science or engineering. Each proposal is assigned to two committee members for review. The committee members can also solicit an external review. After several weeks of review, the entire committee convenes to discuss the relative merits of each proposal and award time based on availability of resources. To apply, the PI must be a researcher or educator at a U.S. academic or non-profit research institution. Proposals are judged on scientific merit, potential for progress, numerical approach, and justification for resources. Allocations are typically awarded for one year, though multi-year allocations may be granted for well-known PIs. PIs can submit renewal or supplemental proposals to the committee to extend their allocation.

PIs are instructed not to share their accounts with others. Instead, they use the Add User Form on the TeraGrid User Portal[5] to request accounts for their project members. PIs can also use this form to remove project members. PIs submit name, telephone, email, and postal address information for the users on their project. For users on multiple projects, each project PI must complete the required information separately for each user to request the user to have access to the project's resources. The PI is notified by postal mail whenever a user is added to their project. All users are required to sign the TeraGrid User Responsibility Form, which educates users about secure and appropriate computing practices.

When a PI's proposal is accepted, or when an active PI requests an account for a project member, TeraGrid allocations staff members enroll the PI or project member in the TeraGrid Central Database, assign a TeraGrid-wide user-

[3] `http://www.opensciencegrid.org`
[4] `http://www.eu-egee.org`
[5] `https://portal.teragrid.org`

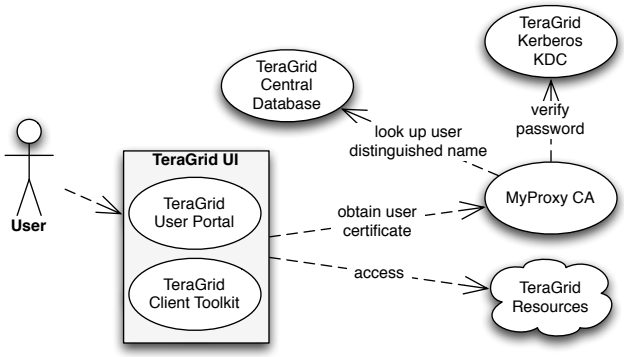


Figure 1: TeraGrid single sign-on provides certificates for secure access to TeraGrid resources.

name and initial password to the researcher, and send the username and password via postal mail to the researcher. The letter distributed with the initial password instructs the researcher to change the password and store the letter in a secure place. If the researcher forgets the password, he or she can call the helpdesk and request that the password be reset to the initial value. If the researcher has lost the letter with the initial password, he or she can call the helpdesk and request that a new letter be sent to their postal address on record. Alternatively, a researcher can reset his or her password via the TeraGrid User Portal, which authenticates the request via the researcher's registered email address. In the future, TeraGrid researchers will be able to set their username and password when they request an account, eliminating the need for passwords to be sent via postal mail.

The process of enrolling a new user into the TeraGrid Central Database also assigns a unique certificate subject distinguished name to the user. The distinguished name includes the user's first and last names, with an optionally appended serial number in case of name conflicts. The database management system ensures that distinguished names are uniquely assigned and are never re-assigned to a different user.

As described later, our federated login solution relies on the fact that the TeraGrid Central Database contains a record for every TeraGrid user, as well as the fact that every TeraGrid user has a TeraGrid-wide username and password.

2.1.2 TeraGrid Single Sign-On

The researcher's TeraGrid-wide username and password enables single sign-on access to all TeraGrid resources. Researchers can use TeraGrid single sign-on from the TeraGrid User Portal (TGUP) and from the command-line (via the TeraGrid Client Toolkit). Upon entering their username and password, researchers obtain a short-lived certificate from a MyProxy[6] Certificate Authority (CA) [1, 6] operated by NCSA. Researchers use this certificate to authenticate to remote login, data transfer, batch job submission, and other services. Furthermore, researchers can delegate a proxy certificate [15] to remote login sessions and batch jobs, allowing those sessions/jobs to access resources on their behalf. Figure 1 presents the TeraGrid single sign-on system architecture.

[6] `http://myproxy.ncsa.uiuc.edu`

The TeraGrid PKI consists of CAs (including the NCSA MyProxy CA) operated by TeraGrid member institutions and other partners. TeraGrid resource providers accept a consistent set of CAs to facilitate single sign-on across the TeraGrid resources. The TeraGrid Security Working Group reviews requests to add or remove CAs and operates by consensus across the TeraGrid members. According to the policy of the working group, new CAs must be accredited by the International Grid Trust Federation (IGTF),[7] the de facto standards body for defining levels of assurance for PKIs in production academic grids around the world. As discussed subsequently, IGTF accreditation was an important step in deploying a new federated CA in TeraGrid in support of single sign-on with federated login.

TeraGrid runs a Kerberos domain to validate usernames and passwords. Kerberos is not typically exposed to end users directly but is instead used by other services (such as the MyProxy CA) as an authentication service.

2.2 InCommon Federation

The InCommon Federation enables users to use their local identity, assigned by their campus, to access services such as academic publications and educational materials, and to collaborate with partners outside the borders of the campus. InCommon facilitates the adoption of standard policies by federation participants on technology issues, legal issues, and acceptable uses of identity information. Several U.S. federal agencies (e.g., NSF, NIH) have joined InCommon, and national-scale infrastructures such as the Ocean Observatories Initiative[8] are exploring its use. InCommon promises to provide a standard interface to the differing campus identity management systems and allow outside leverage of local identities without the need to understand the nuances at each campus.

Many federation members use the Shibboleth[9] software for expressing and exchanging identity information between organizations. Shibboleth allows organizations to federate identity information. In practical terms, this means a user from one institution can authenticate at their home institution and have the resulting identity (identifier and/or attributes) made available to a second institution for the purposes of accessing resources at that second institution. Shibboleth is commonly used in privacy-preserving applications, where access to resources is granted based on the user's attributes (e.g., "University of Illinois student") without requiring disclosure of the user's name or other identifying information. For example, many universities partner with online content providers to enable students to access journal articles using Shibboleth attributes. Shibboleth implements the SAML Web Browser Single Sign-On protocols,[10] which work well for browser-based applications but do not translate directly to the command-line, complex-workflow, unattended/batch processes that make up a significant proportion of TeraGrid computing workloads.

As of January 2010, the InCommon Federation includes over 200 universities, representing over 4 million users. Of the 38 institutions that each represent over 50 TeraGrid users, 24 (67%) are currently InCommon members. While

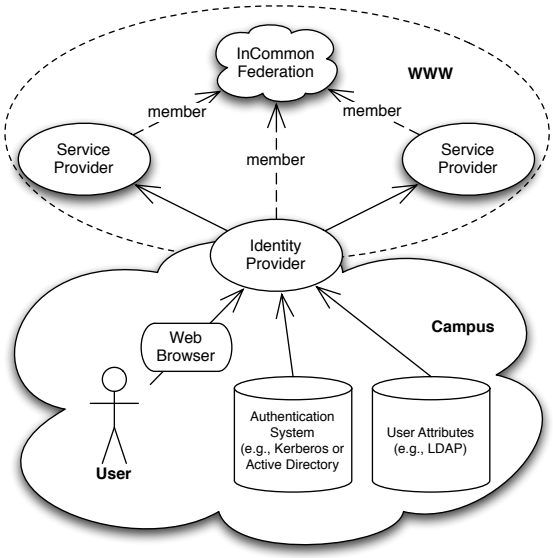


Figure 2: The InCommon Federation defines standard behavior, attributes, and protocols. The campus identity provider converts the user's campus identity into standard SAML format for access to web services.

InCommon membership continues to grow, many TeraGrid users come from campuses that are not (yet) InCommon members. InCommon member ProtectNetwork[11] operates an open identity provider that can provide logins for these users.

As depicted in Figure 2, the operational components of the InCommon Federation are the *identity providers*, *service providers*, and the *federation* that brings them together. Identity providers convert the user's campus identity (identifier and/or attributes) into the standard SAML format, providing single sign-on to multiple service providers and supporting anonymity, pseudonymity, and other privacy controls. SAML identity providers rely on campus authentication systems (such as Kerberos) and attribute stores (such as LDAP) to authenticate users and provide identity information. Service providers consume SAML assertions from identity providers to determine a user's identifier and/or attributes for making access control decisions and providing a personalized user experience. SAML metadata, distributed centrally by the federation, identifies the federation members and provides public keys, resource endpoints (URLs), and other information about the members that helps identity providers and service providers establish trust and interoperate.

3. APPROACH

Recall that our goal is to enable TeraGrid researchers to use the authentication method of their home organization for access to TeraGrid. We achieve this goal by implementing a federated login capability that leverages the InCommon Federation to provide a bridge from campus authentication to the existing TeraGrid authentication architecture. In this section, we present the details of our developed solution,

[7] `http://www.igtf.net`
[8] `http://ooi.oceanleadership.org`
[9] `http://shibboleth.internet2.edu`
[10] `http://saml.xml.org/saml-specifications`
[11] `http://www.protectnetwork.org`

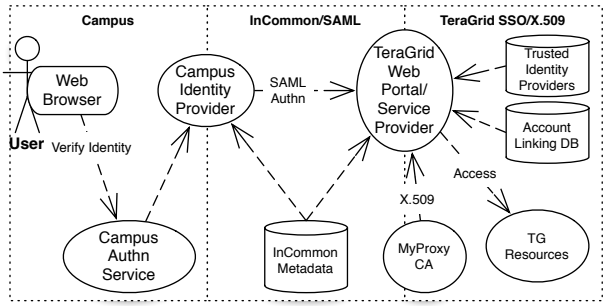


Figure 3: Federated login to TeraGrid relies on translation of credentials between the campus domain, InCommon, and the TeraGrid single sign-on system.

which at its core combines *account linking* and *credential translation*. Our solution builds on the InCommon Federation and existing TeraGrid authentication architecture described in the previous section.

Figure 3 shows a conceptual overview of the credential translation processes. The translation at left between the campus domain and InCommon is handled by Shibboleth or a similar SAML identity provider. The translation at right between InCommon and the existing TeraGrid single sign-on system constitutes our contribution and the focus of this paper. This translation uses the account linking process to bind SAML identities to existing TeraGrid identities.

3.1 Account Linking

The *account linking* process binds the researcher's campus identity, conveyed via InCommon/SAML, to his or her TeraGrid identity, as stored in the TeraGrid Central Database (TGCDB). When the researcher visits the TeraGrid federated login web site, which implements a standard InCommon SAML service provider using the Shibboleth software, he or she sees a prompt to select an InCommon identity provider (i.e., the researcher's home campus) in order to initiate authentication. The Shibboleth software redirects the researcher to the selected identity provider, where the researcher logs in. The identity provider then redirects the researcher back to the TeraGrid site with a SAML authentication assertion, according to the SAML protocols. At this point the account linking component is activated. It first searches the account-link database (actually a table in the existing user database) for an entry matching the researcher's authenticated campus (SAML) identifier. If found, the entry identifies the TeraGrid username linked to that campus identity, allowing the researcher's TeraGrid login to proceed. If no entry is found, the federated login site prompts the researcher for his or her TeraGrid-wide username and password. If the username and password verify (via the TeraGrid Kerberos service), the federated login site creates a new entry in the account-link database linking the TeraGrid account with the campus identity. Then the researcher's TeraGrid login can proceed with that TeraGrid-wide username. When the researcher returns to the site at a later time, the account-link entry will be in place, so the researcher will be able to log in using his or her campus identity without being prompted again for a TeraGrid-wide username and password.

It is important to note that the account linking process does not replace the TeraGrid allocations process. Rather, the account linking process relies on the allocations process for identity vetting and authorization of TeraGrid users. The federated login capability provides only a new authentication method for vetted TeraGrid researchers.

TeraGrid users may link identities from multiple identity providers to their TeraGrid account, allowing researchers associated with multiple research institutions to log in to TeraGrid using whichever identity provider is convenient at the time. However, to avoid account sharing (which is a violation of TeraGrid policy), researchers may link at most one identity from each identity provider with their TeraGrid account. For example, a professor may not link his or her graduate students' campus identities with his or her TeraGrid account. Instead, the TeraGrid policy requires each professor, graduate student, etc., to obtain their own individual TeraGrid account. After login, TeraGrid users may view and delete their account links.

Account links expire one year after creation, at which point the user is required to perform the account linking process again, to re-verify the binding between the user's federated identity and his or her TeraGrid account. This periodic verification of the binding protects against stale or reassigned campus identities (e.g., when a student graduates). When federating with each campus, TeraGrid staff members confirm with the campus operators that campus procedures ensure that identities are never re-assigned within a one year interval.

3.2 Credential Translation

The account linking process facilitates a browser-based, federated login to TeraGrid systems. However, as discussed previously, a significant proportion of TeraGrid use cases and workloads are command-line, complex-workflow, and/or unattended/batch processes, which are not well supported by browser-based authentication (i.e., SAML Web Browser Single Sign-On). So, the TeraGrid federated login employs *credential translation* to convert the browser-based credential to a credential that supports these use cases.

Specifically, the TeraGrid federated login converts the authentication assertion, provided by an InCommon-member identity provider, to an X.509 certificate, provided by a certificate authority (CA) trusted by TeraGrid. TeraGrid has a significant investment in a certificate-based single sign-on infrastructure. Support for certificate-based authentication in remote login (GSISSH), job submission (GRAM), and file transfer (GridFTP) protocols enables today's interactive TeraGrid use cases. Furthermore, proxy certificate delegation [15] enables complex, multi-tier workflows and batch processing in TeraGrid.

Through TeraGrid's federated login capability, TeraGrid researchers can use their campus login to obtain certificates for web and desktop applications. After federated login, the TeraGrid web site presents a menu of options. Researchers can launch remote login and file transfer applets in their browser, authenticating with a certificate loaded into their browser session. Additionally, researchers can launch an application that delivers a certificate to the local filesystem, ready to be used with desktop applications such as those provided by the TeraGrid Client Toolkit. Implementation details are provided in later sections.

In summary, the researcher's federated login to TeraGrid

requires multiple credential translation steps. First, the local campus identity provider translates a local campus credential (such as a Kerberos username and password) to a SAML authentication assertion as specified by InCommon. Then, TeraGrid's federated login system translates the SAML assertion to an X.509 certificate. Finally, TeraGrid resource providers translate the certificate to a local resource login (i.e., a Unix account).

3.3 Trust Establishment

Establishing trust is critical to successfully bridging from campus identity providers to TeraGrid resource providers. Deploying the TeraGrid federated login required negotiation with InCommon members (to release identities to TeraGrid) and accreditation of our CA by IGTF (so the certificates will be accepted by TeraGrid members).

3.3.1 Campus Federation

When TeraGrid became a member of the InCommon Federation, it was not automatically entitled to obtain authentication assertions from InCommon-member identity providers. First, TeraGrid needed to register its federated login service provider with the federation, so its information would be included in the federation metadata, enabling it to be recognized by identity providers. This registration is a lightweight task, requiring only a few minutes of effort.

Following that registration, and of significant effort to arrange, the identity providers need to configure their local policies to release identity information to the TeraGrid's federated login service. Specifically, the federated login service depends on receiving a persistent user identifier from the identity provider via the eduPersonPrincipalName (ePPN) or eduPersonTargetedID (ePTID) attribute defined by the eduPerson specification.[12]

In our effort to have identity providers release ePPNs or ePTIDs to TeraGrid, we encountered three categories of identity providers:

- The first type of identity provider was willing to release ePPNs or ePTIDs to any InCommon-member service provider by default. In this case, after reviewing the published policies of the identity provider, we asked a TeraGrid user associated with that identity provider to help us with testing. After a successful test (i.e., a valid assertion with ePPN or ePTID was received), we added that identity provider to the supported list.

- The second type of identity provider was willing to release ePPNs or ePTIDs on request. In this case, we sent email to the contact address found in InCommon Federation metadata, explaining our application and requesting the needed attribute. Once we received a reply that our request was approved, we proceeded with testing as in the first case.

- The third type of identity provider required local sponsorship and review of our request. In this case, we sent a list of TeraGrid PIs affiliated with the institution to the identity provider contact and worked with them to identify sponsors and follow the local approval process. For some of these campuses, the review is still in progress or stalled.

[12] `http://middleware.internet2.edu/eduperson`

Since federating with campuses was a manual, campus-by-campus process, and there is no method to discern what behavior a campus would present until they were engaged, we focused our efforts on campuses with over 50 TeraGrid users. Of the 38 target institutions, 24 (67%) were InCommon members. To date, we have successfully federated with 16 of those. We have also federated by request with 11 additional campuses outside our initial target list, bringing our current total number of supported campuses to 27.

3.3.2 PKI Federation

Translating SAML authentication assertions from InCommon members to certificates accepted by TeraGrid resource providers and peer grids required us to deploy a certificate authority (CA) and obtain accreditation of the CA from the International Grid Trust Federation (IGTF), to satisfy TeraGrid Security Working Group policies. The IGTF consists of three regional Policy Management Authorities (PMAs). The Americas Grid PMA (TAGPMA)[13] covers the U.S. region.

Worldwide participation in the IGTF ensures that certificates issued by accredited CAs can be accepted by TeraGrid and peer grids around the world. While today's academic SAML federations are national in scope, with limited international inter-federation, translating SAML assertions to internationally accepted certificates supports international science projects such as the Worldwide Large Hadron Collider Computing Grid (WLCG).[14]

The IGTF currently supports accreditation under three CA profiles: Classic, Member Integrated Credential Services (MICS), and Short-Lived Credential Services (SLCS).[15] For Classic CAs, subscriber identity vetting is performed by registration authority (RA) staff persons. In contrast, MICS and SLCS CAs leverage an existing identity management system for vetting certificate requests. We pursued accreditation for our federated CA under the SLCS profile, since our CA leverages the TeraGrid Central Database and identity providers in the InCommon Federation.

SLCS CAs issue short-lived certificates. The short certificate lifetime acts as a countermeasure against credential theft and misuse. The maximum lifetime of one million seconds (or about twelve days) was determined through a requirements-gathering process in the Global Grid Forum [12] and was later incorporated into the SLCS profile.

IGTF profiles require that CAs operate according to community standards. Each CA must publish a Certificate Policy and Certification Practices Statement (CP/CPS) according to RFC 3647 [7]. NCSA's CP/CPS documents are published on the NCSA CA web site.[16] Certificates and Certificate Revocation Lists (CRLs) must conform to RFC 5280 [8] and the Open Grid Forum Grid Certificate Profile [10]. Additionally, since SLCS CAs are online and automated, and therefore subject to network-based attacks, the SLCS profile requires that the CA private key be protected in a FIPS 140 level 2 rated hardware security module [13].

The TAGPMA review process includes a presentation to the TAGPMA membership at a regularly scheduled meeting and a checklist-based review of the CA's policies and operations, followed by a vote for acceptance by the TAGPMA

[13] `http://www.tagpma.org`
[14] `http://lcg.web.cern.ch`
[15] `http://www.tagpma.org/authn_profiles`
[16] `http://ca.ncsa.uiuc.edu`

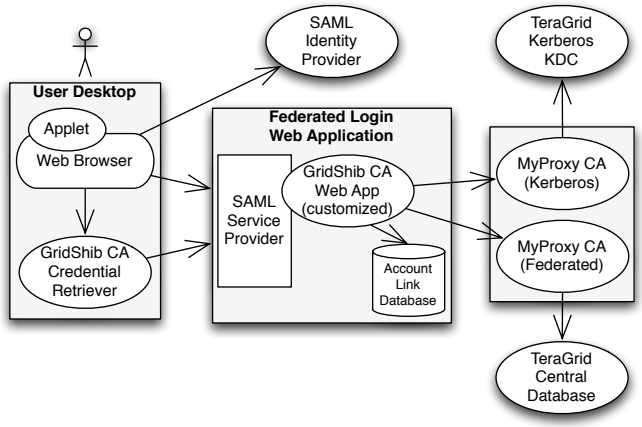


Figure 4: The TeraGrid federated login system provides certificates, issued by a MyProxy CA, for web and desktop applications. The web application binds campus identities to TeraGrid identities via an account-link database.

membership. NCSA began the TAGPMA review process for the federated CA in March 2009 and obtained certification in May 2009. NCSA has been a TAGPMA member since 2005, and this was our third CA to be accredited via the TAGPMA process. Approved CAs are included in the IGTF CA distribution, as well as the TERENA Academic CA Repository (TACAR).[17]

3.4 System Architecture

Figure 4 presents the components of the TeraGrid federated login system. The federated login web application is a SAML service provider, which consumes SAML authentication assertions from InCommon-member identity providers, via the Shibboleth software implementation. The web application has a local PostgreSQL database that stores the account linking information. We decided to (initially) maintain this information in a local database separate from the TeraGrid Central Database (TGCDB), to obtain local database performance and simplify the initial implementation. However, we plan to migrate it to the TGCDB (also PostgreSQL) when we integrate the federation functionality with the TeraGrid User Portal (see Section 6.1).

The web application interacts with two MyProxy CA instances (via the simple MyProxy protocol [2]) for verifying TeraGrid passwords and obtaining short-lived certificates. The first MyProxy CA instance was already in existence (certified by TAGPMA in March 2007) serving TeraGrid single sign-on. It verifies the user's TeraGrid-wide username and password and issues short-lived certificates. In the federated login application, we use this MyProxy instance to verify TeraGrid (Kerberos) passwords at account linking time. Since the web application already contained MyProxy client libraries, using the MyProxy interface to Kerberos rather than interacting with Kerberos directly simplified the web application. The second MyProxy CA instance is the new federated CA, certified by TAGPMA in May 2009. It issues certificates based on federated login. It trusts the federated login web application to properly validate SAML authentication assertions (using Shibboleth) and map campus identities to TeraGrid usernames. The web application sends the authenticated TeraGrid username to MyProxy, which issues a short-lived certificate corresponding to that username. The web application authenticates to MyProxy using its own trusted certificate. The federated MyProxy instance will only accept requests properly authenticated using that certificate. Both MyProxy instances map TeraGrid usernames to certificate subject distinguished names via the TGCDB.

When the TeraGrid user launches one of the browser applets that require a certificate for authentication to TeraGrid resources, the federated login web application, via the MyProxy API, generates a new RSA keypair associated with the user's web session (via state in the web server referenced by a session cookie) and issues a certificate request containing the RSA public key to MyProxy, which returns a short-lived, signed certificate for the user to the web application. The applets can then access the private key and certificate for authentication on the user's behalf. Similarly, when the TeraGrid user selects the credential retrieval desktop application, the browser downloads and launches the application via Java Web Start [11]. The desktop application then generates a new RSA keypair and issues a certificate request to the web application, which passes it to MyProxy and returns the signed certificate to the desktop application, which writes the certificate and private key to the filesystem for access by TeraGrid client applications. The credential retrieval application and components of the web application are reused from the GridShib CA software as developed by the GridShib project [18].[18]

3.5 Current Status

The TeraGrid federated login service[19] is in production, supporting logins from 27 institutions. After accreditation by TAGPMA in May 2009, the site entered a friendly-user beta testing period, where we solicited test users from each supported campus to try the service and give their feedback. We announced the service to all TeraGrid researchers via TeraGrid News on September 1, 2009.

As of February 2010, we have 72 entries in the identity-mapping table from 21 (of the 27 available) institutions, and we have issued over 800 certificates. The most popular application is the remote login GSI-SSHTerm applet,[20] followed closely by the credential retrieval desktop application.

4. SECURITY CONSIDERATIONS

Security was a primary consideration throughout the design and deployment of the federated login service. We highlight security considerations of particular interest in this section.

4.1 Trust Architecture

Adding federated identity to the TeraGrid single sign-on model gives rise to two meaningful changes to the trust relationships in the TeraGrid security architecture.

First, the InCommon identity providers add a new set of trusted entities. Identity providers are trusted to correctly

[17] http://www.tacar.org

[18] http://gridshib.globus.org

[19] https://go.teragrid.org

[20] https://sourceforge.net/projects/gsi-sshterm

authenticate users, disallow the reuse of identifiers, and adhere to other basic policies, as discussed in the following section. Identity providers also play a role in incident response as discussed in Section 4.6.

Second, the federated MyProxy CA outsources authentication to the web front-end. In the current TeraGrid User Portal, a user presents a username and password, which are passed to the MyProxy CA for validation before issuance of a credential. In the federated identity model, the web application presents just a username to the MyProxy CA and authenticates using a trusted certificate specific to the web application instead of the user. The MyProxy CA trusts that the web application has done appropriate authentication of the user. This increases the ramifications of a compromised web application.

The MyProxy CA could be modified to require and validate some proof that the web application actually authenticated the user. One way to provide this validation could be to implement SAML delegation.[21] The ShibGrid project [14] modified MyProxy to validate SAML authentication assertions obtained by the web application. While that implementation does not use SAML delegation, it provides some additional protection. This capability could be added to the TeraGrid service, but it would increase the complexity of the solution.

4.2 Peering with Identity Providers

As discussed in Section 3.3.1, federating with campus identity providers is a manual process. Identity providers decide whether they are willing to release user identifiers to the TeraGrid service. Likewise, TeraGrid staff members, in their role as administrators of the federation service, decide whether to peer with a given campus identity provider. The federated login service is explicitly configured with a list of trusted identity providers (i.e., not all InCommon-member identity providers are automatically accepted). Our review process confirms that the identity provider: (1) serves TeraGrid users; (2) is operated by a known and respected organization; and (3) operates a trustworthy authentication service and provides globally-unique and non-reassigned identifiers, so that subscribers are uniquely identified.

So far, the issue of identifier re-assignment has blocked us from peering with a few campus identity providers. Our annual verification process allows us to support campuses that re-assign identifiers only after a one year or greater hiatus period. We have found in some cases, campuses will re-assign identifiers more quickly for a subset of their population (e.g., undergraduate students and/or visitors), and we are working with those campuses to identity a method to distinguish between those identities that meet our requirements (i.e., those not re-assigned more quickly than our threshold) and those that don't. InCommon's new Identity Assurance program[22] may help with this issue.

4.3 Disallowing Account Sharing

As discussed in Section 2.1.1, TeraGrid policy forbids account sharing. This policy is primarily for clarity during incident response, since multiple users sharing an account complicates the process of determining if suspect account activity was performed by the authorized account holder or by an unauthorized party using the stolen password of the account holder. To enforce this policy, we allow only one identifier per identity provider to be linked with a particular TeraGrid identity.

4.4 Web Application Security

We use multiple methods in the web front end to protect against web-based attacks. The web front end accepts connections only via HTTPS, which provides certificate-based authentication of the service to the web browser and privacy of network data (including SAML assertions, cookies, and certificate requests). To protect against cross-site request forgery (CSRF) attacks, the GridShib CA software uses standard anti-CSRF mechanisms (cookies and hidden form fields) to ensure that web sessions follow an approved workflow, i.e., requiring the user to always visit the login page before requesting a certificate, so a malicious site can not redirect the user's browser directly to the certificate-request form to force a malicious certificate issuance.

The account-link database is configured to allow only local access, and anonymous read access to the database is disabled. The username and password for accessing the database is stored outside publicly accessible web space, and is readable only by the web server process. This configuration gives the server-side web application read and write access to the database while preventing all client-side web access.

The trusted certificate used to request user certificates from the federated MyProxy CA is stored on the web server outside publicly accessible web space and is readable only by the web server process.

Remote login to the web server is restricted to a small set of remote hosts through the use of an iptables-based firewall. Additionally, SSH access is limited to a small number of administrators, who must log in with a one time password (OTP), e.g., by using a CRYPTOCard token generator.

4.5 MyProxy CA Security

The back-end MyProxy CA is secured according to IGTF standards. The CA private key is protected in FIPS 140 level 2 rated hardware security modules. The servers are located on a dedicated network, behind a hardware firewall with a restrictive policy, with network-based and host-based intrusion detection. The firewall allows network connections to the MyProxy CA instance used by the web application only from the host on which that application resides. System logs are streamed to a dedicated syslog collector host, where they are monitored by the NCSA security team. The CA issues a certificate revocation list (CRL) daily or immediately after any revocation.

4.6 Incident Response

The federated login system architecture provides multiple methods for responding to account compromises and other security incidents. In case a federated identity is deemed suspect, the account link for that identity can be disabled in the account-link database by administrators so it can no longer be used to obtain certificates. In case an identity provider is deemed suspect, it can be removed by an administrator from the list of trusted identity providers so assertions from that provider can no longer be used to log in. Extensive CA logging enables administrators to quickly identify certificates associated with a compromise so they can be revoked.

TeraGrid incident response is coordinated through the se-

[21] `http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-delegation.html`

[22] `http://www.incommonfederation.org/assurance`

curity working group. In response to compromise, TeraGrid resource providers can locally disable accounts, and TeraGrid staff can centrally disable or reset TeraGrid-wide passwords.

InCommon metadata contains operational contact information for each identity provider that TeraGrid security staff can utilize during incident response. Additionally, work is underway in the Committee on Institutional Cooperation[23] Identity Management Taskforce to propose a set of policies and additional available information for incident response in federated identity environments such as InCommon.

Like all IGTF CAs, the federated NCSA CA publishes operational contact information on its home page and in metadata files included in the IGTF CA distribution. The IGTF Risk Assessment Team[24] is available for coordinating response to incidents and vulnerabilities impacting IGTF CAs.

5. LESSONS LEARNED

In this section we discuss some of the lessons learned during the deployment of our solution and establishment of trust with identity providers in InCommon.

5.1 Effort for Trust Establishment

As we described previously in Section 3.3.1, while InCommon defines standard (SAML) profiles for identity and attribute transmission and an automated means of metadata distribution, simply being a member of InCommon as a service provider does not guarantee that any particular identity provider will release user attributes to that service provider. Nor does it provide guarantees about identifier persistence in that ePPN identifiers can be potentially re-issued (e.g., after a student leaves the student's identifier could be re-assigned to a new incoming student).

The process of contacting identity providers to arrange attribute release and establish their policies on identifier reissuance is very time consuming. This manual, campus-by-campus effort will be very difficult to scale to the hundreds of campuses associated with TeraGrid researchers, not to mention the thousands of research institutions in the U.S. from where future TeraGrid users might come.

We look forward to deployment of user-driven attribute release in the InCommon Federation, which would avoid the need for manual policy changes by campus operators. User-driven attribute release, via tools such as uApprove,[25] allows users to review and consent to the release of requested attributes when they access the service.

5.2 Testing

Another complexity encountered during attribute release testing was that the identity provider administrators at campuses were rarely TeraGrid users. This meant that only our end users, who are not generally Shibboleth experts, could test the system from end-to-end, as they were the only ones with accounts at both the identity provider and the TeraGrid. Adding a simple test application that could be used by identity provider operators to more fully test the attribute release process, without needing to have a TeraGrid account, would be a useful addition to this trust establishment procedure.

5.3 Software Issues

A major source of issues during our beta testing period was the lack of constraint as to the contents of eduPersonTargetedID (ePTID) values. We found significant variety in the formatting and character sets of ePTID values across campuses, which clashed with several assumptions in our software:

- The various ePTID values triggered exceptions in the GridShib CA identifier sanitizing routines, which attempted to sanitize data from the identity provider to protect against accidental or malicious string encoding that could cause problems. These routines were too aggressive in removing "invalid characters", thereby corrupting the identifiers, and we were forced to abandon such sanitization.

- There was also an assumption in the original software of the identifiers being usable as filenames to maintain an audit record of issued credentials (a requirement of IGTF accreditation). However, some of the characters were meaningful to the file manipulation routines (e.g., forward slashes which represent a path separator under Unix). Hence the approach of using the ePTID was abandoned and instead we used a hash of the distinguished name with a constrained character set.

- Finally, our web site originally displayed the ePTID value to the user after login. While this approach worked with eduPersonPrincipalName values, which are reasonably similar to users' campus usernames and email addresses, we found that the lengthy ePTID string with its broad range of characters distracted and confused users, who expect to see their friendly campus username.

In summary, we have learned to treat ePTIDs as opaque blobs unsuitable for use as a string representation of an identifier and have strengthened the underlying GridShib CA identifier-handling code to support the full range of ePTID values.

6. FUTURE WORK

We consider this work to be just a first step toward enabling federated login to TeraGrid and other U.S. cyberinfrastructure. We envision the following future work.

6.1 Integration with TeraGrid User Portal

The next step for the TeraGrid effort is to integrate federated login with the TeraGrid User Portal (TGUP). Currently, the federated login site is separate from the TGUP, and the TGUP itself requires login with TeraGrid-wide username and password. Integration with the TGUP will provide a more coherent experience to TeraGrid researchers, as well as make TGUP functionality (such as management of TeraGrid allocations) accessible via federated login.

The TeraGrid project is in the process of integrating the Partnership Online Proposal System (POPS)[26] with the user portal, which opens up the possibility of federated logins

[23] http://www.cic.net
[24] http://tagpma.es.net/wiki/bin/view/IGTF-RAT
[25] http://www.switch.ch/aai/support/tools
[26] https://pops-submit.teragrid.org

for TeraGrid proposal submission, potentially eliminating the need for TeraGrid-specific passwords as described in the following section.

6.2 Eliminating TeraGrid Passwords

The account linking process as described so far requires TeraGrid researchers to log in with their TeraGrid username and password at least once per year to maintain the link with their campus identity. This method provides a transition for existing TeraGrid users from daily use of a TeraGrid-specific password to daily use of campus credentials for TeraGrid access, but it does not entirely obviate the need for TeraGrid-specific passwords.

In the future, we plan to integrate account linking with the TeraGrid allocations process, giving TeraGrid researchers the option of never using a TeraGrid-specific password. In this scenario, TeraGrid researchers would authenticate with their campus identity when submitting a proposal for TeraGrid access. A researcher's campus identity will be linked with the proposal at that point, so if the proposal is accepted and TeraGrid access is granted, the researcher's TeraGrid account will be linked with the campus identity when the TeraGrid account is created.

Likewise, project members to be added to a TeraGrid allocation will first authenticate with their campus identity and register a TeraGrid account linked with that campus identity. Then, the project PI will lookup the prospective member's account and add the member to the TeraGrid project. Thus, PIs and other project members will have their campus identities linked with their TeraGrid accounts when the TeraGrid accounts are created, so researchers will be able to access TeraGrid resources using their campus logins without ever having a TeraGrid-specific password. These linked identities could be re-verified each year as part of the allocations renewal process.

It is an open question whether TeraGrid could ever truly eliminate TeraGrid-specific passwords for all users. While we expect many users would prefer to use a federated login, some users may still desire TeraGrid-specific passwords by preference or special requirements.

6.3 Access Based on Attributes

These is a small amount of access to TeraGrid today that is not based on the peer-review process previously described, but is instead granted to a class or workshop for educational purposes. In theory, this access could be granted based on a user's attribute, namely their membership in the class, if it were asserted by their identity provider. Working with campuses to grant access to TeraGrid resources based on such attributes is another area of future investigation.

6.4 Alternative Authentication Technologies

While InCommon and SAML appear to be the most popular technology for federated identity at the home institutions of most TeraGrid users, other web-based authentication methods such as OpenID[27] are popular in the commercial space. We plan on investigating the support of these technologies in our federation model.

6.5 CILogon

Expanding federated login to other U.S. cyberinfrastructure is another area of future work. Relying on the TeraGrid allocations process for identity vetting restricts the availability of the TeraGrid federated login service to registered TeraGrid users. The CILogon project[28] is deploying a modified version of the TeraGrid federated login service that removes the TeraGrid dependencies. The CILogon Service will directly leverage campus identity vetting for certificate issuance. The InCommon Silver Identity Assurance Profile, which maps to NIST Level of Assurance (LOA) 2 [5], provides identity assertions which meet IGTF SLCS profile requirements [3].

Scaling the CILogon Service to serve the national cyberinfrastructure will be a significant challenge. Federating with thousands of U.S. research institutions will require moving beyond the manual campus-by-campus trust establishment process. Providing a usable method for choosing among thousands of available identity providers for a given login is an unsolved challenge. Certainly today's interfaces, where users select their identity provider from a list, will not scale.

7. RELATED WORK

The two areas of related work we find most relevant to the TeraGrid federated login service are (1) similar efforts to bridge SAML and PKI for grids in Europe and (2) TeraGrid's Science Gateways program.

7.1 European SAML-PKI Bridging Efforts

Many European countries have established national SAML federations, with multiple national-scale efforts to link with PKIs in support of cyberinfrastructure.

In Switzerland, SWITCH operates the SWITCHaai federation[29] deployed by most Swiss universities supporting e-learning, e-conferencing, and document exchange services. The IGTF-accredited SWITCH Short Lived Credential Service (SLCS) issues certificates based on successful authentication at a SWITCHaai identity provider.

In Germany, the IGTF-accredited DFN-SLCS CA[30] issues certificates to users of the DFN-AAI federation[31] of universities, technical colleges, and research organizations in Germany.

In the UK, JANET, the national education and research network, operates the UK Access Management Federation for Education and Research,[32] with over 700 members. The SARoNGS Credential Translation Service [16] issues certificates to users of the UK National Grid Service[33] based on successful authentication in the UK Access Management Federation.

Additionally, the Trans-European Research and Education Networking Association (TERENA) has recently developed the TERENA Certificate Service (TCS),[34] which leverages the national SAML-based federations across Europe to deliver certificates to tens of thousands of grid users. Initial TCS partners include the national grid projects and SAML federations of Denmark, Finland, Netherlands, Norway, and Sweden.

[27] http://openid.net

[28] http://www.cilogon.org

[29] http://www.switch.ch/aa

[30] http://www.pki.dfn.de

[31] https://www.aai.dfn.de

[32] http://www.ukfederation.org.uk

[33] http://www.ngs.ac.uk

[34] https://www.terena.org/activities/tcs

Our work to implement federated login for TeraGrid benefited from the examples provided by these related efforts and discussions in IGTF on lessons learned and best practices for bridging SAML and PKI for grids.

7.2 TeraGrid Science Gateways Program

Considering that our work to deploy federated login for TeraGrid is motivated by the desires to make secure access to TeraGrid more convenient for researchers as well as reduce TeraGrid's identity management burdens (e.g., password resets), we find similar motivations for the security design of the TeraGrid Science Gateway program [4, 17]. TeraGrid science gateways[35] provide community-based access to TeraGrid resources, typically via web portals with custom interfaces and applications for specific science communities. The gateway program is part of TeraGrid's effort to serve the larger science community, while continuing to provide high-end computing services to a smaller number of leading-edge researchers. TeraGrid's gateways are designed to serve orders of magnitude more users than can be supported by TeraGrid's existing accounting procedures.

To achieve this goal, TeraGrid provides community allocations to gateways. Gateway PIs and staff are registered in the TeraGrid Central Database (TGCDB), but the gateways manage their own user registration. Gateways access community accounts on TeraGrid resources, with the gateway taking responsibility for isolating its users from one another, so the TeraGrid resource providers are not burdened with managing orders of magnitude more local accounts. Since TeraGrid's federated login capability is based on TGCDB registration, science gateway users do not benefit directly. However, we hope science gateways will provide their own federated login capability. For one proposal, see [9].

8. ALTERNATIVE APPROACHES

A question often posed is what is needed in order to implement a user authentication solution based entirely on SAML or PKI instead of a SAML to PKI bridge. There are significant components missing for each approach, as we describe in the following subsections, that led us to the bridge approach.

8.1 End-to-End PKI Solution?

The TeraGrid has a PKI solution in place with its existing single sign-on system as described in Section 2.1.2. However, ideally TeraGrid would not need to issue certificates, but instead would rely on certificates issued by the user's home organization, taking advantage of the in-person vetting that is (or at least could be) accomplished by that organization. However, despite some progress, we are seeing very limited deployment of externally usable PKIs at universities, as compared with the number of universities that have joined the InCommon Federation. It is the broad and increasing adoption of InCommon in the organizations representing TeraGrid users that led us to build on it, rather than any technical aspect of the SAML technology.

Note that users with credentials from trusted certificate authorities at universities that do operate a PKI can bind, through existing mechanisms in the TeraGrid User Portal, the identity asserted by those credentials to their existing TeraGrid account and access the TeraGrid with those credentials. In order for such certificate authorities to be considered trusted by the TeraGrid they must have achieved accreditation by the International Grid Trust Federation as described in Section 2.1.2.

[35] `http://www.teragrid.org/gateways`

8.2 End-to-End SAML Solution?

To replace the PKI currently in use for single sign-on in the TeraGrid today would not only require that TeraGrid modify a large software deployment base, but would also require addressing functional limitations in SAML, namely:

- Support for clients other than web browsers. Many of the science applications supported by TeraGrid involve desktop applications rather than or in addition to web browsers.
- Delegation support. Our architecture supports authentication on behalf of the user by the web application. It also supports authentication by unattended processes, for example, when the initiating user is offline. (SAML delegation may address this requirement.)
- International federation support. SAML federations have not (yet) reached the global scope of the International Grid Trust Federation as needed to support large grid applications.

Until these issues are addressed, we do not envision a migration away from PKI to be a practical option for TeraGrid.

9. CONCLUSION

In conclusion, we have presented TeraGrid's new federated login capability, which enables TeraGrid users to authenticate using their home organization credentials for secure access to high performance computers, data resources, and high-end experimental facilities. This capability binds campus identities to TeraGrid identities (via *account linking*) and issues certificates based on SAML assertions (via *credential translation*). It is the first effort to leverage federated authentication for access to national-scale research cyberinfrastructure in the United States.

It is our opinion that the world is unlikely to ever settle on a single authentication technology, due to varied technical requirements, as well as significant social and economic issues. Therefore, we believe that the bridging approach described in this article is not simply a short-term hack, but rather an approach that will continue to be required and further refined over time.

10. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 0503697.

11. REFERENCES

[1] T. Barton, J. Basney, T. Freeman, T. Scavo, F. Siebenlist, V. Welch, R. Ananthakrishnan, B. Baker, M. Goode, and K. Keahey. Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, Gridshib, and MyProxy. In *Proceedings of the 5th Annual PKI R&D Workshop*, April 2006.

[2] J. Basney. MyProxy Protocol. Global Grid Forum GFD-E.54, November 2005.

[3] J. Basney. Mapping InCommon Bronze and Silver Identity Assurance Profiles to TAGPMA SLCS Requirements, March 2009. `http://sl.cilogon.org/incommon-slcs-map.pdf`.

[4] J. Basney, S. Martin, J. Navarro, M. Pierce, T. Scavo, L. Strand, T. Uram, N. Wilkins-Diehr, W. Wu, and C. Youn. The Problem Solving Environments of TeraGrid, Science Gateways, and the Intersection of the Two. *IEEE International Conference on eScience*, pages 725–734, 2008.

[5] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic Authentication Guideline. NIST Special Publication 800-63, April 2006.

[6] S. Chan and M. Andrews. Simplifying Public Key Credential Management Through Online Certificate Authorities and PAM. In *Proceedings of the 5th Annual PKI R&D Workshop*, April 2006.

[7] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu. Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework. IETF RFC 3647, November 2003.

[8] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF RFC 5280, May 2008.

[9] T. Fleury, Y. Liu, T. Scavo, and V. Welch. A Web Browser SSO Model for Science Gateways. In *Proceedings of the 2009 TeraGrid Conference*, June 2009.

[10] D. Groep, M. Helm, J. Jensen, M. Sova, S. Rea, R. Karlsen-Masur, U. Epting, and M. Jones. Grid Certificate Profile. Open Grid Forum GFD-C.125, March 2008.

[11] A. Herrick. Java Network Launching Protocol & API Specification. JSR-56, 2005.

[12] S. Mullen, M. Crawford, M. Lorch, and D. Skow. Site Requirements for Grid Authentication, Authorization and Accounting. Global Grid Forum GFD-I.032, October 2004.

[13] NIST. Security Requirements for Cryptographic Modules. Federal Information Processing Standards (FIPS) Publication 140-2, May 2001.

[14] D. Spence, N. Geddes, J. Jensen, A. Richards, M. Viljoen, A. Martin, M. Dovey, M. Norman, K. Tang, A. Trefethen, D. Wallom, R. Allan, and D. Meredith. ShibGrid: Shibboleth Access for the UK National Grid Service. In *Proceedings of the International Conference on e-Science and Grid Computing*, December 2006.

[15] S. Tuecke, V. Welch, D. Engert, L. Pearlman, and M. Thompson. Internet X.509 Public Key Infrastructure Proxy Certificate Profile. IETF RFC 3820, June 2004.

[16] X. D. Wang, M. Jones, J. Jensen, A. Richards, D. Wallom, T. Ma, R. Frank, D. Spence, S. Young, C. Devereux, and N. Geddes. Shibboleth Access for Resources on the National Grid Service (SARoNGS). *International Symposium on Information Assurance and Security*, 2:338–341, 2009.

[17] V. Welch, J. Barlow, J. Basney, D. Marcusiu, and N. Wilkins-Diehr. A AAAA model to support science gateways with community accounts. *Concurrency and Computation: Practice and Experience*, 19(6):893–904, 2007.

[18] V. Welch, T. Barton, K. Keahey, and F. Siebenlist. Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration. In *Proceedings of the 4th Annual PKI R&D Workshop*, April 2005.

Federated Login to TeraGrid

Jim Basney
Terry Fleury
Von Welch

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

This material is based upon work supported by the National Science Foundation under Grant No. 0503697

Goal

- Enable researchers to use the authentication method of their home organization for access to TeraGrid
  - Researchers don't need to use TeraGrid-specific credentials
  - Avoid distribution of TeraGrid-specific passwords
  - Avoid TeraGrid password reset requests
  - Better integrate TeraGrid with campus resources
  - Provision TeraGrid resources according to campus-based identity vetting and authorization

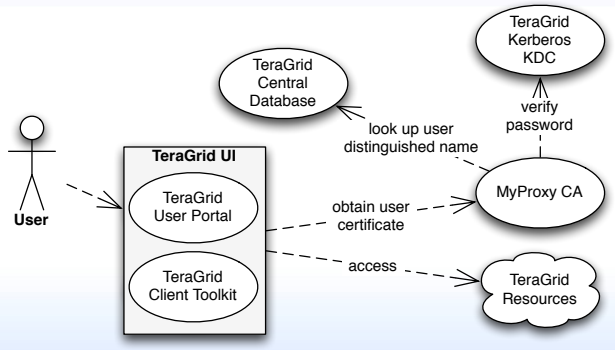
Challenges

- Support TeraGrid usage models
  - Interactive browser and command-line access
  - Multi-stage, unattended batch workflows
- Establish trust among campuses, TeraGrid members, and peer grids (OSG, EGEE)

TeraGrid Allocations

- Resources allocated by peer review
- Project principal investigators add user accounts via the User Portal
- Central Database (TGCDB) contains records for all users
- TeraGrid-wide username and password assigned to every user

TeraGrid Single Sign-On



TeraGrid PKI

- TeraGrid PKI consists of CAs operated by TeraGrid member institutions and other partners
- TeraGrid resource providers trust a consistent set of Cas
  - Provides consistent experience for users
  - Determined by consensus through Security Working Group
  - CAs accredited by International Grid Trust Federation (IGTF)

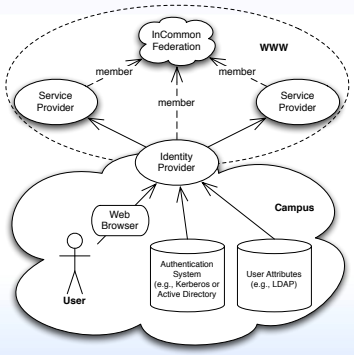
IGTF
International Grid Trust Federation
AP | EU | TAG

InCommon Federation

- InCommon facilitates use of campus identity with external service providers
  - By supporting adoption of standard mechanisms and policies
  - By distributing metadata that identifies members
- Uses SAML Web Browser Single Sign-On protocols
  - Shibboleth implementation from Internet2
  - Work well for browser-based applications, but not command-line or batch workflows
- InCommon represents >200 institutions (>4m users)
  - Of 38 institutions with over 50 TG users, 24 (67%) are currently InCommon members

InCommon

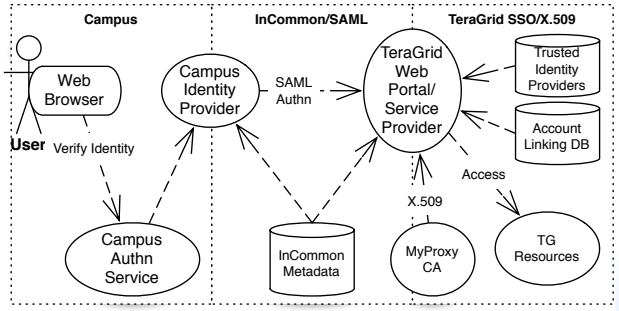
InCommon Federation



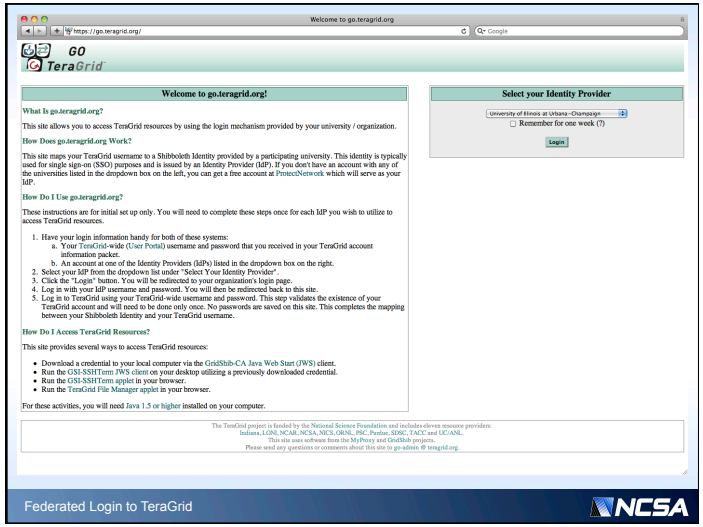
Our Approach

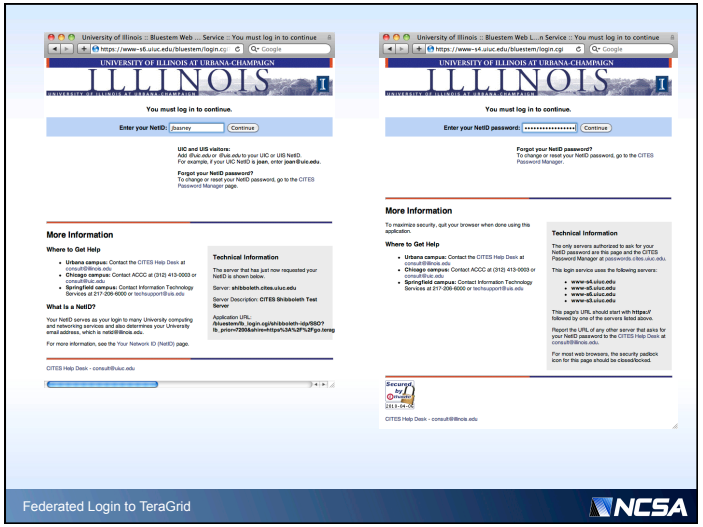
- Account Linking
  - Bind the researcher's campus identity (conveyed via InCommon/ SAML) to his/her existing TeraGrid identity (TGCDB)
    - InCommon motivates our use of SAML
  - Rely on the existing TeraGrid allocations process for identity vetting and authorization
  - Rely on campus for authentication of a persistent user identifier
- Credential Translation
  - Convert from a browser-based (SAML) credential to a certificate for command-line, workflow, and batch processes
    - Deliver certificate to desktop and web session
  - Rely on the existing TeraGrid PKI
    - Adding a new certificate authority

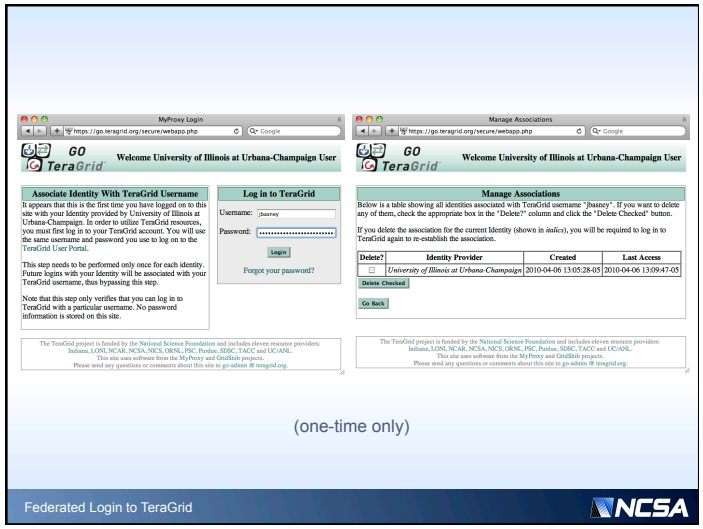
Our Approach

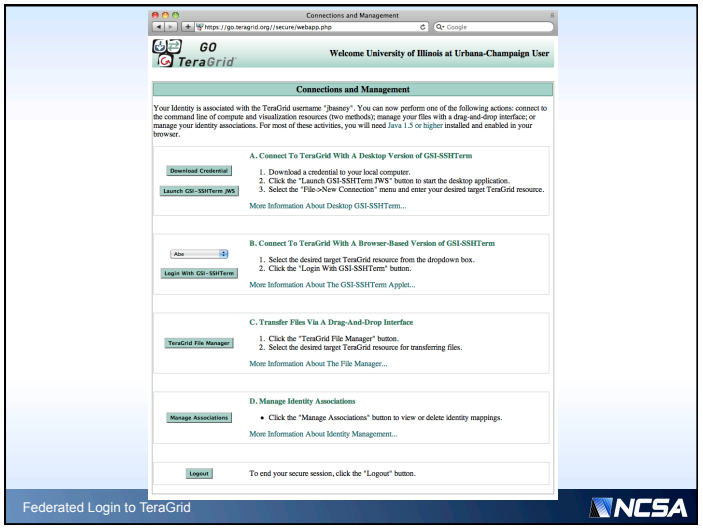


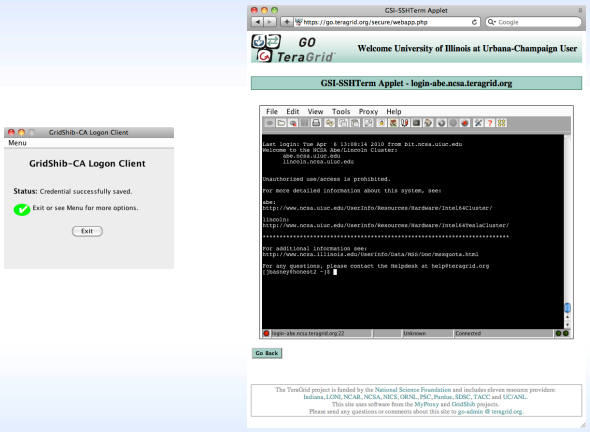
User Experience


Welcome to go.teragrid.org
Welcome to go.teragrid.org!
What Is go.teragrid.org?
How Does go.teragrid.org Work?
How Do I Use go.teragrid.org?
How Do I Access TeraGrid Resources?
Select your Identity Provider
Federated Login to TeraGrid
NCSA

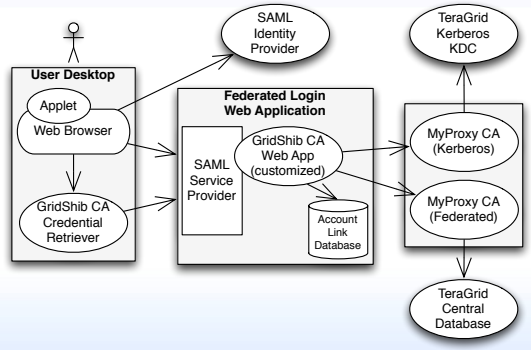

You must log in to continue.
More Information
Federated Login to TeraGrid
NCSA


MyProxy Login
Welcome University of Illinois at Urbana-Champaign User
Associate Identity With TeraGrid Username
Log in to TeraGrid
Manage Associations
Welcome University of Illinois at Urbana-Champaign User
(one-time only)
Federated Login to TeraGrid
NCSA


Connections and Management
Welcome University of Illinois at Urbana-Champaign User
Connections and Management
A. Connect To TeraGrid With A Desktop Version of GSI-SSHTerm
B. Connect To TeraGrid With A Browser-Based Version of GSI-SSHTerm
C. Transfer Files Via A Drag-And-Drop Interface
D. Manage Identity Associations
Federated Login to TeraGrid
NCSA



TeraGrid Federated Login System



Trust Establishment

- Campus and InCommon
- TeraGrid PKI

Trust Establishment Process: Campus

- Join the InCommon Federation
- Add service provider to InCommon metadata
- Request identity providers to release identity information (a manual, campus-by-campus process)
  - Some released identifiers automatically to all InCommon members
  - Some released identifiers on email request
  - Some required local sponsorship and review
- Current status:
  - Targeted 38 campuses with over 50 TeraGrid users
  - 24 (67%) are InCommon members
  - 16 (of the 24) successfully federated to-date
  - 11 additional campuses federated outside the target list

Trust Establishment Process: PKI

- Publish Certificate Policy and Certification Practices Statement (CP/CPS) according to RFC 3647
- Present CA to regional IGTF policy management authority – The Americas Grid PMA (TAGPMA)
- Checklist-based review by TAGPMA of CA's policies and operations
- Vote for acceptance by TAGPMA members
- Current status:
  - Submitted to TAGPMA (March 2009)
  - Approved by TAGPMA (May 2009)
- CA certificate included in TERENA Academic CA Repository (TACAR)

Security Considerations

Security Considerations

- Changes to TeraGrid trust architecture
  - Adding InCommon identity providers as trusted entities
  - Adding web authentication as a trusted method
- Peering with identity providers (IdPs)
  - IdP decides whether to release identifiers to TeraGrid
  - TeraGrid decides to accept IdP assertions – review includes:
    - IdP serves TeraGrid users
    - IdP is operated by a known and respected organization
    - IdP operates a trustworthy authentication service
    - IdP provides globally-unique and non-reassigned identifiers

Security Considerations

- Web application security
  - Use HTTPS for privacy and authentication
  - Cross-Site Request Forgery (CSRF) attack protections (cookies and hidden form fields)
  - Locked down servers (firewalls, OTP for admin access, etc.)
- CA security
  - FIPS 140 level 2 rated hardware security modules
  - Locked down servers

Security Considerations

- Disallowing account sharing
  - Account sharing complicates incident response
  - Allow only one identifier per identity provider to be linked with a given TeraGrid identity
- Incident response
  - Actions may include:
    - Disable account links
    - Disable identity provider trust
    - Revoke certificates
  - Coordinate response with TeraGrid security working group, InCommon, and IGTF

Related Work

- Federated CAs (some accredited by IGTF) in Europe:
  - Switzerland: SWITCH SLCS CA for SWITCHaai federation
  - Germany: DFN-SLCS CA for DFN-AAI federation
  - UK: SARoNGS Credential Translation Service for UK Access Management federation
  - TERENA Certificate Service for national federations (Denmark, Finland, Netherlands, Norway, Sweden, and more)
- TeraGrid Science Gateways
  - Web-based community access to TeraGrid resources
  - Gateways manage their own user registration and authentication
    - May independently support federated login

Status

- In production at https://go.teragrid.org since Sep 2009
  - Supporting logins from 27 institutions
  - Issued >800 certificates so far
- Work in progress:
  - Integrate with TeraGrid User Portal (https://portal.teragrid.org)
  - CILogon Project (www.cilogon.org)
    - Provide certificates to all InCommon members (not just TeraGrid users)
- Other possible future work for TeraGrid:
  - Phase out TeraGrid passwords
  - Attribute-based authorization
  - Support for OpenID

- Questions? Comments?
- Contact: jbasney@illinois.edu

An Identity Provider to manage Reliable Digital Identities for SOA and the Web

Ivonne Thomas
Hasso-Plattner-Institute for IT-Systems Engineering
Prof.-Dr.-Helmert-Str. 2-3
D-14482 Potsdam
ivonne.thomas@hpi.uni-potsdam.de

Christoph Meinel
Hasso-Plattner-Institute for IT-Systems Engineering
Prof.-Dr.-Helmert-Str. 2-3
D-14482 Potsdam
meinel@hpi.uni-potsdam.de

ABSTRACT

In this paper, we describe the implementation of our identity provider, based on open web service standards, which has been extended to distinguish between different qualities of identity attributes; therefore enabling a relying party to distinguish between verified and unverified digital identities.

Our contribution is the definition and representation of identity meta information for identity attributes on the identity provider side and the conveyance of this information as Identity Attribute Context Classes to a relying party. As a main result, we propose a format and semantic to include identity attribute meta information into security token which are sent from the identity provider to a relying party in addition to the attribute value itself.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*

General Terms

Security

Keywords

SOA Security, Identity Management, Identity Provider, Attribute Management

1. INTRODUCTION

Digital Identity Management broadly refers to the establishment and controlled use of a persons "real-life" identity as digital identities in computer networks. Looking at the current online world, performing transactions as online banking, online shopping or communicating in social networks has become an inherent part of life. Hereby, personal, identity-related data plays a major role, since for many activities a service provider requires details about the identity of a user, be it to offer personalized services or to hold it liable in case anything bad happens. Examples include: the purchase of a good, that requires payment and delivery, or the provision of tailored recommendations based on the history of past purchases.

A digital identity usually comprises a limited set of attributes of a "real-life identity" that characterizes this entity (cf. also [23] or [7]). Unfortunately, managing numerous digital identities and associated authentication credentials is cumbersome for most computer users. Users do not only have difficulties to remember their passwords, they also bear a great burden to keep their account information up-to-date.

To overcome the limitations of the closed domain, open identity management models emerged as a way of sharing identity information across several trust domains in a controlled manner. The basic idea is having several places to manage a user's identity data (so called identity providers) and to exchange identity attributes between entities holding identity information (the identity providers) and those consuming it (the relying parties). Open protocols and standards exists to exchange identity attributes as security tokens between identity providers and relying parties (cf. e.g. OASIS Identity Metasystem Interoperability specification 1.0 [19]).

Nevertheless, when we look at the Internet today, we still find an environment of mostly isolated domains. The reasons for the pre-dominance of the isolated model are comprehensible. Isolation allows organizations to retain control over their identity management systems. As organizations usually have different legal and technical requirements for identity management, they find it difficult to give up this control.

However, with regard to the Internet, we can find many identity attributes which do not require strong verification. Often the user can enter information into his account which does not require any verification. It really depends on what a digital identity is used for. If the user logs on to a site to prove on repeat visits that it is the same user, it does not matter whether his digital identity matches with his "real-life identity" as long as it is always the same digital identity he uses to log on. Only if critical transactions are performed, as ordering an item or paying for a service, the integrity of provided user data is required to hold the user liable in case anything bad happens. Current approaches for sharing identity data between domains as proposed by the open identity management models mainly considers the attribute value itself, but hardly how this value was collected or whether any

IDtrust '10, April 13-15, 2010, Gaithersburg, MD

verification process took place.

In order to enable service providers to rely on information from a foreign source, an identity management for the Internet should be able to deal with attributes with a strong verification besides attributes without any verification which are managed by the users themselves. Moreover, it should allow a relying party (such as a service) to assess the value of received identity information in terms of correctness and integrity.

In Thomas et al. [22], we argued that this assessment should be done on the granularity level of the identity data – meaning, that the decision to trust should not only be made between the issuing and the relying party on a all-comprising level, but for each identity attribute, which is exchanged, separately. To give an example, we could consider a university which is trusted to make right assertions about whether a user is a student, but not about whether this user pays its telephone bills.

In this paper, we concentrate on the information required in addition to the attribute value itself to make right assertion about the credibility of an identity attribute. This meta identity information is all information additionally to the attribute value itself which enables a relying party to decide whether it trusts the received value with regard to an intended transaction. To be specific, we provide an identity provider which

- is based on open web service standards, such as WS-Trust, SAML and WS-Metadata-Exchange
- allows the definition of identity meta data and
- conveys identity meta data as so called Attribute Context Classes in SAML security tokens to a relying party

The rest of this paper is structured as follows. Section 2 shows how a scenario could look like which opens up current identity islands by using identity information from many sources across the Internet. In Section 3 we lay some foundations by giving a short introduction to claim-based identity management and the Identity Metasystem. It follows an overview of related work in the area of assurance frameworks as well as a discussion of their limitations in Section 4. After this, Section 5 introduces the trust model, that we use to identify and classify identity meta data that a relying party requires to assess identity information from a foreign source. Section 6 describes the implementation of our identity provider with regard to the definition and exchange of meta data between independent trust domains. In the centre of this section is our extension to the SAML 2.0 token format to convey meta information as part of the security token. Finally, Section 7 concludes the paper and highlights future work.

2. MOTIVATING EXAMPLE

Basically, we can make two observations with regard to the storage and administration of identity information on the Internet, today. The first observation is that basically every service provider on the Internet manages information which is specific to its domain, namely the information which was created during the interaction with a customer and the system, such as a customer number. A second observation is that information stored in independent domains is often redundant, because certain pieces of a subject's identity are required by every service or web site provider. Examples include: the name and address of a person or its birthday. Hence, basically every service or web site provider has identity information, i.e. information about its user's digital identities, which he could provide to other participants (given the user's consent) and basically every service or web site provider also consumes certain information which it requests from the user and which it does not necessarily need to manage itself. A possible solution towards a more effective management of identity information is demonstrated in Figure 1. Instead of entering the same information into different user accounts, the user could reference to another account which already contains this information. For example, the newspaper publisher would receive the assertion that its customer is a student directly from the users university and the information about the user's banking account information directly from the bank.

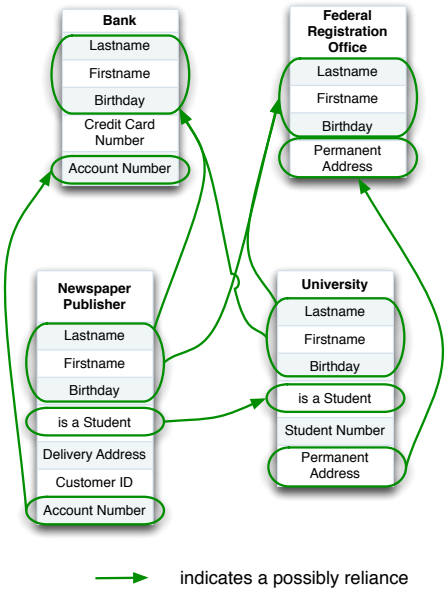


Figure 1: Usecase showing independent identity domains and potential reliance on other Identity Providers

3. BACKGROUND

3.1 Claim-based Identity Management

In order to implement a scenario such as introduced in Section 2, identity management concepts are required that take the decentralized nature of the Internet into account. Open identity management models evolved to address exactly this requirement. Instead of having isolated identity silos as with the traditional approaches, open identity management models are based on the idea of having several places to manage a users identity data (so called identity providers) and to share the identity information between these places and the places where this information is needed.

A concrete implementation of such an open identity management model offers the claim-based identity management. Claim-based identity management uses the notion of *claims*

to describe identity attributes. A claim is an identity attribute named with an abstract identifier (e.g. a URI), which applications and services can use to specify the attributes they need as for example a name or a user's address. Given as a URI, claims provide a platform-independent way to present identity information and are well integrated into the open web service standards such as SAML [8], WS-Trust [15] or WS-Policy [6] which can be used to request and exchange identity information as claims.

3.2 The Identity Metasystem

As claim-based identity management provides interoperability among different identity systems, it is also used as one possibility to implement a related concept, the concept of an Identity Metasystem. Identity Metasystems provide an identity layer on top of existing identity systems and promise an easier management of digital identities among the Internet. This layer abstracts from concrete technologies and provides the necessary mechanisms to describe, exchange and distribute identity information across identity management solutions.

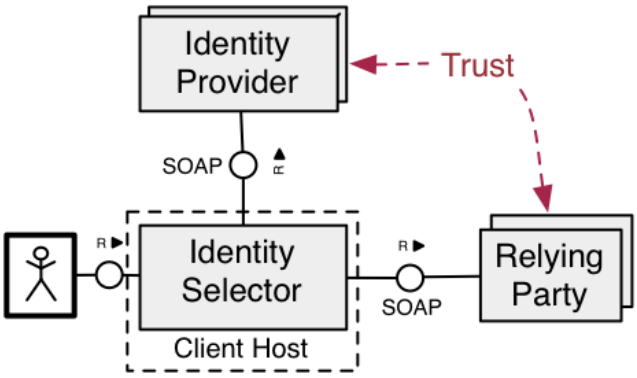


Figure 2: Participants involved in the Identity Metasystem (FMC Block Diagram [12])

To do so, the Identity Metasystem distinguishes three different types of participants as denoted in Figure 2: the consumer of identity information (relying parties), authorities which manage and provide users' digital identities (identity provider) as well as a component to choose a digital identity, called identity selector, and the user. In fact, putting the user in the center of all decision processes regarding his identity and creating a consistent and justifiable user experience belongs to the main principles of Identity Metasystems. These principles which explain sucesses and failures of identity management systems have been written down by Kim Cameron in the *Laws of Identity* [7].

The *relying party* is a service or Web site, which requires a certain set of user attributes / claims to perform a certain action. Instead of managing this information itself, it allows users to authenticate themselves at a federated identity provider and then relies on the assertion issued by this *identity provider*.

An *identity provider* (IdP) holds digital identities of registered users for the purpose of provisioning these identities, or portions of them, to a party willing to rely on this information (the *relying party*). Upon successful registration the identity provider issues a so-called Information Card, which holds all necessary meta data about the interaction between the user and the identity provider, including the URI to contact the IdP, the authentication to the IdP, the claims the IdP can assert as well as supported token types. It is important to note, that Information Cards do not contain any claim values, only the information how to connect to an identity provider to obtain asserted claims as security tokens.

Finally, the identity selector is a piece of software on the user's system which handles the communication between the relying party and the identity provider and provides a consistent user interface to manage Information Cards. Upon request, the identity selector retrieves the policy of the relying party, matches the requirements with the Information Cards of the user and presents the user with a selection of suitable identity providers, from which he can choose. The identity selector takes care of performing the authentication procedure between the user and IdP (e.g. by requesting a password or digital signature) and sends an request for a security token to the identity provider. Upon successful authentication, the identity provider answers with a security token, which the user can use to prove his identity to the relying party.

4. RELATED WORK

The need to trust on information received from a foreign party is inherent to open identity management systems. If a relying party has to rely on identity information received from a foreign party, the need for assurance that the information is reliable is a natural requirement prior to using it. In order to address this need, several initiatives around the world have defined assurance frameworks which cluster trust requirements into different levels of trust. A level of trust or level of assurance (LoA) reflects the degree of confidence that a relying party can assign to the assertions made by another identity provider with respect to a user's identity information.

4.1 Assurance Frameworks

In the area of authentication trust level, the UK Office of the e-Envoy has published a document called "Registration and Authentication – E-Government Strategy Framework Policy and Guideline" [20]. In this document the initial registration process of a person with the system as well as the authentication process for a user's engagement in an e-government transaction are defined. Depending on the severity of consequences that might arise from unauthorized access, four authentication trust levels are defined, reaching from Level 0 for minimal damage up to Level 3 for substantial damage. The more severe the likely consequences, the more confidence in an asserted identity will be required when engaging in a transaction. For example, for filing an income tax return electronically, an authentication trust level of two is needed, which is reached when the client can present a credential (preferable a digital certificate) and can proof his right to that credential, e.g. by signing it with his private key.

The e-Authentication Initiative, another approach, is a major project of the e-government program of the US. The core concept is a federated architecture with multiple e-government applications and credential providers. In order to assist agencies in determining the appropriate level of identity assurance for electronic transactions, the initiative

has published a policy called "E- Authentication Guidance for Federal Agencies" (OMB M-04-04) [10]. The document defines four assurance levels, which are based on the risks associated with an authentication error. The four assurance levels reach from "little or no confidence in the asserted identity" to "very high confidence in the asserted identity".

In order to determine the required level of assurance, a risk assessment is accomplished for each transaction. Hereby, the potential harm and its likelihood of occurrence are identified. The technical requirements that apply for each assurance level are described in a recommendation of the National Institute of Standards and Technology (NIST), which is called "Electronic Authentication Guideline"(NIST 800-63) [17]. This document states specific technical requirements for each of the four levels for the token type, the authentication protocol as well as the types of attacks which need to be prevented.

A quite comprehensive approach that extends the OMB/NIST levels has been proposed by InCommon, a federation of more than 100 members from industry, government and the higher education sector [11]. InCommon uses the Shibboleth specifications and defines an Identity Assurance Assessment Framework. Aspects covered are Business, Policy and Operational Factors, Registration and Identity Proofing, Digital Electronic Credential Technology, Credential Issuance and Management, Security and Management of Authentication Events, Identity Information Management, the Identity Assertion Content as well as the Technical Environment.

Further approaches have been developed as part of the Liberty Alliance project's Identity Assurance Framework [1] as well as in the context of the european Stork project [5].

4.2 Limitations

Current approaches for assurance frameworks as described in the previous section provide a comprehensive assessment for identity providers by defining(gathering) trust requirements with regard to all the processes, technologies, technical infrastructure and further protection in place that have an influence on the degree of confidence into the assertion's contents made by an identity provider. The result is a global trust semantics, which allows a classification of identity providers with respect to different levels of trust. Such a classification can serve as the input to policy frameworks as well as a base for contracts and inter-organizational agreements.

Although current approaches provide a quite comprehensive assessment, a number of limitations exists. Existing assurance frameworks mostly refer to the identity as a whole, but do not refer to trust requirements of specific attributes. It is for example not possible to distinguish between self-asserted attributes an identity provider might manage besides attributes that were verified. Especially with regard to platforms of non-institutional providers such as Facebook, users often prefer using pseudonyms when acting in these communities. In fact, in blogs and forum discussions, anonymity of users is a frequent requirement. Also for over-18-services, anonymity of the users often is in favor while at the same time a verified assertion of a user's age is required. For these purposes, an identity provider could manage self-asserted attributes besides verified attributes. When doing so, reflecting these differences in the assertions is a major requirement.

Also, using existing assurance frameworks, it is hard to reflect possible changes of a user's identity trust level over time. As identity proofing processes are cost-intensive and time-consuming due to the effort required to verify a user's identity attributes, a verification of an attribute might not be desired as long as a user is not involved in transactions that demand a higher trust level. Therefore a user might decide to register with an identity provider without proper identity proofing, having for example his/her name self-asserted and getting involved in the identity proofing only upon concrete requirement. This requires a different trust level per user and does not allow to rate an identity provider as a whole.

Furthermore, identity providers are inherently different due to their affiliation with an organization or institution and might be suitable for asserting certain identity attributes only to a limited extent. For example, a banking identity provider will be in particular suitable to assert that a user can pay for a certain service, but might have weak records of the user's status as a student while for a university's identity provider it would probably be the opposite. In fact, such a diversity of identity provisioning sources is intended in the user-centric model which aims at reflecting the way identities are managed in the real world.

Taking all these facts into account, current approaches are likely to work for federations in which members have similar trust requirements, but are less likely to work when applied to the open market and user-centric models.

In our approach we aim at providing identity meta information for identity attributes in order to allow an identity provider to manage a mix of verified and not-verified attributes and more importantly in order to enable a relying party to distinguish between these different qualities of trust.

4.3 Levels of Assurance for Attributes

Work regarding trust levels for attributes has been conducted by Chadwick et al. in [9]. Chadwick et al. build on NIST's concept of assurance levels. Similar to our work, they propose to have separate metrics for identity proofing processes (expressed in the Registration LOA) and the authentication of a subject (expressed in the Authentication LOA). Authentication LOA and Registration LOA are combined to a Session LOA and sent in each assertion from an identity provider to a service provider. Compared to this, our work is targeted more towards the relying party site. In our work, we aim at providing more choices for a relying party's access control decisions by conveying not only a trust level, but also trust-related information to be evaluated during access control. For this purpose, we propose to extend existing protocols by so called Attribute Context Classes that contain, besides a basic trust level, further meta data to enable the relying party to assess the trustworthiness of the received information.

5. A LAYERED TRUST MODEL

This section presents our trust model used by a relying party such as a service provider to accept identity information from a foreign partner and to perform access control decisions based on the received information. In this model we basically distinguish between two types of trust. First, a trust relationship is required between the service provider and the identity provider in order to trust the correctness of the assertions and second, for a concrete transaction, the

service provider has to decide whether the identity-based information in the assertions are sufficient to reach a certain trust level which is required to perform the request. While in the first case, the trust relationship is of a long-running kind, the trust establishment in the second case is part of identity-based access control mechanisms. We call the first kind of trust, *organizational trust* and the second kind *identity trust*. The following section gives a detailed characterization and comparison.

5.1 The Concept of Organizational Trust

Organizational trust refers to the quality of the trust relationship between the participants of a SOA or web-based scenario. When service consumers and service providers are located within the same trust domain, registration, authentication and management of participants happen under the same administrative control and are, therefore, usually fully trusted. However, with regard to cross-organizational scenarios involving services from different organizations, trust between the participants of a SOA is not given per default. Models for identity management as federated identity management establish cross-organizational trust by setting up federation agreements and contracts to extend the trust domain of an organization to the federation. Having a federation or not, whenever organizational borders are crossed, the question of whether the partner is trusted arises. Factors as past experience, the minimum trust settings for, for example, registration and authentication of users or the reputation of a company are important properties to assess the trustworthiness of the potential business partner. Also, the kind of business relationship is an important factor. A B2B relationship is usually much more trustworthy than a B2C relationship due to contracts which manifest certain obligations and procedures of the business partners. In order to classify different qualities of trust relationships, assurance frameworks exists to help business partners to assess their identity management services. (cf. Section 4). However, a detailed assessment is not always feasible. Sometimes the decision to trust is founded on much fewer assessments. Especially in the user-centric model, a relying party such as an online store might decide to trust an identity provider based on soft criteria as the reputation or global image of the company running the identity service rather than on verifiable facts.

In our trust model, we assume that any kind of assessment has been done by the relying party and led to a classification of identity providers into two (trusted, not trusted) or more levels of trust. It is important to note that this decision is specific to a relying party and can be based on strong contracts, the certification of an identity provider by a trusted authority, past experiences just as any other trust criteria that the service provider regards as appropriate.

5.1.1 Formalism

On an abstract level, we can express the quality of any trust relationship as a mapping from a set of *Trust Criteria (TC)* to a level of trust or level of assurance *(LoA)*:

$$isTrusted_{uni} : (TC_1, ...TC_n) \mapsto LoA$$

This is exactly what assurance frameworks do. Assurance frameworks define a mapping from certain trust criteria to a level of trust, which in almost all frameworks is one of $\{1, 2, 3, 4\}$.

In the trust model underlying our implementation, we use a simplified variant of this function with two trust levels $\{trusted, untrusted\}$. Our trust criteria is the identity provider as a whole (*Issuer*):

$$isTrusted : Issuer \mapsto \{trusted, untrusted\}$$

5.2 The Concept of Identity Trust

Identity trust refers to the trust an entity such as a service provider has into the identity of a subject and its behavior. While the organizational trust level indicates the credibility of the issuer of assertions, the identity trust level indicates the trustworthiness of the subject about which assertions are made. Identity trust is established by credentials that verify properties of the subject. In the claim-based identity management model, these required properties to build up trust (trust requirements) are expressed as claims and exchanged in security tokens. In order to assess the trust into the identity of a subject, such as a user, a relying party needs to assess the received tokens. Hereby, several factors influence the trustworthiness. In order to identify these factors, we use our model of a digital identity.

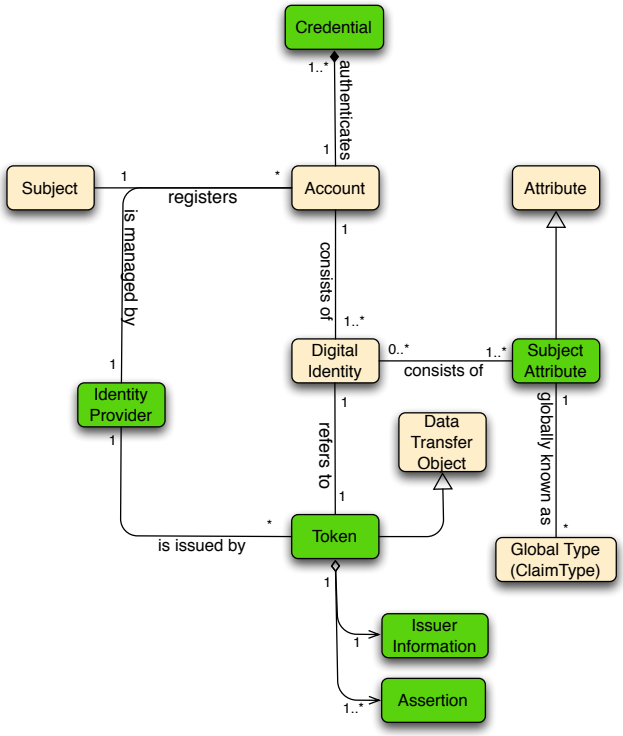


Figure 3: Model of a digital identity based on [13]

Figure 3 shows our model of a digital identity which we extended from Menzel et al.[13]. This model shows the major relationships between the identity provider, the concept of a digital identity, accounts as well as token and authentication credentials. As can be seen from the picture, a digital identity consists of several *Subject Attributes* and is hold in an

Account. Each *Account* can comprise several *Digital Identities*. Using this model, we can identify the aspects that have an influence on the overall trust into an identity. These are (as marked in green):

TC-1 Trust into the authentication process and the subject-to-account mapping.
TC-1 refers to the trust that an identity provider associates a specific subject with the correct record in the identity provider database during an authentication event.

TC-2 Trust into the subject's attributes.
TC-2 refers to the process of identity / attribute proofing and the mechanisms used to verify a specific attribute.

TC-3 Trust into the token.
TC-3 refers to the characteristics of the data transfer between the identity provider and a service provider, e.g. the nature of the token and mechanisms used to protect the token from being forged, replayed or altered.

All these factors are subject to vary between different digital identities of the same or different users within an identity provider. In this case, a relying party needs to check on them per transaction. For example, if an identity provider offers various ways of authentication, the relying party requires to know whether the user typed in a password or presented a signed certificate. The same holds for the subject's attributes. If the process of identity proofing varies between different attributes or different users, a relying party requires to know whether the user presented her/his ID card upon registration or whether the name was self-assigned. Of course, if these factors are static, it is reasonable to consider them as part of the organizational trust relationship as it is usually done in current frameworks.

As we aim in our identity provider to provide digital identities with varying qualities of user attributes, we focus on TC-2 and define a metric on the subject attributes.

5.3 Formalisms

We define *AttributeTrust* to be a function which returns the strength of the attribute proofing process in dependence of the issuer and a certain attribute.

$$AttributeTrust : (Issuer, Attribute) \mapsto AttributeLoA$$

As with the *isTrusted* function defined in 5.1.1, it requires a common semantic of the *AttributeLoA*. Again, it is possible to cluster different trust requirements into levels of assurance. Caution has to be taken as trust requirements usually differ between attribute groups, for example processes to verify a name might be different from processes to verify a membership or the ownership of a specific email-address.

In the trust model underlying our implementation, we use a variant of this function which uses two trust levels with a common semantic $\{verified, unverified\}$ and leave the specifics for each attribute to be checked separately. We define, *isVerified* to be a function which returns whether an identity attribute/ claim was verified by the identity provider.

$$isVerified : (Issuer, Attribute) \mapsto \{verified, unverified\}$$

Depending on the needs, we plan to extend this function in future implementations.

To derive the overall credibility, we combine the results of the functions *isTrusted* and *isVerified*. The way, in which both results are combined shall be defined by a function h, which can be application-specific or globally defined. The function h describes, in which way the fact whether an identity attribute has been verified is combined with the fact whether this has been done by a trusted identity provider. To follow our observation, we would define the credibility of a claim to be 1 only if the claim was verified and issued by a trusted issuer. In all other cases, it is 0. A mathematical definition for h is given below.

$$credibility(issuer, claim) \mapsto h(isTrusted(issuer), isVerified(issuer, claim))$$

with h e.g. defined as

$$h : \{trusted, untrusted\} \times \{verified, unverified\} \mapsto \{1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 1, & \text{if } b_1 = trusted \text{ and } b_2 = verified \\ 0, & otherwise \end{cases}$$

Of course, alternative definitions of h are possible to model other trust behavior. In [22], we give for example the following definition of h which distinguishes three different levels of trust.

$$h : \{trusted, untrusted\} \times \{verified, unverified\} \mapsto \{2, 1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 2, & \text{if } b_1 = trusted \text{ and } b_2 = verified \\ 1, & \text{if } b_1 = untrusted \text{ and } b_2 = verified \\ 0, & \text{otherwise} \end{cases}$$

Please refer to [22] for further details.

5.4 Comparison

Table 1 summarizes the concepts of organizational trust and identity trust and compares them. As Organizational Trust refers to the quality of the trust relationship between organization, it implicitly answers the question: "Can we trust the issuer of a token?". The decision to trust another entity as an identity provider in a SOA or web-based infrastructure, is a decision which is drawn before any messages start flying around. Usually, federation agreements or similar contracts are negotiated and signed when setting up the federation. These decisions are then configured in the infrastructure. As compared to this, identity trust is the trust between the subject of the transaction and the service provider. It is service-call specific and therefore is negotiated each time, a call for a new transaction receives.

6. IMPLEMENTING AN IDENTITY PROVIDER FOR VERIFIED DIGITAL IDENTITIES

This section describes our implementation of a trust-aware claim-based identity provider. The section starts with a short description of the technical and functional characteristics of the existing identity provider. After this, Section 6.2 shows a use case which demonstrates the use of identity meta data in our identity provider. The next sections give

Organizational Trust	Identity Trust
refers to the quality of the trust relationship between organizations	refers to the identity associated with a transaction
Can we trust the issuer of a security token?	Can we trust the subject in the token?
determined out-of-band	determined during service call
configurable	negotiable

Table 1: Comparison of Identity Trust and Organizational Trust

insights into our implementation. We describe, how we defined a data structure to express identity meta data as so called Attribute Context Classes and how we extended the SAML 2.0 assertion specification to send identity meta data as part of security token.

6.1 Existing Identity Provider

This section gives a short overview about our implementation of an identity provider which is in the focus of this paper.

6.1.1 Functional Details

Our prototype is an implementation of an identity provider for service-oriented architectures as well as web applications which features

- a security token service in accordance to the WS-Trust specification 1.3 [16]
- an information card provider based on the specification of SAML 1.1, SAML 2.0 as well as Information Card
- an OpenID Provider according to the OpenID 2.0 Authentication specification [21]

It provides

- security token service functionality including
  - a WS-Meta Data Exchange endpoint to request meta data
  - requesting, issuing and signing of security tokens
  - support for authentication via username token or certificate
- information card provider functionality including
  - issuance of information cards for digital identities
  - creation, editing and deletion of claim types
  - support for various identity selectors
- general identity management system functionality including
  - the creation, editing and deletion of multiple digital identities per user
  - creation, editing and deletion of claims
  - assignment of attributes to digital identities

6.1.2 Technical Details

The prototype is developed in Java utilizing a number of open-source libraries. Most important are Suns web service stack Metro [4] for handling web services and supporting web service security mechanisms such as the security token service, openid4java to provide support of the OpenID 2.0 Authentication protocol [3] as well as maven [2] to provide configuration and deployment options.

A single Web application makes up the prototype, which is deployed and run in Apache Tomcat. The web application offers a web interface as well as a web service-based interface.

6.2 Prototype Use Case

This section describes a small use case which demonstrates the use of identity attributes with different qualities in our identity provider. Figure 4 shows the attribute management page of our identity provider. On this page, a user can manage its identity attributes and assign them to digital identities, which are shown on the right-hand side. As can be seen in Figure 4, a user can have several attributes of the same type such as the *E-Mail Address* or *Given Name*. The type of the identity attribute is mapped to the protocol specific type defined by the protocol which is used to request attributes. In case of Information Card, the type is mapped to the global claim types and in case of OpenID the type refers to the attributes defined by the OpenID community that can be used with OpenID Attribute Exchange (cf. e.g. AXSchema.org). For each stored identity attribute the information whether this attribute has been verified during the collection of the data is shown. Moreover, additional information about the verification process is available as can be seen in Figure 5. Figure 5 shows all available identity meta data for a specific attribute type and for all attributes which have been verified. In this example, the user has registered three different email addresses - two of which are verified and one which is unverified. Looking at the verification details, we find additional information for the two which have been verified. One important information in the meta data is the source of the identity attribute. The source is the entity which provided the data. For example, it is possible that the verification process is the same, but has been performed by different identity providers. One use case that shows the relevance of this is the following: If, for example, an identity provider is federated with another partner and the user decides to link its accounts and to share a certain attribute, so that this attribute is available in both identity providers, the source would indicate the original identity provider that verified that attribute. In case this attribute is issued to another party, the information who verified the attribute will be of interests for the relying party to assess the trustworthiness of the information as the organizational trust might differ between the issuing and the verifying identity provider. As such a federation scenario still bears many open questions, it is due to future work. In our example in the current implementation, the source is in one case an authority (the company itself) and in the other case the user, who has provided the data. In addition to the source, the verification method is detailed in the identity meta data. Again in our example, this is for the first email address, the company itself who is acting as an email provider and in the second case, in which the user had entered the data, a verification email had been sent.

Upon request, this information is sent as part of the se-

curity token to a requesting party. Our demo application to show the use of the identity provider in a complete scenario is a classical web site for an online store selling music files which is shown in Figure 6. To complete the purchase, several personal attributes are requested from the user, such as his name, address and payment information. Furthermore, the music stores requires a valid email address to deliver the purchased mp3 files to. Therefore, once the store receives a security token from the identity provider, it will check whether the provided email address fulfills this requirement.

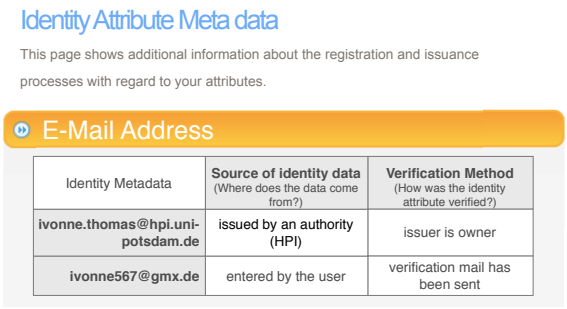


Figure 5: Identity Provider prototype screenshot showing identity meta information for the email address of a user.

6.3 Using Identity Meta Data

This section describes selected implementation aspects with regard to the use of identity meta data in the identity provider.

6.3.1 *Identity and Organizational Trust*

Given the classification into Organizational and Identity Trust as described in Section 5, this section shows its appliance in the identity meta system upon which our prototype implementation is based. As said before, there are three different types of participants in the identity meta system: the identity providers, the relying parties and the clients/users.

A relying party usually specifies a list of identity provider it trusts to make right assertions. Using the notion of claims, the relying party can express for a list of claims the issuer(s) it will accept tokens from. When receiving a security token, the relying party verifies the issuer of the token by checking whether the signature of the token matches the certificate of one of his trusted identity providers. This is in accordance to our notion of Organizational Trust. Only upon correct verification, the relying party will continue with the information in the token.

The information in the token is required to build up identity trust, that is the trust that the requesting user is in fact entitled to access the system. Therefore, the relying party lists the required identity information as claims in its policy. Upon retrieval of this information from a user's identity provider, the relying party checks the value of the identity data with its access control policy and makes an entitlement decision. We store for each claim in accordance of the issuer certain attribute meta data information. This is on one hand the information whether a claim value has been verified (the verification status) and on the other side certain verification details. While the verification status is one of *verified*, *unverified* or *unknown* for all claims, the verification details can differ tremendously between different types of claims. Therefore, we keep the data structure at this point very general and easily adaptable and extensible. The next section goes into detail about this.

In order to model organizational trust, at the moment we simply store for each issuer of security tokens, whether we trust this issuer to make right assertions. As a possible refinement in future work, one could also store certain meta information, which is specific to this issuer, such as identity provider meta information. Such meta information could include, but is not limited to the authentication process supported by an identity provider as well as aspects concerning the storage and management of tokens.

6.3.2 *Attribute Context Classes*

We use so called Attribute Context Classes to define meta data for claims. The notion of Attribute Context Classes has been inspired by a former specification in the SAML community, that is the one of the so-called Authentication Context Classes [18]. Authentication Context Classes are a concept which was introduced in SAML 2.0 and which allows to specify meta data for the authentication used between two parties. As the security of an authentication mechanism depends highly on the values, which characterize such an authentication method, SAML Authentication Classes offer the possibility to describe the authentication process in much more detail. While with SAML 1.1 it was only possible to state that an authentication process was performed using a specific authentication method as, for example, a password, Kerberos or a hardware token, SAML 2.0 now allows to specify how the authentication was performed in addition to the fact that it was performed. This way it is possible to state whether a password with a length of two characters was used or a password with six characters, which was well-chosen and has a limit of three false attempts.

For the identity meta data, we adapted the idea and defined our own data model, which contains the following elements:

- **Attribute Context** This data element holds the attribute context, which is comprised of all additional information to the attribute value itself. This element is the upper container for all identity metadata.
- **Attribute Data Source** This data element indicates the source from which the attribute value was originally received and is part of the Attribute Context. This can be for example another identity provider, some authority as a certificate authority or the user himself who entered the data.
- **Verification Context** This data element holds the verification context, which comprises all information related to the verification of an identity attribute value. The Verification Context is one specific context within the Attribute Context.
- **Verification Status** This data element indicates the verification status of an identity attribute value, which should be one of *verified*, *not verified* or *unknown*. The verification status is part of the verification context.
- **Verification Context Declaration** The verification context declaration holds the verification process details. Such a detail could for example be the method

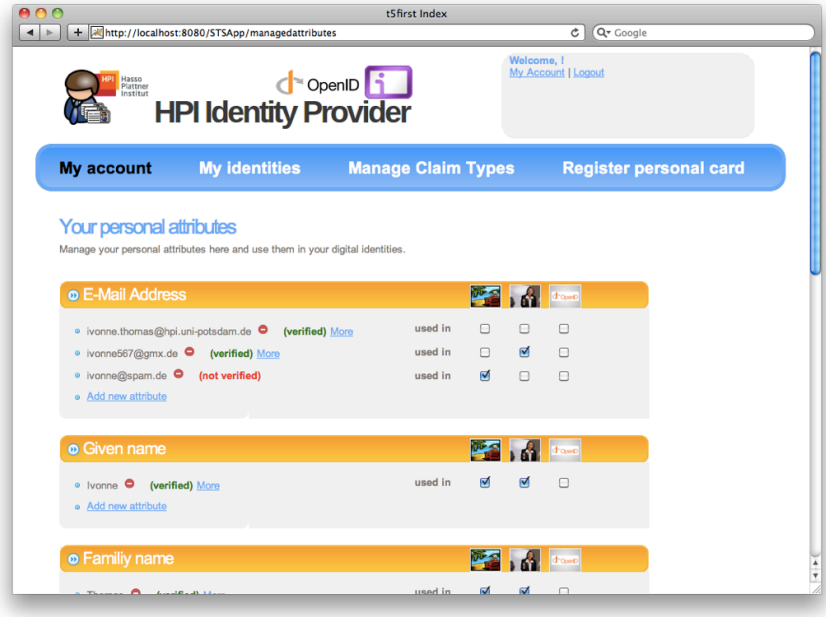


Figure 4: Identity Provider prototype screenshot showing the management of verified and unverified identity attributes.

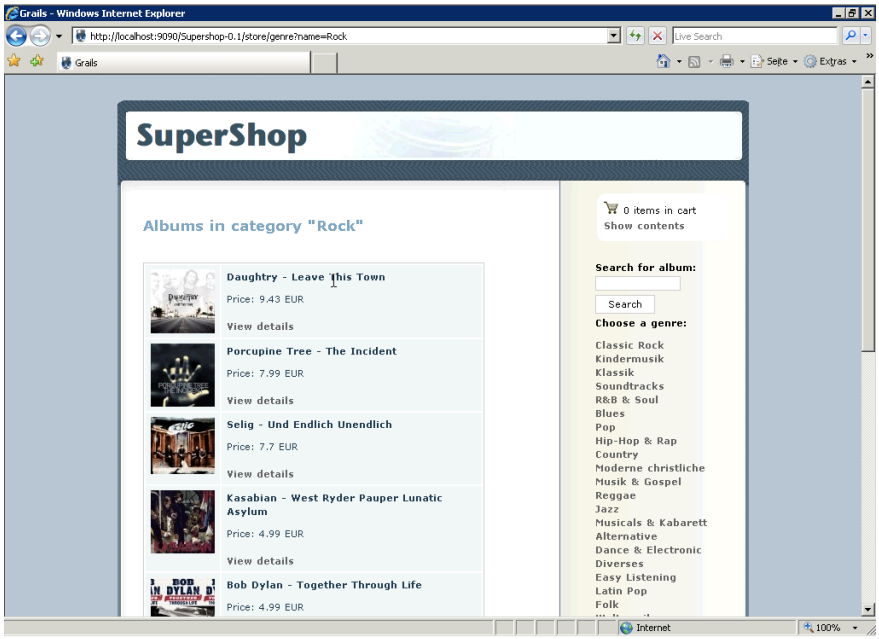


Figure 6: The MP3 Store Scenario: Acting as a relying party of the identity provider.

that has been used for verifying the correctness of the attribute. Further extensions are possible and should be added here. The verification context declaration besides the verification status make up the verification context.

6.4 SAML Attribute Statement Extentions

In order to exchange identity meta data as part of SAML assertions, we introduce extensions to the SAML 2.0 schema. These extensions allow to specify an attribute context to hold further information about an attribute value. The XML schema in Listing 1 presents our extensions, which are defined in a new namespace: *http://de.hpi.ip/saml20/ext*.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://de.hpi.ip/saml20/ext"
  elementFormDefault="qualified">
  <!--
    Extension to
    saml:AttributeStatements to add
    attribute context information
  -->
  <xs:element
    name="AttributeContext">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="AttributeDataSource"
          type="xs:string" />
        <xs:element
          ref="VerificationContext" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element
    name="VerificationContext">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="VerificationStatus"
          type="xs:string" />
        <xs:element
          ref="VerificationContextDecl" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element
    name="VerificationContextDecl">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          name="VerificationMethod"
          type="xs:anyType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 1: XML schema definition of identity meta-data extensions

The root element is the *AttributeContext*, which is added to the complex type *Attribute* of the SAML 2.0 namespace. It is meant to contain all meta information about the attribute value. The attribute value is defined on the same level in the SAML 2.0 type *AttributeValue*. The *AttributeContext* contains the data source of the attribute value as well as a verification context, which is meant to contain all information about the verification of the attribute. This includes the verification status besides further information about the verification process comprised in an element named *VerificationContext* as for example the verification method. The verification method is dependent on the attribute type. Therefore this element can encompass any element structure and is intended to be extended by a suitable data structure to describe an attributes verification. All additional elements are listed in the following with a brief explanation of their meanings:

- **AttributeContext** This element holds the attribute context. This element can be used within the SAML *AttributeStatement* element.
- **AttributeDataSource** This element holds the Attribute Data Source.
- **VerificationContext** This element holds the verification context.
- **VerificationStatus** This element holds the verification status. This element's data type is intentionally defined as a general string to allow possibly extensions later on.
- **VerificationContextDecl** The element holds the verification context declaration.

Listing 2 gives an example that uses the introduced schema. According to our use case described in Section 6.2, the assertion states that the email address of the user is `staff@company.de` and has been verified. The method used for verification is a confirmation email which has been sent to the user.

```
<saml:Assertion
  xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
  xmlns:samlext="http://de.hpi.ip/saml20/ext">
  [...]
  <saml:Subject>
    <saml:NameID>MaxMustermann
    </saml:NameID>
  </saml:Subject>
  [...]
  <saml:AttributeStatement>
    <saml:Attribute
      xmlns:x500="urn:oasis:names:tc:SAML:1.1:
      nameid-format:emailAddress"
      FriendlyName="emailAddress">
      <saml:AttributeValue
        xsi:type="xs:string">staff@company.de
      </saml:AttributeValue>
      <samlext:AttributeContext>
        <samlext:AttributeDataSource>
          user
        </samlext:AttributeDataSource>
        <samlext:VerificationContext>
          <samlext:VerificationStatus>
            verified
          </samlext:VerificationStatus>
          <samlext:VerificationContextDecl>
            <samlext:VerificationMethod
              xmlns:samlextEmail=
              "http://de.hpi.ip/saml20/ext/email">
              <ConfirmationEmailReceived />
```

```
            </samlext:VerificationMethod>
          </samlext:VerificationContextDecl>
        </samlext:VerificationContext>
      </samlext:AttributeContext>
    </saml:Attribute>
  </saml:AttributeStatement>
</saml:Assertion>
```

Listing 2: Example for a SAML security token containing identity meta information.

7. CONCLUSION

Past experiences have shown that there would be no single center to the world of information. In order to get from the isolated model, in which each consumer of identity information manages this information himself to an identity management which takes the decentralized nature of the Internet into account, we argue that consumers of identity information need to be able to assess and distinguish the quality of the information they receive. In particular, with regard to the launch of electronic ID cards as fostered by several european governments, different sources of identity information will have a different quality in terms of correctness and integrity. To have this information integrated into current identity management models is the essence of this paper. Therefore, we defined a data structure to express identity meta data as so called Attribute Context Classes and extended the SAML 2.0 assertion specification to send identity meta data as part of security token. As a proof of concept, we presented an identity provider which is able to manage user-defined digital identities besides verified digital identities. Therefore, for each identity attribute, a so called claim, an attribute context is stored to hold information such as the method of verification that has been used to verify the claim.

As part of future work, we plan to extend the definition of required claims in web service policies by a policy reflecting the additional identity meta data required to assess a claim value.

8. REFERENCES

[1] Liberty Identity Assurance Framework. http://www.projectliberty.org/content/download/4315/28869/file/liberty-identity-assurance-framework-v1.1.pdf, 2007.

[2] Apache Maven. http://maven.apache.org/, 2009.

[3] openid4java - Project Hosting on Google Code. http://code.google.com/p/openid4java/, 2009.

[4] The Sun Metro Web Service Framework. https://metro.dev.java.net/, 2009.

[5] The Stork Project Page. http://www.novay.nl/okb/projects/stork/4561, 2010.

[6] S. Bajaj, D. Box, D. Chappell, F. Curbera, G. Daniels, P. Hallam-Baker, M. Hondo, C. Kaler, D. Langworthy, A. Nadalin, N. Nagaratnam, H. Prafullchandra, C. von Riegen, D. Roth, J. Schlimmer, C. Sharp, J. Shewchuk, A. Vedamuthu, Ümit Yalçinalp, and D. Orchard. Web Services Policy 1.2. Technical report, W3C, http://www.w3.org/Submission/WS-Policy/, April 2006.

[7] K. Cameron. The Laws of Identity, 2005.

[8] S. Cantor, J. Kemp, E. Maler, and R. Philpott. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.02. OASIS Standard Specification, 2005.

[9] D. W. Chadwick and G. Inman. Attribute aggregation in federated identity management. *Computer*, 42:33–40, 2009.

[10] e-Authentication Initiative, US. E-Authentication Guidance for Federal Agencies. http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf, 2007.

[11] InCommon Federation. Identity Assurance Assessment Framework. http://www.incommonfederation.org/docs/assurance/InC_IAAF_1.0_Final.pdf, 2008.

[12] A. Knoepfel, B. Groene, and P. Tabeling. *Fundamental Modeling Concepts*. John Wiley & Sons Ltd, 2005.

[13] M. Menzel and C. Meinel. A security meta-model for service-oriented architectures. *Services Computing, IEEE International Conference on*, 0:251–259, 2009.

[14] Microsoft. Microsoft's Vision for an Identity Metasystem, May 2005.

[15] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, and H. Granqvist. WS-Trust 1.3. OASIS Standard Specification, 2007. OASIS Standard.

[16] A. Nadalin, M. Goodner, M. Gudgin, A. Barbir, and H. Granqvist. WS-Trust 1.3. http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.pdf, 2007. OASIS Standard.

[17] National Institute of Standards and Technology. Electronic Authentication Guideline. http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf, 2006.

[18] OASIS. Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.02 . *OASIS Standard Specification*, March 2005.

[19] OASIS. Identity Metasystem Interoperability Version 1.0 . *OASIS Standard*, July 2009.

[20] Office of the e-Envoy, UK. Registration and Authentication - e-Government Strategy Framework Policy and Guidelines. http://www.cabinetoffice.gov.uk/csia/documents/pdf/RegAndAuthentn0209v3.pdf, 2002.

[21] OpenID Authentication 2.0 - Final Specification. http://openid.net/specs, 2009.

[22] I. Thomas and C. Meinel. Enhancing claim-based identity management by adding a credibility level to the notion of claims. In *SCC '09: Proceedings of the 2009 IEEE International Conference on Services Computing*, pages 243–250, Washington, DC, USA, 2009. IEEE Computer Society.

[23] P. Windley. *Digital Identity*. O'Reilly, 2005.

HPI Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

An Identity Provider to manage Reliable Digital Identities for SOA and the Web

Ivonne Thomas, Prof. Christoph Meinel

Research School on "Service-Oriented Systems Engineering"
Hasso-Plattner-Institute, University of Potsdam

April 2010

Short Introduction

- PhD. Student at the Hasso-Plattner-Institute (HPI), University of Potsdam
- Member of the Research School for Service-Oriented Systems Engineering
- 3rd year
- Research Focus on
  - Security
  - Service-oriented Architectures

Motivation

The law states that southkorean web sites with at least 100,000 daily visitors must force users to register with verifiable real names.

Real Name Policy Act, South Korea

- Very controversial!, BUT:
  - we find different requirements for the reliability of identity attributes in the online world
  - users have verified identities besides anonymous identities
  - user need to decide which identity to use in correspondence with the provider

Identity Assurance Frameworks

- need to trust on information from a foreign party is inherent to open identity management systems!
- basic principle: **cluster trust requirements** into levels of trust
- A level of trust (level of assurance (LoA))
  - reflects the **degree of confidence** that a relying party can assign to the assertions made by another identity provider with respect to a users identity information
- Several initiatives have formed and proposed approaches

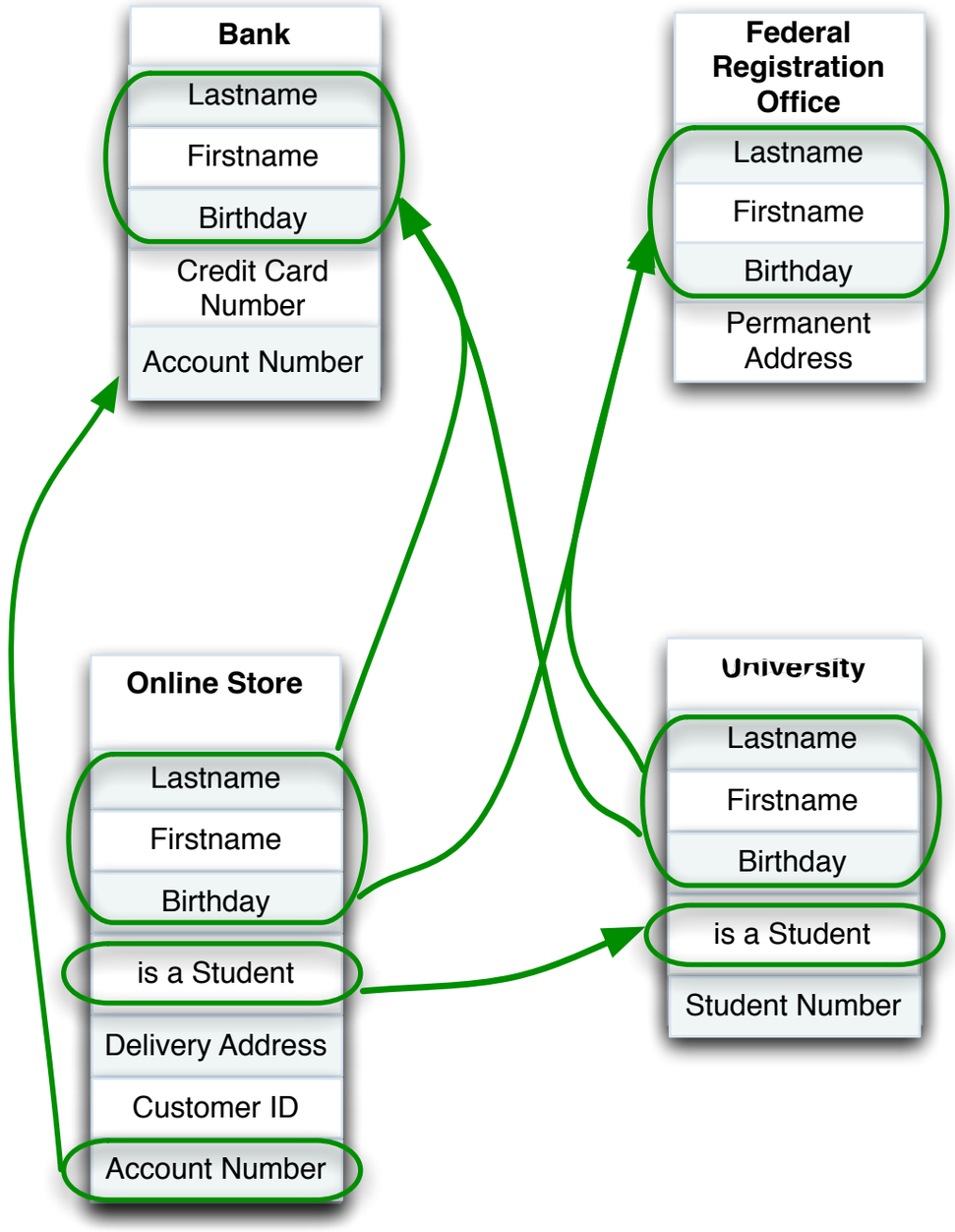
Identity Assurance Frameworks Examples

- UK Office of the e-Envoy
  - ”Registration and Authentication – E-Government Strategy Frame-work Policy and Guideline”
- US e-Authentication Initiative
  - ”E- Authentication Guidance for Federal Agencies” (OMB M-04-04)
- NIST
  - ”Electronic Authentication Guideline”(NIST 800- 63)
- InCommon federation
  - Identity Assurance Assessment Framework
  - Bronce and Silver Profile

Assurance Frameworks Limitations

- Identity is mostly considered as a whole
  - no distinction between different qualities of trust
- no changes of a trust level over time
  - identity attributes are gathered during the registration and often fix
- hard to reflect the uniqueness of identity providers with regard to their ability to assert certain identity attributes

Everybody is Identity Provider
Everybody is Relying Party

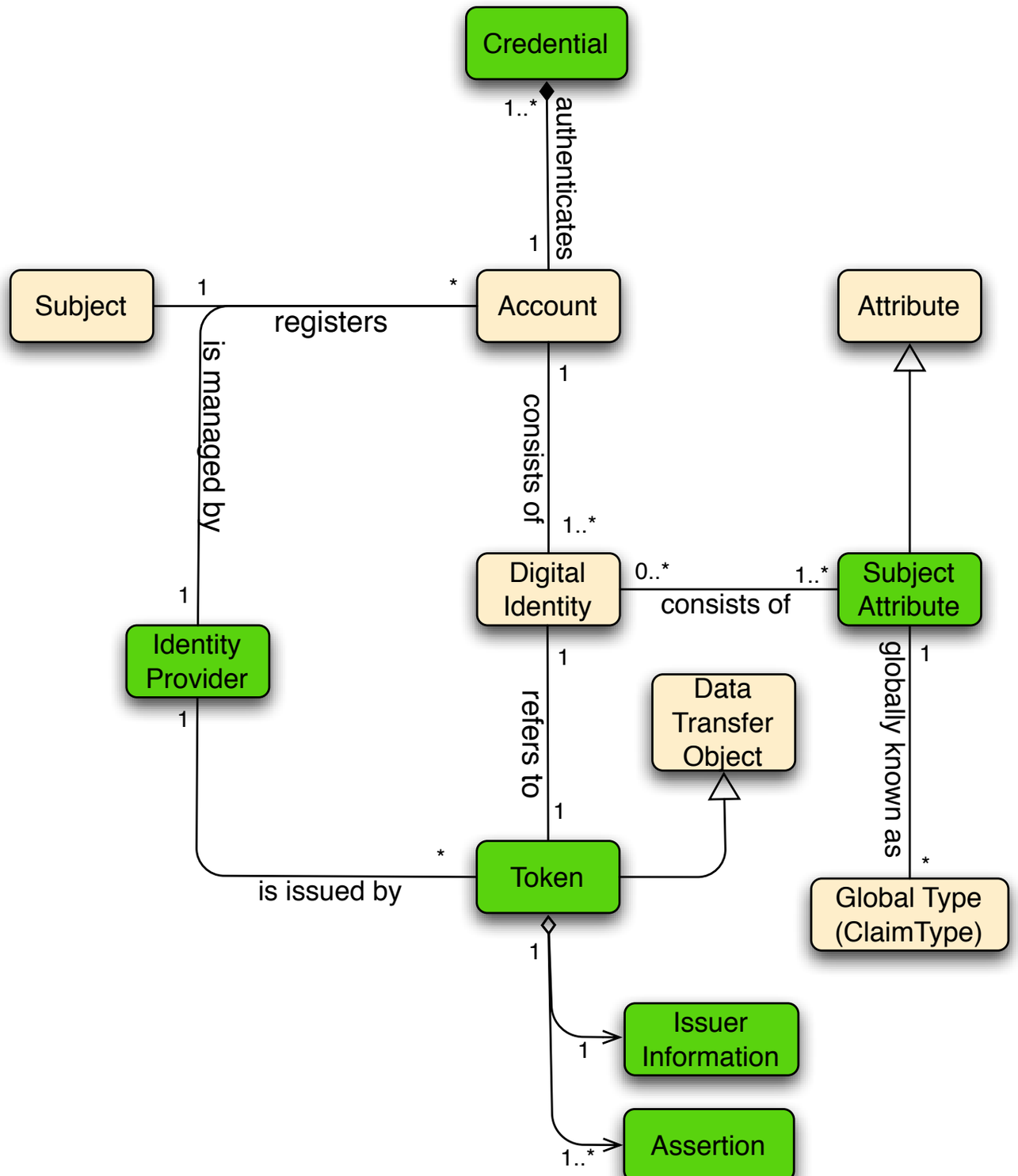


- Every Participant on the Internet
  - needs identity information
  - has identity information, he could share
- Aim:
  - Decentralized storage of identity information to
    - reduce redundancy
    - ease maintenance

AGENDA

- Motivation & Introduction
- Related Work: Assurance Frameworks
  - Limitations
- The need for Levels of Assurance for Attributes
  - Our Model of a Digital Identity
  - A Layered Trust Model
- An Identity Provider to manage Reliable Digital Identities
  - Identity Meta Information
  - SAML Attribute Statement Extensions
  - Demo
- Conclusion

Model of a digital identity

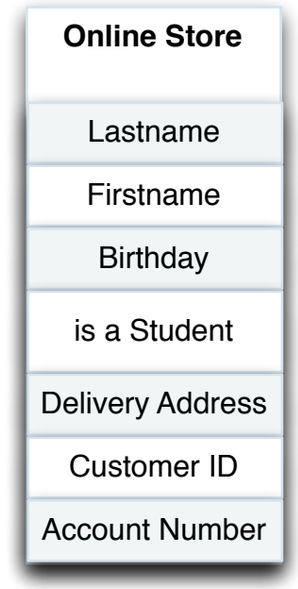


- **TC-0** Trust into the Identity Provider
- **TC-1** Trust into the authentication process and the subject-to-account mapping
- **TC-2** Trust into the subject's attributes.
- **TC-3** Trust into the token.

What about trust?

- Claim-based Identity Management allows
  - to state the attributes a relying party requires on a per-claim basis

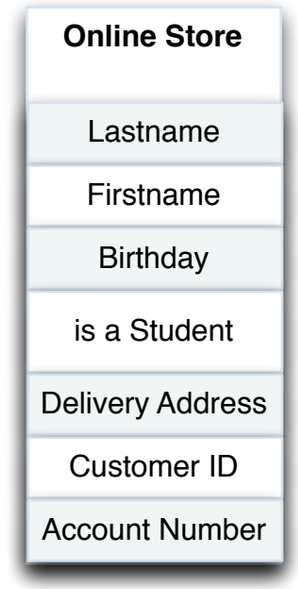
- Trust
  - is usually defined in a general manner
    - between organizations
    - complex contracts balance the risk between independent organizations



What about trust?

- Claim-based Identity Management allows
  - to state the attributes a relying party requires on a per-claim basis

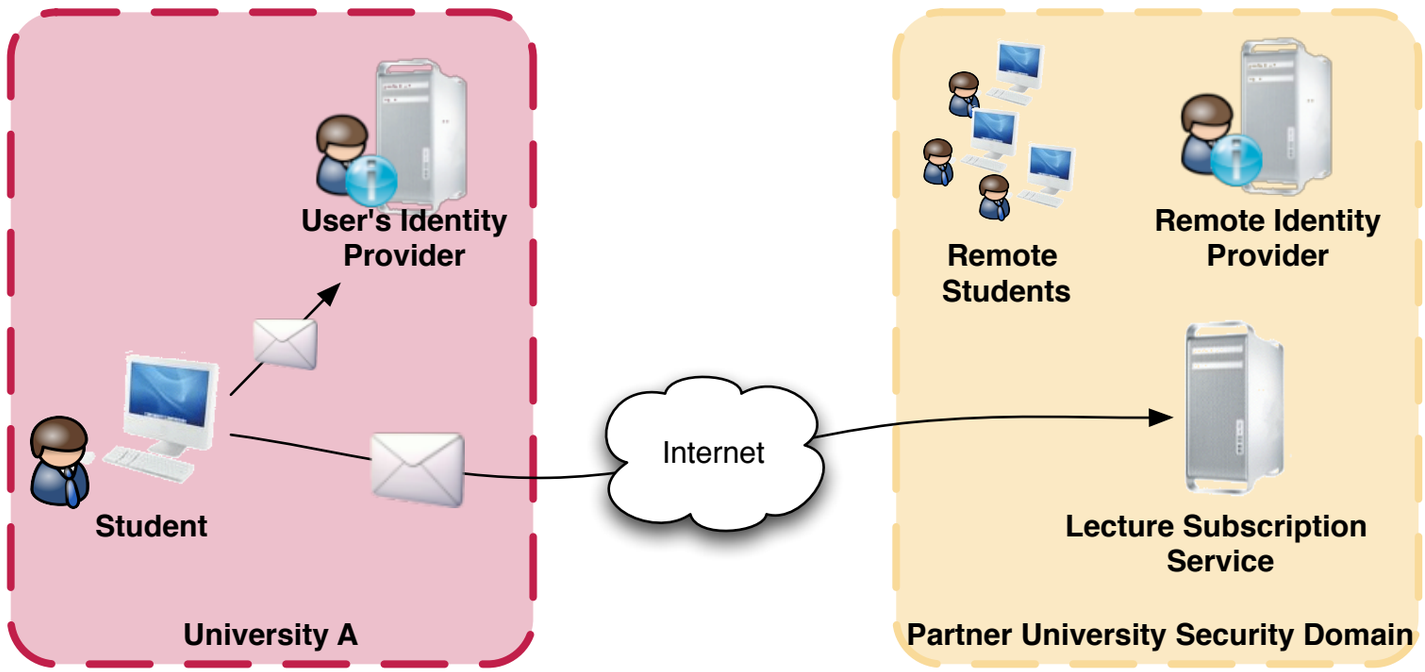
- Trust
  - is usually defined in a general manner
    - between organizations
    - complex contracts balance the risk between independent organizations



Layered Trust Model

Trust is required on two levels

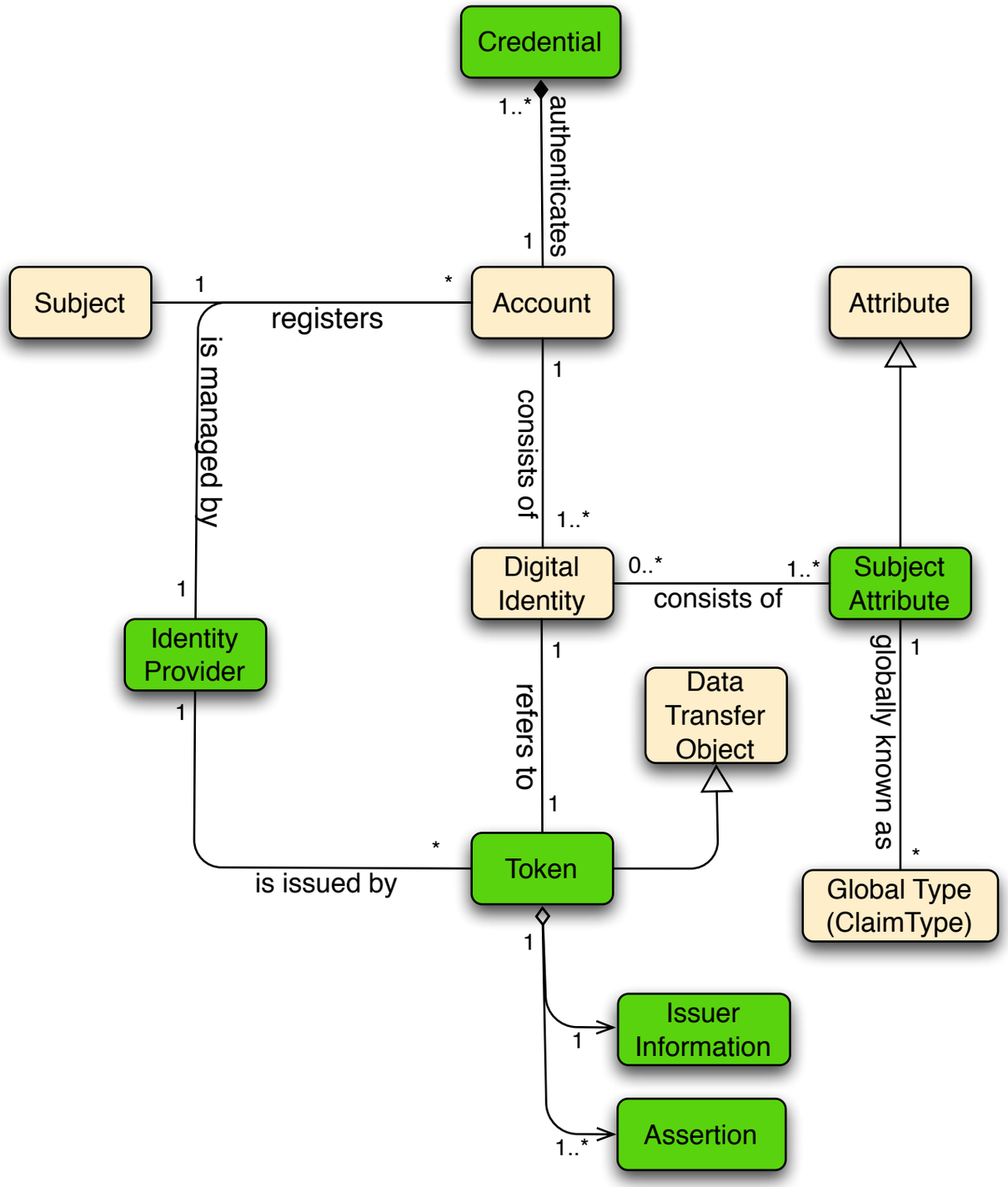
- between the service provider and the identity provider
  - general requirement to trust the **issuer** of an assertion
  - **= Organizational Trust**
- for a request: between the service provider and the requester
  - for a concrete request to trust the **subject** of an assertion
  - **= Identity Trust**



Comparison

Organizational Trust	Identity Trust
refers to the quality of the trust relationship between organizations	refers to the identity associated with a transaction
Can we trust the issuer of a security token?	Can we trust the subject in the token?
determined out-of-band	determined during service call
configurable	negotiable

Model of a digital identity revised



- **Organizational Trust**
  - **Identity Provider**
  - **TC-1** Trust into the authentication process and the subject-to-account mapping
  - **TC-3** Trust into the token.
- **Identity Trust**
  - **TC-2** Trust into the subject's attributes.

Formalism

Two aspects

- Is the issuer of the assertion trusted? (Organizational Trust)

$$isTrusted : Issuer \quad \mapsto \{trusted, untrusted\}$$

- Has the attribute been verified by the issuer? (Identity Trust)

$$isVerified : (Issuer, Claim) \quad \mapsto \{verified, unverified\}$$

- Trust into a claim

Formalism

Two aspects

- Is the issuer of the assertion trusted? (Organizational Trust)

$$isTrusted : Issuer \quad \mapsto \{trusted, untrusted\}$$

- Has the attribute been verified by the issuer? (Identity Trust)

$$isVerified : (Issuer, Claim) \quad \mapsto \{verified, unverified\}$$

- Trust into a claim

$$h : \{trusted, untrusted\} \times \{ verified, unverified\} \mapsto \{1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 1, & \text{if } b_1 = trusted \text{ and } b_2 = verified \\ 0, & \text{otherwise} \end{cases}$$

Formalism

Two aspects

- Is the issuer of the assertion trusted? (Organizational Trust)

$$isTrusted : Issuer \quad \mapsto \{trusted, untrusted\}$$

- Has the attribute been verified by the issuer? (Identity Trust)

$$isVerified : (Issuer, Claim) \quad \mapsto \{verified, unverified\}$$

- Trust into a claim

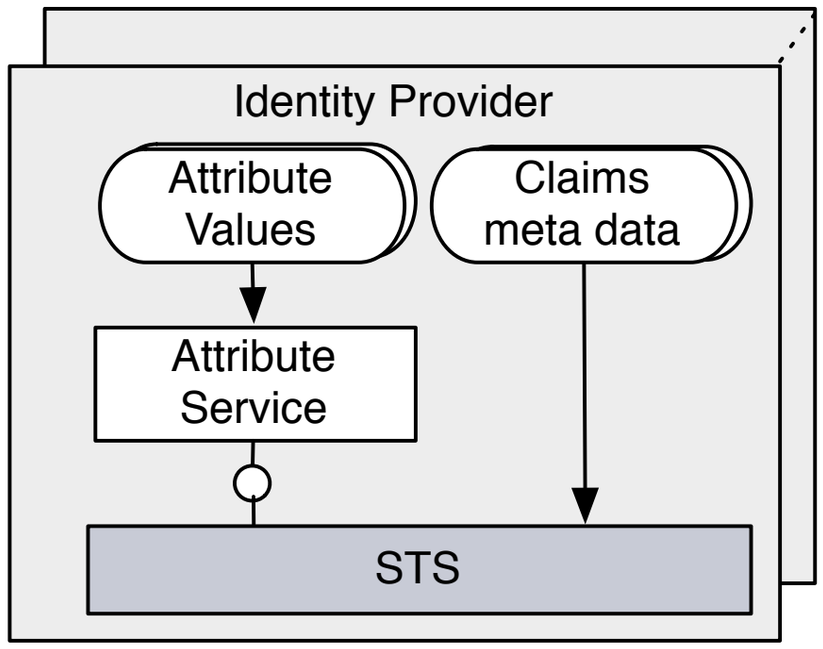
$$h : \{trusted, untrusted\} \times \{ verified, unverified\} \mapsto \{2, 1, 0\}$$

$$h : (b_1, b_2) \mapsto \begin{cases} 2, & \text{if } b_1 = trusted \text{ and } b_2 = verified \\ 1, & \text{if } b_1 = untrusted \text{ and } b_2 = verified \\ 0, & \text{otherwise} \end{cases}$$

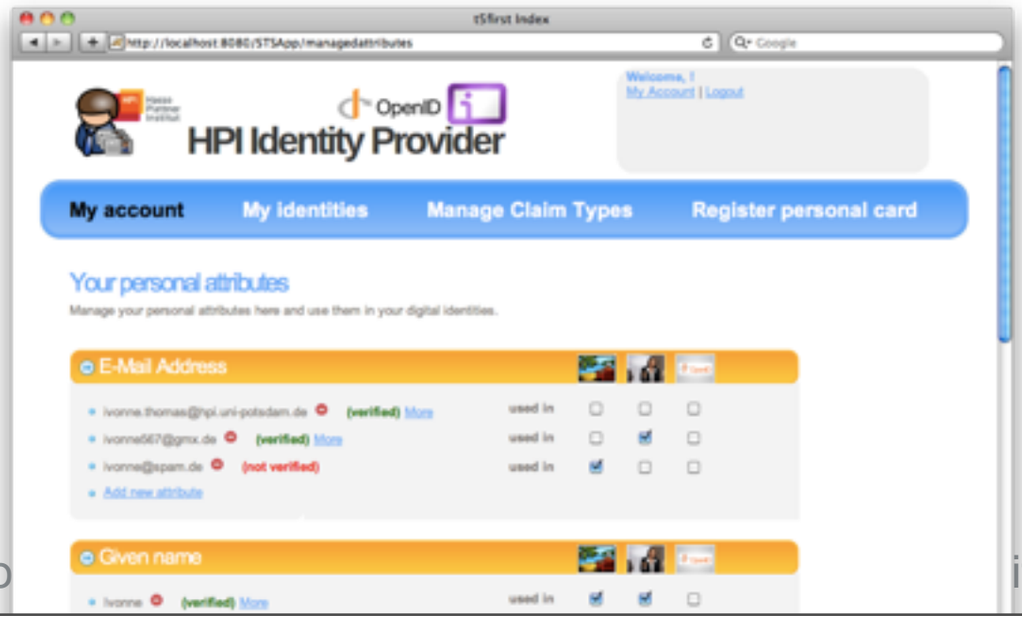
AGENDA

- Motivation & Introduction
- Related Work: Assurance Frameworks
  - Limitations
- The need for Levels of Assurance for Attributes
  - Our Model of a Digital Identity
  - A Layered Trust Model
- An Identity Provider to manage Reliable Digital Identities
  - Identity Meta Information
  - SAML Attribute Statement Extentions
  - Demo
- Conclusion

Identity Provider



- Identity Provider
  - Add, Edit, Remove ClaimTypes
  - Compose ClaimTypes to Digital Identities
  - Request identity information and receive security tokens
  - different protocols are possible:WS-Trust, OpenID



Identity Meta Information

Identity Attribute Metadata

Identity Attribute Meta data

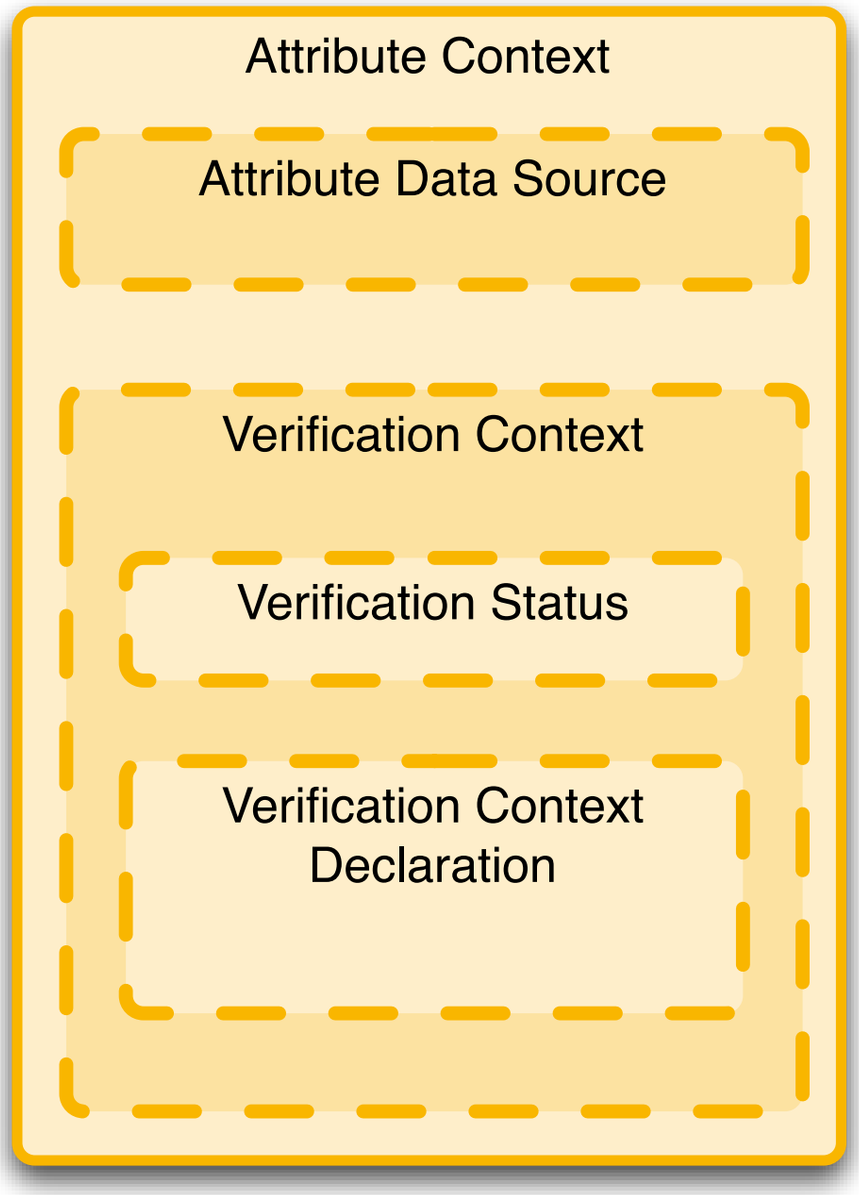
This page shows additional information about the registration and issuance processes with regard to your attributes.

E-Mail Address

Identity Metadata	**Source of identity data** (Where does the data come from?)	**Verification Method** (How was the identity attribute verified?)
ivonne.thomas@hpi.uni-potsdam.de	issued by an authority (HPI)	issuer is owner
ivonne567@gmx.de	entered by the user	verification mail has been sent

Attribute Context Classes

For each attribute, we store additional trust information:



SAML Attribute Statement Extensions Example


```
[...]
<saml:Subject>
  <saml:NameID>MaxMustermann
  </saml:NameID>
</saml:Subject>
[...]
<saml:AttributeStatement>
  <saml:Attribute
     xmlns:x500="urn:oasis:names:tc:SAML:1.1:
     nameid-format:emailAddress"
     FriendlyName="emailAddress">
     <saml:AttributeValue
        xsi:type="xs:string">staff@company.de
     </saml:AttributeValue>
     <samlext:AttributeContext>
       <samlext:AttributeDataSource>
          user
       </samlext:AttributeDataSource>
       <samlext:VerificationContext>
         <samlext:VerificationStatus>
            verified
         </samlext:VerificationStatus>
         <samlext:VerificationContextDecl>
           <samlext:VerificationMethod
              xmlns:samlextEmail=
              "http://de.hpi.ip/saml20/ext/email">
              <ConfirmationEmailReceived />
           </samlext:VerificationMethod>
         </samlext:VerificationContextDecl>
       </samlext:VerificationContext>
     </samlext:AttributeContext>
  </saml:Attribute>
</saml:AttributeStatement>
```

AGENDA

- Motivation & Introduction
- Related Work: Assurance Frameworks
  - Limitations
- The need for Levels of Assurance for Attributes
  - Our Model of a Digital Identity
  - A Layered Trust Model
- An Identity Provider to manage Reliable Digital Identities
  - Identity Meta Information
  - SAML Attribute Statement Extensions
  - Demo
- Conclusion

Conclusion

- SOA requires an open, decentralized Identity Management
- Everybody is Identity Provider as well as Relying Party

- Claims express the identity attributes a relying party requires
- use Claims Meta Information in order to enable a relying party to rely on identity data from remote resources

CardSpace-Liberty Integration for CardSpace Users

Haitham S. Al-Sinani*
Information Security Group
Royal Holloway, University of London
http://www.isg.rhul.ac.uk
H.Al-Sinani@rhul.ac.uk

Waleed A. Alrodhan
Information Security Group
Royal Holloway, University of London
http://www.isg.rhul.ac.uk
W.A.Alrodhan@rhul.ac.uk

Chris J. Mitchell
Information Security Group
Royal Holloway, University of London
http://www.isg.rhul.ac.uk
C.Mitchell@rhul.ac.uk

ABSTRACT

Whilst the growing number of identity management systems have the potential to reduce the threat of identity attacks, major deployment problems remain because of the lack of interoperability between such systems. In this paper we propose a novel scheme to provide interoperability between two of the most widely discussed identity management systems, namely Microsoft CardSpace and Liberty. In this scheme, CardSpace users are able to obtain an assertion token from a Liberty-enabled identity provider that will satisfy the security requirements of a CardSpace-enabled relying party. We specify the operation of the integration scheme and also describe an implementation of a proof-of-concept prototype. Additionally, security and operational analyses are provided.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection

General Terms

Security

Keywords

Identity Management, CardSpace, Liberty Alliance Project, Interoperability, SAML, Browser Extension

1. INTRODUCTION

In line with the continuing increase in the number of online services requiring authentication, there has been a proportional rise in the number of digital identities needed for authentication purposes. This has contributed to the recent rapid growth in identity-oriented attacks, such as phishing, pharming, etc. In an attempt to mitigate such attacks, a number of identity management systems have been proposed.

Identity management deals with uniquely identifying individuals in a system, and with effectively controlling access to the system resources by managing the rights and privileges associated with digital identities. The most important service provided by an identity management system is authentication. Such a system may also support other services, such as pre-authentication, authorisation, single sign-on, identity repository management, user self-service registration, and audit. Examples of identity management systems include CardSpace[1], Liberty[2], OpenID[3], and Shibboleth[4] [5, 8, 17, 46, 50].

Most identity management architectures involve the following main roles.

1. The identity provider (IdP), which issues an identity token to a user.

2. The service provider (SP), or the relying party (RP) in CardSpace terminology, which consumes the identity token issued by the IdP in order to identify the user, before granting him/her access.

3. The user, also known as the principal.

4. The user agent, i.e. software employed by a user to send requests to webservers and receive data from them, such as a web browser. Typically, the user agent processes protocol messages on behalf of the user, and prompts the user to make decisions, provide secrets, etc.

An identity provider supplies a user agent with an authentication token that can be consumed by a particular service provider. Whilst one service provider might solely support CardSpace, another might only support Liberty. Therefore, to make these systems available to the largest possible group of users, effective interoperability between systems is needed. In this paper we investigate a case involving a CardSpace-enabled relying party, a Liberty-enabled identity provider, and a user agent that is (only) CardSpace-enabled. The goal is to develop an approach to integration that is as transparent as possible to both identity providers and relying parties.

*This author is sponsored by the Diwan of Royal Court, Sultanate of Oman.

IDtrust '10, April 13–15, 2010, Gaithersburg, MD

[1] `http://msdn.microsoft.com/en-us/library/aa480189.aspx`
[2] `http://www.projectliberty.org/`
[3] `http://openid.net/`
[4] `http://shibboleth.internet2.edu/`

We have chosen to consider the integration of Liberty with CardSpace because of Liberty's wide adoption (see section 2.2.1). Currently, it is a leading identity management architecture, that has gained the acceptance of a number of technology-leading companies and organisations. Complementing this, the wide use of Windows, recent versions of which incorporate CardSpace, means that enabling interoperation between the two systems is likely to be of significance for large numbers of identity management users and service providers. Another reason for choosing Liberty is because of the similarity between the message flows in its ID-FF profile and CardSpace.

The remainder of the paper is organised as follows. Section 2 presents an overview of CardSpace and Liberty, and section 3 contains the proposed integration scheme. In section 4, we provide an operational analysis of the scheme and, in section 5, we describe a prototype implementation. Section 6 highlights possible areas for related work, and, finally, section 7 concludes the paper.

2. CARDSPACE AND LIBERTY

We provide an introduction to the CardSpace and Liberty identity management systems. SAML is also briefly outlined.

2.1 CardSpace

We first give a general introduction to CardSpace, covering relevant operational aspects.

2.1.1 Introduction to CardSpace

CardSpace is Microsoft's implementation of a digital identity metasystem, in which users can manage digital identities issued by a variety of identity providers, and use them in a range of contexts to access online services. In CardSpace, digital identities are represented to users as Information Cards (or InfoCards). From the CardSpace perspective, InfoCards are XML-based files that list the types of claim made by one party about itself or another party. CardSpace is designed to reduce reliance on username-password authentication, and to provide a consistent authentication experience across the Web to improve user understanding of the authentication process. It is claimed that CardSpace is also designed to reflect the seven identity laws promulgated by Microsoft [6, 10, 17, 34].

The concept of an InfoCard is inspired by real-world cards, such as driving licences and credit cards. A user can employ one InfoCard with multiple websites. Alternatively, just as different physical ID cards are used in distinct situations, separate InfoCards can be used at different websites, helping to enhance user privacy and security. If InfoCards are obtained from different IdPs, the credentials referred to by such cards are stored in distinct locations, potentially improving reliability and security, as well as giving users flexibility in choosing points of trust.

There are two types of InfoCards: personal (self-issued) cards and managed cards. Personal cards are created by users themselves, and the claims listed in such an InfoCard are asserted by the self-issued identity provider (SIP) that co-exists with the CardSpace identity selector on the user machine. In this paper we use personal cards to enable interoperation between CardSpace and Liberty. Managed cards, on the other hand, are obtained from remote identity providers.

The InfoCards themselves do not contain any sensitive information; instead an InfoCard carries metadata that indicates the types of personal data that are associated with this identity, and from where assertions regarding this data can be obtained. The data referred to by personal cards is stored on the user machine, whereas the data referred to by a managed card is held by the identity provider that issued it [6, 16, 18, 24, 34, 35, 38].

By default, CardSpace is supported in Internet Explorer (IE) from version 7 onwards. Extensions to other browsers, such as Firefox[5], and Safari[6] also exist. Microsoft has recently released an updated version of CardSpace, known as Windows CardSpace 2.0 Beta 2[7]. However, in this paper we refer throughout to the CardSpace version that is shipped by default as part of Windows Vista and Windows 7, which has also been approved as an OASIS standard under the name 'Identity Metasystem Interoperability Version 1.0' (IMI 1.0) [28].

2.1.2 CardSpace Personal Cards

The core idea introduced in this paper is to use CardSpace personal cards to make Liberty identity providers available via the CardSpace identity selector. We therefore next describe CardSpace personal cards.

Creation of Personal Cards.

Prerequisites for use of a CardSpace personal card include:

1. a CardSpace-enabled RP; and
2. a CardSpace-enabled user agent, e.g. a web browser capable of invoking the CardSpace identity selector, such as those shipped as part of Windows Vista and Windows 7.

The identity selector allows a user to create a personal card and populate its fields with self-asserted claims. To protect users from disclosing sensitive information, CardSpace restricts the contents of personal cards to non-sensitive data, such as that published in telephone directories. Personal cards currently only support 14 editable claim types, namely *First Name, Last Name, Email Address, Street, City, State, Postal Code, Country/Region, Home Phone, Other Phone, Mobile Phone, Date of Birth, Gender*, and *Web Page*. Data inserted in personal cards is stored in encrypted form on the user machine.

When a user creates a new personal card, CardSpace generates an ID and a master key for this card. The card ID is a globally unique identifier (GUID), and the master key is 32 bytes of random data.

Using Personal Cards.

When using personal cards, CardSpace adopts the following protocol. We describe the protocol for the case where the RP does not employ a security token service (STS[8]).

1. User agent → RP. HTTP/S request: GET (login page).

[5] `https://addons.mozilla.org/en-US/firefox/addon/10292`
[6] `http://www.hccp.org/safari-plug-in.html`
[7] `http://technet.microsoft.com/en-us/library/dd996657(WS.10).aspx`
[8] The STS is responsible for security policy and token management within an IdP and, optionally, within an RP [27].

2. RP → user agent. HTTP/S response. A login page is returned containing the CardSpace-enabling tags in which the RP security policy is embedded.

3. User → user agent. The user agent offers the user the option to use CardSpace (e.g. via a button on the RP web page); selection of this option causes the agent to invoke the CardSpace identity selector, passing the RP policy to the selector. Note that if this is the first time that this RP has been contacted, the identity selector will display the identity of the RP, giving the user the option either to proceed or to abort the protocol.

4. User agent → user agent (identity selector → InfoCards). The CardSpace identity selector, after evaluating the RP security policy, highlights the InfoCards that match the policy, and greys out those that do not. InfoCards previously used for this particular RP are displayed in the upper half of the selector screen.

5. User → user agent (user → identity selector). The user chooses a personal card. (Alternatively, the user could create and choose a new personal card). The user can also preview the card (with its associated claims) to see which claim values are being released. Note that the selected InfoCard may contain several claims, but only the claims explicitly requested in the RP security policy will be passed to the requesting RP.

6. User agent ⇌ user agent (identity selector ⇌ SIP). The identity selector creates and sends a SAML-based Request Security Token (RST) to the SIP, which responds with a SAML-based Request Security Token Response (RSTR).

7. User agent → user agent (identity selector → user agent). The RSTR is then passed to the user agent, which forwards it to the RP.

8. RP → user. The RP validates the token, and, if satisfied, grants access to the user.

The managed card operational protocol is similar, except that the remote IdP specified in the InfoCard is contacted instead of the SIP. The CardSpace identity selector then uses the standard identity metasystem protocols (see section 2.1.3) to first retrieve the IdP security policy[9] and then obtain a security token representing the selected digital identity from the STS of the remote IdP. The identity selector then passes the received token to the user agent, optionally after first obtaining permission from the user[10] [27, 41].

For CardSpace to work, both the RP and the IdP must be CardSpace-enabled. The problem that we address here is the incompatibility issue that will occur if the RP is CardSpace-enabled whereas the IdP is not, but is instead Liberty-enabled. Addressing this issue could help to extend the applicability of CardSpace.

[9]Depending on the IdP security policy, the user may be requested to provide credentials for authentication to the selected IdP. The authentication methods currently supported by CardSpace include username-password authentication, a KerberosV5 service ticket, an X.509v3 certificate, and a self-issued token.

[10]This may involve presenting the user with a 'display token', prepared by the remote IdP, listing the claim values asserted in the 'real' security token; the identity selector will only continue if the user is willing to release such values.

Private Personal Identifiers.

The private personal identifier (PPID) is a unique identifier linking a specific InfoCard to a particular RP [6, 7, 38]. CardSpace RPs can use the PPID along with a digital signature to authenticate a user.

When a user uses a personal card at an RP for the first time, CardSpace generates a site-specific:

- PPID by combining the card ID with data taken from the RP certificate; and
- signature key pair by combining the card master key with data taken from the RP certificate.

In both cases, the domain name or IP address of the RP is used if no RP certificate is available.

Since the PPID and key pair are RP-specific, the PPID does not function as a global user identifier, helping to enhance user privacy. In addition, compromising the PPID and key pair for one RP does not allow an adversary to impersonate the user at other RPs. The CardSpace identity selector only displays a shortened version of the PPID to protect against social engineering attacks and to improve readability.

When a user first registers with an RP, the RP retrieves the PPID and the public key from the received authentication token, and stores them. If a personal InfoCard is re-used at a site, the supplied authentication token will contain the same PPID and public key as used previously, signed using the corresponding private key. The RP compares the received PPID and public key with its stored values, and verifies the digital signature. If all checks succeed it has assurance that it is the same user.

The PPID could be used on its own as a shared secret to authenticate a user to an RP. However, it is recommended that the associated (public) signature verification key, as held by the RP, should also always be used to verify the signed authentication token to provide a more robust authentication method [6].

2.1.3 CardSpace Protocols

In order to maximise interoperability with non-Windows platforms, CardSpace has been specifically designed to use open standards-based protocols, notably the WS-* standards, the most significant of which are listed below.

WS-Policy/WS-SecurityPolicy is used to describe security policies [3, 21]. Note that a website can also describe its policy in HTML/XHTML.

WS-MetadataExchange is used to fetch security policies and exchange service description metadata over the Internet [4]. Note that a website can also transmit its security policy using HTTP/S.

WS-Trust is used to acquire security tokens (e.g. SAML tokens) from IdPs [2].

WS-Security is used to securely deliver security tokens to RPs [37]. Note that HTTP/S can also be used.

2.1.4 Proof Keys

A SAML security token can be coupled with cryptographic evidence to demonstrate the sender's rightful possession of the token. A 'proof key' is a key associated with a security token, and the data string used to demonstrate the sender's

knowledge of that key (e.g. through the inclusion of a digital signature or MAC computed using the key) is called the 'proof-of-possession' of the security token [27, 38].

A security token can be associated with two types of proof key.

1. Symmetric proof keys

   If a symmetric key token is requested, a symmetric proof key is established between the identity selector and the CardSpace-enabled IdP [38], which is then revealed to the RP. This key is used to prove the subject's rightful possession of the security token. Whilst the use of such a key may optimise token processing in terms of speed and efficiency [36], it involves revealing the identity of the RP to the IdP, which is not ideal from a privacy perspective.

2. Asymmetric proof keys

   If an asymmetric key token is requested, the identity selector generates an ephemeral RSA key pair and sends the public part of the key to the CardSpace-enabled IdP. The identity selector also sends a supporting signature to prove ownership of the corresponding private key [38]. If approved by the IdP, the public part is sent to the RP in the security token. The private part of the RSA key pair is then used to prove the subject's rightful possession of the security token. Although the use of such a key may not be as efficient as the symmetric approach, it helps to protect user privacy since the identity of the RP does not need to be disclosed to the IdP.

It merits mentioning that the default behaviour of the CardSpace identity selector is different in the special case of browser-based client interactions with a website, in which case 'bearer' tokens are requested. Because a web browser is only capable of submitting a token to a website passively over HTTP without any proof-of-possession, bearer tokens with no proof keys are used [36].

2.2 Liberty

We next give a general introduction to Liberty, covering relevant operational aspects.

2.2.1 Introduction to Liberty

The Liberty Alliance is a large consortium, established in 2001 by approximately 30 organisations; it now has a global membership of more than 150[11]. The Liberty Alliance Project (or simply Liberty) builds open, standards-based specifications for federated identity, provides interoperability testing, and helps to prevent identity theft. Liberty also aims to establish best practices and business guidelines for identity federation. According to its website, Liberty has been widely adopted with, as of 2006, more than one billion Liberty-enabled identities and devices[12]. As of mid 2009, the work of the Liberty Alliance is being adopted by the Kantara Initiative[13].

Figure 1 shows the general Liberty model, which is essentially a single sign-on (SSO) model [11]. In this model, a

[11] `http://www.projectliberty.org/liberty/membership/current_members/`

[12] `http://www.projectliberty.org/liberty/adoption/`

[13] `http://kantarainitiative.org/`

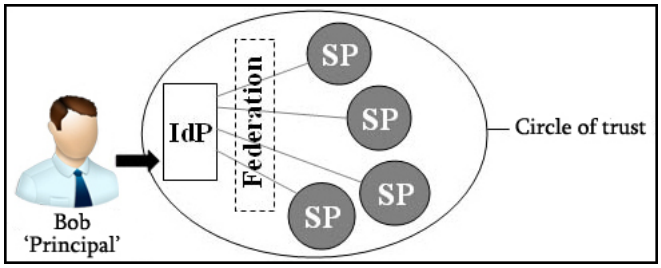


Figure 1: The Liberty model

principal (or a user) can federate its various identities to a single identity issued by an identity provider, so that the user can access services provided by service providers belonging to the same circle of trust by authenticating just once to the identity provider. This relies on a pre-established relationship between the identity provider and every service provider in the circle of trust.

The Liberty specifications are divided into three frameworks: the identity federation framework (ID-FF) [49], the identity web services framework (ID-WSF) [47] and the service interface specifications (ID-SIS) [30]. In this paper we focus on the ID-FF. The ID-FF provides approaches for implementing federation and SSO, including supporting mechanisms such as session management and identity/account linkage.

2.2.2 Liberty Functional Requirements

The Liberty architecture [49] supports the following activities.

Identity federation This is the process of linking a user's SP identity with a specific IdP (given user consent). At the time of federation, two user pseudonyms[14] are created for the IdP-SP association, one for use by each party. De-federation is the reverse process.

Single sign-on This feature enables a user to log in once to an IdP in a Liberty circle of trust and subsequently use SPs belonging to this circle without the need to log in again. Global log-out is the reverse process.

Anonymity A Liberty SP may request a Liberty IdP to supply a temporary pseudonym that will preserve the anonymity of a user. This identifier may be used to obtain information for or about the user (given their consent) without requiring the user to consent to a long term relationship with the SP [49].

2.2.3 Single Sign-on and Federation Profiles

The Liberty ID-FF protocol specification [14] defines the SSO and federation protocol. The ID-FF bindings and profile specification [12] defines profiles, i.e. mappings of ID-FF protocol messages to particular communication protocols (e.g. HTTP [22]). The latter document also describes the common interactions and processing rules for these profiles.

The single sign-on and federation protocol has three associated profiles, summarised below.

[14] A pseudonym is an opaque but unique handle (identifier) for the user, enabling the user's real identity to remain private. Pseudonyms can be temporary or persistent, and are included in SAML tokens exchanged between a Liberty IdP and SP.

Liberty artifact profile The Liberty artifact profile involves embedding an artifact (i.e. an opaque handle) in a URI exchanged between the IdP and SP via Web redirection, and also requires direct (background) communication between the SP and IdP [49]. The SP uses the artifact to retrieve the full SAML assertion from the IdP. As it requires direct SP-IdP communication, which is inconsistent with the CardSpace approach[15], the proposed scheme does not support this profile.

Liberty browser post profile JavaScript-enabled browsers can perform an HTTP redirect between IdPs and SPs by using JavaScript to automatically send a form (containing the authentication data). This profile embeds the entire SAML assertion in an HTML form. As a result, it does not use an artifact and does not require any direct communication between the SP and the IdP. The scheme proposed here supports this profile.

Liberty-enabled client (and proxy) profile This profile defines interactions between Liberty-enabled clients (and/or proxies), SPs, and IdPs. A Liberty-enabled client (LEC) is a user agent that can directly communicate with the IdP that the user intends to use to support its interactions with an SP. In addition, the LEC sends and receives Liberty messages in the body of HTTP requests/responses using 'post', rather than relying upon HTTP redirects and encoding protocol parameters into URLs. Therefore, LECs do not impose any restrictions on the size of the protocol messages. Interactions between a user agent and an IdP are SOAP-based, and the protocol messages include Liberty-specified HTTP headers.

Although it adds complexity, this profile seems like a natural fit to the proposed scheme. We propose to use the CardSpace identity selector to act as a Liberty-enabled client. In our scheme, the identities of the IdPs are stored on CardSpace personal cards.

2.2.4 *Proof Keys*

The Liberty ID-FF supports SAML 2.0 assertions as a security token type. The SAML 2.0 specifications offer three proof-of-possession methods (also referred to as subject confirmation methods): Holder-of-Key (HoK), Sender-Vouches, and bearer [13].

The HoK method [45] can be used to address both the symmetric and asymmetric proof-of-possession requirements of a CardSpace-enabled RP.

2.3 SAML

SAML is an XML-based standard for exchanging identity-related information across the Internet. The SAML specifications cover four major elements.

A SAML assertion can contain three types of statement:

1. an authentication statement, asserting that a user was authenticated at a particular time using a particular authentication method;
2. an attribute statement, asserting that a user is associated with certain attributes; and
3. an authorization decision statement, asserting that a particular user is permitted to perform a certain action on a specific resource.

SAML protocols define data structures for sending SAML requests and returning assertions.

SAML bindings map SAML protocol messages onto standard communication protocols, e.g. HTTP.

SAML profiles describe how SAML assertions, protocols and bindings are combined together to support a particular use case.

SAML 1.0 [26] was first adopted as an OASIS standard in 2002; a minor revision, SAML 1.1 [33], was formally adopted in 2003. A major revision led to SAML 2.0 [13], which became a standard in 2005. The differences[16] between version 1.1 and 2.0 are significant, and SAML assertions of the two types are incompatible.

Finally note that the CardSpace SIP currently only issues tokens conforming to SAML 1.1 [38], whereas the Liberty specifications require IdPs to generate assertions using SAML 2.0 syntax.

3. THE INTEGRATION SCHEME

This section provides an overview of the scheme, and also gives a brief description of its protocol flow. However, we first highlight the main differences between the integration scheme proposed here and a previously proposed scheme of this type.

3.1 Previous Work

The integration scheme proposed here builds on a previous proposal for CardSpace-Liberty integration [1], referred to below as the AM scheme. Whilst the scheme proposed here has some properties in common with this previous proposal, for example both approaches concentrate on supporting integration at the client rather than at the server, there are a number of important differences.

Instead of focusing on CardSpace users only, as is the case with the scheme described here, the AM scheme allows for full interoperability even in the case where the SP is Liberty-enabled and the IdP is CardSpace-enabled. However, since no prototype has been developed, issues which might arise during deployment have not been explored. By contrast, the scheme described below has been prototyped, and hence greater confidence can be derived in its practicality.

One important goal for any identity management system is ease of use. However, user interface issues, notably the operation of the integration software on the client platform, have not been explored for the AM scheme, whereas the proposal here addresses this through a combination of a browser extension and the CardSpace interface. In addition, whereas the relationship between the integration software and the web browser is not specified for the AM scheme, this issue has been resolved for the scheme presented here by implementing the functionality in a web browser plug-in residing on the user machine.

The means by which the integration software is triggered is also not clear for the AM scheme. For example, if the integration software is assumed to run at all times, then

[15] In CardSpace, all RP-IdP communications must go through the identity selector on the user machine.

[16] `https://spaces.internet2.edu/display/SHIB/SAMLDiffs`

problems arise if the user wants to use CardSpace or Liberty without integration. By contrast, several ways of addressing this particular issue are described in sections 3.2.3 and 4.3.

The AM scheme does not address how to handle the private personal identifier (PPID), described in section 2.1.2, when supporting interoperation between RPs and Liberty-enabled IdPs. Additionally, it is not clear whether providing the full address of the IdP is the responsibility of the RP, the integration software, or the user. These issues are addressed in sections 3.2 and 5.

3.2 Integration Protocol

We now present the novel protocol.

3.2.1 System Parties

As stated earlier, the integration scheme addresses the incompatibility issue arising if the RP is CardSpace-enabled and the IdP is Liberty-enabled. The parties involved are as follows.

1. A CardSpace-enabled RP.
2. A CardSpace-enabled user agent (e.g. a suitable web browser).
3. A Liberty-enabled IdP.
4. The integration browser extension (which must first be installed).

Note that there is no need for a Liberty-enabled user agent. Instead the user only needs to install the integration browser extension.

Figure 2 gives a simplified picture of the high-level interactions between system parties on the user machine. The parties shown are the browser extension, the user agent (browser), the identity selector, and the SIP. The arrows indicate information flows.

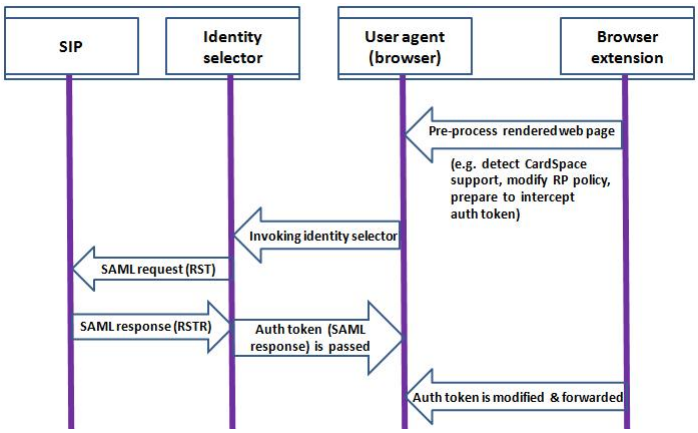


Figure 2: Data flows between client parties

3.2.2 Preconditions

The scheme has the following requirements.

- The user must have an existing relationship with a CardSpace RP.
- The user must have an existing relationship with a Liberty-enabled IdP, and hence the IdP has a means of authenticating the user.
- The CardSpace-enabled RP must not employ an STS (see section 4.7). Instead, the RP must express its security policy using HTML/XHTML, and interactions between the CardSpace identity selector and the RP must be based on HTTP/S via a web browser. This is because of the use of a browser extension (see section 3.2.4) in the scheme, and a browser extension by itself is incapable of managing the necessary communications with an STS.
- The CardSpace-enabled RP must support SAML 2.0 (see section 2.3).
- As well as being able to verify the InfoCard signature, the CardSpace-enabled RP must be able to verify the the IdP digital signature in the provided SAML token.
- The Liberty-enabled IdP must be prepared to provide SAML assertions for SPs for which a federation agreement does not exist for the user concerned[17]. In the absence of the IdP-SP-specific user pseudonyms (which would exist if federation had occurred) the IdP is prepared to use the InfoCard PPID for the user in place of Liberty pseudonyms in the SAML request and response messages (and in the created SAML assertion). This avoids changes to the Liberty message formats, but does require a minor policy/operational change to the Liberty-enabled IdP.

3.2.3 LibertyCards

Either prior to, or during, use of the integration protocol, the user must create a special personal card, referred to as a LibertyCard, which will represent the Liberty IdP. This card must contain the URL of the Liberty IdP it represents, and must also contain a predefined sequence of characters, e.g. the word 'Liberty', which will be used to trigger the integration software (see section 4.3).

The browser extension, described in section 3.2.4, must process the policy statement provided by the RP before it is passed to the identity selector. It must first decide whether or not the RP policy requirements can be met by one or more of the LibertyCards; if not then it leaves the policy statement unchanged, and the browser extension plays no further active part in processing. However, if use of a LibertyCard is appropriate, then the browser extension changes the policy to include the types of claim employed by LibertyCards. For example, if the URL of the Liberty IdP is stored in the *web page* field of the LibertyCard, then the browser extension must modify the RP security policy to add the *web page* claim (see section 5.3.1 for further details). Note that adding the claim types to the RP security policy is necessary to ensure that the token supplied by the SIP contains the values of these claims, which can then be processed by the browser extension; otherwise these values would not be available to the browser extension[18].

[17] It is thus not necessary for the user to Liberty-federate the IdP with the RP (which would in any case be difficult to achieve given that we are not requiring the RP to be Liberty-enabled).

[18] Unfortunately, whilst necessary for the operation of the browser extension, adding claims to the RP policy means that CardSpace-compliant IdPs for which the user has 'managed' InfoCards, and which might otherwise be acceptable to the RP, cannot be selected by the user.

One approach that would avoid the need to store the URL of the IdP in a personal card would involve the browser extension prompting the user to enter the URL of the IdP that they wish to contact, after they have selected a card. This could occur as part of step 8 in section 3.2.5. However this approach is not adopted here because it would require the user to manually enter the URL every time a LibertyCard is used, causing usability issues.

3.2.4 Browser Extension

The integration scheme is based on a browser extension that is able to:

- automatically execute;
- read and inspect browser-rendered web pages;
- modify rendered web pages if certain conditions hold;
- intercept, inspect and modify messages exchanged between a CardSpace identity selector and a CardSpace-enabled RP (via a browser);
- automatically forward security tokens (via browser-based HTTP redirects) to Liberty-enabled IdPs and to CardSpace-enabled RPs; and
- provide a means for a user to enable or disable it.

3.2.5 Protocol Operation

Figure 3 gives a simplified sketch of the integration scheme. The protocol operates as follows (with step numbers as shown in figure 3). Steps 1, 2, 4–7 and 12 of the integration scheme are the same as steps 1, 2, 3–6 and 8, respectively, of the CardSpace personal card protocol given in section 2.1.2, and hence are not described again here.

3. User agent → user agent (browser extension → browser). The browser extension scans the login page to detect whether the RP website supports CardSpace. If so, it starts to process the browser-rendered login page, including embedding a function into the page to intercept the authentication token that will later be returned by the CardSpace identity selector. If not, the browser extension terminates.

8. User agent → user agent (identity selector → browser extension). Unlike in the 'standard' case, the RSTR is not sent to the RP; instead the browser extension intercepts the RSTR (a SAML authentication response), converts it into a SAML authentication request, and forwards it to the appropriate Liberty-enabled IdP. Note that the detailed format of the SAML authentication request will depend on the Liberty profile being used (see discussion below).

9. Liberty-enabled IdP ⇌ user. If necessary, the Liberty-enabled IdP authenticates the user.

10. Liberty-enabled IdP → user agent. The IdP sends a SAML authentication response to the user agent. This response is also Liberty profile-dependent (see discussion below).

11. User agent → RP. The user agent forwards the token to the RP, optionally after first obtaining permission from the user (see section 4.4).

The detailed operation of steps 8 and 10 is dependent on the Liberty profile in use between the user agent and the IdP. The construction of the SAML authentication request in step 8 differs depending on whether the Liberty browser post (LBP) profile or the Liberty-enabled client (LEC) profile is in use. For example, the URI identifier 'URI: `http://projectliberty.org/profiles/brws-post`' must be used when employing the LBP profile, whereas 'URI: `http://projectliberty.org/profiles/lecp`' must be used when employing the LEC profile. In addition, when using the LEC profile, the authentication request must be submitted to the IdP as a SOAP [25] request with a Liberty-enabled header, whereas when using LBP, the authentication request to the IdP can be embedded in an HTML form.

The details of steps 10 and 11 differ significantly depending on which of the two Liberty profiles is in use. In the LEC profile, in step 10 the IdP returns the authentication response to the client (which is responsible for forwarding it to the specified SP). In the LBP profile, however, the IdP sends the HTML form carrying the authentication response to the user agent, and redirects the user via the user agent to the specified SP. Such a procedure would deny the browser extension the opportunity to intercept the communication and give the user the choice whether or not to allow the token to be sent to the RP (as is normally the case for CardSpace). We therefore require a small modification to the way that the Liberty-enabled IdP operates. The IdP must be modified to redirect the user agent to a web page at the IdP server, rather than at the RP, thereby giving the browser extension control. This could be achieved by requiring the IdP to set the action attribute[19] of the HTML form to an empty string or to #[20]. In step 11, the browser extension resets the action attribute to the URL address of the appropriate CardSpace RP, and, after obtaining user permission to release the authentication token to the given RP, automatically submits the HTML form, redirecting the user agent to the RP website. This small change to the normal operation of the Liberty IdP helps to enhance user control (see sections 4.4 and 5.3.3), hence implementing Microsoft's first identity law [6, 10, 17, 34]. It merits mentioning that both the LBP and LEC profiles require the SP URL address to be specified as the value of the '<lib:AssertionConsumerServiceURL>' statement in the SAML authentication request [12]. To keep the changes at the IdP side to a minimum, the value of this field could be set to #, implicitly instructing the IdP to include this value instead of the SP's URL in the action attribute of the HTML form sent back to the user agent. Further discussion of the LBP and LEC profiles is given in section 4.2.

Given that we have assumed that the RP supports SAML 2.0 tokens, there is no need to modify the proof-of-possession data since the RP can use the Liberty ID-FF supported HoK [45] method (which can be symmetric or asymmetric) to express its proof-of-possession requirements. However, a

[19] Observe that, in the standard LBP profile case, the action attribute of the HTML form is set to the URL address of the requesting SP, and the IdP redirects the user agent to that SP.

[20] Note that whilst this has been shown to work successfully with IE7 and IE8, other browsers may not support an action attribute of an empty string or hash (#); hence setting the action attribute to a relative URL for the IdP login page may be required for such browsers.

symmetric proof key should only be used if the user is willing to disclose the identity of the RP to the IdP, and if the RP holds a valid certificate. For browser-based applications (and also where no proof-of-possession is needed), the proposed scheme supports bearer tokens [13, 36, 38].

Finally observe that the additional steps above can be integrated into the current CardSpace framework relatively easily, as the prototype implementation shows.

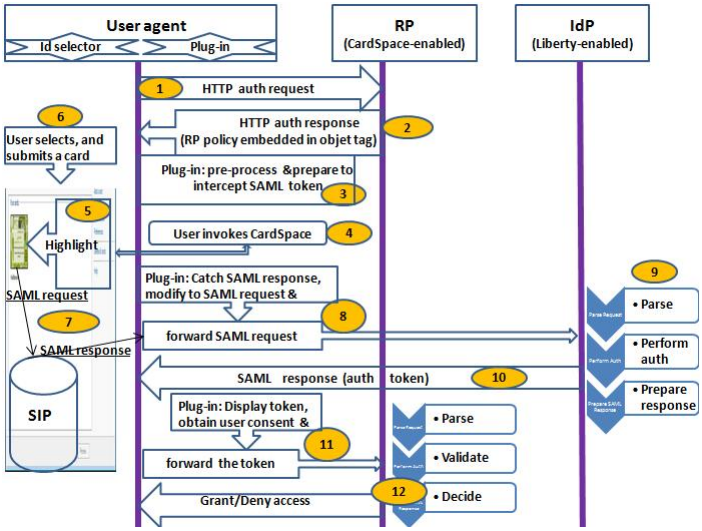


Figure 3: Protocol exchanges

4. DISCUSSION AND ANALYSIS

We now consider implementation and applicability issues of the scheme.

4.1 Differences in Scope

There is a key difference between the Liberty ID-FF and CardSpace frameworks. CardSpace allows IdPs to assert a range of attributes about users (including simple authentication assertions), whereas Liberty ID-FF only supports authentication assertions. In CardSpace, the user attributes to be asserted are specified in a SAML attribute statement contained in a SAML request that can be processed by the local SIP or the remote CardSpace-enabled IdP. However, a Liberty ID-FF conformant IdP is only required to generate SAML authentication statements (and not assert user attributes), which gives rise to an interoperation problem. Two possible solutions are as follows.

1. It could be assumed that the CardSpace RP is only concerned with user authentication (which seems likely to be a common case). In such a case a LibertyCard contains the IdP URL and the trigger word, and a LibertyCard will only be used if the RP policy requests an assertion solely of the PPID attribute, e.g. by including '`http://schemas.xmlsoap.org/ws/2005/05/identity/claims/privatepersonalidentifier`' in the list of required claims. In such a case, the browser extension will modify the RP policy to ensure it includes the fields used in LibertyCards (see section 3.2.3). On selection of a LibertyCard, the browser extension (as in step 8 in section 3.2.5) intercepts, creates and forwards a SAML authentication request to the user-selected IdP. While this is a straightforward task, it limits the scope of applicability of the scheme.

2. Alternatively, it could be assumed that the CardSpace-enabled RP is concerned with both user authentication and the assertion of user attributes, and that the RP policy permits assertions (for user attributes only) to be provided by the SIP. In this case, along with requiring the PPID, the RP security policy would also specify the attributes required, leading the identity selector to highlight the user-created LibertyCards that satisfy the requirements. To ensure that no changes are required at either the RP or the IdP, the browser extension could store attribute assertions created by the SIP. The browser extension would then create the SAML authentication request according to the Liberty ID-FF standards, and forward it to the specified IdP. When the browser extension receives the response containing the authentication assertion from the IdP, it would add appropriate attribute assertion(s) from its local cache and then forward the entire package to the RP. However, if the RP security policy dictates that security tokens must be wholly signed by the issuing IdP, then this solution would fail.

The prototype implementation, described in section 5, implements the first approach.

4.2 Liberty Profiles

To maximise applicability, the integration scheme supports both the Liberty browser post (LBP) and Liberty-enabled client (LEC) profiles, introduced in section 2.2.3. However, the prototype described in section 5 only implements the LBP profile.

In the LEC profile, interactions between a user agent and an IdP are SOAP-based, and the protocol messages include Liberty-specified HTTP headers indicating that the sender is Liberty-enabled. Under the LEC profile, the client must submit the authentication request to the IdP as a SOAP request, whereas, when using the LBP profile, the request can be embedded in an HTML form containing a field called 'LAREQ' set to the '<lib:AuthnRequest>' protocol message [12, 14]. In order to support both profiles, the integration software must therefore be capable of supporting both forms of communications with the IdP.

The two profiles have many properties in common. For example, they both support SAML. In both profiles, the HTML form containing the authentication response must be sent to the user agent using an HTTP POST; this form must contain the field 'LARES' with value equal to the authentication response, as defined in the Liberty protocol schema [14]. In both profiles, the value of the 'LARES' field must be encoded using a base-64 transformation [23].

Despite the differences between the profiles, the protocol steps given in section 3.2.5 apply to both profiles.

4.3 Triggering the Browser Extension

As stated in section 3.1, the means by which the integration software is triggered needs to be chosen carefully. The means included in the scheme described in section 3.2.3 is to include a trigger sequence (e.g. the word 'Liberty') in a specific field of a LibertyCard. This is also the method used

in the prototype described in section 5. However, other approaches could be used, e.g. as follows.

1. The browser extension could start whenever CardSpace is triggered. When a user submits an InfoCard, the browser extension would offer the user two options (based on HTML forms): to continue to use CardSpace as usual, or to use a Liberty-enabled IdP. This approach gives a greater degree of user control, and hence implements Microsoft's first identity law [6, 10, 17, 34]. However, it is not particularly convenient, since it would always require users to choose whether or not to use the integration software.

2. Alternatively, the browser extension could ask the user whether they wish to activate the integration protocol (e.g. via a JavaScript pop-up box). This has advantages and disadvantages similar to those of the first alternative.

4.4 Token Forwarding

The means by which the security token is forwarded to the RP needs to be chosen carefully. We refer to the numbered protocol steps given in section 3.2.5.

The responsibility for delivering the security token could be given to the Liberty IdP (as is normally the case when using the LBP profile). In this case the RP address could be added to the SAML authentication request (as prepared in step 8) so that the IdP knows which RP it must forward the token to (again as is normally the case for the Liberty profiles). Although this would avoid the need for changes to the normal operation of the Liberty IdP and potentially also help auditing, such an approach has privacy implications since the IdP would learn the identity of the RP.

As a result, as specified in step 11 of the proposed scheme, the responsibility for sending the security token to the RP is given to the user agent. Thus a means is required for giving the browser extension the address of the RP, so that it can forward the token. We next consider three possible ways in which the RP address might be made available.

- The RP address could be stored in the browser extension itself. Whilst this puts the user in control, it is not user-friendly, as it would require users to manually add the address of each RP into the code of the browser extension.

- After the security token is returned from the Liberty IdP, the browser extension could ask the user to enter the RP address, e.g. using a JavaScript pop-up box or an HTML form. This has advantages and disadvantages similar to those of the previous alternative.

- The browser extension could store the RP address encrypted in a cookie as part of step 3, so that the browser extension can obtain the address in step 11. In order to adhere to cookie security rules [31], this must be done in such a way that the browser believes it is communicating with the same domain when the cookie is set and when it is retrieved[21].

  To achieve this, the browser extension encrypts and stores the RP address in a cookie in step 3, before the identity selector is invoked. As part of step 8, the browser extension retrieves the encrypted value from the cookie and sends it to the IdP as a hidden HTML variable in an HTML form or as a query URL parameter. As part of step 10, the IdP returns the encrypted RP address to the user agent (again as a hidden form variable or as a URL parameter[22]). In step 11, the browser extension obtains the encrypted value and decrypts it to obtain the RP address.

  Note that the IdP is unable to read the RP address, hence protecting user privacy, since it is encrypted using a key known only to the browser extension. If the IdP, however, needs the RP address for auditing purposes (e.g. for legal reasons), or the IdP policy requires the disclosure of the RP identity (e.g. so it can encrypt the security token using the RP's public key), then the RP address could be sent in plain text to the IdP.

[21] Note that creation of and access to the cookie can be handled by the browser extension transparently to RPs and IdPs.

4.5 Defeating Phishing

Use of LibertyCards helps to mitigate the risk of phishing. The LibertyCard contains the URL of the IdP entered by the user, and the user will only be forwarded to that IdP, i.e. the RP will not be able to redirect the user to an IdP of its choice. By contrast, in the Liberty artifact and Liberty browser post profiles (and in OpenID [44, 48]), a malicious SP might redirect a user to a fake IdP, which could then capture the user credentials. This is a particular threat for static credentials, such as usernames and passwords.

4.6 Integration at the Client Side

Some IdPs and RPs/SPs may not be prepared to accept the burden of supporting two identity management systems simultaneously, at least unless there is a significant financial incentive. Currently, major Internet players, such as MSN[23], do not provide any means of interoperating between identity management systems. As a result, a client-side technique for supporting interoperation could be practically useful.

In addition, building the integration scheme on the client means that the performance of the server is not affected, since the integration overhead is handled by the client. Such an approach also reduces the load on the network.

4.7 STS-enhanced RPs

STS-enhanced RPs are not supported by the integration scheme. This is because use of an STS involves direct communication (i.e. not via a browser) between the CardSpace identity selector and the RP STS [27], which the integration browser extension is currently not capable of intercepting. For example, the identity selector directly contacts the RP STS to obtain its security policy using WS-MetadataExchange.

In the scheme described in this paper, the interaction with the RP uses HTTP/HTML via a web browser. This is a simpler and probably more common scenario for RP interactions [19]. As discussed in section 2.1.3, an RP security policy can be expressed using HTML, and both the policy and the security token can be exchanged using HTTP/S. Therefore, to act as a CardSpace-enabled RP, a website is not

[22] The use of HTML forms (with the POST method) is preferable to query URL parameters, since the latter may suffer from size restrictions; hence the former approach is used in the prototype implementation described in section 5.

[23] `http://www.msn.com`

required to implement any of the WS-* specifications [19, 27].

4.8 Applicability of the Scheme

Although the proposed integration scheme is presented as Liberty-specific, we suspect that the scheme could also be applicable for SAML-compliant IdPs; this, nevertheless, requires certain modifications to the current scheme. For example, the technical differences[24] between Liberty ID-FF 1.2 and SAML 2.0 must be carefully examined. However, given that SAML 2.0 is the successor to SAML 1.1, Liberty ID-FF 1.2 and Shibboleth 1.3 [15], a mapping seems likely to be possible.

Reconfiguring the integration scheme to interoperate with SAML-aware IdPs potentially significantly increases its applicability and practicality. For example, the exchange of identity attributes, which is not supported under the current scheme, would then be feasible. The reconfiguration of the scheme remains possible future work.

5. PROTOTYPE REALISATION

This section provides technical details of a prototype implementation of the integration scheme when used with the Liberty browser post profile. A number of prototype-specific properties and possible limitations of the current prototype are also described.

5.1 User Registration

Prior to use, the user must have accounts with a CardSpace RP and a Liberty-enabled IdP. The user must also create a LibertyCard for the relevant Liberty IdP (or it could be created at the time of use). This involves invoking the CardSpace identity selector and inserting the URL of the target Liberty IdP in the *web page* field[25] and the trigger word (*Liberty*) in the *city* field. For ease of identification, the user can give the personal card a meaningful name, e.g. of the target IdP site. The user can also upload an image for the card, e.g. containing the logo of the intended IdP or simply of Liberty. When a user wishes to use a particular Liberty IdP, the user simply chooses the corresponding card. An example of a LibertyCard is shown in figure 4.

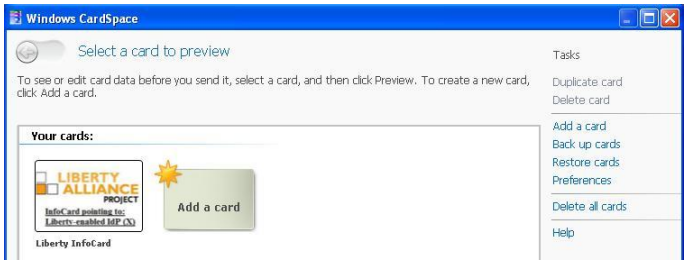


Figure 4: A LibertyCard

[24] `https://spaces.internet2.edu/display/SHIB/SAMLLibertyDiffs`

[25] The *web page* field was chosen to contain the Liberty IdP URL since it seems the logical choice; however, this is an implementation option.

5.2 Implementation Details

The prototype, described in section 5.3, was coded as a client-side plug-in[26] using JavaScript [40, 42], chosen to maximise portability. Indeed, JavaScript[27] appears to be the most widely browser-supported and commonly used client-side scripting language across the Web today. Use of browser-specific client-side scripting languages, e.g. VBScript, was ruled out to ensure the widest applicability [20].

The implementation uses the Document Object Model (DOM) [32] to inspect and manipulate HTML [43] pages and XML [9] documents. Since the DOM defines the objects and properties of all document elements and the methods to access them, a client-side scripting language can read and modify the contents of a web page or completely alter its appearance [20].

The prototype does not use any of the published CardSpace application programming interfaces (APIs). This will ease migration of the plug-in to other CardSpace-like systems such as the Linux/Mac-based DigitalMe[28] and the Firefox/Safari InfoCard extensions.

5.3 Operation of the Prototype

In this section we consider specific operational aspects of the prototype. We refer throughout to the numbered protocol steps given in section 3.2.5.

5.3.1 Prototype-specific Operational Details

In step 3, before the HTML login page is displayed, the plug-in uses the DOM to perform the following processes.

1. The plug-in scans the web page in the following way[29].
   (a) It searches through the HTML elements of the web page to detect whether any HTML forms are present. If so, it searches each form, scanning through each of its child elements for an HTML object tag.
   (b) If an object tag is found, it retrieves and examines its type. If it is of type 'application/x-informationCard' (which signals website support for CardSpace), it continues; otherwise it aborts.
   (c) It then searches through the param tags (child elements of the retrieved CardSpace object tag) for the 'requiredClaims' tag, which lists the claims required by the RP security policy.
   (d) If the required claims include attributes other than the PPID claim, then the plug-in terminates, giving CardSpace the opportunity to operate normally. However, if only the PPID claim is requested, then the plug-in adds the *city* and *web page* claims to the 'requiredClaims' tag, marking them as mandatory (see section 3.2.3).

[26] We use the term *plug-in* to refer to any client-side browser extension, such as a user script, plug-in, etc.

[27] Throughout the description the term *JavaScript* is, for simplicity, used to refer to all variants of the language.

[28] `http://code.bandit-project.org/trac/wiki/DigitalMe`

[29] The relevant user guide [27] specifies two HTML extension formats for invoking an identity selector from a web page, both of which include placing the CardSpace object tag inside an HTML form. This motivates the choice of the web page search method.

2. The plug-in adds a JavaScript function to the head section of the HTML page to intercept the XML-based authentication token before it is sent back to the RP (such a token will be sent by the identity selector in step 8).

3. The plug-in obtains the current action attribute of the CardSpace HTML form, encrypts it using AES [39] with a secret key known only to the plug-in, and then stores it in a cookie. This attribute specifies the URL address of a web page at the CardSpace-enabled RP to which the authentication token must be forwarded for processing. If the obtained attribute is not a fully qualified domain name address, the JavaScript inherent properties, e.g. *document.location.protocol* and *document.location.host*, are used to help reconstruct the full URL address.

4. After storing it, the plug-in changes the current action attribute of the CardSpace HTML form to point to the newly created 'interception' function (see step 2 above).

5. The plug-in creates and appends an 'invisible' HTML form to the HTML page to be used later for sending the SAML token request to the Liberty-enabled IdP.

In step 8 the plug-in uses the DOM to perform the following steps.

1. It intercepts the RSTR message sent by the CardSpace identity selector using the added function (see above).

2. It parses the intercepted token. If the *city* field contains the word Liberty, the plug-in proceeds; if not, normal operation of CardSpace continues. It also reads the *web page* field to discover the URL address of the IdP. In addition, all other fields, including the PPID and InfoCard public key with its digital signature, are parsed. The *city*, *web page*, and *PPID* fields are contained in a SAML attribute statement, whereas the public key and signature values are contained in a SAML signature statement.

   The plug-in uses an XML parser built into the browser to read and manipulate the intercepted XML token. The plug-in passes the token to the parser, which reads it and converts it into an XML DOM object that can be accessed and manipulated by JavaScript. The DOM views the XML token as a tree-structure, thereby enabling JavaScript to traverse the DOM tree to read (and possibly modify) the content of the token elements. New elements can also be created where necessary.

3. It converts the token format from a SAML response message into a SAML request message, compatible with Liberty-conformant IdPs supporting the browser post profile. This involves converting a SAML 1.1-based RSTR into a SAML 2.0 authentication request. Moreover, as outlined in section 3.2.2, the plug-in adds the PPID and the InfoCard public key along with its signature to the SAML request message, because the token must be signed by the Liberty-enabled IdP to provide integrity and authenticity services.

4. It writes the entire SAML request message as a hidden variable into the invisible HTML form created earlier.

5. It retrieves the encrypted RP URL from the cookie, and writes it into the invisible form as a hidden variable.

6. It writes the URL address of the Liberty IdP into the action attribute of the invisible form.

7. It auto-submits the HTML form (transparently to the user), using the JavaScript method 'click()' on the 'submit' tag.

5.3.2 Liberty IdP-specific Details

For steps 8 to 10, we have created an experimental website to act as a Liberty-enabled IdP supporting the Liberty browser post profile. PHP is used to enable the IdP to parse the SAML request and perform the user authentication. The user credentials, i.e. username and password, that the IdP uses to authenticate the user are stored in a MySQL database. They are salted, hashed with SHA-1, and protected against SQL injection attacks. PHP supports a variety of XML parsers, such as XML DOM, Expat parser, and SimpleXML. The prototype uses XML DOM.

5.3.3 User Consent and Token Forwarding

In step 11, the plug-in operates as follows.

1. It obtains the encrypted value of the RP URL from the appropriate HTML hidden variable, decrypts it using its internally stored secret key, and inserts it into the action attribute of the HTML form carrying the received SAML token.

2. The plug-in then displays the token to the user and requests consent to proceed. The displayed token indicates the types of information the authentication token is carrying, as well as the exact URL address of the RP to which the token will be forwarded. The JavaScript 'confirm()' pop-up box is used to achieve this.

3. If the user approves the token, the plug-in seamlessly submits it to the RP using the JavaScript 'click()' method.

5.3.4 CardSpace RP-specific Details

To test the prototype, we built an experimental website to act as a CardSpace-enabled RP. On receipt of the SAML authentication token, the RP uses PHP in step 11 to parse and validate the received token. As is the case with the Liberty IdP, the user identifying data is salted, hashed and stored in a MySQL database that is resistant to SQL injection attacks. The validation process includes verifying the digital signatures and checking the conditions, e.g. time stamps, included in the token. The PPID and the InfoCard public key in the token are compared to the values stored in the RP database, and the authentication status is also checked.

5.3.5 Other Issues

The JavaScript-driven plug-in was built using IE7PRO, an IE extension, chosen to expedite the prototype implementation. Users of the prototype must therefore install IE7PRO, freely available at the IE7PRO website[30], prior to installing the integration plug-in. To enable or disable

[30] `http://www.ie7pro.com`

the integration prototype, a user can simply tick or un-tick the appropriate entry in the 'IE7PRO Preferences' interface. This provides the means to achieve the final objective listed in section 3.2.4.

Finally note that the integration plug-in does not require any changes to default IE security settings, thereby avoiding potential vulnerabilities resulting from lowering browser security settings.

5.4 Limitations

The current version of the prototype has not been tested with CardSpace relying parties using TLS/SSL. Therefore, we are not able to provide precise operational and performance details in this case.

If the RP has a certificate, then the identity selector will, by default, encrypt the SAML-based RSTR message using the public key of the requesting RP. Clearly, the plug-in does not have access to the RP's private key, and hence will not be able to decrypt the token. Therefore, it will not know whether to trigger the integration protocol, and will be unable both to discover which IdP it must contact, and to obtain the user identifier (the PPID).

One solution to these issues would be for the plug-in to first ask the user whether the integration protocol should be activated (e.g. via a JavaScript prompt window), and, if so, it should then forward the SAML token to the RP and notify the RP to wait for another token. The RP should decrypt the token, read the PPID, and then wait. At the same time, the plug-in should prompt the user to enter the URL of the Liberty-enabled IdP, and then create and send a SAML request message to the Liberty IdP, which authenticates the user and responds with a SAML response token. The plug-in could then, optionally, seek user consent, and, if the user approves, the plug-in would then forward the token to the RP. The RP must issue the plug-in with a nonce (and a time-stamp) which the plug-in sends back with the second token to both link the two tokens together and help protect against replay and guessing attacks.

One of the most obvious drawbacks to this solution is that it requires changes at the CardSpace-enabled RP, as the RP must be reconfigured to accept two tokens. However, this would not be a major change since both tokens will be constructed using SAML, and since the RP is not required to directly contact the Liberty-enabled IdP. Therefore, the major overheard remains with the client. Nevertheless, we are working on a revised version of the prototype that is fully compatible with SSL/TLS encryption but without the requirement of RP reconfiguration.

The integration plug-in must scan every browser-rendered web page to detect whether it supports CardSpace, and this may affect system performance. However, informal tests on the prototype suggest that this is not a serious issue. In addition, the plug-in can be configured so that it only operates with certain websites.

The integration plug-in has not been tested with CardSpace 2.0, because it was completed well before its release. Therefore, we are not yet able to provide precise operational details for this version.

Finally note that some older browsers (or browsers with scripting disabled) may not be able to run the integration plug-in, as it was built using JavaScript. However, most modern browsers support JavaScript (or ECMAscript), and hence building the prototype in JavaScript is not a major usability obstacle.

6. RELATED WORK

The Bandit[31] and Concordia[32] projects are currently developing open source technologies to support interoperation between identity management systems. Unlike the integration scheme proposed in this paper, these systems are not based on client-side models. Concordia has proposed a CardSpace and SAML/WS-Federation integration model. This could be used as the basis for supporting Liberty/CardSpace interoperation by taking advantage of the similarities between the Liberty ID-FF SSO profiles and the SAML SSO profiles.

Another scheme supporting interoperation between CardSpace and Liberty has been proposed by Jørstad et al. [29]. In this scheme, the IdP is responsible for supporting interoperation. The IdP must therefore perform the potentially onerous task of maintaining two different identity management schemes. In addition, this scheme requires the user to possess a mobile phone supporting the Short Message Service (SMS). Moreover, the IdP must always perform the same user authentication technique, regardless of the identity management system the user is attempting to use. The IdP simply sends an SMS to the user, and, in order to be authenticated, the user must confirm receipt of the SMS. This confirmation is also an implicit user approval for the IdP to send a security token to the RP. By contrast, the scheme proposed in this paper does not require use of a handheld device, and does not enforce a specific authentication method.

Finally, we observe that Liberty is apparently also working on a scheme somewhat similar to that described here. No specifications have yet been released, but the plans are described in a presentation available at the Liberty website[33].

7. CONCLUSIONS AND FUTURE WORK

We have proposed a means of interoperation between two leading identity management systems, namely CardSpace and Liberty. CardSpace users are able to obtain an assertion token from a Liberty-enabled identity provider that satisfies the security requirements of a CardSpace-enabled relying party. The scheme uses a client-side browser extension, and requires no major changes to servers. It uses the CardSpace identity selector interface to integrate Liberty identity providers with CardSpace relying parties. The scheme extends the use of personal cards to allow for such interoperability.

The integration scheme takes advantage of the similarity between the Liberty ID-FF and the CardSpace frameworks, and this should help to reduce the effort required for full system integration. Also, implementation of the scheme does not require technical co-operation between Microsoft and Liberty.

Planned future work includes investigating the possibility of using the CardSpace identity selector to enable access to identity providers of other identity management systems, such as OpenID and Shibboleth. In addition, we also plan to

[31] http://www.bandit-project.org

[32] http://www.projectconcordia.org

[33] http://www.projectliberty.org/liberty/content/download/4541/31033/file/20080ICP-Cardspace-DIDW.pdf

investigate the possibility of extending the proposed integration protocol to support CardSpace-enabled relying parties that employ security token services.

8. REFERENCES

[1] W. A. Alrodhan and C. J. Mitchell. A client-side CardSpace-Liberty integration architecture. In *Proceedings of the 7th Symposium on Identity and Trust on the Internet (IDtrust 08)*, pages 1–7. ACM, New York, NY, USA, 2008.

[2] S. Anderson et al. *Web Services Trust Language (WS-Trust)*. Actional Corporation, BEA Systems, Computer Associates International, International Business Machines Corporation, Layer 7 Technologies, Microsoft Corporation, Oblix, OpenNetwork Technologies, Ping Identity Corporation, Reactivity, RSA Security, and VeriSign, 2005. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-trust/ws-trust.pdf`.

[3] S. Bajaj et al. *Web Services Policy Framework (WS-Policy)*. BEA Systems, International Business Machines Corporation, Microsoft Corporation, SAP AG, Sonic Software, and VeriSign, 2006. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-polfram/ws-policy-2006-03-01.pdf`.

[4] K. Ballinger et al. *Web Services Metadata Exchange (WS-MetadataExchange)*. BEA Systems, Computer Associates International, International Business Machines Corporation, Microsoft Corporation, SAP AG, Sun Microsystems, and webMethods, 2006. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-mex/metadataexchange.pdf`.

[5] A. Berger. *Identity Management Systems — Introducing Yourself to the Internet*. VDM Verlag, Saarbrücken, Germany, 2008.

[6] V. Bertocci, G. Serack, and C. Baker. *Understanding Windows CardSpace: An Introduction to the Concepts and Challenges of Digital Identities*. Addison-Wesley, Reading, Massachusetts, USA, 2008.

[7] K. Bhargavan, C. Fournet, A. D. Gordon, and N. Swamy. Verified implementations of the information card federated identity-management protocol. In *Proceedings of the 2008 ACM symposium on Information, Computer and Communications Security (ASIACCS 08)*, pages 123–135. ACM, New York, NY, USA, 2008.

[8] D. Birch. *Digital Identity Management: Technological, Business and Social Implications*. Gower Publishing, Farnham, UK, 2007.

[9] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau (editors). *Extensible Markup Language (XML) 1.0*. W3C Recommendation, 5th edition, 2008. `http://www.w3.org/TR/xml/`.

[10] K. Cameron. *The Laws of Identity*. Microsoft Corporation, 2005. `http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf`.

[11] K. Cameron and M. B. Jones. *Design Rationale behind the Identity Metasystem Architecture*. Microsoft Corporation, 2006. `http://www.identityblog.com/wp-content/resources/design_rationale.pdf`.

[12] S. Cantor, J. Kemp, and D. Champagne (editors). *Liberty ID-FF Bindings and Profiles Specification*. Liberty Alliance Project, 2004. `http://www.projectliberty.org/liberty/content/download/319/2369/file/draft-liberty-idff-bindings-profiles-1.2-errata-v2.0.pdf`.

[13] S. Cantor, J. Kemp, R. Philpott, and E. Maler (editors). *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS, 2005. `http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf`.

[14] S. Cantor and J. Kemp (editors). *Liberty ID-FF Protocols and Schema Specification*. Liberty Alliance Project, 2005. `http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_ff_1_2_specifications`.

[15] S. Cantor (editor). *Shibboleth Architecture — Protocols and Profiles*, 2005. `http://shibboleth.internet2.edu/docs/internet2-mace-shibboleth-arch-protocols-200509.pdf`.

[16] D. Chadwick. FileSpace: an alternative to CardSpace that supports multiple token authorisation and portability between devices. In *Proceedings of the 8th Symposium on Identity and Trust on the Internet (IDtrust 09)*, pages 94–102. ACM, New York, NY, USA, 2009.

[17] D. W. Chadwick. Federated identity management. In A. Aldini, G. Barthe, and R. Gorrieri, editors, *Foundations of Security Analysis and Design V, FOSAD 2007/2008/2009 Tutorial Lectures*, volume 5705 of *Lecture Notes in Computer Science*, pages 96–120. Springer, Berlin/Heidelberg, Germany, 2009.

[18] D. W. Chadwick and G. Inman. Attribute aggregation in federated identity management. *IEEE Computer*, 42(5):33–40, 2009.

[19] D. Chappell. *Introducing Windows CardSpace*. MSDN, 2006. `http://msdn.microsoft.com/en-us/library/aa480189.aspx`.

[20] N. Daswani, C. Kern, and A. Kesavan. *Foundations of Security: What Every Programmer Needs to Know*. Apress, Berkeley, CA, USA, 2007.

[21] G. Della-Libera et al. *Web Services Security Policy Language (WS-Security Policy)*. International Business Machines Corporation, Microsoft Corporation, RSA Security, and VeriSign, 2005. `http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-secpol/ws-secpol.pdf`.

[22] R. Fielding, J. Getty, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. *Hypertext Transfer Protocol — HTTP/1.1*. RFC 2616, The Internet Society, 1999. `http://tools.ietf.org/html/rfc2616`.

[23] N. Freed and N. Borenstein. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. RFC 2045, Internet Engineering Task Force, 1996. `http://www.ietf.org/rfc/rfc2045.txt`.

[24] S. Gajek, J. Schwenk, M. Steiner, and C. Xuan. Risks of the CardSpace protocol. In *Proceedings of the 12th International Conference on Information Security (ISC 09)*, pages 278–293. Springer-Verlag, Berlin/Heidelberg, Germany, 2009.

[25] M. Gudgin, M. Hadley, N. Mendelsohn, J.-J. Moreau, H. F. Nielsen, A. Karmarkar, and Y. Lafon (editors). *SOAP Version 1.2 Part 1: Messaging Framework*. W3C Recommendation, 2007. http://www.w3.org/TR/soap12-part1/.

[26] P. Hallam-Baker and E. Maler (editors). *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.0*. OASIS, 2002. http://www.oasis-open.org/specs/#samlv1.0.

[27] M. B. Jones. *A Guide to Using the Identity Selector Interoperability Profile V1.5 within Web Applications and Browsers*. Microsoft Corporation, 2008.

[28] M. B. Jones and M. McIntosh (editors). *Identity Metasystem Interoperability Version 1.0 (IMI 1.0)*. OASIS Standard, 2009. http://docs.oasis-open.org/imi/identity/v1.0/identity.html.

[29] I. Jørstad, D. Van Thuan, T. Jønvik, and D. Van Thanh. Bridging CardSpace and Liberty Alliance with SIM authentication. In *Proceedings of the 10th International Conference on Intelligence in Next Generation Networks (ICIN 07)*, pages 8–13. Adera, BP 196 - 33608 Pessac Cedex, France, 2007.

[30] S. Kellomäki and R. Lockhart (editors). *Liberty ID-SIS Employee Profile Service Specification*. Liberty Alliance Project, 2005. http://www.projectliberty.org/liberty/content/download/1031/7155/file/liberty-idsis-ep-v1.1.pdf.

[31] D. Kristol. *HTTP State Management Mechanism*. RFC 2045, Internet Engineering Task Force, 2000. http://tools.ietf.org/html/rfc2965.

[32] A. Le Hors, P. L. Hégaret, L. Wood, G. Nicol, J. Robie, M. Champion, and S. Byrne (editors). *Document Object Model (DOM) Level 2 Core Specification*. W3C Recommendation, 2000. http://www.w3.org/TR/DOM-Level-2-Core/.

[33] E. Maler, P. Mishra, and R. Philpott (editors). *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1*. OASIS, 2003. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf.

[34] M. Mercuri. *Beginning Information Cards and CardSpace: From Novice to Professional*. Apress, New York, USA, 2007.

[35] Microsoft Corporation. *Microsoft's Vision for an Identity Metasystem*, May 2005. http://msdn.microsoft.com/en-us/library/ms996422.aspx.

[36] Microsoft Corporation and Ping Identity Corporation. *An Implementer's Guide to the Identity Selector Interoperability Profile v1.5*, 2008.

[37] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker (editors). *Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)*. OASIS Standard Specification, 2006. http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SOAPMessageSecurity.pdf.

[38] A. Nanda and M. B. Jones. *Identity Selector Interoperability Profile V1.5*. Microsoft Corporation, 2008. http://www.identityblog.com/wp-content/resources/2008/Identity_Selector_Interoperability_Profile_V1.5.pdf.

[39] National Institute of Standards and Technology (NIST). *Announcing the Advanced Encryption Standard (AES)*, FIPS 197, November 2001. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[40] T. Negrino and D. Smith. *JavaScript and Ajax for the Web: Visual QuickStart Guide*. Peachpit Press, Berkeley, CA, USA, 7th edition, 2008.

[41] R. Oppliger, S. Gajek, and R. Hauser. Security of Microsoft's identity metasystem and CardSpace. In *Proceedings of the Kommunikation in Verteilten Systemen (KiVS 07)*, pages 63–74. VDE Publishing House, Berlin, Germany, 2007.

[42] T. A. Powell and F. Schneider. *Javascript: The Complete Reference*. McGraw-Hill Osborne Media, Berkeley, CA, USA, 2nd edition, 2004.

[43] D. Raggett, A. L. Hors, and I. Jacobs (editors). *HTML 4.01 Specification*. W3C Recommendation, 1999. http://www.w3.org/TR/html401/.

[44] D. Recordon, L. Rae, and C. Messina. *OpenID: The Definitive Guide*. O'Reilly Media, Sebastopol, CA, USA, 2010.

[45] T. Scavo (editor). *SAML V2.0 Holder-of-Key Assertion Profile Version 1.0*. OASIS, 2009. http://www.oasis-open.org/committees/download.php/34962/sstc-saml2-holder-of-key-cd-03.pdf.

[46] D. Todorov. *Mechanics of User Identification and Authentication: Fundamentals of Identity Management*. Auerbach Publications, New York, USA, 2007.

[47] J. Tourzan and Y. Koga (editors). *Liberty ID-WSF Web Services Framework Overview*. Liberty Alliance Project, 2005. http://www.projectliberty.org/liberty/content/download/1307/8286/file/liberty-idwsf-overview-v1.1.pdf.

[48] R. Ur Rehman. *Get Ready for OpenID*. Conformix Technologies, Chesterbrook, Pennsylvania, USA, 2008.

[49] T. Wason (editor). *Liberty ID-FF Architecture Overview*. Liberty Alliance Project, 2003. http://www.telenor.com/rd/idm/liberty-idff-arch-overview-v1.2.pdf.

[50] G. Williamson, D. Yip, I. Sharoni, and K. Spaulding. *Identity Management: A Primer*. MC Press, Big Sandy, TX, USA, 2009.

CardSpace-Liberty Integration for CardSpace Users

IDtrust 2010

13/4/2010

Haitham Al-Sinani
Information Security Group
Royal Holloway, University of London

H.Al-Sinani@rhul.ac.uk
http://isg.rhul.ac.uk/

Acknowledgments

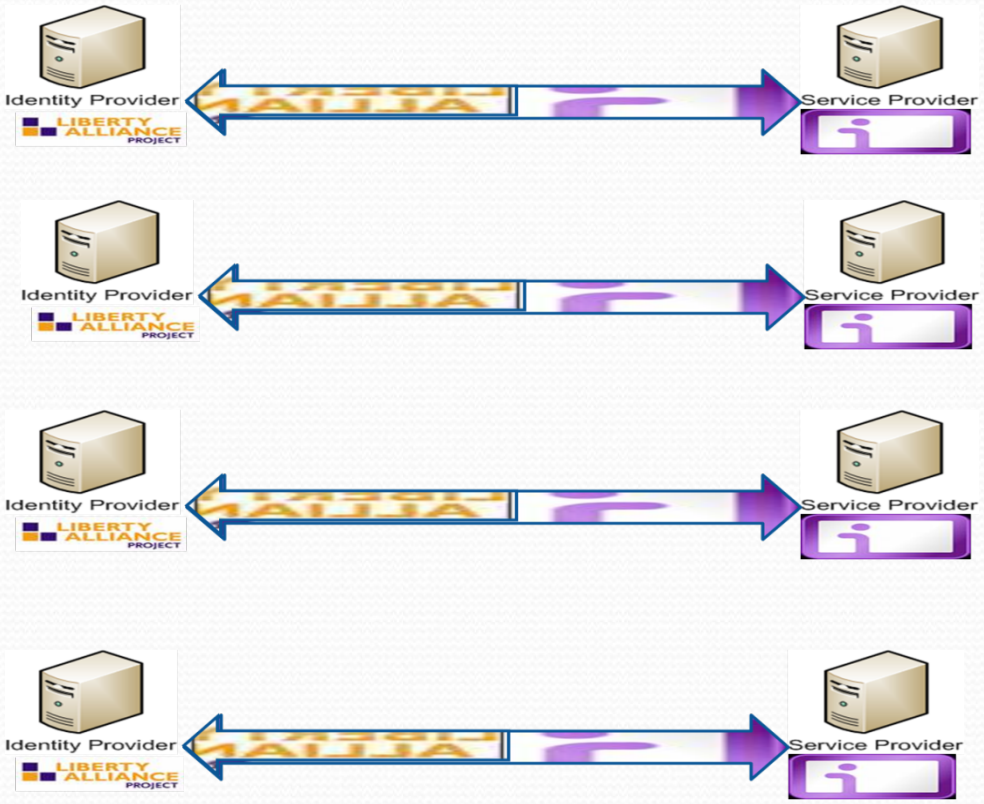
Haitham Al-Sinani is sponsored by the Diwan of Royal Court, Sultanate of Oman.

Joint work with Waleed Alrodhan and Chris Mitchell.

Third party images are gratefully acknowledged, and hyper-linked to their original sources.

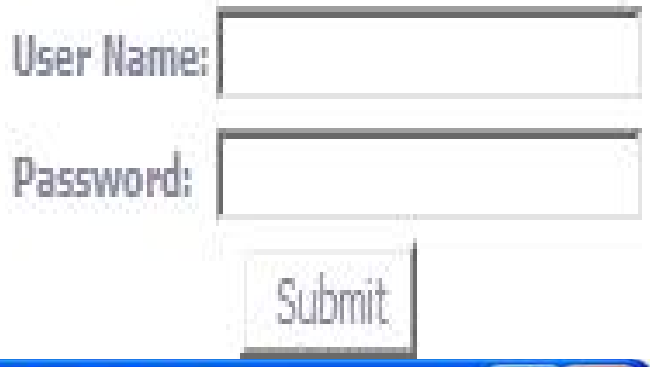
Agenda

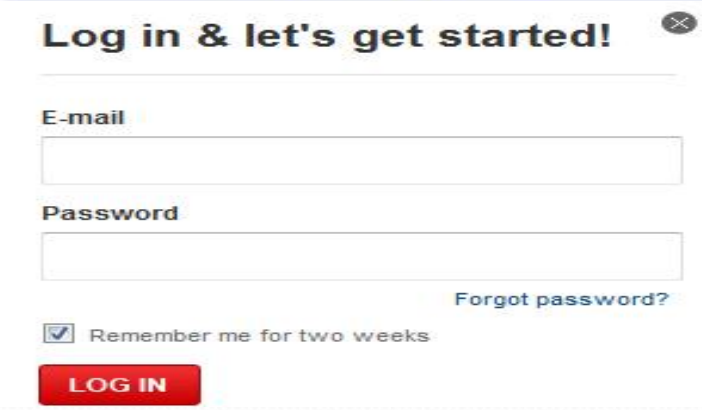
1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A



User Identities

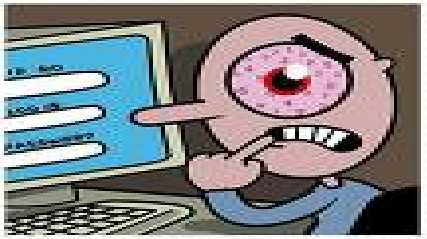
Multiple identities for multiple accounts



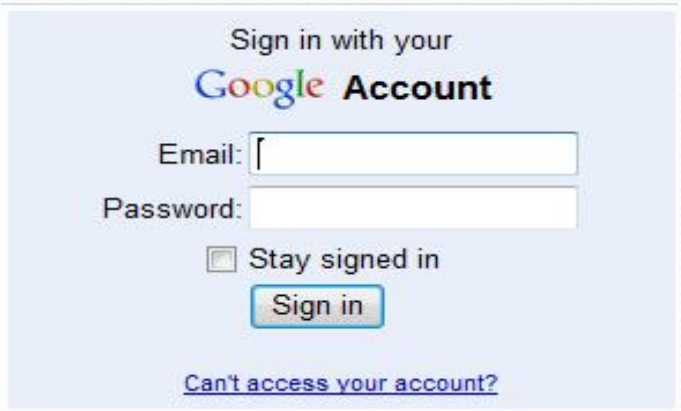


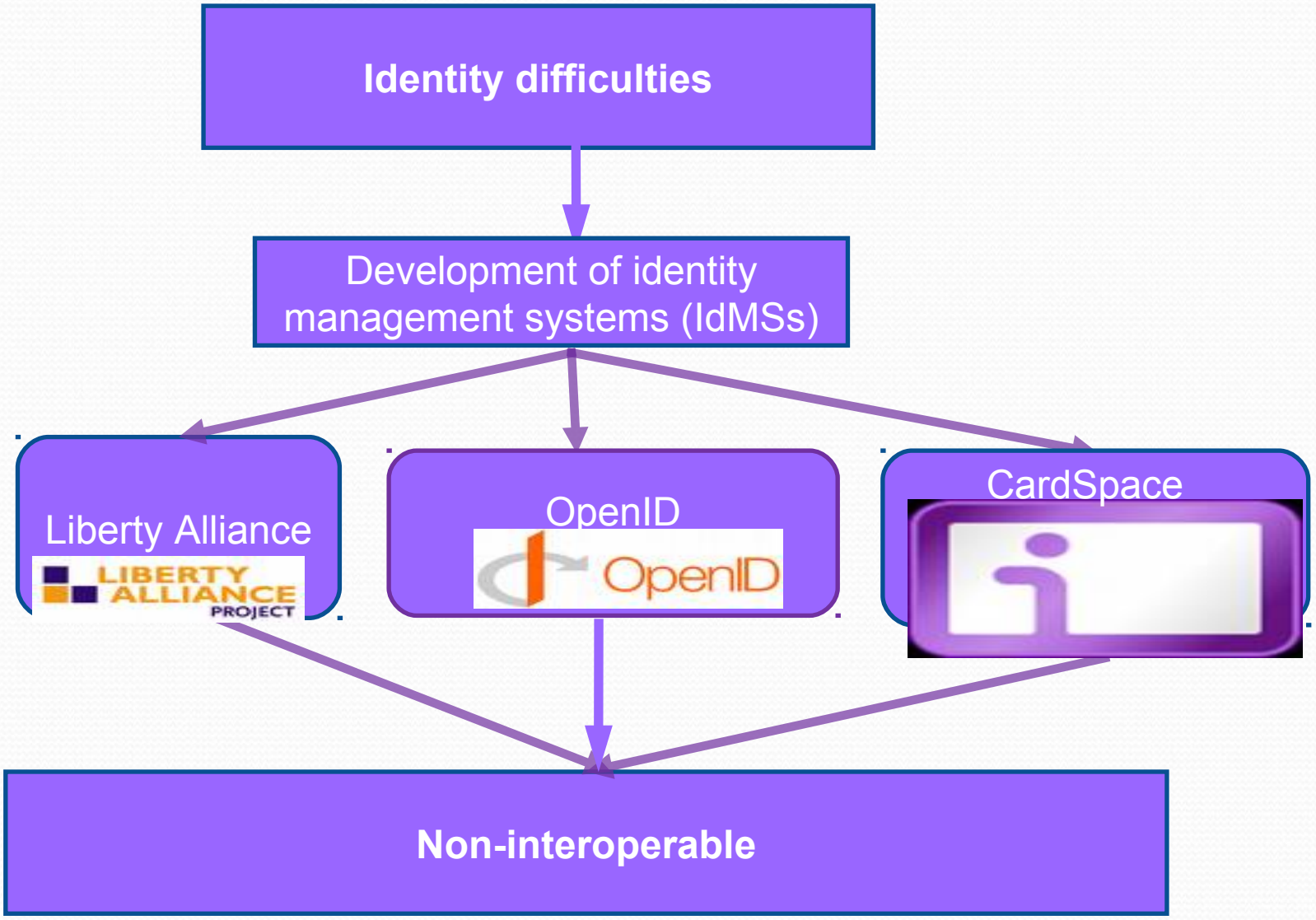
Multiple Identities

- Hard to manage multiple identities (hence poor security practises)



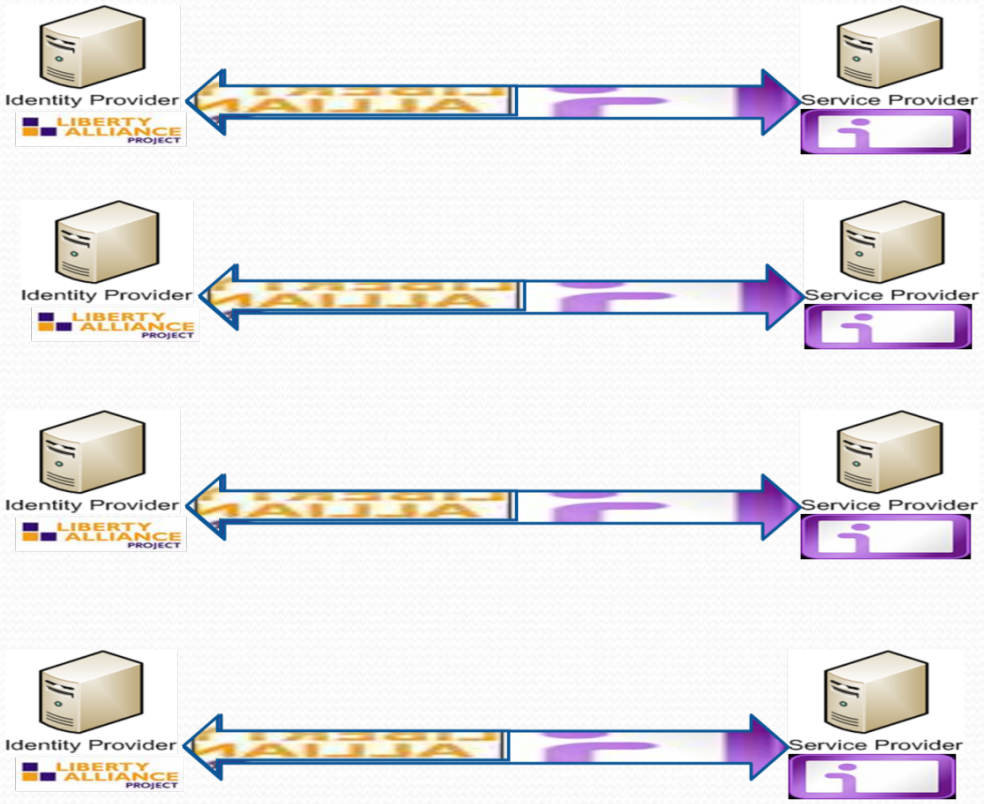
- May result in identity theft

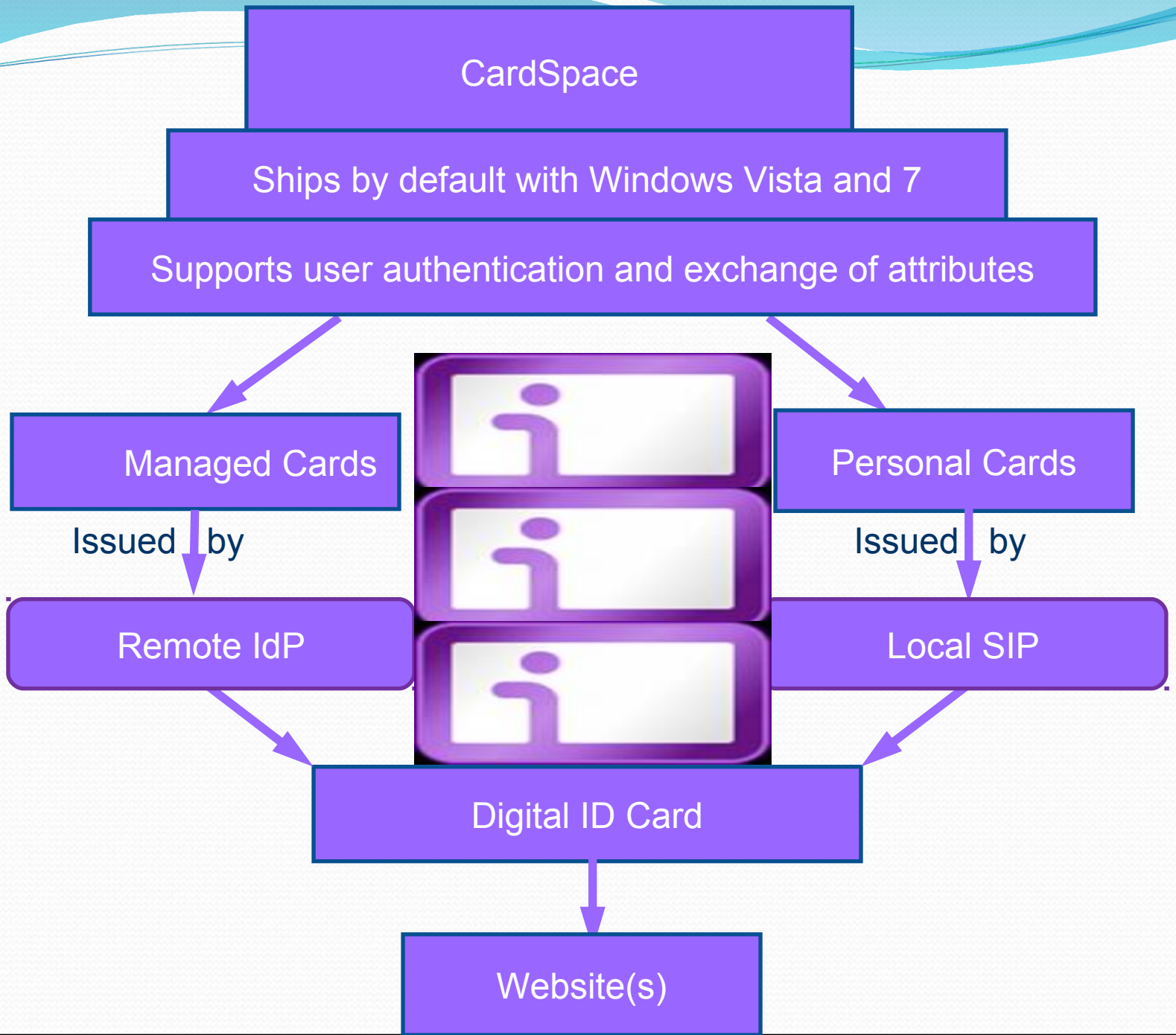



Identity difficulties
Development of identity management systems (IdMSs)
Liberty Alliance
LIBERTY ALLIANCE PROJECT
OpenID
OpenID
CardSpace
Non-interoperable

Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A




CardSpace
Ships by default with Windows Vista and 7
Supports user authentication and exchange of attributes
Managed Cards
Issued by
Remote IdP
Personal Cards
Issued by
Local SIP
Digital ID Card
Website(s)

Acronyms:
RP: Relying Party, e.g. website.
SIP: Self-issued Identity Provider.
CIdS: CardSpace Identity Selector.
RST: Request Security Token
RSTR: Request Security Token Response

CardSpace – SIP Mode

RP

2.

"Can I have a SAML token, containing First Name, E-mail, PPID, issued by SIP, please?"

RP Policy

3. CIdS highlights InfoCards that satisfy the RP policy

1 Request protected resource

7. Token is presented

4. User picks a card

6. Token is created (RSTR)

5. Token is requested (RST)

SIP

Acronyms:
UA: User Agent, e.g. web browser (IE8).
RP: Relying Party, e.g. website.
CIdS: CardSpace Identity Selector.
SIP: Self Issued Identity Provider.

CardSpace – SIP M

1. **UA → RP:** HTTP/S Request, GET (Login Page).
2. **RP → UA:** HTTP/S Response, Login Page + RP Policy.
3. **User → UA:** CardSpace option clicked, and CIdS invoked.
4. **UA ↔ CIdS:** RP policy passed, matching InfoCards highlighted, the rest greyed out.
5. **User ↔ CIdS:** Picks/sends an InfoCard.
6. **CIdS ↔ SIP:** Exchange of RST & RSTR.
7. **CIdS → UA → RP:** RSTR.
8. **User ↔ RP:** Grants/denies access.

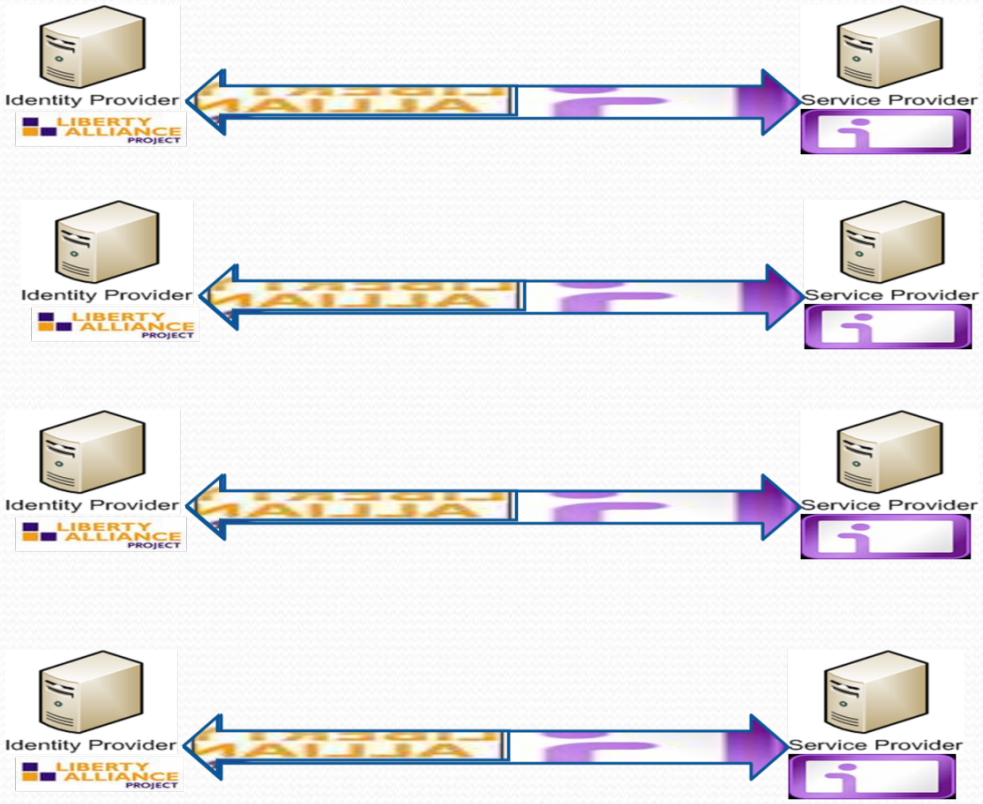
```
<?xml version="1.0" encoding="UTF-8" ?>
<saml:Assertion MajorVersion="1" MinorVersion="1"
AssertionID="SamlSecurityToken-eb6f5355-594d-4001-ae2c-ccc06698db08"
Issuer="http://schemas.xmlsoap.org/ws/2005/05/identity/issuer/self"
IssueInstant="2010-03-04T23:13:28.209Z"
xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
<saml:Conditions NotBefore="2010-03-04T23:13:28.209Z"
NotOnOrAfter="2010-03-05T00:13:28.209Z">
<saml:AudienceRestrictionCondition>
 <saml:Audience>https://www.myopenid.com/signin_password</saml:Audience>
 </saml:AudienceRestrictionCondition>
 </saml:Conditions>
<saml:AttributeStatement>
<saml:Subject>
<saml:SubjectConfirmation>
 <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:ConfirmationMethod>
 </saml:SubjectConfirmation>
 </saml:Subject>
<s
At
<
<
<
<S
<S
...
</Signature>
</saml:Assertion>
```

myOpenID™

WELCOME TO MYOPENID

Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A



Liberty Alliance Project

Consortium of (150+) companies interested in SSO & IdM

As of 2006, more than one billion Liberty-enabled identities & devices

Builds open standard-based specifications for an 'open' XML-based SSO system

Identity Web Services Framework (ID-WSF)

Provides the framework for building interoperable identity-based web services.

Authentication, Discovery, Invocation, Interaction

Identity Federation Framework (ID-FF)

Enables Identity federation and management through features such as identity/account linkage, simplified sign on, and simple session management

Service Interface Specifications (ID-SIS)

- Personal Profile
- Employee Profile
- Business Profile
- Calendar
- Presence
- Alert
- Wallet
- Payment
- Notification
- Availability
- Geolocation
- Gaming
- SMS/MMS Messaging

LIBERTY ALLIANCE PROJECT

LIBERTY ALLIANCE PROJECT

Liberty Profiles

‘The combination of message content specification and message transport mechanisms for a single client type is termed a Liberty profile [1]’

[1] S. Cantor, J. Kemp, and D. Champagne (editors). Liberty ID-FF Bindings and Profiles Specification. Liberty Alliance Project, 2004.

Liberty Artifact

Supported (in the integration scheme)	✗
Prototyped	✗

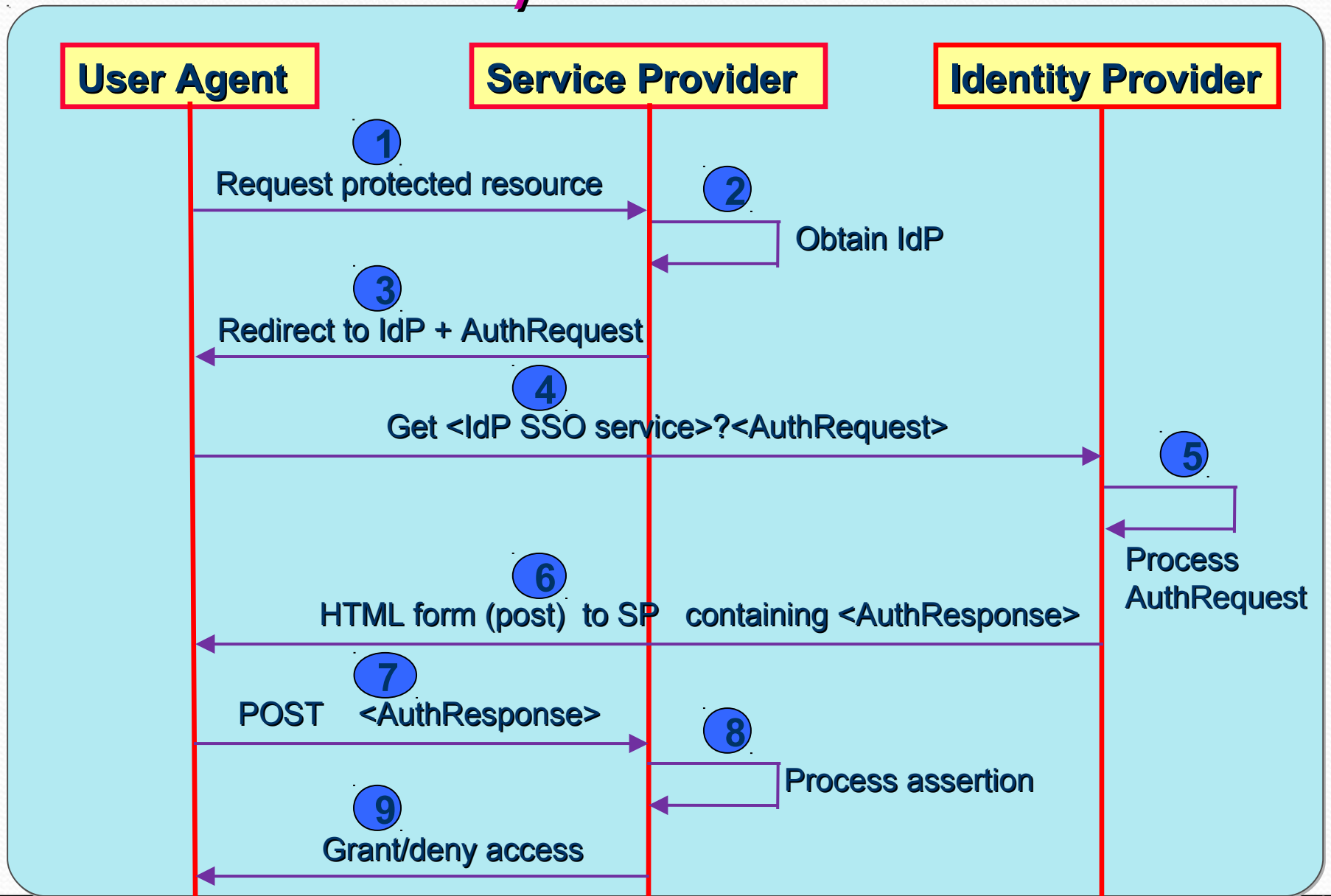
Liberty-Enabled Client (LEC)

Supported	✓
Prototyped	✗

Liberty Browser Post

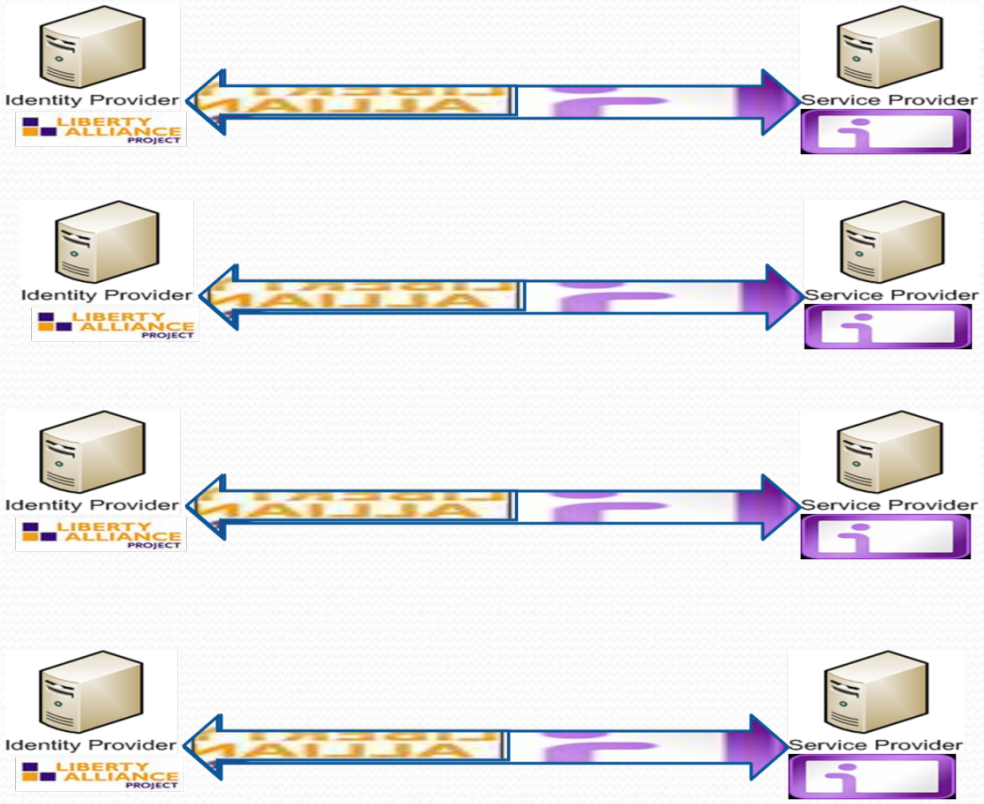
Supported	✓
Prototyped	✓

Liberty Browser Post

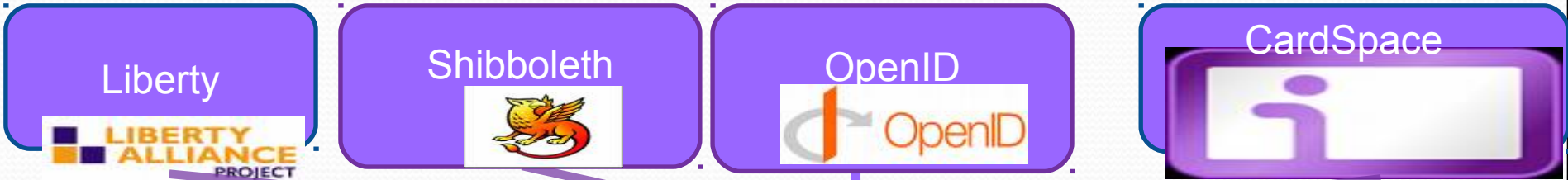


Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A



Interoperability --- Motivation



Identity systems are proliferating ...

Each system offers somewhat distinct user experience

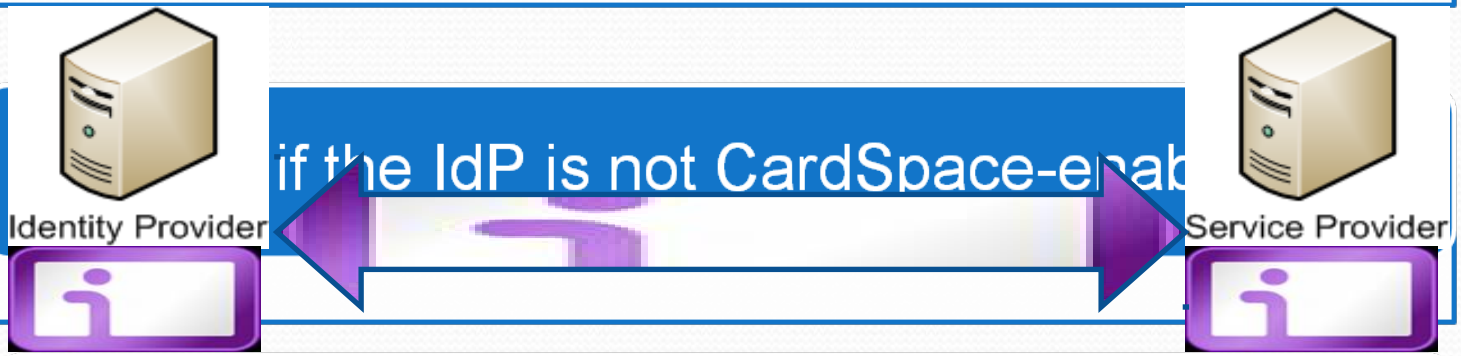
Different experiences may lead to user confusion, which in turn, could lead to phishing, pharming, etc

Interoperation could lead to consistent user experience

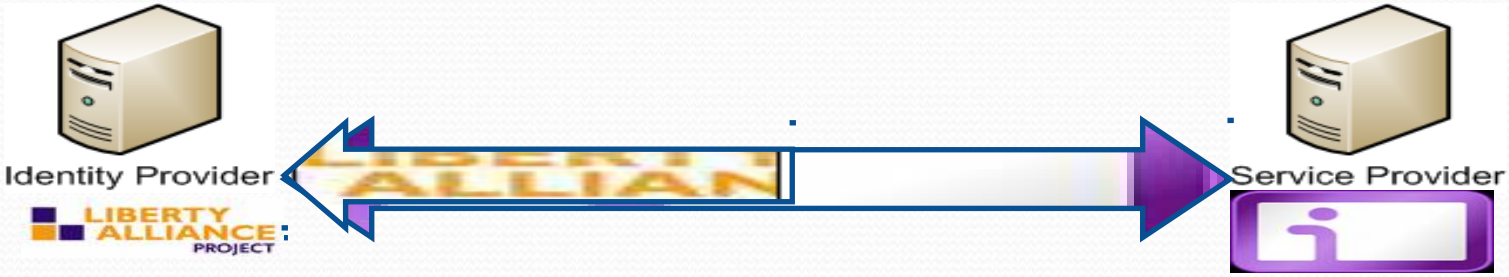
Hence, better security ...

CardSpace Liberty

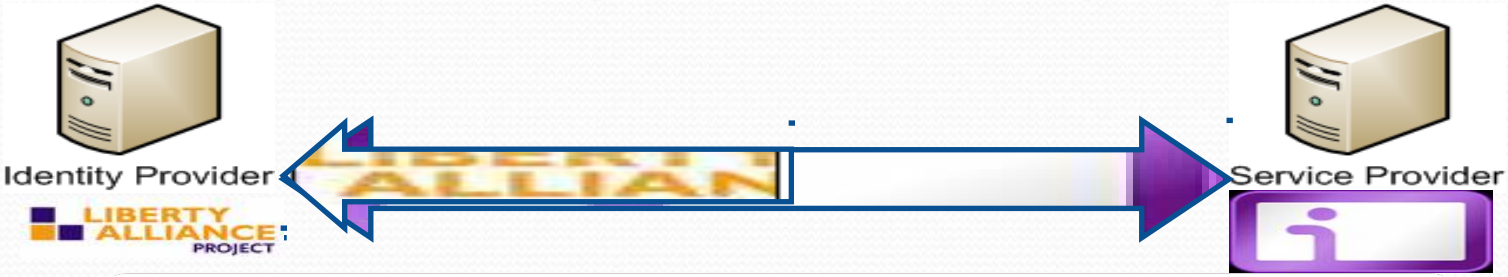
For CardSpace to work, both the RP and IdP must be CardSpace-enabled.



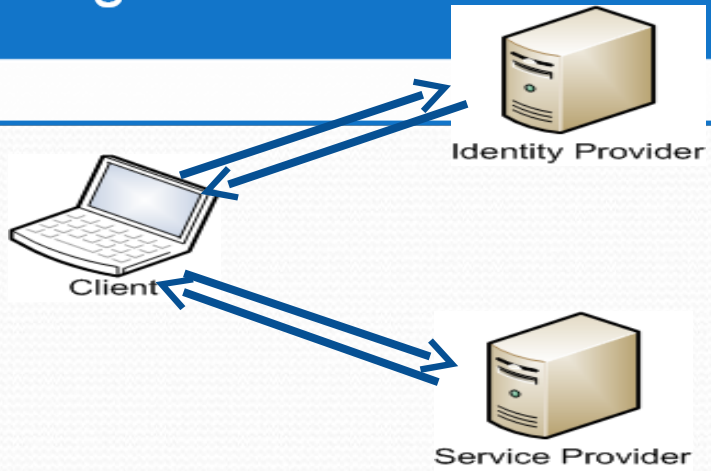
What if the IdP is Liberty-enabled?



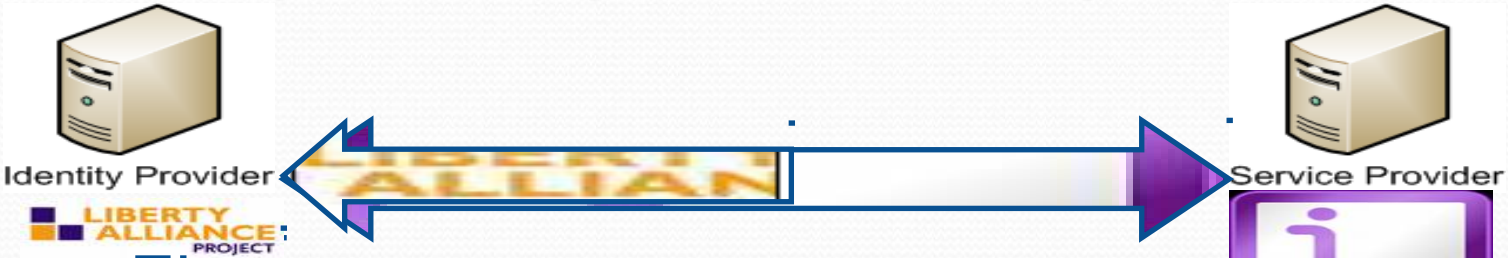
Why CardSpace-Liberty? (1/3)



- Both use SAML
- Both adhere to the general IdP-SP model



Why CardSpace-Liberty? (2/3)



Wide adoption

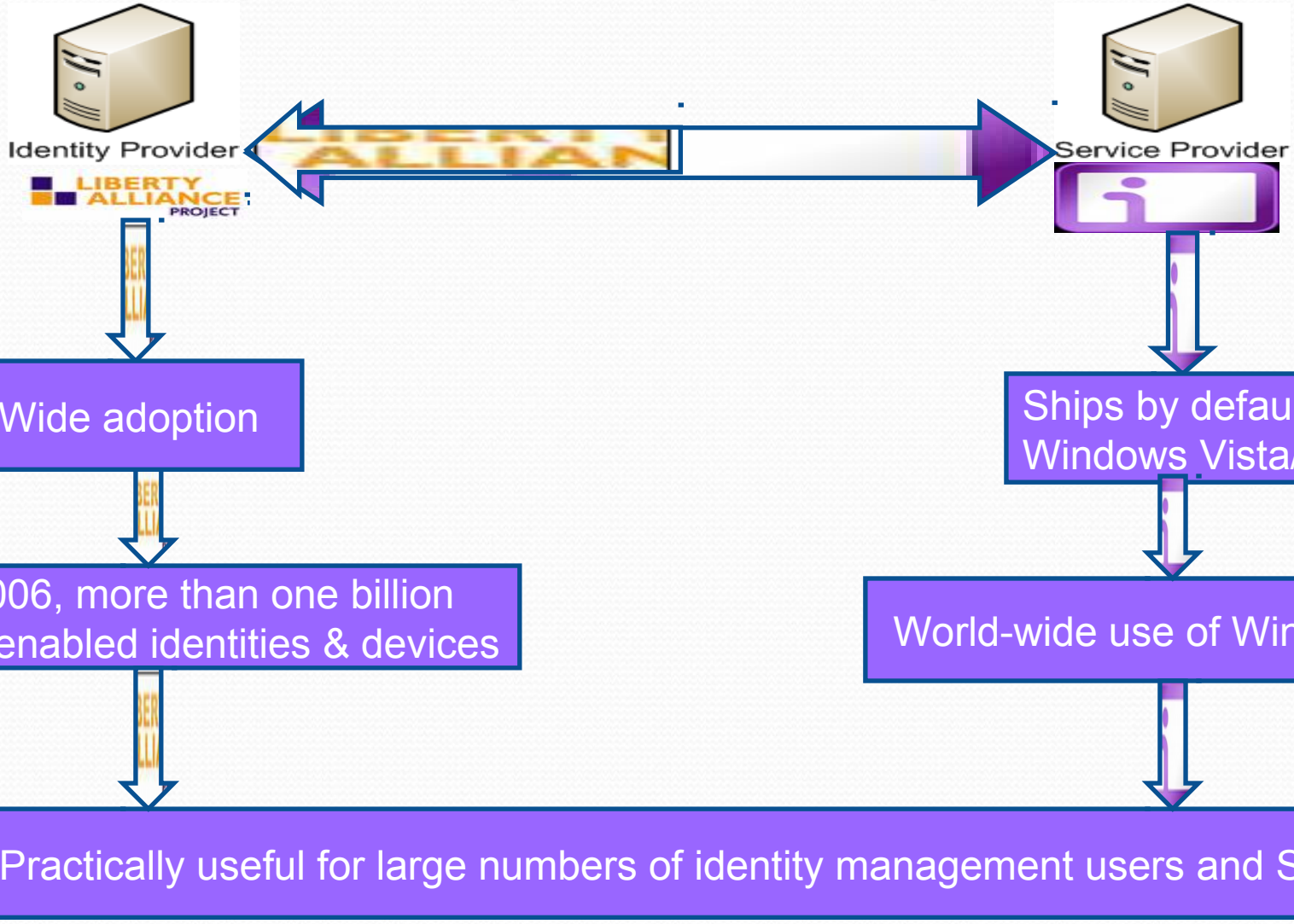
Slow adoption

As of 2006, more than one billion Liberty-enabled identities & devices

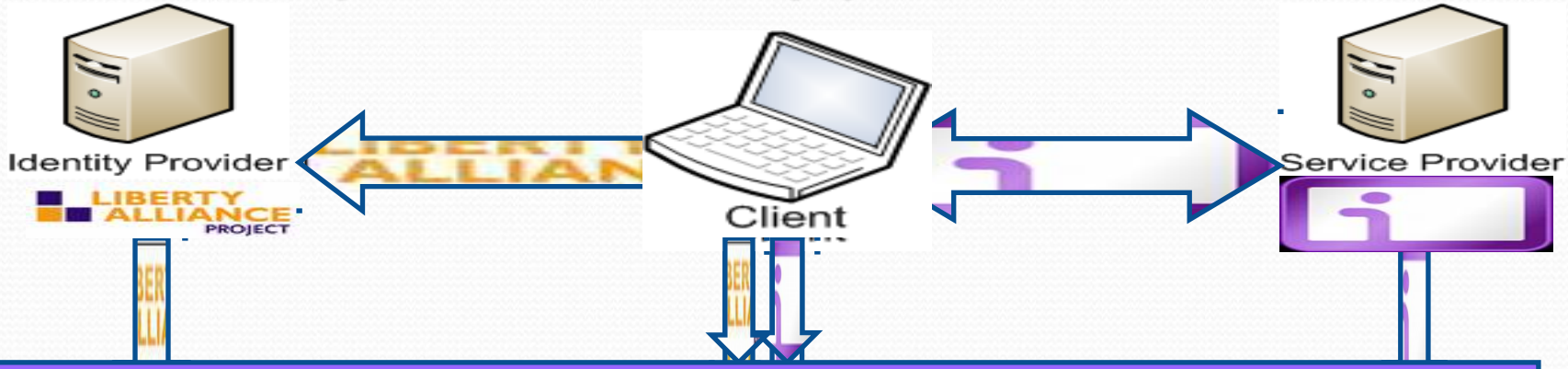
Interoperation could extend the applicability of CardSpace

Hence, adoption is likely to increase

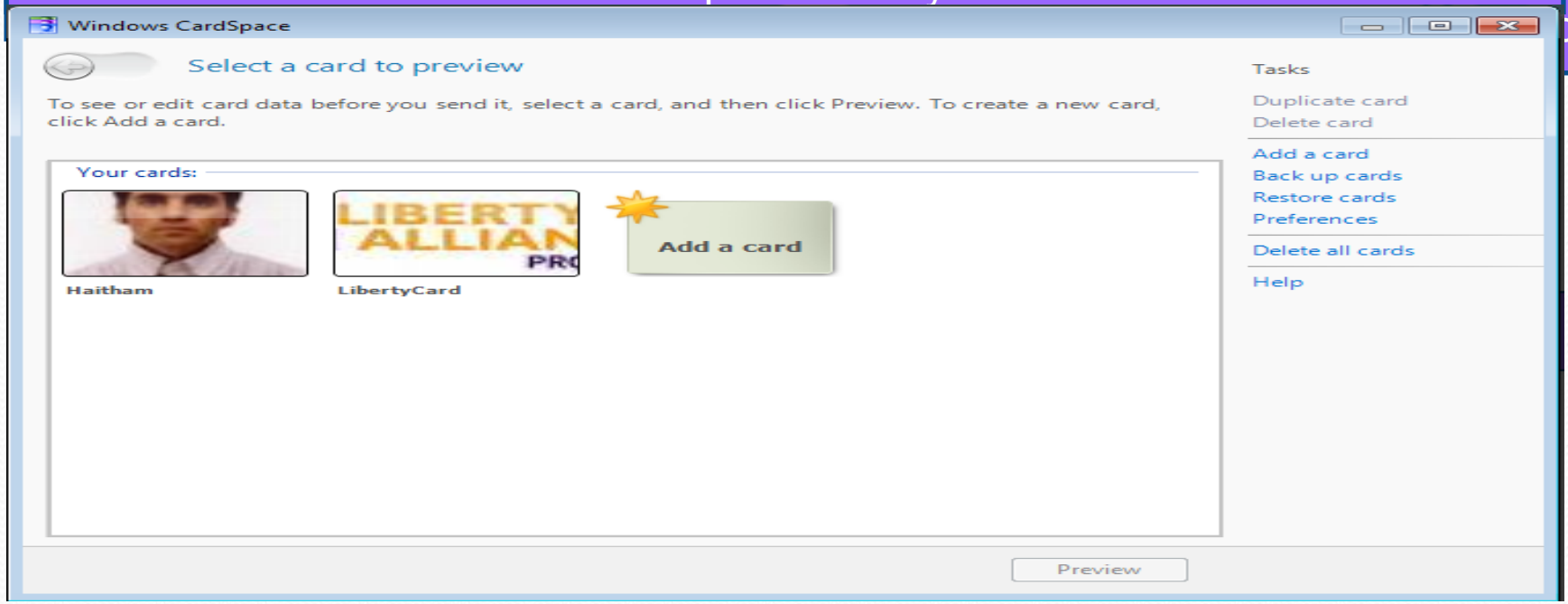
Why CardSpace-Liberty? (3/3)



Interoperation Support --- Where?



CardSpace personal cards are used to make Liberty IdPs available via the CardSpace identity selector



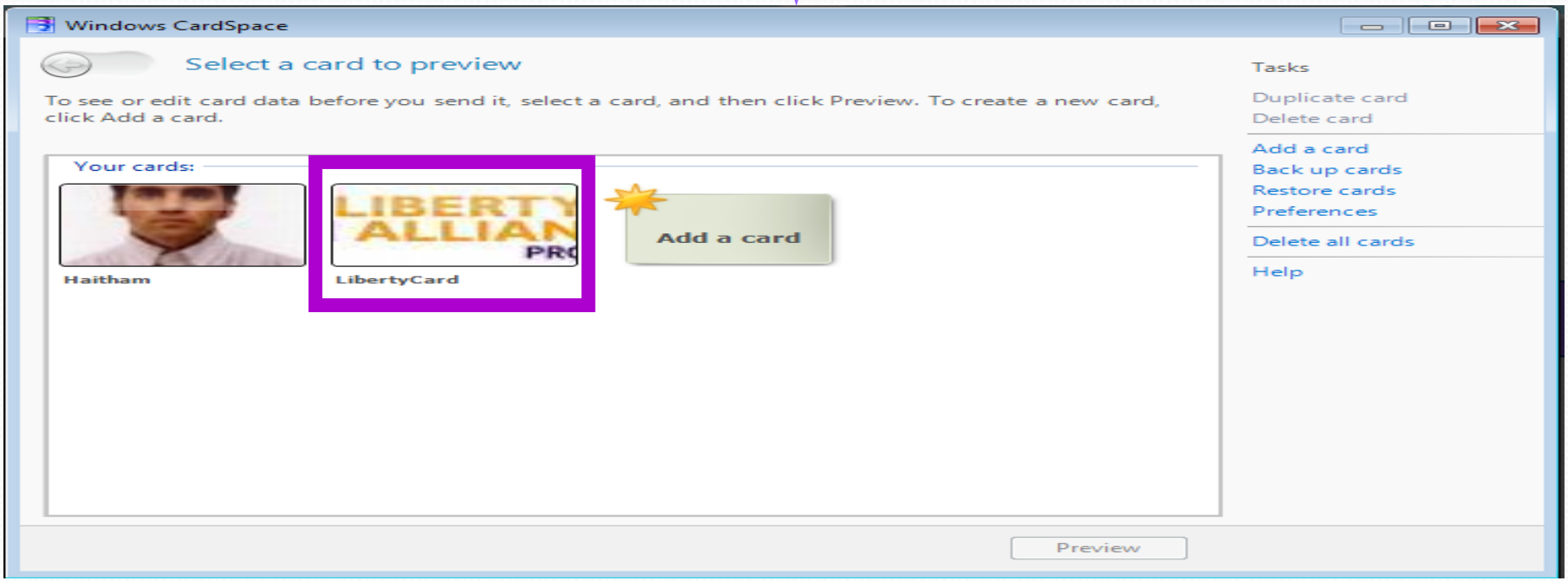
Integration scheme - Preconditions

- The user must have accounts with a CardSpace-enabled RP and a Liberty-enabled IdP.
- The CardSpace RP must be able to verify the IdP-enabled signature on the provided SAML token.
- The CardSpace RP must not use an STS.
- The Liberty IdP must use the InfoCard PPID for the user in place of Liberty pseudonyms in SAML tokens.

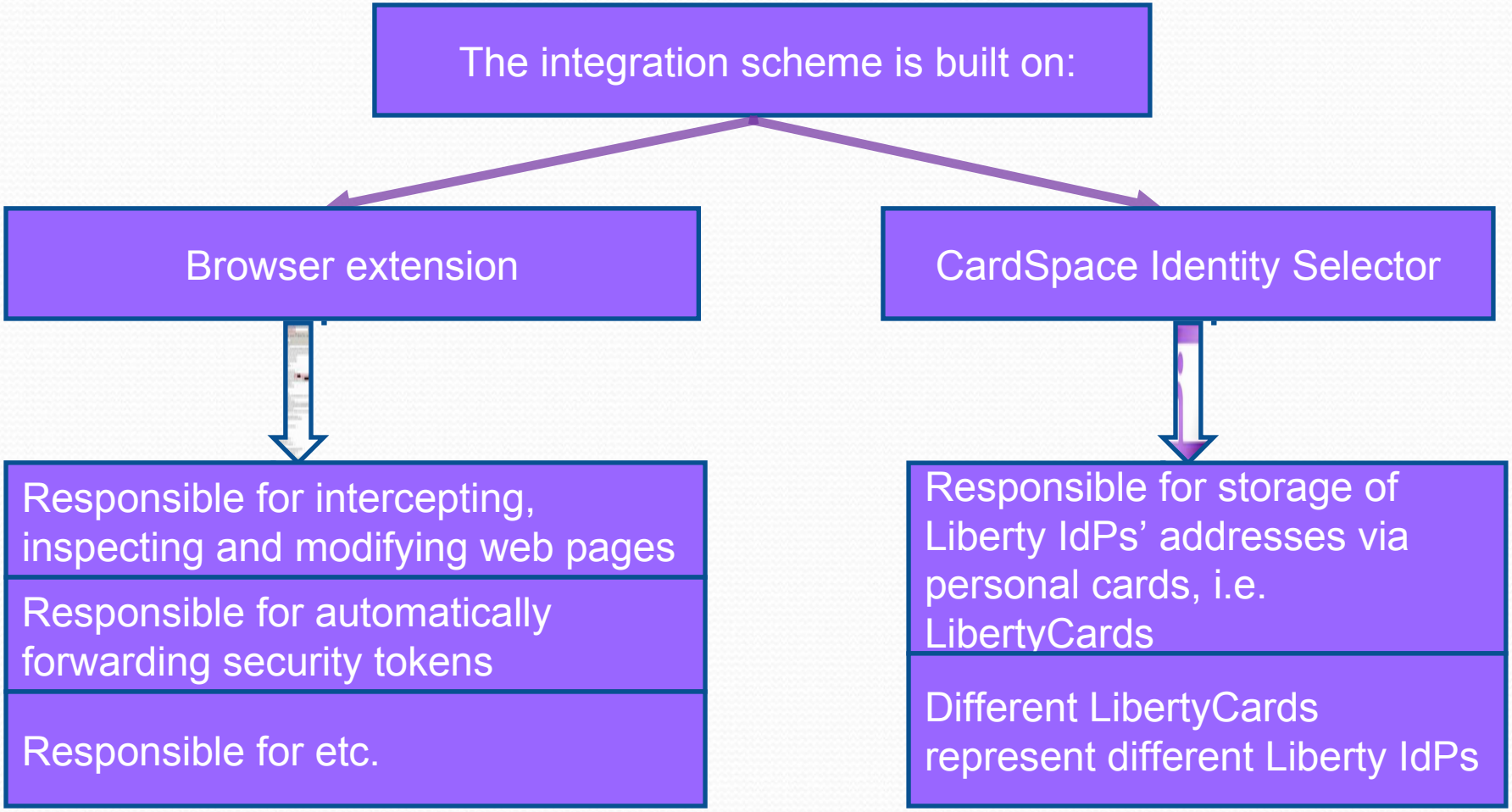
Integration scheme - LibertyCards

The user must create a LibertyCard, which contains (at least):

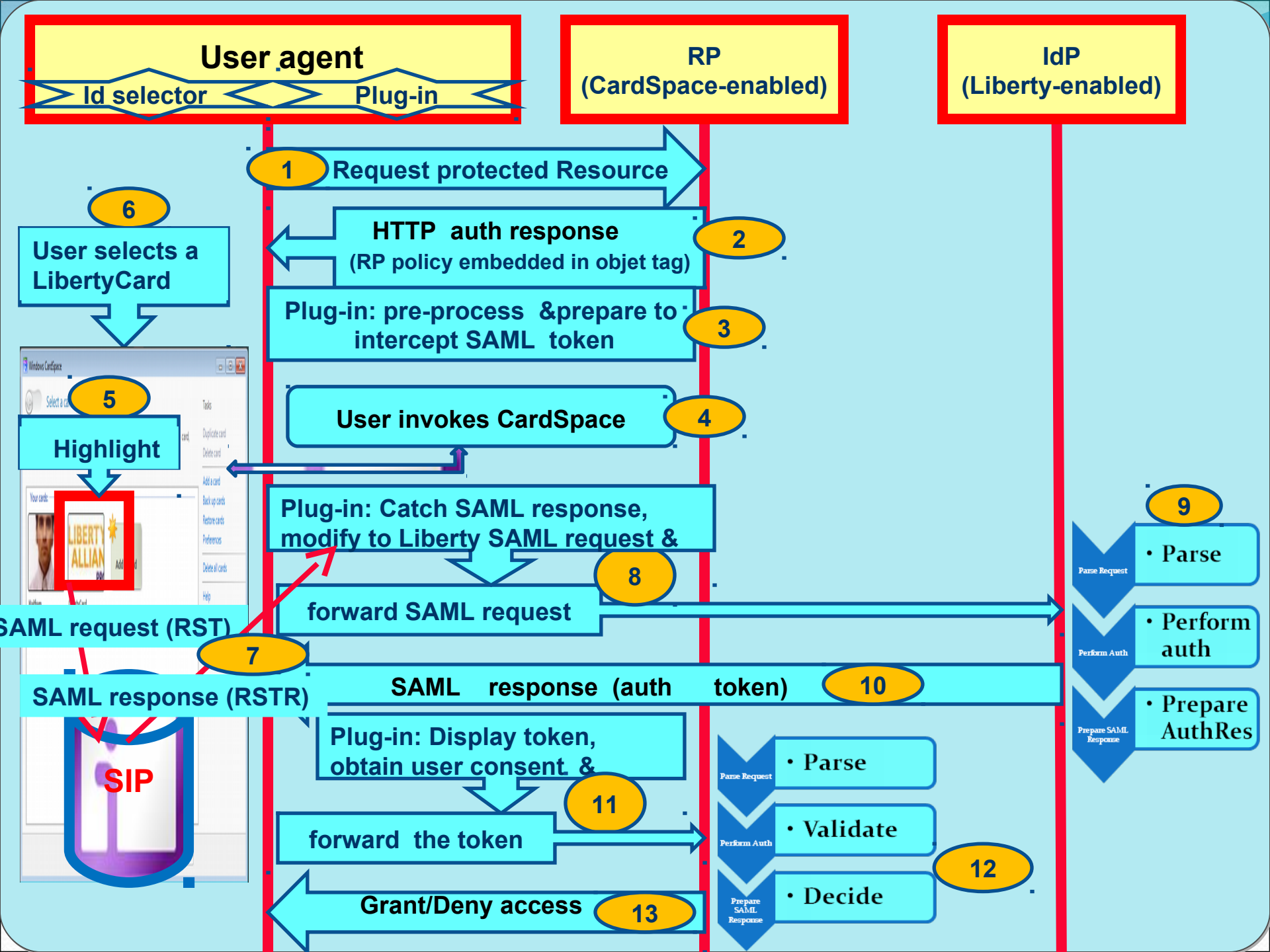
- Address of the Liberty IdP
- Trigger sequence, e.g “Liberty”



Integration Scheme - How?

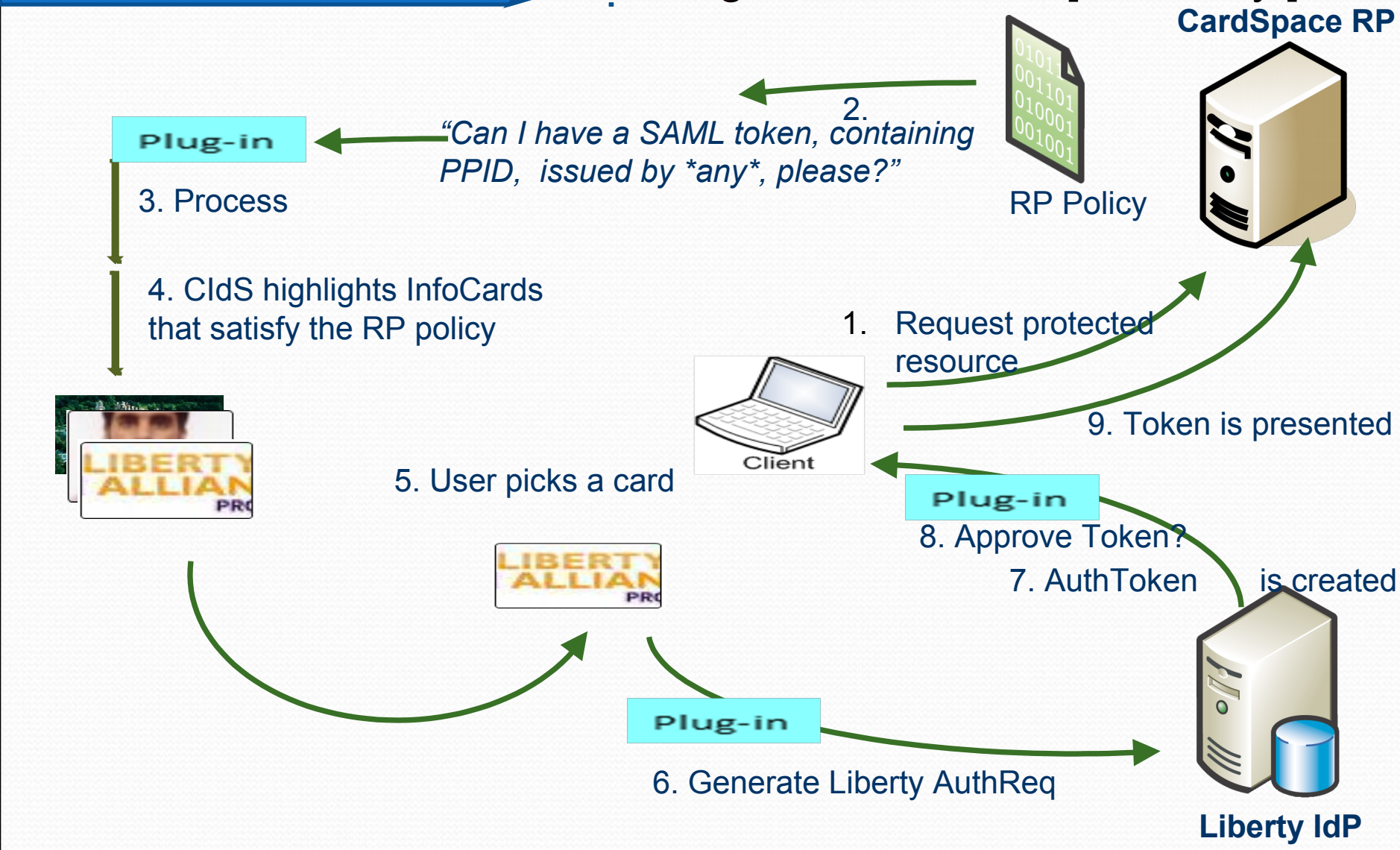


Integration Protocol [Detailed View]


User agent
Id selector
Plug-in
RP
(CardSpace-enabled)
IdP
(Liberty-enabled)
1
Request protected Resource
HTTP auth response
(RP policy embedded in objet tag)
2
Plug-in: pre-process &prepare to intercept SAML token
3
User invokes CardSpace
4
6
User selects a LibertyCard
5
Highlight
Plug-in: Catch SAML response, modify to Liberty SAML request &
8
forward SAML request
SAML request (RST)
7
SAML response (RSTR)
SAML response (auth token)
10
9
Parse Request
• Parse
Perform Auth
• Perform auth
Prepare SAML Response
• Prepare AuthRes
Plug-in: Display token, obtain user consent &
11
forward the token
Parse Request
• Parse
Perform Auth
• Validate
Prepare SAML Response
• Decide
12
Grant/Deny access
13
SIP

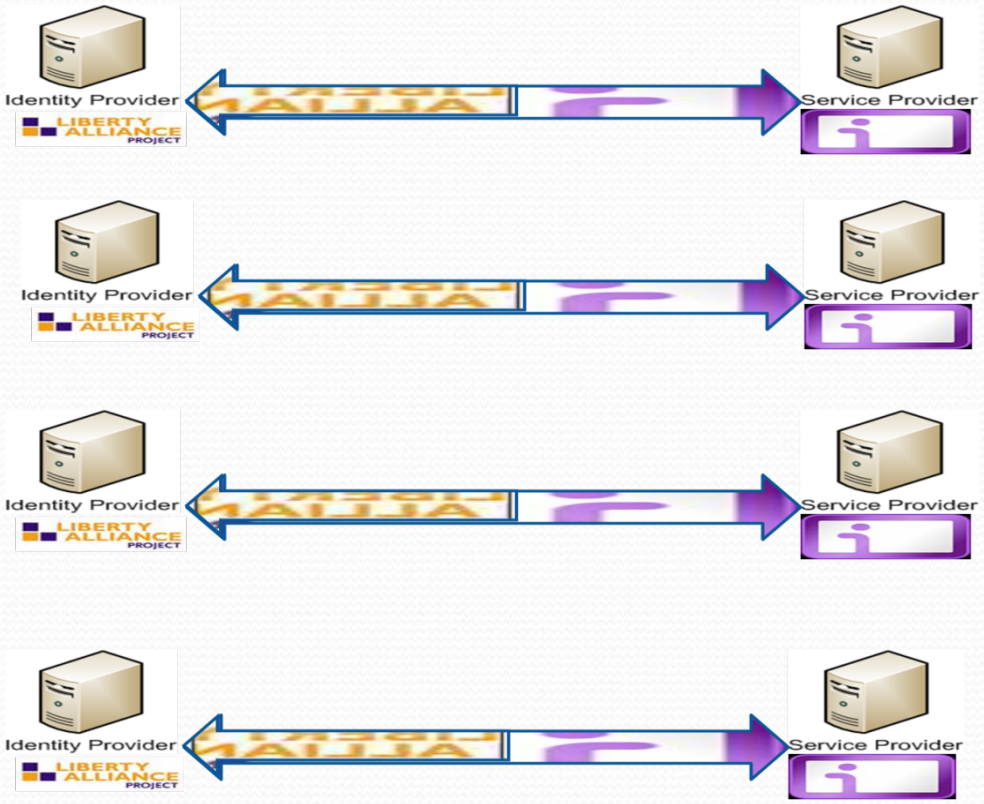
Acronyms:
RP: Relying Party, e.g. website.
IdP: Identity Provider , e.g. Website.
CIdS: CardSpace Identity Selector.

Integration Scheme [summary]



Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A



Integration Scheme - Analyses (1/2)

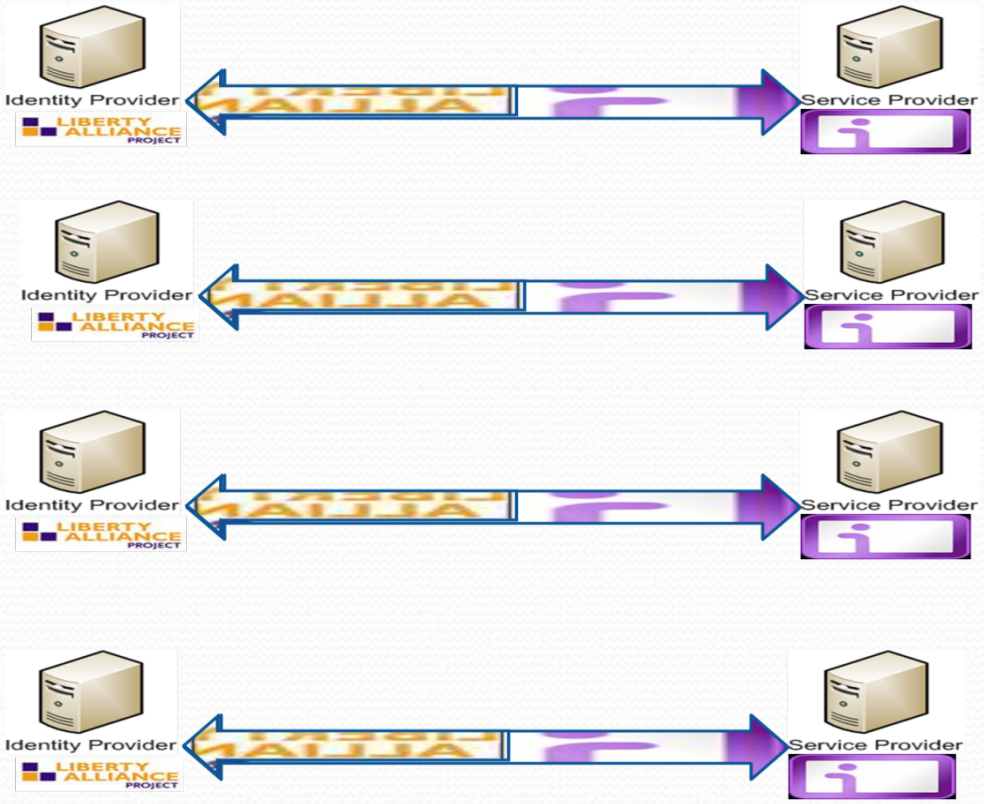
- Defeats Phishing; the RP is unable to redirect the user to an IdP of its choosing.
- No requirements for Microsoft/Liberty cooperation.
- No changes to default IE security settings.
- No changes to CardSpace identity selector.

Integration Scheme - Analyses (2/2)

- CardSpace and Liberty ID-FF have somewhat different scopes.
- Despite being presented as Liberty-specific, the scheme could be extended to support SAML-compliant IdPs.
- The current version of prototype does not support https-based RPs.

Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A

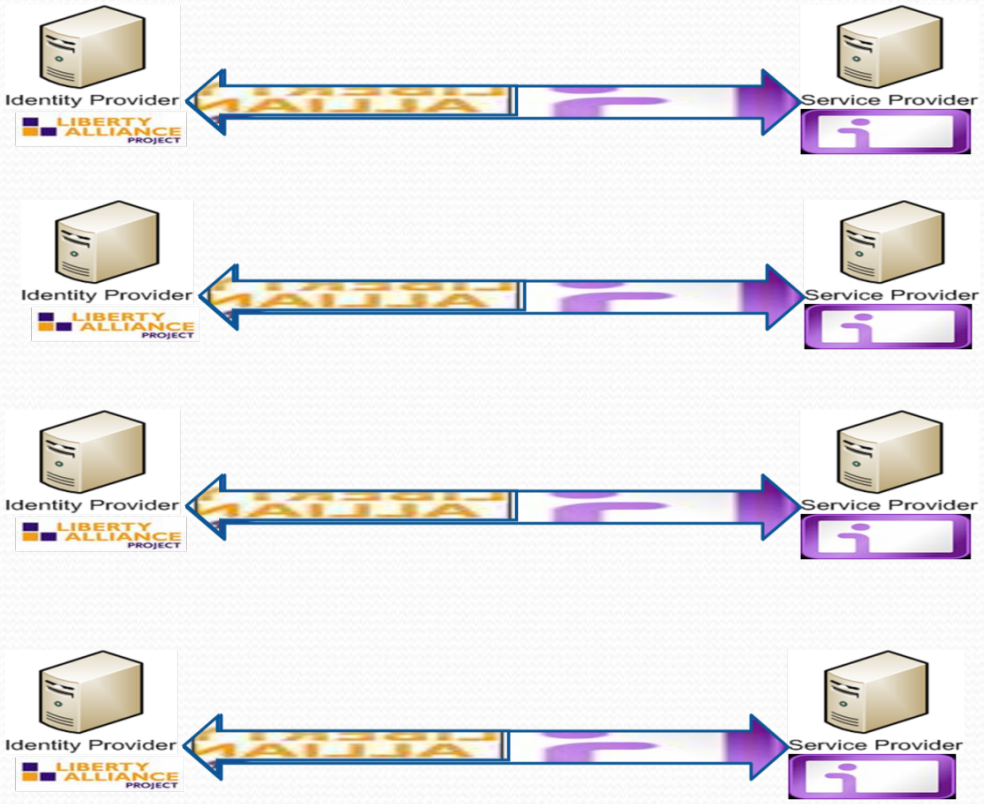


Concluding Remarks

- Interoperation between IdMSs could lead to usability and security improvements.
- Presented a CardSpace-Liberty integration scheme, based on a browser extension and the CardSpace identity selector.
- To be presented: a CardSpace-OpenID integration scheme, also based on a browser extension and the CardSpace identity selector.

Agenda

1. Introduction
2. CardSpace
3. Liberty
4. Integration Scheme
5. Analyses
6. Concluding remarks
7. Q/A



Thank you!

Any Questions?

Information Security Group

CardSpace-Liberty Integration for CardSpace Users

IDtrust 2010

13/4/2010

Haitham Al-Sinani
Information Security Group
Royal Holloway, University of London

H.Al-Sinani@rhul.ac.uk
http://isg.rhul.ac.uk/

An Attribute-based Authorization Policy Framework with Dynamic Conflict Resolution

Apurva Mohan
School of Electrical and Computer Engineering
Georgia institute of Technology
Atlanta, GA, USA
apurva@gatech.edu

Douglas M. Blough
School of Electrical and Computer Engineering
Georgia institute of Technology
Atlanta, GA, USA
doug.blough@ece.gatech.edu

ABSTRACT

Policy-based authorization systems are becoming more common as information systems become larger and more complex. In these systems, to authorize a requester to access a particular resource, the authorization system must verify that the policy authorizes the access. The overall authorization policy may consist of a number of policy groups, where each group consists of policies defined by different entities. Each policy contains a number of authorization rules. The access request is evaluated against these policies, which may produce conflicting authorization decisions. To resolve these conflicts and to reach a unique decision for the access request at the rule and policy level, rule and policy combination algorithms are used. In the current systems, these rule and policy combination algorithms are defined on a static basis during policy composition, which is not desirable in dynamic systems with fast changing environments.

In this paper, we motivate the need for changing the rule and policy combination algorithms dynamically based on contextual information. We propose a framework that supports this functionality and also eliminates the need to recompose policies if the owner decides to change the combination algorithm. It provides a novel method to dynamically add and remove specialized policies, while retaining the clarity and modularity in the policies. The proposed framework also provides a mechanism to reduce the set of potential target matches, thereby increasing the efficiency of the evaluation mechanism. We developed a prototype system to demonstrate the usefulness of this framework by extending some basic capabilities of the XACML policy language. We implemented these enhancements by adding two specialized modules and several new combination algorithms to the Sun XACML engine.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection; D.4.6 [**Operating Systems**]: Security and protection

IDtrust '10 April 13-15, 2010, Gaithersburg, MD

General Terms

Security, Languages, Performance

Keywords

Attribute-based authorization, authorization policy, conflict resolution

1. INTRODUCTION

As information systems become more complex and distributed in nature, system administrators and users need authorization systems which can help them share their resources, data and applications with a large number of users without compromising security and privacy. Although traditional authorization systems address the basic problem of granting access to only authorized individuals, they do not provide a number of desired features of modern authorization systems. These include 1) easily changing authorization based on accessor roles, group memberships, institutional affiliations, location etc., 2) multiple authorities jointly making authorization decision, 3) dynamically changing authorization based on accessor attributes, and 4) GUI-based general purpose tools for description and management of authorization rules. Some traditional authorization systems provide some of these functions on an ad-hoc basis. Although policies have always been part of authorization systems, they were mostly buried in other functional code and hence were difficult to compose and analyze.

Modern policy-based authorization systems provide most of these features. They have a separate policy module that can be queried to make authorization decisions. This module makes decisions taking into consideration all applicable policies for a particular access request. These policies may be defined by multiple authorities. The policies may have different or even conflicting authorization decisions for the same access request. Policy languages use policy combination algorithms (PCA) to resolve such conflicts. These algorithms take the authorization decision from each policy as input and apply some standard logic to come up with a final decision[1].

These PCAs are currently chosen at the time of policy composition and hence they are static. In highly dynamic environments, this is not desirable and there may be a need to select these PCAs dynamically. In this case, it will be useful to have a mechanism to select a suitable PCA based

[1]For efficiency reasons, policy engines only evaluate policies until they reach a final decision based on the combination algorithm.

on the dynamic contextual information available to the system. More discussion on this issue along with a motivating scenario is presented in Section 3.

PCAs used in current systems are also very restricted. There are a number of conflict resolution logics in general purpose computing which are not expressible as PCAs in authorization languages. Examples of these logics include hierarchy-based resolution, priority-based resolution, taking a simple majority vote, and taking a weighted majority vote. There is a need to include algorithms such as these as PCAs in authorization languages to provide more functionality and flexibility in defining policies.

Having a context-aware authorization system also provides the capability to define different policies for different contexts. These contexts can be distinguished by contextual data or environmental attributes. In this case, the policies will be modular making them easy to comprehend and analyze. Without the ability to choose the applicable policies based on contextual information, the policy composer is forced to duplicate each access control rule with and without the contextual information in the same policy. Although the same access control decision can be achieved in both approaches, the latter makes it difficult to analyze the policies and the effect of making changes to them. Also if policies are chosen dynamically, only a small set of rules will be evaluated for their applicability for this request. This reduces the number of matches with potential policy targets thereby lowering computation time.

Another advantage of using context-aware authorization is that a specialized policy created for some specific purpose can be added and removed from consideration dynamically without changing the existing policies. This is especially useful for systems that have to adhere to certain temporary authorization requirements which require special authorization rules. This is also useful in cases where the specialized policy is composed by some entity other than the one who usually creates and maintains authorization policies.

The main contributions of this paper are: 1) proposing a framework where authorization for a particular access request is decided dynamically based on context information, 2) supporting dynamic conflict resolution where PCAs are chosen at run time based on context information, 3) providing the ability to dynamically include (remove) specialized, short-term or add-on policies to (from) the authorization policy set, 4) increasing the efficiency of policy target matching during authorization, 5) increasing the modularity and clarity of the policies, 6) building a prototype authorization system to demonstrate the concepts, and 7) evaluating efficiency of the policy evaluation for the proposed framework.

2. ATTRIBUTE-BASED AUTHORIZATION SYSTEMS

In this section, we first introduce the basic constructs of attribute-based policy languages. We then describe some basic concepts of attribute-based authorization systems, define attribute-based policies, and policy combination algorithms used in conflict resolution.

2.1 Brief Introduction to Policy Languages

In this sub-section, we introduce the basic elements of attribute-based authorization policy languages. Although here we use eXtensible Access Control Markup Language (XACML) as an example to introduce the primary elements, these elements are similar in other policy languages as well.

XACML is an OASIS standard that describes a policy language for representing authorization policies and an access control decision request/response language [2]. XACML is based on XML. It describes general access control requirements while allowing for extensions for defining new functions, data types and combination logics. The language has syntax for defining authorization policies and building a request/response to validate authorization requests against the policies. The response contains one of the four possible outcomes of policy evaluation - Permit, Deny, Indeterminate (an error occurred or some required value was missing, so a decision cannot be made) or Not Applicable (the request can't be answered by this service).

XACML has a Policy Enforcement Point (PEP) that actually protects the resource and a Policy Decision Point (PDP) that evaluates the access request against the policies. The PEP receives the access request from the requesting user and forwards it to the PDP which makes the decision in consultation with the policies. If the access is allowed, the PEP release the resource to the requesting user. The main components of a XACML policy are described below:

Policy - An XACML policy contains a set of rules with the subject and environment attributes, resources and corresponding actions. If multiple rules are applicable to a particular request, then the rule combination algorithm (RCA) combines the rules and resoles any conflict in their decisions. XACML supports the following RCA's - Deny-overrides (Ordered and Unordered),Permit-overrides (Ordered and Unordered), and First-applicable.

Policy Set - A policy set is a container which contains other policies or policy set. One or more of these policies or policy sets may be applicable to a particular access request. If more than one are applicable, then the Policy Combination Algorithms (PCA) are used to combine the policies and resolve any conflicts in their decisions. XACML supports the following PCA's - Deny-overrides (Ordered and Unordered),Permit-overrides (Ordered and Unordered), First-applicable, and Only-one-applicable.

Target - A Target is basically a set of conditions for the Subject, Resource and Action that must be met for a Policy Set, Policy or Rule to apply to a given request.

Rule - The rule is the core representation of the access control logic with the subject, resource, action and environment fields. It is a boolean function, which evaluates to true if the subject, resource, action and environment fields in the request matches with the fields in the rule.

2.2 Authorization Policy

In an attribute-based system, objects are protected by administrator (or object owner) defined policies. These policies define a set of verifiable attributes (with pre-defined values) against each resource for a set of privileges. These attributes are either the characteristics of the user or the environment. These attributes must be presented to the authorization module and verified by it in order to authorize the accessing user to access the requested object with specific privileges. Since the attributes have to be verifiable, they have to be certified by some entity which is trusted by the authorization module.

An attribute-based authorization policy is formally defined below.

Definition 1 : Let $\mathcal{SA}$, $\mathcal{RA}$ and $\mathcal{EA}$ represent the Subject, Resource and Environmental attributes respectively, each of which is well defined set of finite cardinality, given as $\mathcal{SA} = \{sa_1, sa_2.....sa_l\}$, $\mathcal{RA} = \{ra_1, ra_2.....ra_m\}$ and $\mathcal{EA} = \{ea_1, ea_2.....ea_n\}$. These attributes can take values $val\_sa_i \subseteq dom(sa_i)(1 < i < l)$, $val\_ra_j \subseteq dom(ra_j)(1 < j < m)$ and $val\_ea_k \subseteq dom(ea_k)(1 < k < n)$.

Attributes can be of two types, one which can take distinct and unconnected values (for e.g. 'role'='doctor' or 'role'='nurse') and another type which can take a single or range of values (for e.g. 'time' is between t_1 and t_2 or 'age' ≤ 21). In the latter case, the values that an attribute can take are connected. Without loss of generality, we define the latter group as attributes which can take either a single value or a range of values. For example, for a range of sa_j, the domain and values are defined as follows:
Attribute Type 1 -
$dom(sa_j) = [sa_j\_val_1, sa_j\_val_2...sa_j\_val_n]$, $val\_sa_j \in dom(sa_j)$;
Attribute Type 2 -
$dom(sa_j) = [low, high]$, $val\_sa_j = [low', high'] \subseteq dom(sa_j)$;
where, $(low' \geq low)$ and $(high' \leq high)$. If val_sa_j takes a distinct value in $[low, high]$, then $low' = high'$.

Definition 2 : Let Action define a set of actions which a subject can execute on resources. $\mathcal{ACT} = \{act_1, act_2.....act_p\}$. For example, the set of actions on a file can be {read, write, delete, append, execute}. Let $\mathcal{D}$ be the set of decisions that can result as a response to a predicate evaluating to true. $\mathcal{D} = \{d_1, d_2.....d_q\}$.

Definition 3 : An access request ($\mathcal{AR}$) is a tuple of the form $< s, r, a >$, where $s \subseteq \{\mathcal{SA}, \mathcal{EA}\}$, $r \subseteq \{\mathcal{RA}\}$ and $a \subseteq \{\mathcal{ACT}\}$. It represents that s is requesting to access r with rights a. A Rule $\mathcal{R}$ has the same format but defines the set s required to access r with rights a.

Definition 4 : A policy is a list of rules given as $\mathcal{P} = (\oplus, < \mathcal{R}_1, \mathcal{R}_2.....\mathcal{R}_s >)$. $\oplus$ is a combination function, which combines the rules to produce a single decision for the policy.

Definition 5 : A Policy Set (PS) is a container which contains a list of policies. It may also contain other policy sets. It is given as $\mathcal{PS} = (\oslash, < \mathcal{PS}_1, \mathcal{PS}_2, ...\mathcal{PS}_i >)$. Each $\mathcal{PS}_t$ represents either a policy set or a single policy[2]. $\oslash$ is a combination function, which combines all the policy sets. This combination function is used to combine policies and policy sets and has no direct relation with the rule combination algorithm.

Conceptually, a policy is a deliberate plan to implement authorization to a particular resource or group of resources. A rule is a component of the policy that defines a specific authorization predicate. A policy set is a container that contains a number of logically connected policies. In a multi-authority setting where the authorization policies for a particular resource are defined by a number of entities, all policies for that particular resource will form a logical policy set. For example, at a university, the firewall policies to protect a lab computer may be a combination of the policy defined centrally by the office of information technology, a specific department policy, a lab firewall policy, and the administrator defined policy for that computer. A policy set encompasses all of these policies. The policies can be defined in a number of policy description languages. Each has its advantages and disadvantages. In describing the policies in this paper, we will use the syntax and structure of XACML [2], which is an OASIS standard. XACML is an attribute-based policy description language and is used for implementing our prototype system. Although we use XACML for discussion and implementation, the model we present in this paper is generic and can be implemented in other policy languages like P3P [4] or EPAL [1].

[2]In which case the set has a single policy and no PCA.

2.3 Combination Algorithms and Conflict Resolution

In a large system, there may be multiple authorities who specify the authorization policies. As such, there can be multiple groups of policies. When a request is evaluated in the system, the authorization module determines which policy sets apply to the particular request. Then it checks which policies among those groups and which rules among those policies are applicable to the request. There can be multiple policy sets and multiple policies in each set applicable to a single access request. Even within each policy there can be multiple rules which apply to the access request. These rules and policies can have a different or even conflicting decision for the request. As such, a mechanism is needed to resolve these conflicts. Policy languages have some rule combination algorithms (RCAs), which evaluate the applicable rules based on the logic of the algorithm and resolve any conflict in their decisions.

Definition 6 : In a single policy, $\mathcal{E}(\mathcal{AR}, \mathcal{R}_i) \rightarrow d_i$, where $\mathcal{E}$ represents the evaluation of the i^{th} rule and d_i is the corresponding decision. The set of all the decisions is given as $\mathcal{D}^{Rule} = (< d_1, d_2, ..., d_x >)$. Rule Combination Algorithm (RCA) is defined as $\{\mathcal{RCA}\ \phi\ \mathcal{D}^{Rule}\} \rightarrow d$, where $d\ \epsilon\ \mathcal{D}$. ϕ represents *'applied to'*.

For example, a policy may use 'deny-overrides' as its RCA. In this case, if the algorithm finds even a single rule that denies the access, its final decision is 'deny'; otherwise its decision is 'permit' even if a single rule permits. If none of the rules either 'permit' or 'deny' the access, then the result is 'Not Applicable'.

For combining the policies and policy groups, policy languages have policy combination algorithms (PCAs). These algorithms work on similar logic as the RCAs. Each policy give a single decision for the access request. The PCA combines these decisions into a single decision by using the PCA logic.

Definition 7 : In the final policy list, $\mathcal{E}(\mathcal{AR}, \mathcal{PS}_i) \rightarrow d_i$, where $\mathcal{E}$ represents the evaluation of the i^{th} policy set and d_i is the corresponding decision. The set of all the decisions is given as $\mathcal{D}^{PS} = \{d_1, d_2, ..., d_x\}$. Policy Combination Algorithm (PCA) is defined as $\{\mathcal{PCA}\ \phi\ \mathcal{D}^{PS}\} \rightarrow d$, where $d\ \epsilon\ \mathcal{D}$.

In the current systems, these RCAs and PCAs are static and are determined at the time of composing the policies.

3. DYNAMIC CONFLICT RESOLUTION

In the last section, we saw how RCAs and PCAs resolve the conflicts among rules and policies to give a unique decision for an access request. We also noted that, in existing systems, these RCAs and PCAs are chosen at the time of composing the policies and hence do not change. This static composition may not be suitable for highly dynamic environments where there is a need to adapt the policies dynamically. If such a mechanism is available, then it can also serve as an easy tool for the policy composer, if he wishes

to change the RCAs and PCAs without recomposing the authorization policies.

Some researchers have proposed static conflict detection and avoidance, arguing that detecting and resolving conflicts in systems with a large number of policies in real time can be a daunting task [26]. We argue that, even though it is a challenging problem, it is a superior approach. Organization policies, regulatory polices, and user policies change regularly. If we perform static conflict analysis, whenever one of the policy changes, new conflicts can arise requiring some party to change their policies. Also, some policies that conflicted before one of the policies changed and were never composed, may now become acceptable. There is no mechanism to reconsider these rejected policies. Also, the static model does not take into account adding and removing specialized and time limited policies to provide flexibility in policy composition and maintenance.

3.1 Motivating Scenario

Let us consider a motivating scenario from the health care domain. Alex is a patient who stores his personal health record (PHR) with his health maintenance organization(HMO) called Superior Health Care (SHC). At SHC the patients' PHRs are stored in a repository where the access to the repository is mediated through a proxy. The proxy stores all the authorization policies. The policies may have multiple groups with policies defined by patients like Alex himself, the hospital which created the record, SHC's organizational policies, federal regulatory policies, and so on. When someone tries to access an EMR for a particular patient, the system will consult the applicable policies to check whether this access is allowed. Assume that, in normal circumstances, the policy combination algorithm used is 'deny-overrides', which is a secure and stringent policy. Suppose that Alex wishes to use a more lenient policy in case of an emergency, where he will share his PHR with any accessor who is authorized by at least one of the applicable policies. In this case, he needs to dynamically change his PCA from 'deny-overrides' to 'permit-overrides' whenever there is an emergency and back to 'deny-overrides' once the emergency is over. The traditional method would require him to change his policies twice to achieve this. If Alex want to have several dynamic options, he will have to change his policy description each time such a dynamic change occurs.

In the proposed model, Alex can define all such dynamic conditions as an attribute-based policy and the evaluation of these policies will determine what PCA will be used for the current access request. The model extends this concept to the selection of the RCA dynamically. It is desirable that the user has the ability to define several dynamic conditions simultaneously, need not change his policy descriptions every time one such condition changes, and also need not keep track of the dynamic changes. This is one of the key advantages of using the proposed system. If Alex tries to achieve the same effect in current policy-based systems with static conflict analysis, when an emergency occurs he will have to recompose his policy with 'permit-overrides' and resolve all conflicts created in the process. When the emergency is over, he will have to recompose his policies with 'deny-overrides' and resolve all conflicts again. He cannot create a special policy for an emergency, because his two policies are inherently contradictory. This puts a heavy burden on the user and also, by definition an emergency comes unexpectedly, therefore Alex cannot be expected to recompose policies when an emergency has already occurred. In current systems, users like Alex do not change their policies on such events. Our novel framework enables users to achieve this with little effort and provides an important new functionality.

3.2 Proposed Model

In this section, we present a novel mechanism to dynamically determine the policies applicable to an access request and to evaluate only the applicable policies. In this model, we evaluate the authorization policies in two stages. In the first stage, we determine which policies are applicable to the current access request and we also dynamically determine which PCA will be used to resolve the conflicts in the authorization decisions. In the second stage, we evaluate only the applicable policies using the PCA selected in the first stage.

During stage one, the total applicable policy set (TAPS) is determined by selecting only those policies where at least one of the authorization rules is applicable to the current access request. If $PS_1, PS_2...PS_n$ are the authorization policy sets, then the TAPS for a particular $\mathcal{AR}$ is given as $TAPS = \oslash\{PS_1, PS_2....PS_n\}$.

The combination algorithm $\oslash$ used is *'all-that-apply'*, which is a new rule combination algorithm defined in Appendix A. The *'all-that-apply'* algorithm has been implemented in our modified XACML engine (see Section 5). To evaluate TAPS, all available policy sets are evaluated as explained in Definition 6. If a policy set has at least one rule that applies to the current access request, we include it in the TAPS. To find an applicable rule, we consider the subject and environment attributes in the access request (which is the set $\{EA \cup SA\}$) along with their boolean relationships. We then match that with the rules in the policy level target. We try to find a rule with the same set $\{EA \cup SA\}$ with the same relationships so that at least one of the attribute combinations matches with those in the $\mathcal{AR}$. EA, SA and RA are specified in Definition 1.

To aid in determining applicable policy sets, we create a meta-policy file called the M-Policy. This file contains one rule for each authorization policy set in the system. This rule is a copy of the policy level target rule included in each set. This rule is a method, in a language such as XACML, to define whether a particular policy is applicable to the given access request and it makes the processing faster. Including it in the M-Policy file has two advantages, namely the processing of the M-Policy file is much faster compared to evaluating the policy level target rule in each file. These rules are optional in XACML. If they are not present, policy evaluation will take longer. Also, we do not use any rule level targets in the XACML policies. As such, we compare the best case performance XACML can offer with our TAPS algorithm. The *'all-that-apply'* algorithm makes it possible to evaluate all target rules at the same place. Each rule in the M-Policy is evaluated (refer to Definition 6). If a rule evaluates to 'permit', it means that the target rule representing the respective policy is true and that policy is applicable. We then include that policy in the TAPS.

To apply the TAPS algorithm to current XACML based systems, we can create an M-Policy file if all the XACML policies in the target system have a policy level target and no overriding rule level target. In systems where either there

are no policy level targets or overriding rule level targets are present, an efficient way to implement the TAPS algorithm is to broadly categorize the available policies and use these categories to select the applicable policies. Although this selection will neither be fine-grained nor accurate, it will still improve the performance of the evaluation system because by using TAPS we can filter out non-applicable policies at an early stage. So, although the performance will not be optimal in this case, it will still be better than the current performance.

The next step in stage one is to determine the applicable PCA (PCA_{apply}) based on a set of environmental attributes, which define the specific conditions under which each of the PCAs is applicable. These environmental attributes essentially define the context of the $\mathcal{AR}$. Some of these attributes might accompany the $\mathcal{AR}$ while others can be provided by an internal or external system entity. We assume that the dynamic decision of which PCA to select is itself based on a policy. Thus, there is a policy set containing the rules governing PCA selection. The PCA rules are defined so that they are mutually exclusive and only one of them is applicable in a particular situation. Although this might seem complex, it is not really so because there are typically a small number of combination algorithms to choose from. This is enforced by using the combination algorithm $\oslash$ *'only-one-applicable'* to choose among the PCAs. *'only-one-applicable'* returns the applicable PCA if one and only one rule evaluates to 'permit'. If zero or more than one rule (and hence the PCA) evaluates to 'permit', then an error code is returned. All rules in the policy set are evaluated and the applicable PCA is selected to be used for resolving conflicts for this access request.

Now in stage two, the final authorization decision is calculated by evaluating the TAPS as $\mathcal{E}(\mathcal{TAPS}) = \{TAPS^{PCA_{apply}}, AR\}\ \phi\ \mathcal{D}^{PS} \rightarrow d$. As defined in Definition 7, in this evaluation, we consider all policies present in the TAPS and evaluate them against the access request $\mathcal{AR}$. The $\oslash$ used in this case is PCA_{apply}, which is calculated in the previous step.

As an example, using this model, Alex can create a PCA selection rule to the effect that if the $\mathcal{E}_{\mathcal{A}} = (\text{'}emergency' = \text{'}true')$, then the PCA 'permit-overrides' is used. The effect will be to allow access to anyone who can satisfy at least one of the applicable policies. On the other hand, in case where $\mathcal{E}_{\mathcal{A}} = (\text{'}emergency' = \text{'}false')$, PCA 'deny-overrides' can be used. This will limit access to holders of those attribute combinations that are not denied by any policy and are allowed access by at least one applicable policy. Since this evaluation is done during each access request, the PCA will change dynamically whenever there is an emergency.

In addition to providing this novel functionality, our framework proposes the use of TAPS to reduce the policy set to be evaluated for each access request. As shown in Section 6, this improves the real time system performance by 4-8 times. Formulation and evaluation of these rules is explained in more detail in Section 4.1.

4. SYSTEM DESIGN AND BACKGROUND MODULES

In this section, we will first present the system design for a generic implementation of this authorization framework, and then describe some background modules used for building the prototype.

4.1 System Design

The proposed system has a two stage authorization process, where in the first stage the applicable policy set and the applicable PCA is determined and in the second stage the applicable policies are evaluated to reach an authorization decision. For the first stage, the policy is created with an index rule for each policy in the TAPS. An index rule is of the form $< \{\mathcal{SA}, \mathcal{RA}, \mathcal{EA}\} : PolicyId >$, where PolicyId is the index id of a particular policy. For example, if policy 'P1234' is applicable to requests in an emergency scenario, then the index rule will be represented as -
$< \{EMT.EMTLicense = \text{'}valid'\} : P1234 >$
$< \{CompanyY.Dispatched = \text{'}true'\} : P1234 >$
$< \{EMT.Employer = \text{'}CompanyY'\} : P1234 >$

The attribute in the index rule is directly provided by an attribute provider (AP)[3]. In this example, the three attributes jointly establish that the EMT's license is valid, he works for company Y and company Y was dispatched to the emergency by the 911 operator. These attributes will be provided by distinct entities. Using them together can establish a complex fact, which cannot be verified by any single entity in the whole system. Note that if an index rule does not contain any attributes i.e. $< * : PolicyId >$, then it is true by default and that policy is always included.

For an access request, the attributes present in the request are compared against the index rules and, in many cases, only a small number of policies will be included in the TAPS. As a result, the policy evaluation stage will be much faster in these cases. The diagram in Figure 1 describes the dynamic authorization process. A similar policy is created with an index rule for each available PCA. Based on the attributes in the index rules, we determine which PCA will be applied to this particular request.

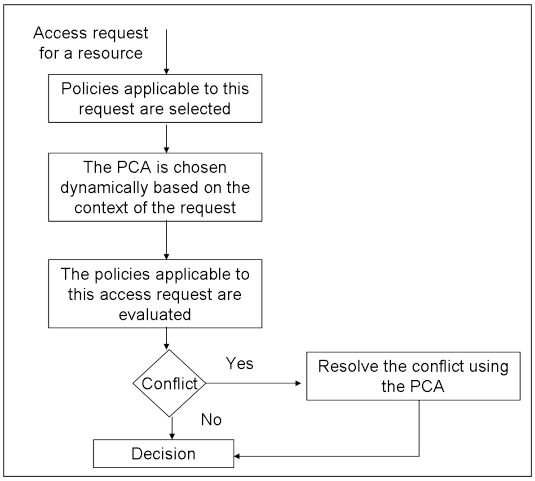


Figure 1: Block diagram of policy evaluation using the proposed framework.

[3]An AP is an entity similar to an identity provider. We define an AP as an entity that can certify certain attribute values for an individual due to its special relationship with the individual. For example, an employer can certify an employee's role in an organization.

4.2 Application Scenario

To understand the implication of using context information in the total applicable policy set (TAPS) evaluation and using dynamic PCA selection, let us again consider the previous health care domain scenario. Assume that Alex's HMO where he stores his PHRs has access policies for data based on criteria like data type, membership type, etc. Alex's policies also apply to his PHR, as described earlier. Now Alex, who lives in Atlanta is planning a trip to Florida for a week and he wants his PHR to be accessible to any physician or 'paramedic in Florida' during that week in case he needs medical help. Using our proposed model, he can add a special policy saying $< \{startdate \leq date \leq enddate\} : P2345 >$, where P2345 describes the special permission to 'physicians' in general and 'paramedics in Florida'. Upon evaluating this index rule, Alex's authorization system will compare the current date with the date range in the index rule and will include P2345 during that particular week. Since the proposed model is attribute based, Alex can take advantage of this by adding multiple attribute combinations. Assume that Alex's location can be tracked from his mobile phone, which communicates that to his authorization system over a secure channel. Then Alex can set the index rule as follows : $< \{startdate \leq date \leq enddate\}, \{location = Florida\} : P2345 >$.

This additional attribute will make sure that the lenient PCA is chosen only when he is physically in Florida[4]. Alex's mobile phone is used to provide his location, but the PHR will be primarily be accesses by the paramedics and physicians using their systems. In the event that he has to cancel his trip, his more lenient policy will not be in effect and his information will not be available to any paramedic in Florida. He also has the convenience of setting this rule once and then forgetting about it, irrespective of whether he actually makes the trip or not.

It is important here to note the difference between creating a new access rule in Alex's policy vs. creating an add-on access policy. While the former is possible using the current authorization systems, it will require Alex to modify his policy by adding new access rules and probably changing the rule combination algorithm. The effects of doing both these actions is hard for an average user to comprehend. If Alex has set his RCA as 'deny-overrides' and he wants to add his new rules to permit access during that particular week, he will need to either change the RCA to 'permit-overrides' or change each of the deny rules in the policy. Doing either is not desirable because his deny rules will be bypassed. In the proposed system, Alex can add a policy to the policy set defining his access policies and change the PCA to 'permit-overrides' for the specified period. Doing so will still keep all of Alex's deny rules unmodified and his policy set will allow access when at least one of his policies allow access, which is what he intended to do. This is hard to do in current systems, because PCA cannot be changed according to dynamic requirements. The resulting policy set is also more modular and analyzing such a policy set is easier. Finally, it saves the effort and complexity of analyzing the effects of changing the RCA or policy rules, not to mention restoring the original state once the specified time has passed. An example XACML policy for Alex is shown in Appendix B.

An additional benefit of our framework is that SHC can create index rules using attributes like 'username'[5], 'datatype', and 'data source' to create index rules to quickly select relevant policies when a physician tries to access Alex's PHR. These relevant policies form the TAPS for this access request. Suppose policy P880 contains Alex's disclosure policies, P130 contains data source's policy, P110 contains HIPPA policy, P112 contains the electronic privacy act[6], and P21 contains the SHC's disclosure policies. SHC's index rules for Alex's PHR are shown below :

$< \{`username = Alex'\} : P880 >$
$< \{`datasourceId = 814820'\} : P130 >$
$< \{`datatype = PHR'\} : P110, P112 >$
$< \{*\} : P21 >$

Note that, in the last index rule, the attribute value is left blank, which results in P21 being included every time. Using this efficient evaluation of TAPS, SHC can quickly determine the policies that need to be evaluated for an access request to Alex's PHR. We report some performance results of the efficiency of TAPS evaluation in Section 6.

5. PROTOTYPE IMPLEMENTATION

In this section, we describe the prototype implementation of the proposed framework. The prototype implementation of the framework extends the functionality of the policy language. The implementation is done using Sun's open-source XACML engine implementation, where we implemented additional modules and PCAs using Java. The generated policies are written in XACML. We use the Sun XACML PDP implementation because its loading and evaluation times are both reasonable when compared to other popular XACML implementations like XACMLLight and XACML Enterprise. Its overall performance is much better than XACMLLight and close to XACML Enterprise. A detailed comparison of the three implementations is done in [25].

The authorization policy consists of multiple policy sets. These sets consist of the system policy, the patient policy, and the data source policy. The system can be extended to consider the data accessor's policy to ensure that the obligations associated with the access request will be honored. The authorization module is set up as shown in Figure 2. The 'Policy Load and Evaluation' and 'Ancillary' modules are part of the standard XACML engine and the 'PSS' and 'PCA Selector' (explained later in this section) are added to the XACML engine. To make the proposed model closely compliant with the existing XACML engine, we have modeled the two new sub-modules as XACML policy sets, so that the XACML policy engine can be used to do these evaluations as well.

Policy Set Selector (PSS) - The PSS takes the authorization policy as the input, which contains all the available policy sets. The schema of the TAPS as a policy file is shown in Figure 3. It is organized in the Subject, Resource, Action and Environment structure. The PSS evaluates each policy set to find out all the sets that are applicable to this access request. The PCA used here is 'all-that-apply', which is es-

[4]We assume that Alex always carries his mobile phone with him because in essence the service is tracking a device and not Alex himself.

[5]The system can use any pseudonym to link Alex's PHR to his policies.

[6]The assumption here is that the rules in these acts can be encoded in a high level language like EPAL or XACML.

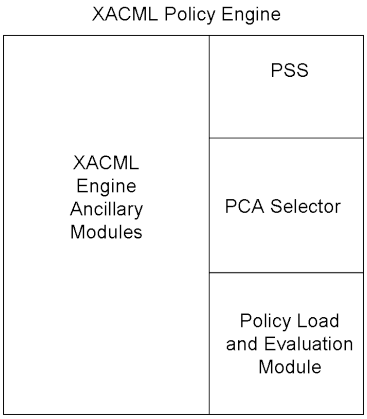


Figure 2: Modified XACML policy engine.

pecially developed for the PSS. The function of this PCA is to evaluate all the policy sets and output all that apply. All the policy sets selected by the PSS are stored in a data structure and only those policy sets are considered in the evaluation phase. As mentioned earlier, this reduces the number of policies to be evaluated for an access request and results in considerable run time performance improvement. A detailed discussion of the performance improvement is given in Section 6.

```
<PolicySet Combination Algo: all-that-apply>
<Policy Description : Policy 1>
    <Subjects>
        Required Attribute Sets
    </ Subjects >
    <Resources>
        Policy set or policy
    </Resources>
    <Actions>
        Include Policy set / policy
    </Actions>
</ Policy>
<Policy Description : Policy 2>
:::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::::::::::::::::::::::::::::::::::::::::::
</Policy>
<Policy Description : Policy 3>
:::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::::::::::::::::::::::::::::::::::::::::::
</Policy>
</ PolicySet>
```

Figure 3: Policy set selector module as a XACML policy set.

PCA Selector - The PCA selector reads the PCA selection file, which is described as a XACML policy. This description is created by the entity that is responsible for making sure that all the relevant policies are taken into consideration. This entity should make sure that the all the available PCAs are encoded as individual policies as shown in Figure 4. This system can be used as a static system by defining the selected PCA with no attributes (hence always applicable) and defining all the other PCAs with attributes that are never true. Although such a configuration may not provide some of the key benefits of the proposed framework, it may sometimes be required for backward compatibility.

The PCA selector file is a policy set as shown in Figure 4. All the PCAs are described as contained policy sets and the combination algorithm used is 'only-one-applicable', which is a standard XACML PCA. It returns 'permit' if one of the policy sets is applicable and 'deny' if zero or more than one policy set are applicable. In case the result is 'permit', the applicable policy set returns the name of the PCA to be used in combining policies. This module provides the novel functionality of selecting the PCA dynamically as described in Section 3.2.

```
<PolicySet Combination Algo: only-one-applicable>
<Policy Description : Policy 1>
    <Subjects>
        Required Attribute Sets
    </ Subjects >
    <Resources>
        PCA Algorithm 1
    </Resources>
    <Actions>
        Use this PCA algorithm
    </Actions>
</ Policy>
<Policy Description : Policy 2>
:::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::::::::::::::::::::::::::::::::::::::::::
</Policy>
<Policy Description : Policy 3>
:::::::::::::::::::::::::::::::::::::::::::::::::
:::::::::::::::::::::::::::::::::::::::::::::::::
</Policy>
</ PolicySet>
```

Figure 4: PCA selector module as a XACML policy set.

To continue with the example in Section 3, the PCA selection policy set will be set as shown in Figure 4. Initially, when there is no emergency, the PCA 'deny-overrides' will be selected. This will be indicated by the attribute 'emergency' being set to false. When there is an emergency, the attribute is set to true and the PCA evaluation will give the output as 'permit-overrides'. The output PCA again becomes 'deny-overrides' once the emergency is over and the corresponding attribute is set to false.

This attribute can be provided by a number of entities like the 'emergency operations center', the '911 operations center', the patient himself or any other entity that the patient's agent trusts to provide this attribute. Although it sometimes might be difficult to ascertain that this particular patient is involved in an emergency, the patient would give more priority to making his PHR available to medical personnel in an emergency rather than to his privacy. Since the entire system can be audited, any breach of privacy can be discovered on audit.

6. PERFORMANCE EVALUATION

In this section, we will discuss the performance evaluation of the various components of the proposed framework. We are basically measuring the following parameters: 1) overhead in evaluating the total applicable policy set (TAPS), 2) overhead in dynamic selection of the PCA, and 3) time saved in evaluating just the TAPS (and evaluating applicable policies) compared to performing a target match on all the available policies (and evaluating applicable policies).

To measure these parameters, we evaluate the following - 1) TAPS evaluation time vs. total number of available policies , 2) PCA evaluation time vs. number of attributes in each index rule, 3) evaluation time vs. number of policies (with and without TAPS). Reasons for choosing these parameters and the evaluation results are discussed in detail in Section 6.2.

6.1 Evaluation Setup

In the evaluation setup, we create XACML policies for the modules described in Section 5. For evaluating the TAPS, we use the schema shown in Figure 3. We setup a XACML policy file with one index rule representing each available policy file (or policy set). Each index rule contains two attributes, both of which are required for access. There are 16 attributes in total and we select 2 out of them randomly. For the experiments, we use 1,2,4 and 8 index rules for each policy file in each run of the experiment. We also vary the total number of available policies from 1 to 10,000 increasing the number of policies by an order of magnitude each time. Most of the real world policies use 10-20 user attributes coming from the organizations LDAP server [22], [3], hence we feel 16 is a representative number. Moreover, this is a configuration parameter and not a limitation because it can be scaled easily. We also scale the number of attributes in one of the experiments (as described in this Section 6.2.2). We believe that most of the real world systems use much less than 10,000 policies. We evaluate performance up to 10,000 policies to observe the system performance over a broad range.

For selecting the PCA, we use the schema shown in Figure 4. Since we have a fixed number of PCA's in the system, we use this evaluation to scale up the number of attributes from 2 to 10,000 in each index rule. This evaluation gives us an estimate of the evaluation time in a system with large number of attributes.

For evaluating the actual policies, we have created policies with 1,2,4 and 8 rules per policy to be used in different runs of the experiment. We created sets of 10, 100, 1,000, and 10,000 policies.

All experiments were run on a single 2.4GHz Intel Dual Core Pentium machine with 2 GB of physical memory.

6.2 Evaluation Results

In this subsection, we present the performance results for the different cases just described.

6.2.1 Case 1

In this case, we evaluate the time consumed in evaluating the TAPS with varying number of total available policies. The RCA used is 'all-that-apply', so the evaluation considers all the policies that apply to a particular access request. We change the number of policies from 1 to 10,000 by increasing the number of policies by an order of magnitude in each step. We also vary the number of index rules applicable to each policy to 1,2,4, and 8 in different runs of the experiment. The result is shown in Figure 5. We observe that the evaluations take almost linear time as shown in this semi-log graph. The evaluation time is within 2 seconds even with 1,000 policies with 8 rules each, whereas with 100 policies with 8 rules each the evaluation time is within 250 milli-seconds.

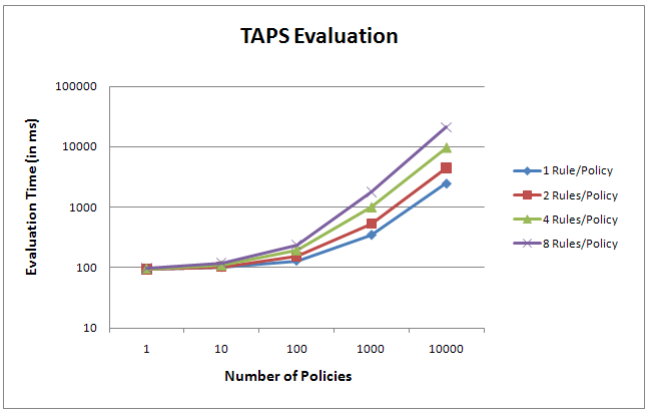


Figure 5: Evaluation time vs. number of available policies.

6.2.2 Case 2

In this case, we evaluate the applicable PCA from a list of PCAs supported by the system. In our prototype system, we have seven PCAs, each denoted as a policy set with its own index rule. We increase the number of attributes used in each index rule to understand the effect of scaling the attributes on performance. We increase the number of attributes from 2 to 10,000. The run time performance is shown in Figure 6. We observe that even with 100 attributes per index rule, the total evaluation time is under 280 milliseconds.

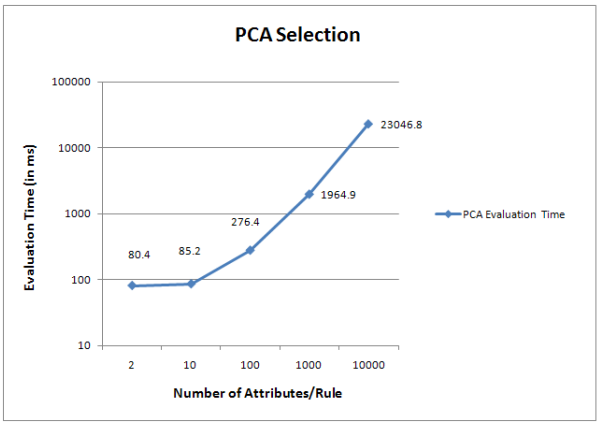


Figure 6: Evaluation vs. number of attributes per index rule.

6.2.3 Case 3

In this case, we evaluate the same set of policies with and without the PSS module and compare the performance of

the two systems. The setup is described in Section 6.1. In each policy file, we have a policy target set up, which is the default method XACML uses to check whether the current policy (file) is applicable to the current request. This target can be set up by resources, subjects, actions, or environments. We set up these targets with applicable subjects values. This allows us to make a direct comparison with our experimental setup. Also, this does not limit the use of target in the experiments conceptually or physically[7]. We first run the test with all the files and let XACML engine perform target matches with all the available policies and evaluate policies where the target matches. Figure 7 shows the result of this evaluation with about 1% of the policies being evaluated.

For comparison with our proposed system, we run the experiment with the same policy set with the PSS module included. We evaluate the TAPS using the index rule method for all the available policies and force the TAPS to be 1% of the total available policies. The resulting TAPS is stored in an array and the XACML engine then performs evaluation of all the files in this array. The combined time for determining the TAPS and evaluating it is shown in Figure 8. We include 1 percent of the total policies in the TAPS, which we believe is more than what most access requests would require, especially in systems with large number of policies. We chose this percentage so that we have a view of the worst case system performance and expect that most real systems will have fewer policies to evaluate per access request and the evaluation times will be lower that what is observed in Figure 8.

Comparing the results in Figure 7 and Figure 8, we observe that using TAPS evaluation with the index rules and then evaluating the applicable policies is about 4-8 times faster than the conventional method. This is specially important in large systems with a lot of policies. Considering the worst case scenario (10,000 policies, 8 rules/policy), the conventional evaluation takes about 210 seconds compared to 26 seconds on our system. In a more common scenario (100 policies, 8 rules/policy), the evaluation times are 1.8 seconds and 0.5 seconds respectively. We argue that this performance improvement is not only significant, but critical for real time systems.

6.2.4 Case 4

In this case, we fix the total number of available policies to 1000 and change the percentage of applicable policies to each access request. We perform this experiment with 15access request. We repeat this experiment for 1,2,4 and 8 rules per policy with and without the PSS system and compare their performance. The results are shown in Figure 9 and Figure 10. We observe that in our proposed model the system evaluation time starts from a very low value and increases linearly. On the other hand in existing systems, it starts at near maximum value and remains almost constant.

7. RELATED WORK

[7]Using target in the policy file is optional in XACML. If no target is used, the only way to check the applicability of the policy is to evaluate it and see if it applies to the current request. This will be slower than matching the target and hence we believe that our comparison is fair because we compare our results with the faster version.

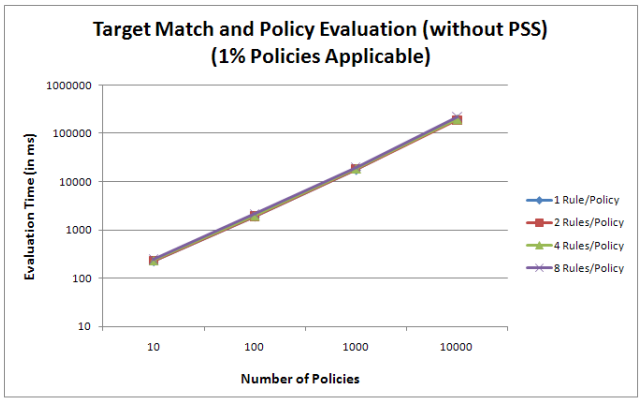


Figure 7: Evaluation time vs. number of total available policies (conventional XACML).

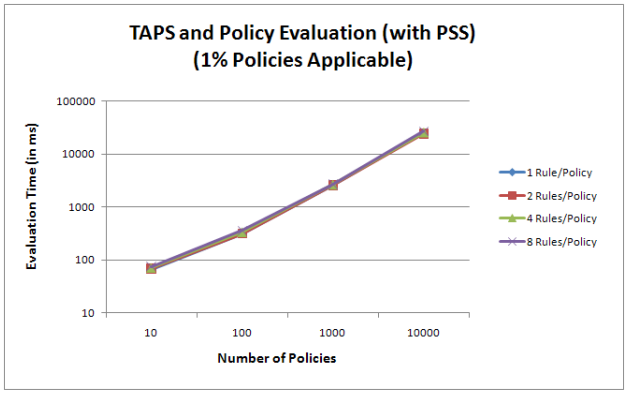


Figure 8: Evaluation time vs. number of total available policies (our proposed framework).

In this section, we review related work in the area of conflict detection, avoidance and resolution works and compare them to our proposed framework.

7.1 Conflict resolution

Mazzoleni, et. al, presented a system for integrating authorization policies for different partners organizations [20]. Their core idea is to find the similarity between a set of policies and to use that information to transform the set of policies into a single transformed policy which applies to the request. In their case, the PCA are static there is no way to choose policies dynamically, whereas in our framework we can choose the PCA dynamically. Our framework also allows multiple policies for the same resource, one of which can be chosen at run time.

Another idea for policy conflict resolution in active databases was proposed by Chomicki et. al, in [10]. Their system is based on the Event-condition-action paradigm in which policies are formulated using ECA rules. A policy generates a conflict when its output contains a set of actions that the policy administrator has specified cannot occur together. This work is specific to dynamically resolving conflicts among actions in a system, whereas our focus is more on a generic policy-based system to protect the resources. In our framework, the policy composers need not have any idea

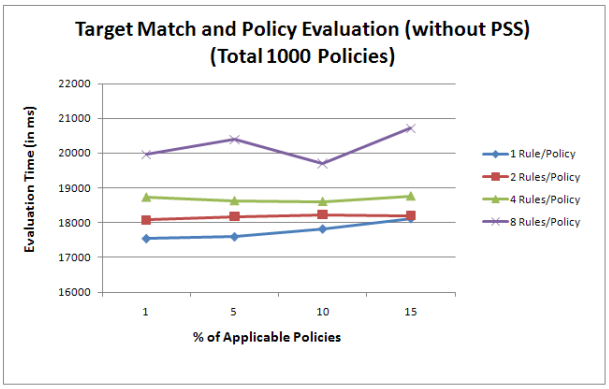


Figure 9: Evaluation time vs. number of total available policies (conventional XACML).

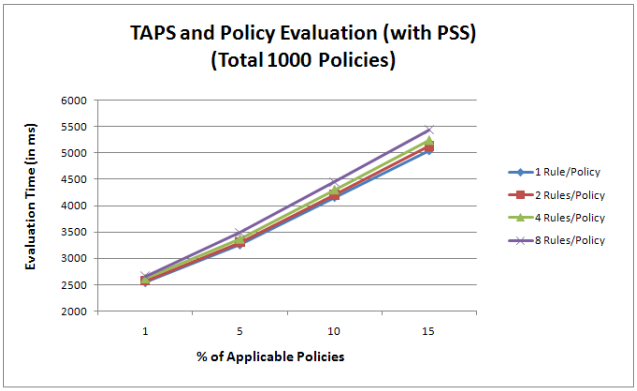


Figure 10: Evaluation time vs. number of total available policies (our proposed framework).

of the possible conflicts in the system, whereas in Chomicki the system administrator specifically defines conflicting actions. Moreover, in our system there can be a number of authorities who can compose the policies and it is not possible for any one authority to have an idea of all the possible conflicts in advance.

7.2 Conflict avoidance

One approach to avoid conflicts in authorization rules is presented by Yu et. al, in [26]. They argue that a large number of rules may apply to a service and detecting and resolving conflicts in real time can be a daunting task. Their system is completely static and assumes that is it always possible to determine priorities ahead of time and avoid conflicts. We argue that this is not possible in dynamic environments and is based on multiple factors like the context of the access request, authorities defining the policies, mandatory policies (like regulatory) vs. optional policies, and environmental factors.

Another approach for avoiding conflicts in policy specification is proposed by Agrawal, et. al, for defining authorization policies for hippocratic databases [5] and [6]. Their system allows system administrators to specify system policies for administration and regulatory compliance and these policies have the highest priority. Users are allowed to specify their privacy preference as long as their policies do not conflict with the system policies. In our framework, the users can specify their preferences even if they have conflicts with the other policies. The users policies may override other polices or be overridden based on context information. Agrawal's framework also does not consider changing system and regulatory policies that may create more conflicts with accepted user policies. Also, it may result in removal of conflicts between the new system policy and previously rejected user policies, which is not handled in this system. In our framework, this will be naturally handled without any action on anyone's part to resolve the conflict.

7.3 Hybrid Approach

Bertino, et. al, presented an approach which is a hybrid of conflict avoidance and conflict resolution [9]. In this work, the authors propose a scheme for supporting multiple access control policies in database systems. Here policies may have 'strong' authorization which are without conflicts or 'weak' authorization with possible conflicts. Compared to this framework, we believe that our approach is more generic because it allows conflicting policies to be composed and resolves conflicts based based on context information. To implement Bertino's proposed system, there should be some static hierarchy (or first specified rule overrides others) for conflict avoidance among strong authorizations. In contrast, our framework will allow dynamic overriding among the authorities.

Another approach to resolving policy conflicts in a hybrid manner is proposed by Jin, et al. [14]. In their work they mention that although resolving conflicts using the static method is easier, it may not be feasible in large systems with large number of policies. The main difference with our framework is that the combination algorithms in their model are defined statically, whereas in our case we decide the combination algorithm at run time based on context information. Also, our framework enables the user to add (remove) PCAs or policies dynamically, an aspect not considered in [14].

8. CONCLUSION

In this paper, we discussed policy-based authorization systems and attribute-based systems. We focus on the multi-authority case, where multiple policies are used to authorize a single access request. In particular, we expose the problems in choosing the PCAs ahead of time i.e. during the policy description. We present a framework to choose the PCA dynamically during run time based on dynamic attributes. The framework also supports choosing the applicable policy sets based on dynamic attributes. This increases the policy evaluation efficiency of the system and modularizes the policies enhancing their analyzability. Using dynamic attributes to determine applicable policy sets at run time provides a novel method to add and remove specialized policies dynamically. We implemented and evaluated a prototype of the authorization system as a module of a modified version of Sun's XACML engine.

9. REFERENCES

[1] Enterprise Privacy Authorization Language (EPAL). *http://www.w3.org/Submission/2003/SUBM-EPAL-20031110/.*

[2] eXtensible Access Control Markup Language (XACML). *www.oasis-open.org/committees/xacml/.*

[3] Ldap authentication attributes. In *http://docs.sun.com/source/817-7647/ldapauth.htmlwp19608*.

[4] P3P: The Platform for Privacy Preferences. *http://www.w3.org/P3P/*.

[5] R. Agrawal, D. Asonov, R. Bayardo, T. Grandison, C. Johnson, and J. Kiernan. Managing disclosure of private health data with hippocratic databases. *IBM Research White Paper*, Januray 2005.

[6] R. Agrawal, P. Bird, T. Grandison, J. Kiernan, S. Logan, and W. Rjaibi. Extending relational database systems to automatically enforce privacy policies. In *ICDE*, pages 1013–1022, April 2005.

[7] A. Barth, J. Mitchell, and J. Rosenstein. Conflict and combination in privacy policy languages. In *Workshop on Privacy in the Electronic Society*, October 2004.

[8] E. Bertino, C. Brodie, S. B. Calo, L. F. Cranor, C. Karat, J. Karat, N. Li, D. Lin, J. Lobo, Q. Ni, P. R. Rao, and X. Wang. Analysis of privacy and security policies. *IBM Journal of Research and Development*, 53, 2009.

[9] E. Bertino, S. Jajodia, and P. Samarati. Supporting multiple access control policies in database systems. In *Proceedings of the IEEE Symposium on Security and Privacy*, May 1996.

[10] J. Chomicki, M. J. Lobo, and S. Naqvi. Conflict resolution using logic programming. *IEEE Transactions on Knowledge and Data Engineering*, 15(1), Januray/February 2003.

[11] K. Fisler, S. Krishnamurthi, L. Meyerovich, and M. Tschantz. Verification and change impact analysis of access control policies. In *International Conference on Software Engineering*, May 2005.

[12] J. Halpern and V. Weissman. Using first-order logic to reason about policies. In *IEEE Computer Security Foundations Workshop*, 2003.

[13] J. Jin, G.-J. Ahn, M. J. Covington, and X. Zhang. Toward an access control model for sharing composite electronic health record. In *4th International Conference on Collaborative Computing*, 2008.

[14] J. Jin, G.-J. Ahn, H. Hu, M. J. Covington, and X. Zhang. Patient-centric authorization framework for sharing electronic health records. In *SACMAT*, 2009.

[15] H. Kamoda, M. Yamaoka, S. Matsuda, K. Broda, and M. Sloman. Policy conflict analysis using free variable tableaux for access control in web services environments. In *WWW Conference*, 2005.

[16] H. Koshutanski and F. Massacci. An access control framework for business processes for web services. In *ACM Workshop on XML Security*, October 2003.

[17] N. Li, Q. Wang, W. Qardaji, E. Bertino, P. Rao, J. Lobo, and D. Lin. Access control policy combining: Theory meets practice. In *ACM SACMAT*, 2009.

[18] E. Lupu and M. Sloman. Conflicts in policy-based distributed systems management. In *IEEE Transactions on Software Engineering*, pages 852–869, Nov/Dec 1999.

[19] A. Masoumzadeh, M. Amini, and R. Jalili. Conflict detection and resolution in context-aware authorization. In *21st International Conference on Advanced Information Networking and Applications Workshops*, May 2007.

[20] P. Mazzoleni, B. Crispo, S. Sivasubramanian, and E. Bertino. Xacml policy integration algorithms. In *ACM Transactions on Information and System Security (TISSEC)*, pages 852–869, February 2008.

[21] A. Mohan, D. Bauer, D. Blough, M. Ahamad, B. Bamba, R. Krishnan, L. Liu, D. Mashima, and B. Palanisamy. A patient-centric, attribute-based, source-verifiable framework for health record sharing. CERCS Tech Report GIT-CERCS-09-11, Georgia Tech, 2009.

[22] L. Ngo and A. Apon. Using shibboleth for authorization and authentication to the subversion version control repository system. In *IEEE ITNG*, 2007.

[23] P. Rao, D. Lin, E. Bertino, N. Li, and J. Lobo. An algebra for fine-grained integration of xacml policies. In *CERIAS Tech Report 2008-21, Purdue University*, 2008.

[24] M. Rouached and C. Godart. Reasoning about events to specify authorization policies for web services composition. In *IEEE International Conference on Web Services (ICWS)*, September 2007.

[25] F. Turkmen and B. Crispo. Performance evaluation of xacml pdp implementations. In *ACM workshop on Secure Web Services*, October 2008.

[26] W. Yu and E. Nayak. An algorithmic approach to authorization rules conflict resolution in software security. In *Annual IEEE International Computer Software and Applications Conference*, July 2008.

APPENDIX

A. 'ALL-THAT-APPLY' COMBINATION ALGORITHM

Definitions:

$P_i = i^{th}$ Authorization policy.
FID = File Identifier.
FID(P_i) = File Identifier for i^{th} authorization policy file.
TAPS = An array to store FIDs. M-Policy = A policy file with index rules to define applicability of authorization policies.

Algorithm:

```
1   Load M-Policy, Access Request (AR)
2   Define TAPS, initialize i=0, counter=0
3   While (M-Policy (index rules))
4       decision = evaluate(index rule i) against AR
5       if (decision == permit)
6       {       { TAPS[counter++] = FID(Pi) }
7       else
8       {       { continue }
9       increment i
10  return TAPS
```

B. ALEX'S POLICY

```
- <Policy PolicyId="Alex's Policy" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  - <Description>
      This is Alex's policy to authorize access to his PHR
    </Description>
  - <Target>
    - <Subjects>
      - <Subject>
        - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">doctor</AttributeValue>
            <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      - <Subject>
        - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">paramedic</AttributeValue>
            <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
    - <Resources>
        <AnyResource/>
      </Resources>
    - <Actions>
        <AnyAction/>
      </Actions>
    </Target>
  - <Rule RuleId="CommitRule" Effect="Permit">
    - <Target>
      - <Subjects>
        - <Subject>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">doctor</AttributeValue>
              <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </SubjectMatch>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">florida</AttributeValue>
              <SubjectAttributeDesignator AttributeId="Alex-Location" DataType="http://www.w3.org/2001/XMLSchema#string"/>
```

```
            </SubjectMatch>
          </Subject>
        - <Subject>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">paramedic</AttributeValue>
              <SubjectAttributeDesignator AttributeId="role" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </SubjectMatch>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">florida</AttributeValue>
              <SubjectAttributeDesignator AttributeId="requester-location" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </SubjectMatch>
          - <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">florida</AttributeValue>
              <SubjectAttributeDesignator AttributeId="Alex-Location" DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </SubjectMatch>
          </Subject>
        </Subjects>
      - <Resources>
        - <Resource>
          - <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">PHR</AttributeValue>
              <ResourceAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" DataType="http://www.w3.or
              /2001/XMLSchema#string"/>
            </ResourceMatch>
          </Resource>
        </Resources>
      - <Actions>
        - <Action>
          - <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
              <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" DataType="http://www.w3.org
              /2001/XMLSchema#string"/>
            </ActionMatch>
          </Action>
        </Actions>
      </Target>
    </Rule>
  </Policy>
```

Figure 11: An example policy for Alex.

An Attribute-based Authorization Policy Framework with Dynamic Conflict Resolution

Apurva Mohan

Douglas M. Blough

Georgia Institute of Technology

Contents

- Problem introduction
- Motivating scenario
- Proposed solution
- Performance of the proposed framework
- Conclusion

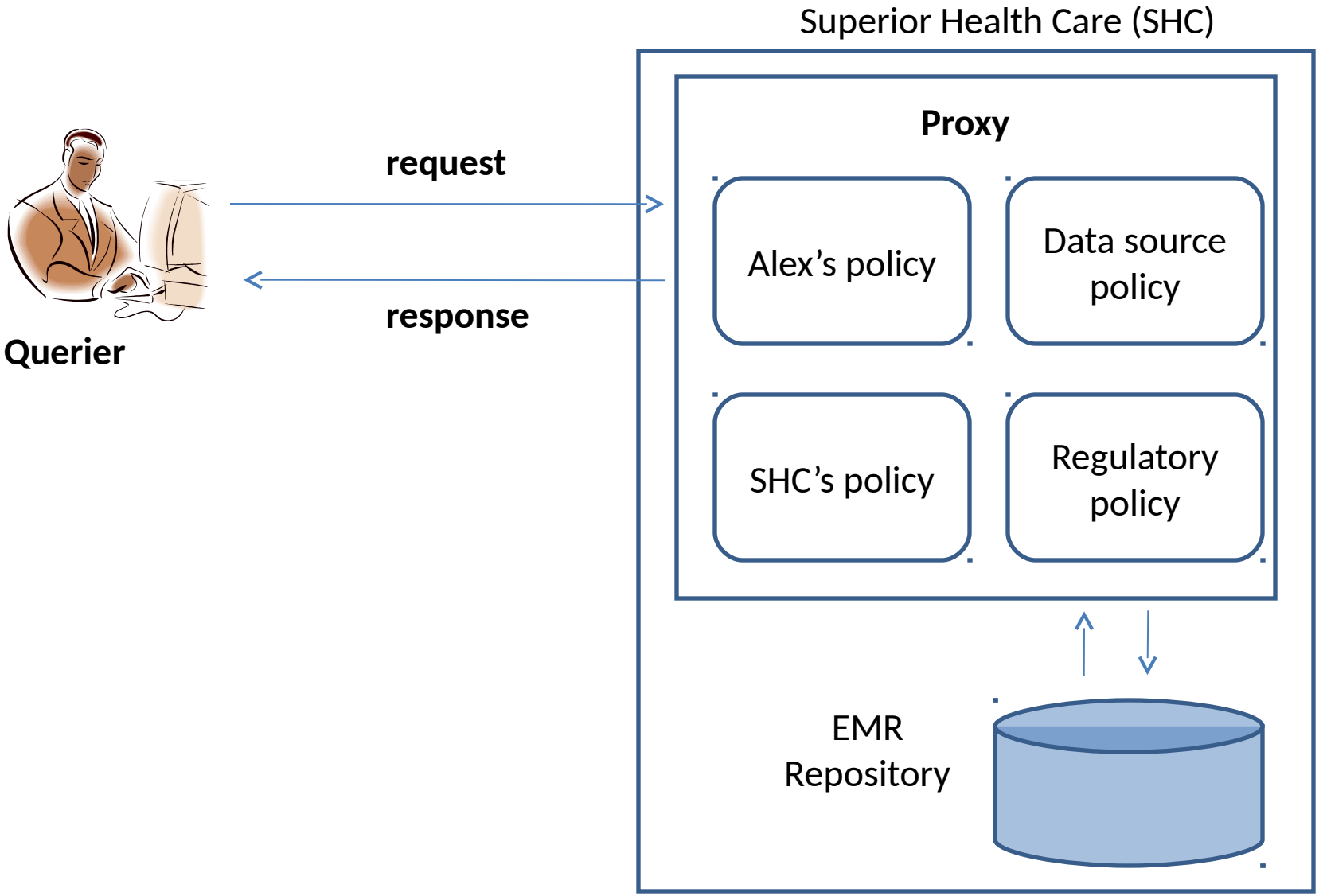
Introduction

- Policy based authorization systems
- Role-based vs. attribute-based systems
- Multi-authority systems
- Conflicts in policy decisions

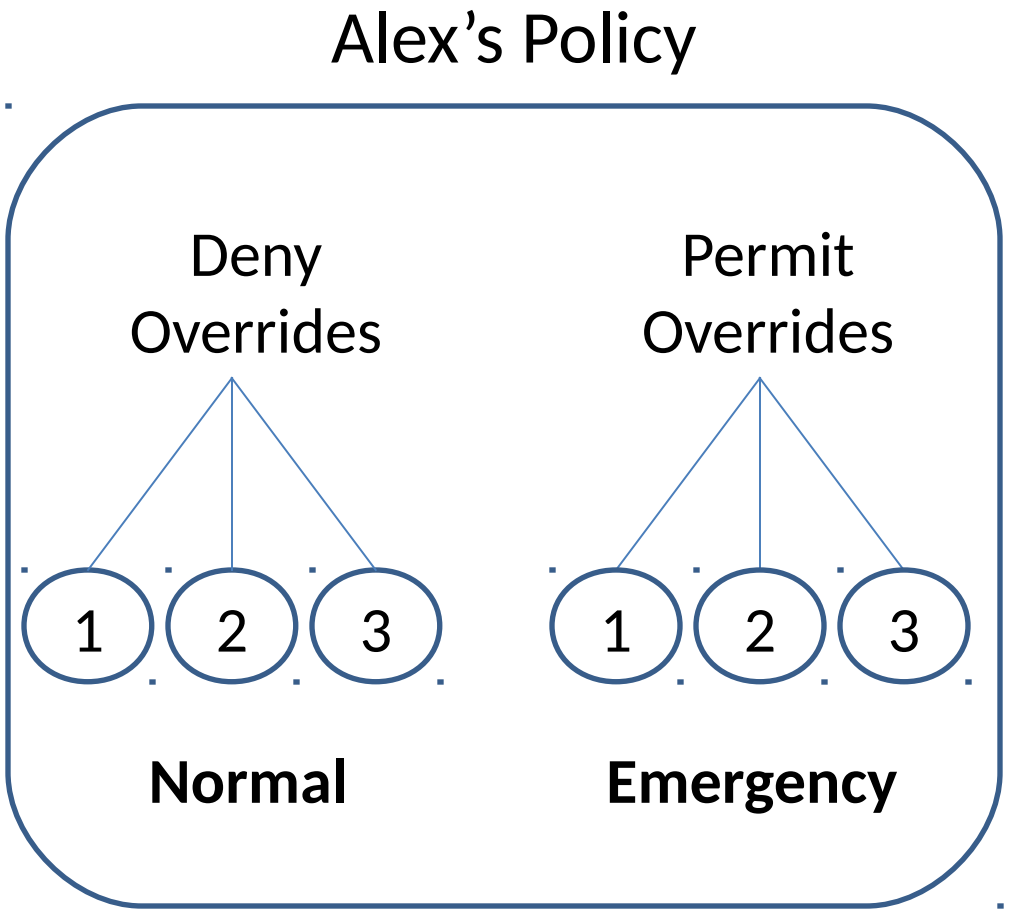
Problem Introduction

- Conflict resolution in current systems is static
- Most policy based systems do not provide modularity
- Difficult to add or remove special purpose policies
- Evaluation of a large number of non-applicable rules
- Fast indexing scheme for finding applicable policies

Motivating Scenario



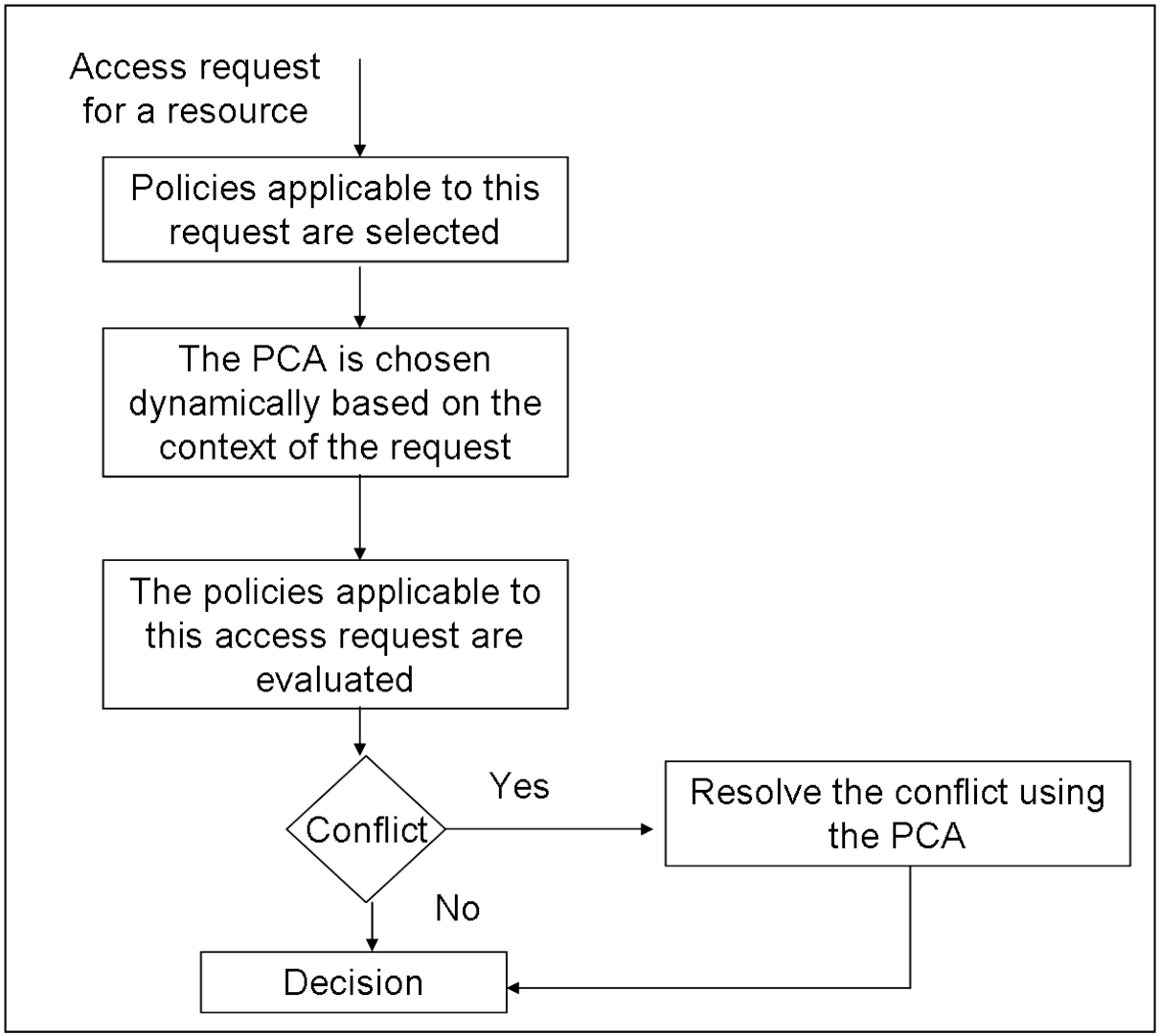
Scenario – Cont.



Proposed Solution

- Dynamic Conflict Resolution
- Decide Applicable policies based on context
- Dynamically include (remove) specialized policies
- Increase modularity of policies
- Increasing the efficiency of policy target matching

Authorization Flow



Proposed Solution - Dynamic Conflict Resolution

<PolicySet Combination Algo: only-one-applicable>

<Policy Description : Policy 1>

    <Subjects>

        Required Attribute Sets

    </ Subjects >

    <Resources>

        PCA Algorithm 1

    </Resources>

    <Actions>

        Use this PCA algorithm

    </Actions>

</ Policy>

<Policy Description : Policy 2>

.....

</Policy>

<Policy Description : Policy 3>

.....

</Policy>

</ PolicySet>

Proposed Solution – Applicable Policies

<PolicySet Combination Algo: all-that-apply>

<Policy Description : Policy 1>

<Subjects>

Required Attribute Sets

</ Subjects >

<Resources>

Policy set or policy

</Resources>

<Actions>

Include Policy set / policy

</Actions>

</ Policy>

<Policy Description : Policy 2>

……

</Policy>

<Policy Description : Policy 3>

……

</Policy>

</ PolicySet>

Motivating Scenario revisited

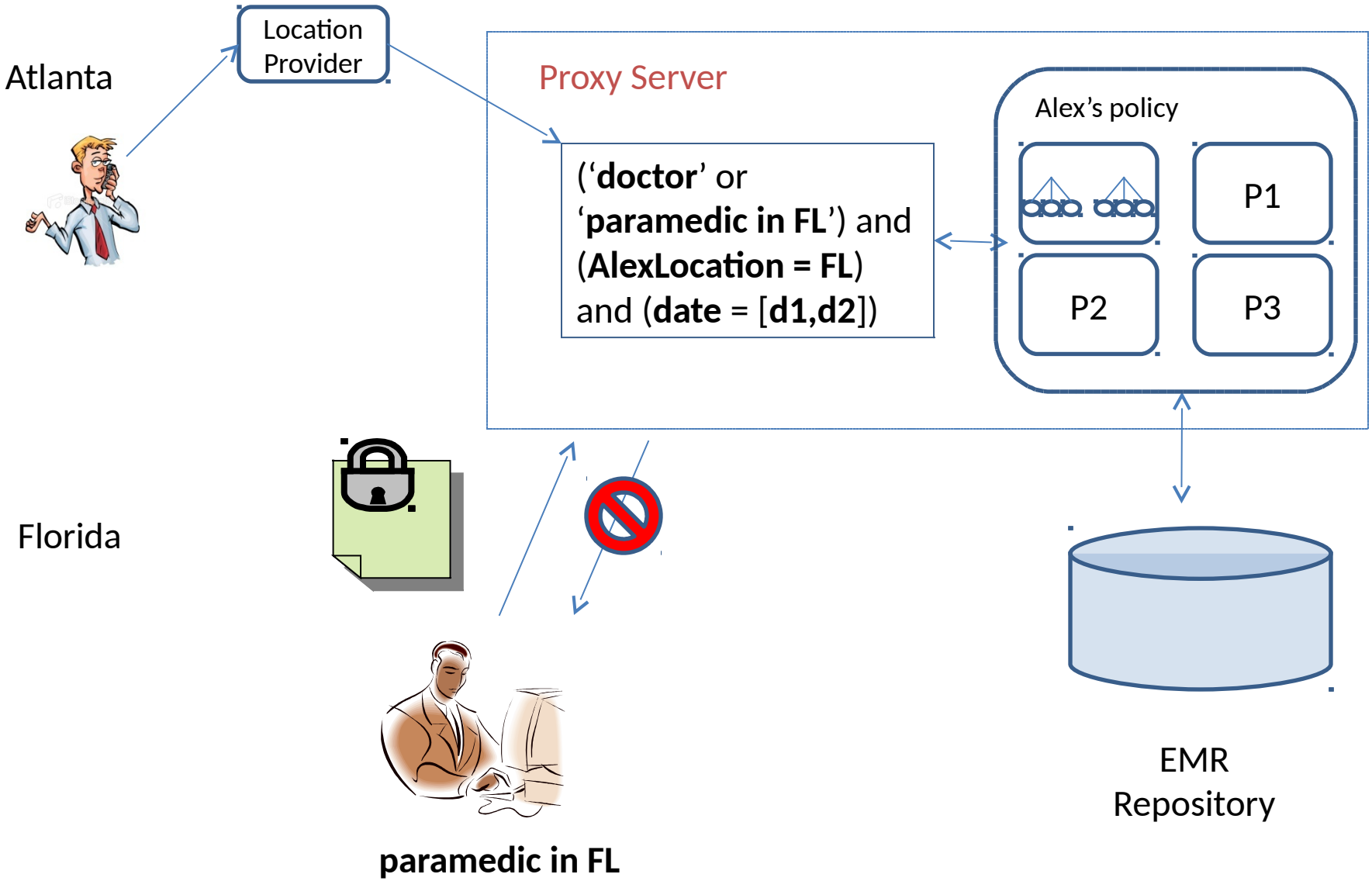
What Alex wants –

- Only his Doctor can access his EMR

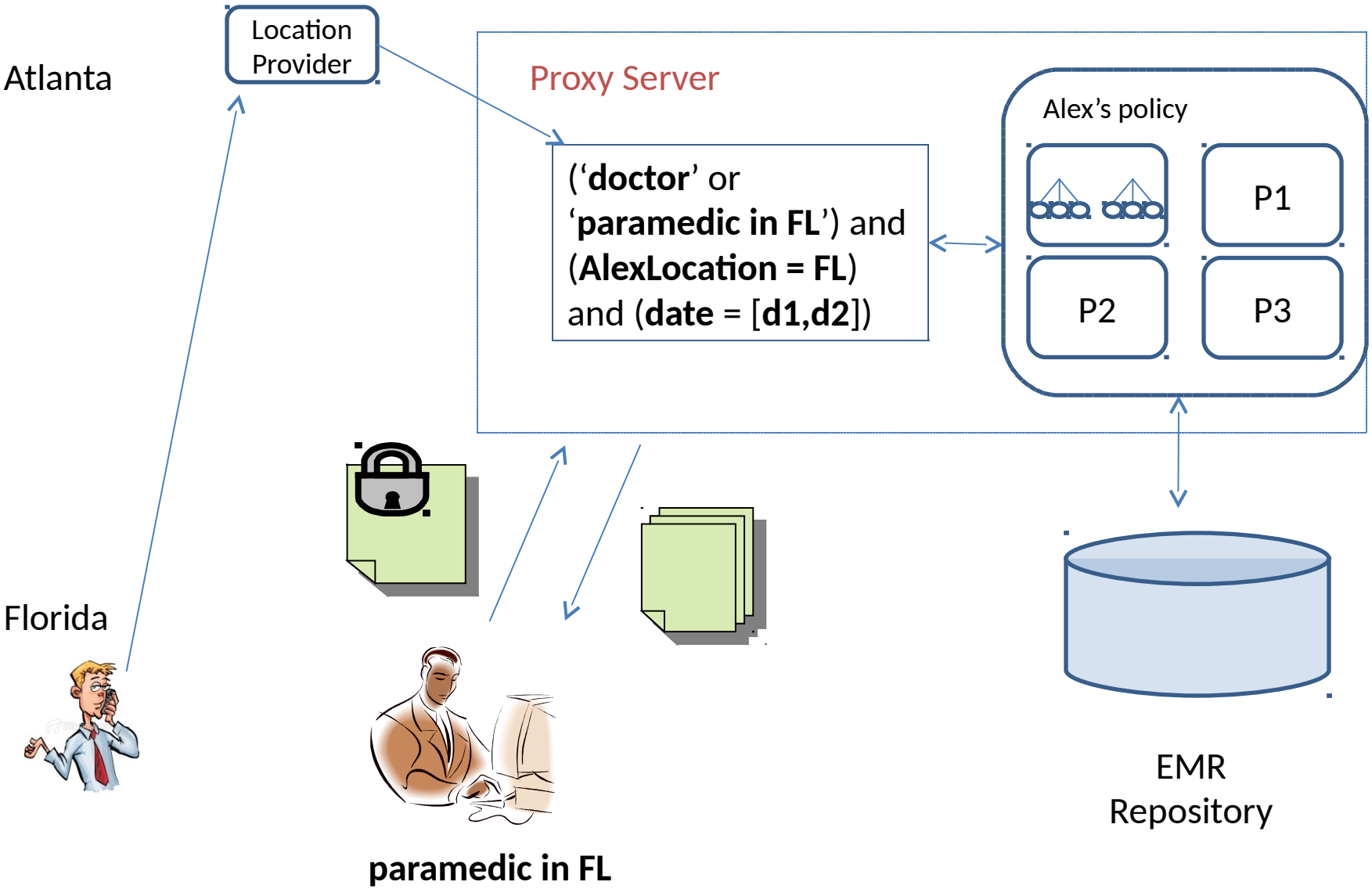
- During his trip, ‘Doctors’ or ‘paramedics in Florida’ can access his EMR

- Attributes used – Alex’s location, Doctor’s credentials, paramedics credentials and location, Alex’s trip duration

Motivating Scenario revisited



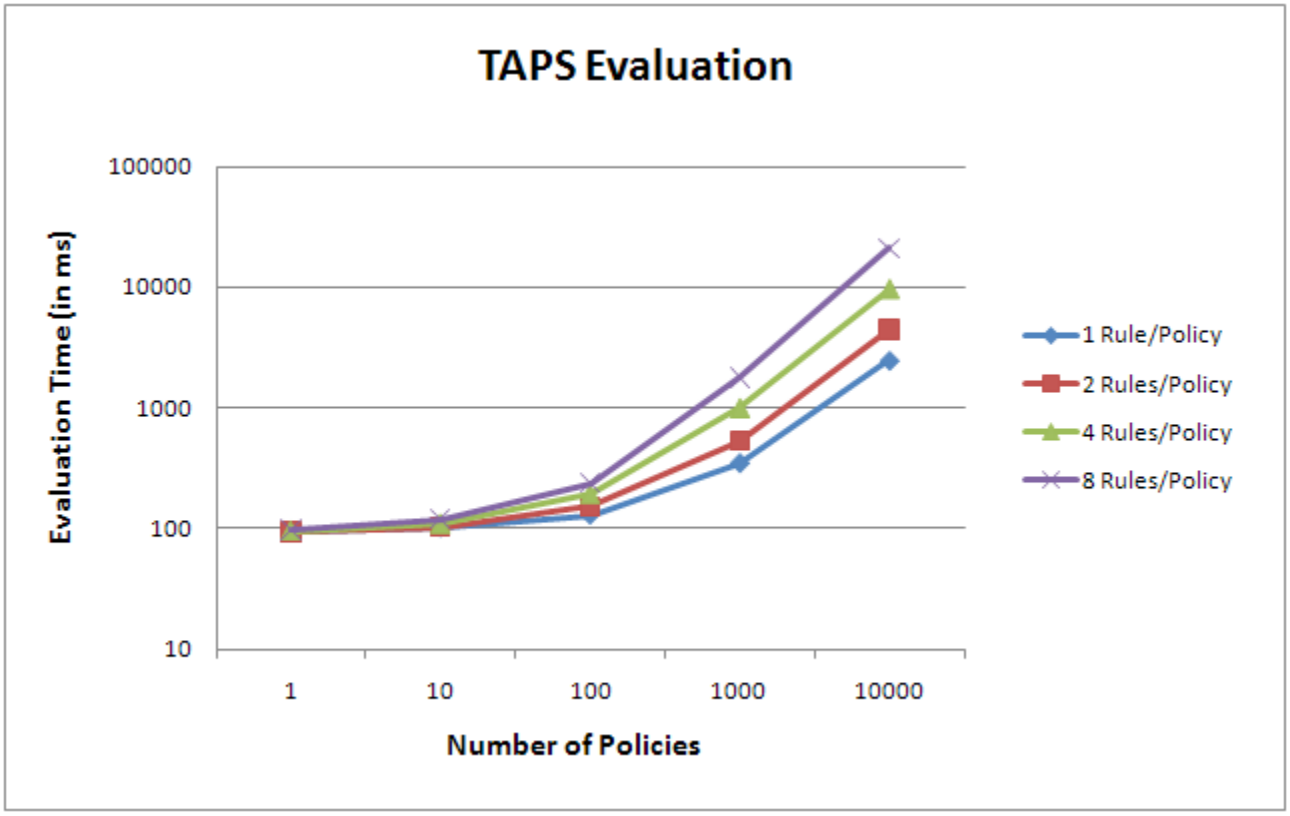
Scenario - Continued



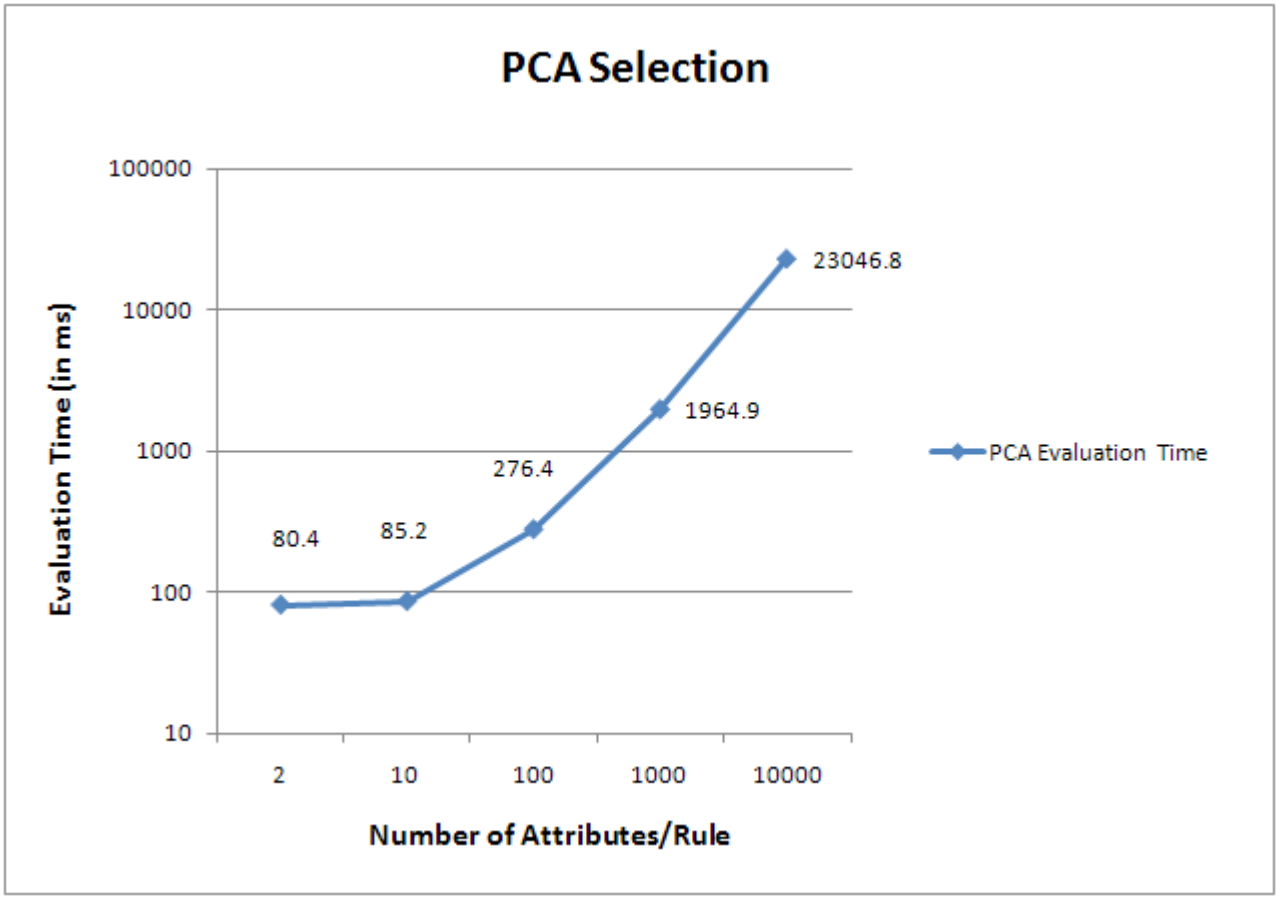
Experimental Setup

- Total Applicable Policy Set evaluation
  - 1,2,4 and 8 rules/policy
  - 1,10, 100, 1000 and 10000 policies
- PCA selection evaluation
  - 7 PCA's, 2-10000 attributes/rule
- Evaluation time
  - 1,2,4,and 8 rules/policy
  - 1,10,100, 1000 and 10000 policies

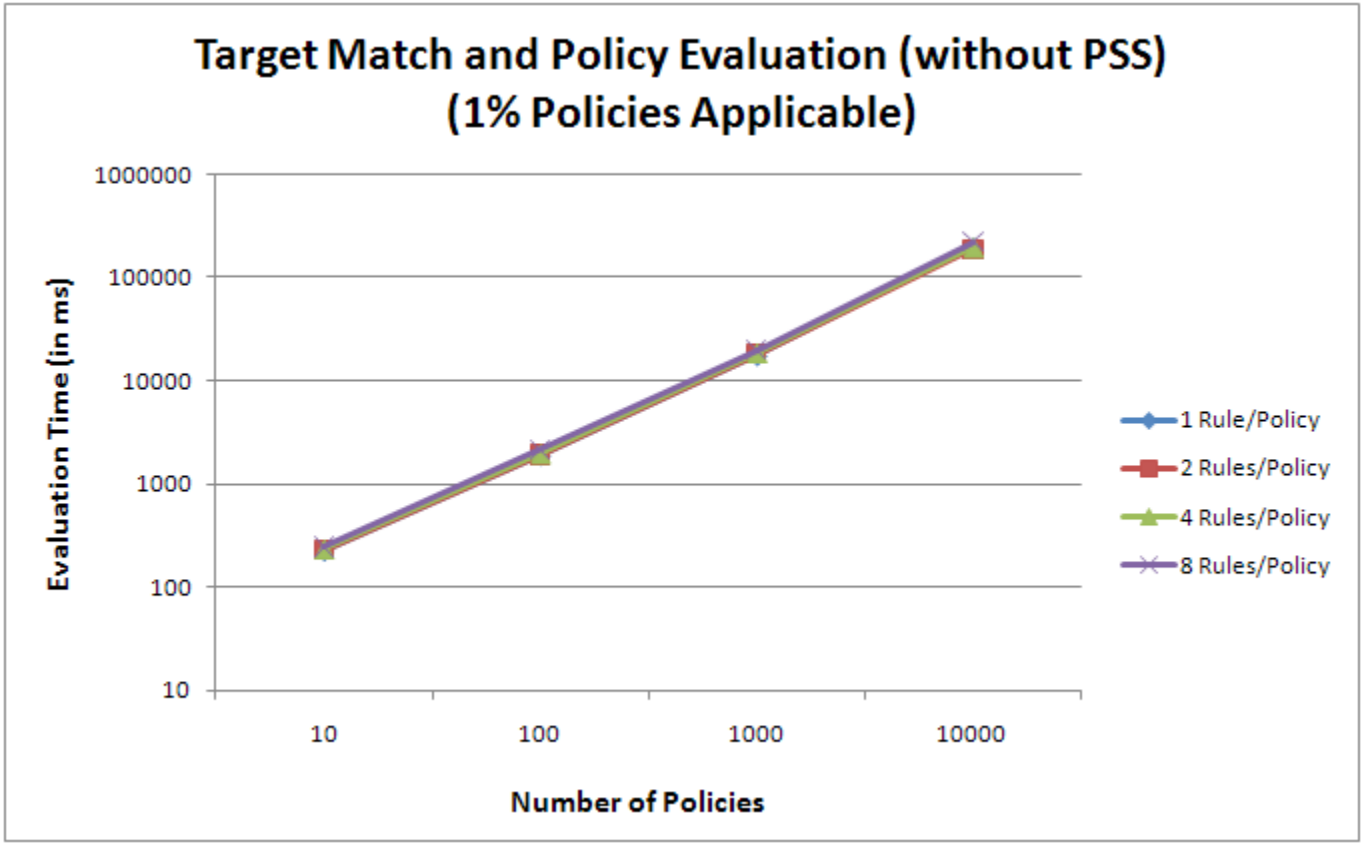
Performance graph - 1



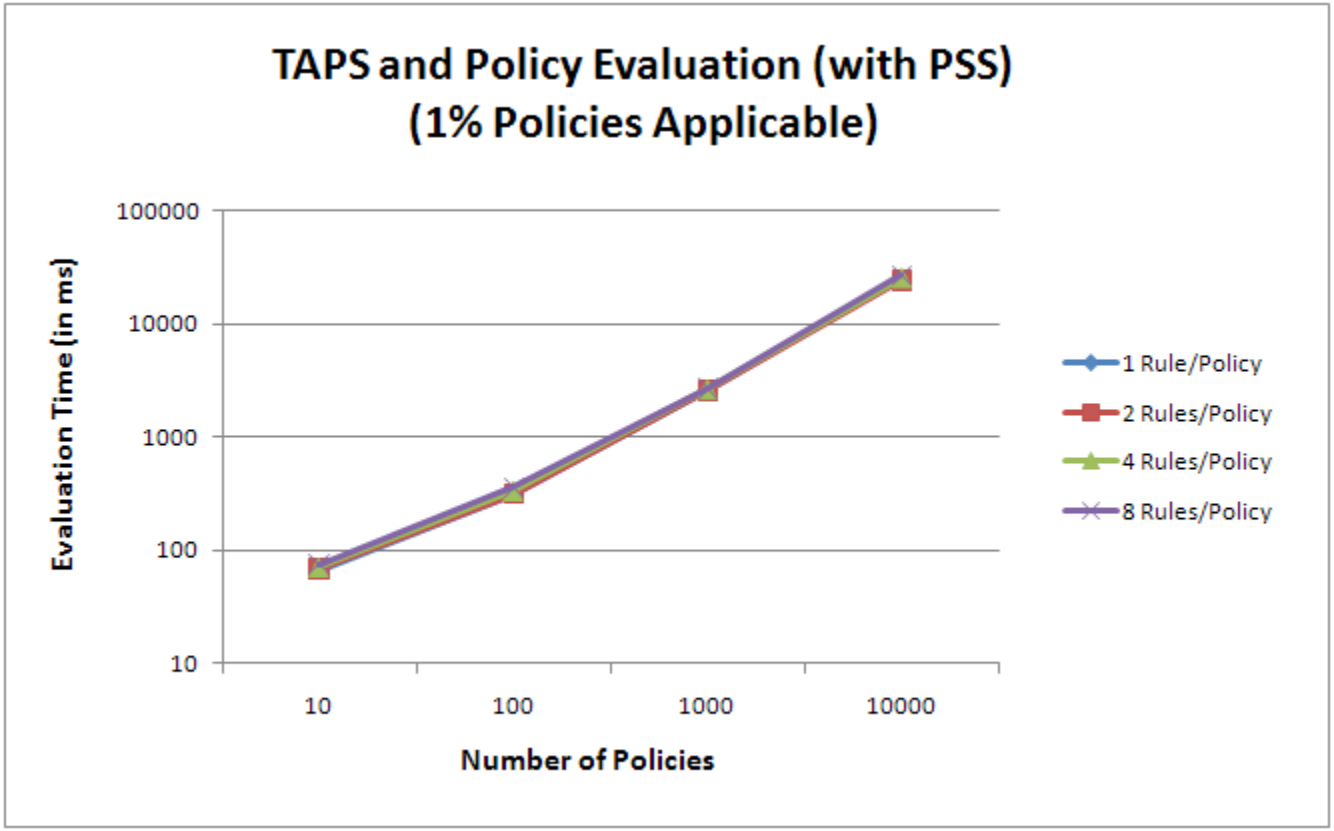
Performance graph - 2



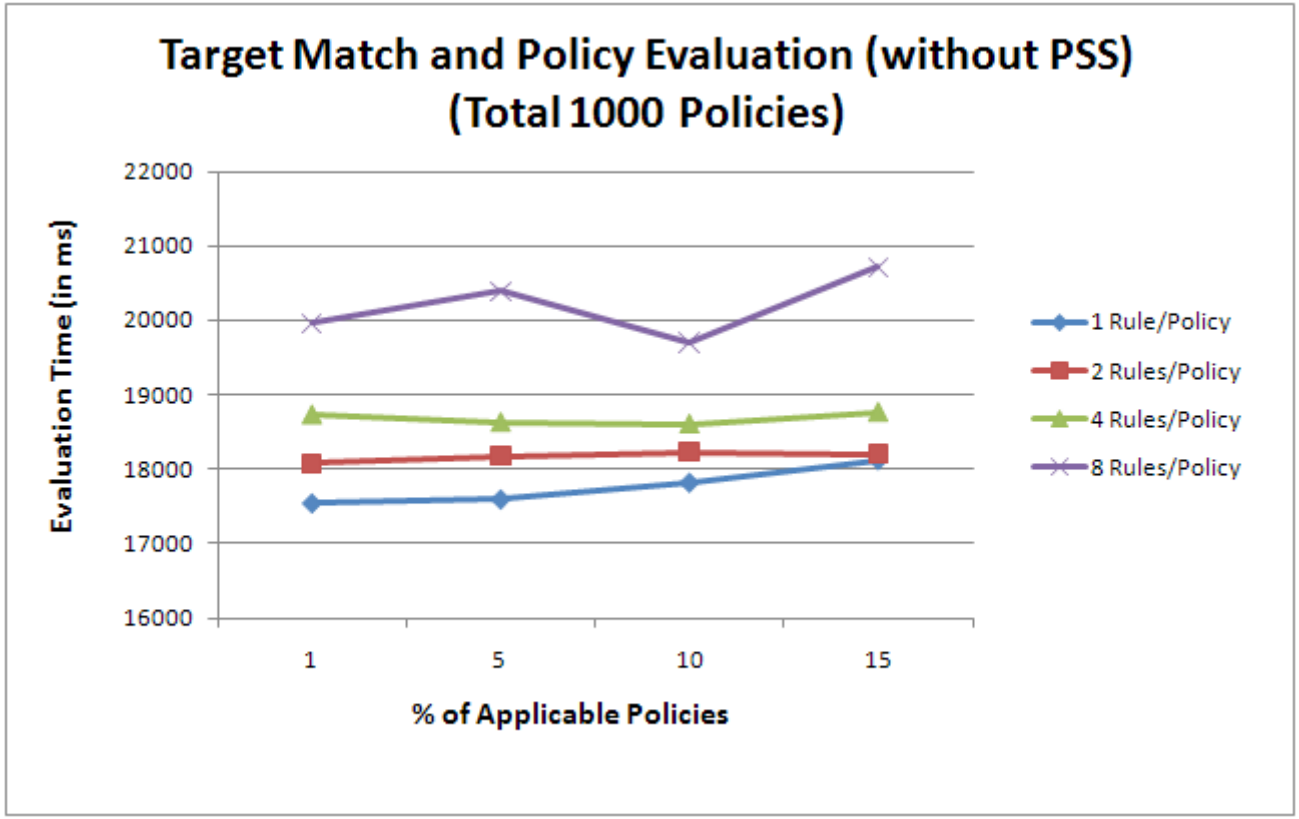
Performance graph - 3



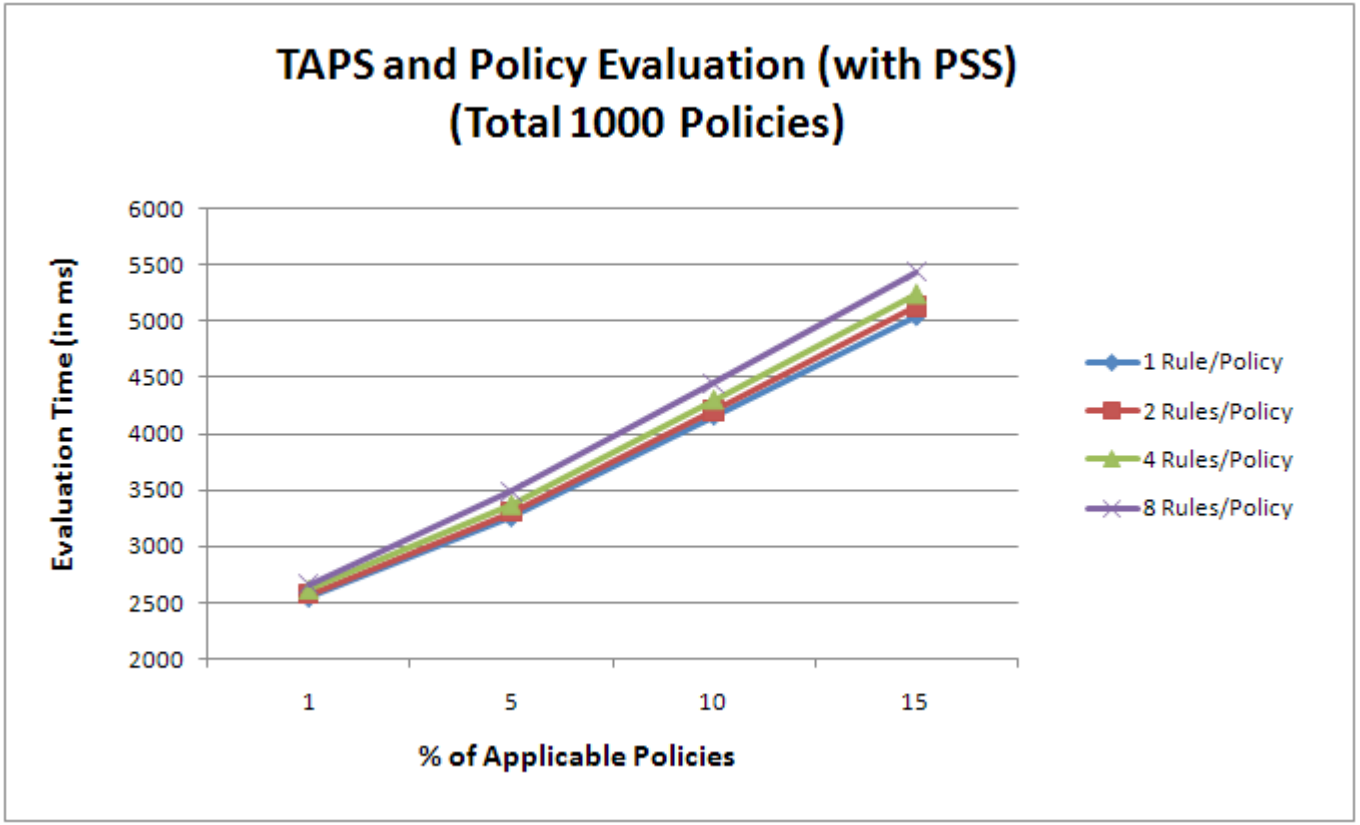
Performance graph - 4



Performance graph - 5



Performance graph - 6



Conclusion

- Proposed a framework for dynamically changing the PCA
- Selecting the applicable policies in a dynamic and efficient manner
- Included modularity in policies
- Add/remove specialized policies dynamically

Questions/Comments?

Computational Techniques for Increasing PKI Policy Comprehension by Human Analysts*

Gabriel A. Weaver
Dartmouth Computer Science Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
gweave01@cs.dartmouth.edu

Scott Rea
Dartmouth Computer Science Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
scott.rea@dartmouth.edu

Sean W. Smith
Dartmouth Computer Science Department
Sudikoff Lab: HB 6211
Hanover, NH 03755
sws@cs.dartmouth.edu

ABSTRACT

Natural-language policies found in X.509 PKI describe an organization's *stated policy* as a set of requirements for trust. The widespread use of X.509 underscores the importance of understanding these requirements. Although many review processes are defined in terms of the semantic structure of these policies, human analysts are confined to working with page-oriented PDF texts. Our research accelerates PKI operations by enabling machines to translate between policy page numbers and policy reference structure. Adapting technologies supporting the analysis of Classical texts, we introduce two new tools. Our *Vertical Variance Reporter* helps analysts efficiently compare the reference structure of two policies. Our *Citation-Aware HTML* enables machines to process human-readable displays of policies in terms of this reference structure. We evaluate these contributions in terms of real-world feedback and observations from organizations that audit or accredit policies.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Methodologies;
D.2.8 [**Software Engineering**]: Metrics

General Terms

Management, Security, Standardization

Keywords

PKI; Certificate Policy Formalization; XML

1. INTRODUCTION

*This work was supported by the NSF (under grant CNS-0448499). The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of any of the sponsors.

IDTrust '10, April 13-15, 2010, Gaithersburg, MD

1.1 Human Analysts and PKI Policy.

Information security policies describe an organization's requirements for protecting their computational and informational assets. In X.509 Public Key Infrastructure (PKI), a natural-language *certificate policy (CP)* is a type of information security policy that documents an organization's set of requirements for trust; furthermore, a *Certification Practice Statement (CPS)* is a natural-language document that describes how the CP is implemented.

As part of the operation of PKI, human policy analysts must regularly retrieve, review and work with certificate policies and the corresponding CPS documents. Often, policy review processes (such as audits, grid accreditation, and bridging) involve comparing a policy or practice statement *under consideration* against a *trusted or accredited* one. During this process, analysts perform several operations on these natural language texts.

- **Finding and retrieving policies**, in practice, is time-consuming and tedious. For instance, in the *International Grid Trust Federation (IGTF)*, although there is a formal distribution of accredited CAs, their corresponding policies documents are not referenced in the distribution metadata. Instead, analysts must manually browse each CA's website (which isn't always listed in the metadata), locate the policy and/or practice statement, and download it.
- **Policy comparison** requires the analyst to compare sections of one policy or practice statement (e.g. "1.1," "3.2.1") with the corresponding sections in another; in theory, these sections should match, but in practice often do not (and may be missing or moved).
- **Policy transform** requires the analyst to manipulate the structure of one policy into another's reference structure (e.g., RFC 2527 or RFC 3647); again, in theory, all policies should match the RFC exactly, but in practice they do not.
- **Policy mapping** requires a combination of policy comparison and policy transform to determine the equivalency of policies and practices within two different PKIs.
- In **compliance evaluation**, the analyst examines how well issued certificates comply with relevant sections of policy. For example, do certificates that have been issued to authenticate to the grid comply with a candidate policy?

- **Content disambiguation** requires the analyst to annotate words and phrases in policy with the specific senses with which they are used. For example, 'reasonable' has a specific legal meaning in Dutch law but not in English law—this caused confusion among policy auditors in the *European Union Grid Policy Management Authority (EUGridPMA)*.

Currently, these review processes are done manually, taking much time and effort. An obstacle hindering all of them is the fact that the processes are all defined in terms of the underlying semantic reference structure of the policies—but human analysts are instead confined to working with the page-oriented PDF text—which may or may not match the reference structure. Auditors therefore must manually translate, in their heads, between policy page numbers and the reference structure in order to do these operations. This forces these operations to be largely manual and/or operate on the entire document. Figure 1 sketches this situation.

1.2 Our Vision

Our overarching research vision is to accelerate PKI policy operations by building automated tools to eliminate slow and error-prone manual processes. In addition to our team's real-world PKI operations experience, we also bring a secret weapon: experience in building automated tools to assist *classics scholars* in overcoming a similar obstacle: doing semantic analysis on page-navigable reference works [20]. (In this earlier paper, we helped apply simple clustering algorithms and text-mining techniques to empirically illustrate how Homeric scholia (scholary comments written in manuscripts) were transmitted, arguably rewriting the past 200 years of theory regarding their transmission.)

As a first step towards achieving this vision, we applied the *Canonical Text Services (CTS) Protocol* (a tool we used in classics work [19]) to construct the *PKI Policy Repository* [18]. Our *PKI Policy Repository* solved the policy retrieval problem. Before, analysts had to manually find and then browse each CA's website. Using the repository, analysts request an arbitrary fragment of policy, the request is encoded as a CTS-URN [10](a hierarchical, machine-actionable, human readable reference string), and the appropriate passage is retrieved. Using this machine-actionable reference framework, we reduced the time to aggregate data for CP comparison by up to 94% (*Policy Reporter*) and reduced the time to map policies from hours to seconds (*Policy Mapper*).

In this current paper, we report on further progress in achieving this resarch vision. In particular, we focus on the *human-computer semantic gap* between the machine representation of PKI policies (structured by page) and the ways in which policy analysts interact with policy (structured by reference scheme). We contribute tools and techniques that use computation to help analysts efficiently compare and browse policies:

- Our *Vertical Variance Reporter* computes and reports differences in the reference structure of two policies.
- Our *Citation-Aware HTML* enables machines to search, to style, and to process human-readable displays of policy in terms of this reference structure.

We also discuss the tools we plan to build next in order to complete the vision.

These tools, in combination with our prior work, provide better quality, reproducible, and reliable data upon which policy auditors can base their trust decisions. Figure 2 sketches how we envision these contributions transforming PKI policy operations.

1.3 This Paper

In Section 2 we describe a set of principles and technologies from the Classics that directly inform our research on PKI policy. Section 3 presents motivation: real-world feedback and observations from organizations—like the FPKIPA-CPWG, EuGridPMA, and TAGPMA—that audit or accredit policies. In Section 4 we describe the design and implementation of our *Vertical Variance Reporter* and *Citation-Aware HTML*—and also discuss the next tools we plan to build. Section 5 gives an experimental evaluation of our *Vertical Variance Reporter* and describes the design of several applications that leverage the properties of our *Citation-Aware HTML*. Section 6 reviews relevant work. Section 7 describes future research directions building upon this work, and Section 8 concludes.

2. MAPPING CLASSICAL TECHNOLOGIES TO PKI

Our work adapts technologies from the Classics to construct computational tools that accelerate traditionally, exclusively-manual PKI policy operations. PKI policies are reference works. Analysts need to be able to align policy sections for comparison. Section 5 of RFC 2527 and Section 6 of RFC 3647 effectively define a canonical structure for *Certificate Policies (CP)* and *Certification Practices Statements (CPS)* for authors and users to understand the meaning and scope of these texts.

Traditionally, PKI policy operations require analysts to manually align policy sections for comparison. However, we can regard these natural language texts as reference works, with canonical structures for authors and users to understand the meaning and scope of these texts. (e.g., Section 5 of RFC 2527 and Section 6 of RFC 3647 define the structure for *Certificate Policies (CP)* and *Certification Practices Statements (CPS)*.)

Prior work in the classics (to which we contributed, in fact) provides technologies to help with analogous tasks for the natural language texts that field studies. We can build on these technologies to solve our PKI problem. In this section, we review some principal building blocks the Classics gives us:

- a data model for canonical texts
- a historical distinction between physical navigation and logical reference, and
- a methodology for working with multiple editions of the same work.

2.1 A Data Model for Canonically Cited Texts

Both theoretical work and hands-on experience with digital texts in the Classics (e.g. Homer and Archimedes [17]) over the past twenty years [11] [9] led us to propose in our previous Classics work [20] that all canonically cited texts possess four properties:

1. citable units of a text are ordered

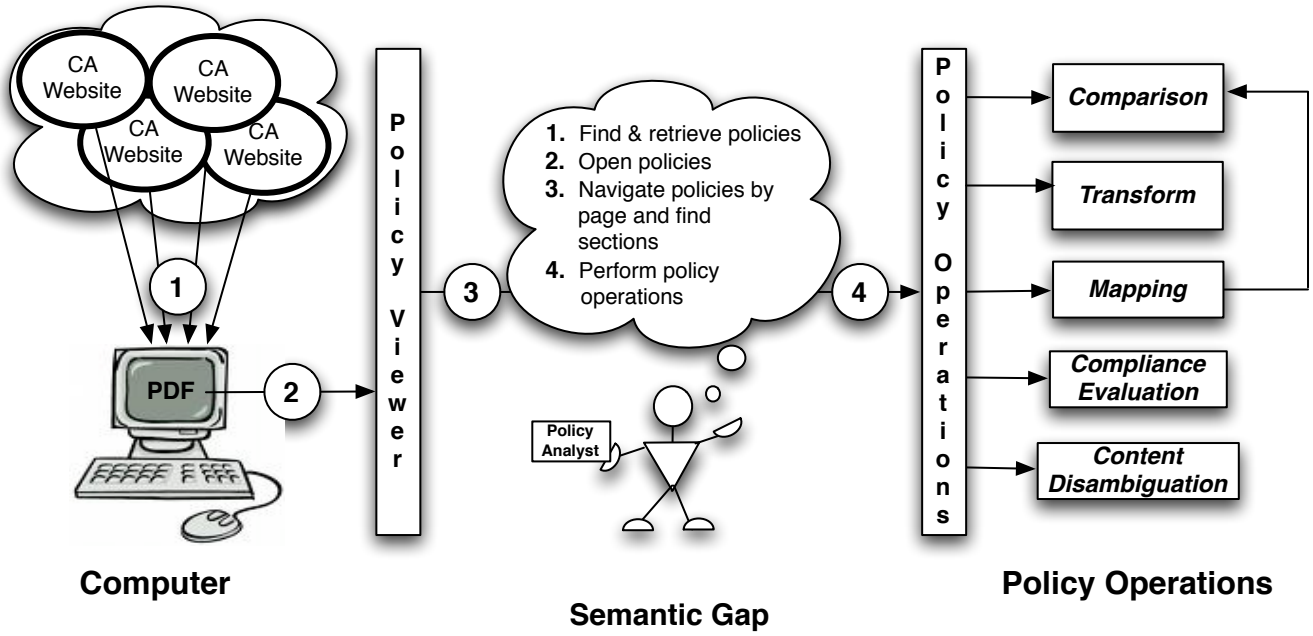


Figure 1: Policy analysts operate on PKI policies by their reference structure, but machine representations of policy like PDF are organized by page. This imposes a semantic gap, forcing policy operations to be largely manual.

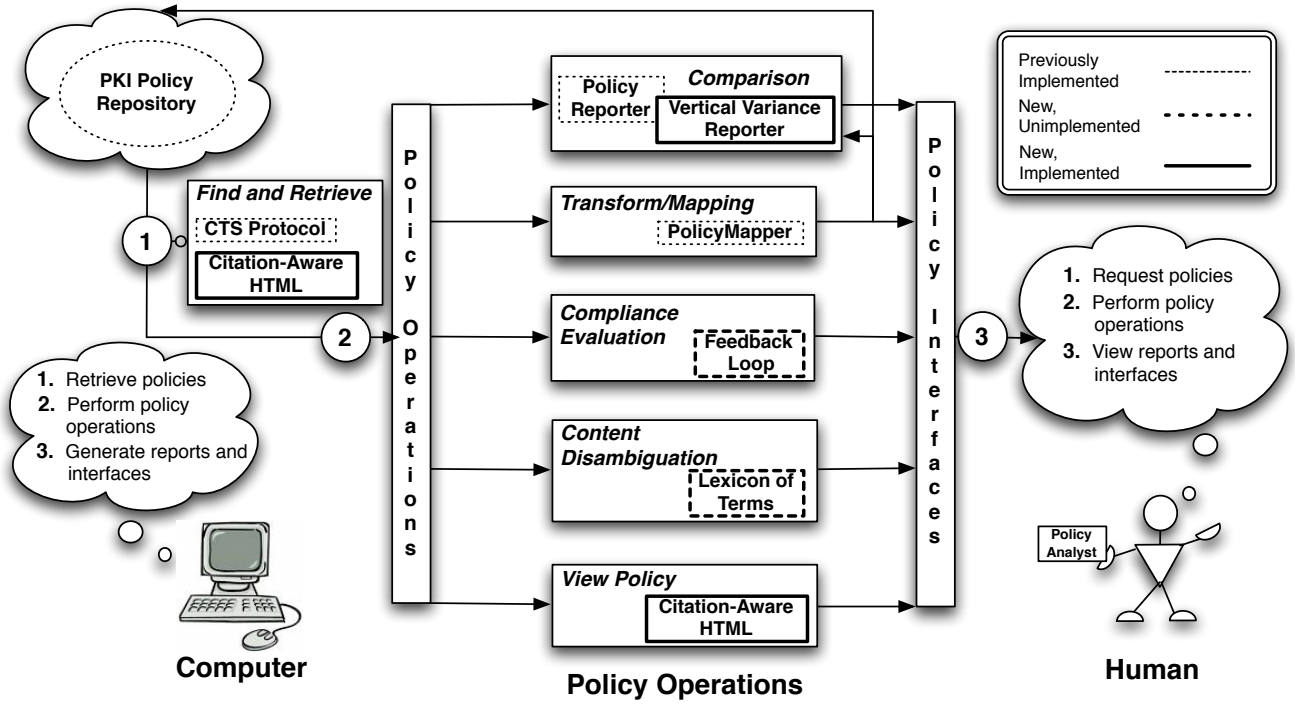


Figure 2: Our representation of policy allows man and machine to directly operate on PKI policy, resulting in reproducible, more reliable data for helping analysts make policy decisions. Please note that previously implemented items were either built or adapted by us.

2. citable units of a text are organized in a (possibly flat) hierarchy

3. versions of a text are related to a notional text in a conceptual hierarchy

4. citable units may include mixed content

The *Canonical Text Services (CTS)* library encodes this data model for canonical texts. Our CTS Protocol [19] defines an HTTP protocol in terms of this data model for referencing and retrieving arbitrary passages of a text.

Our initial work applying Classics tools to PKI contributed the *PKI Policy Repository*, consisting of a CTS server loaded with validated, XML PKI policies. We encoded PKI policies using *Text Encoding Initiative (TEI) P5 Lite*, an XML standard for representing texts in digital form [2]. Like previous efforts to encode policies using XML [5] [4], we modeled a security policy as a tree. This tree corresponded directly to both the hierarchy in the second property of our data model for canonically cited texts and the outline of provisions in Section 5 of RFC 2527 [7] and Section 6 of RFC 3647 [8]. Given a policy's text, we only mark up this hierarchical *reference structure*. By keeping the markup light, we reduce the complexity of encoding a policy.

2.2 Physical Navigation and Logical Reference

The Classics also teaches us the important distinction between physical navigation and logical reference. Originally, when texts such as Homer appeared on manuscripts (MSS), one could reference individual books or lines of the poem, but resolving the reference to a passage of text required manually flipping through the physical MSS folios. With the arrival of the *book* (as opposed to manuscript), the page number and table of contents enabled scholars to quickly resolve logical references (such as "Book 9 of the *Odyssey*") to physical pages for that particular printing. However, over time these tools for physical navigation were used as a citation mechanism [15]. Disciplines outside of the Classics and law, who stuck with logical citation schemes, began citing works in terms of the page. For examples, professors who reference pages rather than logical sections in their syllabi must update their syllabus if the textbook edition or printing changes. CTS advances the historical evolution of text, enabling people and processes to retrieve and navigate texts by their logical structure.

Once policy analysts can use computers to retrieve passages by logical citation, they are no longer required to manually translate, in their heads, between policy page numbers and the reference structure used by many policy operations. In actual practice policies are represented as untagged PDFs that are structured according to the page. Even services such as Google books do not allow one to explicitly retrieve or search within a specific section of a text.

Our overall research vision frees the analyst to continually work in logical reference coordinates whether retrieving, comparing, or mapping a certificate policy. Translation from these logical coordinates to a physical coordinate scheme (byte offsets in a file) is outsourced to the computer. Since the computer can perform this translation, many policy operations can also be augmented with computational tools.

2.3 Working with Multiple Editions

Combining the above properties of canonically cited texts with a citation by logical reference provides Classical scholars with a framework to analyze multiple editions of a text. Versions of a text are related to a notional text (the work) in a conceptual hierarchy. For example, the various translations and editions of Homer's *Odyssey* can be viewed as descendants of a notional work. Although versions may differ, they share (more or less) a common logical reference structure. Book 9 of the *Odyssey* contains Odysseus' adventures with the cyclops Polyphemus regardless of the edition or translation.

Classical scholars also realized that editions may contain slight variations both in logical reference structure, and in textual content. To address these problems, Nagy introduced the concepts of *vertical variance* and *horizontal variance*, distinguishing between differences in structure and content respectively [14].

In PKI operations, we can view the RFC 2527 and RFC 3647 policy formats as notional works according to which individual CAs author editions. Like Classical scholars, policy analysts analyze multiple editions of a text using a common set of logical reference coordinates. Furthermore, different editions may differ in terms of structure or textual content. Like passages in Homer, PKI policy sections may be added or deleted over time. Unlike Homer however, PKI policy passages are identified not just by passage reference (e.g., "(9)") but also by headers that describe the purpose of the section (e.g., "Other Business and Legal Matters"). Therefore, passage reference does not necessarily correlate with section semantics. (This would be like Polyphemus the cyclops occurring in Book 6 rather than Book 9 of the *Odyssey*!) Headers may be relocated and paired with a different passage reference, identifying a different but semantically equivalent section to the corresponding section in the canonical reference structure.

To address these problems in PKI, we developed the *Vertical Variance Reporter* to compute and report vertical variance between multiple editions of a policy under these conditions, enabling policy analysts to see the mapping between two policies' reference structures.

3. REAL-WORLD MOTIVATION

3.1 Feedback

In our prior PKI policy tool work, we developed the *PKI Policy Repository*, *Policy Reporter*, and *Policy Builder*. When we presented these tools to the FPKIPA-CPWG, EuGridPMA, and TAGPMA, these organizations gave us feedback.

Many analysts agreed that a policy repository was desirable for finding policies, understanding the *actual* content of real-world policies, and dynamically creating new policies from previously-accredited, well-understood policies. However, they cited three major obstacles preventing the adoption of our approach: *encoding speed*, *policy variation*, and *display quality*. This current paper contributes solutions to the last two concerns as part of a larger strategy to increase encoding speed—and discusses our plan to eliminate the remaining obstacle.

- *Encoding Speed.* Based upon our prior evaluation of the *Policy Reporter*, we could encode a policy in 4-6 hours by copying and pasting policy content from a PDF into a TEI-XML file.

- *Policy Variation.* Once a policy was encoded and loaded into the *PKI Policy Repository*, analysts could retrieve and run analyses on multiple editions of one or more policy sections, expressed as a set of passage references. However, this approach implicitly assumed that passage reference correlate to section semantics. In the real-world, headers may be relocated and paired with a different passage reference, identifying a different but semantically-equivalent section to that listed in RFC 2527 or 3647. Analysts urged us to generalize our approach to handle the relocation of headers.

- *Display Quality.* Our *PKI Policy Repository* is primarily a service for computer programs; analysts wanted a more human-friendly display of our XML policies. Paragraphs, images, and tables needed to be clearly displayed. Although analysts saw the potential of augmenting their policy operations with computational tools, they required a way to view the XML policy using the traditional typographical conventions that reflect policy structure (for example, using different sized fonts to denote sections and subsections of a policy).

3.2 Observations

In addition to gaining feedback from our work, attending meetings of these accrediting organizations allowed us to directly observe presentations, discussions, and business procedures which would benefit from our computational framework once it could accommodate vertical variance and provide a better human interface for browsing policies.

Policy analysts *manually align* policy provisions before they can compare their content. However, the real world makes this task harder than one expects. Sometimes a policy under consideration contains additional sections that do not map to the trusted or accredited policy. Furthermore, such non-standard sections may contradict statements made in other, standard sections of policy (analysts at the FPKIPA-CPWG call this the *whitespace problem*). Such contradictions, if present in an accredited policy, increase the risk accepted by an accrediting organization. However, a tool that measured the vertical variance of a policy would allow analysts to quickly identify non-standard sections of a candidate policy where these contradictions are likely to occur.

Analysts' current approaches to finding, searching, annotating, and evaluating policies could be accelerated with better human interfaces for browsing policies. Although the IGTF provides a formal distribution of accredited CAs, the corresponding policies themselves are not referenced in the distribution metadata. Analysts searching for terms over the entire text of a PDF policy complained that one could not restrict the search space to a particular section or range of sections. Analysts manually generate matrices consisting of policy sections and comments—so a framework that supported annotation of policy would allow them to dynamically generate these comparison matrices.

Researchers at Trinity College, Dublin presented a suite of unit tests for measuring the validity of a certificate relative to a policy [3]; we saw the potential for combining these automated tools with our suite of policy creation and analysis tools for allowing policy analysts, both non-technical and technically-inclined, to experiment with how modifying a policy's text impacts certificate validity.

4. OUR COMPUTATIONAL TOOLS

As noted above, the policy analysts at the FPKIPA-CPWG, EUGridPMA, and TAGPMA cited three major obstacles to our prior contribution: *encoding speed*, *policy variation*, and *display quality*. We now discuss the tools we built (and the tools are still building) to address these obstacles—and further manual bottlenecks we perceive.

4.1 Completed Tools

4.1.1 Vertical Variance Reporter

Our *Vertical Variance Reporter* addresses the practitioner community's concern over *policy variation*.

In order to determine the *actual* reference structure of a policy rather than imposing an *idealized, trusted structure* such as RFC 2527 or RFC 3647, we extract section identifiers (passage references and their corresponding headers) from its table of contents. Parsing relies upon a library of regular expressions we built to parse common formats for tables of contents. Iterating through these sections, we output a list of section identifiers for the *Vertical Variance Reporter*.

Our *Vertical Variance Reporter* takes two lists of section identifiers as input and computes a mapping between the two that preserves semantic-equivalence. Think about the section identifiers in the *policy under consideration* as being mapped, by some unknown function, to the section identifiers in the accredited policy. We want a way to automatically discover and then calculate this function (or at least a good approximation thereof; the human can do the rest).

To do this, we use one of the secret weapons inspired by the Classical notion of vertical variance: a *confusion matrix* built using the *Levenshtein* metric for semantic distance. [1] The *Vertical Variance Reporter* first records the distance between section headers in the source and target policies. Our tool then processes the confusion matrix to report a bidirectional mapping, classifying policy sections as matched, relocated, or unmapped.[2] In the next few paragraphs, we provide more details about how we compute the confusion matrix and then use it to infer a mapping.

We use a confusion matrix to (1) detect passage references in the *trusted or accredited policy* that are missing from the *policy under consideration*, (2) identify sections in the policy under consideration whose headers are within epsilon of a section header (via the Levenshtein distance) from the accredited policy, and (3) identify sections in the policy under consideration which are further than epsilon away from any of the target policy headers. The rows of the confusion matrix are indexed by the *possible* passage references within source policy given the target. These index values directly correspond to the passage references in the target policy which are used to index columns.

Our tool computes the confusion matrix by iterating over each of the passage references in the target policy and first testing whether it is enumerated in the source policy section list. If the target passage reference does not appear in the source list, a -1 is recorded in the confusion matrix for the entire row. If the source section list does contain the target passage reference, then we calculate the Levenshtein distance between the target header for the current target passage reference and each of the headers in the source. Re-

[1] We use the Levenshtein distance but another metric could be used instead.
[2] It should be noted that this technique may prove useful in clustering documents based upon their reference structure.

sults are recorded in a two dimensional matrix where rows correspond to *possible* passage references within a source policy given the target policy and columns correspond to the target policy's passage references.

The *Vertical Variance Reporter* infers a mapping from two confusion matrices, one comparing sections in the source to those in the target, the other comparing sections in the target to those in the source. In this way, we obtain (1) a list of omitted target references, (2) a list of matched source headers (identified by passage reference), and (3) a list of unmatched source headers. From the target-to-source matrix, we obtain a list of additional source references, a list of matched target headers, and a list of unmatched target headers. By processing these lists our tool is able to classify a section as mapped or unmapped. Mapped sections may be exact matches where the passage references in source and target are equal and the Levenshtein distance is 1, fuzzy matches where the passage references may be different or (inclusive) the Levenshtein distance exceeds a threshold (we used 0.90). Source sections may be unmapped because their passage reference is not present in the target document and their headers fail to match (additional sections) or simply because their headers failed to match any of the target headers (unmatched sections). Table 1 (located at the end of this paper) shows and discusses excerpts of reports generated by our *Vertical Variance Reporter*.

4.1.2 Citation-Aware HTML

In order to address the practitioner community's concern over *display quality*. we developed *Citation-Aware HTML*, which makes it possible for human analysts to search, to style, and in general to manipulate policy in the browser according to logical reference,

Given a list of section identifiers, we use Lucene [12] to index and search Google's OCR HTML for the corresponding byte offset at which the section begins.[3] Our HTML generation process then iterates through these locations, extracting the textual content contained between the start of the section and the next successfully-translated section (or end of file).

Citation-Aware HTML classifies HTML elements using CTS-URNs via the *class* attribute and thereby relates the content spanned by those elements to a policy's reference scheme via machine-actionable reference. Our *Citation-Aware HTML*, like TEI-XML representations of policy, encodes the hierarchy of citable units within a policy. An important consequence of this is that the mapping of *citation nodes* (citable units represented by the *Document Object Model, DOM)* between TEI-XML and HTML is bijective: changes to any citation node in either format can be mirrored in the other since one can generate either format by processing the other.

Our *Citation-Aware HTML* format allows humans to view text using traditional typographical conventions that reflect policy structure while gaining the benefits of navigation by logical reference. Although this technique could be applied to any HTML document, parsing Google's OCR allows us to extract CSS styling information so that eventually we can maintain the typographical conventions in the original PDF policy. This will allow us to faithfully reproduce the *display* of paragraphs, lists, and tables and may be useful for their eventual *encoding* in TEI-XML. Furthermore, our technique lends itself to several policy-browsing applications whose design we discuss below.

[3] Note that we are using Lucene to translate a logical reference coordinate system to a physical coordinate system (bytes) for our machine representation (HTML file).

4.2 Tools Still Under Development

4.2.1 Policy Encoding Toolchain

We are addressing the practitioner community's concern over *encoding speed* with our *Policy Encoding Toolchain*. Encoding a PDF policy with our *Policy Encoding Toolchain* requires the following three steps: (1) use Google Docs to generate Google's OCR HTML output for a given PDF policy, (2) parse this HTML to generate a TEI-XML encoding as well as CSS styling information, and (3) generate a high-quality, human-readable view of the policy that faithfully recreates the typography seen in Google's OCR HTML.

Extracting section lists from a policy's table of contents as well as generating *Citation-Aware HTML* are both components of our toolchain that have value in and of themselves. In order to generate TEI-XML from Google's HTML, we must be able to generate a list of sections describing the reference structure we are trying to represent. Our *Vertical Variance Reporter* compares the vertical variance of two policies, allowing us to evaluate the quality of the encoding of a policy using a given list of section headers. However, this same tool is also useful to policy analysts in comparing a *policy under consideration* to a *trusted or accredited policy*. Our *Citation-Aware HTML* is a product of our envisioned toolchain. However, this same format has independent utility as a key component of several of our policy browsing applications which we will now describe.

4.2.2 Policy Browsing

Policy-browsing applications based upon our *Citation-Aware HTML* include a search utility for finding policies or searching within arbitrary sections of policy, a policy annotation framework generalizing the idea of using typographical cues (font size, color, etc) to reflect policy structure, and a policy feedback loop for dynamic certificate validation which relies upon the bijective mapping between HTML and TEI-XML.

Citation-Aware Searching.

Since the *class* attribute of each citation node is annotated with its corresponding CTS-URN, search engines that index *Citation-Aware HTML* should, in theory, be CTS-URN aware. This means that one could search for all IGTF policies, all policies from a particular CA, a particular version of a policy, or a particular passage of a policy by searching for a particular CTS-URN. At the very least, retrieval of a particular edition should be possible since *Citation-Aware HTML* contains a URN in its page metadata. Just as one can use geographic coordinates to restrict a search to a particular region, so can one use CTS-URNs as textual coordinates to restrict a search to a particular region of text.

Policy Annotation Framework.

Although Google's OCR HTML styles content to mimic page typography, for applications like annotating policy, our *Citation-Aware HTML* enables one to style content with respect to its reference scheme. For example, auditors could highlight various policy sections to indicate the presence of an annotation.[4] Alternatively, auditors could just color-code

[4] These annotations could be mined and presented in a matrix.

policy sections to indicate the various levels of compliance or issues that need further review.

Policy Feedback Loop.

Our *Policy-Driven Feedback Loop* allows analysts to empirically explore the effect that changing a policy would have on an actual PKI infrastructure. Figure 3 illustrates our design that would enable policy analysts to iteratively evaluate the effects of changing policy on certificate validity. First, policy analysts issue a request for a passage of policy against which to check the validity of a corpus of certificates. Using a CTS *GetPassage* request, the corresponding TEI-XML is retrieved and used to generate a suite of unit tests. The test results are then presented by controlling the styling of our *Citation-Aware HTML* for the requested policy passage. For example, the RFC 2119 significance level of violated policy assertions could be indicated with different colors, the number of certificates failing to comply with an assertion could be indicated by font size. Policy writers could then adjust the required value or significance of a policy assertion and POST the updated HTML. Since the mapping between TEI-XML and HTML citation nodes is bijective we can construct a feedback loop: the HTML citation nodes can be used to recover the XML. New unit tests can then be generated and new results presented back to the analyst.

The feedback loop depends upon enriching the reference model for policy with assertions on certificate content. Rather than hand-coding unit tests for *every* new version of a policy, we hand tag the expected value, relation, and significance of each machine-enforceable policy statement *once* within the TEI-XML. Our previously-developed *RFC 2119 analysis* tool leveraged the well-defined semantics of MUST, SHALL, and OPTIONAL. Since these words are technical terms, we were able to process occurrences of these words as tokens with a specific meaning. Similarly, by enriching our reference model with a representation for assertions on certificate content, we hope to gradually develop a lexicon of technical terms for disambiguating content and gradually make larger and larger portions of human-readable policy machine-actionable.

Using our extended policy representation, we walk the tree of citation nodes of the requested policy passage and generate a unit-test suite, much as a compiler walks an *Abstract Syntax Tree (AST)*. The expected value, relation, and significance encoded by our model of assertions, are treated as parameters for generating each unit test. Each citable assertion results in the generation of a unit test whose name encodes its corresponding citation node and significance. The unit tests are executed, results interpreted, and used to generate a CSS style to be included in the *Citation-Aware HTML* for the requested passage. Policy analysts may change the values in the assertions, choosing terms from a *controlled vocabulary* derived from our lexicon.

5. EVALUATION

In this section we present empirical and anecdotal evidence to argue that our *Vertical Variance Reporter* and *Citation-Aware HTML* tools satisfy many of the requirements inspired by feedback and observations from real-world policy analysts. (As noted earlier, our other tools are still in development.)

5.1 Vertical Variance Reporter

The Vertical Variance Reporter addresses the need to be able to understand how the structure of policies differs so that one can quickly determine which sections of a *policy under consideration* can be compared to an *accredited or trusted policy*. In this section, we discuss results from experimental evaluations of how the section identifier extraction process affects the ability to infer a policy mapping between source and target policies. During the discussion of results, we will also mention how this tool relates to the feedback and observations from real-world policy analysts.

5.1.1 Parsing Sections from Tables of Contents

The *Vertical Variance Reporter* computes a semantics-preserving mapping between two lists of section identifiers. Our main technique for generating these lists is to parse the table of contents for a policy in Google's OCR HTML output. In order to make claims on how well the reference structure described in a policy's table of contents (TOC) maps to a target reference structure (such as RFC 3647), we need to be sure that we can correctly extract section identifiers from table of contents formatted in Google's OCR HTML. In the first evalution, we chose 10 policies, generated Google's OCR HTML, extracted their tables of contents, and parsed them for section identifiers. (As noted earlier, we are currently building a tool to automate this encoding process.)

Table 2 shows results for the final step: parsing section identifiers from tables of contents.

As one can see, parsing the table of contents of these policies takes only seconds and we successfully extract every header contained therein. It should be noted that the extracted headers may contain minor artifacts from the extraction process such as rogue page numbers and page headers. These artifacts can be easily fixed either with some quick manual editing or global find and replace. The results of Evaluation 1 allow us to say that our section lists, accurately reflect the policy structure described in a policy's table of contents.

5.1.2 Computing Vertical Variance Using Tables of Contents

The second evaluation uses our *Vertical Variance Reporter* to compute the vertical variance between the same 10 source policies and the structure of RFC 2527 or RFC 3647 depending upon the source policy. We use the section lists derived from the tables of contents. This evaluation allows us to see how well the documented structure of a source policy maps to the RFC standard. Results are presented in Table 3.

Looking at the results we see that the AustrianGrid table of contents' closely follows RFC 3647 (containing 267 of the 270 RFC sections) while the TACC Root policy appears to be missing many sections (containing 67 of those 270 sections). Looking at the ULAGrid policy we see that it contains 271 citable units whereas RFC 3647 only contains 270. This indicates an additional section which the report will identify. This kind of information is a useful first step for solving the *whitespace problem*; it identifies sections to policy analysts that are non-standard and therefore may contain potentially contradictory information. Our mapping from the Austrian Grid TOC to RFC 3647 shows that 260 out of 267 citable units were successfully mapped and that the other 7 units were classified as unmapped. Only 65 of the already-reduced 67 sections in the table of contents for

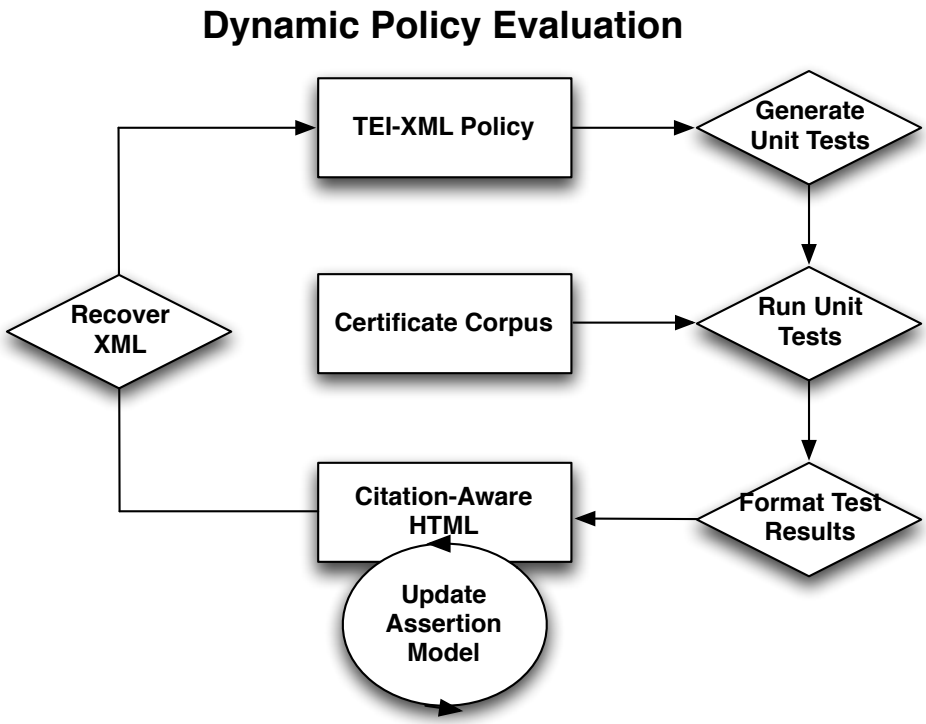


Figure 3: Dynamic Policy Evaluation will allow the policy analyst to treat *Citation-Aware HTML* policies as a form for configuring a certificate policy validation engine. Results of testing the modified policy against a corpus of certificates will be highlighted within the submitted text according to degree of compliance and significance of policy assertion.

TACC Root, actually corresponded to sections seen in RFC 3647. Notice that the mapping from RFC 3647 to Austrian Grid is consistent with its inverse, indicating that we are mapping the same 260 citable units in both directions.

5.1.3 Computing Vertical Variance Using Enhanced Section Lists

Evaluation 3 uses additional sources of information to increase the size of the source section list which we will refer to as $TOC+$. Increasing our section lists is necessary since the tables of contents of some policies do not contain all of the sections *actually* contained in the policy. In Table 3, we see that the DFN-PKI 2.2 policy only contains 79 out of 270 possible sections from RFC 3647. However, looking at the policy text, one sees several sections which its table of contents does not enumerate. Because of this, we paired unmatched passage references from Evaluation 2 with section headers from the target policy, searched for them within our source policy, and if the search returned a unique hit, folded them into our source section header list. Table 3 shows results of this experiment.

Looking at the results, we see that in some cases, this technique increased the size of the enhanced section lists ($|TOC+|$). DFN-PKI 2.2 went from having 79 citable units to 203 citable units. TACC-MICS' policy went from 151 citable units to 270 citable units. This was because TACC-MICS' policy did not enumerate level 3 citation nodes (e.g. "1.3.2") but only levels 1 and 2 (e.g. "1", "1.3" respectively). Many of these newly-inventoried sections could be resolved to an RFC 3647 section: 200 of the 203 citation nodes in the DFN-PKI 2.2 policy could be mapped to RFC 3647. However, some policies did not benefit at all from this approach, the TACC Root policy, with only 67 sections inventoried remained unchanged. On the flip side of the coin, the Austrian Grid policy, with only 3 fewer sections than that of RFC 3647 also remained unchanged. It should be noted that in general, inferring all mappings took between 9 and 45 seconds. Generating enhanced section lists took between 8 and 76 seconds depending upon the size of the section list to be augmented. We ran our evaluations on a MacBook Pro running MacOS 10.5 on a 2.33 GHz Intel Core 2 Duo processor and 2 GB 667 MHz of DDR2 SDRAM.

5.1.4 Comparing Enhanced Section Lists to Ground Truth

Evaluation 4 uses a ground-truth list of policy headers to generate results as in Evaluations 2 and 3. We manually went through each policy and compiled a list of headers in the actual CP or CPS. We then ran the *Vertical Variance Reporter* to infer a mapping between our ground truth lists ($GroundTruth$) and our enhanced section header lists, allowing us to quantify how well we approximate actual policy structure. Table 4 shows results of this experiment.

Our results in Table 5 indicate that headers extracted using our enhanced section list methodology ($|TOC+|$) approximated the actual structure of policies in our corpus with 90.9% to 100% accuracy. Most policies follow the standard format described in RFC 2527 and RFC 3647. The FBCA CP was an exception as it contained 28 non-standard provisions with citation depth 4. For example, Section 6.2.3.4 is found in FBCA CP but is not found in RFC 3647. If one considers only provisions between depths 1-3 inclusive, then we successfully identify between 97.8%

and 100% of all actual provisions. Furthermore, we were able to map our $|TOC+|$ headers to 89.0% to 99.6% of all $GroundTruth$ headers.

5.2 Citation-Aware HTML

As discussed earlier, we developed *Citation-Aware HTML* in direct response to real-world feedback on our *PKI Policy Repository*. In direct feedback, analysts wanted a human-friendly display of XML policies with paragraphs, images, and tables within the policies preserved and presented. In observing policy organizations, we also saw the potential to use better human interfaces for browsing policies to accelerate and improve the process of searching, annotating, and evaluating policies.

5.2.1 Addressing Feedback

Our *Citation-Aware HTML* gives policy analysts a more human-friendly display of XML policies with the potential to exactly replicate the presentational results of Google's OCR output. Currently, we have a basic algorithm for encoding paragraphs. Given that Google does not display embedded images or explicitly *encode* tables in their OCR output, we will hand code image references. The *display* of paragraph, lists and tables will be preserved through styling information which we extract from Google's OCR. However, should individual rows or cells of a table need to be referenced and retrieved by machine, then hand coding their semantic structure within the TEI-XML will become necessary. It should be noted in spite of these limitations, we expect that using our *Policy Encoding Toolchain* to generate XML for most of the policy combined with manual encoding of images and tables as needed, will significantly reduce policy encoding speed.

5.2.2 Leveraging Observations:

Our design descriptions for *Policy-Aware Searching*, a *Policy Annotation Framework*, and a *Policy Feedback Loop for Certificate Validation* all rely upon key properties of *Citation-Aware HTML* to help analysts search, annotate, and evaluate policies. First, we classify citation nodes in the HTML with CTS-URNs, a reference string whose semantics are well-understood and that machines can process, whether to index content for searching or style content according to some meaningful convention. Secondly, we leverage the second fundamental property of canonically cited texts to realize that the mapping between citation nodes in TEI-XML and HTML is bijective. This allows us to create a dynamic *policy feedback loop* that technical and non-technical policy analysts can use to dynamically evaluate the consequences of changes in policy.

6. RELATED WORK

Semantic HTML and Semantic CSS advocates write HTML and CSS that emphasizes the meaning of the text over its presentation [13]. Our *Citation-Aware HTML* subscribes to this philosophy but goes further by embedding URNs to associate semantics with page content. Additionally, others have recommended using Google OCR to convert PDF files into text [1].

The *Policy-Driven Feedback Loop* directly builds upon work done by David O'Callaghan at Trinity College, Dublin [3]. His work will provide us with target and source languages for our policy assertion to unit test compiler. Inglesant, Chadwick, and Sasse developed a controlled vocabulary for configuring access control policies expressed in XML [6]. Our work takes a similar approach, encoding select portions of natural language PKI policies, and deriving a controlled vocabulary from a lexicon of observed words and phrases.

Our work builds upon established standards and mature technologies. TEI P5 [2] represents 15 years of research in encoding texts with XML. The CTS Protocol [19] has been in development for 5 years and is based upon over 20 years of experience [9] in computing with a variety of digitized texts.[5]

7. FUTURE WORK

Using our tools to quantify vertical variance and browse policy in terms of its underlying structure, we will build an IGTF PKI Repository based upon the policies in its distribution. Using confusion matrices we will quantify the structural variance in the IGTF's policies. Knowing which sections of policy are semantically comparable, we will then be able to quantify their horizontal variance.

Two approaches we will employ in quantifying horizontal variance include adding structure to our TEI-XML editions of policy, and using text mining, much as we did in [20], to identify patterns in content with respect to a text's structure. Extending our markup with other data structures, such as assertions, represents a general approach. Most people roughly agree upon the reference structure of a policy. The data models arising from interpreting the text varies greatly. We intend to continue to make content machine-actionable by extending our markup to include structures of interest and to document content values in a machine-actionable lexicon. However, our approach also enables us to use textual content alone to extract topics relevant to trust decisions. With the IGTF repository, we will train classifiers to find all information in a document relevant to a topic. This is of special interest to the FPKIPA-CPWG.

8. CONCLUSION

The *Vertical Variance Reporter* and *Citation-Aware HTML* are our solutions to challenges posed by real-world policy reviewers all over the world. Our *Vertical Variance Reporter* allows analysts to quickly compare the reference structures of two policies and find semantically-equivalent sections between them. Our *Citation-Aware HTML* not only gives policy analysts a nicely-formatted view of policy but also allows us to create a variety of applications for searching, annotating, and evaluating policy. By aligning the textual coordinate systems of man and machine, we have narrowed the *human-computer security policy gap*. Given that human-judgement alone can actually weaken the effects of a security policy [16], we intend to continue exploring how computational tools can support human judgements in the analysis and enforcement of security policy.

9. REFERENCES

[1] Amit Agarwal. Perform OCR with Google Docs âĂŞ Turn Images Into Editable Documents. Retrieved on November 20, 2009 from `http://www.labnol.org/internet/perform-ocr-with-google-docs/10059/`.

[5] We used this experience in designing the CTS Protocol, requiring compatibility with texts encoded in TEI, DocBook, or any other valid XML format encoding a citation scheme.

[2] L. Burnard and S. Bauman. TEI P5: Guidelines for electronic text encoding and interchange. *Text Encoding Initiative Consortium. Retrieved July*, 11:2008, 2007.

[3] David O' Callaghan. Automated Certificate Checks, 2009.

[4] V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak. An Innovative Policy-Based Cross Certification Methodology for Public Key Infrastructures. In *EuroPKI*, 2005.

[5] V. Casola, A. Mazzeo, N. Mazzocca, and V. Vittorini. Policy Formalization to Combine Separate Systems into Larger Connected Network of Trust. In *Net-Con*, page 425, 2002.

[6] David W. Chadwick and A. Sasse. The Virtuous Circle of Expressing Authorization Policies. In *Semantic Web Policy Workshop*, 2006.

[7] S. Chokhani and W. Ford. RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, March 1999.

[8] S. Chokhani, W. Ford, R. Sabett, C. Merrill, and S. Wu. RFC 3657: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, November 2003.

[9] Gregory Crane. The Perseus Digital Library. Retrieved May 29, 2009 from `http://www.perseus.tufts.edu/hopper/`.

[10] D.Smith. CTS-URNs: Overview, December 2008. Retrieved May 29, 2009 from `http://chs75.harvard.edu/projects/diginc/techpub/cts-urn-overview`.

[11] C. Dué, M. Ebbott, C. Blackwell, and D. Smith. The Homer Multitext Project, 2007. Retrieved May 29, 2009 from `http://chs.harvard.edu/chs/homer_multitext`.

[12] Welcome to Lucene! Retrieved November 20, 2009 from `http://lucene.apache.org/`.

[13] Antonio Lupetti. CSS coding: semantic approach in naming convention. Retrieved on November 20, 2009 from `http://woork.blogspot.com/2008/11/css-coding-semantic-approach-in-namin%g.html`.

[14] Gregory Nagy. Editing the Text: West's Iliad. *Homer's Text and Language*, pages 54–56, 2004.

[15] L.D. Reynolds and N.G. Wilson. *Scribes and scholars*. Clarendon Press, 1967.

[16] Stephanie A. Trudeau, Sara Sinclair, and Sean Smith. The Effects of Introspection on Creating Privacy Policy. In *Workshop on Privacy in the Electronic Society*, 2009.

[17] G. Weaver. Semantic and Visual Encoding of Diagrams. Technical Report TR2009-654, Dartmouth College, Computer Science, Hanover, NH, August 2009.

[18] G. Weaver, S. Rea, and S. Smith. A Computational Framework for Certificate Policy Operations. In *Public Key Infrastructure: EuroPKI 2009*. Springer-Verlag LNCS., 2009. To appear.

[19] G. Weaver and D. Smith. Canonical Text Services (CTS). Retrieved May 29, 2009 from `http://cts3.sourceforge.net/`.

[20] G. Weaver and D. Smith. Applying Domain Knowledge from Structured Citation Formats to Text and Data Mining: Examples Using the CITE Architecture. In *Text Mining Services*, page 129, 2009.

Mapping	AustrianGrid Reff	Section Class	(3647 Reff, Score)
$S->T$	1.1	MATCH	(1.1, 1.0)
$S->T$	4.9.2	MATCH	(4.6.2, 0.92), (4.9.2, 1.0), (4.9.14, 0.94)
$S->T$	4.9.5	UNMATCHED	na
$S->T$	6.2.10	UNMAPPED	(6.2.11, 1.0)
$T->S$	6.2.11	ADDITIONAL	na

Passage Ref	AustrianGrid Header	3647 Header
1.1	Overview	Overview
4.6.2	Who may request renewal	Who may request renewal
4.9.2	Who can request revocation	Who can request revocation
4.9.5	Time within which CA must process the revocation request	Time within which CA must procnal
4.9.14	Who can request suspension	Who can request suspension
6.2.10	Cryptographic module rating	Method of destroying private key
6.2.11	n/a	Cryptographic Module Rating

Table 1: Excerpts from a report quantifying the vertical variance of AustrianGrid versus RFC 3647. Row 1 shows that section 1.1 in the Austrian Grid policy exactly matches that of section 1.1 in RFC 3647. However, the mapping from Austrian Grid to RFC 3647 can be more complex. Section headers from the policy under consideration may be ambiguous or not correspond to the accredited policy as shown in rows 2 and 3. Section headers from the accredited policy may be missing in the policy under consideration (as Row 5 seems to indicate for 6.2.11) or relocated. However, looking at Row 4 indicates that section 6.2.11 was moved to section 6.2.10 in the Austrian Grid policy.

Policy	Version	Time (s)	Reff Misses
AustrianGrid	1.2.0	4	0
DFN-PKI	2.1	2	0
DFN-PKI	2.2	2	0
FBCA	2.11	2	0
IRAN Grid	1.3	5	0
IRAN Grid	2.0	2	0
TACC-MICS	1.1	2	0
TACC-Classic	1.2	5	0
TACC-Root	1.2	2	0
ULAGrid	1.0.0	2	0

Table 2: Evaluation 1 shows how we we can parse tables of contents to get an inventory of policy sections. For each of the policies, we parse without missing any sections. This indicates that our section inventories accurately reflect the table of contents (TOC).

TOC	$\|TOC\| : \|TOC+\| : \|RFC\|$	TOC− >RFC						RFC− >TOC				
		Mapped		Unmapped		$\|TOC\|$	$\|TOC+\|$	Mapped		Unmapped		$\|RFC\|$
AustrianGrid	267:267:270	260	260	7	7	267	267	260	260	10	10	270
DFN-PKI-2.1	37:80:270	35	78	2	2	37	80	35	78	235	192	270
DFN-PKI-2.2	79:203:270	75	200	4	3	79	203	75	200	195	70	270
FBCA_CP	281:281:270	242	245	39	36	281	281	242	245	28	25	270
IRAN-GRID-1.3	156:156:193	98	110	58	46	156	156	98	110	95	83	193
IRAN-GRID-2.0	273:273:270	264	264	9	9	273	273	264	264	6	6	270
TACC-MICS_1_1	151:191:270	149	190	2	1	151	191	149	190	121	80	270
TACC_Classic1.2	266:270:270	258	264	8	6	266	270	258	264	12	6	270
TACC_Root_1_2	67:67:270	65	65	2	2	67	67	65	65	205	205	270
ULAGrid_1_0_0	271:271:270	268	268	3	3	271	271	268	268	2	2	270

Table 3: Evaluations 2 and 3 show how well we can classify policy sections as mapped or unmapped. The second evaluation only uses sections from a policy's table of contents (TOC), which the third evaulation uses an enriched list ($TOC+$). In 44 sections, we generate a report for the Austrian Grid that successfully identifies a mapping for 260 of the 267 sections in that policy. We added section headers from RFC 3647 to the headers parsed from DFN's version 2.2 table of contents, resulting in mapping 200 rather than 75 sections.

CP or CPS	$\|GroundTruth\| : \|TOC+\|$	GroundTruth− >TOC+			TOC+− >GroundTruth		
		Mapped	Unmapped	$\|GroundTruth\|$	Mapped	Unmapped	$\|TOC+\|$
AustrianGrid	267:267	265	2	267	265	2	267
DFN-PKI-2.1	80:80	79	1	80	79	1	80
DFN-PKI-2.2	207:203	201	6	207	201	2	203
FBCA_CP	309:281	275	34	309	275	6	281
IRAN-GRID-1.3	157:156	145	12	157	145	11	156
IRAN-GRID-2.0	273:273	270	3	273	270	3	273
TACC-MICS_1_1	192:191	188	4	192	188	3	191
TACC_Classic1.2	270:270	267	3	270	267	3	270
TACC_Root_1_2	68:68	67	1	68	67	1	68
ULAGrid_1_0_0	271:271	270	1	271	270	1	271

Table 4: Evaluation 4 shows how well our method in Evaluation 3 approximates actual policy structure. Looking at TACC Root's CP, we see that only 1 additional provision was identified by manual cataloging rather than automatic extraction. Similarly, only 4 more provisions were identified in DFN-PKI v.2.2. In general our approximation is quite good except for the FBCA CP in which 28, non-standard provisions with citation-depth 4 were identified (e.g. 1.6.2.1).

CP or CPS	$\|TOC+\|/\|GroundTruth\|$	$\|TOC+Mapped\|/\|GroundTruth\|$
AustrianGrid	100%	99.3%
DFN-PKI-2.1	100%	98.8%
DFN-PKI-2.2	98.1%	97.1%
FBCA_CP	90.9%	89.0%
IRAN-GRID-1.3	99.4%	92.4%
IRAN-GRID-2.0	100%	98.9%
TACC-MICS_1_1	99.5%	97.9%
TACC_Classic1.2	100%	98.9%
TACC_Root_1_2	100%	98.5%
ULAGrid_1_0_0	100%	99.6%

Table 5: Using the results in Table 4, we are able to see that our method in Evaluation 3 was able to identify between 90.9% and 100% of all *actual* provisions. Furthermore, we were able to map the $|TOC+|$ headers to between 89.0% and 99.6% of all $GroundTruth$ headers.

Computational Techniques for Increasing PKI Policy Comprehension by Human Analysts

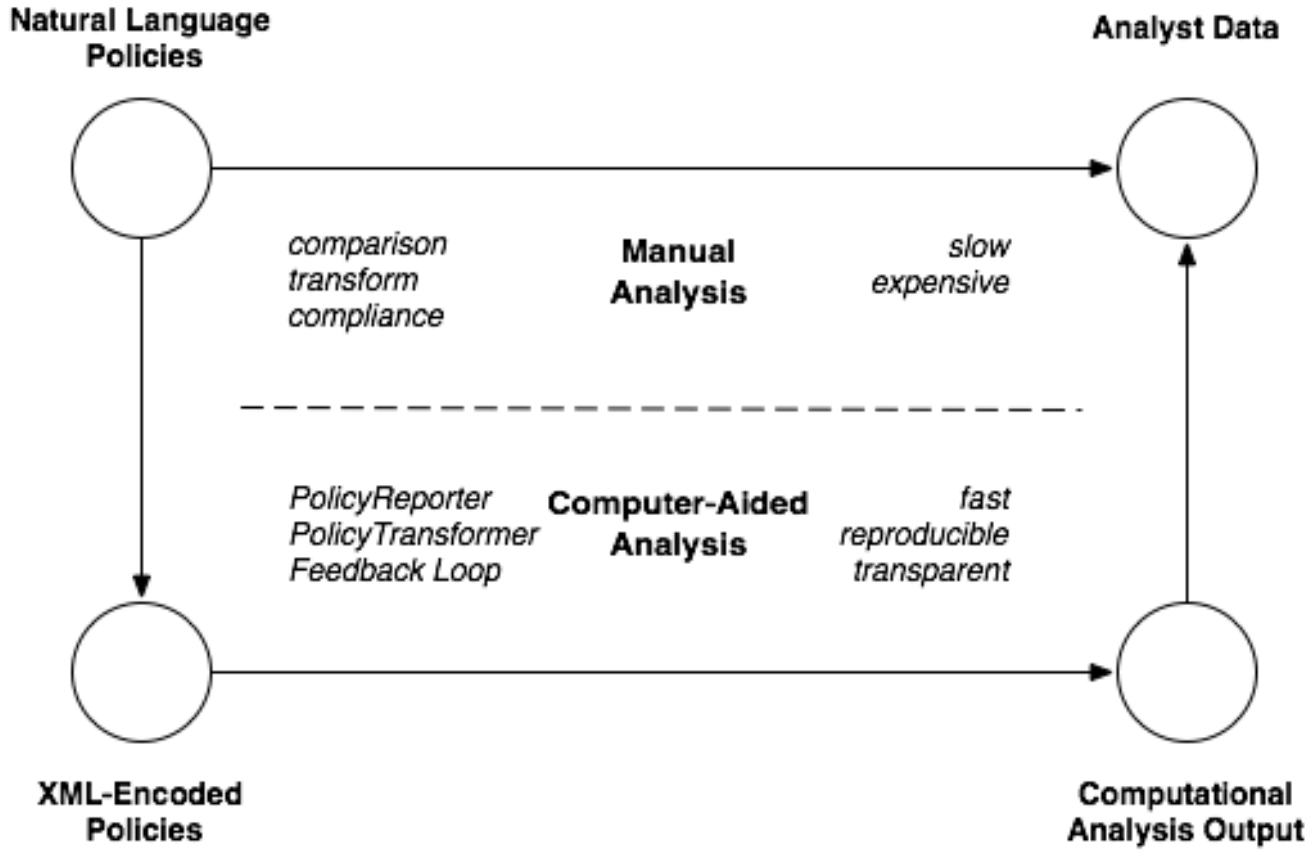
Gabriel A. Weaver, Scott Rea, Sean W. Smith

Dartmouth College

Introduction

- PKI policies define expectations for trust
- Policy review processes include
  - PKI compliance audit,
  - mapping for bridging,
  - and grid accreditation.

Our High-Level Goal



Our Contributions

- **We claim**
  - A human-computer semantic gap forces PKI policy operations to be largely manual.
- **We bridge**
  - That gap with computational tools to accelerate some of these operations based upon real-world feedback.
- **We propose**
  - Future work to accelerate additional policy operations.

Surveying the Semantic Gap

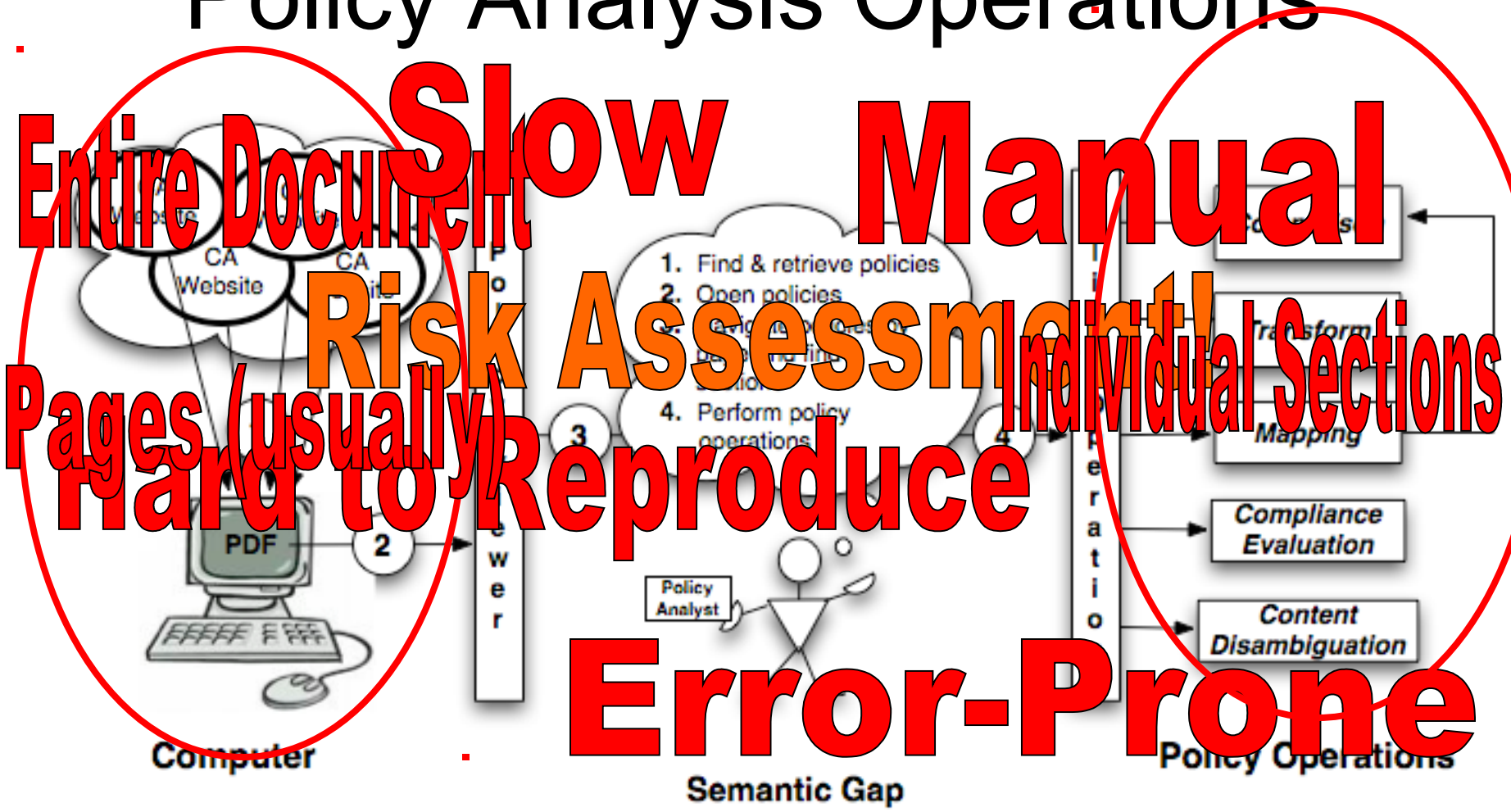
The Problem

Its cloudy in the human-computer semantic gap.

Our Approach

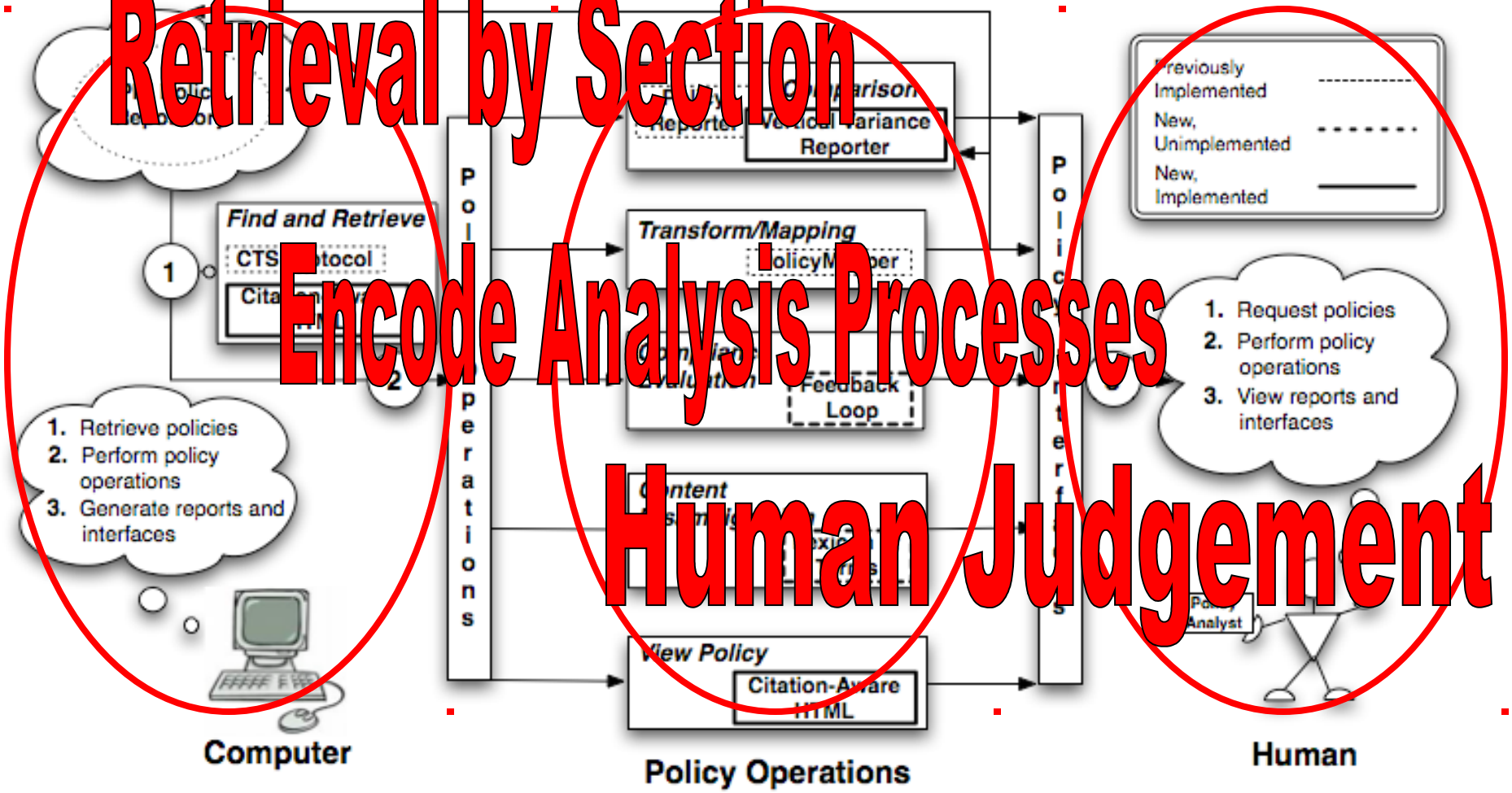
Trust depends upon knowing what to expect.

Policy Analysis Operations



Building Tools for Policy Analysts

Bridging the Gap



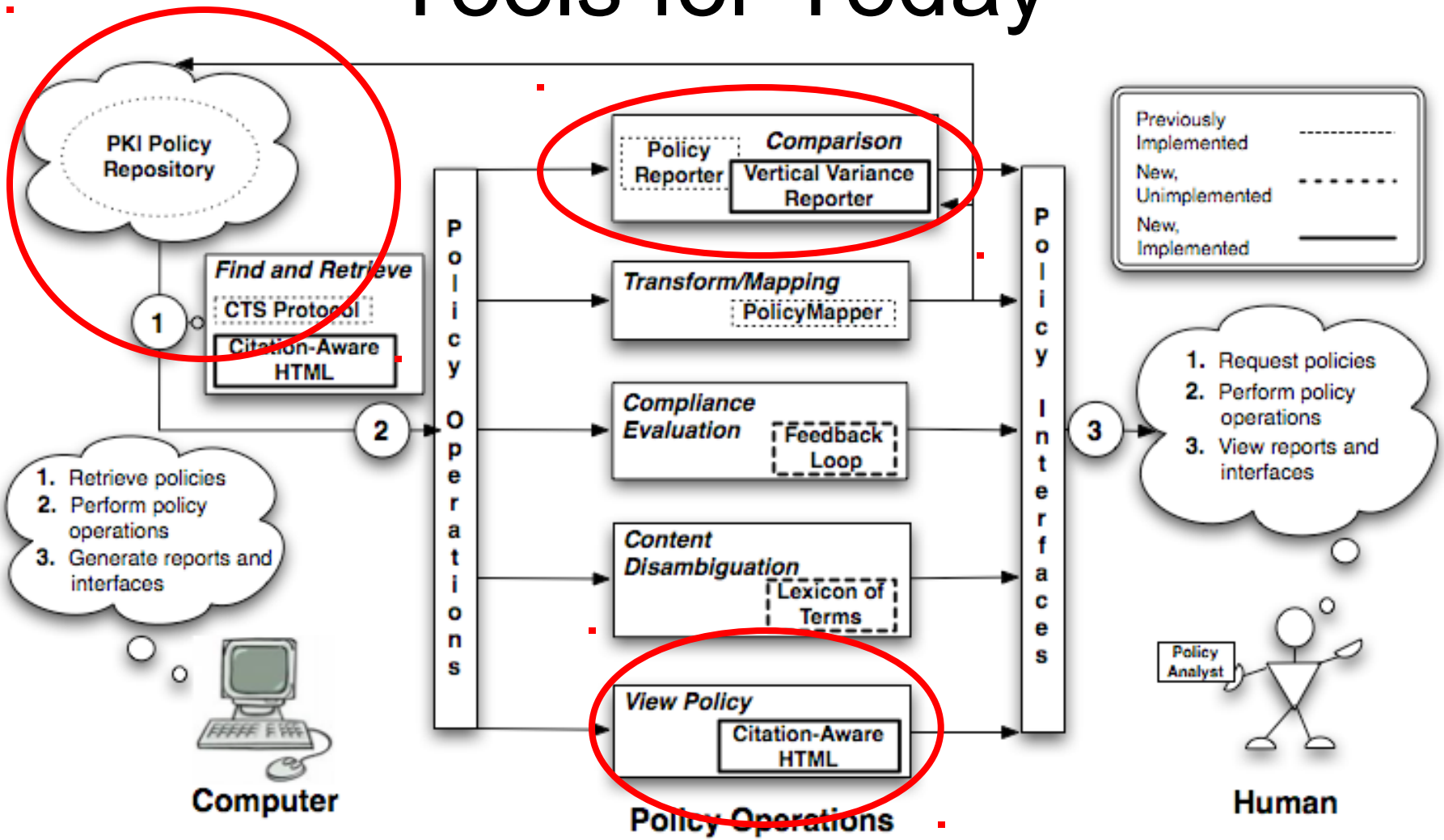
Formalizing Certificate Policy

We claim that computationally processing machine-actionable CP/CPSs is more *efficient* and *consistent*

- *Identification*
  - CTS-URNs
- *Representation*
  - TEI-XML encoding of reference structure (2527, 3647)

Semantics	CTS-URN	OID
All texts in the 'pkipolicy' namespace.	urn:cts:pki:pkipolicy	n/a
The ULAGrid CP (and CPS).	urn:cts:pki:pkipolicy.ulagrid	1.3.6.1.4.1.19286.2.2.2
A specific edition of the ULAGrid CP.	urn:cts:pki:pkipolicy.ulagrid.version1	1.3.6.1.4.1.19286.2.2.2.1.0.0
The ULAGrid CP's 'Technical Security Controls'	urn:cts:pki:pkipolicy.ulagrid.version1:6	1.3.6.1.4.1.19286.2.2.2.1.0.0.6
The ULAGrid's policy unit on Key pair generation	urn:cts:pki:pkipolicy.ulagrid.version1:6.1.1	1.3.6.1.4.1.19286.2.2.2.1.0.0.6.1.1

Tools for Today



Retrieval

PKI Policy Repository

- Last year only a handful of policies
- Feedback:
  - Needed more policies to be useful and prove viability
- Response:
  - Today ~200 IGTF CP/CPSs
  - Beta version on Google AppEngine (slow but stable)
- **Demo!** @ http://pkipolicy.appspot.com/

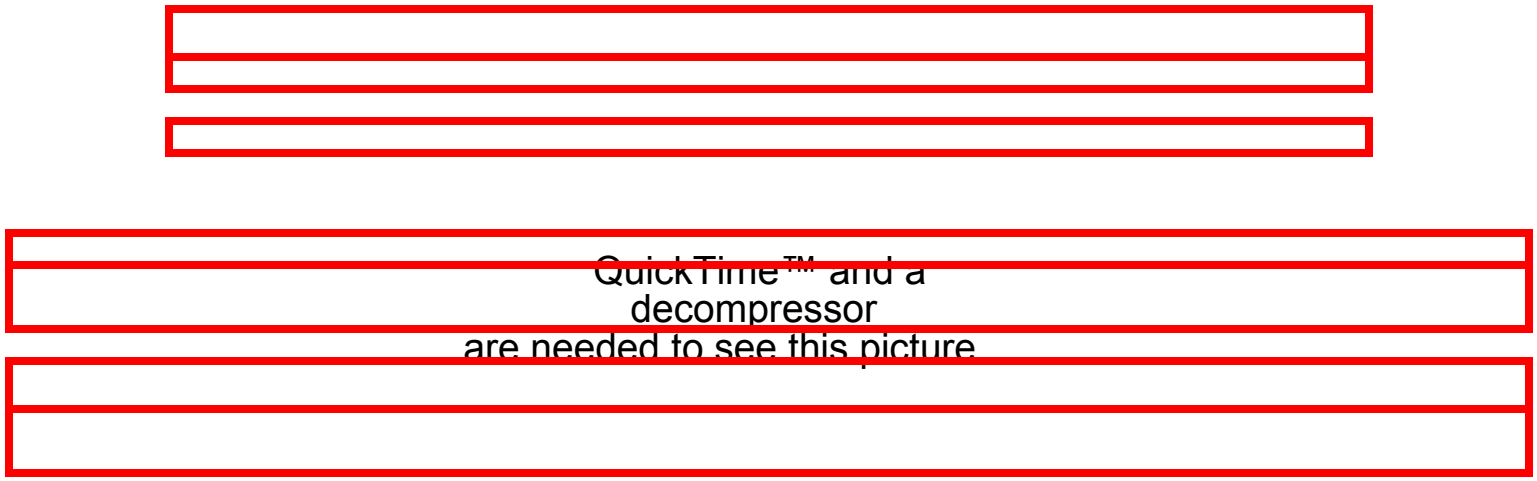
Comparison

PKI Policy Reporter

Provide more, higher-quality information for comparing CPs.

- Generate a report given a set of *policy sections and analyses.*
- **Demo!**
- Feedback:
  - Not all policies rigorously obey 2527/3647 format
  - Sections may *mean* different things across versions
- Response:
  - We created the Vertical Variance Reporter to see how policies *structurally* differ.

Vertical Variance Reporter



Viewing

Policy Reader

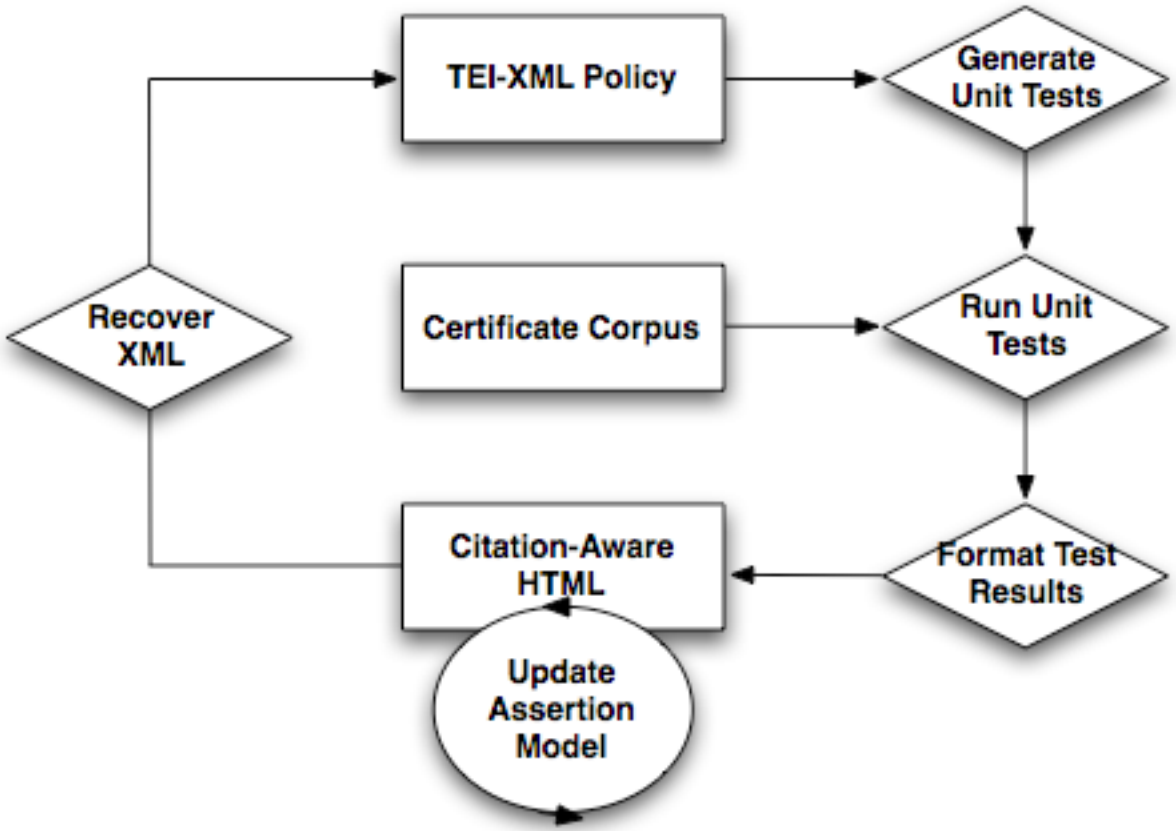
- Feedback:
  - PKI Policy Repository's interface not analyst-friendly.
- Response:
  - We developed the PolicyReader to transform TEI-XML policies into a more familiar format.
- **Demo!**

Future Work

Policy Searcher

- Feedback:
  - It would be nice to search a PKI Policy Repository
- Response:
  - We prototyped a PolicySearcher to search a repository.
- **Demo!**

Policy Compliance



Conclusions

- **We claim**
  - That a human-computer semantic gap arises from systems that primarily work on texts as files or a sequence of pages.
- **We bridge**
  - That gap with computational tools to process these reference structures and try to quantify variance.
- **We propose**
  - Additional tools to go beyond limitations of manual analyses.

Thank You

http://pkipolicy.appspot.com/
Gabriel.A.Weaver@dartmouth.edu

Other Slides

PKI Policy Mapper

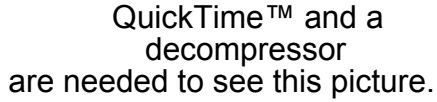
Transform the content of a CP/CPS in 2527 format into 3647 format.

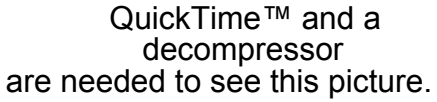
- Mapping a 2527 to 3647 requires 20% more effort than two 3647 CPs. Avg. mapping takes 80-120 hours in a bridge context.
- RFC 3647 defines tables, takes many hours
- Our mapping transforms 2527 to 3647 in seconds
- Demo
- We can flexibly configure the mapping
- Discovered errors in the transformation table (2.1 -> 2.6.4)

Experimental Evaluation

QuickTime™ and a
decompressor
are needed to see this picture.

QuickTime™ and a
decompressor
are needed to see this picture.

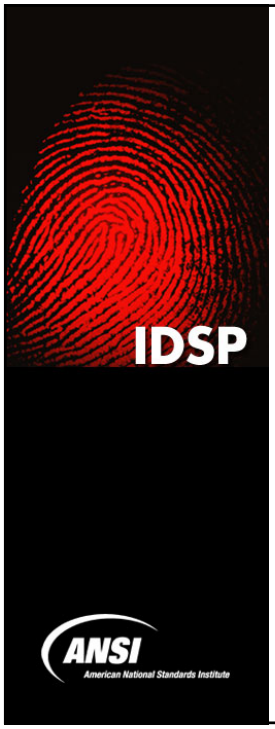

QuickTime™ and a
decompressor
are needed to see this picture.


QuickTime™ and a
decompressor
are needed to see this picture.

Additional Results

Previous Evaluation Results

- Policy Translation *(Policy Mapper)*:
  - Reduced time to perform task from a few days to a few seconds.
- Policy Comparison *(Policy Reporter)*
  - reduced part of the policy comparison process by up to 94%



Report of the IDSP Workshop on Identity Verification

Presented By:

Jim McCabe
Senior Director, IDSP
American National Standards Institute

IDtrust 2010
April 13, 2010

What is IDSP?

- ANSI is a not-for-profit membership organization that administers and coordinates the U.S. voluntary standards system
- Standards Panels provide a forum where subject matter experts from the private and public sectors work cooperatively to identify standards needed to address emerging national priorities
- Identity Theft Prevention and Identity Management Standards Panel (IDSP) is a cross-sector coordinating body whose objective is to facilitate the development, promulgation and use of standards and guidelines to combat ID theft and fraud
  - Identify existing standards, guidelines and best practices
  - Analyze gaps, need for new standards, leading to improvements
  - Make recommendations widely available to businesses, government, consumers

Workshop Participants

- North American Security Products Organization (NASPO)
- National Institute of Standards & Technology (NIST)
- Dept. of Homeland Security (DHS)
- General Services Administration (GSA)
- National Assn for Public Health Statistics & Information Systems (NAPHSIS)
- American Assn of Motor Vehicle Administrators (AAMVA)
- Colorado Div. of Motor Vehicles
- Coalition for a Secure Driver's License
- Social Security Administration
- Others

The Identity Verification (ID-V) Problem Starts at Issuance

- Fraudsters exploit circularity of agencies relying on but not authenticating primary USA "identity" documents issued by other agencies (birth certificates, Social Security numbers / cards, state-issued driver's licenses / ID cards)
- Intelligence Reform and Terrorism Prevention Act of 2004 (IRTPA) requires verification of identity prior to issuance of birth certificates
  - IRTPA regulations have not been released even in draft form
- REAL ID Act of 2005 requires verification of identity prior to issuance of driver's licenses / ID cards
  - Does not provide guidance on how to corroborate a claim of identity under different circumstances

Birth Certificates Especially Problematic

- Birth certificates considered an acceptable breeder document in many states but typically not verified by the issuing agency
- No biometric linking individual to birth record
- Within 57 jurisdictions, there are 6,400 registrars and 14,000 variations of certified birth certificates
- Person obtaining certified copy may not have legal rights to record—some states have "open" records policies
- Birth certificate may not be valid for person presenting it
- Information on birth certificate may not be factual
- Death records may be absent or delayed

Solutions in Progress

- National Assn for Public Health Statistics and Information Systems (NAPHSIS) developing security guidelines
  - Recommending states have "closed" record policies
  - Focusing on physical security of vital records offices
- NAPHSIS looking to expand Electronic Verification of Vital Event (EVVE) system currently only available in some states (Feb 2010 update: 19 states online w/EVVE; implementation in progress in 4 more states and NY City)
  - Provides government-to-government verification of birth and death information
  - Earlier IDSP report encouraged this expansion

Solutions in Progress (contd.)

- HHS CDC / National Center for Health Statistics charged with rulemaking under section 7211 of IRTPA
- Will regulate how states issue vital records but states may decide not to comply
- Will reduce number of birth certificate forms to about 57
- Earlier IDSP report noted that rulemaking has been delayed and recommended that these standards are needed now

IDSP

Workshop Recommendation

- Issuers of primary USA "identity" documents need a process by which they can achieve a level of assurance whether to accept or reject a person's claim of identity
  - One or more practical methods to verify identity with very high confidence, high confidence, some confidence or low/no confidence
- Guidelines on identity verification should be developed with a view toward eventual development of an American National Standard

IDSP

Envisioned Benefits

- Enhanced security / credibility of identity vetting processes and foundational identity documents
- Enhanced security / credibility of credentials issued downstream based on the presentation of these foundational documents as evidence of identity
  - Other government credentials (FIPS 201 PIV cards, U.S. passports, Medicare / Medicaid cards)
  - Commercial credentials (credit / charge cards)
- Will help to reduce identity theft
- Will help to protect Americans from terrorist attacks
- And more . . .

IDSP

Project Phases

- Phase 1 – Concept Formulation – 8 months
  - How to build certainty in a claimed identity
  - Criteria for the acceptance/rejection of a claim
  - Methods for the detection of fraud
  - Deliver draft Guideline
- Phase 2 – Testing – 4 months
  - State vital record offices (birth certificate issuance)
  - State DMVs (DL & ID card issuance)
  - Release of Guideline
- Phase 3 – Standardization – 8-12 months
  - ANSI/NASPO-IDV-2010 Methods for the Verification of Personal Identity

IDSP

Timeline

- Initial IDSP workshop meetings July – Sept 2008
- Project plan developed / team formed led by NASPO
- Concept formulation meetings Oct, Dec 2008, Feb 2009
- IDSP workshop report and NASPO ID-V project then proceeded on parallel tracks. Both released Oct 2009.
- March 29, 2010 – NASPO formally announces its intention to develop an American National Standard
  - Project Initiation Notification System (PINS) 30 day announcement for public comment in April 9 edition of ANSI *Standards Action* www.ansi.org/standardsaction

IDSP

Conceptual Approach for Identity Verification Guidelines

Presented By:

Brian Zimmer
Panel Member, IDSP
President, Coalition for a Secure Driver's License

IDtrust 2010
April 13, 2010

The Chosen Concept for Verified Identity

- An aggregation of evidence / adjudication process
  - Accreditation of Identity Adjudicators
  - An "Identity Resume"
  - An in-person meeting & biometric capture
  - Verification of key items of corroborative evidence
  - Use of acceptance/rejection criteria
  - A two step exceptions process
  - Binding of the person to the verified identity
  - Possible issuance of a ID-V token or certificate
  - Detailed procedures to be followed for the whole adjudication process

IDSP

Key Concepts

- Selection and training of identity adjudicators to manage, administer and effect the process (Background check)
- Use of an identity resume to define the identity, gather information, detect fraud and reduce uncertainty
- An in-person meeting to provide opportunities for candidate / adjudicator interaction, observation and biometric capture
- Preparation and implementation of a personalized plan for verification of evidence
- Procedures for verification of the origins and continuous use of identity for both USA and foreign born persons

IDSP

Key Concepts (cont.)

- Use of a contra indications format for the documentation and presentation of raw results
- Procedures for evaluation and aggregation of evidence and mitigation of significant contra indications
- Procedures for identification of critical combinations of findings to enable fraud detection
- Criteria (thresholds) for acceptance or rejection of the claimed identity optimized to the needs of the relying party
- A two step process to deal with problem cases
- Biometric binding of the person to the verified identity followed by registration of results

Deliverable Content

Part 1 – Resume Introduction

Part 2 – Identity Resume RFI

Part 3 – Resume Preparation Instructions

Part 4 – Adjudication Process Description

Part 5 – Adjudicator Responsibilities

Part 6 – Adjudication Procedures

The Identity Resume

- This is a request for information that will enable:
  a) definition of a person's identity
  b) corroborative evidence to be collected
  c) uncertainty inherent in corroborative evidence to be reduced
  d) symptoms of identity fraud to be detected
  - Information items c) & d) in the Resume are based on:
  - for c) – an analysis of the uncertainty or risk associated with each item of corroborative evidence and how to reduce it
  - for d) – an analysis of behavior expected by each type of identity fraud leading to the inclusion of "imposter traps"

IDSP

Resume Content

1. Your origins
2. Your early years
3. Your family
4. Any name changes?
5. Your education & training
6. Places you have lived
7. Your licenses
8. Your citizenship
9. Your work history
10. Your memberships
11. Your ownerships
12. Unique events/experiences
13. Your special skills
14. Personal information
15. Your ID Documents
16. Additional corrob. evidence

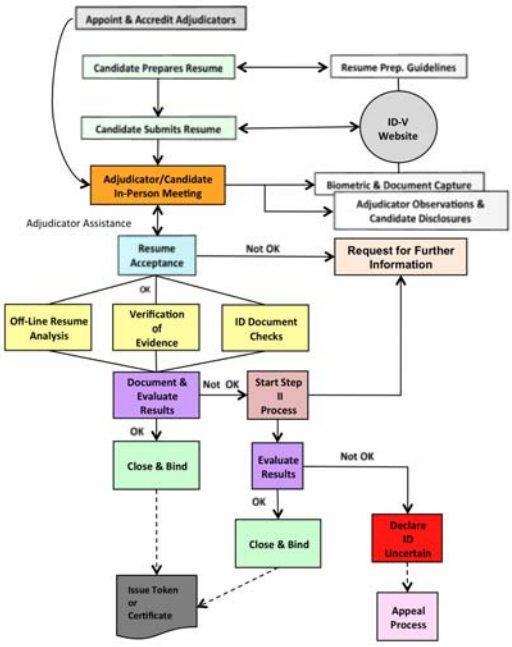
IDSP

Adjudicator Responsibilities

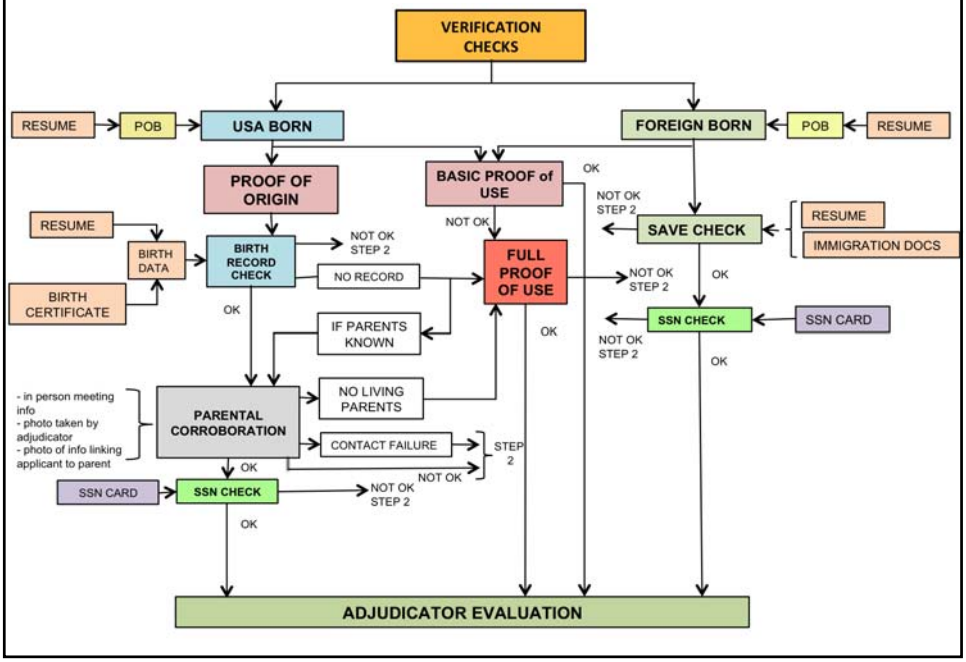
- Prior to the In-Person Meeting
- During the In-Person Meeting
- Following the In-Person Meeting
  - Analysis of the Resume
  - Document Authentication
  - Verification of Corroborative Evidence
- Documentation of Findings & Contra Indications
- Assessment of Impacts on Proof of Origin & Use
- Evaluation, Decision & Action

IDSP

ID-V Process Diagram



Verification of Corroborative Evidence



Contra Indications – The Results Format

Source of Findings		No.	ID-V Contra Indication FINDINGS	Applicable Findings	Proof of Origin IMPACT	Proof of Use IMPACT	Suggested ACTION (e.g. findings to compare and combine)
In-Person Meeting	General Observations	1	Applicant Incoherent				
		2	Inadequate Resume				
		3	Heavy Assistance Required				
		4	Mother Tongue Mismatch				
		5	No Additional Disclosures				
		6	No SSN				
		7	USA Born Problem				
		8	Origins Unknown				
		9	Poor or No Knowledge of Places Lived				
		10	Poor or No Knowledge of Early Years				
		11	25 Year Use Problem				
		12	No Contact Disclosures				
		13	Uncooperative/Hostile Attitude				
	Identity Document	14	No Documents Tendered				
		15	No Picture ID				

IDSP

IDtrust 2010

Slide 22

Proof of Origin Flags

Checks	Results	Red Flag	Yellow Flag	Green Flag	Critical Combination
Birth Certificate	Available				
	Not Available				Check Other ID Docs
	Problem Mitigated				
Birth Record	Full Match				
	Partial Match				
	Problem Mitigated				
	No Matching Record				
Parental (or Equivalent) Corroboration	Strong Corroboration				
	Moderate Corroboration				
	Problem Mitigated				
	No Corroboration				
SSN	Full Match				
	No Matching Record				Check Other ID Docs
	Problem Mitigated				

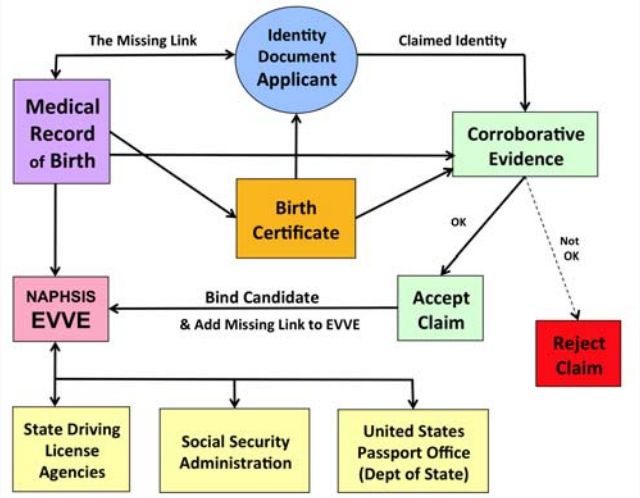
IDSP

Longer Term Goal: PIV Synchronization

PIV Assurance Levels		1	2	3	4
Proof Required	Corroborative Evidence	Self Assertion	Higher Risk	Moderate Risk	Primary ID Document Grade
Origins of the Person	Birth Certificate		Non existence tolerated	Non existence must be fully mitigated. Possession of other ID Docs must be fully mitigated	Non existence must be fully mitigated by hard evidence. Possession of other ID Docs must be fully mitigated
	Birth Record		Non existence must be fully mitigated	Non existence must be fully mitigated by hard evidence	Non existence must be fully mitigated by hard evidence
	Parental Corroboration		Trusted corroborator evidence acceptable if 25 years of use is proven	Mandatory	Mandatory with uncertainty reduction
	SSN		Mandatory	Mandatory	Mandatory
	Problem Mitigation Evidence		All critical combinations, red flags and high impact contra indications must be mitigated	All high impact contra indications, red flag problems and critical combinations must be fully mitigated	All problems must be fully mitigated by hard evidence
Continuous Use of Original Identity	Taken from 10 life history categories		10 years of continuous use with evidence selected from any of 10 life history categories. All changes of identity must be fully explained	15 years of continuous use with evidence selected from 5 out of 10 life history categories. All changes of identity must be fully explained	25 years of continuous use with evidence selected from 5 out of 10 life history categories. All changes of identity must be fully explained

IDSP

Custodian of the Results ?



IDSP

ANSI American National Standards Institute

IDSP

Thank You.

To obtain the IDSP workshop report
http://webstore.ansi.org/identitytheft

To obtain a summary of the NASPO ID-V project
http://www.naspo.info/PDFiles/ID-V_Project.pdf

For further information, contact
Graham Whitehead, NASPO, gdw@naspo.info

Jim McCabe
jmccabe@ansi.org
212.642.8921
www.ansi.org/idsp

Brian Zimmer
BrianZimmer@IDSecurityNow.org
202-312-1540
www.secure-license.org

Four Bridges Forum: How Federated Identity Trust Hubs Improve Identity Management

The Federal PKI

Tim Pinegar
Federal PKI Architecture
Protiviti Government Services
tim.pinegar@pgs.protiviti.com

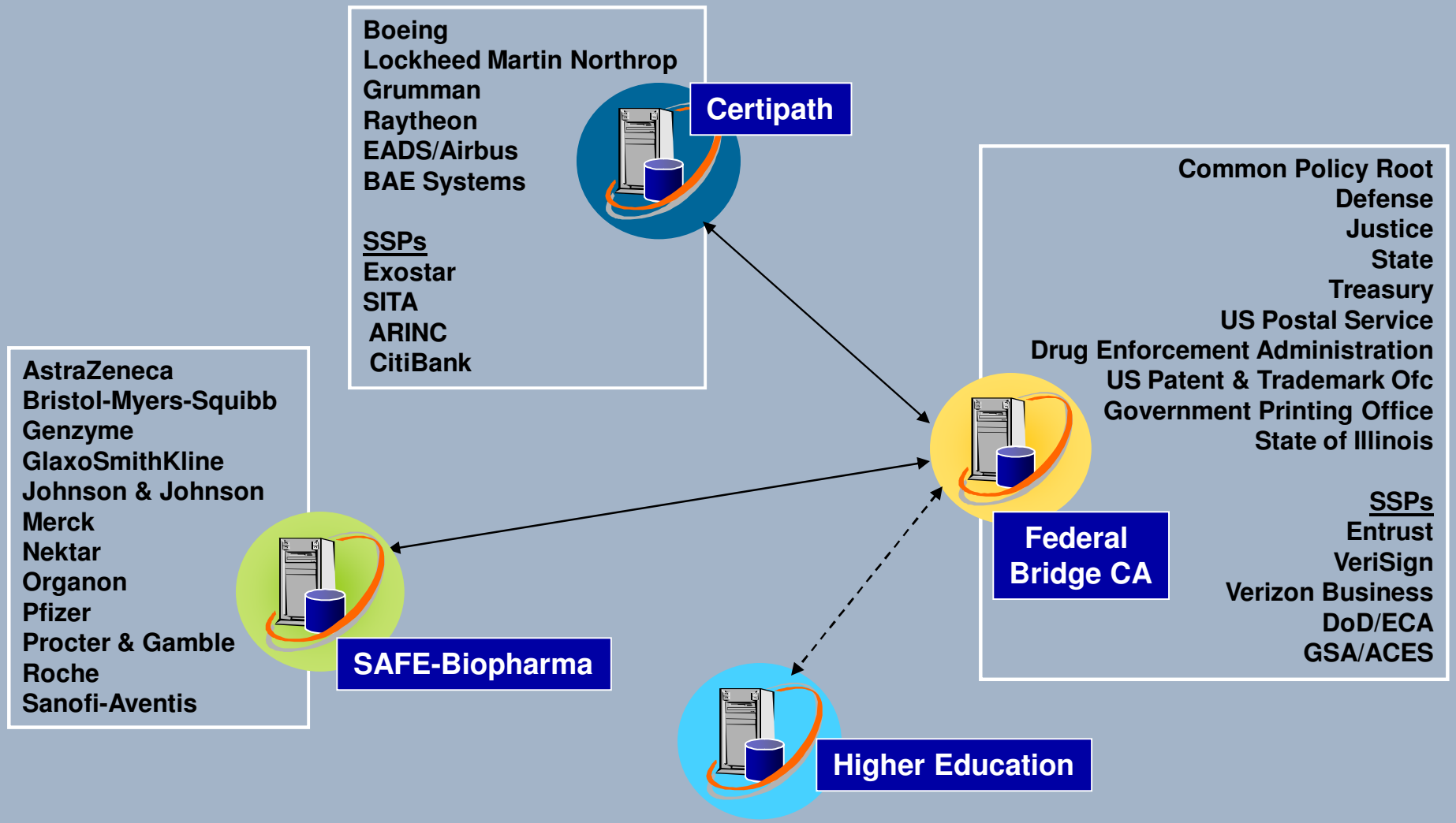
What is the 4BF?

- A consortium of public key infrastructure (PKI) bridges each serving a major community of interest;
- Leveraging government and non-government federated identities;
- Based on a common foundation of trust;
- Laying the groundwork for a global trust network.

Who is the 4BF?

- The Federal PKI Architecture (formerly the Federal Bridge Certificate Authority or FBCA), established to enable trusted transactions within the government and between government and its industry partners.
- SAFE- BioPharma Association, founded by global pharmaceutical organizations to develop and manage digital identity and signature standards for the pharmaceutical and healthcare industries.
- CertiPath, establishing interoperable trusted identities for collaboration within the Aerospace and Defense industry via a standards based PKI bridge.
- The Higher Education Bridge Certificate Authority (HEBCA), developed to facilitate trusted electronic communications within and between institutions of higher education as well as with federal and state governments.

The Four Bridges



The Federal Bridge

- The FBCA is the identity trust hub that enables peer-to-peer transactions between its member organizations, both Federal and non-Federal;
- Source of interoperability for ALL Federal Agency HSPD-12 credentials (5.09 million and counting as of 12/2009);
- Enables Agencies to validate each other's PIV cards for physical access;
- Validate desktop and network logins;
- Support high assurance authentication to Agency Level 3 & 4 applications using government and private sector credentials

4BF Timeline

- 2003 – NIH and Higher Ed demonstrate Bridge-to-Bridge interoperability.
- 2004 – Aerospace Industry starts Certipath Bridge.
- 2004 – Pharmaceutical Industry announces SAFE Bridge.
- 2006 - CertiPath cross-certifies with the FPKI.
- 2008 - BioPharma cross-certifies with the FPKI.
- 2008 – Inaugural meeting of representatives of the four bridges.
- 2008 - 4BF Audit Working Group is formed to define a standard baseline for PKI audit comprehensiveness and quality that incorporates international standards.
- 2008 – 4BF Agreement to Cooperate is signed.
- 2009 – 4BF launches formal outreach campaign.

Why was the 4BF Formed?

- To address and resolve common issues affecting PKI-based identity federations;
- To stimulate greater use of high assurance electronic identity credentials by raising awareness of the benefits to relying party applications;
- To target outreach to government program managers, application owners, and industry partners who can reap immediate benefits from use of PKI bridges; and
- To stimulate global interoperability via the 4BF trust infrastructure.

Benefits of the 4BF?

- Leverage PIV certificates beyond internal agency systems to improve the ROI of PIV system infrastructure.
- Source of interoperability with a business partners in the aerospace, defense and bio-pharmaceutical communities
- Trust of 4BF identity credentials provides “real time” scalability;
- Facilitating identity portability

For Further Information

- Contacts:
  - Judith.Spencer@gsa.gov
  - Tim.Pinegar@pgs.protiviti.com
  - Mollie Shields-Uehling, (mollie@safe-biopharma.org)
  - Scott.Rea@Dartmouth.edu
  - Jeff.Nigriny@certipath.com
- Websites:
  - http://www.safe-biopharma.org/index.htm
  - http://www.certipath.com
  - http://www.idmanagement.gov


4 B F

The 4BF
The Four Bridges Forum

The SAFE-BioPharma Digital Identity and Signature Standard

Mollie Shields Uehling
CEO
SAFE-BioPharma Association

Moving into the Digital Age: BioPharma and Its Partners

- Revolution in life sciences and medical research
- Cost and complexity has created crisis in R&D productivity
- Need for rapid, close collaboration between pharma, healthcare providers, government agencies and research institutions
- FDA and EMEA moving to fully electronic submission, review and response
- Healthcare mandate for eMRs for every American by 2014 presents wealth of opportunity for information for research and clinical decision-making
- Fundamental to interoperability in sensitive electronic exchanges of information are trusted identities and legal signatures.

SAFE-BioPharma Association

- Strategic initiative started by biopharmaceutical industry
  - Member-governed non-profit collaboration incorporated May 2005
  - Trusted identity and non-repudiable digital signature
  - Single interoperable digital identity across industry
  - Technology and vendor neutral
  - Interoperable with Federal agencies
  - Based on leading government technical and identity proofing standards
  - Wrapped in a legal, governance and risk mitigation model
  - Recognized by world's leading regulatory authorities
- To facilitate the transformation of the industry to fully electronic business and regulatory processes

SAFE-BioPharma 2005-2010

- Regulatory engagement and recognition – US, European Union, Japan
- Improving usability
  - Pilots, early adopters
  - Resulted in expansion of the standard
  - Improvements in identity proofing process and digital signing options
- Building the interoperable network:
  - Issuers, digital signing, and business applications
  - Cross-certification with FBCA
  - EU qualified certificates; Safe Harbor certification
- Supporting use
  - First, ELNs (basic laboratory research)
  - Then digitally signed regulatory submissions
  - Now workflow between several/many partners for auth & signing in federated approach

The SAFE-BioPharma Framework

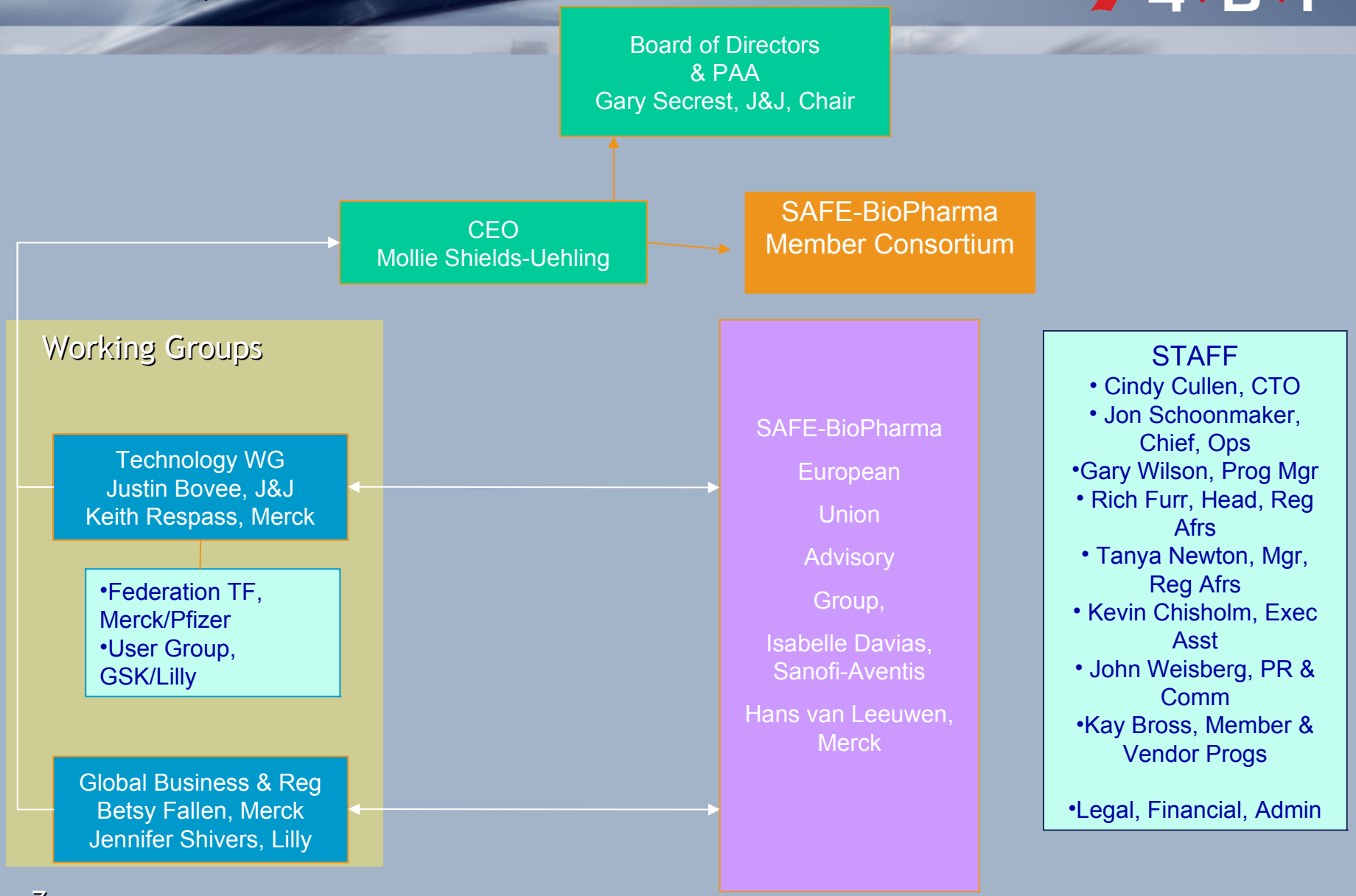
Legal, governance, risk mitigation – contract based

Existing Technical and Identity Standards – NIST,OMB, Federal PKI

- *Identity verification*
- *Manage identity life cycle*
- *Comply with referenced standards*
- *Security, audit & control requirements*
- *Certification*

- *Accept digitally signed transactions*
- *Agree to limited liability caps*
- *Agree to dispute resolution*
- *Agree to identity assurance*
- *Agree to self-audit & meet SAFE requirements*

A Non-Profit, Member-Driven Standards Association



SAFE-BioPharma Association – Non-Profit Standards Collaboration

Standards	Standards-Related Services	Collaborative Association
Standard Development & Maintenance **Certification:** - Products - Issuers **Standards engagement: HL7, HITSP, CDISC, IHE, Kantara** **Working Groups** –Technical –Federation –Users Group –Global Business & Reg –Implementation –SAFE EU Advisory Council **Regulatory alignment:** –FDA; EMEA; NCAs, MHLW	**Manage member –funded shared infrastructure** **Operation of SAFE bridge** **Cross-cert with FBCA** **Vendor partner program** **Implementation tools** **Incubating Innovation** **Credentials Issuance Model** **Antecedent Data ID Proofing** **EU qualified digital identities** **Zero footprint token** **Hosted digital signing**	**Stakeholder outreach** **Education & advocacy** **Policy engagement** **Industry awareness & engagement** **4BF – network of trusted bridges** **Information/Best Practices Forum** **Media: local, national, trade, international**

Options for Flexible Use

- Two levels of trust:
  - Basic Assurance for authentication
  - Medium Assurance for trusted identity uniquely linked to digital signature and EU-qualified
- Three digital signing technologies:
  - Software
  - Hardware (zero footprint now undergoing FIPS certification)
  - Roaming
- Three identity-proofing options
  - Antecedent – enterprise and on-line
  - Trusted agent
  - Notary – including office/home notary services

Member Public Key Infrastructure Options

- Internal infrastructure
  - Cross certified with SAFE-BioPharma Bridge
  - BMS, J&J
- Outsourced infrastructure
  - Cross-certified with SAFE-BioPharma Bridge:
    - Chosen Security
    - Citibank
    - Entrust
    - IdenTrust
    - TransSped
- SAFE-BioPharma tiered services infrastructure:
  - External partners
  - Regulatory uses
  - Healthcare providers
  - Members

Assurance and Identity Proofing Services*

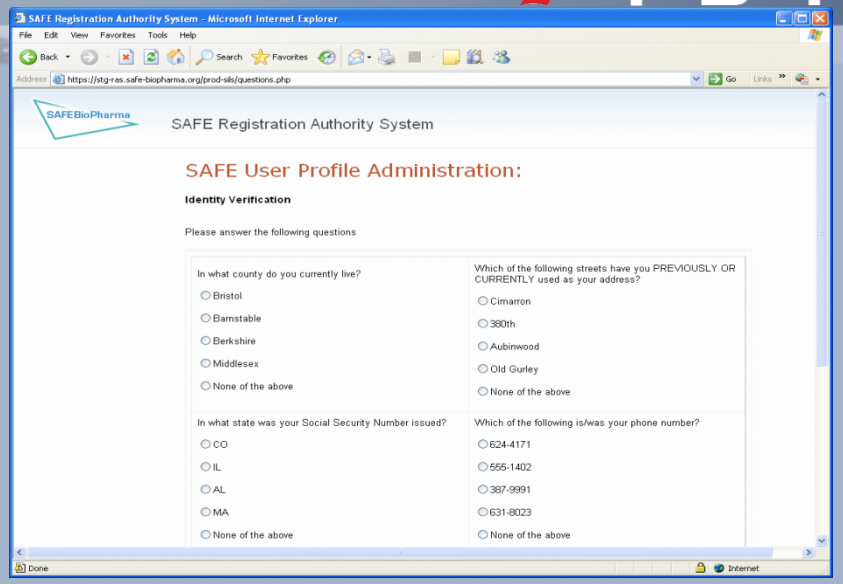
Credential Type	Assurance Level ====> Credential Usage	ID Proofing Method				
		Basic *Basic only*	Enterprise Antecedent *Basic & Medium*	3rd Party Antecedent *Medium*	Trusted Agent *Medium*	Notary *Medium*
Basic Software	*Authentication*	✓	✓			
Medium Software	*Auth &/or Signing*		✓	✓	✓	✓
Hardware Token	*Auth &/or Signing*		✓	✓	✓	✓
Roaming Digital ID	*Auth &/or Signing*		✓	✓	✓	✓
One Time Password (OTP)	*Authentication*					

★ 3rd-Party Antecedent currently available in USA

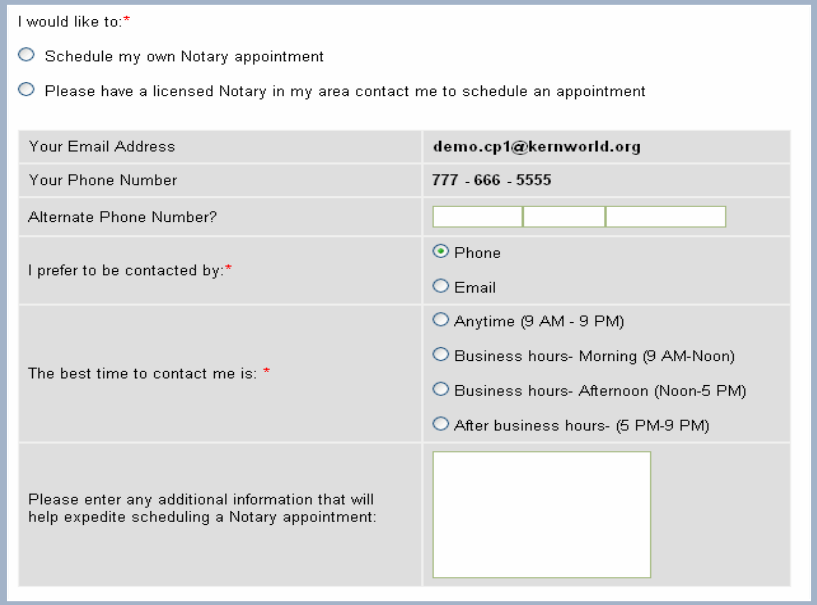
*Provided through outsourced services supplier.

On-Line Antecedent Process

- ID Vetting Successful:
  - Applicant Passes 3rd Party Antecedent identity proofing
  - Moved to RA queue for processing and Certificate Issuance steps.
  - It's a matter of minutes end-to-end.



- ID Vetting Not Successful:
  - Unable to verify identity via 3rd Party Antecedent
  - Process reverts to Notary Process with two service options:
    - User locates notary
    - RAS/NNA will have a local notary contact the Applicant directly



SAFE-BioPharma and Regulators

- European Medicines Agency (EMA) and FDA are on paths to requiring fully electronic submissions within the next few years
- FDA and EMA helped write SAFE-BioPharma standard; engaged since inception
  - FDA has received 10,000s of SAFE-BioPharma submissions since 9/06
  - EMA eCTD pilot successfully completed; EMA ESG to go live this year
- Japan pilot underway – exchanging business, regulatory and clinical documents between pharmas, hospitals, and regulatory agencies

Examples of How SAFE-BioPharma Is Being Used

Use Case	Company
ELNs – basic research	Abbott (including China), BMS, GSK, Pfizer, SA
Contracts, SOWs	J&J, Premier, Oxford, MWB Consulting
Physician Signatures	SNAP Diagnostics
Purchasing	Premier
Alliance management	BMS
External Partner Authentication	BMS, GSK
Regulatory Submissions	AZ, BMS, GSK, SA, Eli Lilly
Document management system	McDougall Scientific
Collaborative research partners	BMS
Paperless business/regulatory environment	Amarin, MWB Consulting. SAFE-BioPharma Assn

Pfizer eLabNotebooks

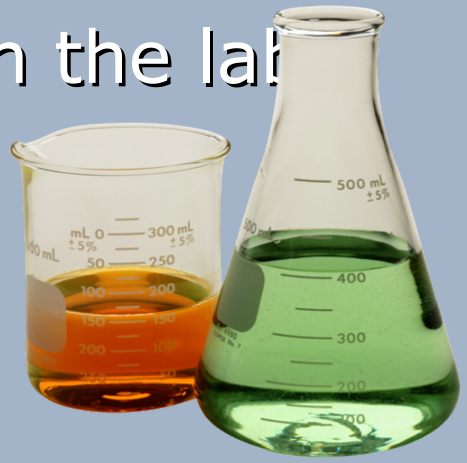
Company Profile:

- Largest *research-based* pharmaceutical
- Global research organizations
- Using paper laboratory notebooks requiring scientists signatures on each experiment!
- Replaced with electronic notebooks and digital signatures

Pfizer ELN Results – Over 1 million digital signatures

Results:

- Less time on paperwork, more in the lab!
  - > 3300 researchers in 280 departments in 20 countries;
  - > 550,000 documents signed
  - >1,000,000 digital signatures!
- 3.3 million pages *not* printed!
  - >16 tons of paper saved
- Better patent defense
  - Signed, time-stamped in timely manner
- Better compliance with internal regulations
- Easier access to research
  - Electronic search of records
- Faster research cycles
  - More time in lab, less on paperwork; No more delays to collect witness signatures



SNAP Diagnostics

Company Profile:

- Leader in diagnostic technology for detection of sleep apnea and analysis of snoring problems
- Provides physicians in the U.S., EU, and Latin America with proprietary diagnostic equipment used in home settings

Scope:

- Records of at-home tests analyzed by company physicians who advise referring physicians re therapeutic approach
- SNAP physicians digitally sign diagnoses and send to personal physician

Results:

- Eliminated paper in day-to-day reviews of diagnostic information
- Eliminated costs associated with handling, signing, shipping, storing and accessing paper

Premier Purchasing

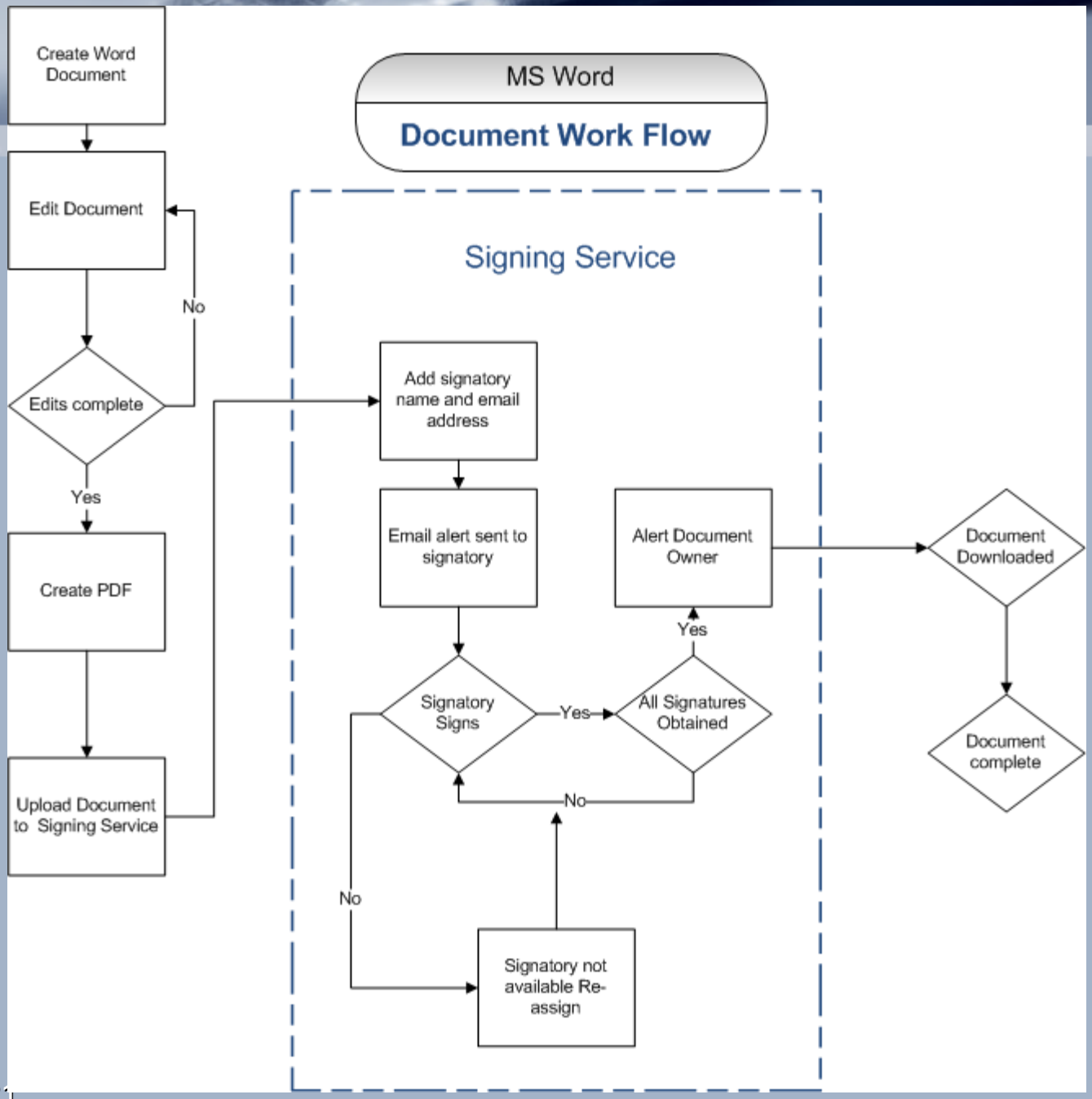
- Company profile
  - Largest Group Purchasing Organization (GPO) in U.S.
  - Owned by non-profit hospitals
  - Serves 2,000 U.S. hospitals and 53,000-plus other healthcare sites
  - Buys from ~700 suppliers
  - http://www.premierinc.com/
- Scope:
  - Eliminate overnight shipping, fax and related workflows for contract origination and amendments
  - Provide SAFE-BioPharma credentials to Premier Sourcing/Procurement employees and their supplier colleagues for signing new and amended supplier contracts
  - eContracting process ~700 companies and thousands of contracts and/or amendments
- Future:
  - Digitally sign and submit required reports to CMS

National Cancer Institute-Bristol-Myers Squibb: The Business Issue

- NCI – pre-imminent cancer research institution
- Collaborates with pharma, biotechs, many research institutions and individual researchers
- Lots of contracts, clinical documents, amendments, signatures
- Bristol-Myers Squibb (BMS) conducts collaborative research with NCI's Clinical Trials Evaluation Program (CTEP) and its many collaborative research partners (cooperating groups)
  - Clinical trial agreements
  - Clinical trial documents
  - Clinical materials orders and supply
- Leverage PIV (Federal credentials) , BMS cross-certified SAFE-compliant credentials and SAFE-BioPharma credentials for authentication and digital signing

The NCI-BMS Project

- Credentials – Medium Assurance
  - BMS
    - BMS SAFE-BioPharma Cross Certified credentials
    - SAFE-BioPharma Credentials
  - NCI
    - PIV cards
  - Cooperative Groups – research organizations – academic, CROs, etc.
    - SAFE-BioPharma Credentials
- Signature Work Flow
  - SAFE-BioPharma Digital Signing Service Pilot – MySignatureBook/MyOneLog-on
  - A “cloud” service


MS Word
Document Work Flow
Signing Service
Create Word Document
Edit Document
Edits complete
No
Yes
Create PDF
Upload Document to Signing Service
Add signatory name and email address
Email alert sent to signatory
Signatory Signs
Yes
No
Signatory not available Re-assign
No
All Signatures Obtained
Yes
Alert Document Owner
Document Downloaded
Document complete


Digital
Signature
Service
SAFE-BioPharma
Cooperative
Group
NATIONAL
CANCER
INSTITUTE
Bristol-Myers Squibb


Digital
Signature
Service
SAFE-BioPharma
Cooperative
Group
NATIONAL
CANCER
INSTITUTE
Bristol-Myers Squibb

Leveraging the Value of the 4BF

- Very challenging economic environment
- Investment flowing to areas that will improve productivity and lower FTE
- Opportunity to leverage network of interoperable credentials and
- To create network of trusted partners
- Move government-business and B2B processes into the cloud
- Bring value to government program managers and private sector business process owners by improving business process efficiencies

Value of SAFE-BioPharma Digital Identity and Signature in the Partnered Digital World

- Established, tested standard meeting needs of governments, regulators, industry
- Ability to identity proof and issue certificate in ~15 minutes (including medical license)
- Contract-based system tailored to pharma/healthcare needs (HIPAA compliance; risk mitigation)
- Secure
- Legal enforceability
- Regulatory recognition and acceptance (US, EU, Japan)
- Global standard and set of services
- Links Federal agencies to pharma and healthcare providers
- Provides interoperability

Interoperable standard that facilitates transition to fully electronic business and regulatory processes

- *Please visit the SAFE-BioPharma website*: http://safe-biopharma.org/
- *Please visit the 4BF website:* http://www.the4bf.com/
- *Pfizer's Implementation of SAFE-BioPharma Digital Signatures in ELNs:* http://www.safe-biopharma.org/images/stories/pfizer%20white%20paper_v1.pdf
- *AstraZeneca's Implementation of SAFE-BioPharma for FDA Submissions:* *http://www.safe-biopharma.org/images/stories/az_safe_final.pdf*
- *Learn more about the SAFE-BioPharma Implementation Toolkit:* http://safe-biopharma.org/index.php?option=com_content&task=view&id=254&Itemid=422
- *Watch the SAFE-BioPharma introductory video:* http://www.phillipsvideopost.com/safe
- *Contact us for more information:*

Mollie Shields Uehling
CEO
mollie@safe-biopharma.org
(201) 292-1861
(201) 925-2173 (cell)

Kay Bross, Director
Member/Vendor Progs
kbross@
(513) 489-3840 (o)
(513) 673-2344 (c)

Jon Schoonmaker
Chief of Operations &
Technical Program
(301) 610-6060
jschoonmaker@safe-biopharma.org

Cindy Cullen
CTO
cindy.cullen@bms.com
(609) 818 4152

Kevin Chisholm, Admin.
Kevin.Chisholm@SAFE-BioPHarma.org
(201) 292-1860

Rich Furr
Head, Reg. Afrs.
RFurr@SAFE-BioPharma.org
(610) 252-5922

Gary Wilson
Prog. Mgr
(781) 962-3172
Gwilson@safe-biopharma.org

Tanya Newton
Manager, Reg Afrs
(908) 213-1069
tanya.newton@safe-biopharma.org


4 B F

The 4BF

The Four Bridges Forum

HEBCA - Higher Education Bridge Certificate Authority

Authentication

- Authentication is the process of obtaining an identification credential (e.g. username/password) from a user and validating those credentials against some authority.
  - If the credentials are valid, the entity that submitted the credentials is considered an authenticated identity.
- Authentication relies on two main elements:
  - A credential that is bound to an identity
  - The ability to verify the credential

Authentication Factors

- Three different Factors of Authentication:
  - Something you know
    - e.g. password, secret, URI, graphic
  - Something you have
    - e.g. key, token, smartcard, badge
  - Something you are
    - e.g. fingerprint, iris scan, face scan, signature

Authentication Factors

- Single Factor of Authentication is most common
  - Passwords (something you know) are the most common single factor
- At least Two Factor Authentication is recommended for securing important assets
  - e.g. ATM card + PIN (have + know)
- 2 x Single Factor Authentication ≠ Two Factor Authentication
  - e.g. Password + Graphic is NOT equivalent to Smartcard + PIN (although it may be better than a single instance of One Factor Authentication)

Password Authentication

- General issues with Authentication using Password technology
  - Passwords easily shared with others (in violation of access policy)
  - Easily captured over a network if no encrypted channel used
  - Vulnerable to dictionary attacks even if encrypted channels are used
  - Weak passwords can be guessed or brute forced offline
  - Vulnerable to keyboard sniffing/logging attacks on public or compromised systems
  - Cannot provide non-repudiation since they generally require that the user be enrolled at the service provider, and so the service provider also knows the user's password
  - Vulnerable to Social Engineering attacks
  - Single factor of Authentication only

Password Authentication

- Specific issues with Authentication using Password technology
  - Too many passwords to remember if requiring a different one for each application
    - Leads to users writing them down and not storing them securely
    - Leads to use of insecure or weak passwords (more secure ones are generally harder to remember)
    - Leads to higher helpdesk costs due to resetting of forgotten passwords.
    - Leads to re-use of passwords outside institutions' domain where protection mechanisms may be much lower

Password Authentication

- Specific issues with Authentication using Password technology
  - Potential single point of failure for multiple applications if same password used
    - Strong passwords not consistently supported in all applications
    - Weak passwords leads to widespread compromises
    - Passwords not consistently protected for all applications
    - Password expiration not synchronized across applications
    - Limited character set for input
    - No control over use of passwords outside organization's domain
    - Offline attacks against passwords may be possible

The PKI Solution

- Solution to Password vulnerabilities - Public Key Infrastructure (PKI)
  - PKI consists of a key pair – 1 public, stored in a certificate, 1 private, stored in a protected file or smartcard
  - Allows exchange of session secrets in a protected (encrypted) manner without disclosing private key
  - PKI lets users authenticate without giving their passwords away to the service that needs to authenticate them
    - Dartmouth's own published password-hunting experiences shows that users happily type their user ID and password into any reasonable-looking web site
    - PKI can be a very effective measure against phishing

PKI Solution

- Solution to Password vulnerabilities - Public Key Infrastructure (PKI)
  - PKI lets users directly authenticate across domains
    - Researchers can collaborate more easily
    - Students can easily access materials from other institutions providing broader educational opportunities
  - PKI allows decentralized handling of authorization
    - Students on a project can get access to a web site or some other resource because Prof Smith delegated it to them
    - PKI simplifies this process – no need for a centralized bureaucracy, lowers overheads associated with research
  - Private key is never sent across the wire so cannot be compromised by sniffing
  - Not vulnerable to dictionary attacks
  - Brute force is not practical for given key lengths
  - Facilitates encryption of sensitive data to protect it even if a data stream or source is captured by a malicious entity

PKI Solution

- Solution to Password vulnerabilities - Public Key Infrastructure (PKI)
  - 1024-bit keys are better than 128 character passwords (they are not subject to a limited character input set)
    - This is far stronger than just about any password based authentication system
    - As one researcher said recently “the Sun will burn out before we break these”

Quote from Prof Smith: *“In the long run: user authentication and authorization in the broader information infrastructure is a widely recognized grand challenge. The best bet will likely be some combination of PKI and user tokens.”*

  - Failing to look ahead in our IT choices means failing in our research and educational mission.

Additional PKI Benefits

- Additional drivers for PKI in Higher Education (besides stronger authentication):
  - Better protection of digital assets from disclosure, theft, tampering, and destruction
  - More efficient workflow in distributed environments
  - Greater ability to collaborate and reliably communicate with colleagues and peers
  - Greater access (and more efficient access) to external resources
  - Facilitation of research funding opportunities
  - Compliance

Additional PKI Benefits

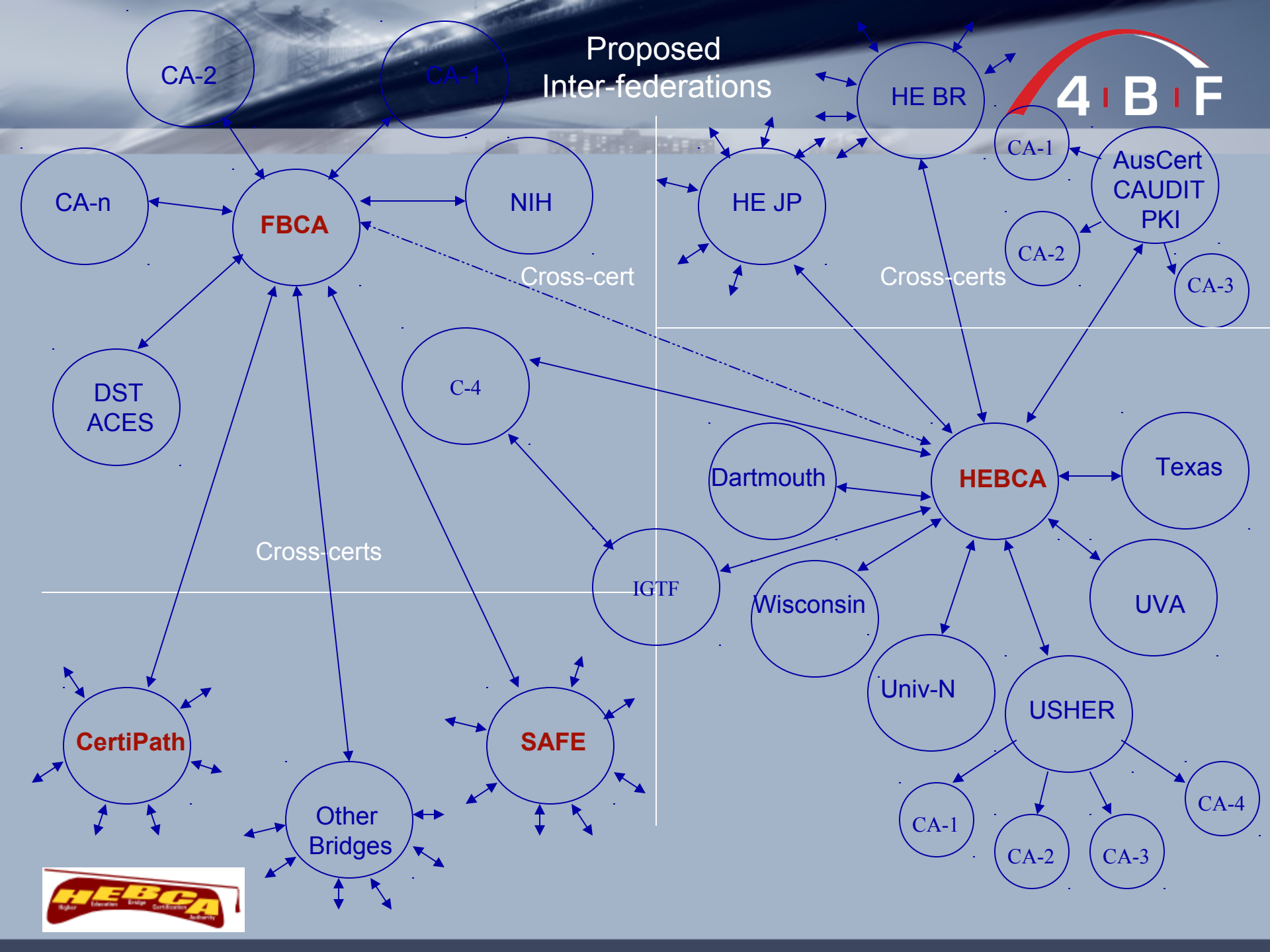
- Applications that utilize PKI in Higher Education
  - Secure Wireless
  - S/MIME email
  - Paperless Office workflow (signed PDF and Word docs)
  - Encrypted File Systems (protecting mobile data assets)
  - Strong SSO
  - Shibboleth/Federations
  - GRID Computing Enabled for Federations
  - E-grants facilitation

HEBCA – A Brief History

- HEBCA started life as pilot project to validate PKI bridge-2-bridge transactions
- Modeled on the successful FBCA, but representing higher education
- Hosted at govt. contractor site, beginning 2001 with involvement from several HE institutions
  - Dartmouth College, University of Wisconsin, University of California – Berkley, University of Alabama, etc.
- EDUCAUSE provided sponsorship to instantiate the infrastructure for real
- Dartmouth College chosen as operating authority in May 2004

HEBCA – A Brief History

- HEBCA rebuilt from the ground up based on prototype infrastructure
- Policy Mapping and technical interoperation completed with FBCA, cross-certification with a limited number of schools and related entities
- HEBCA is ready for production, but still operates in a “Test” mode today
- Steps are underway to migrate infrastructure to a long term commercial operation


Proposed
Inter-federations
4 I B I F
CA-2
CA-1
HE BR
CA-1
AusCert
CAUDIT
PKI
CA-n
FBCA
NIH
HE JP
CA-2
CA-3
Cross-cert
Cross-certs
DST
ACES
C-4
Dartmouth
HEBCA
Texas
Cross-certs
IGTF
Wisconsin
UVA
Univ-N
USHER
CertiPath
SAFE
Other
Bridges
CA-1
CA-2
CA-3
CA-4

HEBCA – A Brief History

- HEBCA provides 5 levels of interoperability
  - Test + 4 levels equivalent to NIST SP800-63
- Audit has been the single most prevalent deterrent to adoption within the community
  - Schools are very consistent and regimented in the processes that they follow for Identity authentication and management, but often do not have formal documentation of those processes, nor audit of those processes by independent 3rd parties.
- Authentication has been the service driving the majority of demand

HEBCA

- HEBCA provides an efficient way for participating organizations to establish trust of any identities issued by other participants
- HEBCA uses technological and policy-based processes to assert the level of assurance that community members can place in a given identity certificate.
- As each participant joins HEBCA, their identity credentialing processes are reviewed and an assurance value is assigned to their certificates on a scale recognized within the community.
- Instead of each member establishing bilateral trust agreements, and reviewing the policies and procedures of each of all the other participants, they can simply trust the validity of the identity which HEBCA has vetted and asserted across its entire system
- HEBCA's participation in the 4BF enables a far greater community of trust for its participants beyond just higher education

HEBCA

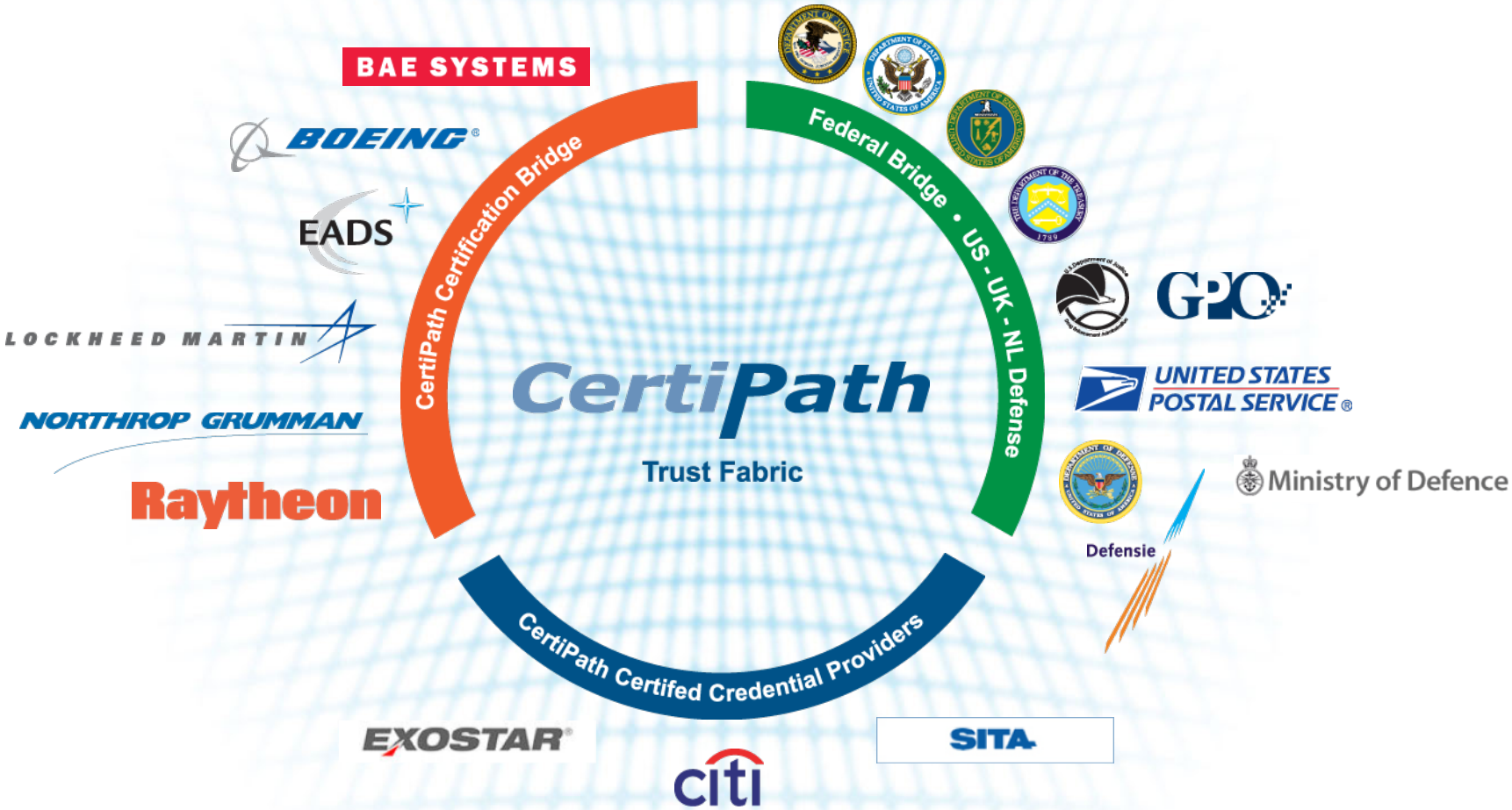
- NOTE: HEBCA is still only operating in “Test” mode
  - Transition is underway to move operations to commercial CA vendor (DigiCert Inc.)
  - Root will be re-issued & participants re-cross-certified
  - Expect full production operations by Q4 2010
  - Scott Rea: scott.rea@dartmouth.edu

The 4BF

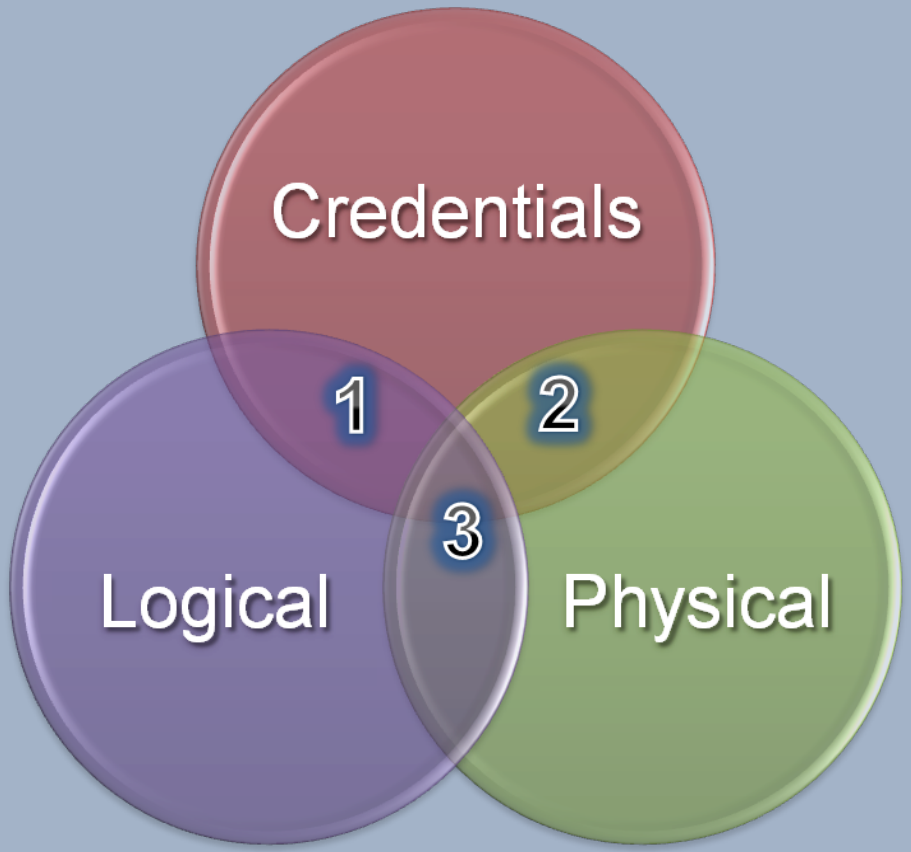
The Four Bridges Forum

Jeff Nigriny
CertiPath

CertiPath Trust Fabric



The “Bridge” between LACS and PACS

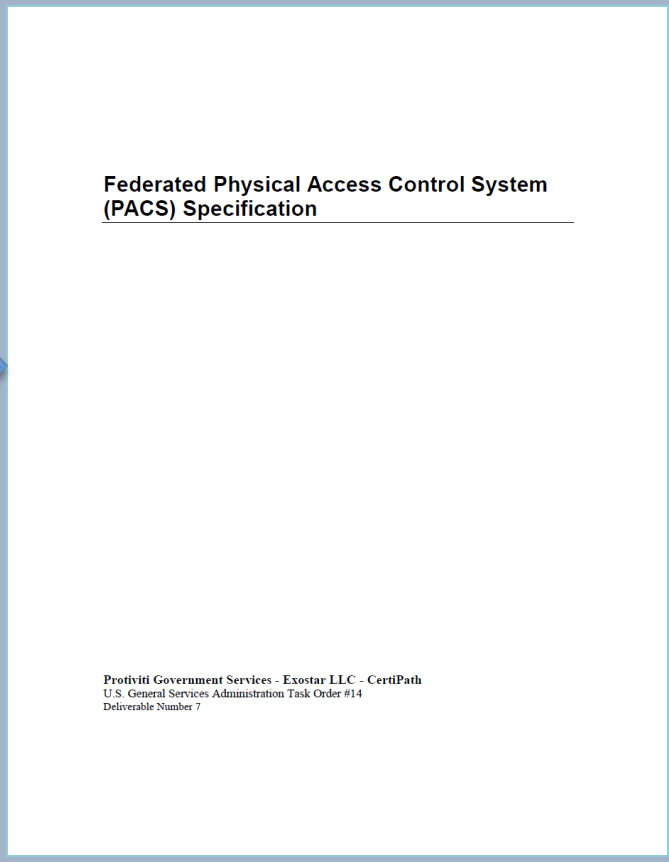


1. Traditional LACS space marked by PKI, OTP, and UID/Password leveraged through Smart Card Logon, Federated Access Gateways, SSL, S/MIME
2. Traditional PACS space marked by Magstripe and Prox, however PKI on PIV/-I and CAC is quickly becoming best practice for Federal Facilities
3. Credentials which work in either application are the missing link to gaining situational awareness through logical and physical networked “intelligence points”

Growing Pains

- PKI in PACS is easier said than done
  - PACS Vendors and integrators are commercially aligned to avoid interoperable credentials
- Poor implementations hurt everyone
- All of the supporting infrastructure for interoperable credential usage in LACS is missing for PACS

GSA Trusted PACS Specification



Version 1 of the Trusted PACS Specification was published by GSA on March 9th, 2010

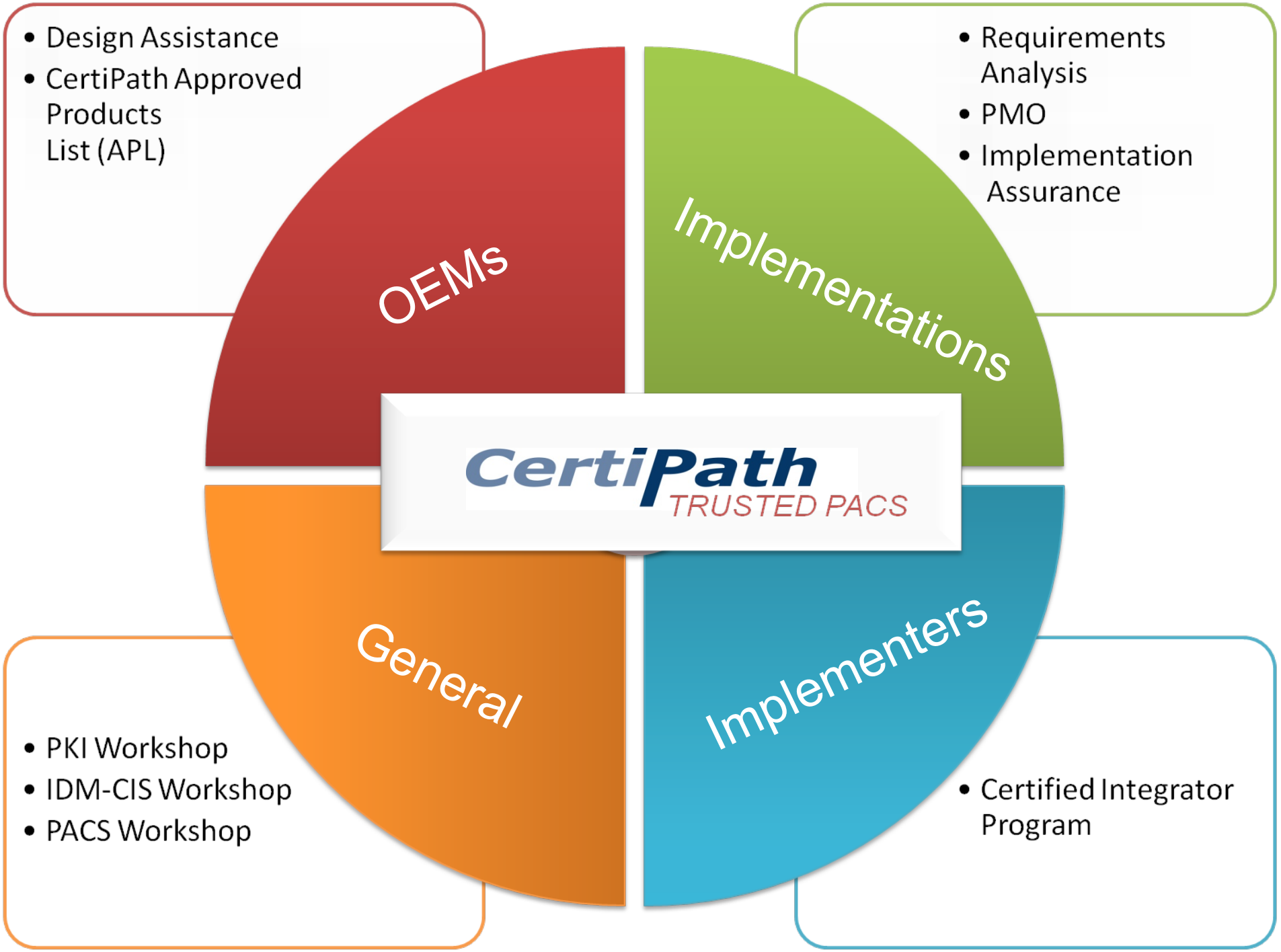
Policy - LACS & Credentials vs. PACS

Interoperable high assurance LACS and Credential standards/policies exist to:

- Define the need
  - Many e.g., OMB M-04-04, SP 800-79, ISO 27799, etc.
- Define the form
  - Many, e.g., x.509, SP 800-73, SAML
- Define audit/C&A
  - Many, e.g. FIPS-201 APL, FISMA, SOX, etc.
- Define interoperability
  - Many, e.g., The 4BF's CPs, OpenID, Kantara
- Define the requirement for industry
  - None

Interoperable high assurance PACS standards/policies exist to:

- Define the need
  - Few e.g., SP 800-116, DTM-09-012
- Define the form
  - Closest to date is TWIC, FRAC
- Define audit/C&A
  - None and worse, FIPS-201 APL is causing confusion
- Define interoperability
  - One, GSA Trusted PACS Specification
- Define the requirement for industry
  - None

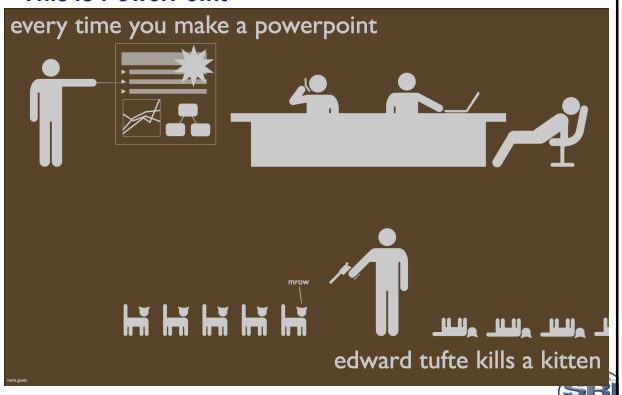

• Design Assistance
• CertiPath Approved Products List (APL)
OEMs
• Requirements Analysis
• PMO
• Implementation Assurance
Implementations
CertiPath
TRUSTED PACS
General
• PKI Workshop
• IDM-CIS Workshop
• PACS Workshop
Implementers
• Certified Integrator Program

Internet Voting – Threat or Menace?

April 14, 2010

Jeremy Epstein
Senior Computer Scientist
SRI International
Arlington VA

This *is* PowerPoint



Outline…

- Background – definitions & requirements
- How Internet voting works
- Some potential solutions
- How can PKI help with Internet voting
- Is it a threat or a menace?

What is Internet voting (i-voting)?

√ 1. Getting information on candidates, contests, etc.
√ 2. Voter registration – get blank form, fill out, submit, receive ACK
√ 3. Absentee ballot request – get request form, fill out, submit, receive ACK
4. Fill out & submit ballot
√ a. Get blank ballot
√ b. Fill out
X c. Return
√ d. Receive ACK

Advantages of i-voting

- "More modern"
- Potential for higher turnout, especially for young voters
- Potential for lower cost
- Reduce precinct staffing issues
- Enable military/overseas voters - Uniformed and Overseas Citizens Absentee Voting Act (UOCAVA) compliance

Voting System Requirements

- Allows each authorized voter to vote exactly once
- Accurately records the votes
- Accurately counts the votes
- Voter can be sure his/her vote is counted, without trusting the other side's people, even if the other side's people are the election officials (*)
- Voter can be sure his/her vote is counted, without trusting the company that made or programmed the voting equipment
- No one can learn how he/she voted without his/her cooperation
- No one can prove how he/she voted even *with* his/her cooperation

} Avoid coercion

(*) Election officials are overwhelmingly honest, but the system can't depend on that

None of these are absolutes – all voting systems make some level of compromises

Unique issues with voting (Internet or otherwise)

- Once-every-four-years voters (can't rely on special-purpose devices, software)
- Process must be understandable to everyone
- Must be usable and accessible to all citizens – including low-income, seniors, non English-speaking
- Process is largely run by minimally trained (but hardworking!) senior citizens
- Many ballot styles – hundreds or thousands per state
- Highly cost sensitive – no one wins election by promising to invest more in elections!

Email ballot submission is a bad idea…

- No privacy
  - Email is store and forward, so any machine (or administrator of the machine) can read the message
- No authentication
  - To/from headers aren't trustworthy
- No integrity
  - Contents of the email may be modified at any hop

PKI can address all of these, *if* you can get certificates to voters that they then have to find and use successfully once every four years

Mail-in ballots (absentee, VBM)

- Privacy – double envelopes, but no protection against vote selling
- Authentication – signatures, but signature checking is weak
- Integrity – controls on physical mail (stronger than email)

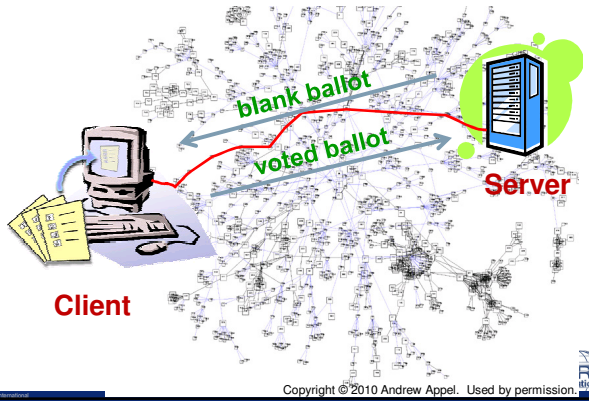
- Lots of historical problems with privacy, esp. in nursing homes
- Overseas VBM has to trust (at least) two countries' mail systems
- Definite risks, but wholesale attacks much harder than email

- Absentee (excused or no-excuses) everywhere in the US
- All-VBM in Oregon, largely VBM in California and Washington

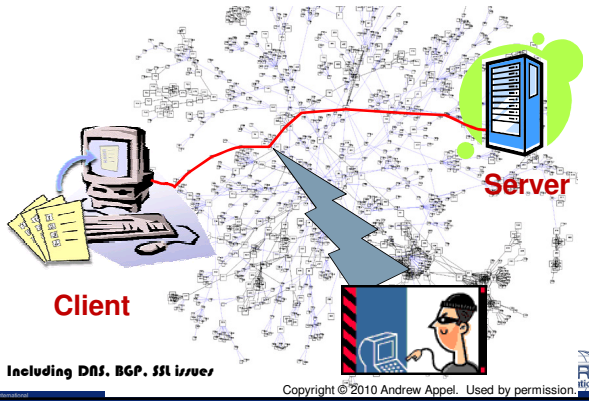
Types of i-voting

- Home-based (personal computer, cellphone, etc)
  - More convenient
  - May be more accessible for voters with disabilities
  - Less expensive for locality
- Kiosk-based (dedicated controlled system)
  - More even playing field (poor voters aren't at a disadvantage)
  - More controlled environment (physical and software controls, voter authentication by a trusted person, reduced risk of in-person coercion)
  - Essentially no different than a precinct-based system

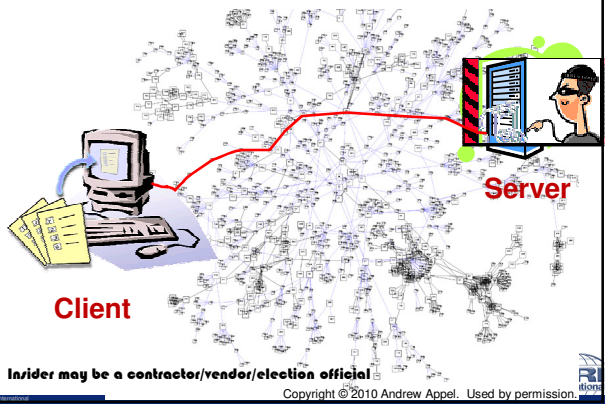
Simple i-voting Protocol



Communications Security

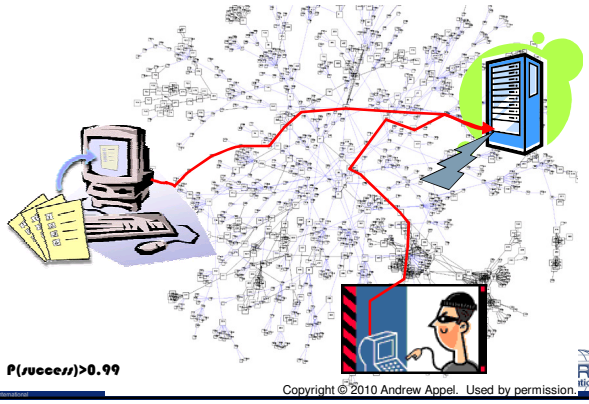


Insider Attacks



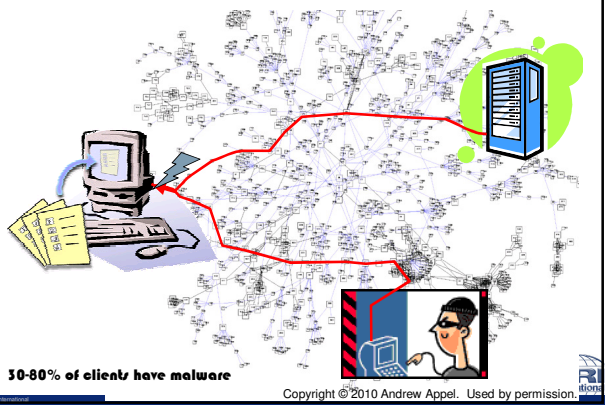
Insider may be a contractor/vendor/election official

Vulnerability in Server to Outside Attacks



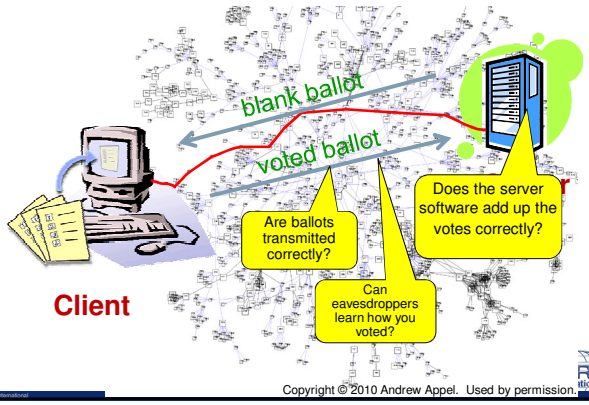
P(success)>0.99

Vulnerability in Clients

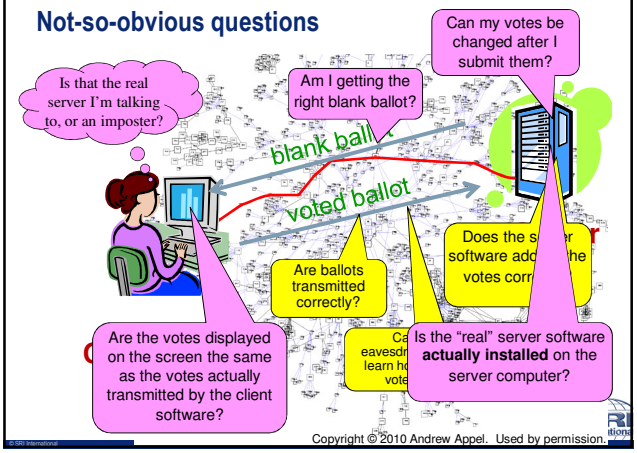


30-80% of clients have malware

Obvious Questions



Not-so-obvious questions



If I can bank online, why can't I vote online?

ZeuS, May 2009

Other online elections

- Shareowners (board of directors, etc)
  - Possibly different threat models - long history of attacks against political elections
  - No requirement for ballot anonymity
- Political primaries (e.g., Democrats Abroad 2008)
  - Run by parties, not the government
  - No requirement for ballot anonymity
  - No requirement for auditability
  - Not governed by any Federal (or sometimes even state) regulations
- Local elections (e.g., Honolulu neighborhood boards 2009)
  - Much lower threat model (less to gain, less to spend)
  - Not governed by any Federal (or sometimes even state) regulations

Some Obvious Solutions That (Might) Work

- Using SSNs as authenticators
- Digital signatures on ballots
- End-to-end crypto
- Out-of-band vote confirmation
- Signed vote summary to voter
- Bullet-proof server to store votes
- Paper backup of votes

Using SSNs to authorize voting

- Can we make it easier to authorize voters for online voting?
- Idea: Have voters sign in by providing their SSN and then cast a ballot, which could be encrypted using their SSN as a key.

Avoids having to create a new identifier that voters can't remember.

Not all US citizens have SSNs, some non-citizens do (www.ssa.gov/pubs/10096.html).
Generally speaking it's illegal.
SSN isn't secret!

Digital Signatures on Ballots

- Can we avoid the need to fully trust the client and server computers?
- Idea: Let each voter (digitally) sign her ballot, and post every ballot on a public (Internet) bulletin board.

Accurate and trustworthy: Each voter can verify that her ballot is present; any member of the public can add up all the posted votes and reconfirm election results.

How do we get keys/certificates to voters?
Complete loss of voter privacy!

Cryptographic End-to-End Protocols

- Can we allow posting votes without compromising voter privacy?
- Idea: Let each voter (digitally) sign her ballot, and post every ballot on a public (Internet) bulletin board. **But use special-purpose encryption protocols to avoid loss of voter privacy**

Each voter can verify (probabilistically) that her ballot is (very likely) present; any member of the public can add up all the posted votes (probabilistically) and reconfirm election results.

Do these protocols actually work? Can they be explained to voters and policymakers? Are policymakers able to evaluate these protocols? Are there hidden vulnerabilities?

Out-of-Band Vote Confirmation (1)

- Can we avoid the threat of malware on the voter's computer?
- Idea: Have a chart of images associated with each candidate and published in the newspaper; server sends back the right image to the voter to prove that the voter's computer transmitted the vote correctly.

Gives warm fuzzies that the voter's intent is captured. If lots of choices, increases effort for malware author to give a "right" image to the voter.

Extra steps for voters (who aren't likely to check). If only one image per candidate, malware can provide the image regardless of the vote cast.

Out-of-Band Vote Confirmation (2)

- Can we avoid the threat of malware on the voter's computer?
- Idea: Have the server call the voter on the phone to read back the votes.

Allows online voting without the slow process of picking candidates from a phone menu.

Extra steps for voters (who aren't likely to check). New opportunity for vote selling. How does a voter prove that the call-back doesn't match what they wanted? Not all voters have phones!

Signed Email with Vote Summary

- Can we avoid the threat of malware on the voter's computer?
- Idea: Have the server send an email back to the voter with a digitally signed copy of their votes.

If voter uses web-based email (e.g., Gmail), can check results anywhere, so compromise of vote casting computer isn't catastrophic.

Extra steps for voters (who aren't likely to check). Many voters won't have access to multiple computers. New opportunity for vote selling. Teaching voters to check email signatures is hard!

Provide a Bullet-Proof Server to Store Votes

- Can we avoid the risk that someone (insider or outsider) will hack into the server and add or change votes?
- ~~Idea: Have understaffed non-technical election officials set up the system.~~
- ~~Idea: Have unaccountable outsourced vendors set up the system.~~
- ~~Idea: Have Google run the election (c.f. Aurora).~~
- Note: This method used by Democrats Abroad for their 2008 pilot program.

Faith based voting?

This is perhaps the biggest threat of all.

Paper Backup of Votes

- Can we use paper backups of votes in case there's a system failure?
- Idea: Print the voted ballots and use those for audits and recounts.
- Note: This method used in Okaloosa County Florida for their 2008 pilot program.

Voter can check that the computer recorded the vote correctly by examining paper. Allows audit to verify that electronic tallies are correct.

Does this actually solve a problem? Why not just mark the paper by hand and send it in the mail?

Use CAC for military voters

- Can we use existing infrastructure to authenticate voters?
- Idea: Have military voters use their Common Access Cards (CAC).

Provides strong remote authentication, with existing processes designed to deter credential sharing.

Military isn't likely to share CAC authentication with 5000 localities nationwide. At best, solves authentication problem.

Where can PKI help?

- For electronic ballot distribution, allow voters to ensure that they got the right ballot
- For acknowledgement of voter registration requests, absentee ballot requests, and completed ballot receipt, signed emails
- Possibly for authentication for military voters using CAC

- *But expecting voters to maintain a PKI certificate for use once every four years is a non-starter*

So is it a threat or a menace?

THREAT

MENACE

So is it a threat or a menace? (Take 2)

THREAT: a warning that something unpleasant is imminent

- Lots of political movement towards i-voting because it sounds like a good idea
- Little understanding by elected officials of the technological risks or similarities and differences compared to e-banking

MENACE: pose a threat to; present a danger to

- I-voting is a danger to accurate vote counting given current and reasonably foreseeable technology
- E2E technologies may reduce the risk for kiosk voting systems

Going forward

- Encourage the *relatively* safe parts
  - Online voter registration (backed up with in-person identity checks)
  - Online absentee ballot requests
  - Online absentee ballot distribution
  - Online absentee ballot receipt acknowledgement
- … and stick with mail-in paper for the critical ballot submission

For more reading/viewing

- Recent OVF/UOCAVA Internet voting debate
  - http://www.youtube.com/OverseasVote
- Open source voting
  - Open Source Digital Voting Foundation - www.osdv.org
  - Elections by The People Foundation – www.electionsbythepeople.org
- NIST End to End Voting Systems Workshop - http://csrc.nist.gov/groups/ST/e2evoting/index.html
- *Internet Voting: Will We Cast Our Next Votes Online?*, Jeremy Epstein, ACM Computing Reviews, December 2009.
- *A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE)*, David Jefferson et al, January 2004 (updated June 2007), www.servesecurityreport.org
- *Report of the National Workshop on Internet Voting: Issues and Research Agenda*, Internet Policy Institute, March 2001.

Independently-Verifiable Secret-Ballot Elections

Poorvi L. Vora
Department of Computer Science
The George Washington University

Outline

- Current voting technology, limitations
- Cryptographic approach; paradigm shift
- “End-to-end” voting systems
- Electronic E2E voting systems

Current Technology

In the world's oldest continuous democracy

- Humboldt County, CA: voting machines dropped 197 votes – Wired, 12-8-2008
- Florida's 13th Congressional District (2006): One in seven votes recorded on voting systems was blank – US Government Accountability Office, 2-8-2008
- Franklin County, Ohio: computer error gave Bush 3,893 extra votes in one precinct – WaPo, 11-6-2004
- In a North Carolina County: 4,500 votes were lost – WaPo, 11-6- 2004

Voting Machine Analysis

- Kohno *et al* (2004): Diebold AccuVote-TS DRE*
  - Voters **can cast unlimited votes** without detection
  - Insiders **can modify votes** and match votes to voters

- Felten (2006)
  - "**Hotel Minibar Keys** Open Diebold Voting Machines

- Bishop, Wagner *et al* (2007): CA "Top to Bottom Review"
  - Voter can **insert a virus** into code
  - Virus can spread through the state's election system

And so on ….

optical scan (Kiayias *et al,* 2007), Ohio voting machines OS + DRE (McDaniel *et al*, 2007); NJ DREs (Appel *et al*, 2009);

*DRE: Direct Recording Electronic

More exhaustive testing?

- Not possible to test large programs for the absence of errors
- Cannot rely only on
  - software and
  - software testing
- Go back to paper, or keep paper back-up

At least “we” can count paper

BUT

- Everyone cannot use paper
- Inefficient and inaccurate counts and recounts

(e.g. Minnesota Senate election)

Problems of integrity remain

- “we” = persons with privilege
- Still need to secure cast ballots till counting

Integrity Issues

Are these our only choices:

- **Trust:**
  - chain of custody of voting systems/paper back-up and
  - those who count

  OR
- **Watch**
  - all locks on all precincts, and
  - all counts

Cryptographic Voting Systems

Paradigm Shift

Audit the Election Not the Equipment

Instead of checking

- all the software, and
- that it will perform **several operations** correctly **every time**

Determine that only the tally is correct, only this time

Encrypted Paper Trail

RECEIPT No. 7151058

PRESIDENTIAL PRIMARY X897

PROPOSITION 51 34M7

PROPOSITION 52 66Y9

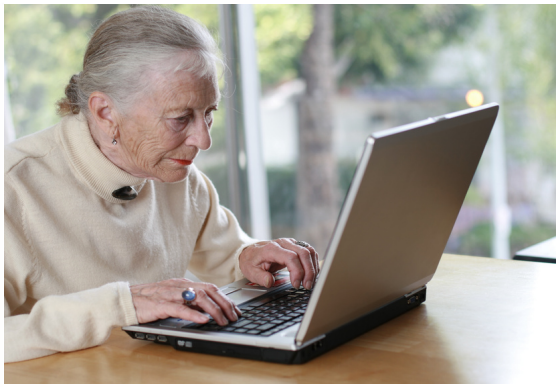
PROPOSITION 53 NG76

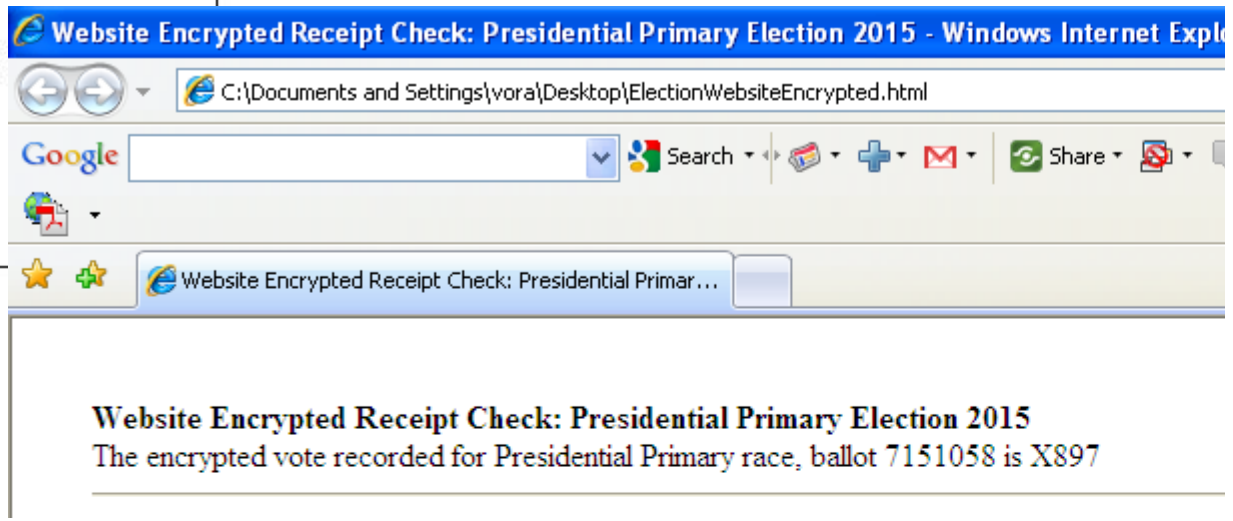
PROPOSITION 54 12AE

PROPOSITION 55 REWQ

PROPOSITION 56 SDU5

PROPOSITION 57 GECZ

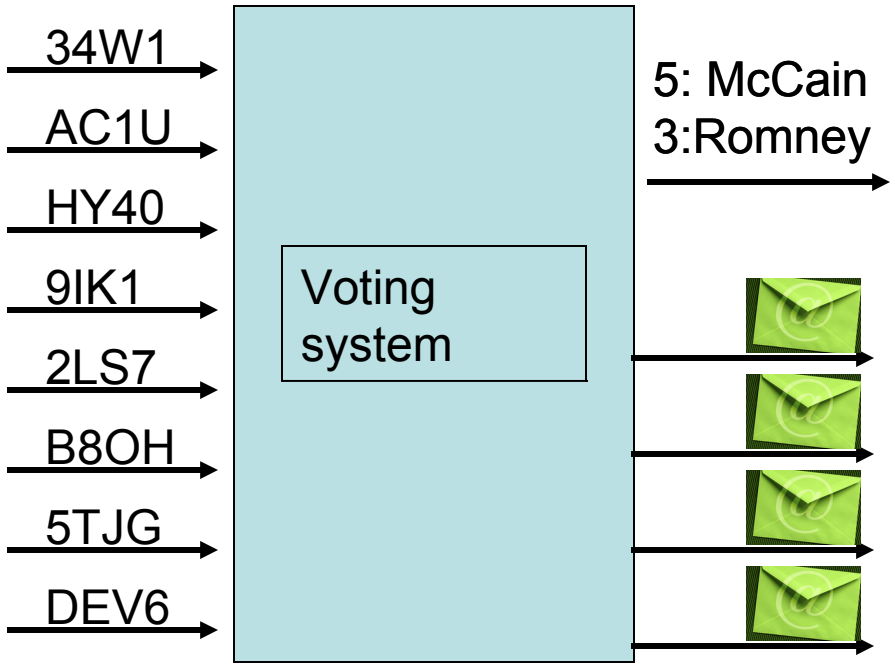




1. Voter Casts Encrypted Vote and Takes Copy out of Polling Booth

2. Voter Checks Receipt on Website/Newspaper

Tally Computation



Tally

Public digital audit trail

- commitment by voting system

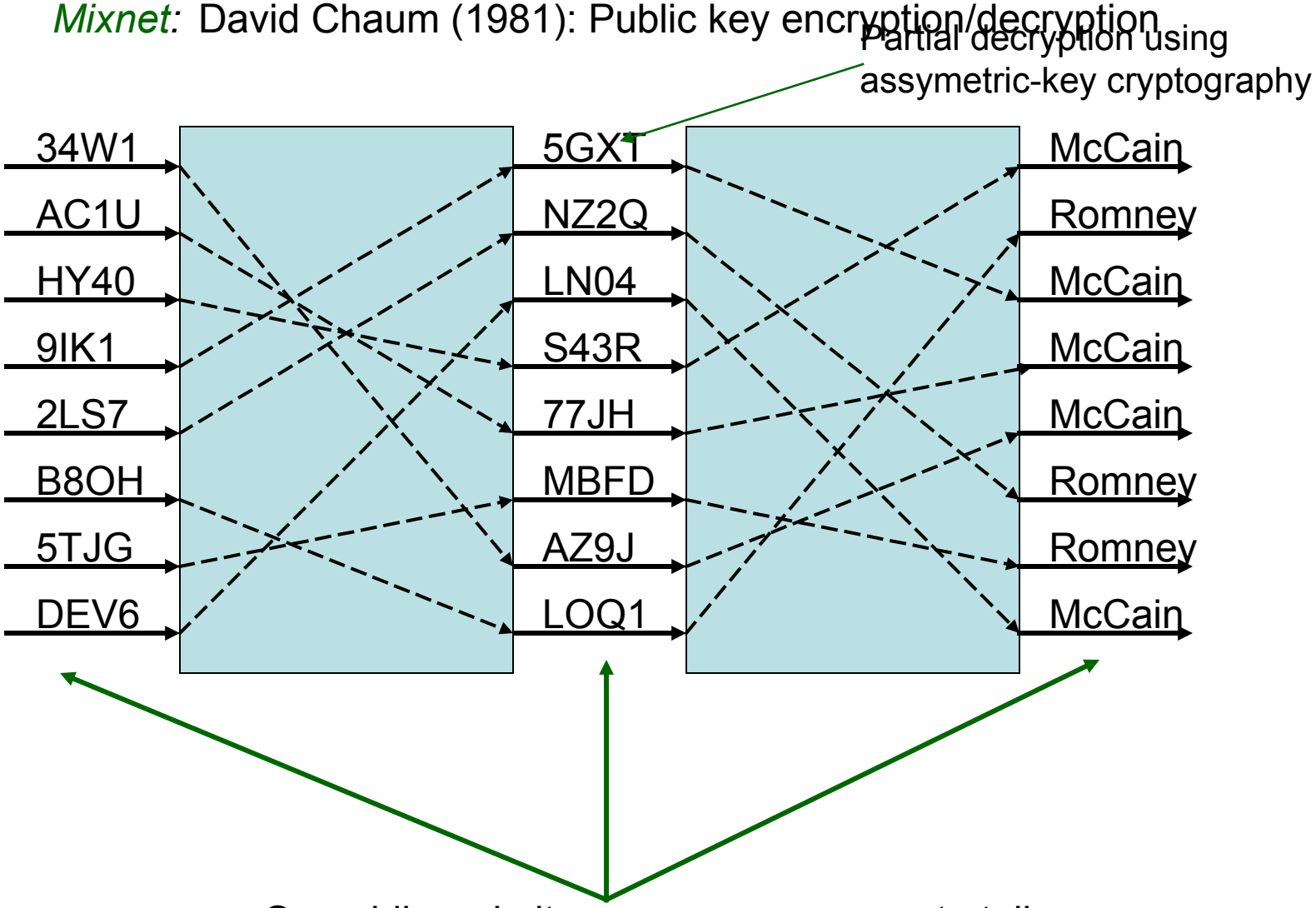
for proof of tally

3. Voting system reveals tally and a digital audit trail to begin the proof of tally correctness

For example: Invention of Secure Electronic Voting

Votes are decrypted and shuffled

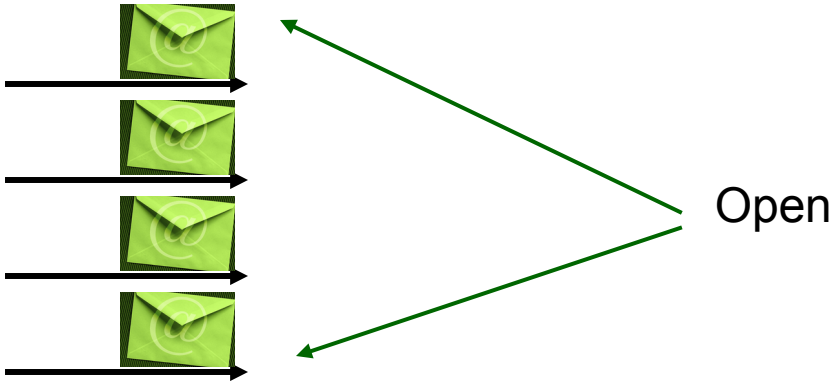
Mixnet: David Chaum (1981): Public key encryption/decryption



Tally Audit

4. Public audit performed by auditors

Successful audit verifies tally without revealing information on votes



Voting protocols can protect

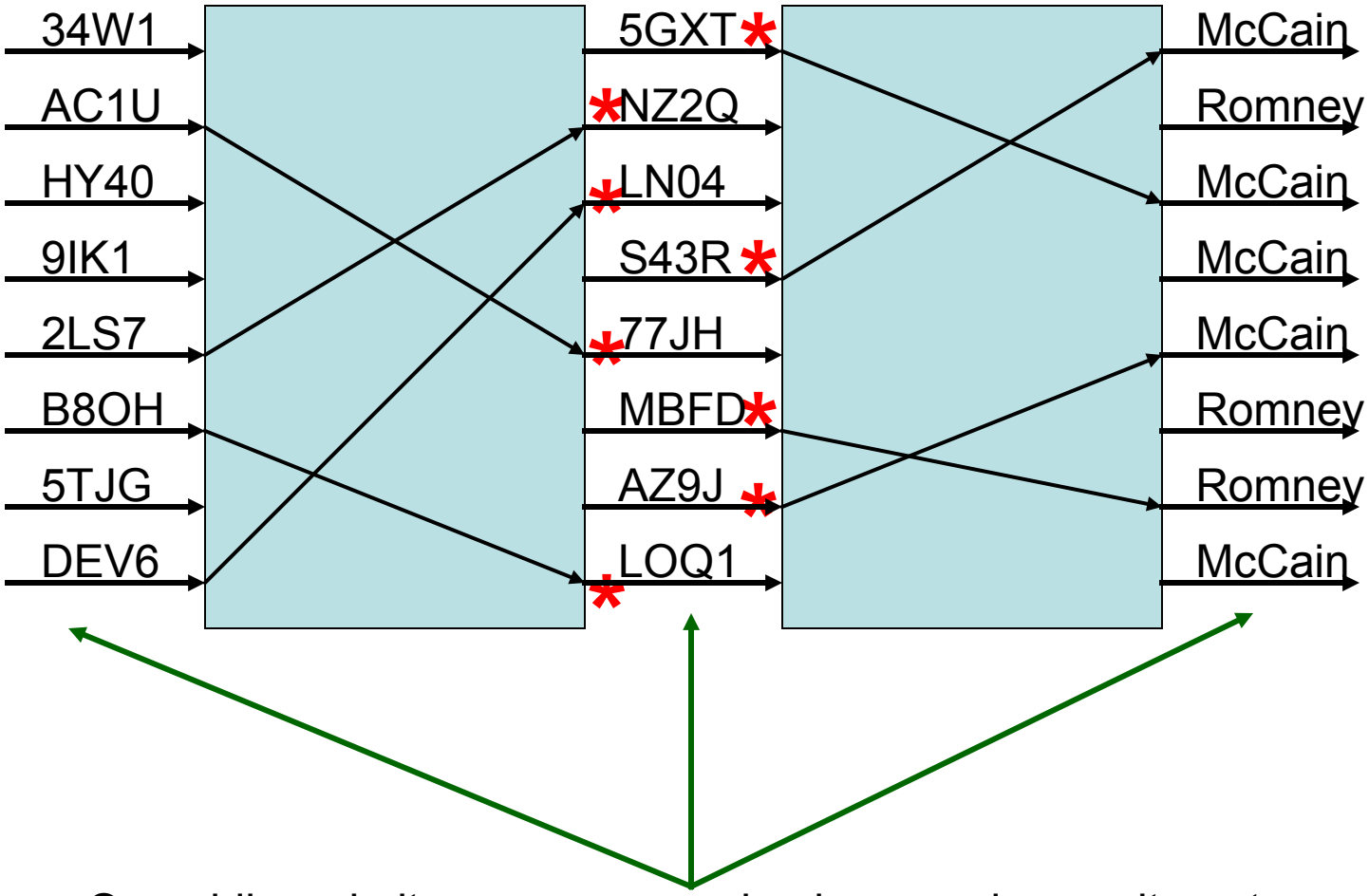
tally integrity or

vote secrecy (but **not both**)

against an adversary who can break the cryptography

For Example: Tally Audit

Jakobsson, Juels, Rivest (2002)



On public website: anyone can check opened commitments

The story so far (in 2002) …

- Very interesting theoretical results

Chaum (1981), Cohen (now Benaloh) and Fischer (1985), Benaloh and Tuinstra (1994), Sako and Kilian (1995),

  - Relevant: zero-knowledge proofs and interactive/non-interactive proofs (*e.g.* Goldwasser-Micali-Rackoff (1985))

- BUT: Computers vote OR humans encrypt votes

- Encryption on trusted machines
  - Cannot use in polling booth
  - Cannot use to vote from home, because
    - Home PCs can have viruses
    - Adversary can threaten or bribe voter

Trusted encryption without trusted encryption device?

E2E Systems: Voter-Verifiable Voting

Voters need not trust encryption device

- **Electronic:**

Chaum (2002-3); Neff (2004); Benaloh (2006); VoteBox (2007)

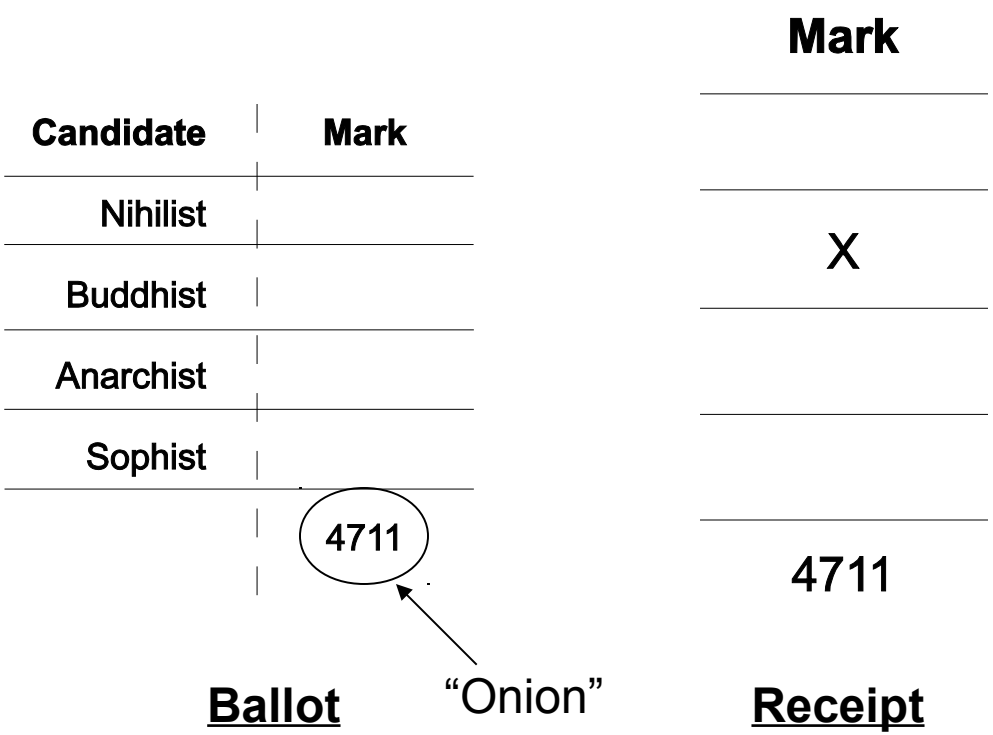
- **Paper Ballots:**

Prêt à Voter (2005); Punchscan (2005); Scratch and Vote (2006); Voting Ducks (2006); Scantegrity (2007)

- **Remote:**

Rijnland Internet Election System (RIES) Netherlands governmental elections (2004, 2006); Helios (2008); not resistant to remote coercion

Example: Prêt à Voter
Ryan *et al*, 2005



1. System encrypts vote
2. Voters can choose to audit the encryption or cast it
3. Audit ballot by opening onion

Picture from Stefan Popoveniuc, PhD Dissertation, GW, 2009

Scantegrity II

Takoma Park Municipal Election: 2009

Scantegrity II front end + Punchscan back-end

UMBC, GW, MIT, Waterloo, UOttawa

First fully-voter-verifiable secret-ballot governmental election

- November 3, 2009: Takoma Park, MD
- Mayor + 6 Council Members
- 1728 votes cast (10,934 registered voters)
- Candidates were ranked by voters (*instant runoff voting*)
- Unique:
  - Public audit of tally
  - Open-source
  - Fully-verifiable by voters

Scantegrity II (2008)

UMBC, GW, MIT, Waterloo, UOttawa

1-392060

Stub Number:

City of Takoma Park, Maryland
MUNICIPAL ELECTION
NOVEMBER 3, 2009

OFFICIAL BALLOT — WARD 1

Ciudad de Takoma Park, Maryland
ELECCIONES MUNICIPALES
3 DE NOVIEMBRE DE 2009

BOLETA OFICIAL— DISTRITO ELECTORAL 1

MAYOR *ALCALDE*			
Rank candidates in order of choice *Clasifique a los candidatos por orden de preferencia*	1st choice *1ra opción*	2nd choice *2da opción*	3rd choice *3ra opción*
Roger B. Schlegel			
Bruce Williams			
Write-In Candidate/*Para añadir a un candidato*			

CITY COUNCIL MEMBER WARD 1 *MIEMBRO DEL CONSEJO DE LA CIUDAD DISTRITO ELECTORAL 1*		
Rank candidates in order of choice *Clasifique a los candidatos por orden de preferencia*	1st choice *1ra opción*	2nd choice *2da opción*
Josh Wright		
Write-In Candidate/*Para añadir a un candidato*		

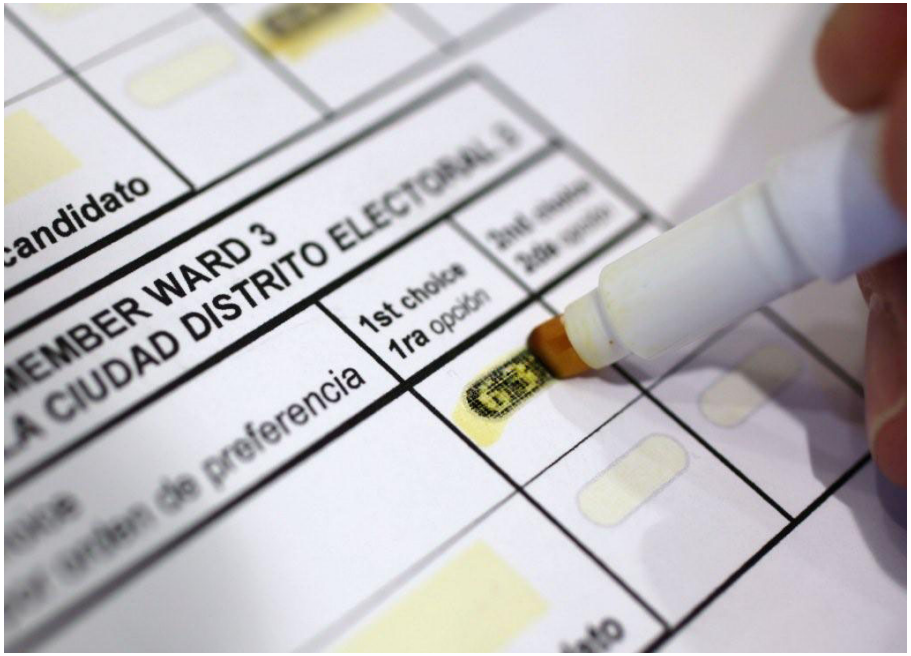


Photo by Alex Rivest

1-634527

Online Verification Number/
Número de Verificación por Internet

INSTRUCTIONS FOR VERIFYING YOUR VOTE ON-LINE AFTER YOU RETURN HOME

PARA LAS INSTRUCCIONES EN ESPAÑOL VEA AL DORSO

You have the **OPTION** of verifying your vote on-line after you return home. **It is not necessary to do so**. You may ignore this step entirely; **your cast ballot will be counted whether or not you do this verification.**

If you wish to verify your vote on-line, perform the following steps:

1. Fill out your ballot according to the instructions provided on the ballot. "Confirmation numbers" will appear inside the ovals you mark.
2. **BEFORE** YOU CAST YOUR BALLOT Record the Online Verification Number and the confirmation numbers below, using the narrow tip of the special pen (note that Wards 1-5 will not have a 3rd choice confirmation number for the city council race).

"On-Line Verification Number" from the bottom right corner of your ballot

Confirmation Numbers	1st Choice	2nd Choice	3rd Choice
Mayor			
City Council Member			

3. **Cast your ballot as usual using the poll-site scanner. DO NOT CAST THIS SHEET, but take it home with you**.
4. After you have returned home, use a computer with an Internet connection to access the City Clerk's web page: **www.takomaparkmd.gov/clerk**. Here you will see instructions for verifying that the confirmation numbers you wrote down are correctly recorded. Note that the confirmation numbers are randomly generated and cannot be used to determine your vote.

Thank you for verifying your vote!
The Takoma Park Board of Elections

Website Verification

- Immediately after election (10-11 pm)
  - Scantegrity count announced
  - Codes made available online
- 81 unique ballot verifications, 64 before Takoma Park complaint deadline (Nov. 6)
- One complaint
  - Codes not clear enough for one voter
  - Voter noted “0”
  - Scantegrity website said “8”
  - Voter trusted Scantegrity code was correct
  - Audit check later revealed Scantegrity code was correct

Audits: (Closed) Manual Vote Count

- November 5, afternoon
- Jointly by Scantegrity and Takoma Park
- Corroborated Scantegrity total
- Few differences, due to difference between:
  - machine reading (by scanner) and
  - human determination of *voter intent*
- Election certified at 7 pm.
  - by Chair, Board of Elections, to City Council

Audits: Encryption Audit

Lillie Coney*

Audited ballots through the day

- Chose about 50 ballots at random
- Exposed all confirmation codes
- Took home copies of marked ballots
- Checked them against commitments when opened after election

With familiarity, voters, including candidate representatives, can do this too

* Associate Director, Electronic Privacy Information Center and Public Policy Coordinator for the National Committee for Voting Integrity (NCVI)

Audits: Digital Audit Trail

Dr. Ben Adida* and Dr. Filip Zagórski+

- Audited the entire digital audit trail and independently confirmed tally correctness
- Provided their own copy of confirmation codes for voter check
- Pointed out discrepancies in documentation

* *Helios* and Center for Research on Computation and Society, Harvard University
+Institute of Mathematics and Computer Science, Wroclaw University of Technology, Poland

Universally Verifiable

Anyone can perform the audits performed by Adida and Zagórski

- BoE Chair expects other voters will, using software provided by Adida and Zagórski
- Voters can write their own software, using Scantegrity public spec

Limitations

- Bulletin Board (website) needs to be secure
  - Ensure that it doesn't present one code to voters, another to auditors
  - Hence Adida and Zagórski made their own copies and requested voters to check
- The cryptographic protocol does not prevent ballot stuffing, we had to use procedures
- Paper ballots are inaccessible to those with motor and visual disabilities

Electronic Independently-Verifiable Elections?

Electronic Audit

- Voter: “Vote for Bob”
- System prints encryption and signs it
- Voter: “I want to audit this encryption”
- System shows that it encrypted vote for Alice
- Voter knows system cheated, but no proof without hard record of “Vote for Bob”
- If we keep hard record, then has to be destroyed if voter chooses to vote, not audit
- Need observers during audit. Can we do that without voting system detecting an audit?

Conclusions

- Can have better integrity of election outcome using E2E systems
- Challenges exist in making E2E systems electronic

Acknowledgements

Collaborators:

Carback, Chaum, Clark, Coney, Essex, van de Graaf, Hall, **Hosp**, **Popoveniuc**, Rivest, Ryan, Shen, Sherman, Wagner

At NIST: Hastings, Kelsey, Peralta, Popoveniuc, Regenscheid

Help with Takoma Park election:

City Clerk and Board of Elections, Takoma Park

Independent auditors: Adida, Coney, Zagórski

Survey: Baumeister

Others: **Florescu**, Jones, Relan, **Rubio,** Sonawane,

Support: NSF IIS 0505510, NSF CNS 0831149, NSF CNS 0937267

School of Engineering and Applied Science, GW: start-up funds

Extras

NIST
National Institute of Standards and Technology

INFORMATION TECHNOLOGY LABORATORY

Cyber and Network Security Program

Using the DNS as a Trust Infrastructure with DNSSEC

Scott Rose
NIST
{scott.rose@nist.gov}
IDTrust 2010, April 14, 2010

DNSSEC

U.S. DEPARTMENT OF HOMELAND SECURITY

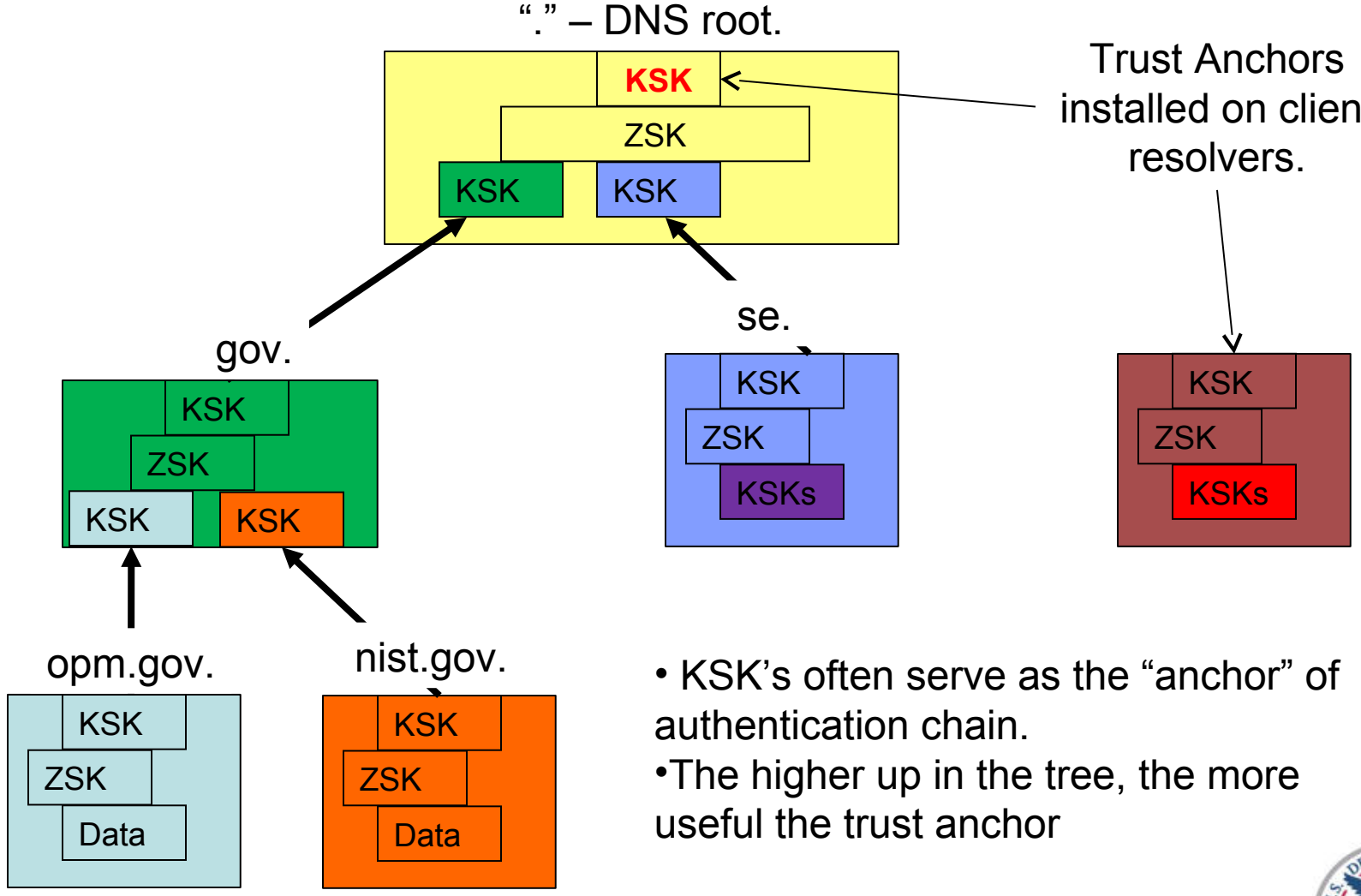
About DNS

- Worldwide database, widest deployed standards-based name system
  - "PKI without the 'K'" – Dan Kaminsky
- Essential component of Internet
  - Robust even in the presence of some errors
  - Often the first part of any Internet transaction
- Due to lightweight, distributed nature, attacks very difficult to detect
  - cache poisoning
  - response re-writing
- In response, the IETF developed the DNS Security Extensions (DNSSEC)

What DNSSEC Provides

- Cryptographic signatures in the DNS
- Integrates with existing server infrastructure and user clients (i.e. Backwards compatible)
- Assures integrity of results returned from DNS queries:
  - Users can validate source authenticity and data integrity
- Checks chain of signatures up to root
  - Protects against tampering in caches, during transmission
- Not provided: confidentiality, security for denial-of-service attacks

DNSSEC Chain of Trust



- KSK's often serve as the "anchor" of authentication chain.
- The higher up in the tree, the more useful the trust anchor

Deployment is Real

- Several TLD's and lower zones are signed now
  - .gov, .org, and country codes like .us, .se, .br…
  - .edu, .net and .com are planning to deploy by 2011
  - Drivers to deploy in .gov – OMB mandate and FISMA
- Root zone to be signed by July 1, 2010
- What's Missing/Still in Development?
  - Application support
  - Stable means to distribute trust anchors
  - Full registrar support

DNSSEC Becoming a Feature

- Tools available
  - Open source software to turnkey appliances
- Becoming available by ISP's (Comcast)
- Integrated into Windows 7 and Windows Server 2008 R2
  - managed via group policy
- Some application patches available
  - Firefox browser and Thunderbird email client
  - Third party plug-ins and patches

So What Does This Get Us?

- Single, distributed, global, lightweight trust infrastructure.
- DNS is a lookup protocol
  - different types of data can be placed in the DNS
    - Example: digital certs, SSH key hashes
  - All would be DNSSEC signed.
- Could we use this to bootstrap trust between organizations?
  - Both would have a common 3rd party trust anchor (root zone for example)
  - Data needed to establish trust in other protocols could be stored in an organization's DNS zone (and signed).

Examples – Bootstrapping Trust

- Crude transport security
  - encoded public keys in DNS CERT RR's to set up secure communication
    - Or SSH key hashes (SSHFP RR's)
  - CERT RR protected by DNSSEC signature
  - IP address of server also protected
  - Not ideal, but could work
    - Need to be sure you are actually talking to the actual server (no IP address spoofing)
- Signed Email
  - user public keys encoded in CERT RRs (e.g. scottr@nist.gov becomes "scottr.nist.gov IN CERT …"

Some Things to Keep in Mind

- DNS has caching and no revoke feature
  - Data is considered valid as long as the signature is valid (replay attacks possible)
  - DNS updates might not be seen until old data times out of caches
- DNSSEC validation would have to be done by the client, or a trusted recursive server
  - Right now, stub clients on desktop/laptop systems rely on an upstream cache to do most of the work (including validation)
  - Do you always trust the recursive server? What about Wi-Fi hotspots?
- No Cross-Signing
  - Hierarchy built upon the existing DNS hierarchy (so "example.com" can't authenticate "sub.example.org")

Resources

- DNSSEC Resources
  - General Information
    - http://www.dnssec.net/
  - NIST DNSSEC Testbed
    - http://www.dnsops.gov/
  - DNSSEC Deployment Initiative
    - http://www.dnssec-deployment.org/
- Root Zone DNSSEC Deployment
  - http://www.root-dnssec.org/

Efficient and Privacy-Preserving Enforcement of Attribute-Based Access Control

Ning Shang*
Microsoft Corporation
One Microsoft Way
Redmond, Washington
nishang@microsoft.com

Federica Paci†
University of Trento
Via Sommarive 14
Povo, Trento, 38123
paci@disi.unitn.it

Elisa Bertino
Purdue University
305 N. University Street
West Lafayette, Indiana
bertino@cs.purdue.edu

ABSTRACT

Modern access control models, developed for protecting data from accesses across the Internet, require to verify the identity of users in order to make sure that users have the required permissions for accessing the data. User's identity consists of data, referred to as *identity attributes*, that encode relevant-security properties of the users. Because identity attributes often convey sensitive information about users, they have to be protected. The Oblivious Commitment-Based Envelope (OCBE) protocols address the protection requirements of both users and service providers. The OCBE protocols makes it possible for a party, referred as sender, to send an encrypted message to a receiver such that the receiver can open the message if and only if its committed value satisfies a predicate and that the sender does not learn anything about the receiver's committed value. The possible predicates are comparison predicates $=, \neq, >, <, \leq, \geq$. In this paper, we present an extension that improves the efficiency of EQ-OCBE protocol, that is, the OCBE protocol for equality predicates. Our extension allows a party to decrypt data sent by a service provider if and only if the party satisfies all the equality conditions in the access control policy.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: [Security and protection]

General Terms

Security

*This work was done while the author was at Purdue University.

†This work was done while the author was at Purdue University.

IDtrust'10 April 13–15, 2010, Gaithersburg, MD

Keywords

Identity, Privacy, Agg-EQ-OCBE

1. INTRODUCTION

Modern data access control models, developed for interactions across different domains and Internet, allow one to specify and enforce access control policies, that is, policies regulating accesses to the protected data, in terms of conditions expressed against *user identity attributes*. Because such attributes often encode relevant-security properties of the users, they have to be protected as well. The implementation of such attribute-based access control models thus requires mechanisms whereby a user obtains access to data if and only if its identity attributes satisfy the service provider[1] policy, whereas the service provider learns nothing about user's identity attributes.

Several approaches based on anonymous credentials [6, 2, 10, 4, 3] have been proposed to allow users to prove that their identity attributes satisfy conditions in the policies by the service provider without revealing the identity attributes in clear. These approaches are based on storing cryptographic commitments of attribute values in certificates and using zero-knowledge proofs protocols [5] to prove properties of these values. A major drawback of those approaches is that, even though the service provider does not learn the attribute values, it learns whether users' identity attributes satisfy its policy conditions and may thus infer information about the values of these attributes.

The Oblivious Commitment-Based Envelope (OCBE) protocols [9] is an approach that addresses such shortcoming and can thus satisfy the protection requirements of both the service providers and the users. The OCBE protocols allow a service provider to send an encrypted message, containing the protected data, to a user such that the user can open the message if and only if the committed value of a specified identity attribute satisfies a predicate. Under such protocol service provider does not learn anything about the user's committed value and does not learn whether the value satisfies the conditions in the access control policy. The possible predicates supported by OCBE are the comparison predicates, that is, $=, \neq, >, <, \leq, \geq$. A major drawback of the OCBE protocol is that it is only able to enforce a condition (consisting of a single predicate) against a single identity attribute. Therefore, if the access control policy requires verifying conditions against several identity attributes, sev-

[1]We use the term 'service provider' to refer to the party managing and securing the protected data.

eral rounds of the protocol have to be carried out which results in inefficient access control. Efficient access control systems are crucial for mobile identity systems and mobile devices.

In this paper, we present the Agg-EQ-OCBE[2] protocol that addresses the efficiency issue of the EQ-OCBE protocol, that is, the OCBE protocol for equality predicates. Our approach provides an efficient approach under which the user can quickly decrypt the data, even when multiple conditions are imposed against its identity attributes. Like the original EQ-OCBE, Agg-EQ-OCBE assures user privacy in that the service provider does not learn the values of the user identity attributes nor whether these attributes verify the access control policies.

The paper is organized as follows. Section 2 reviews the EQ-OCBE protocol. Section 3 presents the Agg-EQ-OCBE protocol. In Section 4 we prove that Agg-EQ-OCBE is secure against a malicious user. Section 5 describes our implementation and performance measurements. Section 6 concludes the paper.

2. OVERVIEW OF THE EQ-OCBE PROTOCOL

We give an overview of the EQ-OCBE protocol in this section. We shall describe the protocol in a more general setting of finite abelian groups. This can be viewed as a natural extension of the originally proposed EQ-OCBE protocol [9].

The EQ-OCBE protocol is built on the Pedersen commitment scheme [12], which is described in [12] in a particular implementation using a subgroup of the multiplicative group of a finite field. Note that this is not intrinsic for the scheme. It also can be implemented using other abelian groups, e.g., elliptic curves over finite fields.

We rewrite the Pedersen commitment scheme as follows.

Definition 1. (The Pedersen Commitment Scheme)
Setup A trusted third party T chooses a finite cyclic group G of large prime order p so that the *computational Diffie-Hellman problem*[3] is hard in G. Write the group operation in G as multiplication. T chooses an element $g \in G$ as a generator, and another element $h \in G$ such that it is hard to find the discrete logarithm of h with respect to g, i.e., an integer α such that $h = g^\alpha$. T may or may not know the number α. T publishes G, p, g and h as the system's parameters.
Commit The domain of committed values is the set of integers $D = \{0, 1, \ldots, p-1\}$. For a party U to commit a value $x \in D$, it randomly chooses $r \in D$, and computes the commitment $c = g^x h^r \in G$.
Open U shows the values x and r to open a commitment c. The verifier checks whether $c = g^x h^r$.

The EQ-OCBE is a Diffie-Hellman-like protocol that allows the user to correctly retrieve the protected data only if the user's committed value equals the one specified by the policy of the service provider. It involves three communication parties: a user U, a service provider SP, and a trusted party T which generates initialization parameters for the protocol to use.

There are several cryptographic components in EQ-OCBE:

- A semantically secure symmetric-key encryption algorithm $\mathcal{E}$ (e.g., AES) with keyspace $\{0,1\}^k$. We use $\mathcal{E}_{\mathsf{Key}}[\mathsf{Message}]$ to denote the encrypted plaintext Message with encryption key Key under the encryption algorithm $\mathcal{E}$.
- A finite cyclic group G of large prime order p, over which the computational Diffie-Hellman problem is intractable. The group operation is written multiplicatively.
- A cryptographic hash function $H(\cdot) : G \to \{0,1\}^k$.

We shall describe how the EQ-OCBE protocol works in our case of policy enforcement for an equality condition.

Protocol 1. (EQ-OCBE)
Parameter generation
T runs a Pedersen commitment setup program to generate system parameters $\mathsf{Param} = \langle G, g, h\rangle$. T also outputs the order of G, p.
Commitment This step is a modified version of the one described in [9]. Instead of requiring T to generate the Pedersen commitment, we let U perform this procedure and ask T to verify the validity of the commitment[4] .

To commit to an element $x \in \mathbb{Z}/(p)$, U randomly chooses $r \in \mathbb{Z}/(p)$, computes the Pedersen commitment $c = g^x h^r$, and sends c to T. T asks U to open the commitment c, and checks that U can indeed commit to the value x. T digitally signs c and send its signature to U. This is an alternative to the **CA-Commit** step in the original EQ-OCBE protocol, in which T sends c to SP. By adopting a public-key infrastructure, T can go off-line after this step. Later in communications, U sends c as well as its signature from T to SP; SP verifies the signature is valid, thus believes that the commitment c is valid. In this way, no further communications are needed between T and U.
Interaction

- U makes a data service request to SP.
- Based on this request, SP sends its policy definition, which requires the value $x_o \in \mathbb{Z}/(p)$ be committed by U.
- Upon receiving this policy, U sends a Pedersen commitment, $c = g^x h^r$, signed by T, to SP.
- After verification of T's signature, SP randomly picks $y \in \mathbb{Z}/(p)^*$, computes $\sigma = (cg^{-x_0})^y$, and sends to U a pair $\langle \eta = h^y, C = \mathcal{E}_{H(\sigma)}[M]\rangle$, where M is the message containing the requested data.

Open Upon receiving $\langle \eta, C\rangle$ from SP, U computes $\sigma' = \eta^r$, and decrypts C using $H(\sigma')$.

The adapted EQ-OCBE protocol above guarantees that U can successfully decrypt the ciphertext if its committed

[2]'Agg' stands for 'aggregated'.
[3]For a cyclic group G of order q, written multiplicatively, the computational Diffie-Hellman problem is the following problem: Given a randomly-chosen generator g of G, and g^a, g^b for random $a, b \in \{0, \ldots, q-1\}$, it is computationally intractable to compute the value g^{ab}.
[4]We say a Pedersen commitment c is valid if its holder, U, is allowed to commit to the value x.

value is equal to the one specified in SP's policy, and that it is computationally infeasible for U to do so if otherwise. SP will not know if the message M has been successfully decrypted, without further communications with U.

3. AGGREGATION OF EQ-OCBE

The modification of the original EQ-OCBE protocol works for one equality condition. In many cases, we want the user to be able to decrypt a message, containing the protected data, if and only if several equality conditions are all satisfied. We can do this by dividing the encryption key into many shares, then performing the EQ-OCBE protocol multiple times, once for each share. More specifically, this can be done as follows.

- Suppose the user U requests data from the service provider SP.
- SP responds with its policy which specifies that n values $x_1, \ldots, x_n \in \mathbb{Z}/(p)$ need to be committed by U in order that U can be served.
- U then sends to SP its n corresponding commitments $c_1, \ldots, c_n$.
- SP chooses $n-1$ random messages $M_1, \ldots, M_{n-1}$, which have the same bit length as the to-be-sent message M (containing the data) and sets
$$M_n = M \bigoplus_{i=1}^{n-1} M_i,$$
where $\oplus$ denotes the bitwise exclusive-or operation.
- SP and U performs the interaction and open procedures as above for n times, for n encrypted M_i.
- U computes
$$M = \bigoplus_{i=1}^{n} M_i.$$

However, such an approach is not very efficient in terms of bandwidth and computation. For n such equality conditions, the number of packets sent in communications and the computational cost increase by approximately n times.

We shall present an aggregated version of the EQ-OCBE protocol, Agg-EQ-OCBE, which handles multiple equality conditions at the same time, without significantly increasing computational cost. Agg-EQ-OCBE also requires less bandwidth compared to the above n-round EQ-OCBE.

Protocol 2. (Agg-EQ-OCBE)
In addition to $\mathcal{E}, H(\cdot)$, and G as in EQ-OCBE, another cryptographic component, a cryptographic hash function $H_1(\cdot) : \{0,1\}^* \to \mathbb{Z}/(p)$, is used.
Parameter generation The system parameters $\mathsf{Param} = \langle G, g, h\rangle$ are generated in the same way as in Protocol 1.
Commitment To commit to an element $x \in \mathbb{Z}/(p)$, U randomly chooses $r \in \mathbb{Z}/(p)$, computes the Pedersen commitment of the hash value $H_1(x)$, $c = g^{H_1(x)}h^r$, and sends c to T. T asks U to open the commitment c by revealing x and r. After verifying that x can be committed by U and indeed $c = g^{H_1(x)}h^r$, T digitally signs c and sends the signature to U. U can hold multiple such commitments corresponding to different committed values.
Interaction (with aggregation)

- U makes a data request to SP.
- Based on this request, SP sends its policy, specifying that n values $x_0^{(i)}, i = 1, \ldots, n$, must be committed by U, i.e., U must hold n commitments $c_i = g^{H_1(x_0^{(i)})}h^{r_i}, i = 1, \ldots, n$, all signed by T, in order to be served.
- Upon receiving this policy, U picks its n corresponding commitments c_i, all signed by T, and sends these commitments together with the signatures to SP. Note that these signatures can be sent in an aggregated way, up to the requirements and design of the system, as described in [8, 1]. We shall use aggregate signature in this protocol.
- SP verifies T's signatures, in an aggregated way, for all commitments c_i. SP computes the aggregate commitment
$$c = \prod_{i=1}^{n} c_i,$$
and the value
$$x_0 = \sum_{i=1}^{n} H_1(x_0^{(i)}) \in \mathbb{Z}/(p).$$
SP randomly picks $y \in \mathbb{Z}/(p)^*$, computes $\sigma = (cg^{-x_0})^y$, and sends to U a pair $\langle \eta = h^y, C = \mathcal{E}_{H(\sigma)}[M]\rangle$, where M is the message related to the requested service.

Open
Upon receiving $\langle \eta, C\rangle$ from SP, U computes
$$r = \sum_{i=1}^{n} r_i,$$
and
$$\sigma' = \eta^r.$$
U then decrypts C using $H(\sigma')$.

Definition 2. (Soundness of Agg-EQ-OCBE)
An Agg-EQ-OCBE protocol is *sound*, if the user U, whose committed values $x_0^{(i)}, i = 1, \ldots, n$ are those specified by SP's policy, can output the plain-text message M with non-negligible probability.

It can be easily seen that Agg-EQ-OCBE is sound. When $c_i = g^{H_1(x_0^{(i)})}h^{r_i}$, we have that
$$\begin{aligned}\sigma = (cg^{-x_0})^y &= (\prod_{i=1}^{n} c_i g^{-x_0})^y \\ &= \left(\left(\prod_{i=1}^{n} g^{H_1(x_0^{(i)})}h^{r_i}\right) g^{-\sum_{i=1}^{n} H_1(x_0^{(i)})}\right)^y \\ &= \left(h^{\sum_{i=1}^{n} r_i}\right)^y = (h^r)^y = (h^y)^r = \eta^r.\end{aligned}$$

4. SECURITY ANALYSIS

Due to the unconditional hiding property of the Pedersen commitment scheme, the service provider SP is not able to learn whether any of the user U's attributes satisfy the required conditions in the policy.

The security analysis of EQ-OCBE [9] implies that when a single commitment is considered, it is hard for a user U to obtain useful information if U's committed value is not equal to that specified by SP, i.e., EQ-OCBE is *oblivious*. It can be easily seen that a similar argument holds true for Agg-EQ-OCBE. For the Agg-EQ-OCBE protocol, we have the additional concern that a user U who does not possess all commitments corresponding to the values specified by the SP may still be able to correctly decrypt the communications. For Example, if the SP's policy requires two commitments $c_1 = g^{21}h^{r_1}, c_2 = g^{35}h^{r_2}$ to be presented, a user U who holds two commitments $c_3 = g^{18}h^{r_3}, c_4 = g^{38}h^{r_4}$ can open the envelope, because the two aggregate commitment $c_1 \cdot c_2$ and $c_3 \cdot c_4$ have $56 = 21 + 35 = 18 + 38$ as their exponents for g, although U does not conform to the policy. The Agg-EQ-OCBE is designed to prevent such an attack from happening.

For the security analysis of Agg-EQ-OCBE, we shall introduce a new and reasonable property for the cryptographic hash function $H_1(\cdot) : \{0,1\}^* \to \mathbb{Z}/(p)$ that we use in Agg-EQ-OCBE. This new definition of property relies on the fact that the range of the hash function is a subset of a group, in which group operations can be considered.

Definition 3. (Group 2nd-preimage resistance)
Let $(\widetilde{G}, +)$ be a finite abelian group of large cardinality[5]. Let $\widetilde{H} : \{0,1\}^* \to \widetilde{G}$ be an unkeyed hash function. We say that $\widetilde{H}(\cdot)$ has the property of *group 2nd-preimage resistance* if for any positive integer m and n sufficiently smaller than $|G|$, and for any given m inputs $x_1, \ldots, x_m$, it is computationally infeasible to find n inputs $y_1, \ldots, y_n$, with

$$\{x_1, \ldots, x_m\} \neq \{y_1, \ldots, y_n\},$$

such that

$$\sum_{i=1}^{m} \widetilde{H}(x_i) = \sum_{i=1}^{n} \widetilde{H}(y_i).$$

Note that the group 2nd-preimage resistance property is stronger than the well-known 2nd-preimage resistance property (cf. e.g. [11]) of cryptographic hash functions, where the latter property is an instance of the former with $m = n = 1$. It is not known yet whether the property of group 2nd-preimage resistance is a consequence of the three basic properties of a general cryptographic hash function: preimage resistance, 2nd-preimage resistance, and collision resistance.

Given this definition, we now can give a security proof of Agg-EQ-OCBE.

Since we assume that $\mathcal{E}$ is a semantically secure symmetric-key encryption algorithm, the ability to decrypt a message is equivalent to the knowledge of the secret encryption key. When the hash function H is modeled as a random oracle, the user U can compute this secret key $H(\sigma)$ only if U can compute the value $\sigma = (cg^{-x_0})^y$. We therefore say the Agg-EQ-OCBE protocol is *secure against the user U* when no polynomial time adversary can win the following game with non-negligible probability.

Game:
Players: challenger $\mathcal{C}$, adversary $\mathcal{A}$
Rules:

- $\mathcal{C}$ generates and sends $\mathsf{Param} = \langle G, g, h\rangle$ to $\mathcal{A}$. $\mathcal{C}$ chooses and sends $x_1, \ldots, x_n \in \mathbb{Z}/(p)$ to $\mathcal{A}$. $\mathcal{C}$ chooses $b \in \mathbb{Z}/(p)^*$, and sends h^b to $\mathcal{A}$.
- $\mathcal{A}$ chooses $y_1, \ldots, y_n, r_1, \ldots, r_n \in \mathbb{Z}/(p)$, with $\{x_1, \ldots, x_n\} \neq \{y_1, \ldots, y_n\}$, and sends $y_i, r_i, 1 \leq i \leq n$ to $\mathcal{C}$. $\mathcal{A}$ outputs a value σ'.
- $\mathcal{C}$ computes $c = \prod_{i=1}^{n} g^{H_1(y_i)}h^{r_i}$, $x = \sum_{i=1}^{n} H_1(x_i)$, and
$$\sigma = (cg^{-x})^b.$$
- $\mathcal{A}$ wins the game if $\sigma' = \sigma$.

THEOREM 1. *Assume that the computational Diffie-Hellman problem is intractable in G. Model H as a random oracle, and assume that H_1 has the property of group 2nd-preimage resistance. Then Agg-EQ-OCBE is secure against the user U.*

The proof of Theorem 1 is reported in Appendix A

5. EXPERIMENTAL RESULTS

We have performed an experimental evaluation to compare the performance of the multiple-round EQ-OCBE and Agg-EQ-OCBE protocols. For multiple-round EQ-OCBE, we generate the Pedersen commitments by committing to the actual values $x_0^{(i)}$ and do not introduce the cryptographic hash function $H_1(\cdot)$. For Agg-EQ-OCBE, we use the hash function $H_1(\cdot)$ and commit to the hash values $H_1(x_0^{(i)})$. The experiment compares the creation time of σ and η at the service provider's side, which consists of the most computationally costly part for both protocols, and the derivation time of σ' from η at the user's side. We also compare the creation time of aggregate commitment and the creation time for σ and η ("envelope"), both at the service provider's side. We do not include communication time and symmetric encryption time in the comparisons, which vary with different network settings and plaintext lengths, in order to focus on the core components of the protocols. We also do not include the signature verification time in the comparison, for the same reason. We expect Agg-EQ-OCBE to outperform multiple-round EQ-OCBE, when the number of involved commitments increases.

In our experiment, we choose the group G to be the rational points of the Jacobian variety (aka. Jacobian group) of a genus 2 curve $C : y^2 = x^5 + 26828108228393556449007 36x^3 + 22659135529599310290211 6x^2 + 25476747159529297178999 18x + 47973099597084896730593 50$ over the prime field $\mathbb{F}_q$, with $q = 5 \cdot 10^{24} + 8503491$ (83 bits). The Jacobian group of this curve has a prime order (164 bits)[6]:

$$p = 24999999999994130438600999402209463966197516075699.$$

The parameter generation program chooses non-unit points g and h in the Jacobian group as the base points for constructing the Pedersen commitments.

In the experiment, we run both multiple-round EQ-OCBE and Agg-EQ-OCBE at the service provider's side for $n(1 \leq n \leq 50)$ Pedersen commitments of randomly generated values $x_0^{(i)}, 1 \leq i \leq n$. We use $x_0^{(i)}$ as the exponents of g for

[5]Let $|G|$ denote the cardinality of a set G, for all G.

[6]The data is taken from [7].

multiple-round EQ-OCBE, and the hash values of $x_0^{(i)}$ as the exponents for Agg-EQ-OCBE, where the hash function $H_1(\cdot) : \{0,1\}^* \rightarrow \mathbb{Z}/(p)$ is built on SHA-1. We also simulate the aggregation of n commitments at the user's side for Agg-EQ-OCBE. For each $n, 1 \leq n \leq 50$, we run 50 rounds of both protocols on n Pedersen commitments. In each round, the n Pedersen commitments under test are different (randomly chosen) and we take the average running times of the 50 rounds. The experimental results are presented in Figures 1: From top to bottom:

- Computation time comparison at service provider's side of multiple-round EQ-OCBE and Agg-EQ-OCBE;
- Computation time comparison at user's side of multiple-round EQ-OCBE and Agg-EQ-OCBE;
- Computation time comparison at service provider's side of commitment aggregation and envelope creation, for Agg-EQ-OCBE.

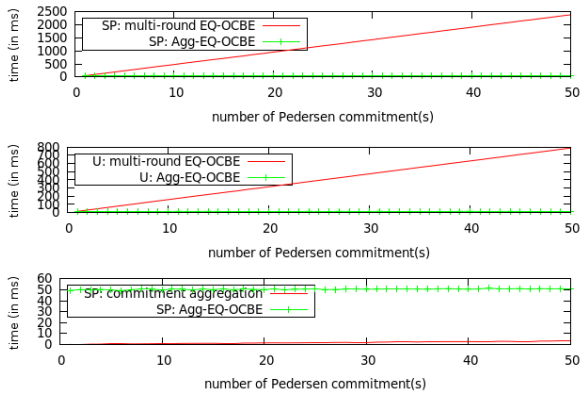


Figure 1: Running time comparison

The experiment was performed on a machine running GNU/Linux kernel version 2.6.9-67.0.1.ELsmp with 4 AMD Opteron (tm) Processor 850 2390MHz and 7.36 Gbytes memory. Only one processor was used for computation. The code is written in C++, and built with gcc version 3.6.4, optimization flag -O2. The code is built over the G2HEC C++ library [13], which implements the arithmetic operations in the Jacobian groups of genus 2 curves.

The experimental results show that while in multi-round EQ-OCBE the running time for composing the EQ-OCBE envelopes linearly increases with the number of involved Pedersen commitments, in Agg-EQ-OCBE it is nearly constant. The experimental results also imply that the overhead of the hash computation introduced in Agg-EQ-OCBE takes negligible time. We have obtained similar results for the envelope opening operations executed at the user's side. We can see that the operation of aggregation of commitments at the service provider's side takes very little time compared to the envelope creation operations. Therefore, Agg-EQ-OCBE is more efficient than the solution based on running EQ-OCBE for multiple rounds.

6. CONCLUSIONS

In this paper, we have proposed, Agg-EQ-OCBE, an extension that improves the efficiency of the EQ-OCBE protocol by allowing a user to decrypt data sent by a service provider if and only if the user satisfies several equality conditions. We have proved the security of our Agg-EQ-OCBE protocol. The experimental results show that the Agg-EQ-OCBE is more efficient than running the EQ-OCBE protocol iteratively for each equality predicate. Future work includes developing efficient OCBE protocols for inequality predicates.

Acknowledgements

The work reported in this paper has been partially supported by the MURI award FA9550-08-1-0265 from the Air Force Office of Scientific Research.

7. REFERENCES

[1] D. Boneh and C. Gentry. Aggregate and verifiably encrypted signatures from bilinear maps. In *Proceedings of Eurocrypt 2003, volume 2656 of LNCS*, pages 416–432. Springer-Verlag, 2003.

[2] S. Brands. Rethinking public key infrastructures and digital certificates: Building in privacy. *MIT Press*, 2000.

[3] J. Camenisch and E. Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proc. Ninth ACM Conf. Computer and Comm. Security*, pages 21–30, 2002.

[4] J. Camenisch and A. Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *Advances in Cryptology, Proc. EUROCRYPT 01*, pages 93–118, 2001.

[5] M. Camenisch, J.and Stadler. Efficient group signature schemes for large groups. *Advances in Cryptology, CRYPTO '97*, pages 410–424, 1997.

[6] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. *Comm. ACM*, 28(10):1030–1044, 1985.

[7] P. Gaudry and É. Schost. Construction of secure random curves of genus 2 over prime fields. In *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 239–256. Springer-Verlag, 2004.

[8] L. Harn. Batch verifying multiple RSA digital signatures. *Electronics Letters*, 34(12):1219–1220, Jun 1998.

[9] J. Li and N. Li. OACerts: Oblivious attribute certificates. *IEEE Transactions on Dependable and Secure Computing*, 3(4):340–352, 2006.

[10] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym systems. In *Proc. Sixth Workshop Selected Areas in Cryptography*, pages 184–199, 1999.

[11] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[12] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 129–140, London, UK, 1992. Springer-Verlag.

[13] N. Shang. G2HEC: A Genus 2 Crypto C++ Library. http://www.math.purdue.edu/~nshang/libg2hec.html.

APPENDIX

A. PROOF OF THEOREM 1

PROOF. We shall show that if there is an adversary $\mathcal{A}$ who wins the game with probability ϵ, we can construct another adversary $\mathcal{B}$ who can either break the group 2nd-preimage resistance property of H_1, or solve the computational Diffie-Hellman problem in G, with the same probability ϵ. Indeed, $\mathcal{B}$ executes the following procedures:

- When given a group G, $h, h^a, h^b \in G$, and $x_1, \ldots, x_n \in \mathbb{Z}/(p)$, $\mathcal{B}$ gives $\mathsf{Param} = \langle G, h^a, h\rangle$ to $\mathcal{A}$. $\mathcal{B}$ also sends $x_1, \ldots, x_n$, and h^b to $\mathcal{A}$. Let $g = h^a$.
- $\mathcal{B}$ receives $y_1, \ldots, y_n, r_1, \ldots, r_n$, and σ' from $\mathcal{A}$, where $\{x_1, \ldots, x_n\} \neq \{y_1, \ldots, y_n\}$.
- $\mathcal{B}$ computes $x = \sum_{i=1}^{n} H_1(x_i)$, $y = \sum_{i=1}^{n} H_1(y_i)$, and checks whether $x = y$. If $x \neq y$, $\mathcal{B}$ computes $r = \sum_{i=1}^{n} r_i$, and outputs
$$\delta = (\sigma'(h^b)^{-r})^{(y-x)^{-1}},$$
where $(y-x)^{-1}$ is the multiplicative inverse of $y-x$ in $\mathbb{Z}/(p)$.

When $\mathcal{A}$ wins the game, we have

$$\begin{aligned}\sigma' &= \left(\left(\prod_{i=1}^{n} g^{H_1(y_i)} h^{r_i}\right) g^{-x}\right)^b \\ &= (g^{y-x} h^r)^b.\end{aligned}$$

If $x = y$, then the group 2nd-preimage resistance property of H_1 fails to hold. Otherwise,

$$\delta = (\sigma'(h^b)^{-r})^{(y-x)^{-1}} = g^b = (h^a)^b = h^{ab},$$

i.e., the computational Diffie-Hellman problem is solved. □

Efficient and Privacy-Preserving Enforcement of Attribute-Based Access Control

Ning Shang [1,3] Federica Paci[1,2] Elisa Bertino[1]

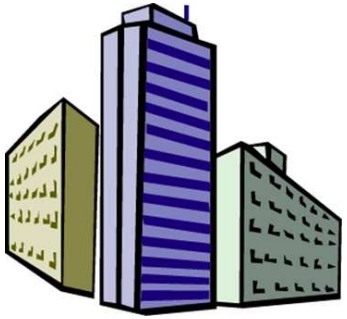
[1]Purdue University, [2]University of Trento, [3]Microsoft

April, 2010

Attribute-based access control - Approach 0

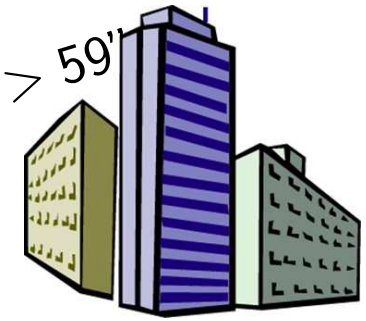
Without privacy

Without privacy

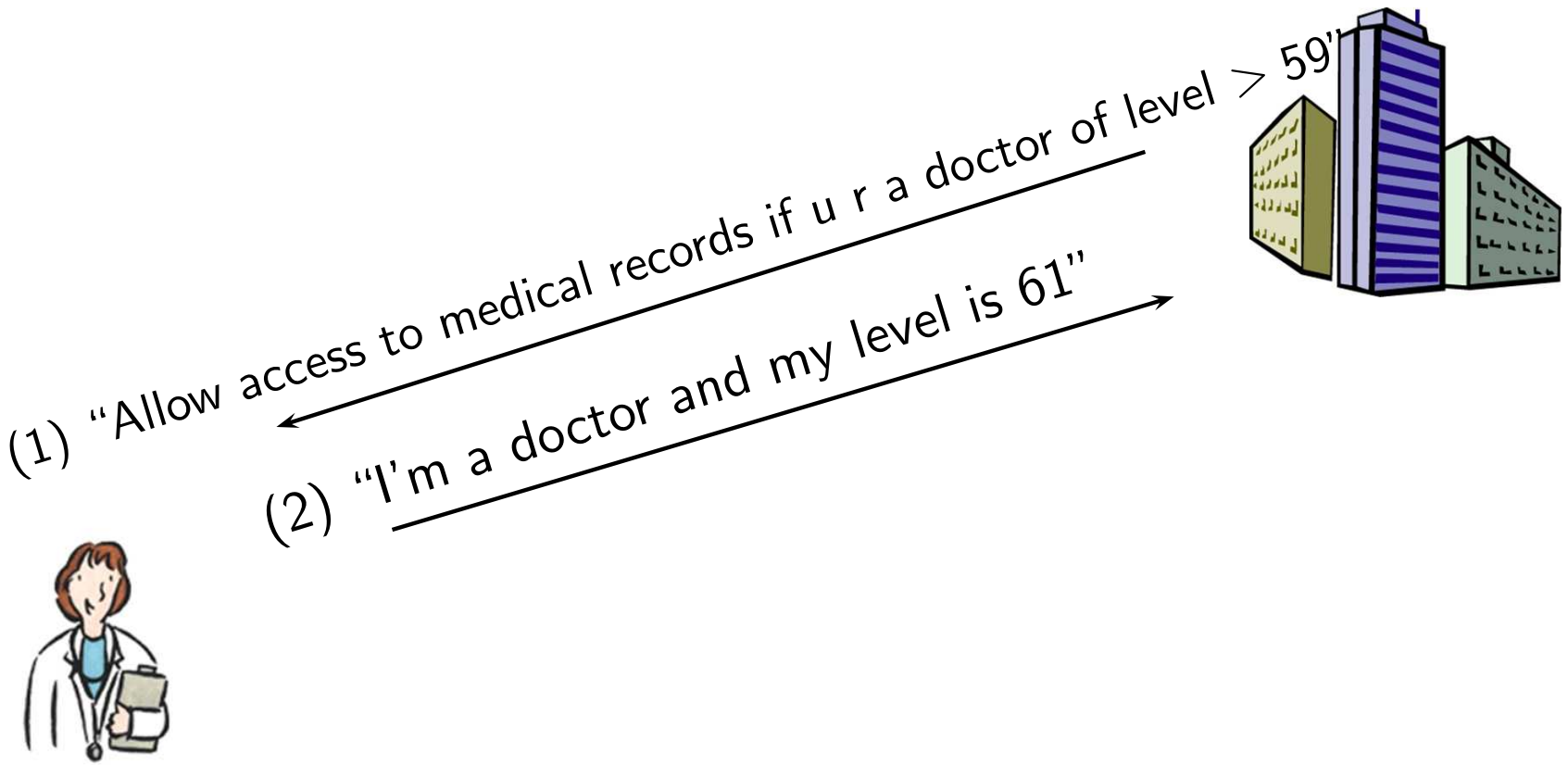


Without privacy

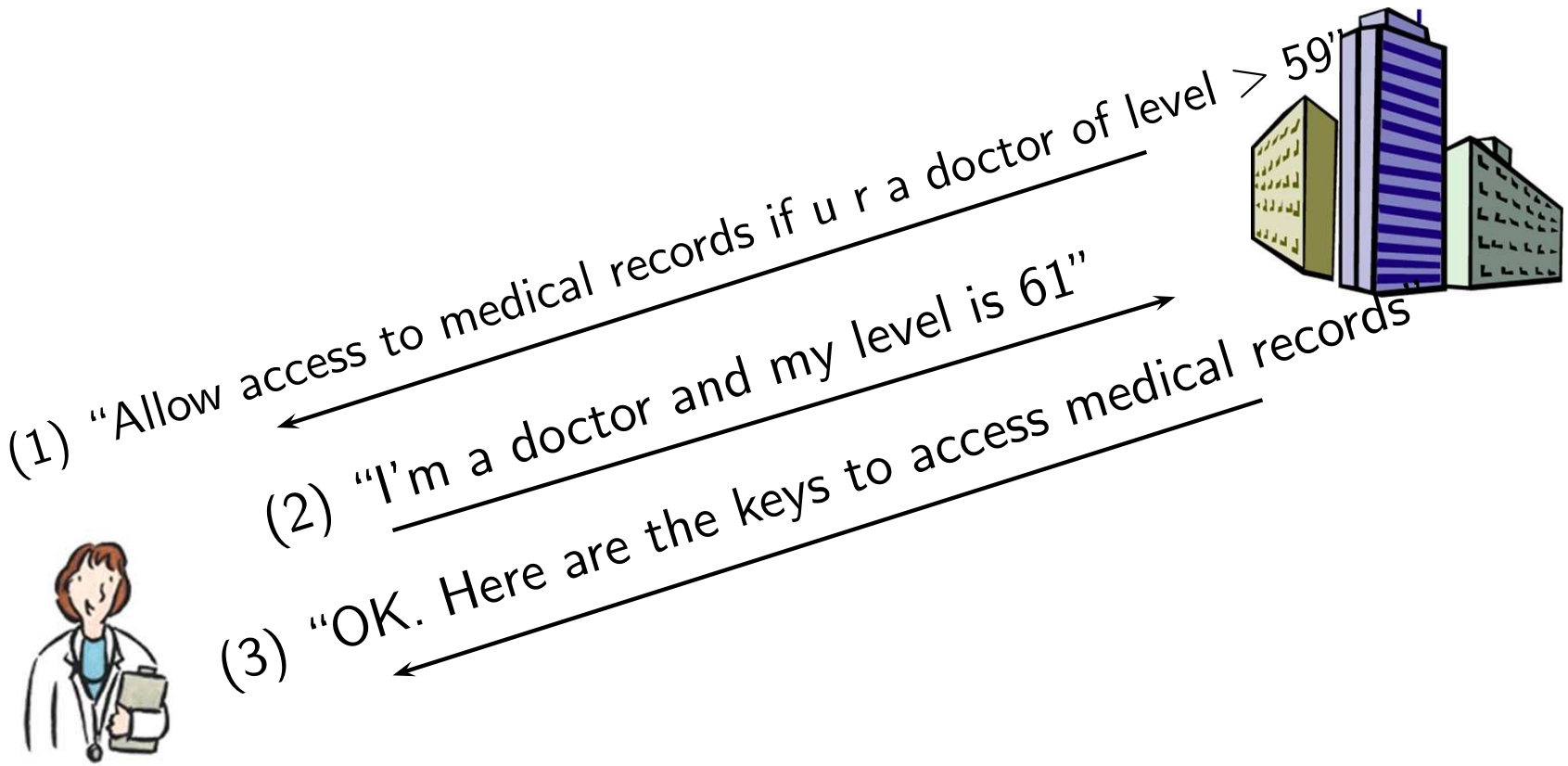
(1) "Allow access to medical records if u r a doctor of level > 59"



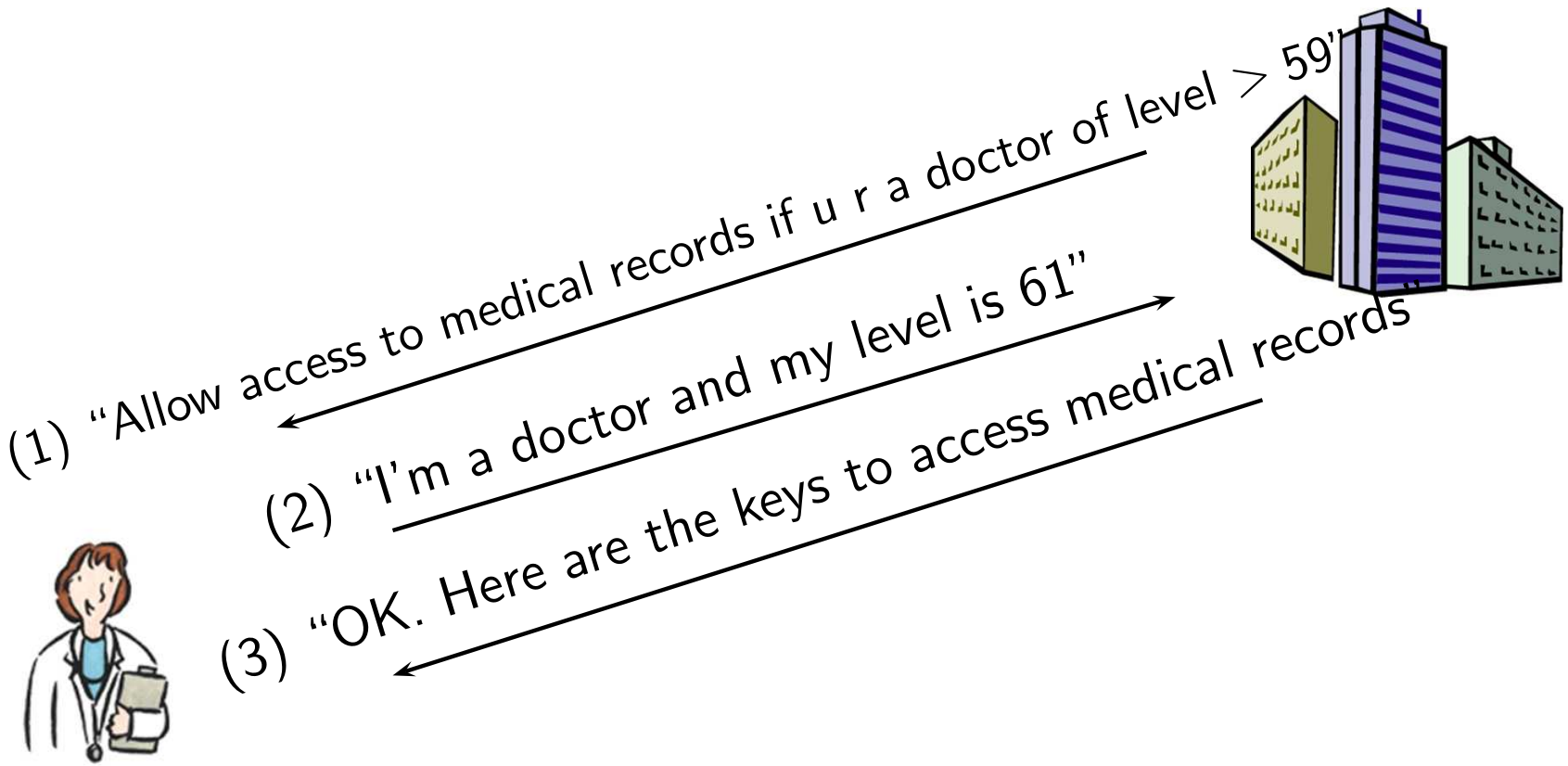
Without privacy



Without privacy



Without privacy

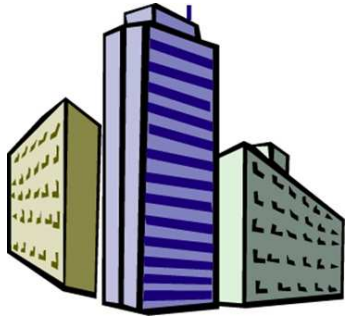


SP knows a lot about user's involved credentials

Attribute-based access control - Approach 1

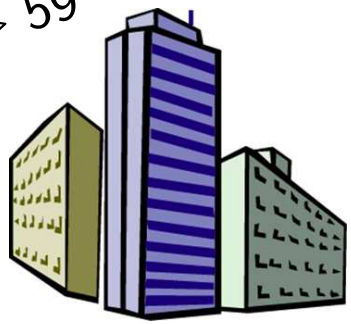
Privacy-preserving via ZKPK

Privacy-preserving via ZKPK

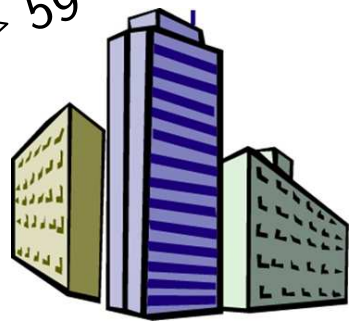


Privacy-preserving via ZKPK

(1) "Allow access to medical records if u can prove u r a doctor of level > 59"



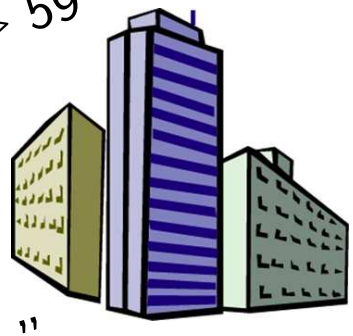
Privacy-preserving via ZKPK



(1) "Allow access to medical records if u can prove u r a doctor of level > 59"

(2) Zero-knowledge proof protocols

Privacy-preserving via ZKPK

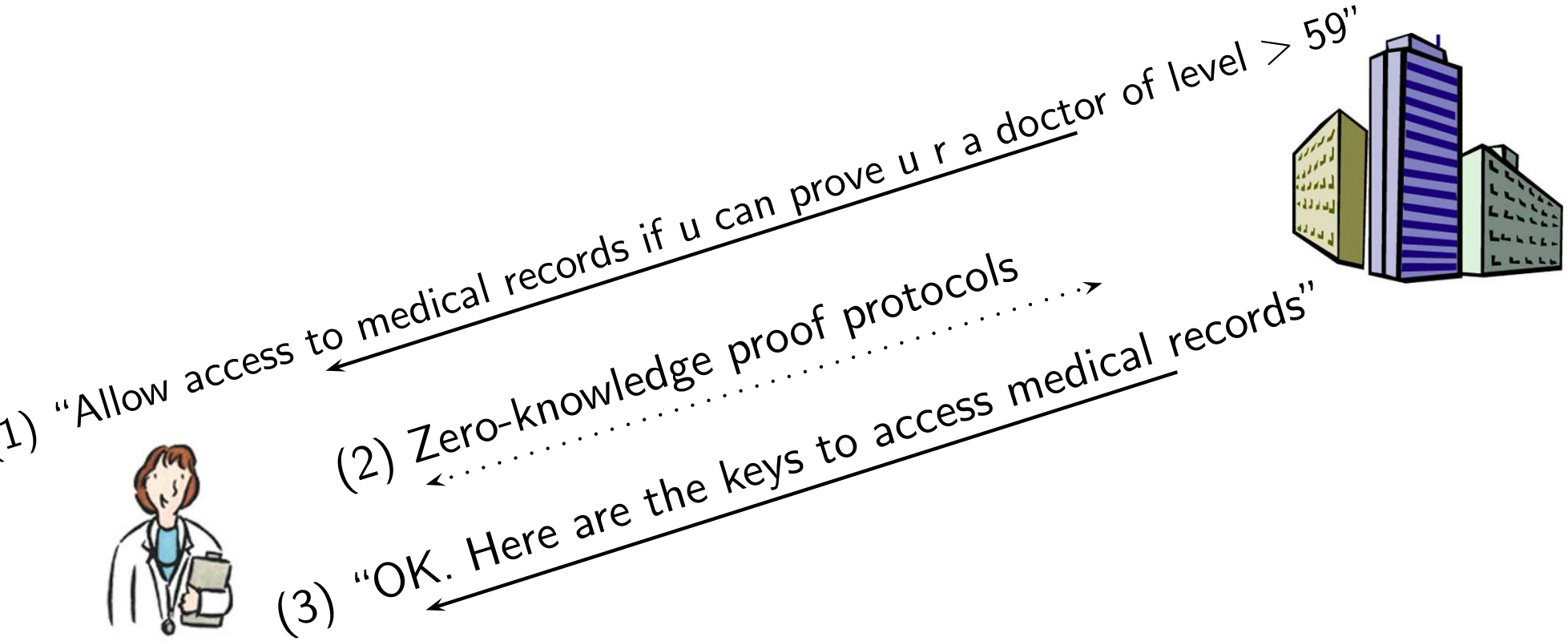


(1) "Allow access to medical records if u can prove u r a doctor of level > 59"

(2) Zero-knowledge proof protocols

(3) "OK. Here are the keys to access medical records"

Privacy-preserving via ZKPK

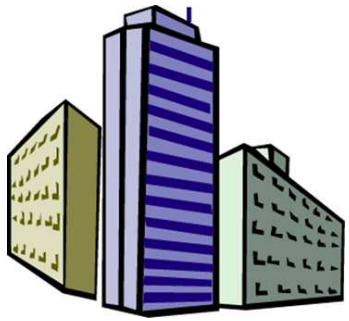


SP knows whether the user's credentials satisfy the requirements or not

Attribute-based access control - Approach 2

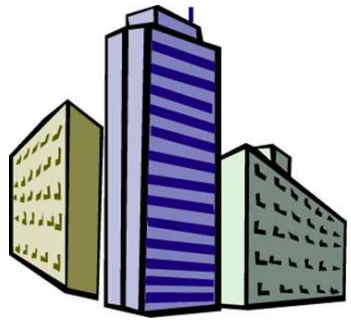
Privacy-preserving via OCBE

Privacy-preserving via OCBE



Privacy-preserving via OCBE

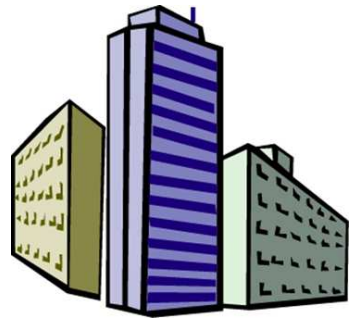
(1) "Allow access to medical records if u r a doctor of level > 59"



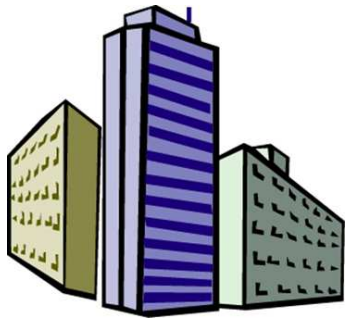
Privacy-preserving via OCBE

(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")



Privacy-preserving via OCBE

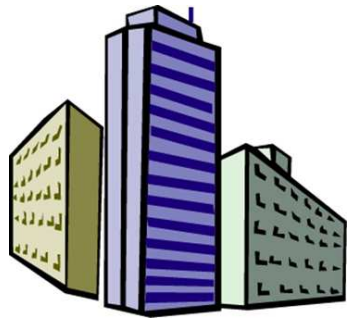


(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")

(3) Envelope(keys to access medical records)

Privacy-preserving via OCBE



(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")

(3) Envelope(keys to access medical records)

User can open the envelope iff its credentials satisfy the policy

SP does not know the outcome of envelope opening

OCBE Overview

OCBE: **O**blivious **C**ommitment-**B**ased **E**nvelope.[1]

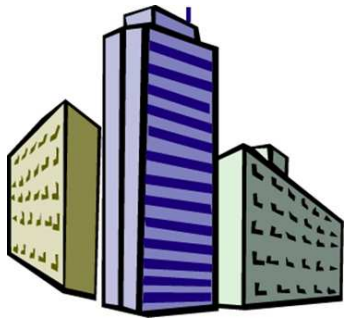
[1] Jiangtao Li and Ninghui Li. OACerts: Oblivious attribute certificates. *IEEE Transactions on Dependable and Secure Computing*, 3(4):340-352, 2006.

OCBE cryptographic building blocks

- $G = \langle g \rangle$: finite cyclic group of order p in which the computationally Diffie-Hellman problem is hard
- Pedersen commitment: $c = g^x h^r$, where $g, h \in G, r \xleftarrow{R} \mathbb{F}_p$
- $\mathcal{E}_K$: symmetric key encryption algorithm with key K
- $H(\cdot)$: cryptographic hash function

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$

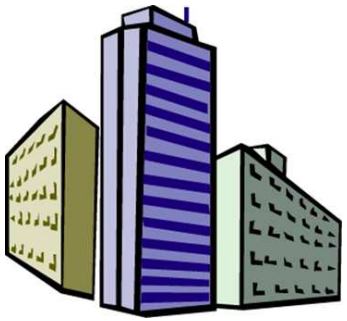


secret r

$c = g^{\mathrm{ID}} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



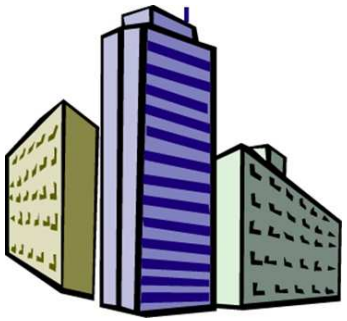
(1) EQ_{x_0}

secret r

$c = g^{ID} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

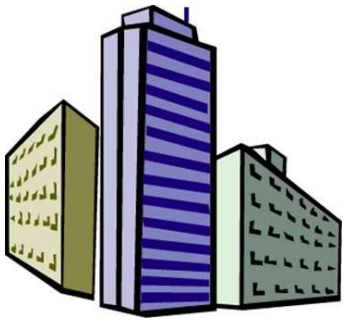
(2) c

secret r

$c = g^{ID} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

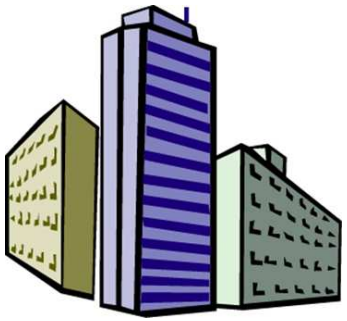
(2) c

secret r
$c = g^{\mathrm{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

(2) c

(4) $\eta = h^y$, $C = \mathcal{E}_{H(\sigma)}[\text{message}]$

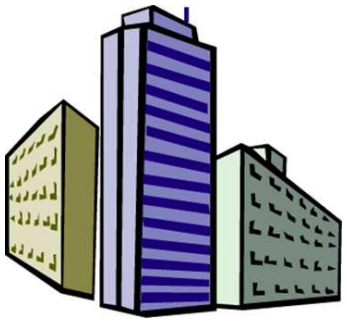
secret r

$c = g^{\text{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

(2) c

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

secret r
$c = g^{\text{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

(5) $\sigma' = \eta^r$, decrypts C with $H(\sigma')$

Other OCBE's

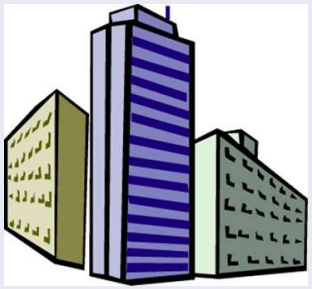
GE-OCBE, LE-OCBE, ... are OCBE protocols for $\geq, \leq, \ldots$ predicates. They are performed in a similar fashion as EQ-OCBE, but generally more expensive.

OCBE features

- Security & privacy: the identity tokens (commitments) are unconditionally hiding and computationally binding
- X.509 integration: the identity tokens can be put into X.509v3 certificate extension fields

Multiple attributes specified in policy

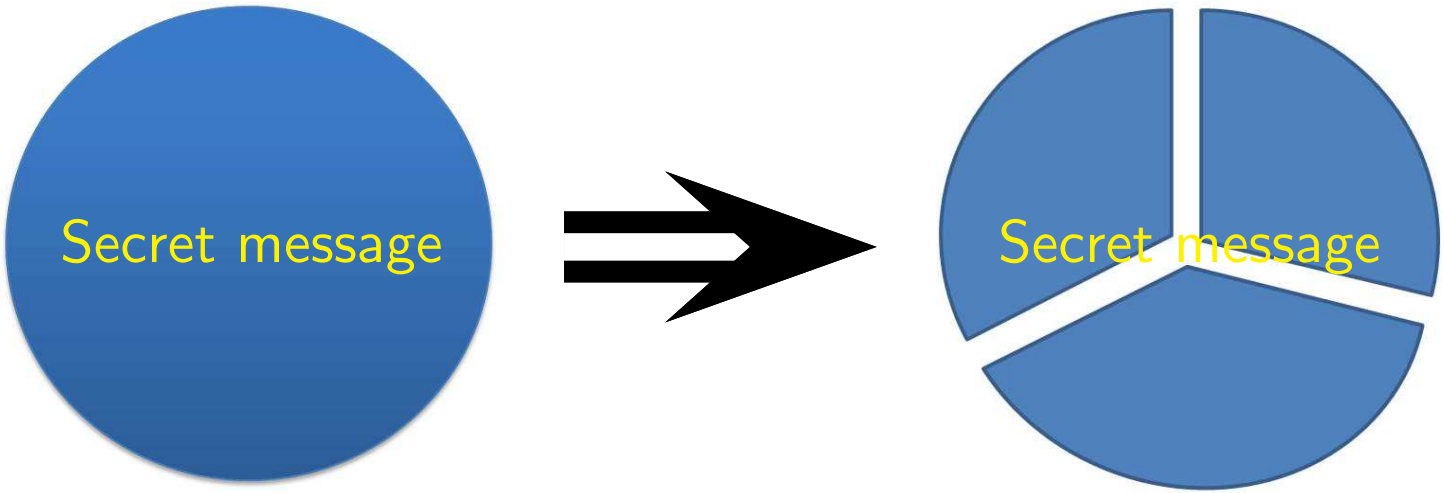
Conjunction of conditions



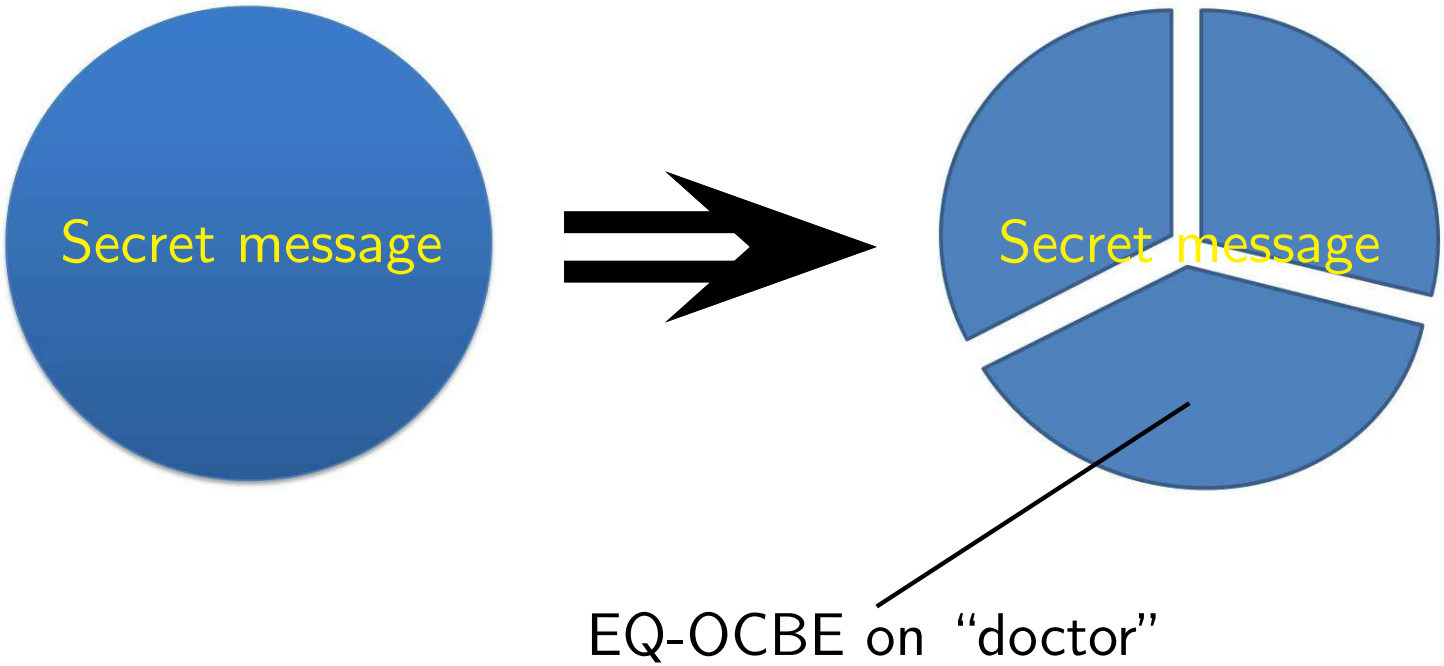
"Allow access if you are a doctor of Hospital A in Indiana"

Multiple attributes: a straightforward solution

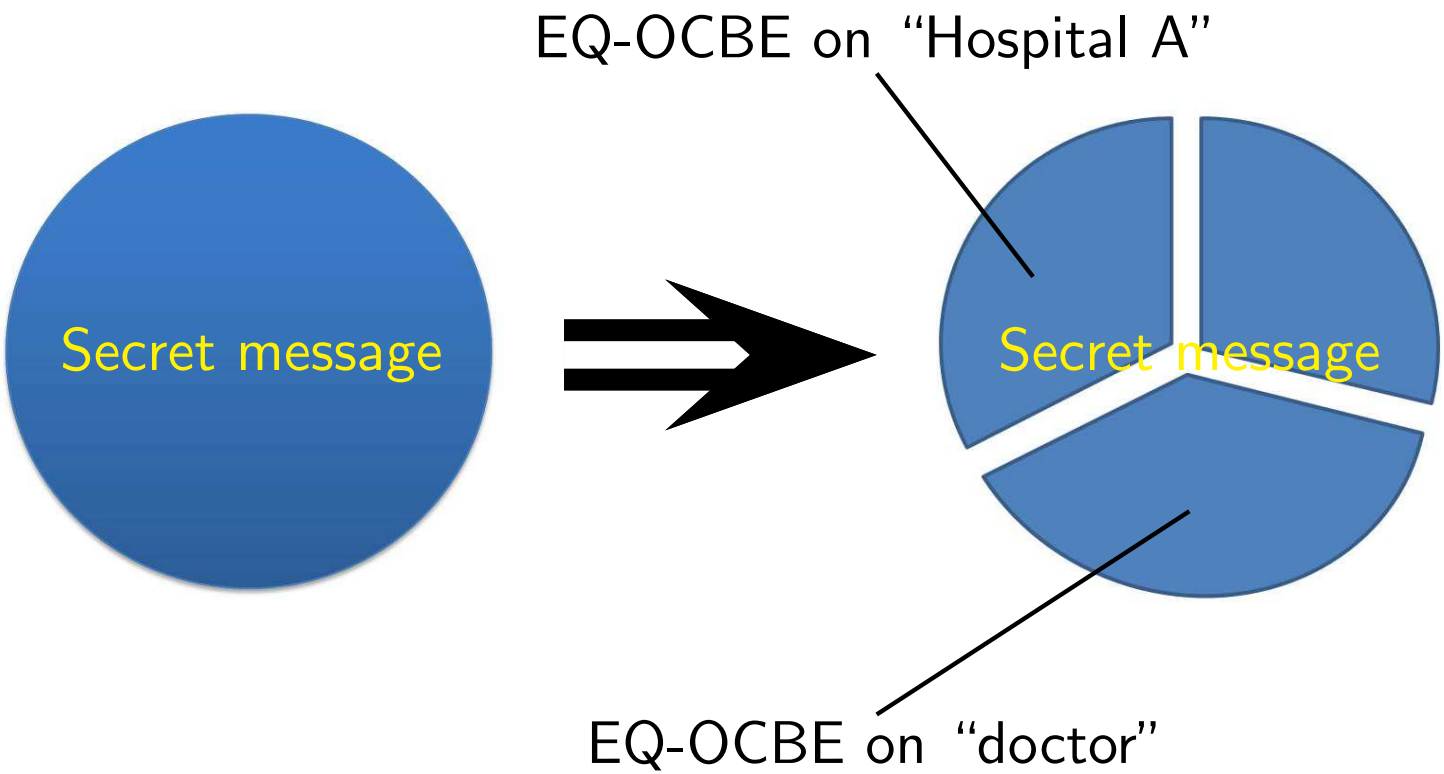
Multiple attributes: a straightforward solution



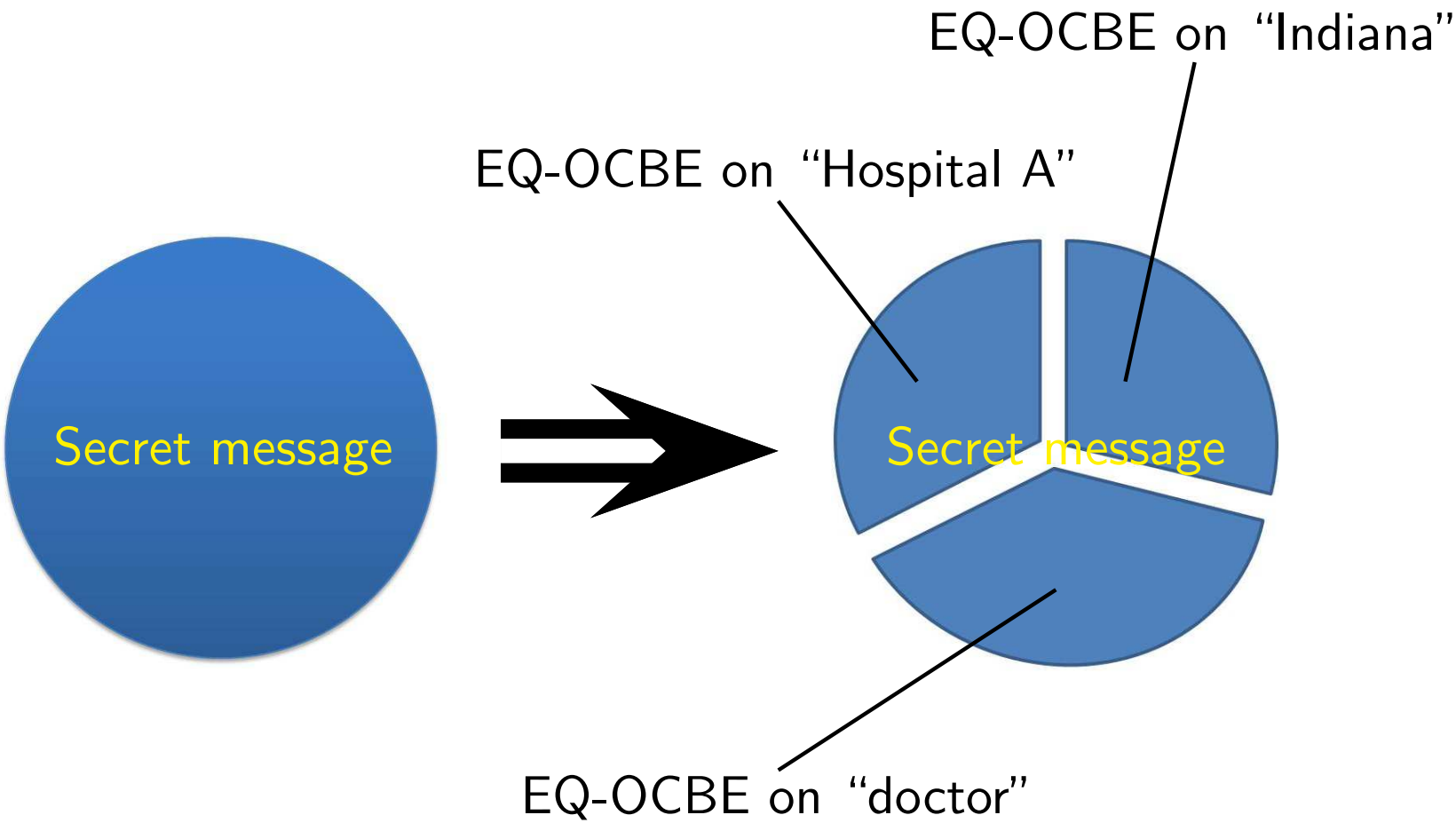
Multiple attributes: a straightforward solution



Multiple attributes: a straightforward solution



Multiple attributes: a straightforward solution



This approach works, but...

It is not very efficient

communication and computation costs increase in proportion to the number of specified attributes

Question

Can we do better?

Answer

Agg-EQ-OCBE:
Aggregate OCBE protocol for equality predicates

- handles multiple equality conditions at the same time, without significantly increasing computational cost
- also requires less bandwidth

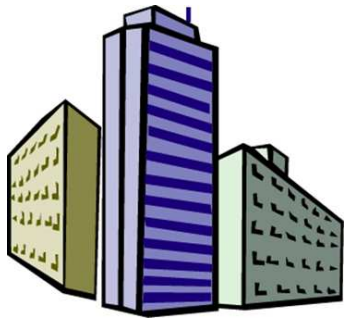
Agg-EQ-OCBE ideas

Techniques to improve the performance

- make use of the algebraic structure and operations in EQ-OCBE
- trade more expensive exponentiation operations for less costly addition and multiplication operations

Agg-EQ-OCBE illustration

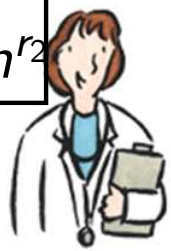
Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2

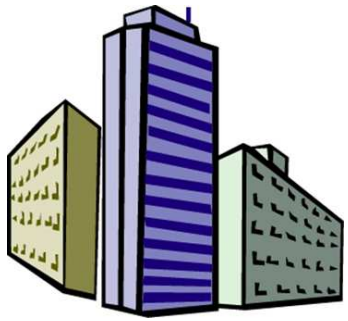
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$

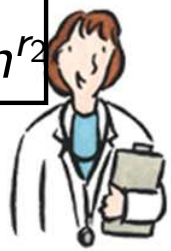


(1) EQ_{x_1}, EQ_{x_2}

secret r_1, r_2

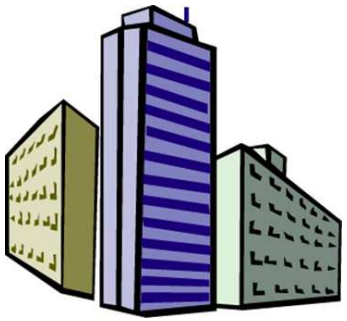
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



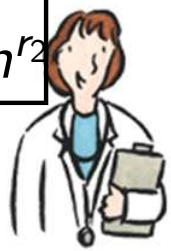
(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

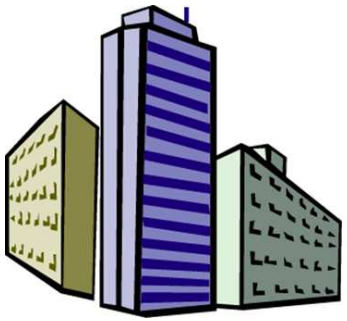
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



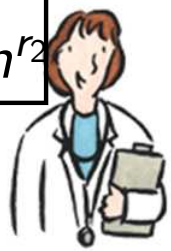
(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



(3) $y \xleftarrow{R} \mathbb{F}_p$,

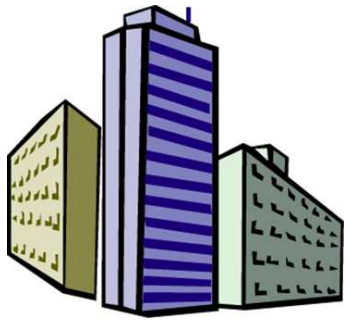
$c = c_1 \cdot c_2$,

$x_0 = x_1 + x_2$,

$\sigma = (cg^{-x_0})^y$

Agg-EQ-OCBE illustration

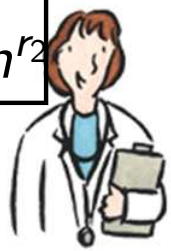
Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2

$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



(1) EQ_{x_1}, EQ_{x_2}

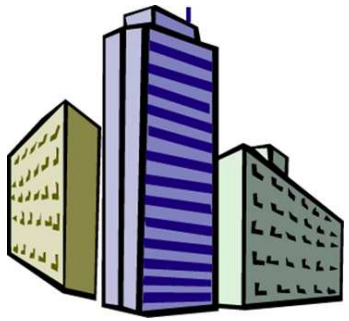
(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

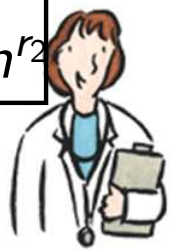
(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = x_1 + x_2$,
$\sigma = (cg^{-x_0})^y$

Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2
$c_1 = g^{21} h^{r_1}$
$c_2 = g^{35} h^{r_2}$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = x_1 + x_2$,
$\sigma = (cg^{-x_0})^y$

(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts C with $H(\sigma')$ to get message

One problem

Collision

Owners of identity token sets

$$S_1 = \{c_1 = g^{21}h^{r_1}, c_2 = g^{35}h^{r_2}\} \text{ and } S_2 = \{c_3 = g^{18}h^{r_3}, c_4 = g^{38}h^{r_4}\}$$

will both open the envelope.

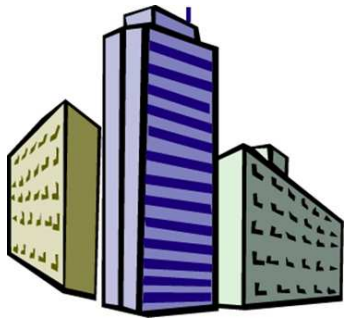
$$\boxed{21 + 35 = 56 = 18 + 38}$$

Solution

Cryptographic hash

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



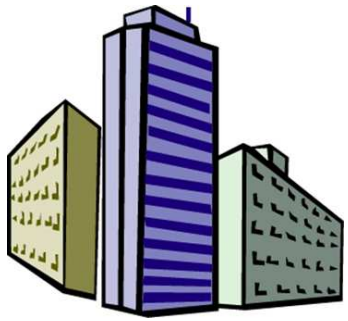
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

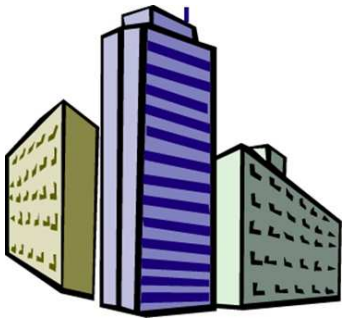
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

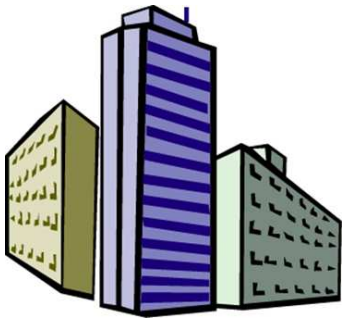
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

(3) $y \xleftarrow{R} \mathbb{F}_p$,

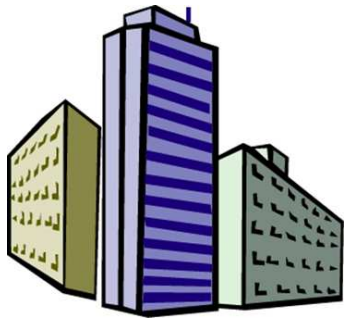
$c = c_1 \cdot c_2$,

$x_0 = H_1(x_1) + H_1(x_2)$,

$\sigma = (cg^{-x_0})^y$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

(1) EQ_{x_1}, EQ_{x_2}

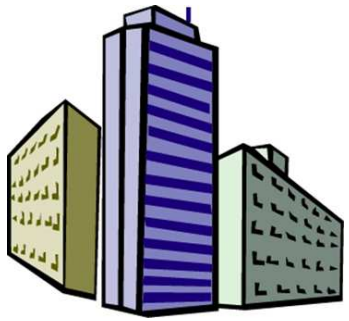
(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = H_1(x_1) + H_1(x_2)$,
$\sigma = (cg^{-x_0})^y$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

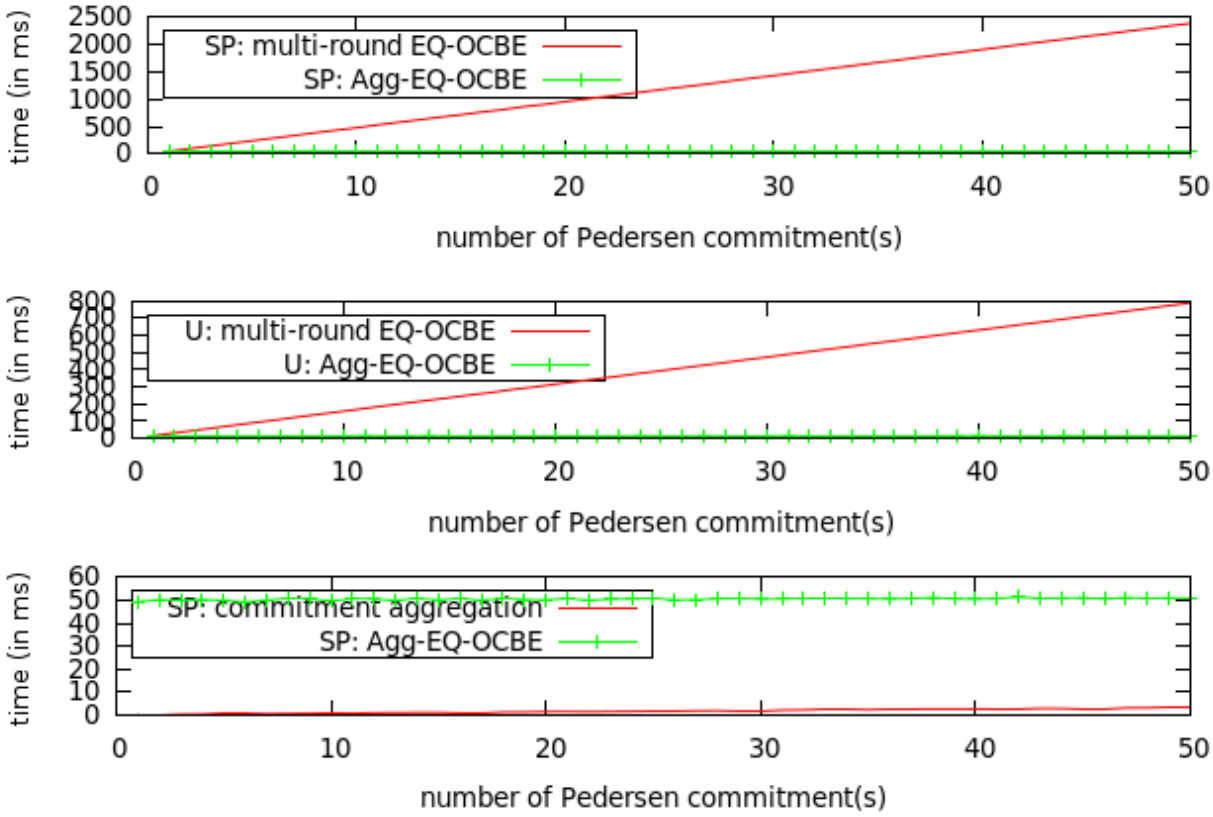
(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = H_1(x_1) + H_1(x_2)$,
$\sigma = (cg^{-x_0})^y$

(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts C with $H(\sigma')$ to get message

Underlying intractability assumptions

- **Group 2nd-preimage resistant hash $\widetilde{H}(\cdot)$**
  Given $(x_1, \ldots, x_m)$, it is hard to find another tuple $(y_1, \ldots, y_n)$ such that
  $$\sum_{i=1}^{m} \widetilde{H}(x_i) = \sum_{i=1}^{n} \widetilde{H}(y_i)$$
- **Computational Diffie-Hellman problem**
  Given g^a, g^b, it is hard to compute g^{ab}, without knowing a and b.

Experimental results



Future work

- **More application scenarios**
- **Aggregate GE-OCBE and other OCBE protocols**
  aggregation works in certain cases, e.g., when sum of attribute values needs to be $\geq$ a threshold value

Summary

- Privacy-preserving attribute-based access control concepts and approaches
- OCBE overview
- Aggregate EQ-OCBE
- Experimental data

The End

Thank you!

Questions?

nshang@cs.purdue.edu

Efficient and Privacy-Preserving Enforcement of Attribute-Based Access Control

Ning Shang [1,3] Federica Paci[1,2] Elisa Bertino[1]

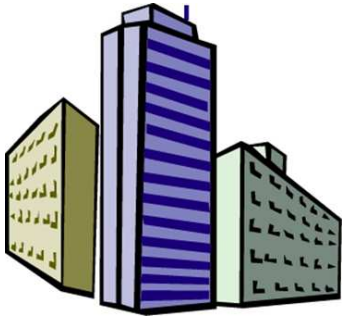
[1]Purdue University, [2]University of Trento, [3]Microsoft

April, 2010

Attribute-based access control - Approach 0

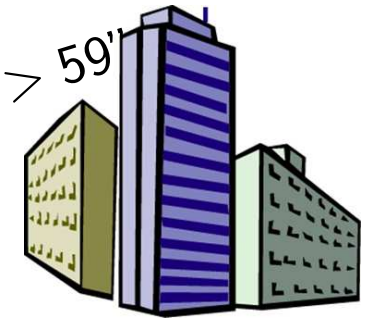
Without privacy

Without privacy

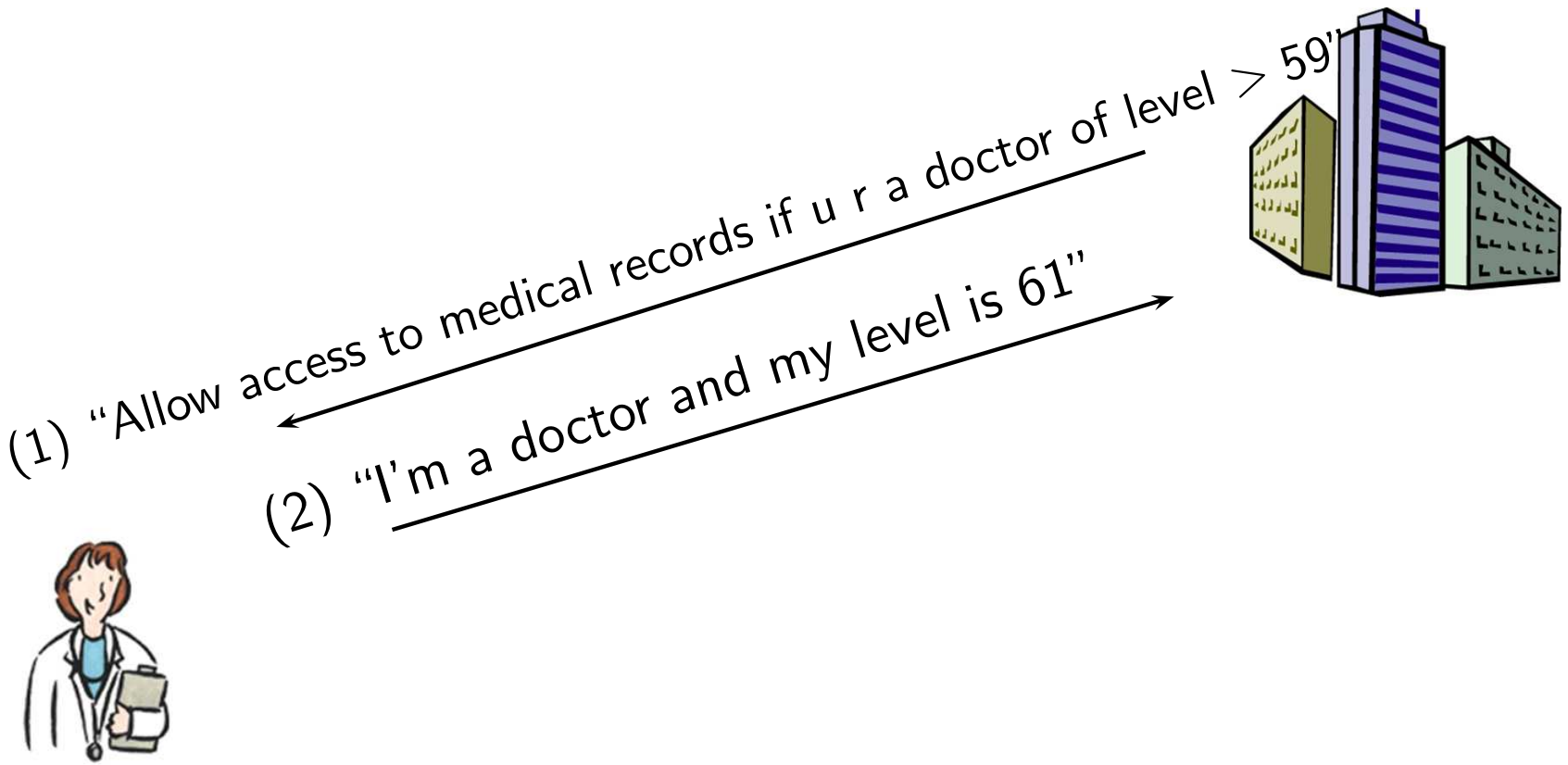


Without privacy

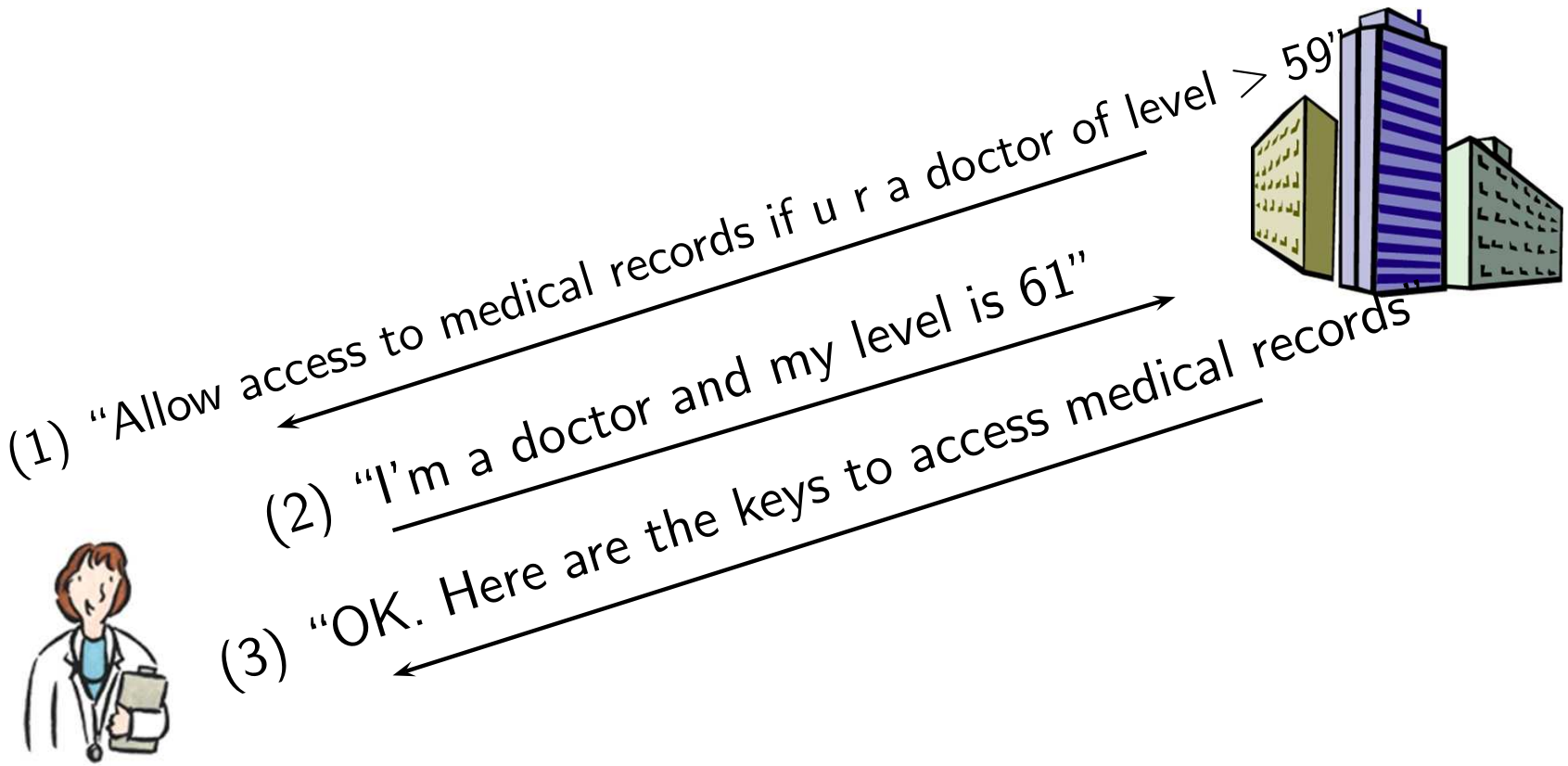
(1) "Allow access to medical records if u r a doctor of level > 59"



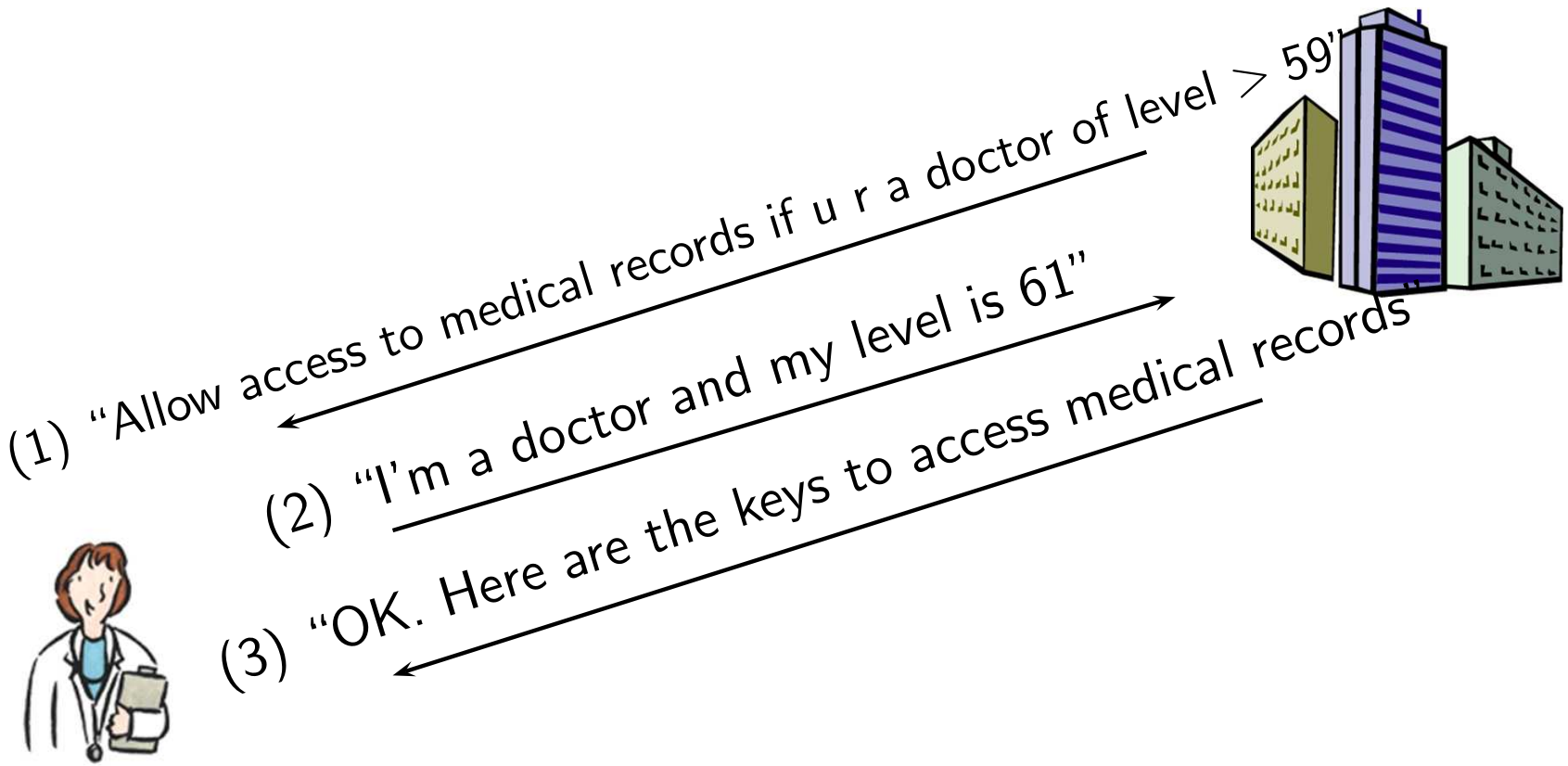
Without privacy



Without privacy



Without privacy

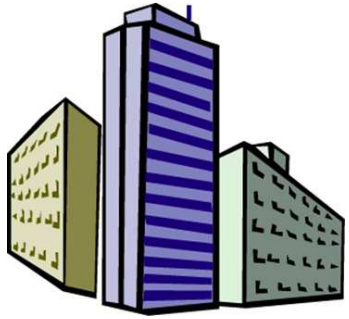


SP knows a lot about user's involved credentials

Attribute-based access control - Approach 1

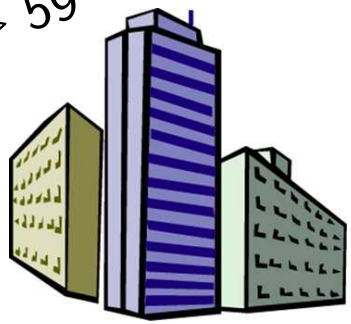
Privacy-preserving via ZKPK

Privacy-preserving via ZKPK

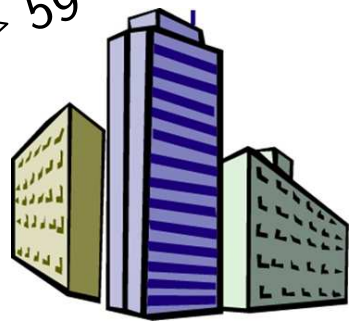


Privacy-preserving via ZKPK

(1) "Allow access to medical records if u can prove u r a doctor of level > 59"



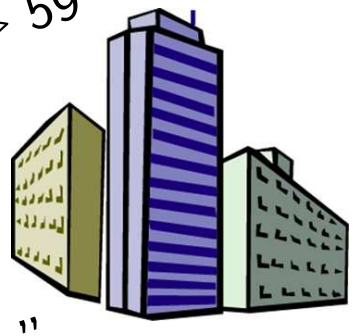
Privacy-preserving via ZKPK



(1) "Allow access to medical records if u can prove u r a doctor of level > 59"

(2) Zero-knowledge proof protocols

Privacy-preserving via ZKPK

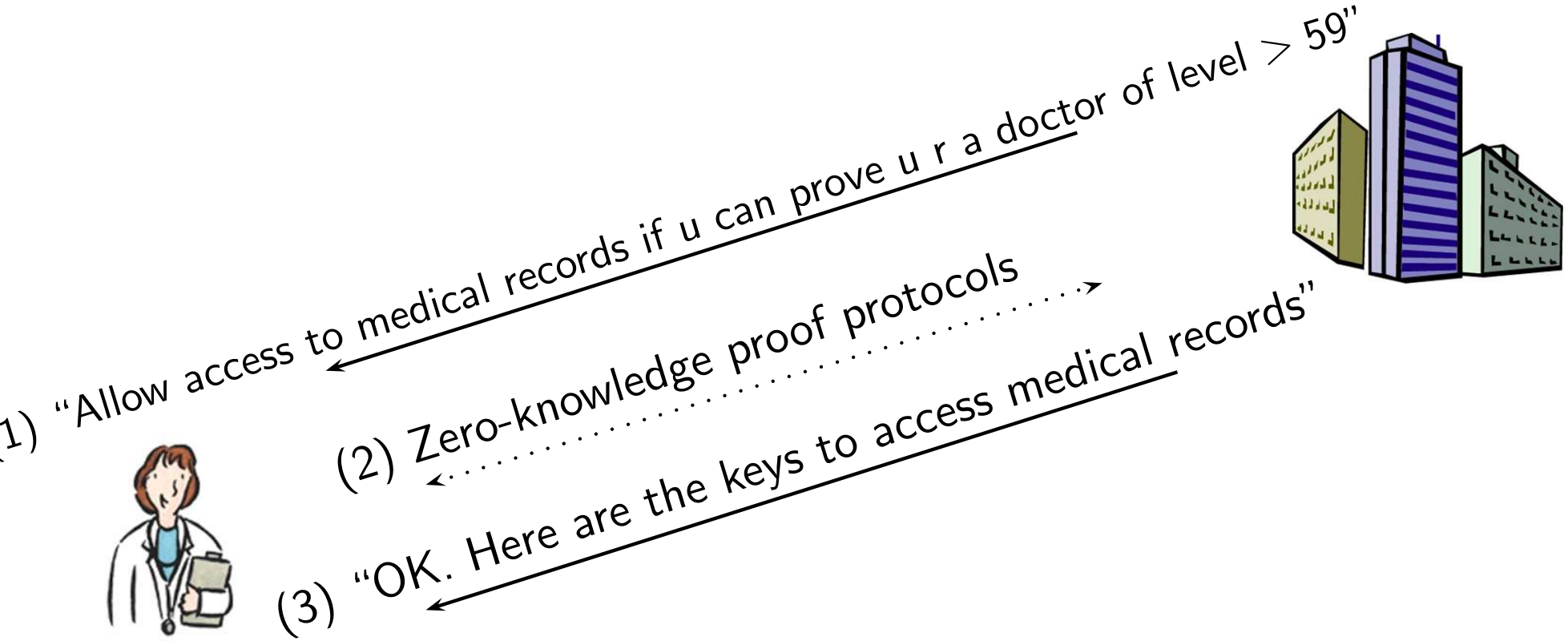


(1) "Allow access to medical records if u can prove u r a doctor of level > 59"

(2) Zero-knowledge proof protocols

(3) "OK. Here are the keys to access medical records"

Privacy-preserving via ZKPK

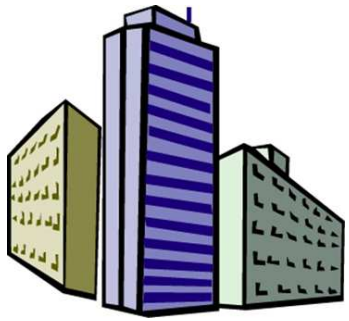


SP knows whether the user's credentials satisfy the requirements or not

Attribute-based access control - Approach 2

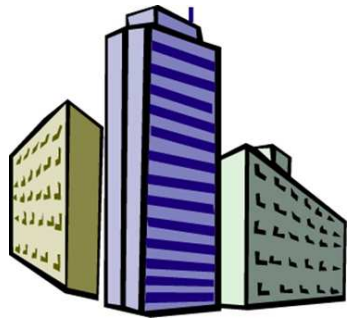
Privacy-preserving via OCBE

Privacy-preserving via OCBE



Privacy-preserving via OCBE

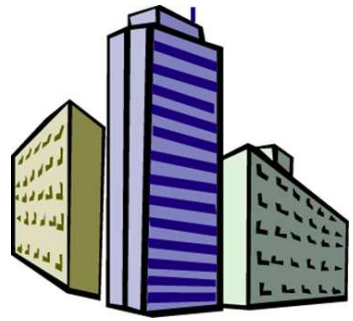
(1) "Allow access to medical records if u r a doctor of level > 59"



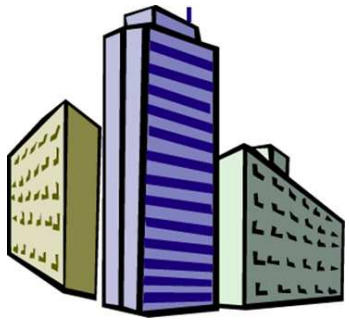
Privacy-preserving via OCBE

(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")



Privacy-preserving via OCBE

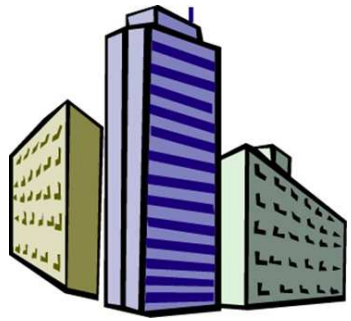


(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")

(3) Envelope(keys to access medical records)

Privacy-preserving via OCBE



(1) "Allow access to medical records if u r a doctor of level $>$ 59"

(2) Commitments("I'm a doctor", "level $=$61")

(3) Envelope(keys to access medical records)

User can open the envelope iff its credentials satisfy the policy

SP does not know the outcome of envelope opening

OCBE Overview

OCBE: **O**blivious **C**ommitment-**B**ased **E**nvelope.[1]

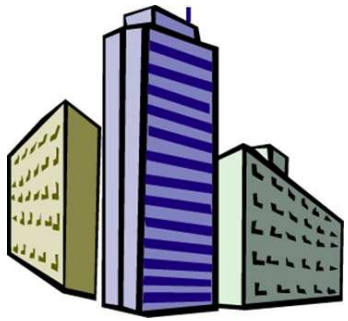
[1] Jiangtao Li and Ninghui Li. OACerts: Oblivious attribute certificates. *IEEE Transactions on Dependable and Secure Computing*, 3(4):340-352, 2006.

OCBE cryptographic building blocks

- $G = \langle g \rangle$: finite cyclic group of order p in which the computationally Diffie-Hellman problem is hard
- Pedersen commitment: $c = g^x h^r$, where $g, h \in G, r \xleftarrow{R} \mathbb{F}_p$
- $\mathcal{E}_K$: symmetric key encryption algorithm with key K
- $H(\cdot)$: cryptographic hash function

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$

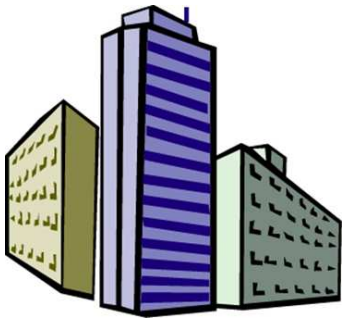


secret r

$c = g^{\mathrm{ID}} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



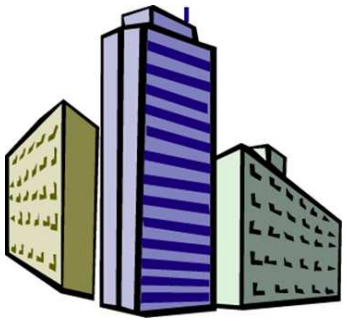
(1) EQ_{x_0}

secret r

$c = g^{ID} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

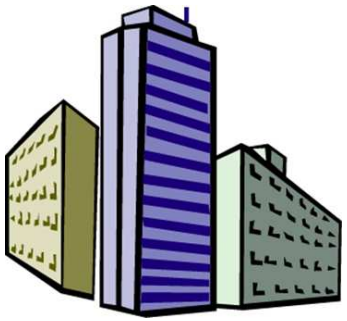
(2) c

secret r

$c = g^{ID} h^r$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

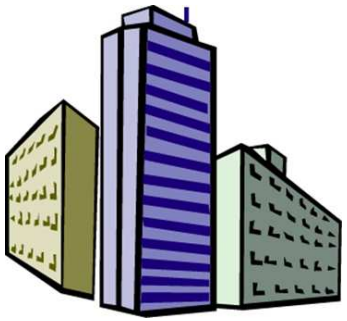
(2) c

secret r
$c = g^{\mathrm{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

(2) c

(4) $\eta = h^y$, $C = \mathcal{E}_{H(\sigma)}[\text{message}]$

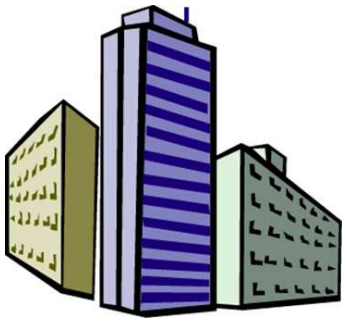
secret r

$c = g^{\text{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

EQ-OCBE: equality predicate

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, $H(\cdot)$



(1) EQ_{x_0}

(2) c

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

secret r
$c = g^{\text{ID}} h^r$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$\sigma = (cg^{-x_0})^y$

(5) $\sigma' = \eta^r$, decrypts C with $H(\sigma')$

Other OCBE's

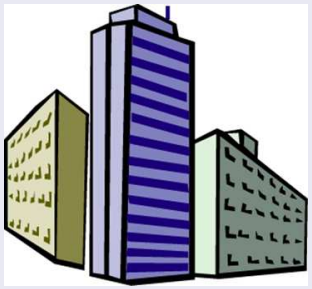
GE-OCBE, LE-OCBE, ... are OCBE protocols for $\geq, \leq, \ldots$ predicates. They are performed in a similar fashion as EQ-OCBE, but generally more expensive.

OCBE features

- Security & privacy: the identity tokens (commitments) are unconditionally hiding and computationally binding
- X.509 integration: the identity tokens can be put into X.509v3 certificate extension fields

Multiple attributes specified in policy

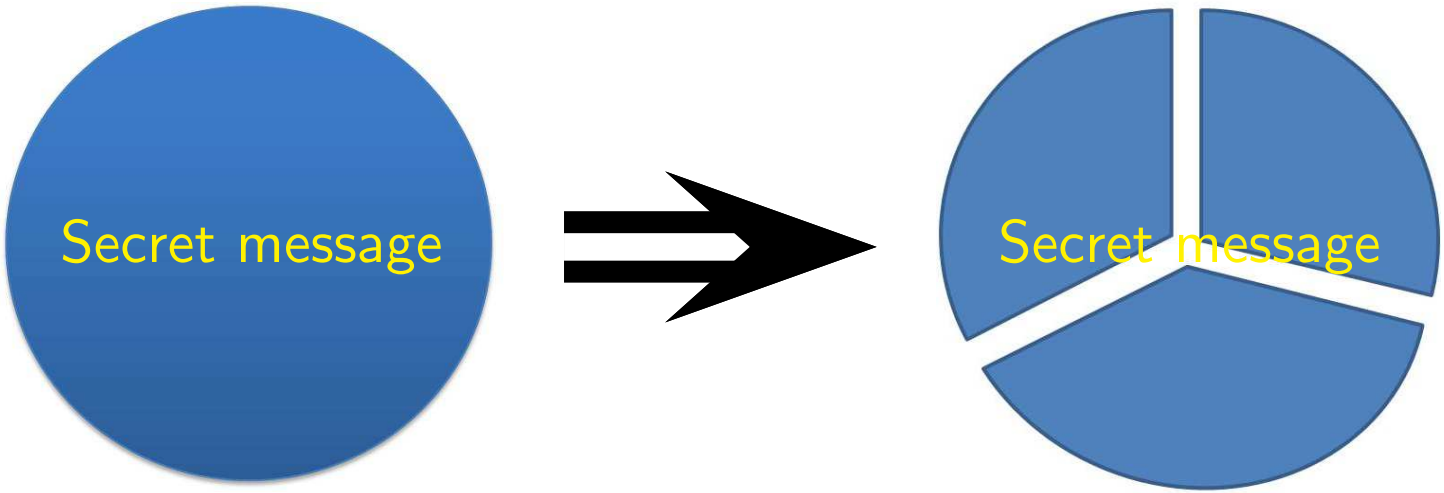
Conjunction of conditions



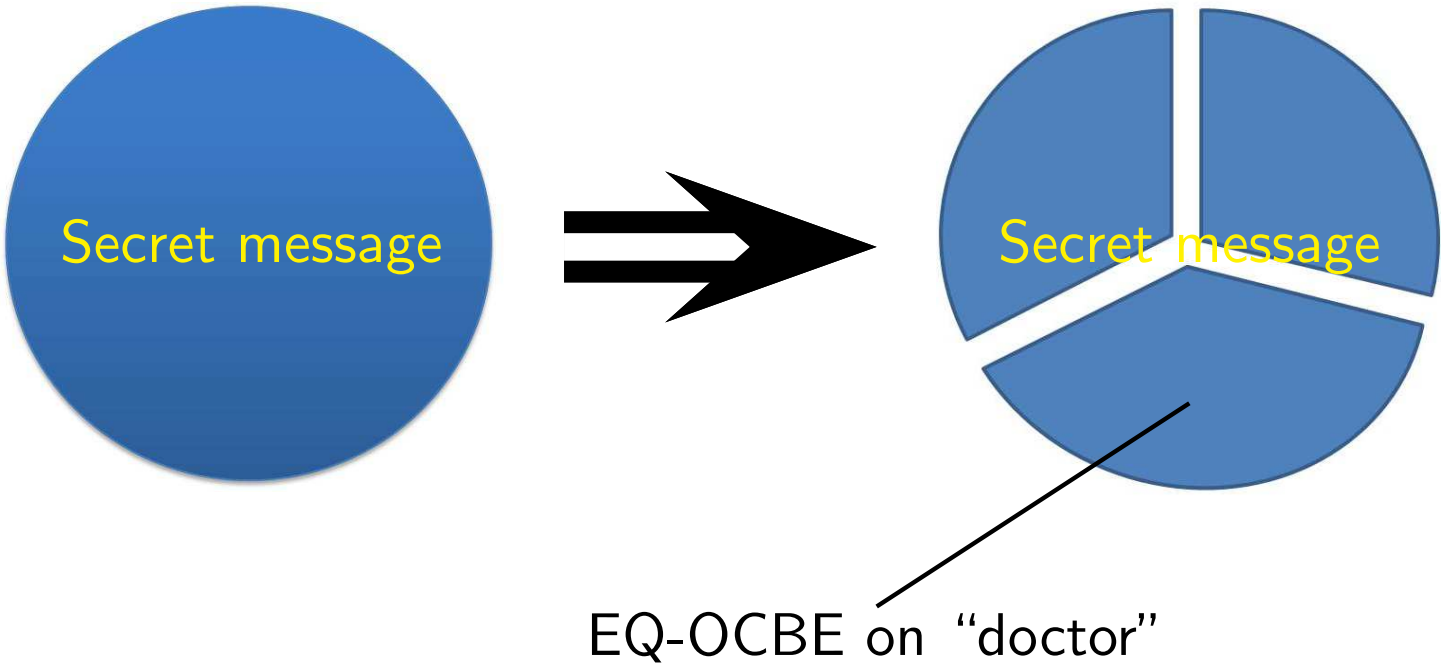
"Allow access if you are a doctor of Hospital A in Indiana"

Multiple attributes: a straightforward solution

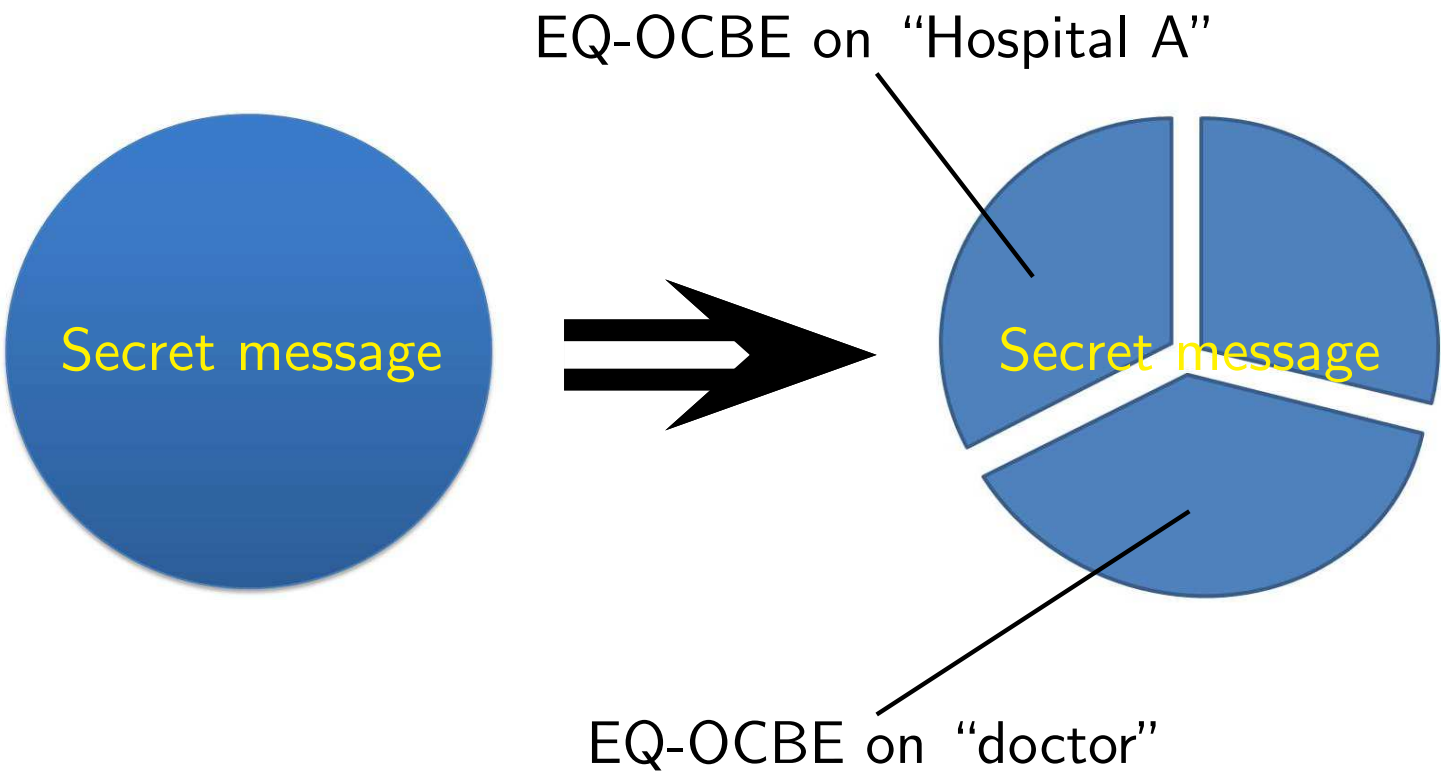
Multiple attributes: a straightforward solution



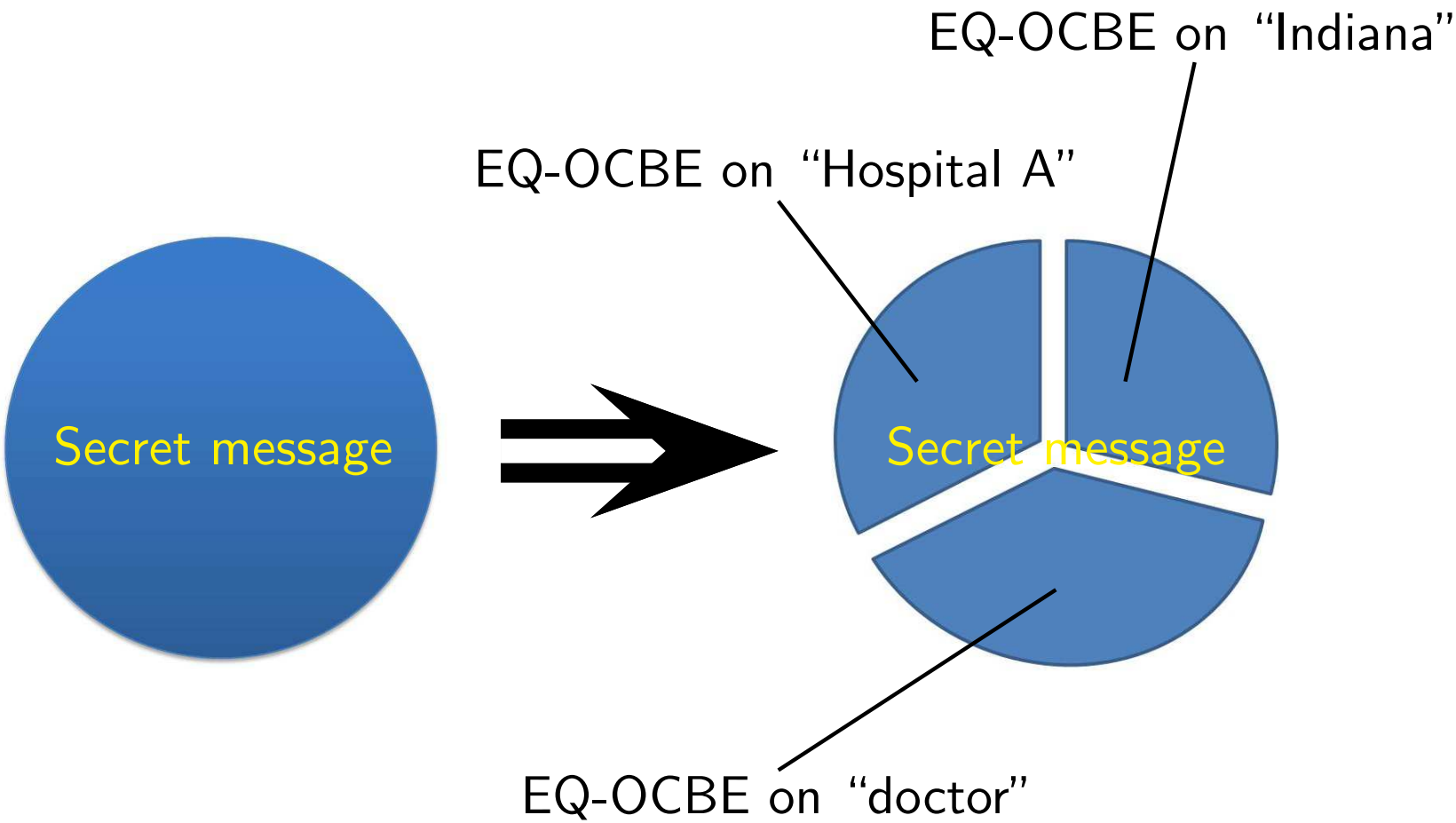
Multiple attributes: a straightforward solution



Multiple attributes: a straightforward solution



Multiple attributes: a straightforward solution



This approach works, but...

It is not very efficient

communication and computation costs increase in proportion to the number of specified attributes

Question

Can we do better?

Answer

Agg-EQ-OCBE:
Aggregate OCBE protocol for equality predicates

- handles multiple equality conditions at the same time, without significantly increasing computational cost
- also requires less bandwidth

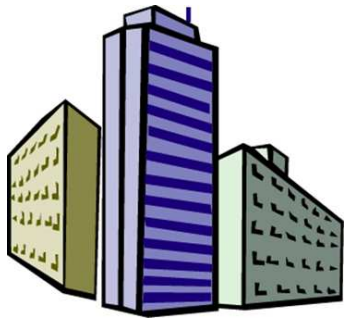
Agg-EQ-OCBE ideas

Techniques to improve the performance

- make use of the algebraic structure and operations in EQ-OCBE
- trade more expensive exponentiation operations for less costly addition and multiplication operations

Agg-EQ-OCBE illustration

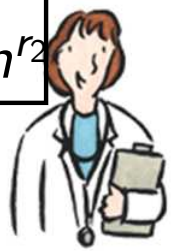
Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2

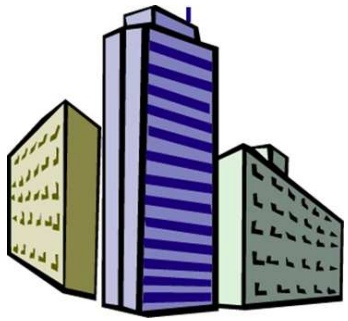
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$

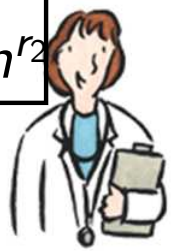


(1) EQ_{x_1}, EQ_{x_2}

secret r_1, r_2

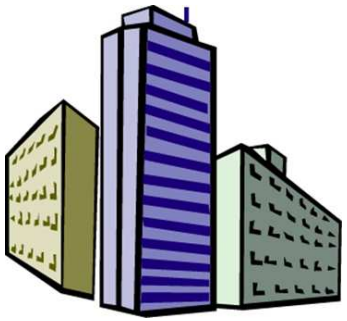
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



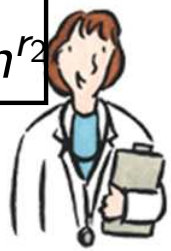
(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

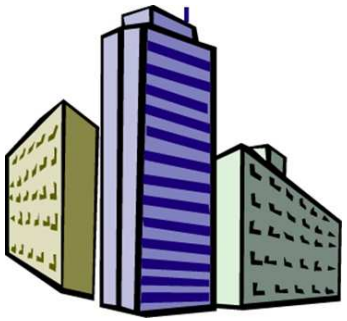
$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



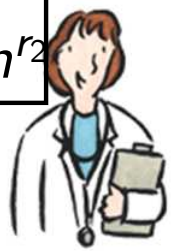
(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



(3) $y \xleftarrow{R} \mathbb{F}_p$,

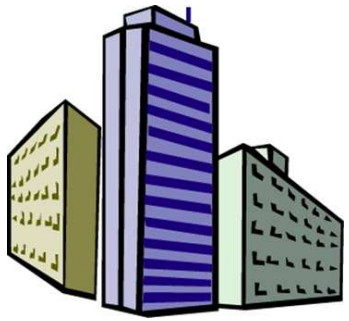
$c = c_1 \cdot c_2$,

$x_0 = x_1 + x_2$,

$\sigma = (cg^{-x_0})^y$

Agg-EQ-OCBE illustration

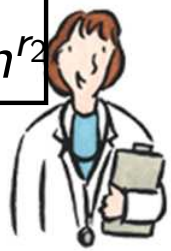
Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2

$c_1 = g^{21} h^{r_1}$

$c_2 = g^{35} h^{r_2}$



(1) EQ_{x_1}, EQ_{x_2}

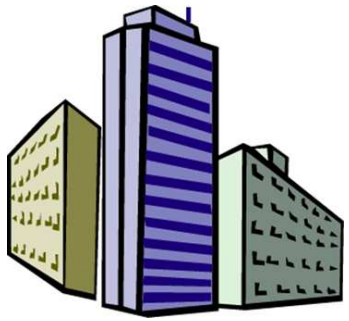
(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

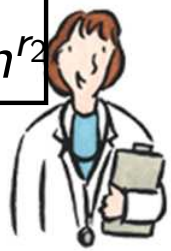
(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = x_1 + x_2$,
$\sigma = (cg^{-x_0})^y$

Agg-EQ-OCBE illustration

Public Param $= \langle G, p, g, h \rangle, \mathcal{E}, H(\cdot)$



secret r_1, r_2
$c_1 = g^{21} h^{r_1}$
$c_2 = g^{35} h^{r_2}$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = x_1 + x_2$,
$\sigma = (cg^{-x_0})^y$

(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts C with $H(\sigma')$ to get message

One problem

Collision

Owners of identity token sets

$$S_1 = \{c_1 = g^{21}h^{r_1}, c_2 = g^{35}h^{r_2}\} \text{ and } S_2 = \{c_3 = g^{18}h^{r_3}, c_4 = g^{38}h^{r_4}\}$$

will both open the envelope.

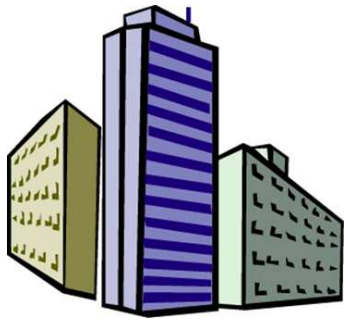
$$\boxed{21 + 35 = 56 = 18 + 38}$$

Solution

Cryptographic hash

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



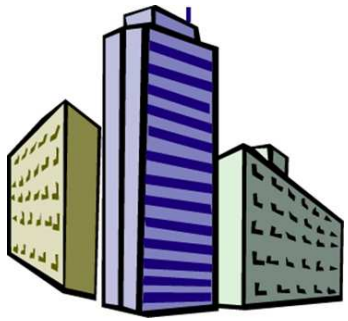
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

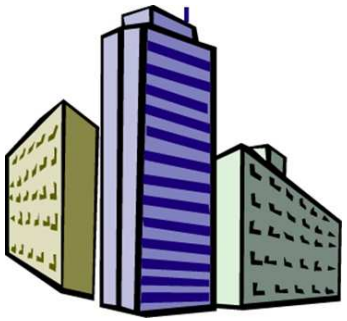
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

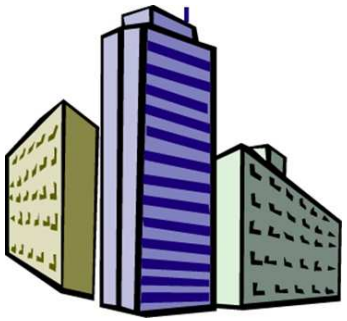
secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

(3) $y \xleftarrow{R} \mathbb{F}_p$,

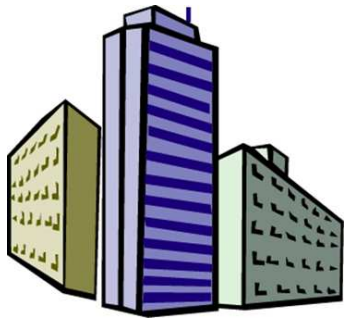
$c = c_1 \cdot c_2$,

$x_0 = H_1(x_1) + H_1(x_2)$,

$\sigma = (cg^{-x_0})^y$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

(1) EQ_{x_1}, EQ_{x_2}

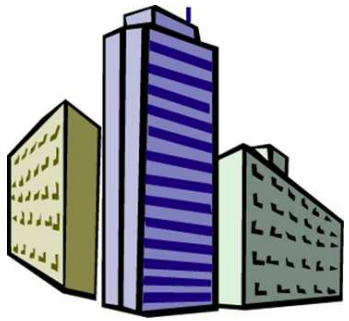
(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = H_1(x_1) + H_1(x_2)$,
$\sigma = (cg^{-x_0})^y$

Aggregate EQ-OCBE

Public Param $= \langle G, p, g, h \rangle$, $\mathcal{E}$, H, $H_1(\cdot)$



(1) EQ_{x_1}, EQ_{x_2}

(2) c_1, c_2

(4) $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{message}]$

secret r_1, r_2

$c_1 = g^{H_1(21)} h^{r_1}$

$c_2 = g^{H_1(35)} h^{r_2}$

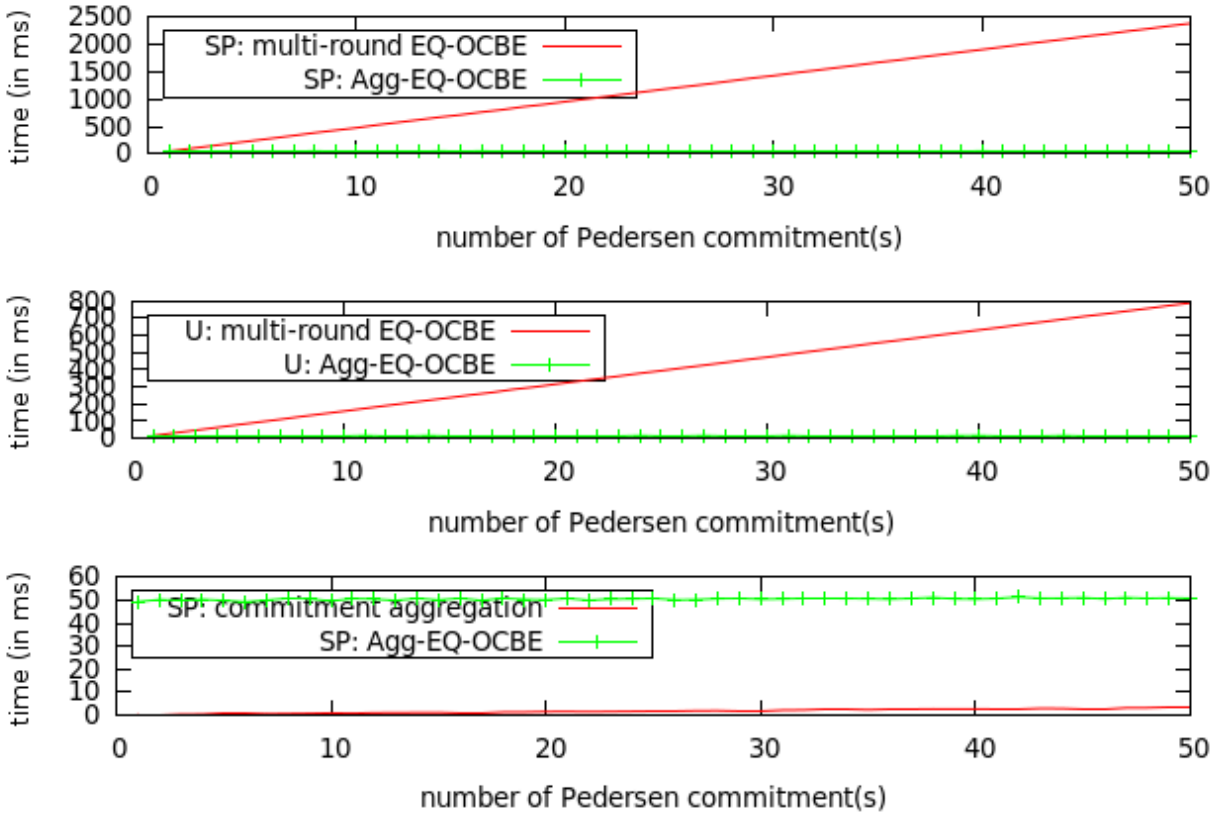
(3) $y \xleftarrow{R} \mathbb{F}_p$,
$c = c_1 \cdot c_2$,
$x_0 = H_1(x_1) + H_1(x_2)$,
$\sigma = (cg^{-x_0})^y$

(5) $r = r_1 + r_2, \sigma' = \eta^r$, decrypts C with $H(\sigma')$ to get message

Underlying intractability assumptions

- **Group 2nd-preimage resistant hash $\widetilde{H}(\cdot)$**
  Given $(x_1, \ldots, x_m)$, it is hard to find another tuple $(y_1, \ldots, y_n)$ such that
  $$\sum_{i=1}^{m} \widetilde{H}(x_i) = \sum_{i=1}^{n} \widetilde{H}(y_i)$$
- **Computational Diffie-Hellman problem**
  Given g^a, g^b, it is hard to compute g^{ab}, without knowing a and b.

Experimental results



Future work

- **More application scenarios**
- **Aggregate GE-OCBE and other OCBE protocols**
  aggregation works in certain cases, e.g., when sum of attribute values needs to be $\geq$ a threshold value

Summary

- Privacy-preserving attribute-based access control concepts and approaches
- OCBE overview
- Aggregate EQ-OCBE
- Experimental data

The End

Thank you!

Questions?

nshang@cs.purdue.edu

Privacy-Preserving DRM

Radia Perlman
Intel
radia@alum.mit.edu

Charlie Kaufman
Microsoft
charliek@microsoft.com

Ray Perlner
NIST
ray.perlner@nist.gov

ABSTRACT

This paper describes and contrasts two families of schemes that enable a user to purchase digital content without revealing to anyone what item he has purchased. One of the basic schemes is based on anonymous cash, and the other on blind decryption. In addition to the basic schemes, we present and compare enhancements to the schemes for supporting additional features such as variable costs, enforcement of access restrictions (such as "over age 21"), and the ability of a user to monitor and prevent covert privacy-leaking between a content-provider-provided box and the content provider. As we will show, the different variants have different properties in terms of amount of privacy leaking, efficiency, and ability for the content provider to prevent sharing of encryption keys or authorization credentials.

Categories and Subject Descriptors

C.2.0 [**Computer Networks**]: General – *Security and protection.*
K.4.1 [**Computers and Society**]: Public Policy Issues – *privacy.*
E.3 [**Data**]: Encryption

General Terms

Algorithms, Design, Economics, Security, Human Factors.

Keywords

Algorithms, Protocols, Blindable Parameterizable Public Key, Privacy, DRM.

1. INTRODUCTION

Most work in the field of Digital Rights Management (DRM) focuses on the problem of preventing its circumvention. This paper looks at a different problem: how to charge for the use of content while allowing the user to maintain her privacy (in the sense of not revealing to the content provider what content was purchased by which user). In some scenarios, privacy is of greater concern to the user than the payment required. This paper presents and contrasts two basic approaches, plus variants, of systems in which content is distributed in encrypted form, and the user pays to receive a decryption key. The first is based on Chaum's anonymous cash [5]. The second is based on blind decryption [15].

In addition to the basic schemes, we provide various methods of enhancing these schemes for functionality such as different costs for different content, ability of a third party to create content to be distributed by the content provider, and enforcement of authorization policies. Additionally, we examine the scenario where, for DRM enforcement reasons, there is a sealed box, provided by the content provider on the user's premises, that communicates with the content provider to acquire keys and does the actual decryption. We examine the problem of whether the user can detect or prevent the sealed box from covertly telling the content provider what content the user is decrypting. We show that it is impossible, if the user is only passively monitoring the channel, for the user to know whether the box is indeed leaking information. We then show a mechanism in which the user can cooperate with the box in forming the message to be sent to the content provider, and be assured there is no collusion going on, without impacting the ability of the content provider to enforce DRM.

Although the focus of this paper is not the cryptography, we do introduce a new variant of asymmetric keys; the ability to have a family of blindable keys, parameterized by an arbitrary string, which we will use to encode information such as authorization policies or monetary units. This functionality can be provided by a somewhat unusual use of identity based encryption (IBE), but we also introduce two alternative algorithms, which lack some of the properties of IBE that are not needed for our application.

We will assume that there are enough items of content distributed by the content provider that the mere fact that a user is doing business with the content provider, and the amount of money the user spends with the content provider, is not a privacy issue. However, as we will show, privacy leaking is not absolute, and some of the solution variants have different tradeoffs.

Encrypted content must be accessed anonymously, though that is not the focus of the paper. Encrypted content might, for instance, be broadcast video, or content posted on the Internet. If the content is broadcast, say from a satellite or

IDtrust '10, April 13-15, 2010, Gaithersburg, MD.

via cable TV, there may be no problem with accessing the encrypted content anonymously. If the encrypted content is downloaded from the Internet, some sort of anonymization technique would be required, e.g., [10], [11], [16].

In addition to the encrypted content for an item, there will be associated metadata that can be used to decrypt the content with the help of the content provider. For example, metadata may include a decryption key for the content, encrypted with the public key of the content provider. Metadata may also contain other information, such as an authorization policy for accessing the content.

Another aspect of DRM, also not the primary focus of this paper, is how to prevent a user from copying content and sharing it with others. There has not been a foolproof technical solution, especially since the analog output of video and audio has to be available. For instance, it is not uncommon for people to carry a camcorder into a theater, record the movie as it is played, and then sell copies later. Various proposed solutions for enforcing DRM include threats of prosecution if caught illegally copying and distributing, watermarking to discover which copy leaked [1], [7], [4], [9], and various software and hardware techniques to prevent copying [14], [12]. Even though there might never be a foolproof technical solution, it is common today for digital content to be distributed with some degree of copy protection, even in software-only systems. This is evidence that content providers believe that copy protection deters a sufficient amount of copying that the complexity (and customer annoyance) of the DRM is of positive value (to the content provider).

So this paper is not about how to make DRM itself more secure; it is instead focused on enhancing DRM with additional functionality.

DRM enforcement commonly involves using a sealed box (e.g., the box that a video satellite provider installs at the user's house with a subscription to his service). We assume in such deployments:

- The box's only means of communication with anything is through a channel that the user can monitor.
- The user can modify messages to/from the box (the user can place an additional box, along the channel that the box uses to communicate with everything else).
- The user cannot examine the logic inside the box to determine whether it is indeed designed not to divulge the user's identity.

This fairly common deployment scenario leads to interesting functional differences between the schemes presented in this paper.

In sections 2 and 3 we present the two basic schemes (anonymous cash in section 2 and blind decryption in section 3.) In section 4, we compare the efficiency of the two schemes and conclude that the blind-decryption-based scheme offers superior performance and lower overhead, while the anonymous-cash-based scheme provides the additional functionality of allowing the content provider to do per-item accounting.

In section 5, we propose modifications of the two schemes that allow the content provider to charge different amounts for various pieces of content, without compromising the user's privacy. In section 6, we propose similar modifications of the two schemes so that the content provider can enforce authorization policies (such as "over 18", "citizen of US", or "citizen of any country except Monaco or Grenada") that might restrict access to some content. We also discuss the comparative implications on our scheme variants when authorization policy might be very complex. In section 7 we consider how the user, while communicating with the content provider using a sealed box can be assured that the box is not covertly leaking information about the user's purchases to the content provider.

2. First Scheme: Basic anonymous-cash-based DRM

2.1 The concept of anonymous cash

Chaum [5] introduced the concept of anonymous cash. The basic idea is that a data structure with a particular syntax, signed with the bank's private key, is worth a fixed amount of cash. The data structure includes a random number large enough to assure that independently chosen values will be unique. The anonymity comes from the construct of blind signatures, where Alice can get the bank to sign something without the bank knowing what it is signing.

Alice chooses a random number R, hashes it, and formats it according to the rules of valid currency. Alice "blinds" it, and presents the blinded result to the bank, which signs the result with its private key. Then Alice applies the blinding function's inverse function ("unblind") to obtain a value we will refer to as "the bank's signature on R".

The bank will not know the values of R that Alice has purchased, so when R is "spent" the purchase cannot be traced to Alice, though the bank will know how many tokens Alice has purchased. Merchants accepting the anonymous cash can verify it is valid by checking the bank's signature. The only problem is assuring that Alice doesn't spend the same valid unit of anonymous cash more than once. If there is only one place accepting the anonymous cash (in this case the content provider), then double spending can be prevented by having the content

provider remember all the R's that have been spent. Alternately, if the bank issuing the anonymous cash is online, then the cash can be spent with multiple merchants, provided that the bank remembers all the R values used and is consulted by each merchant on each transaction before the anonymous cash is accepted.

Chaum, Fiat, and Naor extended the notion of electronic cash to allow for an offline bank [6]. In this scheme, Alice might successfully spend digital cash multiple times, but once the bank collects the transactions (the spent cash), the culprit's identity will be revealed.

The latter anonymous cash scheme is more complex and expensive and our application does not require the off-line assumption. We will therefore use the simple notion of random R's, that have been blindly signed in advance to indicate that the holder of the signed R is allowed to trade that R for a unit of merchandise.

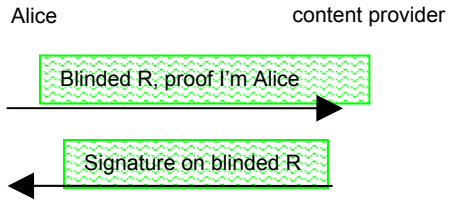
2.2 Using anonymous cash for DRM

In our application there is no reason for there to be a third party (the bank) providing general purpose tokens that can be spent with multiple merchants. Alice can directly purchase tokens from the content provider.

2.2.1 Obtaining cash

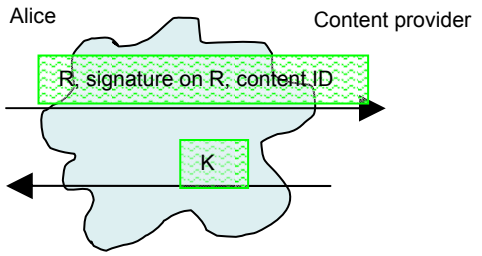
This will be done non-anonymously, in a conversation that must be authenticated and encrypted. The shaded text box indicates encryption.

Alice must pay for the cash through some mechanism such as a credit card, or having a pre-paid account with the content provider debited when she obtains cash.



2.2.2 Purchasing content

To purchase content, Alice presents the anonymous cash, together with the metadata for the content she wishes to access, and the content provider returns the content key. This interaction must be both **anonymous** (because the content provider will know what content is being requested and must not know who is requesting it) and **encrypted** (since otherwise an eavesdropper could steal the cash or the content key). The cloud in the diagram indicates an anonymization infrastructure. Note that an anonymization infrastructure is very expensive in terms of computation and bandwidth [10].



Since the transaction where Alice is requesting a content key must be anonymous and encrypted, the metadata for an item could simply be the item's ID, and the content provider would keep a table of (content ID, content key) pairs. (In contrast, as we will see in section 3, in the blind decryption scheme, the metadata for an item must be {K}P, i.e., the content key encrypted with the content provider's public key.)

However, it might be preferable, even in the anonymous cash scheme, for the metadata to be {K}P rather than simply a "content ID" if:

- the content is to be prepared by a 3rd party; otherwise, it would be necessary for the 3rd party to securely relay the content key for that content to the content provider.
- it were inconvenient for the content provider to securely keep a large table of (content ID, key) pairs.

3. Second scheme: Blind decryption

In this second scheme, we use blind decryption instead of blind signatures. Blind decryption is similar in spirit to blind signatures, but there are more algorithms that work for blind decryption than blind signatures because blind decryption does not require a "public" key. Blind decryption works with various schemes including RSA keys (as with blind signatures), Diffie-Hellman keys, and IBE (identity based encryption).

3.1 Mechanics of Blind Decryption

3.1.1 RSA Keys

With RSA keys, blind decryption is a simple variant of blind signatures. If the content provider's public RSA key is (e,n), with the private key being (d,n), then the encrypted data key K will consist of K^e mod n.

To obtain K, Alice blinds K^e mod n by choosing a random number R, "encrypting" R with the content provider's public key to obtain R^e mod n, multiplying the two quantities together to obtain (K^e * R^e mod n), and presenting the result to the content provider, which uses its private key by raising to d mod n, resulting in K*R mod n, which it returns. Alice divides by R mod n to obtain K.

3.1.2 Diffie-Hellman Keys

Blind decryption can work with Diffie-Hellman keys, chosen from any Diffie-Hellman group, including elliptic curves. We will call the operations "multiplication" and "exponentiation" although, in the literature, elliptic curve operations are usually called "addition" and "multiplication". But we find the description with multiplication and exponentiation more clear for people who are familiar with Diffie-Hellman but not with elliptic curves. That way the formulae work with both mod p Diffie-Hellman and with elliptic curves. Note: the Diffie-Hellman blind decryption we are presenting is a simplification of one presented in [15], and it works for blind decryption, but would not work as a blind signature scheme. Also, for brevity, assume the operations are being done mod p (rather than having us say "mod p" each time).

Assume the content provider's public Diffie-Hellman key is g^x, and the private key is x.

A content key K is of the form g^{xy}. If the encryption algorithm requires a particular form factor for the key, such as being 128 bits, then some function would be performed on g^{xy} to convert it to the right form factor, such as a cryptographic hash.

The metadata associated with the item that is encrypted with key g^{xy} includes g^y.

In other words, g^{xy} (or more likely a cryptographic hash of g^{xy}) is used as a symmetric encryption key (for any symmetric key algorithm such as AES) to encrypt the content, and the metadata includes g^y. To decrypt the content, Alice must obtain g^{xy}. If blinding were not necessary, Alice could send the content provider g^y and have the content provider apply its private key (i.e., exponentiate by x) and return g^{xy} mod p. But we need this operation to be blinded.

Each item of content distributed by a particular content provider is encrypted with a different key (a different y was chosen), but they all use the same secret x. The value y is independently and randomly chosen for each item.

To blind g^y mod p so that the content provider cannot know which key Alice is purchasing, Alice chooses a value z and computes z^{-1} mod q, where q is the order of the cyclic group generated by g. For mod p groups, q is a large factor of p-1. She raises g^y to z to obtain g^{yz} and sends that to the content provider.

The content provider raises this to its private key (x) and returns to Alice: g^{xyz}.

Alice unblinds g^{xyz} by exponentiating by z^{-1} to obtain the content decryption key g^{xy}.

3.1.3 Identity-based encryption(IBE)

The Boneh-Franklin (BF) scheme used in IBE [2] can also be used by our scheme for blind decryption, although we will be using it in a different way. In IBE, as traditionally used, there is a master key generator. Anyone knowing the domain parameters can generate a public key from a string, and the master key generator calculates the corresponding private key (using the domain secret), and gives the private key to the public key's rightful owner.

However, in our schemes, there is only one "rightful public key owner" -- the content provider. In the way we use the BF scheme, the content provider will act as the master key generator, in the sense of knowing the domain secret, but it will not give private keys to anyone (other than calculating its own private key). Other parties will never know any private keys; they will only know the domain parameters in order to obtain the content provider public key.

In "normal" IBE, there would be a family of public keys, parameterized with a string "ID". At this point in the paper, we only need a single public key (the content provider's public key), so we can assume that "ID" is a constant. Later in the paper (section 6.3.3) we will want to use a string to create a family of keys, but they will all still be public keys belonging to the content provider.

To create a blindable public key, we will modify a simplified version of the Boneh-Franklin IBE scheme. The BF scheme uses a bilinear map $\hat{e}(P,Q)$, (usually a twisted Weil or Tate pairing) which maps two order q elliptic curve points to an order q finite field element, and has the property that $\hat{e}(P^a, Q^b) = \hat{e}(P,Q)^{ab}$, for points P, Q and integers a, b. The security of BF relies upon the Bilinear Diffie-Hellman assumption that given P, P^r, P^s, P^t, it is difficult to find $\hat{e}(P,P)^{rst}$.

In the case of the basic IBE scheme, a trusted server called the private key generator chooses a secret integer s and an elliptic curve point P, and it publishes as system parameters P^s, P, and a specification of the group that P lives in. The private key generator can generate a private key corresponding to any public key, "ID", by using a special hash function H to map "ID" to an element of the group generated by P. We will write H("ID") as P^t, despite the fact that no party, including the key generator, will be able to compute t. This notation (P^t) is simply used here to make the bilinear Diffie-Hellman problem embedded in the scheme more transparent. The private key corresponding to "ID" is $H("ID")^s$, which may also be written as P^{ts}. To obtain a shared secret key with the holder of the public key, "ID", an encryptor chooses a random number r and transmits P^r. The shared secret is then $\hat{e}(P,P)^{rst}$, which is calculated as $\hat{e}(P^s, H("ID"))^r = \hat{e}(P^s, P^t)^r$ by the encryptor and $\hat{e}(P^r, H("ID")^s) = \hat{e}(P^r, P^{ts})$ by the holder of the public key "ID".

Blinding may be added as follows: suppose a message is encrypted with $\hat{e}(P^r, H("ID")^s)$, and you know P^r and "ID". You want to decrypt the message with the help of the "ID" holder, but you don't want him to find out which value of

P^r was used, since that would unambiguously identify the message you are trying to decrypt. You can do this by choosing a random blinding factor, b, and sending P^{rb} to the holder of "ID". He will send back $\hat{e}(P^{rb}, H("ID")^s) = \hat{e}(P^{rb}, P^{ts}) = \hat{e}(P,P)^{brst}$. You can now get $\hat{e}(P,P)^{rst}$, by raising $\hat{e}(P,P)^{brst}$ to the $b^{-1}(\text{mod } q)$.

3.2 Purchasing Content with Blind Decryption

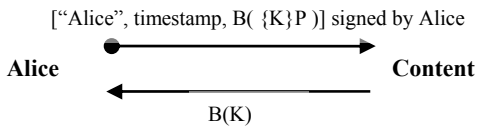
In the anonymous cash scheme, when Alice is purchasing a content key, she must do it anonymously, and the conversation must be encrypted. In our blind decryption scheme, it is not necessary for the conversation to be anonymous or encrypted, but it does need to be integrity-protected (signed by Alice).

There is no need for an anonymizing network. The content provider will know which user (Alice) is accessing an item, and it can debit her account at that time, but it will not know *which* item Alice is accessing.

The protocol for requesting decryption is for Alice to send the content provider a message containing Alice's identity (so her account can be charged for the decryption), along with an encrypted blob (consisting of the blinded encrypted key) that the content provider will "decrypt" with its private key. ("Decrypt" is in quotes because the result will still be encrypted with the blinding function). This message must be signed by Alice, e.g., with a MAC using a secret key Alice shares with the content provider, or signed with her private key, because her account will be debited for the cost of the decryption and we must assure that a third party cannot request a decryption be charged to Alice. It also must be resilient against replays, so an eavesdropper cannot cause Alice to be charged multiple times for the same decryption.

A simple method of avoiding replays without adding messages is for Alice to include a timestamp, have the content provider store the timestamp of the previous decryption request from Alice, and ensure that the timestamps from Alice are monotonically increasing. A sequence number could be used instead of a timestamp.

Alice will not be anonymous in this scheme. She will authenticate to the content provider, and her account will be debited for each decryption of a content key she requests. The content provider will know that Alice has purchased *some* content, but not *which* content.



Using blind decryption to obtain a specific encryption key

4. Comparison of the basic schemes

4.1 Efficiency

The blind decryption scheme is dramatically more efficient than the anonymous cash scheme because the blind decryption scheme does not need an anonymization infrastructure. Also, the anonymous cash scheme needs two conversations: a (nonanymous) conversation to purchase tokens, followed by an anonymous, encrypted conversation to request (and pay for) a content key. In contrast blind decryption only needs a single interaction; debiting Alice's account and having Alice request a content key are done in the same (nonanonymous) two-message exchange.

Another important difference is that with the blind decryption scheme, the content provider only requires a single private key operation (to blindly decrypt {K}P). The anonymous cash scheme requires one private key operation for the content provider to blindly sign each token, as well as a private key operation to establish the server-side-authenticated encrypted channel required for content key requests. The anonymous cash scheme is also likely to require an additional private key operation to set up the encrypted conversation in which Alice purchases tokens, although it could be done with a long-term shared secret key between Alice and the content provider, and many tokens can be purchased in the same conversation. Additionally, although we showed a protocol, where the metadata is the content ID, and retrieving the content key is a table lookup, that scheme requires the content provider to keep a large database (keys for all the content items). As such, it is likely preferable for the metadata to be {K}P, in which case the anonymous cash scheme would require at least three private key operations for the content provider, versus one for the blind decryption scheme.

The main expense of the anonymous cash scheme (compared to the blind decryption scheme) is the cost of the anonymization infrastructure, both in bandwidth and computation, placing computational burdens not just on Alice and the content provider but also on the relay nodes. Although obtaining the encrypted content (in either scheme) might in some cases require an anonymization network, there are scenarios (such as acquiring content through broadcast video) in which the blind decryption scheme would not need such a channel. However, the anonymous cash scheme will *always* require the existence of an anonymization infrastructure (though in most descriptions of anonymous cash in the literature, this important detail is omitted).

4.2 Per-item accounting

The anonymous cash scheme allows the content provider to know how many people have purchased each item of content, (although it does not know specifically which people have purchased which content). In contrast, the blind decryption scheme does not allow this.

It might be important in some applications for the content provider to know how many people have purchased each item, in order to determine the royalty amount for each content contributor. However, many schemes deployed today (e.g., premium TV channels that show many movies) do not have any mechanism for the content provider to know how many people have watched specific movies. Payment to receive a premium channel is a flat rate regardless of how much or which content is accessed within that channel. So in many applications this per-item accounting is not required.

5. Variable Charging

It is possible that some content might cost more than other content. With the anonymous cash scheme, it is simple to charge different amounts for different content, since the content provider knows which key is being requested. So, the content provider could require n tokens to purchase an item worth n units.

This straightforward approach does not work in the blind decryption scheme, since the content provider does *not* know which key it is decrypting.

5.1 Multiple Keys

In blind decryption, a piece of content that costs n units of money could require n encryption keys and n decryption requests. So for instance, the metadata for an item costing n units could contain, for i = 1 through n, $\{K_i\}P$. Alice would need to decrypt each of the K_i and then perhaps $\oplus$ them or hash them together to obtain the content key.

Note that requiring n decryptions or requiring n blindly signed tokens to purchase an item worth n units puts a burden of n-1 additional private key operations on the content provider in either scheme (either it has to blindly sign n tokens or do n blind decryptions).

5.2 Multiple-value tokens and multiple-value public keys

Instead of making an item worth n units require n private key operations, we can make it require, say, $\log_2$ n operations, using either anonymous cash or blind decryption, by having the content provider have different public key pairs for different denominations of money.

For instance, with the anonymous cash scheme the content provider could have public keys: P1 worth 1 unit, P2 worth 10 units, and P3 worth 100 units. When Alice purchases anonymous cash, she can specify the denomination that she would like. If she specifies she wants a 100-unit token, the content provider would debit her account 100 units of money and blindly sign the token with public key P3. To purchase something worth 14 units, she could present 14 single tokens, or a 10-unit token plus 4 singles.

This savings can also be done with blind decryption. Suppose there was an item worth 14 units. (Assuming the denominations of the content provider's public key are 1, 10, and 100), the metadata associated with the 14-unit item would contain 5 wrapped keys; ((unit=10, {K1}P2), (unit=1, {K2}P1), (unit=1, {K3}P1), (unit=1, {K4}P1), (unit=1, {K5}P1)). Alice would need to do 5 blind decryptions, each time specifying the unit, e.g.,

> ["Alice", timestamp, B({K}P2) **unit=10**] signed by Alice

And the 5 keys would be cryptographically combined to form the content key.

Note that if the metadata gives Alice the choice of unwrapping 14 single-unit keys, or 5 variable-unit keys (e.g., a ten and 4 ones), then these keys could not be simply be hashed together to form the content key. Either the function would have to be $\oplus$ (where it is easy to make two different sets yield the same answer), or if a hash was used, you'd wind up with two different quantities, say K1, and K2. The real content key C could be stored in the metadata as {C}K1 and {C}K2, so that C would be retrievable whether Alice had computed K1 or K2.

5.3 Issue: Privacy and large-unit tokens or decryptions

If all G-rated content cost 1 unit, and all X-rated content cost 10 units, the variable charging could leak information. In the anonymous cash scheme, Alice could buy anything she wants with (lots of) unit tokens, and the content provider would not know who was purchasing the expensive content. Or even the fact that she has purchased a large denomination note does not mean she is intending to buy a single expensive item, since she could pay for multiple single-unit purchases in the same transaction with a single, large denomination note.

With the blind decryption scheme, Alice is not anonymous, and has to unwrap the content in the same denominations that it was wrapped. To help protect privacy:

- Alice could spread decryptions over time, so the content provider wouldn't be able to tell the exact amount of any item (e.g., for a 14-unit item, she could request decryption of the 10-unit key at a different time from requesting the 4 single-unit keys).
- The content provider could provide metadata for a 14-unit item that would allow retrieving the item using n single-unit decryptions, rather than the smaller number of decryptions possible using larger denomination keys. Both types of metadata could be provided, giving Alice the choice. So, the content key could be the $\oplus$ of 14 single-unit decryptions in the metadata, or the $\oplus$ of a ten-unit decryption plus four single-units.

To avoid having users opt for unwrapping content using single unit keys (putting a computational burden on the content provider), the content provider could provide other content (rather than just X-rated content) that is worth more than one unit, for instance a package of all the Disney movies together, or entire seasons of "Little House on the Prairie". Or, the content provider could provide a discount for using the larger-unit keys (the metadata for a 14-unit item could give Alice the choice of unwrapping 14 single-unit keys, or, say, a 10-unit key and two single-unit keys, so that the item would cost only 12 units if she uses the larger denomination key. In the case of purchasing anonymous cash, the content provider might provide discounts for large-value tokens, e.g., charging 9 units to obtain a 10-unit token.

6. Authorization Categories

In some cases it is not sufficient to pay for content; one must also be authorized to purchase that particular content. For example, X-rated content might only be legally purchasable by someone over age 21. Or some other content might only be legal to sell to citizens of some countries. The system must allow anonymous purchase, but only to qualified individuals.

In sections 6.1, 6.2, and 6.3 we discuss three methods of providing for authorization, and if/how each of the two basic schemes can be modified with each of these:

- Authorization secrets used as credentials
- Authorization secrets used as content key components
- Authorization category-specific public keys

The various approaches have different tradeoffs in terms of amount of privacy information leaked, efficiency, functionality, and ability to prevent credential sharing. Section 6.4 compares the three methods, while section 6.5 gives cryptographic techniques that can be used to make authorization category-specific public keys more efficient.

Regardless of the method used to add authorization, the authorization policy for an item must appear in the metadata in cleartext, so that Alice can tell what types of authorization she must obtain in order to purchase the item. We will use the term "ACL" (Access Control List) to mean the authorization policy associated with an item, and we assume it can consist of any Boolean combinations of groups, roles, identities, attributes, etc.

An obvious concern is that any sort of authorization secret could be copied and sent to non-authorized users. However, this is not a special concern with authorization, since this is also true of the content keys. The entire system depends on some sort of DRM enforcement to hinder sharing of content keys as well as authorization secrets. One mechanism, which we will explore in greater depth in section 7, is to use a sealed box like the one that comes with a subscription to satellite TV or cable. But software-only DRM schemes are prevalent today, even though they aren't 100% effective, so they must be sufficiently effective at deterring sharing to satisfy the content providers.

Assume that for each authorization category (e.g., over 21, citizen of country X) there is a server that can determine whether someone is a member of the relevant group or has the relevant attribute. If Alice can prove to that server that she has attribute Z, that server presents her with a secret, S_Z. To prevent an eavesdropper from stealing the secret, the conversation in which Alice obtains S_Z must be encrypted. To prevent Alice from sharing S_Z with unauthorized users, some sort of DRM scheme must be in place.

Since it is common to have multiple users in the same household sharing a system, and they might have different authorizations (e.g. the system may be shared by parents who are over 21 and children who are not), there must be some ability to maintain multiple distinct accounts. There will also need to be some sort of login, so that the system knows on which user's behalf it is acting. The system should keep a database, for each user, of items such as authorization secrets, content keys, and anonymous cash tokens.

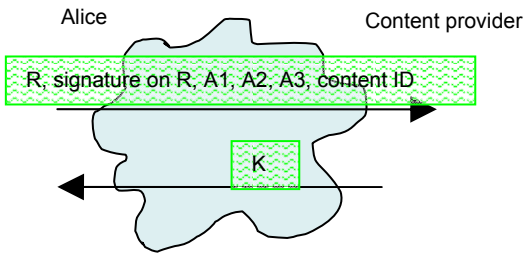
When anyone in the household purchases a content key, it would be a matter of policy whether that key would also be made available to all the household accounts that would be authorized to view that content, or whether each account would need to purchase the content separately. It might be a privacy concern, for instance, for household members to see which items have already been purchased by some other household member.

To lessen the threat of authorized users sharing authorization secrets with others, given that a DRM scheme is likely not to be 100% effective, the authorization secret can, in some of our schemes, be changed periodically, and then authorized users will need to get the new value when their old value becomes invalid. In one of our schemes (authorization category-specific public keys), there are no authorization secrets to share.

6.1 Authorization secrets as credentials

This scheme only works with the anonymous cash scheme. When Alice is anonymously requesting a decryption, she presents all the authorization secrets (A1, A2, A3) that prove she satisfies the ACL for the requested item, along with anonymous cash. It will be known which authorization secrets Alice has ever obtained, but not whether she ever uses them to purchase ACL-restricted content. For maximum privacy, it might be best for Alice to automatically request all authorization keys for which she is eligible so as not to leak any hints about what kinds of content she might be seeking. An authorization secret would only need to be obtained once (per user), and that

would enable that user to access any content that requires that authorization.



Content provider looks up ACL associated with "content ID", and verifies that A1, A2, and A3 are sufficient credentials to satisfy the ACL

It is straightforward to accommodate complicated authorization policy, e.g., of legal age in the country of residence. Since the ACL is part of the metadata, the client can calculate what credential secrets need to be sent to satisfy the policy. The content provider can know what the policy for that content is in one of two ways:

- The content provider stores, for each item of content, (content ID, key, ACL)
- To save the content provider from keeping such a large table, the metadata for the content would be [{K}P, ACL] signed by content provider.

However, there is a potential for privacy leaking. If there is a group with a very small number of members, and someone requests access to something requiring being a member of that group, there is no way to avoid leaking that someone from that group accessed that item. Even if all groups were large, it could be that the intersection of several groups could be very small. If access to the item is the AND of a bunch of groups, it is unavoidable (with this scheme) to divulge that someone who is in the intersection of all the groups has accessed the item.

The issue is with the OR of several groups. Suppose the ACL says that you must be accredited as fluent in at least 3 languages, and Alice happens to know Bulgarian, Bengali, and Navajo. When the anonymous requester presents those three credentials, it will narrow the potential requesters to a very small set, even though each of the groups is large, and even though the ACL would usually allow for satisfaction while still being part of a very large potential set (e.g., with English, French, Spanish).

One feature of this scheme (as opposed to the one we will present in the following section), is that it is relatively easy to periodically change the authorization secrets, to mitigate against some stealing of credentials. When an authorization secret has changed, the user will have to obtain the new secret.

6.2 Authorization secrets as content key components

This variant works with either anonymous cash or blind decryption. We assume that Alice obtains a (symmetric) encryption key for each authorization category that she qualifies for. As with section 6.1, it will be known which authorization secrets Alice obtained, but not whether she ever purchases content requiring them.

This scheme can handle any Boolean combination of authorization categories. To access an item that requires, say, authorizations X and Y, Alice would need to have obtained authorization secret keys K_X and K_Y, in addition to the K wrapped inside the metadata. So, the metadata might consist of: ($\{K\}P$, $\{K_1\}K_X$, $\{K_2\}K_Y$). The decryption key for the content could be, for instance, $h(K,K_1,K_2)$. Alice unwraps $\{K\}P$ with the help of the content provider, but is able to unwrap K_1 and K_2 because she knows K_X and K_Y.

The OR operation would require organizing the metadata to give the client the choice as to what to unwrap. For example, if the ACL was "citizen of US OR citizen of Canada", the metadata might contain (("citizen of US", $\{\{K\}P\}K_{US}$) , ("citizen of Canada", $\{\{K\}P\}K_{CANADA}$)).

If there were an ACL such as "citizen of any country other than Monaco" this would require a large amount of metadata, since that would be the OR of hundreds of countries. In contrast, the authorization claim secrets scheme (6.1) only requires that Alice present the single authorization claim secret for some country other than Monaco (we won't worry about whether someone who is a dual citizen is allowed to see content in this case).

In this scheme (using the authorization secret as a decryption key), it is not as easy to periodically change an authorization secret as it would be in scheme 6.1. It could be done, but it would involve preparing new metadata for all affected content.

6.3 Authorization category-specific public keys

In this scheme, the content provider has different public keys, one for each authorization group. In the blind decryption scheme, this would mean that an encryption key for an item would be wrapped with a category-specific public key. In the anonymous cash scheme, it would mean that the cash token would be signed with a category-specific public key. In other words, in the blind decryption request, Alice would specify "blindly unwrap this using your 'US-citizen' key", and in the anonymous cash purchasing request, Alice would specify "blindly sign this using your 'US-citizen' key".

These could be completely independent keys, or they could be derived cryptographically using any of the methods that we will present in section 6.5.

6.3.1 Boolean combinations with blind decryption

With blind decryption, Boolean combinations of authorization categories can be handled the same way as in scheme 6.2. In other words, an item requiring authorizations A_1 AND A_2 could be encrypted with $h(K_1, K_2)$ and include as metadata $(A_1: \{K_1\}P_{A1})$ and $(A_2: \{K_2\}P_{A2})$. Alice would have to unwrap both keys to read the item. The keys would have to be half the price of the intended cost of the item. The metadata for A_1 OR A_2 would be similar, but just have a single K, such that unwrapping either quantity will work, as in: $((A_1: \{K\}P_{A1})$ OR $(A_2: \{K\}P_{A2}))$, and either of those unwrappings would be the actual cost of the item.

6.3.2 Boolean combinations with anonymous cash

With anonymous cash (assuming the metadata is just the content ID), it works somewhat like scheme 6.1, in that a cash token signed with an authorization-specific key works both as a unit of currency and as proof of authorization. If Alice has to prove A_1 OR A_2, she merely presents a token signed with either the A_1-specific public key or the A_2-specific public key. If Alice has to prove A_1 AND A_2, during the anonymous content request, she could present two (half-price) tokens, one signed with A_1 and one signed with A_2.

6.3.3 ACL-specific keys

An alternative for Boolean combinations is to have a public key which is specific to the entire ACL, e.g., a specific public key for "(paid up member of ACM OR IEEE) AND citizen of US". In other words, in the blind decryption scheme, the metadata would consist of $\{K\}P_{ACL\text{-}string}$. In the anonymous cash scheme, the client would request a cash token signed with the ACL-specific key $P_{ACL\text{-}string}$.

That approach has the disadvantages of

- requiring a lot of content provider keys (but in section 6.5 we will explain how that can be practical), and
- leaking privacy, because although there might be a lot of items of content requiring each of the component authorization categories, there might be very few (or even just a single one) with the specific combination of those categories in the ACL.

6.4 Comparison of 6.3 with 6.1 and 6.2

With authorization-specific content keys, Alice cannot cheat by stealing authorization secrets, since when she requests cash tokens or requests blind decryption, she is not anonymous, and the content provider checks her authorizations by looking them up in her profile. However, it has a serious privacy disadvantage relative to the other two schemes: the content provider will know how many decryptions Alice is asking for, for each ACL.

On the other hand, using either authorization scheme raises revocation issues to a greater or lesser extent: An authorization secret could be stolen, or Alice might no longer be authorized in some category (say, her membership in an organization has lapsed). If communication to the content provider is done with a sealed box, or with reasonably trusted DRM software, then the content provider could keep the authorization secrets in the client up to date. For instance, if "current member of ACM" is required for some types of content, the content provider could communicate with ACM periodically to get its list of members, and then install the "ACM" authorization secret into the boxes (or software) of all the authorized users, and remove the secret from boxes (DRM software) of users who were, but are no longer, members.

Given that even with DRM, authorization secrets might be stolen by determined attackers, it is an advantage of scheme 6.1 that the secrets can be changed periodically.

In contrast, with multiple content provider public keys (6.3), revocation is very simple. All that is required is that the content provider keep track of all of Alice's authorizations. If, for instance, her membership in an organization lapses, that organization would inform the content provider, which would remove membership in that organization from Alice's profile and no longer allow Alice to decrypt anything requiring that authorization. With anonymous cash-based authorization-specific content provider schemes, once Alice has obtained authorization-specific cash tokens it will not be possible to take them back (unless enforced through the DRM software/hardware).

6.5 Blindable Parameterizable Keys

In this section, we present a new cryptographic tool; blindable parameterizable keys, and give several ways of accomplishing this. Armed with such functions, the content provider can have a family of keys, parameterized by the ACL.

6.5.1 Using Identity Based Encryption

The notion of keys parameterized by a string sounds a lot like IBE [17] [2], and indeed the same math can be used for parameterizable blind decryption (but not blind signatures), but we are using IBE in a different way.

We described in section 3.1.3 how to use IBE for blind decryption, but we were not parameterizing the single content provider public key. To make the scheme work with a different public key for every ACL string, we make it more like IBE in the sense that the public key used *is* derived from the ACL string. The rest of the system still works as it did in section 3.1.3 – the content provider

knows the domain secret and can convert any public key into a private key, and the clients never need to know any private keys, just the domain parameters.

6.5.2 Parameterized Diffie-Hellman

Parameterization can be done with our Diffie-Hellman variant of blind decryption. Alice would only need to know "g" and "p". The content provider would only need to know a single secret "x". The metadata for content for "over 21", would consist of (g^y mod p, "over 21"). The content key for that data would be calculated by calculating S=h(x, "over 21") and then raising the metadata to S to obtain the content key g^{yS} mod p.

Alice blinds g^y mod p by choosing a random z, calculating the inverse exponent z^{-1} for mod p exponentiation, and presenting that along with the string "over 21". The content provider uses the string "over 21" to calculate S and returns g^{yzS} mod p. Alice exponentiates, mod p, by z^{-1} to obtain g^{yS} mod p, the content key.

6.5.3 Parameterizable RSA

Note that the schemes we present in sections 6.5.1 and 6.5.2 work for blind decryption but not blind signatures, so neither of them would work for anonymous cash. A scheme that might work as a blindable parameterizable public key scheme is RSA, where the content provider's public key, instead of being (e,n), is simply the modulus n. The public exponent for a given ACL would be the hash of that ACL string.

RSA is clearly not secure if multiple users use the same modulus, since knowledge of a key pair allows you to factor the modulus [3], but we are not proposing that. Instead we are proposing a single user (the content provider) using modulus n, but using a family of exponent pairs parameterized with a string.

It is a good idea for all the public exponents to be relatively prime, so that Alice can't get the decryption of something encrypted with an exponent that she isn't authorized for, by requesting one or more decryptions using exponents she is authorized for and multiplying or dividing the results. With exponents being hashes, this threat is unlikely to ever happen in practice, but it is possible (with some computational cost) to make all the exponents prime by not simply hashing the ACL string, but instead, hashing the ACL string, padding with some number (e.g., 32) of zero bits, and then finding the first prime greater than that.

7. DRM-Enforcement Sealed Box

This section considers the implications on the design in the common deployment scenario where the content provider provides a sealed box, and communication between the "user" and the content provider is actually done between the box and the content provider. We assume that the user can communicate with the box, to tell it which content the user would like to access.

We assume the box is reasonably difficult to tamper with, and an additional hindrance would be that tampering with it would be illegal. A plausible deployment of such a "box" might be a smart card or other sealed module that installs into the user's PC.

7.1 Hindering Copying of Authorization Keys

In many of the variants we have presented, a user collects content keys and authorization keys. So, an obvious implication is that one person can obtain a key to decrypt a piece of content, or an authorization key for "over 21", and widely distribute it.

However, each box will be known to the content provider. Either the content provider will know a public key for each box, or will have a shared secret key with each box. Communication is between the server and the box, and any information that must be kept from the user (such as an authorization key) can either be done through an encrypted channel (such as SSL) between the box and the server, or can be returned to the box encrypted with a key known only to that box. Content and authorization keys, as well as the private key for a particular box will be stored inside the box, and the box would be designed to make it be very difficult to extract keys from the box.

If a determined user does extract keys from a box, all is not lost. It still would be difficult to insert such keys into other boxes. In other words, assuming a reasonably competent job of engineering the boxes to be tamper-resistant, it would not only take a great deal of ingenuity and lack of fear of prosecution to extract the keys from one box, but it would take an equal amount of tampering to insert keys *into* a box, since an untampered-with box would only accept such keys during communication with the content provider.

If the identity key for a particular box were compromised, that might enable simulating an entire box in software (and therefore it would not take much effort to deploy clones), but the compromise of that one box would become known to the content provider quickly (as, for instance, the owner of that box would be charged for all content requested by any clone), and the content provider would revoke the key for that box. Although the content keys and authorization keys known to that compromised box might still be publicly known, it would still be difficult to install these keys into existing boxes.

7.2 Monitoring Privacy Preservation

The box is provided by the content provider, so even if in theory the protocol is intended to enable preserving the user's privacy, the content provider might be motivated to cheat.

Communication is between the box and the content provider, but as we said in the introduction, the user can monitor what is transmitted.

In the anonymous cash scheme, when decryptions are requested, this must be done over an encrypted channel, with a key between the box and the content provider. The user cannot tell what the box is saying. The box could easily be (intentionally) leaking its identity when it asks for a decryption of a particular piece of content.

In the blind decryption scheme, it is also possible for the box to cheat in a way that the user cannot detect through passive monitoring. When the box asks for decryption of a piece of content, the communication is not encrypted, so the user can indeed verify that what the box transmits is "["Alice", timestamp, B({K}P1)] signed by Alice". However, there are several ways for the box to cheat in a way that would be undetectable by Alice, even though Alice can see what it is transmitting.

First we will explain how the box can cheat, and then explain in section 7.3, with a protocol between Alice and the box, how we can allow Alice to enforce privacy protection without interfering with the (legitimate) DRM-enforcing protocol between the box and the content provider.

7.2.1 Cheating with a weak blinding function

There is no way for the user Alice to know whether the box is truly choosing a random number for the blinding function, or whether it is sneakily identifying the content Alice is purchasing, by using a blinding function predictable to the content provider.

An example method for the box to cheat and let the content provider know which item Alice is requesting, without Alice being able to detect that it is cheating, is as follows:

The random number it uses for the blinding could be a hash of the secret the box shares with the content provider, and the time. The granularity of time units must be small enough so that consecutive decryption requests would have different blinding quantities, but large enough so that it is not expensive for the content provider to do a brute force search on all possible blinding functions derived that way until it obtains a K with recognizable formatting. Recognizable format, for instance, might be where K in {K}P was padded with specific structure, e. g., according to the PKCS #1 standard [13].

7.2.2 Cheating by using the integrity check

If the integrity check between the box and the content provider is a shared secret key, the key will not be known to Alice, because the content provider does not want Alice to be able to ask for content keys.

In this case, the box can leak, say, the ID of the content that Alice is requesting, by adding the ID of the content to the integrity check. For example, if the proper integrity check for the message

"Alice", timestamp, B({K}P1)

using the shared secret K is "X", and the ID of the content being requested by Alice is n, then instead of sending X as the integrity check, the box could send X+n. To retrieve "n", the content provider computes the correct integrity check for the message (X) and subtracts it from the integrity check as sent by the box.

There really is no way to fix this, so the integrity check must be a public key-based signature, where Alice must have access to the box's public key so she can verify that the box is providing valid signatures.

However, there is still a problem. In many public key signature schemes, e.g., ElGamal, there is a per-message random number x, where $g^x \bmod p$ is part of the signature. The box could choose an x that leaks the ID of the content being requested. For example, the box could try lots of x's, until it finds one for which the lower bits of $g^x \bmod p$ reveal the ID of the content. If it were exactly the ID of the content, Alice would be able to detect this; however, there are ways for the box to do this undetectably to Alice. For example, if the box shares a secret S with the content provider, and if both the box and the content provider remember the timestamp T of the last request to the content provider, the box could compute T encrypted with S, take the bottom n bits of {T}S (where "n" is the number of bits in a content ID), $\oplus$ the result with the content ID to obtain the quantity Q, and find an x such that the bottom n bits of $g^x \bmod p$ is Q.

Thus there really is no way for Alice to passively monitor the channel and be reassured that the box is indeed preserving her privacy, in either the anonymous cash scheme or the blind signature scheme.

However, in section 7.3 we will provide a mechanism for Alice to interact with the box and be assured that the box is not colluding with the content provider. The only way this can work is with the blind decryption scheme using public key signatures for the integrity check. We will show how Alice can protect against both methods of the box cheating (weak blinding function and leaky integrity check).

7.2.3 Cheating by using the timestamp, or timing

If the timestamp has sufficient granularity, it would be possible for the box to leak information in the low order bits of the timestamp. Also, it might be possible for the box to covertly signal information to the content provider based on when it sends requests. Both of these threats are easily countered, as explained in section 7.3.3.

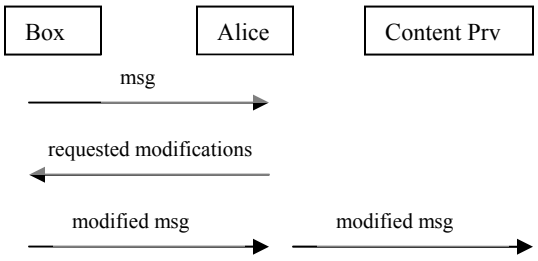
7.3 User-enforced Privacy Protection

With the anonymous cash approach, the user has no recourse other than trusting that the content provider's box is indeed protecting the user's privacy, because the conversation between the box and the content provider must be encrypted. The DRM system will not allow Alice to monitor the conversation (e.g., by letting the encryption

be between Alice and the content provider rather than the box and the content provider) because she is not allowed to see the content key.

However, it is possible, with the blind decryption schemes, to have a protocol between Alice and the box in which Alice can be assured that her privacy is being protected.

The basics of the protocol are that the box emits a message it would like to send to the content provider. Because Alice sits between the box and the rest of the world, Alice can choose either to send this message on to the content provider or to intercept the message. If she intercepts the message, she can send it back to the box, together with instructions for modifying the request. The box then modifies the message it would have sent, using Alice's instructions. Alice will be able to verify that the box incorporated Alice's R into the message the box sends to the content provider.



7.3.1 Foiling weak blinding

As we discussed in section 7.2, with the blind decryption scheme, the box could choose blinding functions that are predictable by the content provider, and thereby allow the content provider to discover which content Alice was accessing. This is unavoidable if Alice is merely passively monitoring the channel.

However, there is a way (with the blind decryption scheme) for Alice to enforce that there be no such convert channel between the box and content provider. The simplest solution (which doesn't quite work, but we will fix it) is to have Alice insert an extra level of blinding in the message to the content provider, and reverse her level of blinding before passing the result back to the box.

In other words, what we'd like is that the box would transmit

- "Alice", timestamp, B({K}P)

to the content provider, but the message would be intercepted by Alice, who would add an extra level of blinding, say with function B2, and forward to the content provider:

- "Alice", timestamp, B2(B({K}P))

The returned message from the content provider will be

- B2(B(K))

Alice would then unblind with B2's inverse and forward B(K) to the box.

But this would not work. The problem is that the message between the box and the content provider needs to be integrity protected; otherwise, anyone could ask for decryptions, and Alice's account would be debited. Even Alice is not trusted (by the content provider) to generate messages, since the content provider wants to keep decrypted content keys inside the closed system (only accessible by the boxes provided by the content provider). Since the message from the box to the content provider is integrity protected, Alice cannot modify it without invalidating the message.

So, the solution is for Alice to interact with the box in order to influence what it uses for blinding.

The constraints are:

- The box cannot trust Alice to do the complete blinding (because Alice is not allowed to see the content key).
- The signed message to the content provider must be generated by the box (since only it is trusted by the content provider to sign messages).
- Alice needs to be able to verify that the box is not attempting to leak information, and that it really is applying the extra level of blinding she requests.

So the protocol is to allow Alice to ask the box to apply an extra level of blinding, with a key that she chooses and specifies to the box. She will be able to verify that her level of blinding has been applied, because she can compare the box's output before and after her blinding function has been applied. The box will be able to unblind with both functions; the blinding function it chose, and the one that Alice chose. The content provider will act as it did before, though if it were attempting to collude with the box, it will notice that the box is no longer colluding with a weak blinding function (since the content provider will not be able to unblind the message from the box to discover what key Alice is attempting to access). If there was no collusion attempt going on between the box and the content provider, the double blinding will be undetectable by the content provider.

7.3.1.1 Using RSA keys

The box originally chooses the blinding function R1, and emits the signed message:

"Alice", timestamp, $R1^e * K^e \bmod n$

Alice intercepts this message, chooses a random R2, and returns the message to the box saying "please add an extra level of blinding using R2."

The box then transmits the signed message:

"Alice", timestamp, $R2^e * R1^e * K^e \bmod n$

Alice examines this by dividing by $R2^e$ mod n, to ensure that the result is what the box originally transmitted ($R1^e$ * K^e mod n). If the answer is correct, she forwards the now doubly blinded message to the content provider

The content provider applies its private key and returns:

R2 * R1 * K mod n

Alice lets the message go to the box, which knows both R1 and R2, and can therefore extract K.

This protocol will work, in the sense that the key will be properly extracted for the content that Alice requested, and also, that Alice is assured that the box has not leaked to the content provider the identity of the content she has requested.

If the content provider had been attempting to collude with the box by having it use a predictable blinding function, the content provider will notice that it is unable to unblind what it received.

7.3.1.2 Using Diffie-Hellman keys

If instead the content provider had a public Diffie-Hellman key, say g^x mod p, then the protocol to extract the encryption key for a piece of content from the metadata for that content, say g^y mod p, would be:

- The box would choose a blinding number z1, exponentiate by z1 mod p, and transmit the signed message:
  - "Alice", g^{y*z1} mod p
- Alice would intercept this, choose her own blinding number z2, and say to the box
  - Add blinding using z2
- The box would then transmit the signed message:
  - "Alice", $g^{y*z1*z2}$ mod p
- Alice raises $g^{y*z1*z2}$ mod p to her number's inverse exponent and verifies that the result is the original one transmitted by the box, i.e., g^{y*z1} mod p
- Alice lets the message go through to the content provider, and allows the return message to go through to the box.

7.3.2 Foiling Leaky Signatures

The other method for the box to cheat and collude with the content provider is by leaking information in the integrity check. If the integrity check is a secret key shared between the box and the content provider, there is nothing Alice can do.

However, if the integrity check is based on the box's public key, then Alice can ensure there is no cheating, as long as she has access to the box's public key (and she monitors that signatures that the box emits are correct).

With RSA keys, and with PKCS #1 v1 padding, there is no problem.

With signatures involving a per-message random number, such as ElGamal, it is possible (as we showed in section 7.2.2) for the box to leak information.

As with double blinding, Alice can enforce that the box is not choosing a bad random number x by allowing Alice to contribute to the random number. The box first presents to Alice the message it would like to send, including g^x mod p. Alice then chooses her own random number y and tells the box to include "y" in its signature. Then she tests whether the box modifies g^x mod p to instead be g^{xy} mod p, and still sends a valid signature.

7.3.3 Foiling other attacks

7.3.3.1 Timestamp

The box could, in theory, leak some information in the least significant bits of the timestamp, assuming the timestamp had sufficient granularity that it could do that while still having a timestamp that was plausible to Alice. If it was using a sequence number, then it could not embed information, since the sequence number would be constrained to be one bigger than the last request.

In some cases Alice might not be keeping sufficient state to be able to monitor the sequence numbers, and therefore it might be more convenient to use a timestamp.

When she is making the request to modify the message, she can also request a specific timestamp, close enough to the actual time so it would still be a valid timestamp, but without the box being able to control the low order bits.

7.3.3.2 Timing

To foil the box leaking information by when it sends requests, Alice can delay a message between the box and the content provider by some amount of time before passing it on.

More broadly, there might be a piece of popular content that many users may attempt to access at the time that it is broadcast for the first time. The fact that someone is asking to access something at just that time would be a clue that the user is likely accessing that particular piece of content.

To mitigate this issue, the content provider should provide the metadata for content well in advance of the broadcast. Even if the data for the content does not exist, there is no reason why the key with which that content will be encrypted could not be chosen and posted well in advance. Then users can collect the metadata for that content and request decryption of the key(s) well in advance of the existence of the content. While this is not strictly something Alice can enforce, she can at least verify that the content provider is consistently making metadata available early.

7.3.3.3 Box-initiated encrypted communication

There are times when the content provider needs to transmit encrypted information to the box; e.g., authorization secrets. If this were done by establishing an encrypted channel between the box and the content provider, then the box could transmit any information it wanted without Alice being able to monitor it. For example, it could inform the content provider which items Alice has recently purchased.

There is no reason for the box to be sending encrypted information to the content provider (other than the blinded content key, which we discussed in section 7.3.1.). But the content provider does need to send encrypted authorization secrets to the box.

Rather than doing this by establishing an encrypted channel between the box and the content provider, authorization secrets can be encrypted by the content provider with the box's public key, or with a shared secret key between the content provider and the box. As long as all of the information from the box to the content provider is unencrypted (again, other than the blinded content key), Alice can prevent the box from leaking information to the content provider.

8. Conclusions

We have examined two families of privacy-preserving DRM schemes, one based on anonymous cash and the other based on blind decryption.

The blind decryption scheme is less expensive, because decryption purchases and decryption requests can occur in the same message. In contrast, the anonymous cash scheme requires a (non-anonymous) communication to purchase tokens and a separate anonymous communication for purchasing decryptions. Also, the anonymous cash scheme requires an anonymization network.

We provided a way (in either scheme) to provide differential costs of items using multiple denomination content provider public keys.

The anonymous cash scheme allows the content provider to do accounting of how many accesses there are for each item of content, which might be important if royalties to the copyright owners of individual items of content are based on number of accesses. The blind decryption scheme does not support this.

We examined several variants for supporting additional authorization. We concluded that authorization encryption keys worked equally well with anonymous cash or blind decryption, and leaked the least privacy information. The authorization claim secret scheme had the advantage that authorization keys could be changed inexpensively. The multiple content provider public key scheme has the privacy disadvantage that it knows the authorization policy of the content that Alice is decrypting. However, it does have the advantage that there are no authorization secrets to steal from authorized users, and revocation of a user's authorization in a category is trivial.

To make it practical to have many content provider public keys, e.g., based on potentially complex authorization categories, we provided a scheme, inspired by IBE, wherein the content provider's Diffie-Hellman key is derived from the authorization string. This is not an IBE scheme because Alice never finds out (or needs to find out), the particular content provider public key. All she needs is the Diffie-Hellman parameters (g and p), and the string, (say "citizen of US AND over 21").

The most likely deployment scenario for this type of application is where communication is not directly between the content provider and an open computer controlled by the user, but rather by a sealed box approved by the content provider and provided by the content provider to sit in the user's house.

We examined the implications of this design. In particular, we concluded there is no way in any of the schemes, if the user can only passively monitor all communication to and from the box, to see if the box is indeed performing the privacy protection protocol properly, rather than covertly leaking to the content provider what the user is accessing.

We concluded that only in the blind decryption scenario would it be possible to enhance the system with a protocol between the user (a computer controlled by the user) and the box, so that the box can continue to enforce the legitimate interests of the content provider, but the user can enforce that the box not covertly leak privacy-compromising information to the content provider. We discussed several ways in which the box could covertly pass information to the content provider that would be undetectable to Alice, if she were only passively monitoring the communication, and we presented methods for Alice to be assured no such covert channel is going on, by allowing Alice to influence the messages between the box and the content provider.

9. Acknowledgements

We would like to thank John Kelsey, Dave Molnar and Hilarie Orman for their helpful comments and advice.

10. REFERENCES

[1] Bender, W., Gruhl, D., Norimoto, N., "Techniques for data hiding", Proc. of SPIE, 1995.

[2] Boneh, D., Franklin, M., "Identity-Based Encryption from the Weil Pairing" *Advances in Cryptology - Proceedings of CRYPTO 2001.*

[3] Boneh, D., "Twenty years of attacks on the RSA cryptosystem", Notices of the AMS, 1999.

[4] Boneh, D., Shaw, J., "Collusion-secure fingerprinting for digital data", CRYPTO 1995.

[5] Chaum, D., “Blind signatures for Untraceable payments”, Advances in Cryptology - proceedings of Crypto 82, 1983.

[6] Chaum, D., Fiat, A., and Naor, M. “Untraceable electronic cash”. Proceedings on Advances in Cryptology (Santa Barbara, California, United States). 1990.

[7] Cox., I., Miller, M., Bloom, J., “Digital Watermarking”, Morgan Kaufmann Publishers Inc., 2001.

[8] Cox, I., Kilian, J., Leighton, T., Shamoon, T., “Secure Spread Spectrum Watermarking for Multimedia”, IEEE Transactions on Image Processing, 1997.

[9] Craver, S., Memon, N., Yeo, B. L., and Yeung, M. M. “Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks and implications”. IEEE Journal. Selec. Areas Comm. 1998.

[10] Dingledine, R., Mathewson, N., Syverson, P. “Tor: The Second-Generation Onion Router”. Usenix Security Symposium, 2004.

[11] M. J. Freedman and R. Morris. Tarzan: A peer-to-peer anonymizing network layer. In 9th ACM Conference on Computer and communication Security, 2002.

[12] Iannella, R., “Digital Rights Management (DRM) Architectures, D-Lib Magazine, June 2001.

[13] Jonsson, J., Kaliski, B., “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”, RFC 3447, February 2003.

[14] Nair, S., Tanenbaum, A., Gheorghe, G., Crispo, B., "Enforcing DRM policies across applications", Proceedings of the 8th ACM workshop on Digital rights management, 2008.

[15] Perlman, R., “The Ephemerizer: Making Data Disappear”, Journal of Information System Security, 2005.

[16] Saint-Jean, F., Johnson, A., Boneh, D., Feigenbaum, J., "Private Web Search", Proceedings of the 2007 ACM workshop on Privacy in electronic society", 2007.

[17] Shamir, A., “Identity-Based Cryptosystems and Signature Schemes”, *Advances in Cryptology: Proceedings of CRYPTO 84.*

[18] Shamir, A., “How to Share a Secret”, Communications of the ACM, v 22 n 11, p 612-613, Nov 1979.

Privacy-preserving DRM

Radia Perlman; Intel Laboratories
Charlie Kaufman; Microsoft
Ray Perlner; NIST

The problem

- Let Alice purchase content
- Without anyone knowing which content she purchased

Basic approach

- Obtain (encrypted) content somehow
  - from satellite TV
  - from the Internet (through an anonymizer)
- Purchase the content key

With wrinkles

- Additional authorization
  - Over 21
  - Citizen of US
  - Citizen of any country other than Monaco
- Differential costs (some things cost more than others)
- Implications of content-provider provided sealed box at customer site

Structure of talk

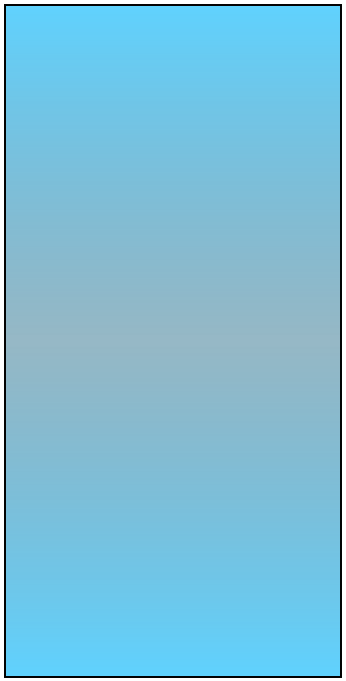
- Two families of schemes
  - anonymous cash
  - blind decryption
- Comparison of these schemes
- Adding wrinkles with each scheme

Encrypted content has metadata

- The metadata might, for instance, contain the content key encrypted with the content provider's public key
- Presenting the metadata to the content provider allows it to return the content key

Encrypted content: metadata {K}P

metadata: {K}P

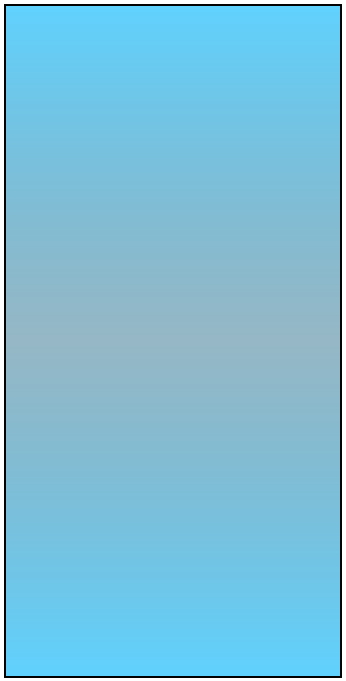


encrypted with K

content provider knows priv key, can decrypt {K}P and return K

Encrypted content: metadata content ID

metadata: content ID



encrypted with K

content provider has table (ID, K) for all items, can look up K

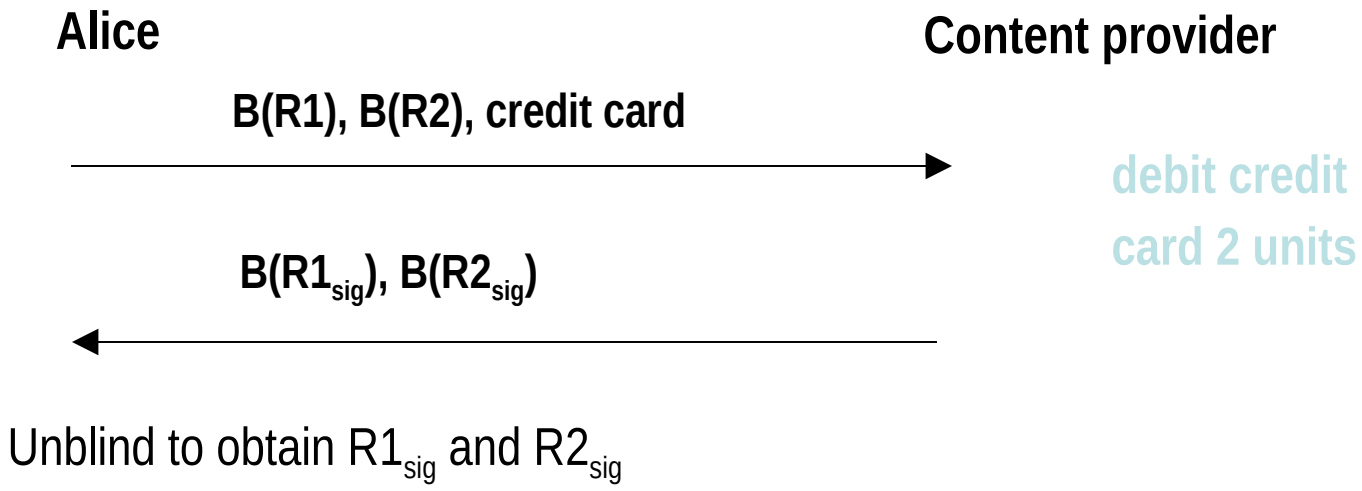
Both schemes use concept of "blinding"

- Alice wants Bob to sign or decrypt "x" with Bob's private key
- Alice creates functions (blind=B, unblind=U) that commute with Bob's public/private key operations
- Sends B(x) to Bob
- Bob applies private key
- Alice takes the result, applies U, to get signed or decrypted x

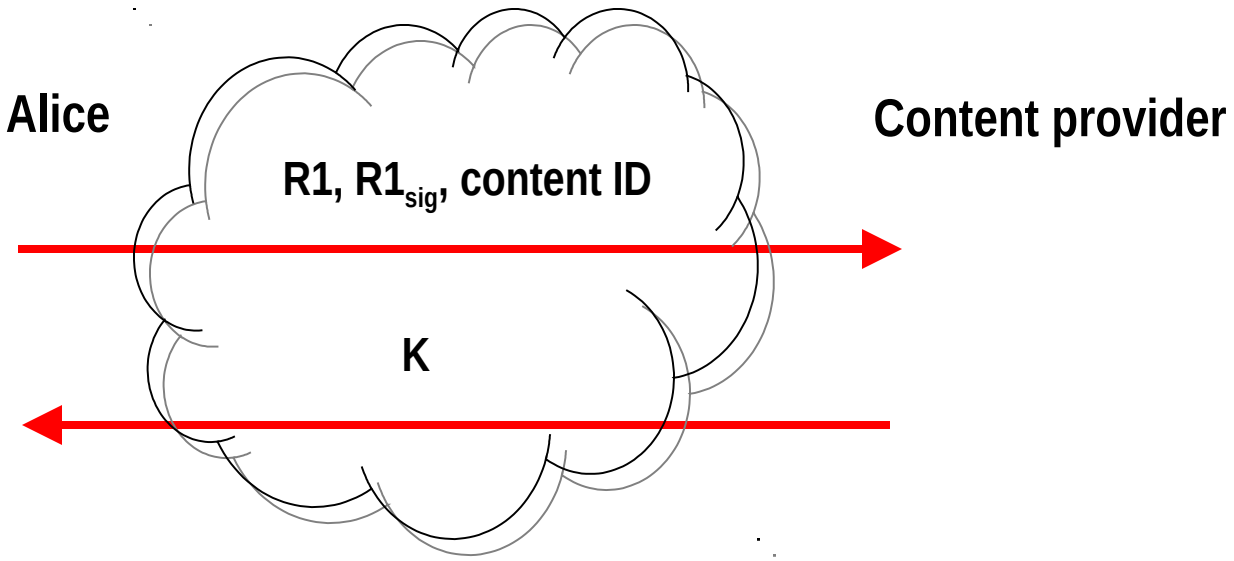
Anonymous Cash

- Chaum scheme for anonymous cash
- Choose random number R, “blind it”, send it to bank to sign, then unblind it. A “token” is R, and the signature on R, say R_{sig}
- Buying content
  - (non-anon) buy tokens, using real money
  - In an anonymous, encrypted conversation, present anonymous cash, ask for particular content key

Anonymous Cash Scheme: Buying tokens

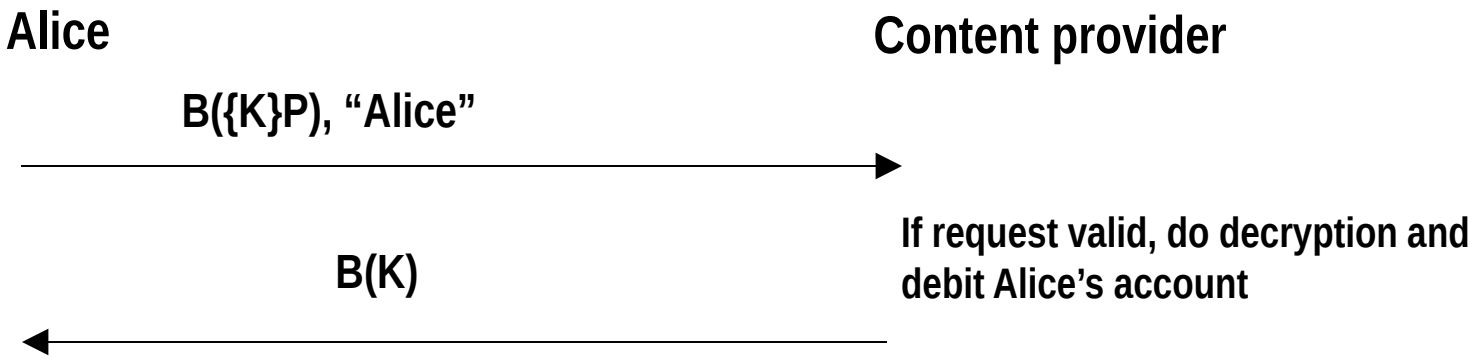


Anonymous Cash Scheme: Purchasing content



Anonymizing cloud, encrypted conversation

Scheme 2: Blind Decryption



Note: conversation must be signed by Alice, plus have timestamp

Comparisons

- Per-item accounting
  - Possible in anonymous cash scheme
  - Not possible in blind decryption scheme
- Efficiency (see next slide)

Blind decryption more efficient

- One conversation, vs anonymous cash
  - one to buy token
  - one by purchase content
- One private key operation for content provider, vs in anonymous cash
  - blindly sign token
  - establish server-side encrypted/authenticated session
- No need for anonymization cloud

First wrinkle: variable charging

Could be trivial with anon cash: present n tokens to buy something worth n units

That would require n private key operations for the content provider (actually 2n because of originally signing them)

Instead, can have different denomination tokens

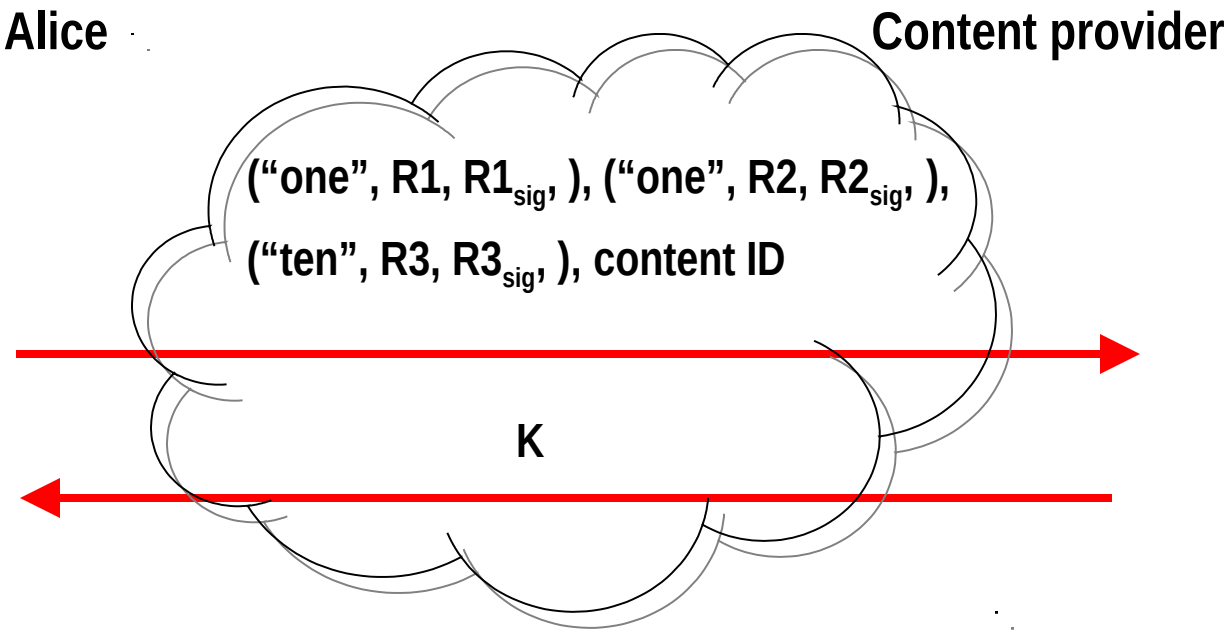
Variable charging: Anonymous cash

- Content provider has different “denomination” public keys, say P1=“one”, P2=“10”, P3=“50”
- When purchasing tokens, ask for denominations
  - I’m Radia
    - I’d like 4 ones: B(R1), B(R2), B(R3), B(R4)
    - And 2 tens: B(R5), B(R6)
    - And 3 fifties: B(R7), B(R8), B(R9)
  - Content provider applies P1 to first 4, P2 to next 2, P3 to next 3
- When (anonymously) purchasing content, provide all the necessary tokens

Anonymous cash, different denominations

- Might be suspicious to get anything more than a one, if all G-rated content was 1, and X-rated was more
- Allow purchase of multiple things in the same transaction, so asking for a large denomination bill isn't suspicious
- Besides you ***could*** purchase with all one's
  - content provider could discourage this paranoia by offering a discount for large denominations

Anonymous Cash : Purchasing content that costs 12 units



Anonymizing cloud, encrypted conversation

Variable Charging: Blind Decryption

- For an item costing 3 units, metadata would have 3 wrapped keys, K1, K2, K3, and content key is h(K1,K2,K3)
- Could also have different denomination content provider public keys, just like anonymous cash
- Metadata for something worth 12 units:
  - "one": {K1}P1, "one": {K2}P1, "ten": {K3}P2
- Request to content provider:
  - "one", B({K1}P1), "ten" B({K3}P2), "Alice"
- Can request the keys at different times

Variable Charging: Blind Decryption

- If Alice is nervous buying something worth more than 10 units, metadata could give the choice of unwrapping 12 individual keys or a 10 and 2 ones. Alice's choice
  - Could unwrap 12 ones, content key is XOR of all of those, or unwrap 2 ones and 1 ten, and content key is also XOR of those 3.
  - Content provider might provide discount for using larger denominations
- Note: the component keys for this content can be purchased at different times

Easy issue: Timing issue

- When something is first broadcast, it might be likely that someone asking for content at that time is buying that content
- So, provide the metadata well in advance

New topic: Additional Authorization

- Suppose you also have to prove “over 21”
- Several scheme, with slightly different properties.
  - authorization secrets used as encryption keys
  - authorization secrets used as credentials
  - different content provider public keys

Leaking of authorization secrets

- Obvious concern
- No matter how the secrets are used, what if they leak out?
- No harder to leak these, or to protect these, than content keys
- So we're assuming some sort of DRM, whether hardware or software
  - Note: software DRM "can't" be secure, but it is widely deployed

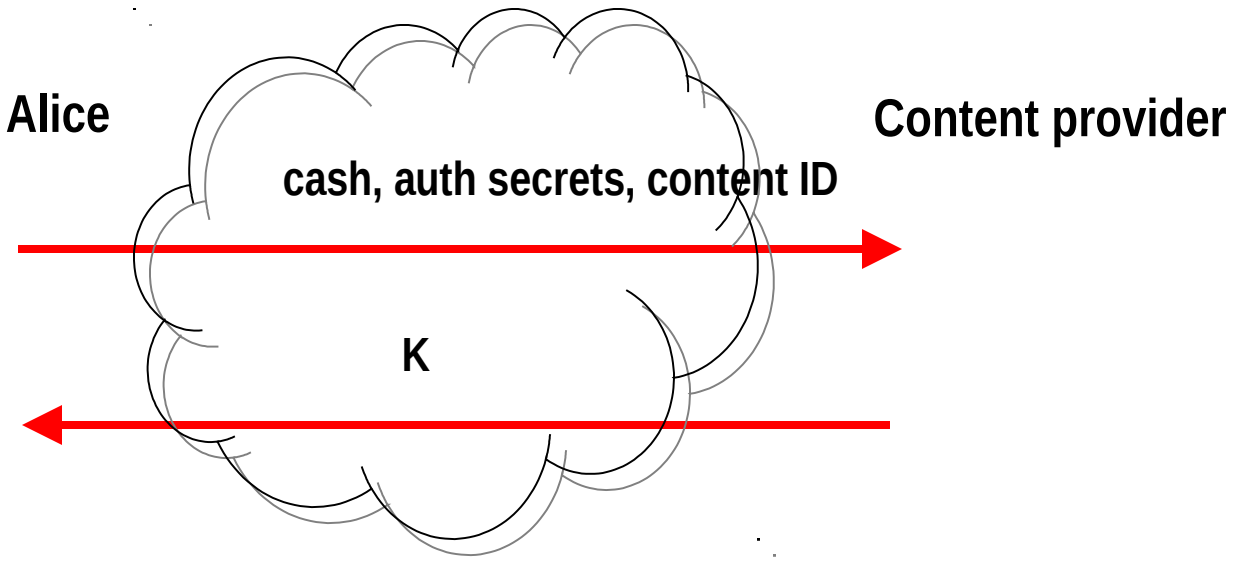
Authorization secrets used as keys

- Metadata would contain
  - ACL: “over 21”, “US”
  - {K}P (blind decryption), or content ID (anonymous cash)
- Alice has already (nonanonymously) obtained and saved K_{21}, and K_{US}.
- Content key would be h(K, K_{21},K_{US})
- Somewhat bulky metadata with the OR of attributes, but everything is doable

Auth secrets as credentials

- Only works with anonymous cash scheme
- Metadata would contain
  - ACL: “over 21”, “US”
  - content ID (anonymous cash)
- Alice has already (nonanonymously) obtained K_{21}, and K_{US}.
- Anonymous, encrypted request
  - K_{21}, K_{US} , content ID
  - Content provider checks ACL to make sure all necessary authorizations are proven, returns K

Anonymous Cash Scheme: Purchasing content



Anonymizing cloud, encrypted conversation

Complex policies

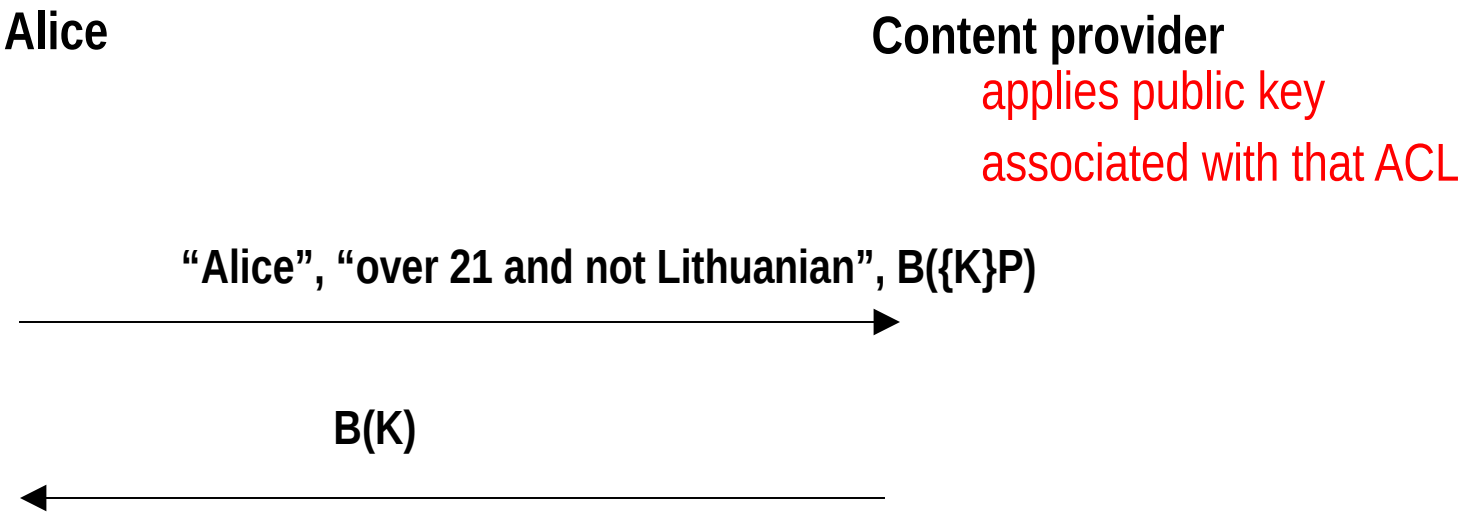
- Easy: ACL is part of metadata. Client figures out what is needed to satisfy it

Comparison

- Privacy issue
  - Could be there is only one Lithuanian with a PhD in Chinese literature with a plumbing license
    - auth secrets as credentials wouldn't be very anonymous then..
    - only relevant if ACL is complicated OR
- Revocation of authorization secrets
  - In credentials scheme, easy to change secret periodically
  - With auth secrets as keys, you'd have to re-encrypt the data

Third possibility: ACL-dependent content-provider keys

ACL-dependent public keys



Note: conversation must be signed by Alice, plus have timestamp

Content provider checks Alice profile to ensure she has attributes

ACL-dependent public keys; cash

- When asking for a token, specify which key (e.g., “US citizen”)
- When purchasing ACL-dependent content, use the relevant cash

Good, bad, and ugly of this variant

- Good
  - No need for authorization secrets
  - No worry about authorization secrets getting shared
  - Revocation of Alice's attributes very easy
- Bad
  - Content provider knows ACL of the content Alice is asking for; could be very few possibilities
  - But could wrap content with more atomic attributes
- Ugly (but not, with cute crypto)
  - Managing all these public keys

Unique ACL

- Could be only one piece of content that has the ACL “plumbing license AND alum of NYU”
- So instead, you could have two keys in the metadata, one wrapped with “plumbing license” public key, and one wrapped with “alum of NYU” public key, and content key is XOR of the two of them

How to do blindable, ACL-parameterized public keys

- Use Diffie-Hellman keys
  - works with elliptic curves, but I'll explain it with modular exponentiation, where it also works
- All Alice knows for content provider's key are the parameters "g" and "p"
- Content provider just needs a single secret, let's call it "S"

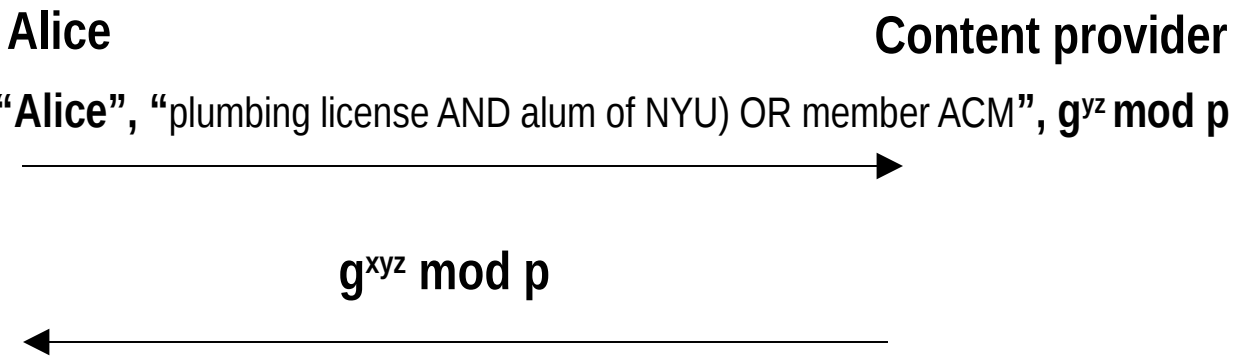
Content provider encrypts an item

- Choose a random number "y"
- Calculate $g^y \bmod p$
- hash S with the ACL, e.g., h(S, "(plumbing license AND alum of NYU) OR member ACM") = x
- Calculate $g^{xy} \bmod p$
- Content key is h($g^{xy} \bmod p$)

Alice wishes to purchase item

- Metadata
  - ACL: plumbing license AND alum of NYU) OR member ACM
  - g^y mod p
- Unblinded: Send all that metadata to content provider, which derives “x” from the ACL, and sends back g^{xy} mod p
- Blinded: Choose z, calculate z^{-1}, raise g^y mod p to z, send ACL and g^{yz} mod p

ACL-dependent public keys



Note: conversation must be signed by Alice, plus have timestamp

Content provider checks Alice profile to ensure she has attributes

Alice raises to z^{-1} to obtain g^{xy} mod p

Note

- Reminiscent of “identity based encryption”
- But it’s not: nobody but content provider can know either public or private key

IBE (Identity Based Encryption)

- This works as well, where the content provider knows the domain secret, its “public key” is the domain parameter, and the ACL is the string

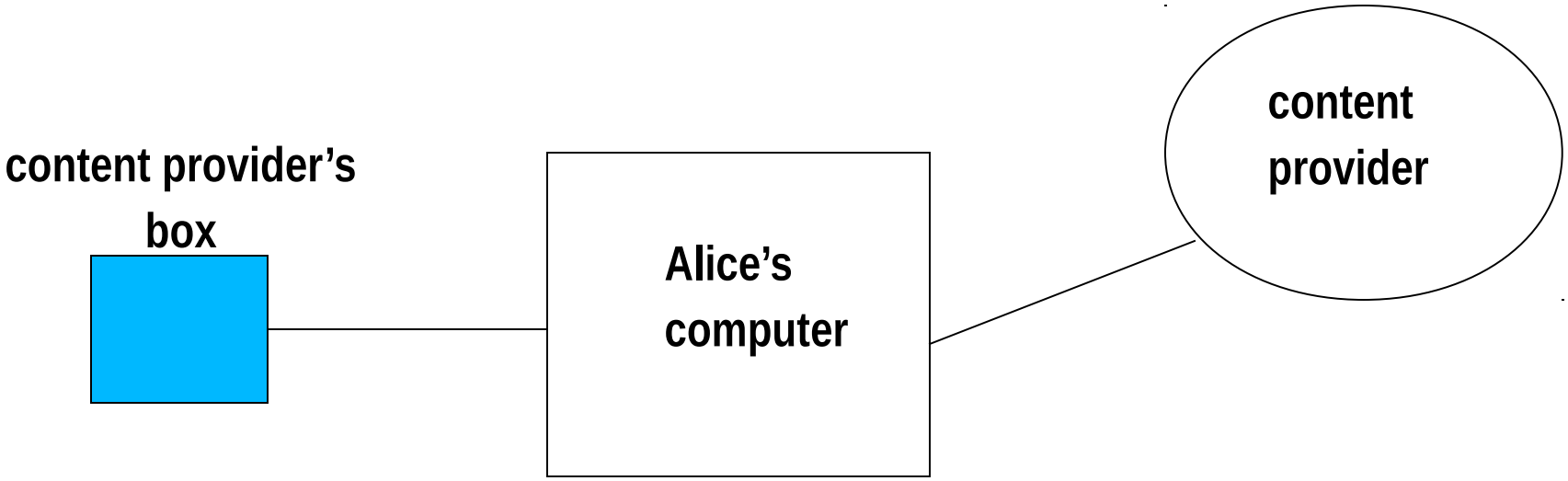
RSA

- This variant may work
- Would be nice to have a proof
- “Public key” is just modulus “n”
- public exponent is h(ACL string)
- “private key” is factorization of n

Most subtle wrinkle: Sealed box

- Common deployment scenario: sealed box at customer premises provided by content provider
- Communication is between box and content provider
- Customer can monitor communication, talk to box, intercept messages

Sealed box provided by content provider



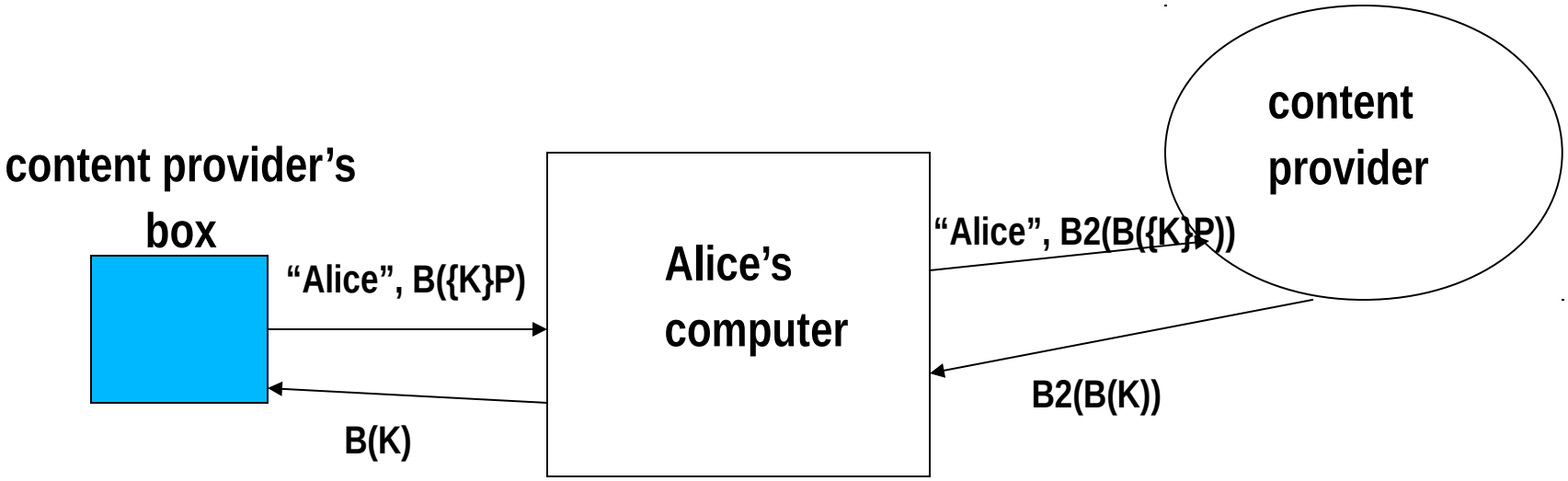
Can Alice tell if box is cheating, and leaking privacy information?

- Anonymous cash scheme
  - Absolutely not: communication between box and content provider must be encrypted with end-to-end key
  - Alice can't tell anything about the conversation
- Blind decryption
  - Looks sort of promising
  - Box says, in cleartext "timestamp, B(metadata)"

But box can cheat

- For instance, blinding function could be purposely weak, so that content provider can tell what content Alice is accessing
- No way for Alice to be able to detect this is going on
- But there's hope

Alice can add extra level of blinding



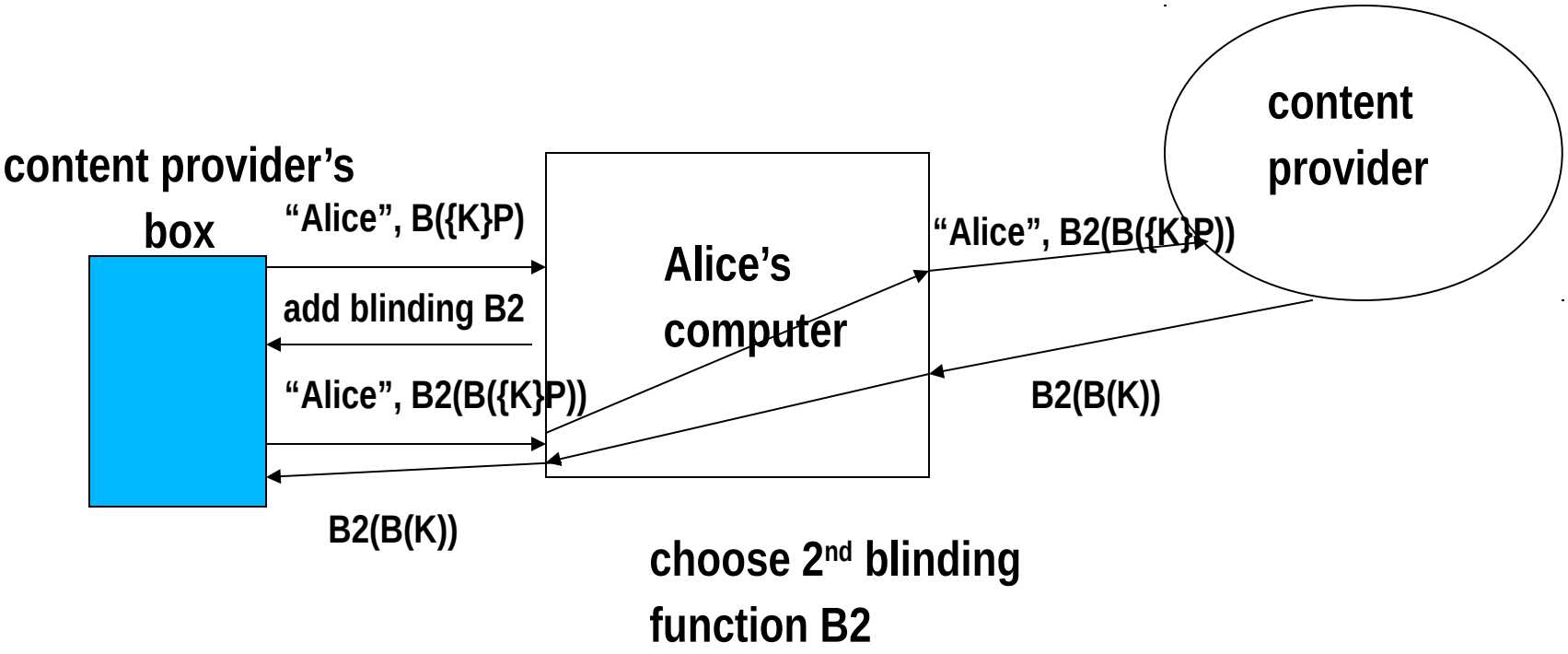
choose 2nd blinding function B2

Note: Alice can't cheat and get access to the content keys. Box can't cheat and tell content provider what key is being decrypted

But it doesn’t quite work

- Problem: There needs to be end-to-end authentication between the box and the content provider, because content provider wants it to be “impossible” to get content keys out of boxes.
- So Alice can’t modify messages between the box and the content provider.

Solution: Alice can tell box to add her chosen 2nd level of blinding



Note: Alice can't cheat and get access to the content keys. Box can't cheat and tell content provider what key is being decrypted

Summary

- Two basic schemes
  - anonymous cash
  - blind decryption
- Wrinkles
  - variable costs
  - supporting arbitrarily complicated ACLs
  - allowing Alice to cooperate with box to preclude covert channel

NIST Hash Competition:
Where we are and what we're learning

Bill Burr

NIST

April 14, 2010

NIST National Institute of Standards and Technology

Cryptographic Hash Function

- Hash functions take a variable-length message ***x*** and reduce it to a shorter fixed-length *message digest* ***hash(x).***
- Core requirement: Use ***hash(x)*** as a stand-in for ***x*** in digital signatures, MACs, file comparisons, etc.
- Many applications: "Swiss army knives" of crypto:
  - Digital signatures (with public key algorithms)
  - Random number generation
  - Key update and derivation
  - One way function
  - Message authentication codes & user authentication (with a secret key)
  - Code recognition (list the hashes of good programs or malware)
  - Commitment schemes and random oracles

NIST National Institute of Standards and Technology

Hash Functions Properties

- Collision Resistance
  - Hard to find x_1 *and* x_2, $x_1 \neq x_2$ such that $hash(x_1)=hash(x_2)$
    - By Birthday paradox, for *n*-bit hash, should take $2^{n/2}$ hashes to find a collision
  - Harder to get CR than we knew before 2004
    - Xiaoyun Wang: differential collision attacks on SHA-1
    - Attacker controls too much (no secret key)
- Preimage Resistance
  - Roughly means the hash is "one-way." That is, given *y*, hard to find *x* such as $y = hash(x)$.
    - For *n*-bit hash, should take 2^n hashes to invert
- Second Preimage Resistance
  - Given an x_1, hard to find $x_2 \neq x_1$, such that $hash(x_1) = hash(x_2)$.

NIST National Institute of Standards and Technology

Hash Function Standards

- MD5: 128-bits, badly broken in 2004
  - Never was a FIPS
- SHA-0: 160-bits designed by NSA 1990
  - FIPS 180, Quickly withdrawn, publicly broken in 1998
- SHA-1: 160-bits, tweak to SHA-0, 1995
  - FIPS 180-1, Wang attack in 2005
    - Now down to estimated 2^{52} work factor
- SHA-2: designed by NSA
  - 224, 256, 384 & 512-bit variants, FIPS 180-3
  - MD5, SHA-0, SHA-1 all in a design family, all broken. SHA-2 is a descendent: is it next?
    - Takes a long time to introduce a new hash function into use

NIST National Institute of Standards and Technology

SHA-3 Hash Competition

- Motivated by collision attacks on commonly used hash algorithms, particularly MD5 & SHA-1
  - No actual collisions yet announced on SHA-1
    - SHA-1 collision work factor may be as low as ≈ 2^{52} operations
      - McDonald, Hawkes and Pieprzyk, Feb 09
- Held 2 hash function workshops in 2005 & 2006
- Proposed criteria for new hash function Jan 2007
  - Many comments received
- "SHA-3" Competition announced Nov. 2, 2007

NIST National Institute of Standards and Technology

SHA-3 Competition Timeline

- ✓ 01/23/07 Draft submission criteria published
- ✓ 11/02/07 Federal Register announcement of SHA-3 Competition
- ✓ 08/31/08 Preliminary submissions due
- ✓ 10/31/08 Submissions due – 64 received
- ✓ 12/09/08 Announced 51 First round candidates
- ✓ 02/25/09 First SHA-3 Candidate Conference, Leuven Belgium
- ✓ 07/24/09 Announced 14 second round candidates
- ✓ 09/15/09 Tweaks accepted, second round began
- ✓ 08/23/10 – 08/24/10 Second SHA-3 Candidate Conference, UCSB
- ➢ 4Q10 Announce finalist candidates
- ➢ 1Q11 Final tweaks of candidates
- ➢ 1Q12 Last SHA-3 Candidate Conference
- ➢ 2Q12 Announce winner
- ➢ 4Q12 FIPS package to Secretary of Commerce

NIST National Institute of Standards and Technology

SHA-3 Participation

- SHA-3 Zoo – ECRYPT II/Graz University
  - Independent summary of cryptanalytic results
- eBash - ECRYPT Benchmarking of All Submitted Hashes
  - Systematic benchmarking of hash functions on many platforms
- ATHENa - Automated Tool for Hardware EvaluatioN
  - Inspired by eBash, but for hardware
  - George Mason Univ., Virginia Tech., Univ. Illinois Chicago
  - ARRA Grant

NIST National Institute of Standards and Technology

The SHA-3 Candidates

- Very international field & some are hard to tie to a particular country
  - US, Canada, China, Singapore, Japan, Korea, Argentina, India, Switzerland, Macedonia,Turkey, Israel, Belgium, France, Norway, Luxembourg and a number of “pan European” submissions
  - Now down to 14 second round candidates
    - 3 are (mainly) from the US
    - 1 each from Singapore, Japan, Israel & Turkey
    - The rest are mainly Western Europe (some are hard to attribute to one country)

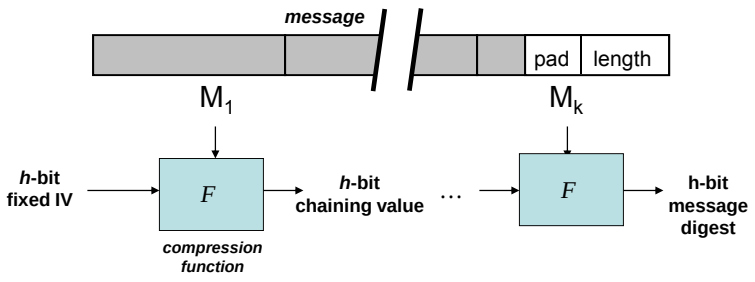
NIST National Institute of Standards and Technology

Second Round SHA-3 Candidates

- BLAKE
  - Swiss, HAIFA
- Blue Midnight Wish
  - Norway, WideP MD
- CubeHash
  - US, Sponge variant
- ECHO
  - France, HAIFA
- Fugue
  - US, Sponge variant
- Grøstl
  - European, WideP MD
- HAMSI
  - Turkey, MD
- JH
  - Singapore, novel construction
- Keccak
  - European, Sponge
- LUFFA
  - Japan, Sponge variant
- SHABAL
  - France, WideP MD
- SHAvite-3
  - Israel, HAIFA
- SIMD
  - France, WideP MD
- SKEIN
  - US, WideP MD (more or less)

NIST National Institute of Standards and Technology

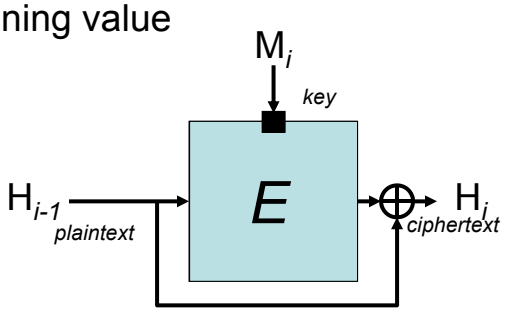
Merkle-Damgard Chaining Mode



NIST National Institute of Standards and Technology

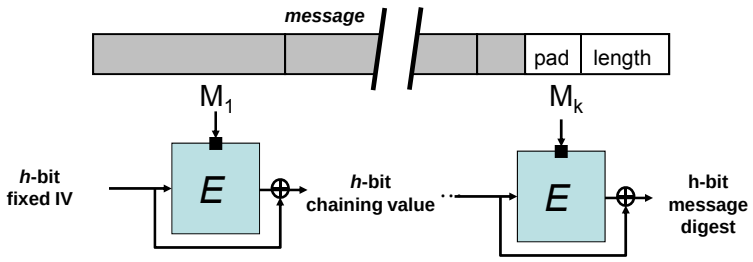
Davies-Meyer Compression Fun.

- Compression function from a block cipher
  - Message block is the key
  - Old chaining value is plaintext
  - New chaining value is ciphertext XOR old chaining value



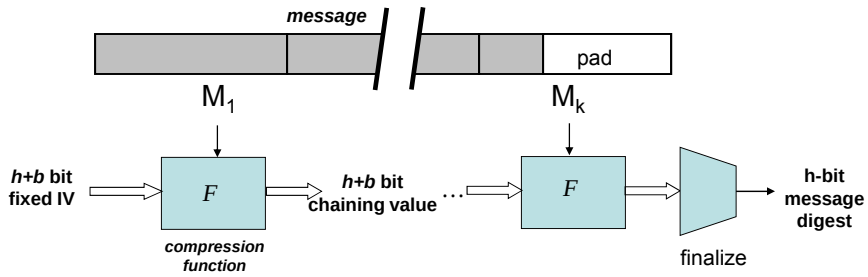
Merkle-Damgard Hash

- Figure illustrates a MD hash with a Davies-Meyer Compression Function



Wide Pipe Merkle-Damgard Hash

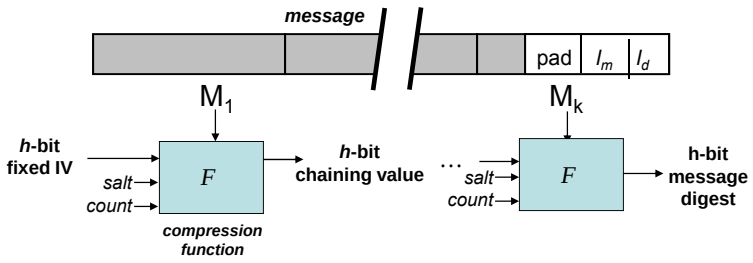
- Chaining variable bigger than hash output
  - greater preimage resistance for big messages
  - Prevents extension attacks



NIST National Institute of Standards and Technology

HAIFA - HAsh Iterative FrAmework

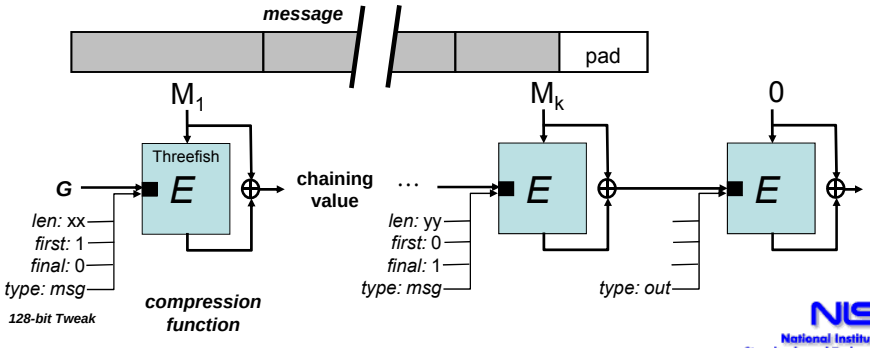
- Biham & Dunkelman
- Incorporates a salt & a bit count in each compression function call
- Shavite-3, ECHO & Blake are HAIFA



NIST National Institute of Standards and Technology

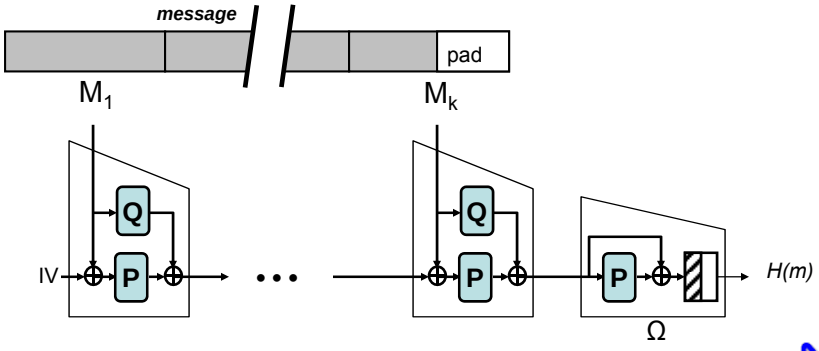
SKEIN – UBI Mode

- Threefish “tweakable” wide block cipher
- Matyas-Meyer-Oseas mode variant
- Message is plaintext, chaining var. is key
- Wide-pipe MD, more or less



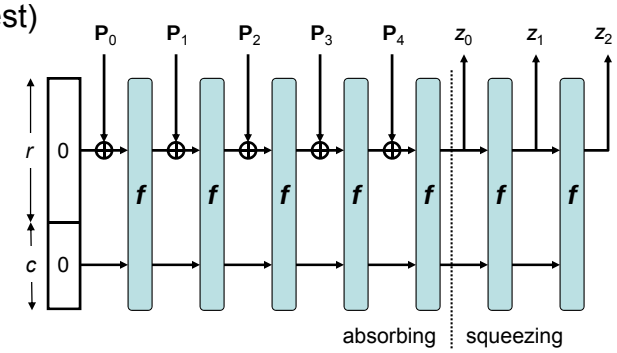
Grøstl

- Wide-pipe MD - truncated in output transformation, Ω
- P & Q are nearly identical fixed permutations
- P & Q use AES s-boxes



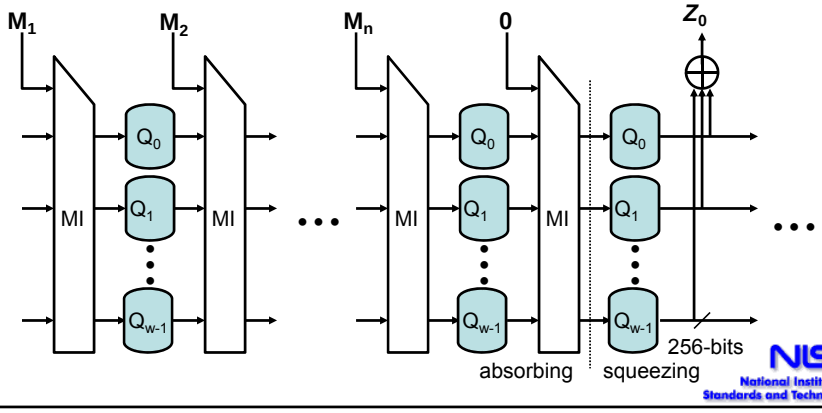
Keccak - Sponge Model

- Compression function is a fixed permutation
- XOR message into part of the chaining variable
- Think of sponge as a stream cipher – absorb the message, then squeeze out the keystream (message digest)



Luffa - Modified Sponge

- Separate linear message injection function, then 3 to 5 256-bit non-linear fixed permutations



Questions and Issues

- Performance issues
  - 32 vs 64-bit, low end vs high end, hardware vs software, parallelism: SIMD & MIMD, long messages vs short
- How important are proofs?
- Primitive reuse
  - AES s-boxes, AES round function, Threefish wide block cipher, cha-cha round, stream cipher
- Does any property above 2^{256} matter?
  - Time + memory? Greater of time or memory?

NIST National Institute of Standards and Technology

Some Comparisons

- Big biters vs lil' nibblers
  - Nibblers (64-bits or less): CubeHash, Hamsi, Fugue
  - Biters (256-1024 bits): all the rest & CubeHash
- Block cipher vs fixed permutation
  - Block Cipher: Skein, Shavite3, BLAKE, SIMD, Shabal, BMW?
  - Fixed Permutation: CubeHash, Luffa, Keccak, Grøstl, ECHO, Hamsi, Fugue, JH
- Unusual Chaining Mode: JH, Shabal
- One engine fits all sizes: Kekkac, CubeHash, JH

NIST National Institute of Standards and Technology

Confusion & Diffusion

- Since Shannon we've talked about "diffusion" and "confusion" in cryptography
- Roughly speaking
  - Diffusion means that one input bit affects the value of every output bit
    - typically get this from XOR & rotation
  - Confusion means that the relationship between inputs and outputs is algebraically complex ("non-linear") so we can't solve the equations

NIST National Institute of Standards and Technology

Non-linearity

- Compression function needs non-linearity or it would be easy to invert
  - Some submissions get non-linearity from S-boxes (substitution tables), often the AES s-boxes
    - Shavite-3, Grøstl, Luffa, Fugue
    - But table lookups are subject to cache timing effects resulting in side channels
      - Do side channels matter for hash functions?
  - Other submissions use only operations
    - AND, Add, & Multiply all contribute to nonlinearity
  - Of course you can generate substitution boxes with logic or circuits, or implement logic circuits with tables
    - May not be equally efficient or practical depending on the design and platform

NIST National Institute of Standards and Technology

Comparison: Non-linearity

	S-box		Logic	
Mode	**AES**	**Bit-slice**	**ARX**	**Bitwise**
Stream sponge	Fugue		CubeHash	
Block sponge		Luffa		Keccak
Wide MD	Grøstl	JH	BMW, Skein	Shabal, SIMD
Narrow MD		HAMSI		
Haifa	Shavite 3, ECHO		BLAKE	

NIST National Institute of Standards and Technology

Basic Operations of SHA-3 Candidates

	NTT	Linear code	S-box	GF MUL	MUL	mADD[1]	ADD/SUB	Boolean
BLAKE						mADD3	ADD	XOR
BMW						mADD17	ADD, SUB	XOR
CubeHash							ADD	XOR
ECHO			AES 8x8	x02, x03				XOR
Fugue			AES 8x8	x04..x07				XOR
Groestl			AES 8x8	x02..x07				XOR
Hamsi		LC[128, 16,70]	Serpent 4x4					XOR
JH			Serpent 4x4	x2, x5				XOR
Keccak								NOT:AND:XOR
Luffa			4x4	x2				XOR
Shabal					x3, x5		ADD, SUB	NOT:AND:XOR
SHAvite-3			AES 8x8	x02, x03				NOT:XOR
SIMD	NTT_{64}				x185,x233	mADD3	ADD	NOT:AND:OR
Skein							ADD	XOR
SHA-256						mADD5		NOT:AND:XOR

[1] mADDn – denotes a multioperand addition with n operands.

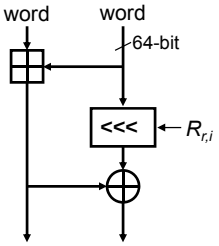
This chart is from a presentation by Kris Gaj & Jens-Peter Kaps of GMU

NIST National Institute of Standards and Technology

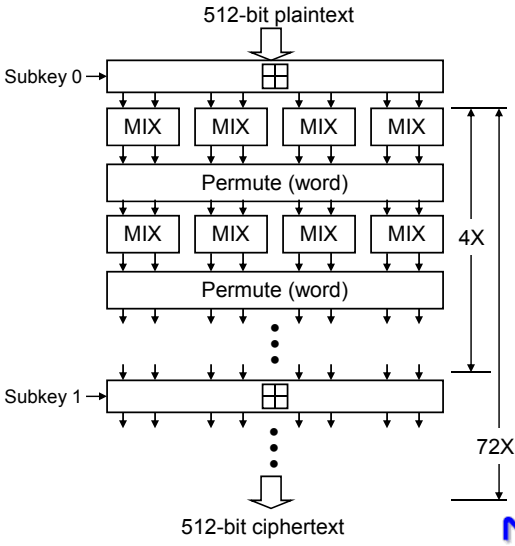
Non-linearity: Skein (ARX)

ARX: Add, Rotate, XOR

The non-linearity in the Threefish block cipher comes from many repetitions of the simple MIX operation

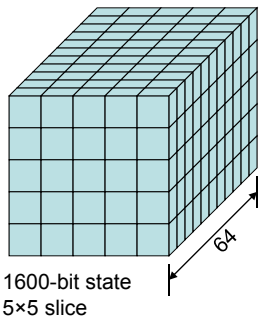


MIX operation

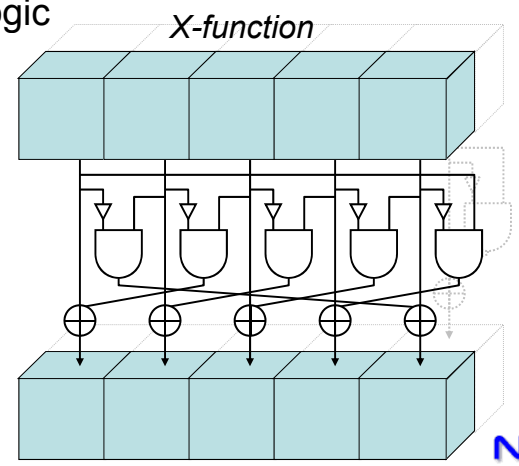


Non-linearity: Keccak (bitwise)

- All nonlinearity from one simple function
  - Only bitwise logic



1600-bit state
5×5 slice
64-bit lane (word)



Multiplicative Complexity

- We can
  - view crypto functions as logic circuits
  - make any circuit from AND and XOR
  - represent a circuit as an equation on GF(2), where AND is multiplication (•) and XOR is addition (+).
  - solve large systems of linear equations (only additions), but solving nonlinear equations (also do multiplication) is much harder
- Cryptography needs a lot of nonlinearity to make solving the equations computationally complex.
- How do we measure nonlinearity?
  - Multiplicative complexity is one answer

NIST National Institute of Standards and Technology

Multiplicative Complexity

- Can divide any circuit into a linear and nonlinear components
- But determining the minimum number of AND gates needed is very hard for nontrivial cases (super exponential)
- Find lowest known upper bound
- Peralta (NIST) and Boyar (SDU) have developed better heuristics for minimizing multiplicative complexity

NIST National Institute of Standards and Technology

Multiplicative Complexity

- Peralta-Boyer heuristics
  - Better logic synthesis ??
    - NIST/SDU patent application
    - Application to AES s-boxes resulted in simplest known circuit
    - Significantly speeded up 2 SHA-3 candidates
- Plan to apply heuristics to SHA-3 finalists
  - How many ANDs for each output bit?
    - Alternative view for cryptanalysis
    - Is anything just too simple?
  - Speed improvements

NIST National Institute of Standards and Technology

Links

- NIST Hash competition page: http://www.nist.gov/hash-competition
- eBASH benchmarking of hash functions: http://bench.cr.yp.to/ebash.html
- Wikipedia: http://en.wikipedia.org/wiki/SHA-3
- ECRYPT II – The SHA-3 Zoo: http://ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo
- Classification of the SHA-3 Candidates: http://eprint.iacr.org/2008/511.pdf
- SHA-3 Engineering Comparison (propaganda, but useful): http://www.skein-hash.info/sha3-engineering
- ATHENa - Automated Tool for Hardware EvaluatioN: http://cryptography.gmu.edu/athena/

NIST National Institute of Standards and Technology

Biometrics-Based Identifiers for Digital Identity Management

Abhilasha Bhargav-Spantzel
Intel Corporation
2191 Laurelwood Avenue
Santa Clara,CA 95054
abhilasha.bhargav-spantzel@intel.com

Anna Squicciarini
College of Information Sciences and Technology
Pennsylvania State University
University Park, PA
16802-6823
asquicciarini@ist.psu.edu

Elisa Bertino
Department of Computer Science
CERIAS
Purdue University
West Lafayette, IN 47906
bertino@cs.purdue.edu

Xiangwei Kong
Information Security Research Center
Dalian University of Technology
Liaoning Province 116023
kongxw@dlut.edu.cn

Weike Zhang
Patent Examination Collaboration Center 33
No. 18 South Fourth Street,Zhongguan Cun
Haidian District, Beijing
100190
zhangweike@sipo.gov.cn

ABSTRACT

We present algorithms to reliably generate biometric identifiers from a user's biometric image which in turn is used for identity verification possibly in conjunction with cryptographic keys. The biometric identifier generation algorithms employ image hashing functions using singular value decomposition and support vector classification techniques. Our algorithms capture generic biometric features that ensure unique and repeatable biometric identifiers. We provide an empirical evaluation of our techniques using 2569 images of 488 different individuals for three types of biometric images; namely fingerprint, iris and face. Based on the biometric type and the classification models, as a result of the empirical evaluation we can generate biometric identifiers ranging from 64 bits up to 214 bits. We provide an example use of the biometric identifiers in privacy preserving multi-factor identity verification based on zero knowledge proofs. Therefore several identity verification factors, including various traditional identity attributes, can be used in conjunction with one or more biometrics of the individual to provide strong identity verification. We also ensure security and privacy of the biometric data. More specifically, we analyze several attack scenarios. We assure privacy of the biometric using the one-way hashing property, in that no information about the original biometric image is revealed from the biometric identifier.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and protection; E.3 [**Data Encryption**]

IDtrust '10, April 13-15, 2010, Gaithersburg, MD

General Terms

Algorithms, Security, Experimentation, Human Factors

Keywords

Security, Privacy, Biometrics, Multi-factor Authentication, Identity, Cryptography

1. INTRODUCTION

To support online activities, such as commerce, healthcare, entertainment and scientific collaboration, it is crucial to be able to verify and protect the digital identity of the individuals involved. Misuse of identity information can result in identity theft, that is, the act of impersonating another's identity by presenting stolen identifiers or proofs of identities. Identity theft has been receiving increasing attention because of its high financial and social costs. An approach that can help in protecting from identity theft is the privacy-preserving multi-factor verification of identity[1]. Such a verification requires an individual to prove his/her identity by proving the knowledge of several *identity attributes* (also called *identifiers*). When talking about identifiers, we distinguish between *weak* and *strong identifiers*. A strong identifier uniquely identifies an individual in a population, whereas a weak identifier can be applied to many individuals in a population. The number and types of strong identifiers used in verification should not be fixed a-priori and each party interested in verifying the identity of an individual should be able to require any combination of such identifiers [3]. Biometric data represent an important class of identity attributes. To fully realize their potential, identity verification protocols should be able to support the use of biometric data in combination with other digital identifiers, such as a social security number (SSN) or a credit card number (CCN). The privacy of the biometric data and other sensitive identifiers should, however, be protected to mitigate attacks

[1]Effective solutions to protect from identity theft require a combination of technical and non-technical measures. Our approach represents one such measure which if used alone, however, may not be sufficient to address all possible threats to the security and privacy of identity information.

such as identity theft. By privacy of the biometric data we mean that minimal information about the biometric is revealed during the biometric verification process, and that this information cannot be reused in contexts outside a given biometric verification.

The use of biometric data in the context of identity attribute verification poses several non trivial challenges because of the inherent features of the biometric data. In general, two subsequent readings of a given biometrics do not result in exactly the same biometric template[2]. Therefore the matching against the stored template is probabilistic. Storing biometric templates in repositories along with other personally identifiable information introduces security and privacy risks [16]. Those databases can be vulnerable to attacks by insiders or external adversaries and may be searched or used for purposes other than the intended one. If the stored biometric templates of an individual are compromised, there could be severe consequences for the individual because of the lack of revocation mechanisms for biometric templates. To overcome the shortcomings of server-based storage and matching, several efforts have been devoted to the development of techniques based on client side matching [26, 27]. Such an approach is convenient as it is relatively simple and cheap to build biometric verification systems supporting biometric storage at the client end able to support local matching. Nevertheless, systems of this type are not secure if the client device is compromised; therefore additional security mechanisms are needed.

Client side verification systems has lead to research on key generation mechanisms that use biometrics [50, 48, 15, 26, 27, 58, 38]. A biometric key (BK for brevity) is never stored at any location and the key generation mechanisms should not allow the re-generation of the BK without the individuals' real biometrics. Note that under those approaches the biometric template is stored; therefore the verification does not involve biometric matching and instead uses the BK. Current techniques, however, are not sufficient because of several unresolved challenges concerning BK generation [35]. In particular, most BK generation approaches [24] do not differentiate between the cryptographic keys, used in the BK generation process, and the specific information retrieved from the actual biometrics. For example in [24] the BK is a repeatable string derived from a user biometrics. The final BK is essentially a pre-defined cryptographic key which can only be derived from information stored by the user and the users biometric information. As such the BK is never stored and cannot be derived without the users biometric information. Other approaches map biometric data into a unique and repeatable binary string [50, 48, 15, 26, 27, 58, 38]. Subsequently, the binary string would be mapped to an encryption key known as the BK by referring to a look-up table. In this work we focus on the repeatable binary string, referred to as the biometric identifier (BID), that is derived from the biometrics.

The goal of this paper is to identify the biometric information necessary and sufficient to generate a BID, which can in turn be used to generate a BK or simply as conventional strong identifiers such as SSN or CCN. To be used as strong identifiers, BIDs need to satisfy two key properties, namely uniqueness and repeatability. Uniqueness of BID ensures that two different individuals do not generate the same BID. If each individual is considered as a class in a given classifier model [22], then for uniqueness property to hold, the BIDs should have large inter-class variation. Repeatability of BID refers to the ability by an individual to re-generate his own BID (small intra-class variation). Another main challenge is to ensure the security and privacy of the biometric data. In particular, it should not be possible to re-create the BID without the original biometrics and the final BID should not leak information about the original biometrics. There are additional challenges with respect to the protection of the BID from brute force attacks conducted by exploiting meta-data stored at the client. As such several well-known solutions to the problem of BK generation have shown to be vulnerable to this threat [35].

[2]The digital representation of a biometric is referred to as *biometric template*.

We develop an approach that does not need to use specific features of the biometrics. We in fact use generic properties of biometric images that are shown to be suitable for multimodal biometric systems [45]. Multimodal biometric systems utilize more than one physiological or behavioral characteristic for enrollment and verification. This is an original contribution of our work as most of today's approaches are designed for a specific biometrics and cannot be trivially generalized to other biometrics. Additionally in the current approach, we depend on cryptographic keys in combination with the biometric data to preserve the privacy of the biometric during biometric verification.

Our Approach. The method for generating BIDs from biometric measurements is characterized by two phases [38]. During the first phase the biometric features are analyzed and used to compute a bit string representing these features. Such bit string should have uniqueness and repeatability properties. The bit string is then used in the second phase to generate a unique BID with the help of some meta-data. If two instances of the bit strings are sufficiently similar, then the BID generated is the same.

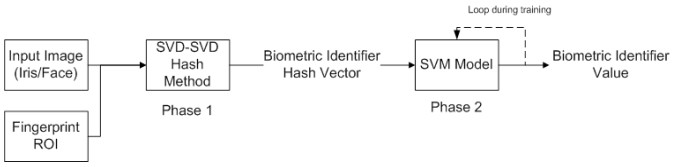


Figure 1: Two main phases of the biometric key generation.

In our approach, in Phase 1, a biometric hash vector is generated. Such biometric hash vector is a bit string which represents the biometrics and is obtained from the biometrics through an image hashing algorithm based on Singular Value Decomposition (SVD) (see Figure 1). In Phase 2, a classifier model based on Support Vector Machines (SVM) is used to classify and rank the resulting biometric hash vector. More specifically, the resulting biometric hash vector is classified to obtain a combination of classes which represent the user's unique and repeatable BID. The meta-data needed to execute Phase 1 and 2 consists of the classifier model and the pseudorandom secrets involved in the hashing algorithm.

The final BID generated at the end of Phase 2 is used for multi-factor identity verification. Identity verification based on the use of BIDs can be executed according to different strategies. For example the BID can be used as a password or as an attribute embedded in a digital certificate. In our approach we focus on the use of BIDs in the context of a privacy-preserving multi-factor cryptographic protocols for identity verification [3]. More specifically such protocol is based on the notion of *proof of identity* which consists of a cryptographic token bound to an individual, versus the actual value of the individuals' identity attribute. A proof is created so that only the individual to whom the proof is bound can properly

use it. Proofs of identity attributes are built using zero knowledge proof of knowledge (ZKPK for brevity) techniques [6, 18]. Efficient mechanisms have been developed to prove the knowledge of multiple strong identifiers stored as cryptographic commitments using aggregated ZKPK protocols [3].

In our approach the BID is used for identity verification based on ZKPK. The BID is used together with a random secret r to generate a Pedersen commitment [9]. This commitment is used to construct a ZKPK proof. This proof is sufficient for verification purposes as it corresponds to the biometrics enrolled in the system. The commitment is enrolled with a party and can be used by any verifying party. The use of ZKPK proof enables us to support *two-factor* (i.e. the BID and the secret random r) verification. At the time of verification the individual needs both to provide r and to reconstruct the BID, to prove knowledge of the value committed at enrollment. To revoke a BID, the commitment corresponding to enrolled biometrics is added to a revocation list which is similar to certificate revocation lists [25] in a public key infrastructure. In our approach, we consider the case where a revocation list consists of the biometric commitments which have been revoked. After a commitment has been published in the revocation list, the individual cannot do a proof of knowledge with that BID because it relies on a revoked commitment.

Contributions. The key contributions of the paper are as follows. First we present algorithms for reliable and secure generation of BIDs from different types of biometrics. We focus on techniques that are suitable for fingerprints, irises and faces. Second, we propose an approach for encoding BIDs into cryptographic biometric commitments that are used in ZKPK at the time of verification. It follows from the zero-knowledge proof protocols that the cryptographic proofs do not leak information except for the fact that the verifier learns that the prover verifies the proof. As such the verifying party obtains no information about the characteristics of the real biometrics from the cryptographic proof. Therefore, multi-factor verification techniques can use one or more biometrics interoperably with one or more non-biometric features to achieve strong identity verification. Our protocols ensure that the privacy of the biometrics is preserved as the final BID does not reveal any information about the original biometric image. We also present a detailed security analysis of the resulting biometric verification system. We provide an empirical analysis of the biometric key generation for different types of biometrics in order to provide evidence of the correctness of the proposed algorithms. Finally, we briefly discuss several use scenarios for our techniques to identify relevant infrastructural and organizational requirements for the use of our technique.

The rest of the paper is organized as follows. In Section 2 we introduce the main algorithms for the BID generation. In Section 3 we present the experimental results. In Section 4 we develop a comprehensive analysis of the proposed solution. In Section 5 we discuss related work. Finally in Section 6 we make some concluding remarks and additional considerations concerning the use of our approach.

2. BIOMETRIC KEY GENERATION ALGORITHMS

In this section we first introduce some preliminary concepts related to the techniques underlying our proposed solution. Then, we discuss the two core algorithms for the BID generation, that is, the SVD based image hashing algorithm and the SVM classification algorithm.

2.1 Preliminary Concepts

Singular Value Decomposition (SVD). SVD is a well known technique for factorizing a $m \times n$ matrix into a diagonal form. As proven by Golub and Loan [23], if A is a real *m-by-n* matrix, two orthogonal matrices exist:

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m \times m} \quad V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$UAV^T = diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n} \qquad p = min\{m, n\}$$

where V^T is the transpose of matrix V and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$. σ_i's, $i = [1 \ldots p]$, are the *singular values* of A, and the vectors u_j, $j = [1 \ldots m]$, and v_k, $k = [1 \ldots n]$, are the jth *left singular vector* and the kth *right singular vector* respectively. $\sigma_i(A)$ denotes the ith largest singular value of A.

The singular values of a matrix A are unique. The singular values σ_i's reflect the variations along the corresponding i singular vectors. It can be shown that computation of the right singular vectors and the singular values can be obtained by computing the eigenvectors and eigenvalues of the symmetric matrix $M = A^T A$ where A^T is the transpose matrix of A.

Support Vector Machines (SVM). SVM [22] is a classifier based on statistical learning technique developed by Vapnik *et al*. [13]. It aims at finding optimal hyperplanes to determine the boundaries with the maximal margin separation between every two classes while training the classifier model. Then additional data, which is not used during the training, is used as test data and can be classified using the separate hyperplanes.

Let $\{x_i, y_i\}$, $i = [1, \ldots, L]$, be a training data vector, where x_i is the data item and y_i, $y_i \in \{-1, +1\}$ is a class label. Given an input vector x, SVM constructs a classifier of the form

$$f(x) = Sign(\Sigma_{i=1}^{L} \alpha_i y_i K(x_i, x) + b)$$

where: α_i, $i = [1, \ldots, L]$, is a non-negative Lagrange multiplier; each multiplier corresponds to an example from the training data; b is a bias constant; and $K(\cdot, \cdot)$ is a kernel function satisfying the conditions of Mercer's theorem [53]. Some frequently used kernel functions are the polynomial kernel $K(x_i, x_j) = (x_i \cdot x_j + 1)^d$ and the Gaussian Radial Basis Function (RBF) $K(x_i, x_j) = e^{-|x_i - x_j|^2 / 2\gamma^2}$. Note that there are several approaches adopting SVM for classification problems with three or more classes as well.

SVM applies to classification of vectors, or uni-attribute time series. To classify multi-attribute data, which are matrices rather than vectors, the multi-attribute data must be transformed into uni-attribute data or vectors. We use the combination of the SVD technique with SVM which has been explored by previous work [31, 37, 55]. SVD is used to reduce multi-attribute biometric data to feature vectors.

2.2 SVD Image Hashing

In this section we describe the hashing mechanism used in Phase 1 of BID generation. The techniques presented build on the basic image hashing process described in [30]. The main steps of the algorithm (summarized in Figure 2) are as follows.

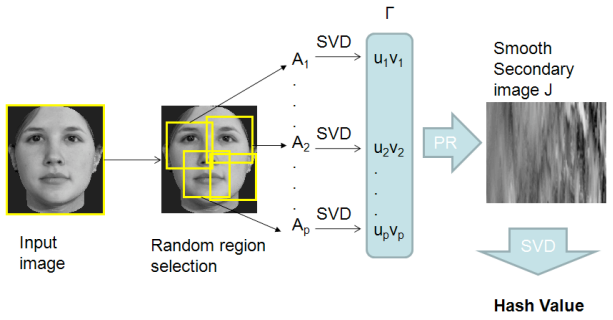


Figure 2: Key steps of the biometric image hashing algorithm.

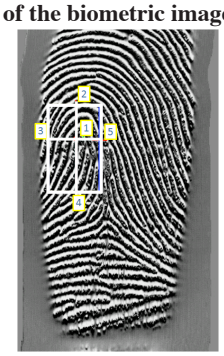


Figure 3: Fingerprint region of interest.

Pre-processing. As a first step the biometric image may be pre-processed so as to obtain a clear well focused biometric image I. Pre-processing provides an effective region in a selected biometric image for subsequent feature extraction. We support three types of biometric data: face, iris and fingerprint.

For the specific case of fingerprint image, as a part of pre-processing, the *region of interest* (ROI) is identified (See step 2 of Algorithm 1). The unique characteristics of the fingerprint are known to be around the core point or delta point [54]. The outside portion of a fingerprint is generally prone to small translations and is typically cropped out. Also, a larger area of the central portion of fingertip skin is in contact with the scanner surface as compared to the peripheries, giving a better image. The center is also better for liveness analysis. Since data such as the rate of perspiration can be measured, the center region is also more robust to pressure dispersion as compared to the other regions. Importantly, as the experimental results show, it preserves enough information to identify individuals. The procedure to determine the ROI corresponds to steps 6-15 of Algorithm 1 (see Figure 3). This ROI is then used as an image input for the rest of the algorithm (step 15 of Algorithm 1).

Feature Extraction. Once the image I of size $n \times n$ is finalized, the features are extracted based on a random region selection. The selection is executed by choosing p semi-global regions based on a pseudorandom (PR) generator that uses a secret key r. The obtained matrices corresponding to the selected sub-images (denoted by ρ_i) are then transformed under matrix invariant functions such as SVD.

The random partitioning of the image introduces unpredictability in the hash values and hence increases the security of the overall system. As long as these sub-images are sufficiently unpredictable, the resulting intermediate hashes are also different with high probability [36]. The squares ρ_i's determined in steps 18–23 and used in the partitioning (see Figure 2) are deliberately chosen to be overlapping to further reduce the vulnerability of the algorithm to malicious tampering. Note that an increased number of squares increases the pseudorandomness in the resulting hash value, and therefore helps in increasing security as explained in Section 4, assuming a secure pseudorandom number generator. As a further advantage, the random partitioning decreases the probability of collision and increases the robustness against noise that may be present in the biometric image. As reported in line 22 of Algorithm 1, the A_i's, $1 \leq i \leq p$, are matrices corresponding to the selected sub-image blocks. Here each element of the matrix A_i corresponds to the 256 grey level value of the pixel of the selected sub-image. The encoding of the actual matrix used in the transformation is done based on the fact that every element in the matrix has a grey value g, $0 \leq g \leq 255$, a position v and a direction d. A single pixel may not have a direction, but for a group of pixels, the grey value may change hence defining a concrete direction. Grouping pixels is important as isolated components may not be robust.

Transformation. Each sub-image $A_i, 1 \leq i \leq p$, is used to perform the SVD transformation. As a result for each A_i a unitary reduction to the diagonal form is performed to obtain $U_i S_i V_i, 1 \leq i \leq p$, such that $A_i = U_i S_i V_i^T$. As such the SVD selects the optimal basis vectors in the L_2 norm[3] sense such that, for any $m \times m$ real matrix A_i, we have

$$(\sigma_k, \overrightarrow{u_k}, \overrightarrow{v_k}) = arg\ min_{a,\overrightarrow{x},\overrightarrow{y}} |A - \Sigma_{l=1}^{k-1} \sigma_l \overrightarrow{u_l} \overrightarrow{v_l}^T - a\overrightarrow{x}\overrightarrow{y}^T|_F^2$$

where: $1 \leq k \leq m$; $a \in \mathbb{R}$; $\overrightarrow{x}, \overrightarrow{y} \in \mathbb{R}^m$; $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_m$ are singular values, $\{\overrightarrow{u_i}\}$ and $\{\overrightarrow{v_i}\}$, $1 \leq i \leq p$, are the corresponding singular vectors; and $(\cdot)^T$ is the transpose operator [30]. By using the SVD we preserve both the magnitude of the important features in singular values and also their location geometry in the singular vectors. The combination of the left most and right most singular vectors which correspond to the largest singular values, in turn, captures the important geometric features in an image in the L_2 norm sense. Therefore as a next step for each A_i, $\overrightarrow{u_i}$, that is, the first left singular vector and $\overrightarrow{v_i}$, that is, the first right singular vector are retrieved. Those vectors are then combined in $\Gamma = \{\overrightarrow{u_1}, \ldots, \overrightarrow{u_p}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$.

The next step is to form a pseudorandom (based on pseudorandom numbers) smooth secondary image J from Γ. J is formed according to an iterative process, at each step of which an element from Γ is selected and added to J. As a first step an element is pseudorandomly selected from Γ and set at the first column of J. Then for the i^{th} column of J, an element from Γ is selected such that it is closest to the $(i-1)^{th}$ column of J in the L_2 norm sense as denoted in step 39 in Algorithm 1. An element can only be chosen once from Γ, therefore an element chosen at the i^{th} step cannot have been chosen at any of the previous $(i-1)^{th}$ steps. Hence after $2p$ steps all the elements of Γ are pseudo-randomly reordered to form the secondary image J of size $m \times 2p$. Note that the secondary image is required to ensure the one-way property of the SVD image hashing algorithm (See the analysis in Section 4).

Once J is formed, SVD is re-applied to it, to finally obtain the image hash vector (steps 49 – 52 of Algorithm 1). The left and right singular vectors are obtained by $J = U_J S_J V_J^T$. Then the

[3] L_2 norm, defined for a vector $\overrightarrow{x} = \{x_1, \ldots, x_n\}$ is denoted by $|\overrightarrow{x}| = \sqrt{\sum_{k=1}^{n} |x_k^2|}$.

singular vectors corresponding to the largest singular values, that is, the first left ($\overrightarrow{u_J}$) and the first right ($\overrightarrow{v_J}$) are chosen. These vectors are simply combined to obtain the final hash value $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$.

2.3 SVM Classification

As discussed in the previous section, from one input biometric sample, a hash vector $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$ of length $m + 2p$ is obtained. Since the hash vectors obtained from different biometric samples of the same user may be the same or may differ from sample to sample, we train a classifier to determine which hash values correspond to a given user (or class), so that at the time of verification, the classifier can identify the correct class of the user. To achieve this goal several biometric samples of different users are taken. Algorithm 1 is run on each sample to get the corresponding hash vector.

These samples are then divided into training and test data to perform the classification. We use K-fold cross-validation to divide the training and testing data. All sample hash vectors are partitioned into K subsamples. Of the K subsamples, a single subsample is retained as the validation data for testing the model, and the remaining K - 1 subsamples are used as training data. The cross-validation process is then repeated K times (the folds), with each of the K subsamples used exactly once as the validation data. The K results from the folds are then averaged to produce a single estimation [2].

The obtained hash vectors do not greatly differ with respect to the Euclidean distance, as inferred through experimental analysis; therefore we use SVM techniques to map the input hash vectors onto a higher dimensional space where a maximal separating hyperplane can be constructed.

As explained in Section 2.1 the hyperplane constructed using SVM is such that it has the maximum distance to the closest points of the training set. These closest points in the training set are called *support vectors*. Here we use the Gaussian radial basis kernel function (RBF for brevity) $K(\overrightarrow{H}_i, \overrightarrow{H}_j) = e^{-|\overrightarrow{H}_i - \overrightarrow{H}_j|^2/2\gamma^2}$ where $\overrightarrow{H}_i$ and $\overrightarrow{H}_j$ are two of the training samples and $\gamma > 0$.

During training, two specific parameters have to be assessed, namely γ used in the RBF kernel function and the penalty parameter C used in the evaluation of an optimal hyperplane balancing the tradeoff between error and margin. To select the pair with the best CV accuracy, all combinations of C and γ are tried using a grid search method [8]. After training, the SVM model encodes all the classes that this SVM classifier has been trained with.

Note that an increased number of classes increases the number of choices for an attacker executing guessing attacks on the SVM model, to guess the right BID. Additional classes can be added to the original SVM classifier model by training additional samples of the given biometrics. These samples have to be carefully added as the added classes, which do not resemble the original biometric classes, would most likely be easily ruled out by an attacker. We therefore employ a strategy to make the additional classes similar to the original set of classes. For each class in the SVM model we define a *protector class* which is similar to the original class so that the cluster formed by the *protector class* is close to the original SVM class, and yet is different enough to be distinguished as a different class. There could be different ways of obtaining the protector classes. The first is to find biometric images of different individuals which look perceptually similar. The second possibil-

Algorithm 1 Generic Biometric Image Hashing Algorithm

Require: Biometric image I
Ensure: The quality of the image is suitable based on biometric.
Input biometric image I
{**Pre-process fingerprint images to calculate ROI**}
if (type(I) == 'fingerprint') **then**
  $point_1$ = Algorithm_R92(I) {Compute core or delta point}
  size = 4 {Set fingerprint ROI threshold size}
  count = 0
  for each line i in orthogonal directions (N,S, E, W) **do**
    repeat
      increment length of line;
      if line encounters a ridge **then**
        $point_i$ = coordinate of intersection of line and ridge
        count++
      end if
    until (count $\neq$size)
  end for
  I = crop($point_2, point_3, point_4, point_5$)
end if
Let resultant image $I \in \mathbb{R}^{n \times n}$ be of size $n \times n$
{**Random Selection**}
Let p be the number of rectangles
Let ρ_i be the i^{th} rectangle and m be the height/width of ρ_i.
for each i where $1 < i < p$ **do**
  Randomly position rectangle ρ_i at (x_i, y_i) such that $x_i + m < n$ and $y_i + m < n$
  Let A_i be the "sub-image" that is formed by taking the portion of image that is in $\rho_i : A_i \in \mathbb{R}^{m \times m}, 1 \leq i \leq p$.
end for
{**First SVD Transformation**}
for each A_i where $1 \leq i \leq p$ **do**
  $A_i = U_i S_i V_i^T$ {Collect singular vectors corresponding to the largest singular value}
  $\overrightarrow{u_i}$ = first left singular vector
  $\overrightarrow{v_i}$ = first right singular vector
end for
$\Gamma = \{\overrightarrow{u_1}, \ldots, \overrightarrow{u_p}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$
Initialize secondary image $J[m, 2p]$ {Constructing secondary image from singular vectors}
for all c where $1 \leq c \leq 2p$ **do**
  Initialize variable e_c corresponding to element in Γ
  if $c = 1$ **then**
    $e_c = PR\_Select(\Gamma)$
  else
    var_loop = true
    while var_loop **do**
      $e_c = min_{k=1}^{2p}(\sqrt{\sum_{l=1}^{c-1}(J(l) - \Gamma(k))^2})$
      if not(e_c already chosen for J) **then**
        var_loop=false
      end if
    end while
  end if
  for all r where $1 \leq r \leq m$ **do**
    $J[r][c] = e_c[r]$
  end for
end for
{**Second SVD Transform**}
$J = U_J S_J V_J^T$ {Collect singular vectors corresponding to the largest singular value}
$\overrightarrow{u_J}$ = first left singular vector
$\overrightarrow{v_J}$ = first right singular vector
$\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$
return Hash Value $\overrightarrow{H}$

ity is to add noise to the original biometric image. For example, the face images could be modified to render naturally asymmetric features to symmetric or changing other specific aspects as the size of the face characteristic such as the eyes, nose and so on. If there are n original classes, then we add a protector class for each, thus resulting in $2n$ classes. We also add other spurious classes which are not similar to the original biometric samples (as the protector classes) but are of the same biometric type.

As a final step, a combination of the classes is chosen based on SVM ranking which provides class prediction confidence of the SVM classifier. More specifically if n is the total number of classes, the final BID is the label of class with the highest confidence label and an unordered combination of the top $t = \frac{n}{2}$ class labels which are listed with decreasing confidence levels. For an attacker to guess the BID, given the SVM classes, the number of choices is $n + \binom{n}{t}$ resulting in the final number of bits as $\log_2(n + \binom{n}{t})$. Considering the FAR for the primary class the final number of bits would be $MIN[\log_2(n), -\log_2(FAR)] + \log_2(\binom{n}{t})$. We typically consider the total number of classes $n > 69$ which leads the number of choices to be $> 2^{64}$, thus making it computationally hard for the attacker to guess the right BID.

3. EXPERIMENTS

In this section we summarize the experimental results we conducted to assess the accuracy and robustness of our approach. We carried out extensive tests for different biometrics, to demonstrate that the relevant criteria required for the security, repeatability and uniqueness of the BID are met. All experiments have been conducted using Microsoft Windows XP Professional 2002 Service pack 1 operating system, with Intel(R) Pentium(R)4 3.20GHz and memory of 512MB.

3.1 Dataset and Experimental Setup

We tested our hashing algorithm (Algorithm 1, Section 2.2) on fingerprint, iris and face data. Summary information about the data used and the obtained results is reported in Table 2. For fingerprints we used FVC [34] databases. The FVC dataset used consists of overall 324 fingerprint images of 59 individuals collected using thermal sweeping and optical sensors. We also used 50 images of 10 individuals generated using the synthetic fingerprint generator SFingeGe v3.0 [7]. Regarding the iris data, the UBIRIS iris Database3 [44] was used which consists of 1695 images of 339 individuals' eyes. Finally for the face data we used the Yale Database of Faces [20] containing 100 images of 10 individuals and the AT&T Database of Faces [1, 46] containing 400 images of 40 individuals. We evaluated our results using the SVM classification algorithm, with K-fold cross validation (CV). Based on the CV accuracy, the False Acceptance Rate (FAR) was calculated. The FRR is calculated as $1 - CV\ Accuracy$, whereas the FAR is calculated as the number of false accepts divided by the number of tries.

The values used in the experiments for the key parameters of Algorithm 1 are reported in Table 1, where n is the size of the image in pixels, p is the number of sub-images, m is the size in pixels for each of the sub-images, and J is the secondary image.

To assess the optimal values for p and m, we ran experiments with various possible combinations of the values and used the one which provided the maximum accuracy. For example for the fingerprint database FVC2004 DB3_B, the value of p was varied between $[10, \ldots, 100]$ and the value of m between $[10, \ldots, 100]$ (See Figure 4); the highest accuracy was found for $p = 50$ and $m = 30$.

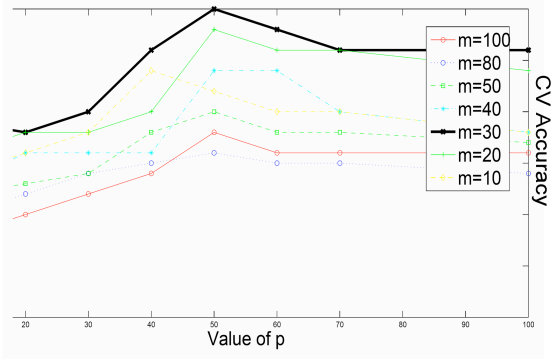


Figure 4: Plot of different values of number of sub-images (p); the image size of sub-images (m); and the corresponding CV accuracy.

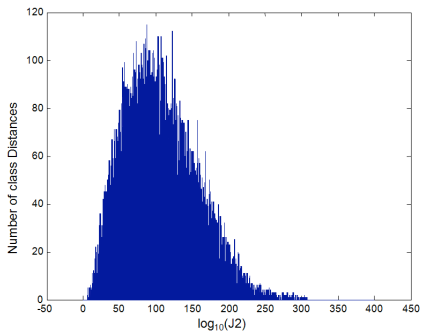


Figure 5: J2 histogram of iris classification.

The code for implementing the various steps is written in MATLAB and the `rand()` function of MATLAB is used as the pseudo random function used in step 21 and 35 of Algorithm 1. The size of the secondary image J is 30×100 leading to the size of $\overrightarrow{uJ} = 30 \times 1$ and $\overrightarrow{vJ} = 100 \times 1$, thus resulting in a hash vector $\overrightarrow{H} = \{\overrightarrow{uJ}, \overrightarrow{vJ}\}$ of 130 dimensions.

For the SVM classification we adopted the LIBSVM [8] package to generate the hash vectors and build the final classifier model. This uses the RBF as the kernel function. Based on experimental analysis, C was set to the range $\{2^5, \ldots, 2^{15}\}$ and γ to $\{2^{-5}, \ldots, 2^3\}$. All combinations C and γ were tried using grid search to select the best CV accuracy based on the input data.

Image type	n	p	m	J size	$\overrightarrow{H}$ size
Fingerprint/Iris/Face	128	50	30	30×100	130

Table 1: Parameter values for experiments on Algorithm 1.

3.2 Experimental Results

We now discuss the results of the experimental evaluation of our approach. First, regarding the time performance, on the average, the hash vector from any given image is generated in 0.9597 seconds. The generation of SVM model for about 220 persons' hash vectors takes 3 or 4 hours. At the testing stage, once the model is

#	Biometric Type	Database Name	Description	# Images	# Persons	CV Accuracy %	FRR %	FAR %
1.	Finger-print	FVC2004, DB3_B	300×480, Thermal Sweeping Sensor	54	9	92.59	7.41	9.26×10^{-03}
2.	Finger-print	FVC2004, DB3_A	300×480, Thermal Sweeping Sensor	150	30	97.33	2.67	9.21×10^{-04}
3.	Finger-print	FVC2004, DB2	328×364, Optical Sensor	120	20	85.83	14.17	7.46×10^{-03}
4.	Finger-print	SFingGe v3.0, Synthetic Generator	288×384	50	10	88	12	1.33×10^{-02}
5.	Iris	UBIRIS.v1 Sessao_1	$800 \times 600 - 24$ bit color	1100	220	87.73	12.27	5.6×10^{-04}
6.	Iris	UBIRIS.v1 Sessao_2	$800 \times 600 - 24$ bit color	595	119	97.65	2.35	1.99×10^{-04}
7.	Face	The Yale Face Database B	$640 \times 480 - 8$ bit gray scale	100	10	99	1	1.11×10^{-03}
8.	Face	AT & T Databases of Faces	$92 \times 112 - 256$ bit gray scale	400	40	98.25	1.75	4.49×10^{-04}

Table 2: Summary of the experimental results of all biometric data types.

generated, it takes approximately 0.001 second to classify the test images.

Regarding the experimental results, the obtained results largely confirm the correctness of our algorithm: in each of the test cases, the accuracy was above 85% cross validation. False acceptance rates were within the interval $[1.99 \times 10^{-04}, 1.33 \times 10^{-02}]$, which translates into the assurance that the chances of accepting an incorrect biometric image are low. The worst observed FAR value is 1.33×10^{-2}, which interestingly is obtained for the images generated by the synthetic fingerprint generator, where the conditions for biometric generation were generally better controlled (e.g., there was no unexpected noise because of human interaction). Regarding FRR, the worst observed FRR value was in conjunction with the worst accuracy results since the FRR result is dependent on the accuracy (see previous section). The worst rate amounts to 14% (test case n. 3) and it is still acceptable, as it is in the same order of similar biometric key generators [24]. Additional insights specific to the different types of tested biometrics are discussed in what follows.

Fingerprint. Two types of Fingerprint Verification Competition (FVC) databases [34] corresponding to two types of sensors were used for the fingerprint biometric experiments. The sensors highly influence the quality of fingerprint images. We define the *quality* of the fingerprint image according to three criteria [28]: (i) high contrast between ridges and valleys, (ii) the image area foreground, and (iii) little scar or latency. As shown by the results, the CV cross validation is above 85% for each data set considered, which confirms the validity of our approach. A first important consideration suggested by the experimental results is that the algorithm performs better in case of large data set (as in the test case n. 2 in Table 2), most likely because of the more accurate training and testing during the configuration phase which helped in finding the optimal configuration parameters. We also notice that on average our algorithm performs better when using the thermal sensor than when using the optical sensor because the thermal sensor captures better quality fingerprint images. We can explain this result by elaborating more on how the quality is affected, in that the quality of the fingerprint image is affected by several human factors such as skin humidity and pressure. If the skin humidity is lower, the image quality of the optical sensor degrades. The skin humidity does not affect the image quality of the thermal sensor because it is the sweeping type. Moreover, regarding pressure, for optical sensor the foreground image is smaller for low pressure, while the fingerprint is smeared for high pressure. This is again not true for thermal sweeping sensor where the image quality is not significantly affected.

Note that the last data set was composed of artificially generated images. We experimented with synthetic fingerprint images as they potentially supply non-biased images and can be created at a low cost. It was difficult to control the randomness which lowered the cross validation classification accuracy to 88%. We believe the results could be improved using synthetic generator version which generates several samples corresponding to a single individual, maintaining the invariant features of an individual for all samples.

Iris. We used the UBRIS.v1 Sessao_1 (Session 1) and UBRIS.v1 Sessao_2 (Session 2) [44, 43] iris databases. For the first image capture session, noise factors, such as reflections, luminosity and contrast, were minimized. In the second session the capture place was changed to introduce a natural luminosity factor. Images collected in the second session simulated the ones captured by a vision system without or with minimal active participation from the subjects, adding possible noise to the resultant images. Note that when capturing iris images, some pre-processing is performed. A sequence of images is obtained rather than a single image. Not all images in the input sequence are clear and sharp enough for recognition. The images may be out of focus, or contain interlacing lines caused by eye motion or have severe occlusions caused by eyelids and eyelashes. Therefore, only high quality images from an input sequence are included in the final database.

Face. We used two databases for these experiments. The first one collected good quality images, in that photos were taken with subjects in frontal pose. Thus the resulting cross validation accuracy was 99%. The second set of tests was performed on images taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position with

tolerance for some side movement. Despite this, the overall cross validation accuracy of this database was 98.25% although the false rejection rate increased by .75%.

4. ANALYSIS

We start with proving some key properties related to uniqueness and repeatability and security properties of the BID generation algorithms. Based on such results we analyze privacy aspects and discuss how to prevent from possible attacks.

4.1 Uniqueness and Repeatability

A criterion frequently used for assessing uniqueness and repeatability in classification is the J_2 function [32]. The key idea of the J_2 function is to compare the *within-class* distance of the various hash vectors (or elements being classified) belonging to a given class, with the *between-class* distance among the various classes. There are two key steps to be taken while evaluating J_2.

The first step is to evaluate the *within-class* scatter matrix S_w: $S_w = \Sigma_{i=1}^{M} S_i P_i$ where M is the total number of classes; $S_i = E[(x-\mu_i)(x-\mu_i)^T]$ is the covariance matrix[4] for a class denoted by w_i where E is the expected value function, x is any vector in class w_i and μ_i is the mean vector of class w_i; and, $P_i = n_i/N$ where n_i is the number of samples in class w_i and N is the total number of samples in all the classes.

The second step is to evaluate the *between-class* scatter matrix S_b:$S_b = \Sigma_{i=1}^{M} P_i(\mu_i - \mu_o)(\mu_i - \mu_o)^T$ where $\mu_o = \Sigma_{i=1}^{M} P_i \mu_i$ is the global mean vector of all the classes.

From the above a covariance matrix of feature vectors with respect to the global mean is evaluated as $S_m = S_w + S_b$. Finally the J_2 criterion is calculated as: $J_2 = \frac{|S_m|}{|S_w|}$ As it is evident from the equation, for good repeatability of correct classification (small within-class distance), and uniqueness (large between-class distance) the value of J_2 should be large.

We carried out additional experiments on all the datasets to estimate J_2 and obtained average values of J_2 for fingerprint as 1.2712×10^{81}, iris as 1.5242×10^{303} and face as 3.7389^{103}. These values of J_2 and the corresponding classification accuracy (See Table 2) provide empirical evidence that the algorithm satisfies the uniqueness requirement on the biometric hashes generated based on the biometric datasets provided.

For clarity, we provide an example of a J_2 histogram for the Iris Session 1 database in Figure 5 (data corresponding to test case n. 5 in Table 2). Note that the J_2 metric requires the calculation of *within class* and *between class* distances of all the possible pairs of data elements. The y axis in the histogram presents the values of log(J2) class distances between any two classes. For instance for a value (120(x-axis),100(y-axis)) means that there are 100 class distances which have the J_2 value of 120. If there are all together $|C|$ number of total classes then the possible permutations of the distances to be tested are $\frac{|C| \times |C-1|}{2}$.

4.2 Biometric Image Keyed Hashing

We analyze the one-way security property of the SVD based biometric image hashing algorithm. More specifically, we show that it is computationally hard, given the BID hash vector $\overrightarrow{H}$ to reconstruct the original biometric image. We prove this result by the following two theorems. First, we prove that it is hard to construct the secondary image from the vector, which is required for reconstructing the original biometrics. The result (Theorem 2) shows that even if the second image is constructed or attacked, it is still hard to obtain the original biometric image I. Our results are based on the combination of mathematical properties of the SVD and the employed hashing technique.

[4]Covariance is the measure of how much two random variables vary together. A covariance matrix is a matrix of covariances between elements of a vector.

Type	n	spurious	η	# bits
Fingerprint	69	-	2.84×10^{19}	64
Fingerprint	139	69+1	2.36×10^{40}	134
Iris	220	-	4.52×10^{64}	214
Iris	119	-	2.43×10^{34}	114
Face	101	50+1	1.01×10^{29}	96

Table 3: Summary of number of SVM classes and entropy.

THEOREM 1. *Let $\overrightarrow{u}_J$ and $\overrightarrow{v}_J$ be the vectors which form the final hash value $H(u_J, v_J)$, and let λ_i be non-zero eigen values of the matrix $J^T J$ where J is the secondary image. If there is no λ_i that is dominant, then it is computationally hard to construct the secondary image from $H(u_J, v_J)$.*

Because in our theoretical results the assumption that there is no dominant eigenvalue is crucial, we have carried out extensive an experimental analysis on the biometric images to assess whether such assumption holds. Our experimental results show that such assumption holds because of the smoothness of the secondary image. A proof sketch of the theorem is reported in Appendix A.

THEOREM 2. *Given the secondary image it is computationally hard to obtain the original image I.*

Proof Sketch in Appendix A.

As a final remark we note that even if the attacker is able to retrieve the biometric image, it cannot reconstruct the hash vector without the knowledge of the secret random value needed during the selection of the p sub-images and to pseudorandomly combine them to form the secondary image J.

4.3 SVM Classes and BID Space

From the empirical analysis during the classification experiments provided in Section 3, we observe that if n is the number of classes, and these classes are listed in decreasing order of their confidence level, the highest confidence class is the same and the unordered set of the following t classes where $(n-1) \geq t \geq \frac{n}{2}$ is the same for the multiple testing rounds in the K-fold validation. In general, for most SVM classification experiments for all three biometrics, the ordering of several of the t classes was swapped with the neighboring classes. Therefore for the final label which denoted the final BID value, we use the class with the highest confidence followed by an unordered combination of the next t classes. For an attacker to guess the right key based on the classifier model, the number of choices would be $\eta = n + \binom{n}{t}$, under the assumption that each class has the same likelihood. Based on the uniqueness analysis from the J_2 metric we observe that the samples considered have

large inter-class distances, thus avoiding centroid formations that would narrow down the attacker's number of choices. As part of future work, we plan to further investigate inference-based attacks on the SVM model, which could potentially help the attacker make better guesses about the combination of classes used for generating the BID.

As noted from the experiments n in our case ranges in the interval $[69, 220]$. Based on the value of n, the resulting η ranges in the interval $[2^{64}, 2^{214}]$. η is proportional to the number of bits needed to encode the BID. More precisely the number of bits, considering the FAR for the primary class, is $MIN[\log_2(n), -\log_2(FAR)] + \log_2(\binom{n}{t})$. This results in the number of bits ranging in the interval $[64, 214]$. A summary of the experimental data corresponding to the biometric type, n, η and final number of bits of the BID is provided in Table 3.

4.4 Privacy and Security Analysis

We now analyze the relevant privacy and security properties of our technique, based on the above results. In addition we briefly analyze how our commitment technique is employed in the multi-factor approach to identity verification.

4.4.1 Privacy Analysis

Privacy in our context includes the following properties: unlinkability of the BID to the source biometric image, anonymity and confidentiality.

Unlinkability: Unlinkability refers to the impossibility of linking the BIB with a source biometric image. This property holds in our approach as a consequence of the irreversibility results of Theorems 1 and 2. The one-way nature of the BID generation process guarantees that there is no way to reconstruct the biometric image from the BID.

Confidentiality: Confidentiality refers to keeping the biometrics confidential throughout all the processing steps of the BID lifecycle. We protect confidentiality of the image as follows. First, once the biometric image is captured, the conversion phase only requires the hashing secrets and the SVM classifier model (referred to as the meta-data). Specifically, only the classifier model is permanently recorded by the system. During the verification phase, only the hash values obtained after processing the biometric images are used. Clear text images and templates are not required, so as to minimize information exposure. Therefore the only code that needs to be trusted to assure confidentiality of the biometric image is the code that given the initial image generates the hash value. Such code must be trusted not to leak the image and to discard the image once the hash value has been generated; the code is small and thus can be easily verified. We remark that confidentiality is preserved even in case an attacker gains partial information related to the BID. Since the BID and the biometric image are unlinkable, the confidentiality of the biometric image is preserved, as given the BID, given the unlinkability of the BID with the biometric image.

Anonymity: Anonymity refers to the property that prevents an individual to be identifiable within a set of subjects [42]. Our approach also assures anonymity, provided that no other identifying information is used in combination with the BID ZKPK proofs needed for verification. The generated BID, in fact, does not reveal any unique physiological information about the user's identity which is one of the key problems in typical matching based biometric verification. Also it follows from the unlinkability and confidentiality properties that the attacker cannot recreate the hash values given the biometric image and also cannot link a BID to an actual individual.

4.4.2 Security Analysis

Security in our system is given by the difficulty of perpetrating impersonation attacks.

We make two key assumptions in order to achieve a high-assurance BID generation. First, we assume that the sensor which captures the biometric image is able to detect *live* images and does not leak the image or information about the image. Second, we assume that the pseudorandom hashing secret used in Phase 1 is not compromised. If at least one of the two assumptions holds, then the BID cannot be compromised, as elaborated further in the analysis below.

We now focus on an attacker trying to impersonate a given user based on the BID and show how our approach withstands these types of attacks. We analyze the attackers' options by considering each of the secrets involved in the system.

The various possible points of attack include (A) biometric image; (B) hashing secrets; (C) classifier model used in Phase 2 (see Figure 2); (D) BID and possibly additional secrets and components depending on other cryptographic components used. The secrets of the system are the hashing secrets used in Phase 1 and the random commitment secret which is used together with the BID to create the cryptographic commitment. The classifier model is not assumed to be secret. Precisely, the classifier model can be revealed without jeopardizing the protocol security if the number of classes n is greater than 69. This is because $n > 69$ (69 is the minimum sample size used in our experiments) would make the number of possibilities greater than 2^{64} thus ensuring computational hardness. As described in Section 4.3, increasing the value of n by adding classes increases the keyspace; making it computationally hard for an attacker to perform a brute force attack.

	A	B	C	D	E	Attack Prevention Summary
1	×					BID cannot be created without hashing secrets.
2	×	×				BID cannot be created without classifier model.
3	×		×			The classifier model does not allow inference of the hashing secret needed construct BID.
4	×	×	×	×		The BID is compromised, but the commitment secret prevents from creating ZKPK.
5				×		The BID is compromised, but the commitment secret prevents from creating ZKPK. No other secrets are leaked.
6		×	×		×	All stored information is compromised but the BID cannot be created without biometric image.

Table 4: Possible security attacks [key: (A) biometric image (B) hashing secrets (C) classifier model (D) BID (E) commitment secret; ×: the value is known to the attacker].

To succeed in an impersonation attack the attacker needs to know all the secrets required to create the BK. In order to gather the other secrets, the attacker would have to pass the verification methods and compromise the system. Bypassing the cryptographic ZKPK protocol is computationally hard [18, 5]. Additionally, the cryptographic ZKPK protocol prevents replay attacks: the attacker cannot use the proofs created during a given biometric verification process

in any another verification process. Table 4 provides a summary of the various cases in which one or more secrets are compromised, and reports possible security implications. Case 1, 2 and 3 address the cases in which the biometric image is known to the attacker, but not the meta-data, which includes the hashing secret and classifier model, nor the random secret in the BID commitment, which are stored by the user. Thus, in these cases the attacker is not able to generate the BID. However, if the attacker knows the BID, then to perform successful verification it also needs the commitment secrets. This scenario is summarized by case 4. As noted earlier the knowledge of the BID does not reveal any information about the biometric image or the secrets involved as shown in case 5.

Finally, an interesting case is when the stored information including the meta-data and the commitment secret are compromised (case 6). In this case, the attacker's best choice as a source of information is the SVM model. However as we show in Section 4.3, for number of classes $n > 69$, the number of choices $> 2^{64}$ which makes it computationally hard for the attacker to guess the right BID.

5. RELATED WORK

Biometrics-based key generation has been extensively investigated in the past years. As mentioned earlier, the biometrics-based key generation is characterized by two stages. At the first stage certain biometric features are used to compute a bit string representing that biometrics. The bit string is then used in the second stage to generate a unique cryptographic key with the help of stored meta data. If two instances of the bit strings are sufficiently similar then the cryptographic key generated is the same. In most approaches, the second stage is independent of the biometrics being used, whereas the first is mostly biometric-specific.

The first approach to biometrics-based key generation is by Soutar *et al*. [50, 49, 48]. They developed methods for generating a repeatable cryptographic key from fingerprints using optical computing and image processing techniques. Following Soutar's work several strategies have been proposed for improving the second-stage of the key generation. Davida *et al*. [15] described a second-stage strategy using error correcting codes (ECC) and how it could be used with first-stage approaches for generating a bitstring representing iris scans[14]. The second-stage approach was significantly improved by Juels *et al*. [26, 27]. The underlying intuition behind the error correction and similar schemes can be understood based on Shamir's secret sharing scheme [47]. The hardness of Shamir's secret sharing scheme is based on the *polynomial reconstruction* problem which is a special case of the Reed-Solomon list decoding problem [4]. In fuzzy vault scheme proposed by Juels [27] based also on ECC, the user adds spurious *chaff* points which make it infeasible for an attacker to reconstruct the polynomial representing the BK.

Since the introduction of the fuzzy vault scheme, several researchers have implemented it in practice [11, 57, 17, 10, 19, 51, 40]. In particular the most recent work is by Nandakumar *et al*. [40] where the fuzzy vault implementation is based on the location of minutia points in a fingerprint. They generated 128 bit keys and obtained an accuracy rate of 91% for high quality images and 82.5% for medium quality images. The FRR was approximately 7% which shows an improvement over several other implementation of this scheme (where the average FRR was from 20-30%). From the experimental point of view, we generate 134 bit keys with the accuracy of 94.96% for high quality images and 86.92% for medium quality images. The FRR was on an average 9.06% which is comparable to the above scheme. From the algorithmic point of view, we use a similar concept of chaff points while adding spurious classes to make it hard for the attacker to guess the correct final key. We do not use ECC to retrieve the final key, but plan to investigate how ECC can be used while finding a list of SVM classes uniquely ordered by the confidence measures (See Section 4.3). A major difference of our approach with respect to the stage-one approaches of the various implementations of the fuzzy-vault is that their feature extraction is specific to the type of biometrics. Dependence on specific features has led to brute force attacks on several fuzzy vault implementations [35]. In our case, we instead use image analysis which can be used for several generic 2D biometric images such as fingerprint, iris and face.

Another scheme which makes use of the *polynomial reconstruction* problem in the second-stage is the scheme proposed by Monrose *et al*. which was originally used for hardening passwords using keystroke data [39] and then extended for use in cryptographic key generation from voice [38]. Let us consider the case when m biometric features are recorded at stage-one. When the system is initialized the main key κ and $2m$ shares of κ are generated using generalized secret sharing scheme. The shares are arranged within an $m \times 2$ table such that κ can be reconstructed from any set of m shares consisting of one share from each row. The selection is based on the biometric features recorded. Monrose *et al*. show that it is computationally infeasible for an attacker to guess the right shares because of the random or spurious shares present in the table. We also add spurious classes in the SVM classification model to make it infeasible for the attacker to guess the BID. Moreover, the features they capture in stage-one for key stroke [39] are durations and latencies, whereas for the voice [38] are the ceptral coefficients. Their experimental evaluation shows an average about 20-30% FRR. This biometric encoding of voices is not comparable with ours as we consider different biometrics which can be represented in 2D images.

Several of the techniques have been recently extended in the context of bio-hashing [33, 29, 12]. The approaches closest to ours are the bio-hashing techniques by Goh and Ngo [21, 41] who propose techniques to compute cryptographic keys from face bitmaps. *Biohashing* is defined as a transformation from representations which have a high number of dimensions and high uncertainty (example face bitmaps) to representations which have a low number of dimensions and zero uncertainty (the derived keys). Like our work, the goal of using the image hashing techniques is to extract bits from face images so that all similarly looking images will produce almost the same bit sequence. However, the work mainly focuses on the first stage of biometrics-based key generation and proposes the potential use of Shamirs secret sharing techniques [47] in the second stage. With respect to the first stage, Goh and Ngo use principal component (PCA) analysis for analyzing the images. This is similar to our use of SVD, as both SVD and PCA are common techniques for analysis of multivariate data. There is a direct relation between PCA and SVD in the case in which principal components are calculated from the covariance matrix. An important capability distinguishing SVD and related methods from PCA methods is the ability of SVD to detect weak signals or patterns in the data which is important in our case as we propose to use our techniques for generic 2D biometric images. The methodologies we employ for stage-one also differs in that the biometric hash vector output from stage-one cannot be simply distinguished using straight forward implementation of hamming distance based analysis as proposed in [21, 41]. We instead combine stage-one and stage-two

with the use of SVM classifiers in stage-two which provides a way to analyze the properties such as inter and intra-class distance of the biometric hash vectors. We provide a detailed analysis of our approach which has not been developed in earlier bio-hashing work.

There are other biometric cryptosystems in which biometric authentication is completely decoupled from the key release mechanism. The biometric template is stored on the device and when the biometric match happens, the cryptographic key is released [52]. This approach however has several vulnerabilities and is not related to our key generation approach.

6. CONCLUSION

In this paper we have presented a novel approach for generating BIDs from 2D biometric images. These BIDs can be used together with other identity attributes in the context of multi-factor identity verification techniques. In the proposed approach the secure management of the BID's random secret is an important issue. To address such issue there are approaches that provide a secure and usable way to manage and store those random secrets. One such approach [56] uses cellular phones based on NFC (Near Field Communication) technology and allows users to store secrets on the phone as well as to split them among various phone components (including an external card) and also on an additional external device for increased security. From the user side, configuration is very easy in that the user has a menu with three security levels (low, medium, high) among which to choose. Each such level corresponds to a different splitting strategy. We refer the reader to [56] for more details.

In addition to the technical solution provided in the paper, we have also investigated organizational requirements based on the potential scenarios where our approach would be most likely used[5]. In particular, the security of the initial enrollment is crucial for the overall process. We have developed cases in which enrollment has high assurance and it is performed at controlled and secure enrollment points. By contrast, in a non-secure enrollment, additional verification steps are needed to attest the biometric key generation software and the storage medium used for storing the user secret keys. We have thus explored the possible media used to store the secrets and benchmarked them to identify the most suitable media. Similar considerations apply to the verification locations, which may be protected or unprotected. Such analysis has been instrumental for clarifying the relevant preconditions that need to be met to successfully apply our approach, and to identify possible non-technical limitations.

We plan to further investigate possible attacks on the classification model to see if guessing attacks can reduce the entropy of the biometric samples considered. The η provided in Section 4 assumes that there are no guessing attacks as the J_2 value is high. However, there may be additional attacks such as those discovered by Mihailescu in [35] relevant to Fuzzy Valut schemes where the entropy of the scheme was significantly reduced as a result of the attacks.

7. REFERENCES

[1] AT & T Databases of Faces. `http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html`.

[2] K-Fold Cross Validation. `http://en.wikipedia.org/wiki/Cross-validation`.

[3] A. Bhargav-Spantzel, A. C. Squicciarini, R. Xue, and E. Bertino. Practical identity theft prevention using aggregated proof of knowledge. Technical report, CS Department, 2006. CERIAS TR 2006-26.

[4] D. Bleichenbacher and P. Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. *Lecture Notes in Computer Science*, 1807:53–77, 2000.

[5] J. Camenisch and A. Lysyanskaya. Efficient non-transferable anonymous multi-show credential system with optional anonymity revocation. In B. Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 93–118. Springer Verlag, 2001.

[6] J. Camenisch and A. Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Advances in Cryptology – CRYPTO '04*, 2004.

[7] R. Cappelli. SFinGe: an approach to synthetic fingerprint generation. In *International Workshop on Biometric Technologies (BT2004)*, pages 147–154, Calgary, Canada, June 2004.

[8] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[9] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, pages 89–105, London, UK, 1993. Springer-Verlag.

[10] Y. Chung, D. Moon, S. Lee, S. Jung, T. Kim, and D. Ahn. Automatic alignment of fingerprint features for fuzzy fingerprint vault. In *In Proceedings of Conference on Information Security and Cryptology*, pages 358–369, Beijing, China, Dec. 2005.

[11] T. C. Clancy, N. Kiyavash, and D. J. Lin. Secure smartcard based fingerprint authentication. In *WBMA '03: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications*, pages 45–52, New York, NY, USA, 2003. ACM Press.

[12] T. Connie, A. Teoh, M. Goh, and D. Ngo. Palmhashing: A novel approach for cancelable biometrics. *Information Processing Letters*, 93(1):1–5, 2005.

[13] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[14] J. Daugman. Biometric personal identification system based on iris analysis. In *United States Patent*, 1994.

[15] G. Davida, Y. Frankel, and B. Matt. The relation of error correction and cryptography to an offine biometric based identication scheme. In *Proceedings of WCC99, Workshop on Coding and Cryptography, 1999.*, 1999.

[16] R. Dhamija and J. D. Tygar. The battle against phishing: Dynamic security skins. In *SOUPS '05: Proceedings of the 2005 Symposium on Usable Privacy and Security*, pages 77–88, New York, NY, USA, 2005. ACM Press.

[5]Details concerning the organizational requirements for our biometric verification protocols are reported in a technical report, which we are unable to refer because of the double blind review requirements.

[17] Y. C. Feng and P. C. Yuen. Protecting face biometric data on smartcard with reed-solomon code. In *Proceedings of CVPR Workshop on Privacy Research In Vision*, page 29, New York, USA, June 2006.

[18] U. Fiege, A. Fiat, and A. Shamir. Zero knowledge proofs of identity. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 210–217, New York, NY, USA, 1987. ACM Press.

[19] M. Freire-Santos, J. Fierrez-Aguilar, and J. Ortega-Garcia. Cryptographic key generation using handwritten signature. In P. J. Flynn and S. Pankanti, editors, *Proceedings of SPIE: Biometric Technology for Human Identification III*, volume 6202, 2006.

[20] A. Georghiades, P. Belhumeur, and D. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Pattern Analysis and Machine Intelligence*, 23(6):643–660, 2001.

[21] A. Goh and D. C. Ngo. Computation of cryptographic keys from face biometrics. In *Communications and Multimedia Security*, volume 2828 of *LNCS*, pages 1–13, 2003.

[22] K.-S. Goh, E. Chang, and K.-T. Cheng. Support vector machine pairwise classifiers with error reduction for image classification. In *MULTIMEDIA '01: Proceedings of the 2001 ACM workshops on Multimedia*, pages 32–37, New York, NY, USA, 2001. ACM Press.

[23] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.

[24] F. Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. *IEEE Transactions on Computers*, 55(9):1081–1088, 2006.

[25] R. Housley, W. Polk, W. Ford, and D. Solo. Internet X.509 public key infrastructure certificate and certificate revocation list (CRL) profile, 2002.

[26] A. Juels and M. Wattenberg. A fuzzy commitment scheme. In *ACM Conference on Computer and Communications Security*, pages 28–36, 1999.

[27] A. Juels and M. Wattenberg. A fuzzy vault scheme. In *Proceedings of IEEE International Symposium on Information Theory, 2002*., 2002.

[28] H. Kang, B. Lee, H. Kim, D. Shin, and J. Kim. A study on performance evaluation of fingerprint sensors. In *Audio and Video Based Biometric Person Authentication*, pages 574–583, 2003.

[29] A. Kong, K.-H. Cheung, D. Zhang, M. Kamel, and J. You. An analysis of biohashing and its variants. *Pattern Recognition*, 39(7):1359–1368, 2006.

[30] S. S. Kozat, R. Venkatesan, and M. K. Mihcak. Robust perceptual image hashing via matrix invariants. In *International Conference on Image Processing*, pages V: 3443–3446, 2004.

[31] C. Li, L. Khan, and B. Prabhakaran. Real-time classification of variable length multi-attribute motions. *Knowledge Information Systems*, 10(2):163–183, 2006.

[32] C.-C. Li and K. S. Fu. Machine-assisted pattern classification in medicine and biology. *Annual Review of Biophysics and Bioengineering*, 9:393–436, 1980.

[33] A. Lumini and L. Nanni. An improved biohashing for human authentication. *Pattern Recognition*, 40(3):1057–1065, 2007.

[34] D. Maio and D. Maltoni. FVC2004: third fingerprint verification competition. `http://bias.csr.unibo.it/fvc2004/`, 2004.

[35] P. Mihailescu. The fuzzy vault for fingerprints is vulnerable to brute force attack. Technical report, University of Göttingen, 2007.

[36] M. K. Mihçak and R. Venkatesan. New iterative geometric methods for robust perceptual image hashing. In *DRM '01: Revised Papers from the ACM CCS-8 Workshop on Security and Privacy in Digital Rights Management*, pages 13–21, London, UK, 2002. Springer-Verlag.

[37] X. min Tao, F. rong Liu, and T. xian Zhou. A novel approach to intrusion detection based on SVD and SVM. *Industrial Electronics Society*, 3(2–6):2028–2033, November 2004.

[38] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice. In *SP '01: Proceedings of the 2001 IEEE Symposium on Security and Privacy*, page 202, Washington, DC, USA, 2001. IEEE Computer Society.

[39] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. In *CCS '99: Proceedings of the 6th ACM conference on Computer and communications security*, pages 73–82, New York, NY, USA, 1999. ACM Press.

[40] K. Nandakumar, A. K. Jain, and S. Pankanti. Fingerprint-based fuzzy vault: Implementation and performance. In *IEEE Transactions on Information Forensics and Security, 2007 (To appear)*, 2007.

[41] D. C. Ngo, A. B. Teoh, and A. Goh. Biometric hash: high-confidence face recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(6):771–775, June 2006.

[42] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability, and pseudonymity - a proposal for terminology. pages 1–9. 2001.

[43] H. Proença and L. A. Alexandre. UBIRIS: a noisy iris image database. In *ICIAP 2005: International Conference on Image Analysis and Processing*, volume 1, pages 970–977, 2005.

[44] H. Proença and L. A. Alexandre. Toward non-cooperative iris recognition: A classification approach using multiple signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence, Special Issue on Biometrics*, 9(4):607–612, July 2007. ISBN 0162-8828.

[45] A. Ross, A. K. Jain, and J.-Z. Qian. Information fusion in biometrics. In *Pattern Recognition Letters*, volume 24, pages 2115–2125, September 2003.

[46] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *IEEE Workshop on Applications of Computer Vision*, Sarasota (Florida), December 1994.

[47] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[48] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar. Biometric encryption^{TM} - enrollment and verification procedures. In *SPIE 98: In Proceedings of Optical Pattern Recognition IX*, volume 3386, pages 24–35, 1998.

[49] C. Soutar, D. Roberge, A. Stoianov, R. Gilroy, and B. V. Kumar. Biometric encryption^{TM} using image processing. In *SPIE 98: In Proceedings of Optical Security and Counterfeit Deterrence Techniques II*, volume 3314, pages 178–188, 1998.

[50] C. Soutar and G. J. Tomko. Secure private key generation using a fingerprint. In *Proceedings of Cardtech/Securetech Conference*, volume 1, pages 245–252, May 1996.

[51] U. Uludag and A. Jain. Securing fingerprint template: Fuzzy vault with helper data. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 163, Washington, DC, USA, 2006. IEEE Computer Society.

[52] U. Uludag, S. Pankanti, S. Prabhakar, and A. Jain. Biometric cryptosystems: Issues and challenges. In *Proceedings of the IEEE, Special Issue on Enabling Security Technologies for Digital Rights Management,,2004*., volume 92, 2004.

[53] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[54] S. Wang and Y. Wang. Fingerprint enhancement in the singular point area. *IEEE Signal Processing Letters*, 11(1):16–19, January 2004.

[55] Y. Wang, Y. Sun, M. Liu, P. Lv, and T. Wu. Automatic inspection of small component on loaded PCB based on SVD and SVM. In *Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications IX*., volume 6315 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, September 2006.

[56] J. Woo, A. Bhargav-Spantzel, A. Squicciarini, and E. Bertino. Verification of receipts from m-commerce transactions on nfc cellular. In *10th IEEE Conference on E-Commerce Technology (CEC 08)*, July 2008.

[57] S. Yang and I. Verbauwhede. Automatic secure fingerprint verification system based on fuzzy vault scheme. In *ICASSP '05: Proceedings of the Acoustics, Speech, and Signal Processing*, volume 5, pages 609–612, Philadelphia, USA, March 2005.

[58] W. Zhang, Y.-J. Chang, and T. Chen. Optimal thresholding for key generation based on biometrics. In *ICIP '04: International Conference on Image Processing*, pages 3451–3454, 2004.

APPENDIX

Proof.[Theorem 1]

If only the final hash value is known to an adversary, then the first step is to approximate the secondary image J (See Figure 2). We prove the hardness by analyzing the following equation which provides a possible approximation of the secondary images –

$$J = \sum_{i=1}^{r} \sqrt{\lambda_i} u_i v_i^T = \sqrt{\lambda_1} u_J v_J^T + \underbrace{\sqrt{\lambda_2} u_2 v_2^T + \sqrt{\lambda_3} u_3 v_3^T + \ldots + \sqrt{\lambda_r} u_r v_r^T}$$

where $r = 2p$; p is the number of sub-images created; and λ_i, $1 \leq i \leq r$ are non-zero eigen values of the matrix $J^T J$ such that $\lambda_1 > \lambda_2 > \ldots > \lambda_r$. Note that J^T is the transpose matrix of J and a positive square root of λ_i is a singular value. The u_i's and v_i's, $i = [1, \ldots, r]$, are eigenvectors of JJ^T and $J^T J$ respectively. Since the final hash value, $[u_J, v_J]$ are known to the adversary, the values which need to be guessed are λ_1 and $\{\lambda_2 u_1 v_1^T + \lambda_3 u_2 v_2^T + \ldots + \lambda_r u_r v_r^T\}$. To guess λ_i's there are infinitely many solutions as any nonnegative eigenvalues can lead to specific eigenvectors that are unitary (i.e. satisfy the definition). Any eigenvalue matrix resulting from this construction will give a solution to the equation and therefore it is computationally hard for the adversary to identify the original value.

If there is a case in which λ_1 is dominant such that the rest of the values $\lambda_2, \ldots, \lambda_r$ are approximately equal to zero, then one could try to guess λ_1 and possibly approximate the secondary image by $\dot{J} = \sqrt{\lambda_1} u_J v_J^T$. It is not trivial to theoretically predict the possible distribution of the values of λ_i's because they are dependent on the type of image and the distribution of the pixel values of those images. Therefore we conducted experimental evaluation on the biometric images and found that the λ_i's are distributed such that there is no one dominant eigenvalue because the secondary image J is a smooth image (i.e. the adjacent pixels of the image do not differ beyond a certain threshold which is determined by the algorithm parameters). We conclude that because of the hardness of guessing the eigenvalues and the lack of dominant eigenvalues the reconstruction of the secondary image J from the resultant hash vector $\overrightarrow{H}$ is computationally hard for the biometric types considered. □

Proof Sketch. [Theorem 2]

If J is known to the adversary, then the first step would be to form each sub-image matrix A_i, where $1 \leq i \leq p$. Note that a combination of all A_i eigenvectors were used to construct J. Each A_i is of the form $A_i = U_i S_i V_i^T$. As in the proof of Theorem 1, an infinite number of eigenvalues exist for constructing infinite A_i which would satisfy the relation. Moreover, using the same reasoning as before, there are no dominant eigenvalues as the p sub-images each of size $m \times m$ are overlapping. Because of the overlap most significant eigenvalues do not differ beyond a certain threshold as determined by the algorithm parameters p and m. In addition the largest eigenvectors (i.e. the left most and the right most vectors of the U_i and V_i matrices respectively) of each sub-image A_i are pseudorandomly combined to form J resulting in the number of choices the attacker would need to try as $p!$. This motivates the need for large values of p (~ 50). As a result guessing the order of each sub-image A_i and hence creating the original image I is computationally hard.

Biometrics Based Identifiers for Digital Identity Management

Abhilasha Bhargav-Spantzel, Anna Squicciarini, Elisa Bertino,
Xiangwei Kong, Weike Zhang

IDTrust 2010
April 14th 2010

Outline

1. Identity Concepts Overview
2. Biometric Systems Overview
3. Biometric Commitments
4. Backup

Digital Identity

- **Digital identity**:
  - nyms.
  - identity attributes or identifiers:
    - strong identifiers (eg. SSN)
    - weak identifiers (eg. age)
- **Owner of an identity attribute**:
  Individual who is
  - issued the identity attribute
  - authoritative of making the claim
- **Identity verification**:
  Claimed attribute is
  - owned by the individual
  - valid

Digital Identity

- **Digital identity**:
  - nyms.
  - identity attributes or identifiers:
    - strong identifiers (eg. SSN)
    - weak identifiers (eg. age)
- **Owner of an identity attribute**:
  Individual who is
  - issued the identity attribute
  - authoritative of making the claim
- **Identity verification**:
  Claimed attribute is
  - owned by the individual
  - valid

Digital Identity

- **Digital identity**:
  - nyms.
  - identity attributes or identifiers:
    - strong identifiers (eg. SSN)
    - weak identifiers (eg. age)
- **Owner of an identity attribute**:
  Individual who is
  - issued the identity attribute
  - authoritative of making the claim
- **Identity verification**:
  Claimed attribute is
  - owned by the individual
  - valid

Digital Identity (cont.)

- **Identity assurance**: Confidence about
  - ownership
  - validity

Identity assurance and linkability

Linkability / Assurance	Strong	Weak
Strong	**SSN** is valid and owned by ***Homer07*** who is **Bob Smith**	***Homer07*** is a **U.S. Citizen**
Weak		

Digital Identity (cont.)

- **Identity assurance**: Confidence about
  - ownership
  - validity

Identity assurance and linkability

Linkability / Assurance	Strong	Weak
Strong	**SSN** is valid and owned by ***Homer07*** who is **Bob Smith**	***Homer07*** is a **U.S. Citizen**
Weak		

Digital Identity (cont.)

- **Identity assurance**: Confidence about
  - ownership
  - validity

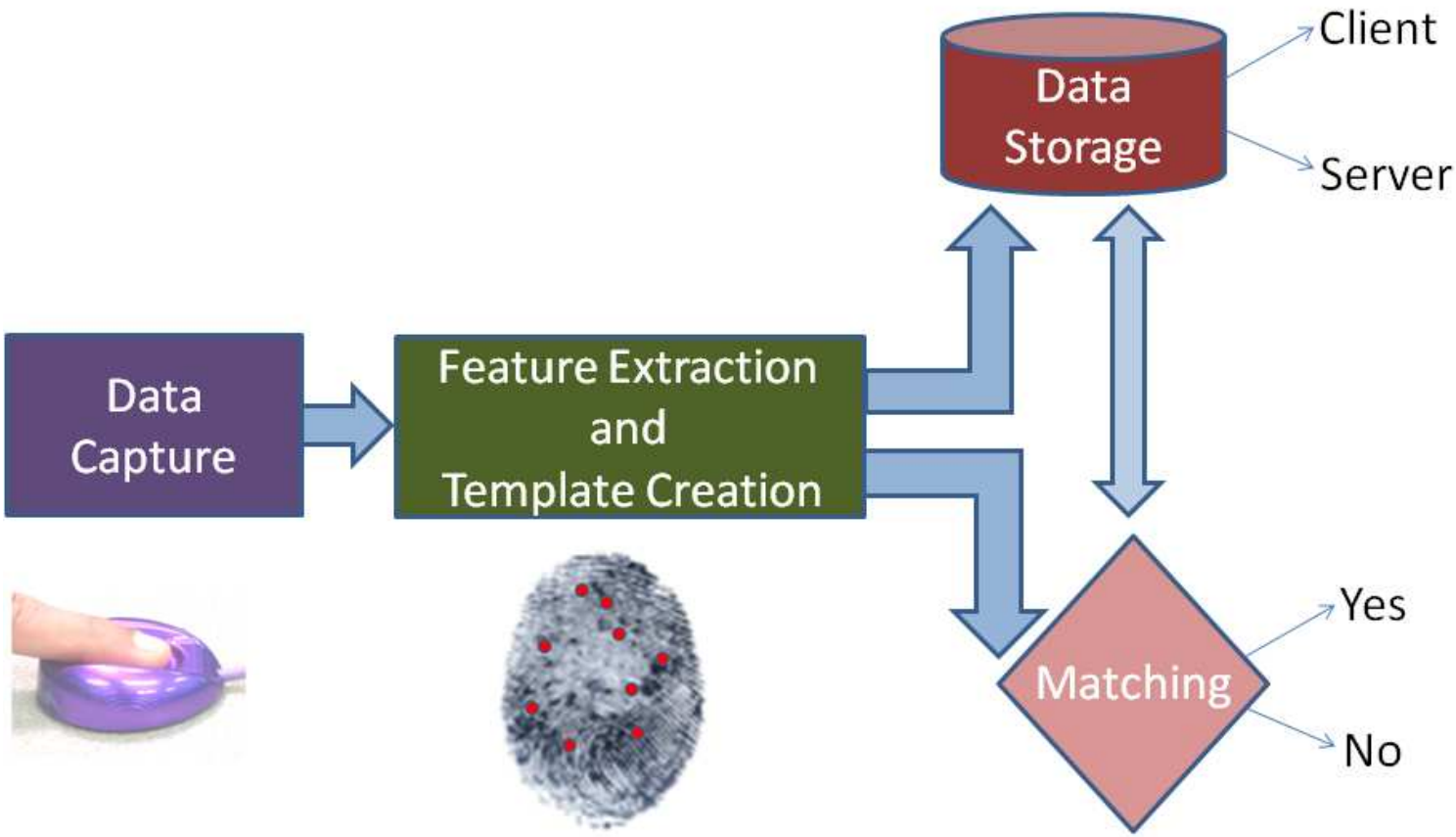
Identity assurance and linkability

Linkability / Assurance	Strong	Weak
Strong	**SSN** is valid and owned by ***Homer07*** who is **Bob Smith**	***Homer07*** is a **U.S. Citizen**
Weak		

Outline

1. Identity Concepts Overview
2. Biometric Systems Overview
3. Biometric Commitments
   - Our Approach
   - Main Techniques
   - Experiments and Results
   - Analysis
   - Related Work
4. Backup

Biometric Matching Based Systems



Biometric Keys: General Idea

- Generating cryptographic keys from biometric measurements:
  - **Phase 1:**
    - Biometric features $\longrightarrow$ bit string
    - Bit string should have *large inter-class variation* and *small intra-class variation*
  - **Phase 2:**
    - Bit string $\xrightarrow{metadata}$ unique key
    - If two instances of bit strings are 'similar' then the key generated is the same

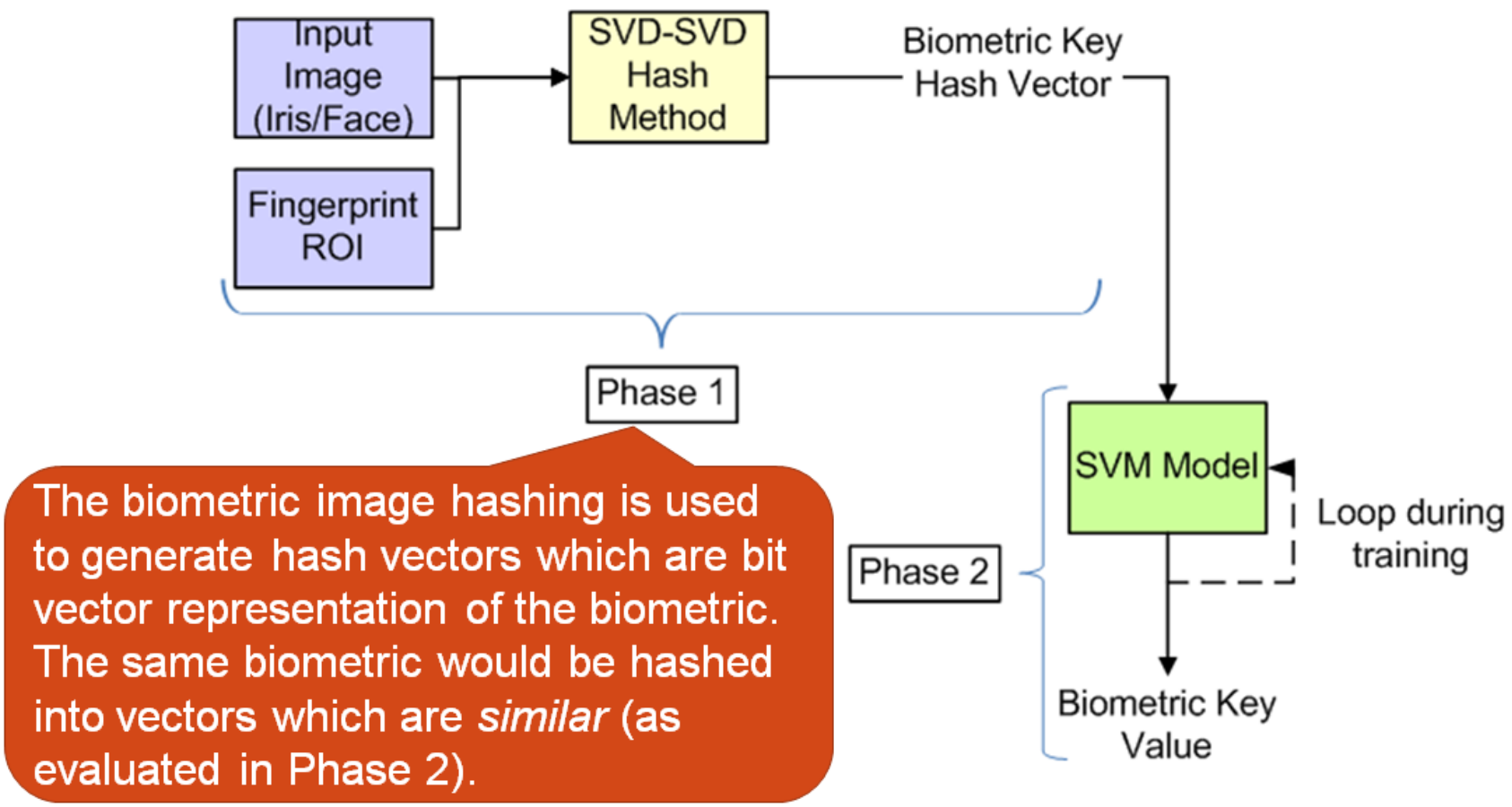
Biometric Keys: General Idea

- Generating cryptographic keys from biometric measurements:
  - **Phase 1:**
    - Biometric features $\longrightarrow$ bit string
    - Bit string should have *large* *inter-class variation* and *small* *intra-class variation*
  - **Phase 2:**
    - Bit string $\xrightarrow{metadata}$ unique key
    - If two instances of bit strings are 'similar' then the key generated is the same

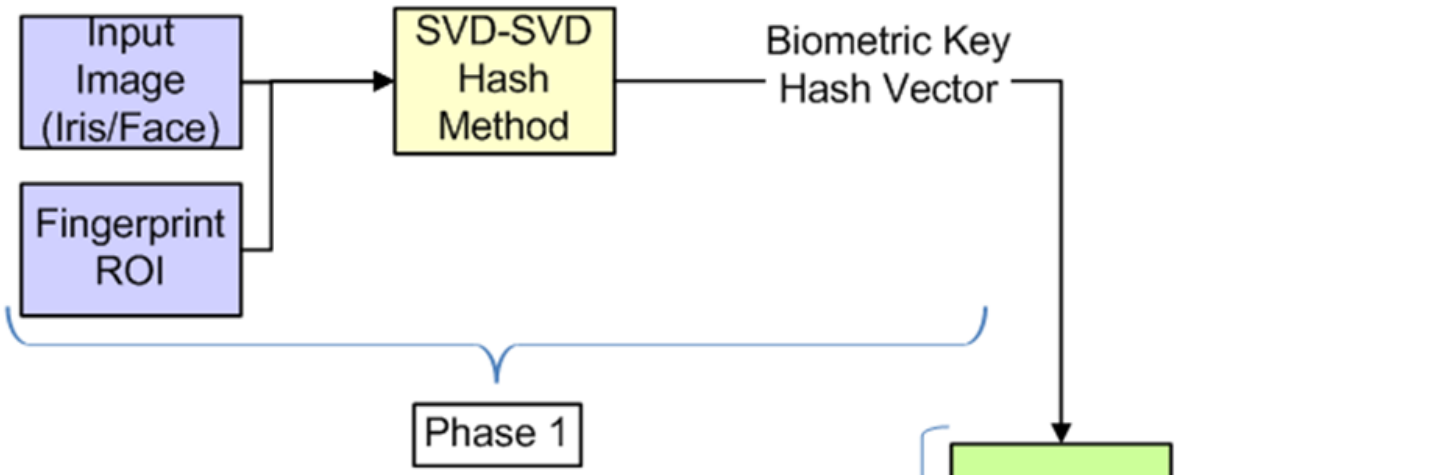
Biometric Keys: General Idea

- Generating cryptographic keys from biometric measurements:
  - **Phase 1:**
    - Biometric features $\longrightarrow$ bit string
    - Bit string should have *large inter-class variation* and *small intra-class variation*
  - **Phase 2:**
    - Bit string $\xrightarrow{metadata}$ unique key
    - If two instances of bit strings are 'similar' then the key generated is the same

Two main phases of the biometric key generation

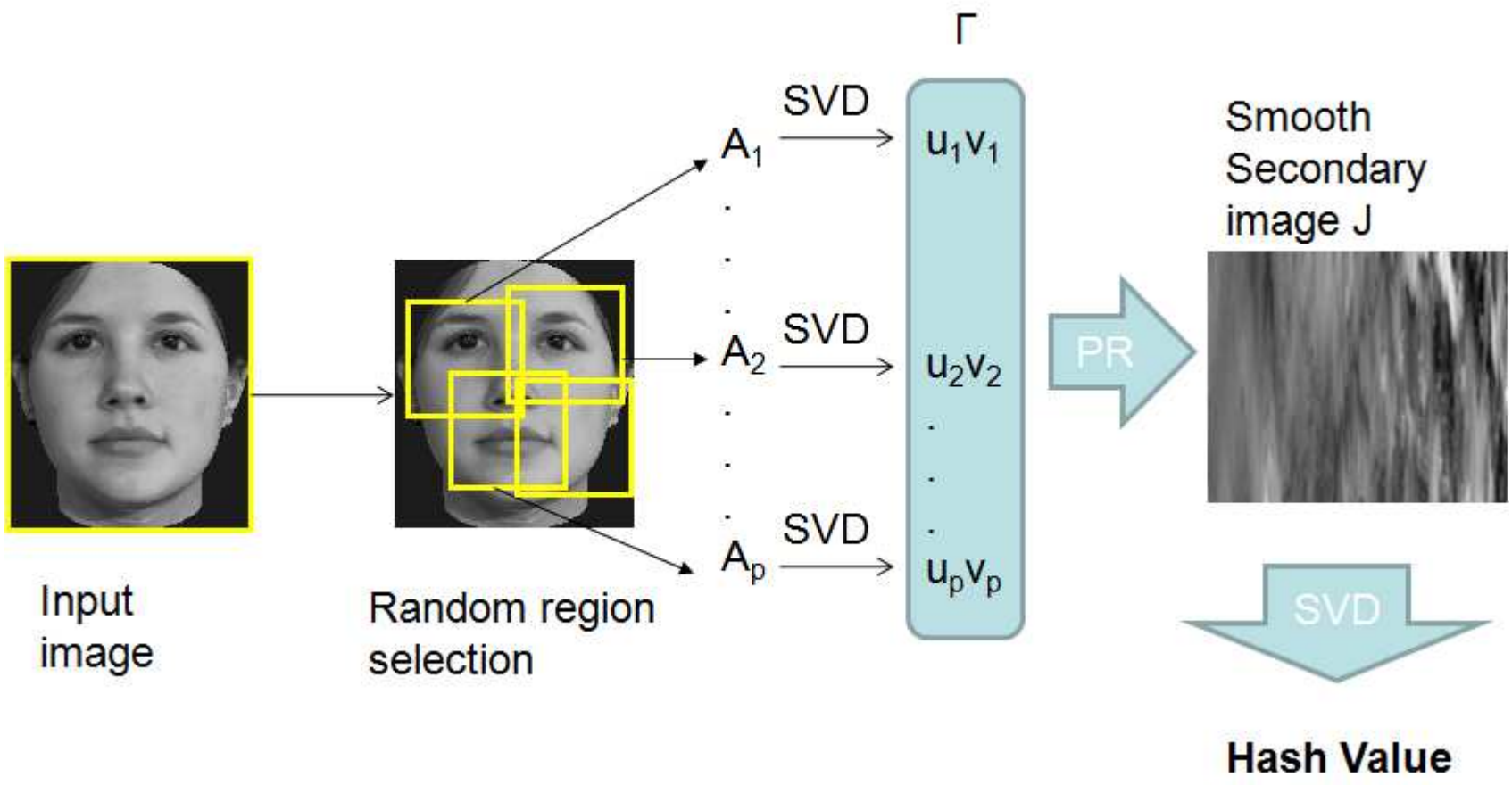


Two main phases of the biometric key generation



The Biometric Key (BK) is the class combination value, which is used with a random secret (r) to form a commitment of the form $g^{BK}h^{r}$ which is then used in zero-knowledge proof verification.

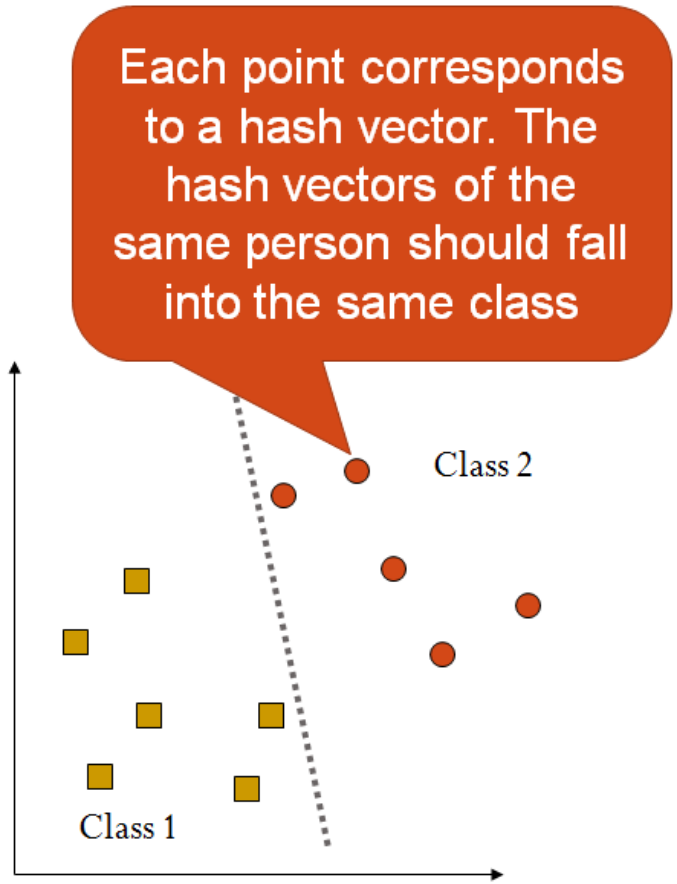
Biometric Hashing Process



Key Steps of Biometric Hashing Algorithm

1. Random selection of A_i from biometric image
2. **First SVD transform:** $A_i = U_i S_i V_i^T \ 1 \leq i \leq p$
3. Random selection of eigenvectors to create secondary image J
4. **Second SVD transform:** $J = U_J S_J V_J^T$
5. **Final hash vector:** $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$

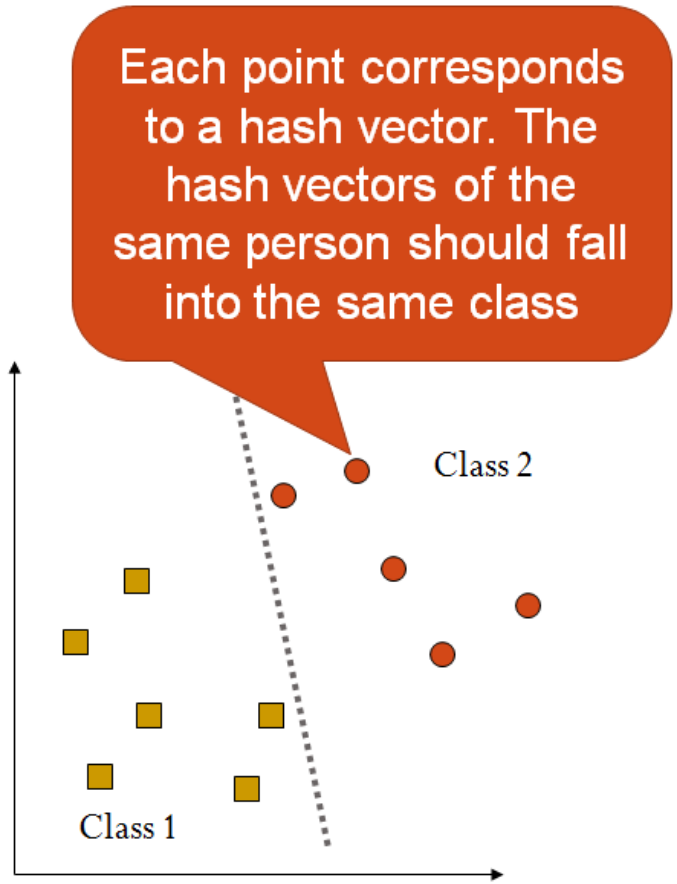
SVM Classification



SVM Usage

- The hash vectors are ranked based on confidence degrees from SVM
- Biometric key: highest confidence class and top n/2 classes (total n classes)
- Attacker choices for brute force $n + \binom{n}{t}$
- For $n > 69$ number of choices is $> 2^{64}$

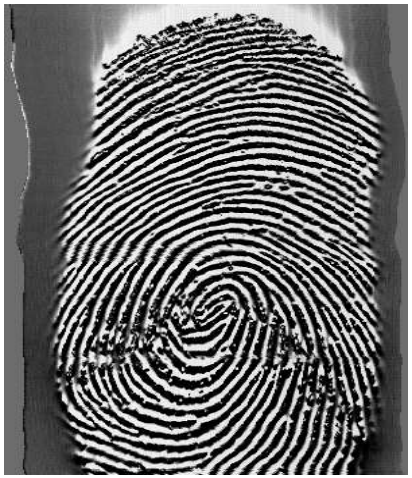
SVM Classification



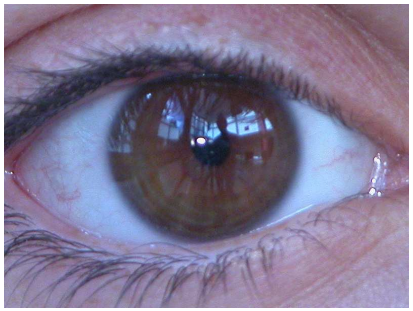
SVM Usage

- The hash vectors are ranked based on confidence degrees from SVM
- Biometric key: highest confidence class and top n/2 classes (total n classes)
- Attacker choices for brute force $n + \binom{n}{t}$
- For $n > 69$ number of choices is $> 2^{64}$

Experimental Samples



Thermal (left) and optical (right) sensor fingerprint samples [324 images - FVC]



Iris sample [1695 images - UBIRIS]

Experimental Samples (cont.)

Yale face samples [100 images]

AT&T face samples [400 images]

Summary of Experimental Results

Type	# Images	# Persons	CV Accuracy %	FAR %
Fingerprint (Thermal)	204	39	94.96	5.09×10^{-03}
Fingerprint (Optical)	120	20	85.83	7.46×10^{-03}
Iris	1695	339	92.69	3.80×10^{-04}
Face Yale	100	10	99	1.11×10^{-03}
Face AT&T	400	40	98.25	4.49×10^{-04}

Uniqueness and Repeatability Analysis

- The metric to measure uniqueness and repeatability is

$$J_2 = \frac{|S_m|}{|S_w|}$$

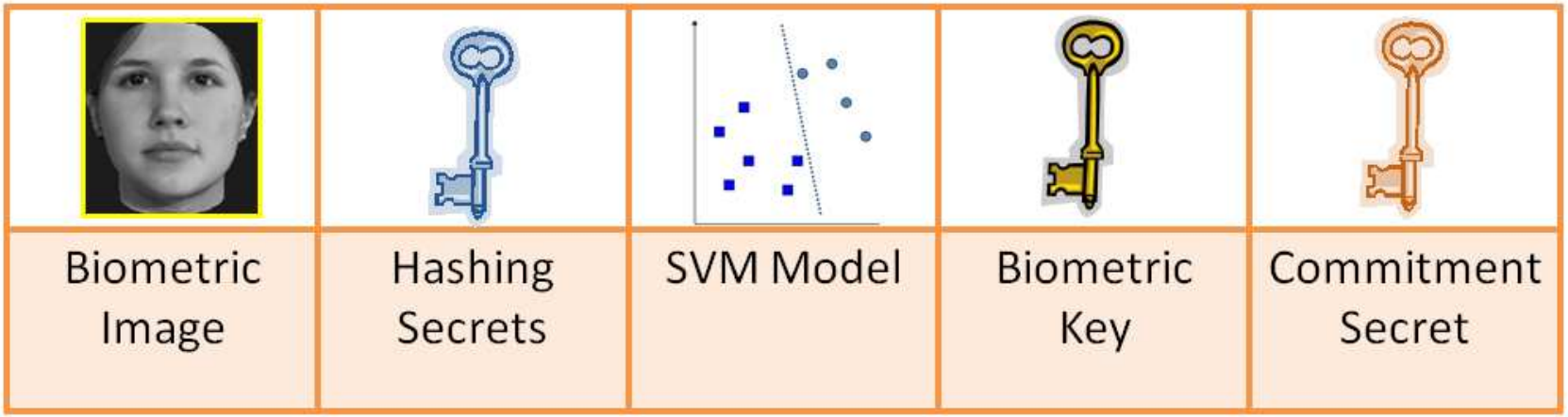
  where S_m is inter-class distance and S_w is intra-class distance

- The average values of J_2 calculated were as follows–
  - Fingerprint : 1.2712×10^{81}
  - Iris : 1.5242×10^{303}
  - Face : 3.7389^{103}

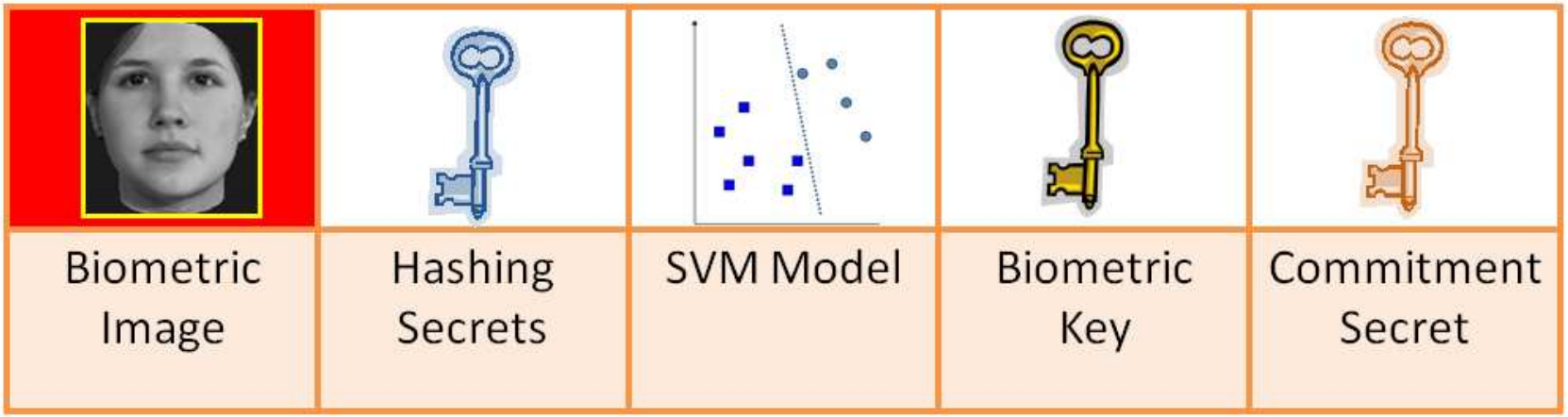
Biometric Key Analysis

Type	n	Spurious classes	η	# of BK bits
Fingerprint	69	-	2.84×10^{19}	64
Fingerprint	139	69+1	2.36×10^{40}	134
Iris	220	-	4.52×10^{64}	214
Iris	119	-	2.43×10^{34}	114
Face	101	50+1	1.01×10^{29}	96

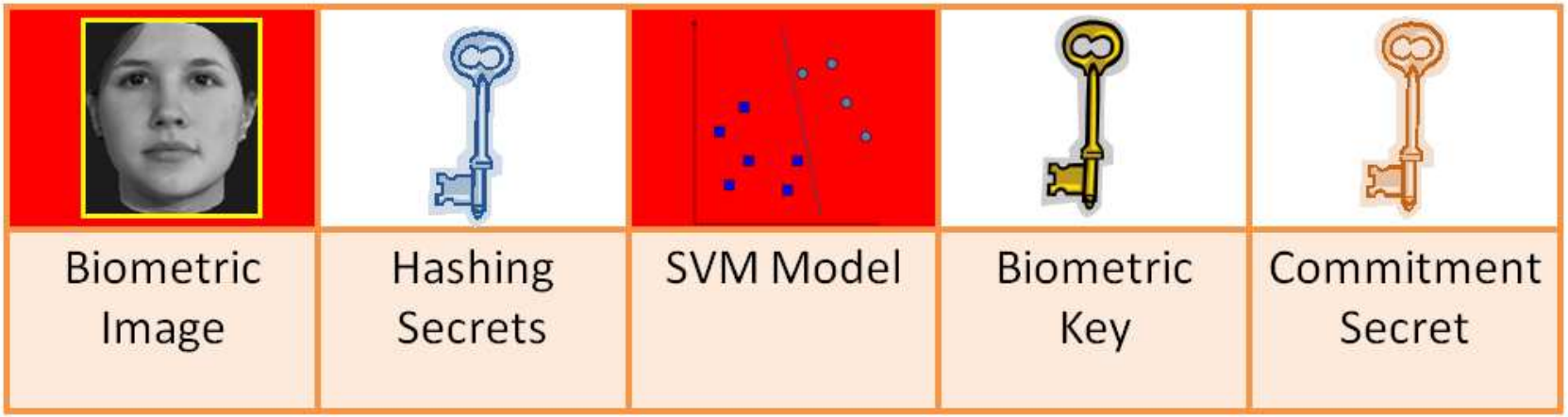
Biometric Verification System Analysis



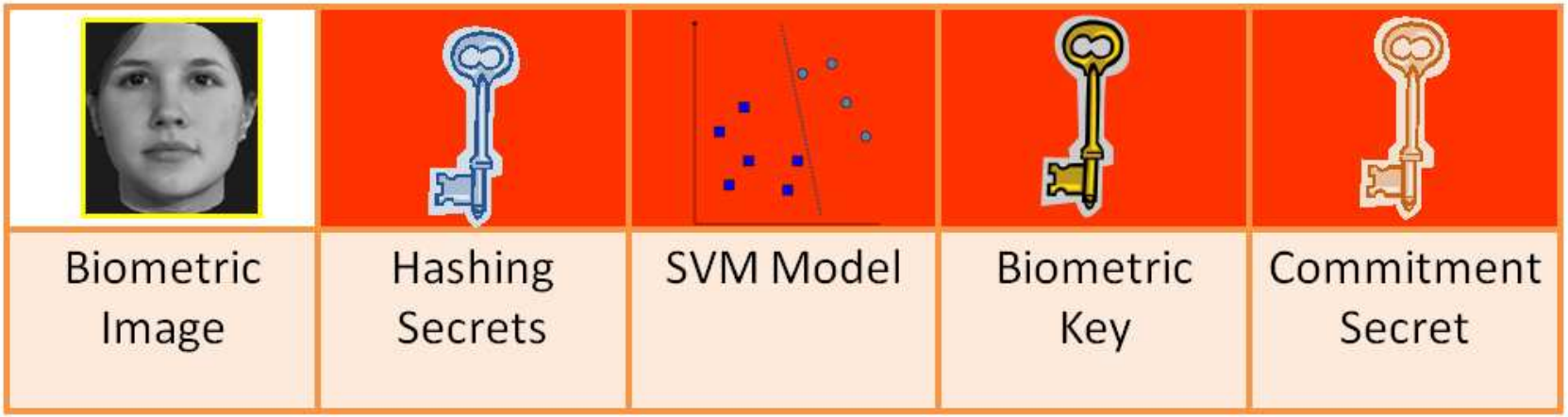
Biometric Verification System Analysis



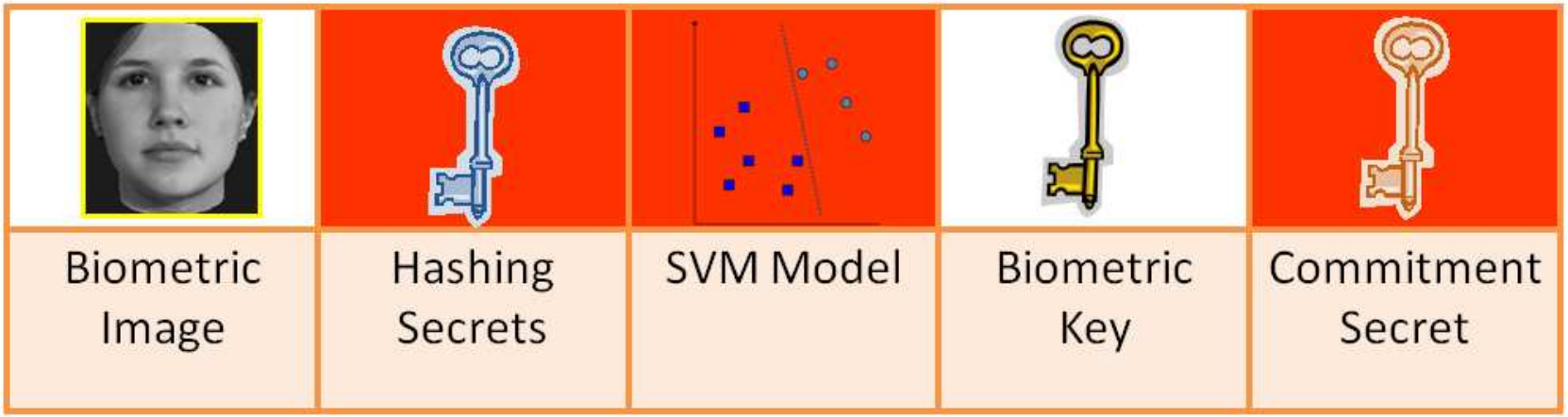
Biometric Verification System Analysis



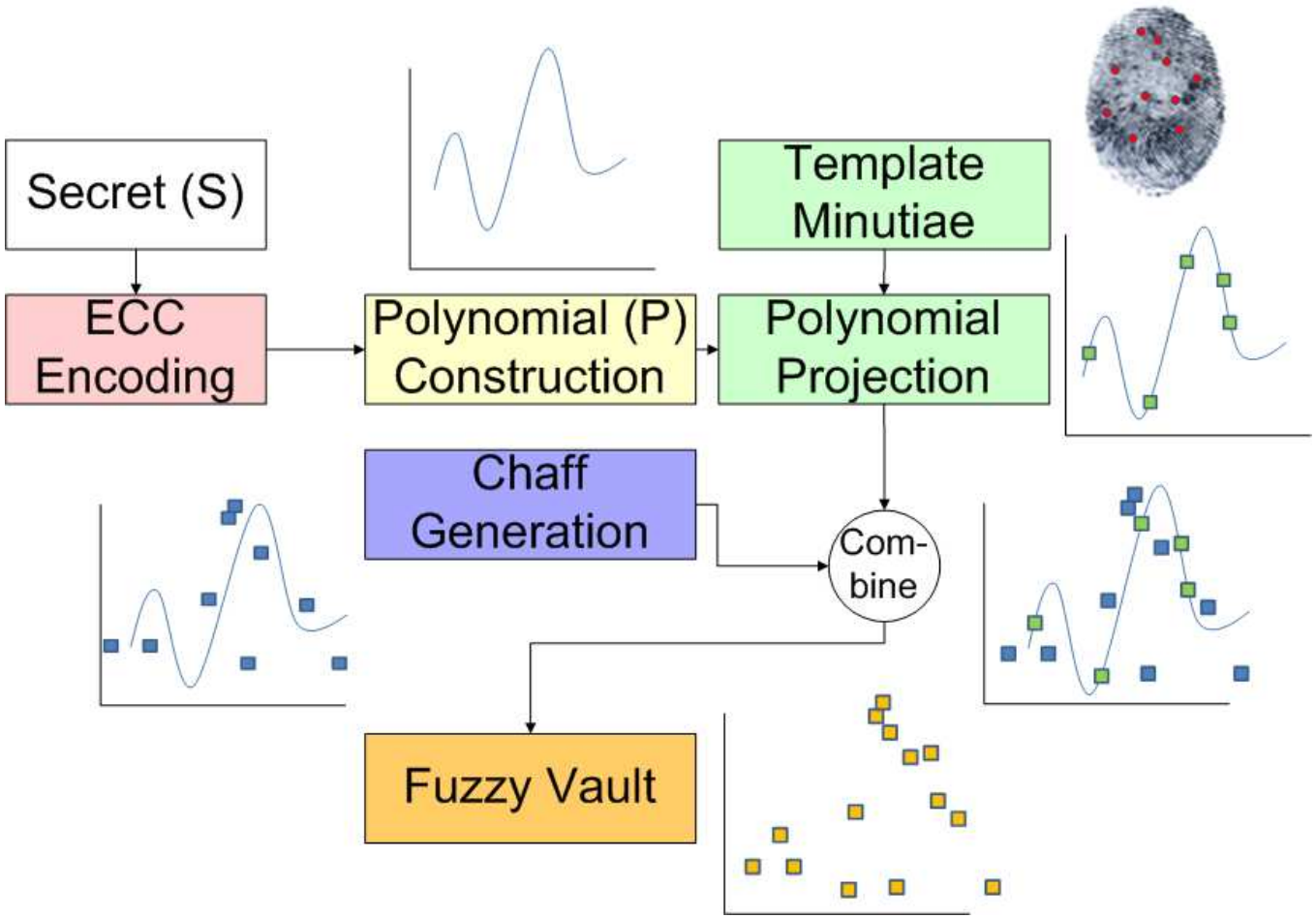
Biometric Verification System Analysis



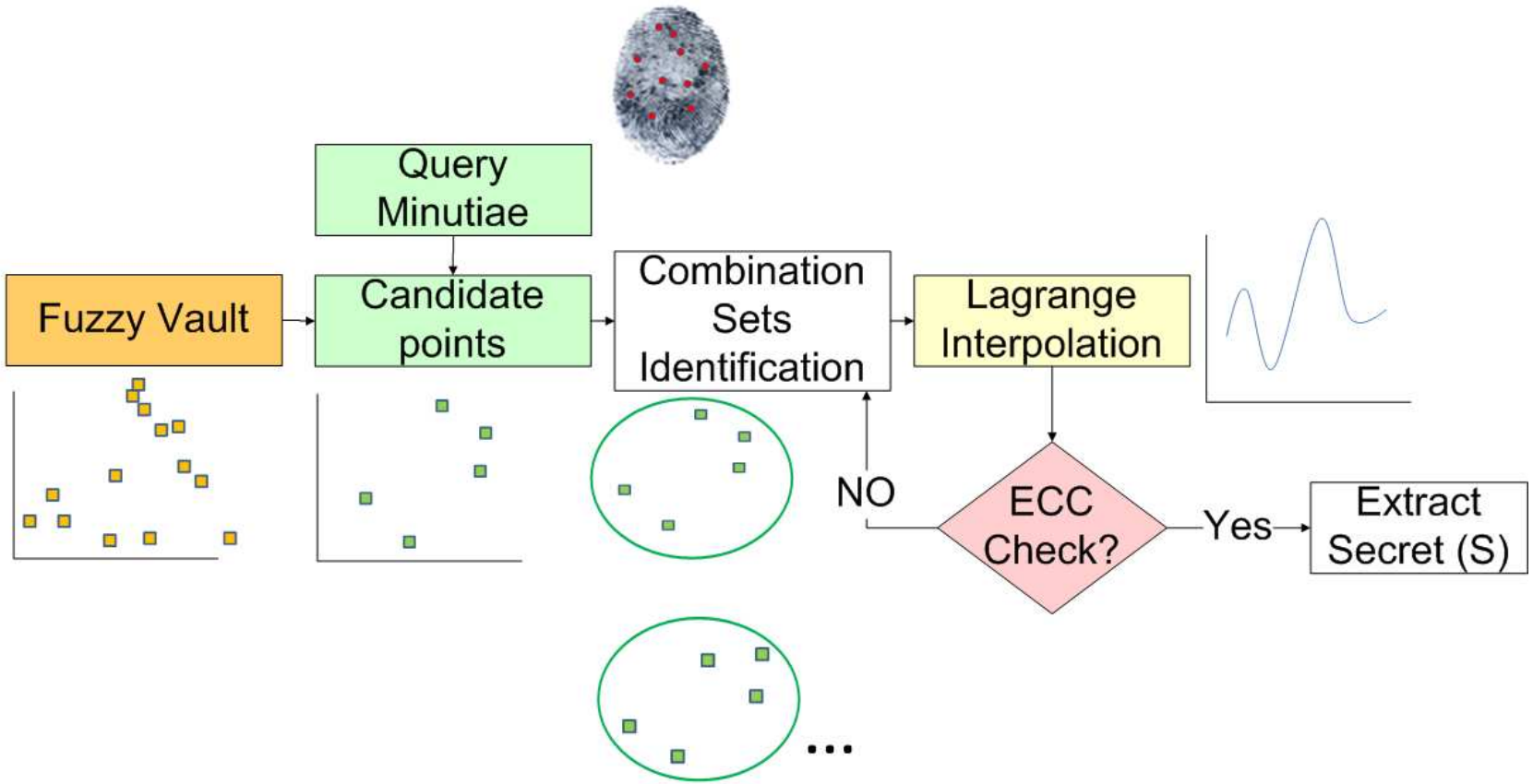
Biometric Verification System Analysis



Fuzzy Vault Scheme



Fuzzy Vault Scheme (cont.)



Fuzzy Vault Scheme - Shortcomings

- **Intra-class variability**: rotation, translation, # minutia points
  - *'helper data'* reduces security
- Increasing the degree of the **polynomial** increases complexity
  - require increased number of minutiae points
- Increasing the number of **chaff points** increases the complexity
  - empirical bound because of minutiae location

Fuzzy Vault Scheme - Shortcomings

- **Intra-class variability**: rotation, translation, # minutia points
  - *'helper data'* reduces security
- Increasing the degree of the **polynomial** increases complexity
  - require increased number of minutiae points
- Increasing the number of **chaff points** increases the complexity
  - empirical bound because of minutiae location

Fuzzy Vault Scheme - Shortcomings

- **Intra-class variability**: rotation, translation, # minutia points
  - *'helper data'* reduces security
- Increasing the degree of the **polynomial** increases complexity
  - require increased number of minutiae points
- Increasing the number of **chaff points** increases the complexity
  - empirical bound because of minutiae location

BioHashing

Goh *et al.* perform *bio-hashing*

- Based on principal component analysis (PCA)
- Focus only on the first phase of key generation.
  Our approach
  - couples phase-one and phase-two of key generation
  - analyzes inter and intra-class variations
  - analyzes security and privacy of the biometric verification system

Thank you!

- Abhilasha Bhargav-Spantzel
  Intel Corporation
  email: abhilasha.bhargav-spantzel@intel.com

Tools used

Singular Value Decomposition (SVD)

If A is a real *m-by-n* matrix, the two orthogonal matrices exist:

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{m \times m} \text{ and } V = [v_1, \ldots, v_n] \in \mathbb{R}^{n \times n}$$

such that

$$UAV^T = diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{m \times n} \qquad p = min\{m, n\}$$

where V^T is the transpose of matrix V and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$. σ_i's are the *singular values* of A and the vectors u_i and v_i are the ith *left singular vector* and the ith *right singular vector* respectively.

Tools used (cont.)

Support Vector Machines (SVM)

- SVM is a classifier based on statistical learning technique developed by Vapnik *et al.*
- It aims at finding optimal hyperplanes to determine the boundaries with the maximal margin separation between every two classes.

SVM applies to classification of vectors, or uni-attribute time series. To classify multi-attribute biometric image data, which are matrices rather than vectors, the multi-attribute data are transformed into uni-attribute data or vectors using SVD.

Tools used (cont.)

Support Vector Machines (SVM)

- SVM is a classifier based on statistical learning technique developed by Vapnik *et al.*
- It aims at finding optimal hyperplanes to determine the boundaries with the maximal margin separation between every two classes.

SVM applies to classification of vectors, or uni-attribute time series. To classify multi-attribute biometric image data, which are matrices rather than vectors, the multi-attribute data are transformed into uni-attribute data or vectors using SVD.

Key Steps of Biometric Hashing Algorithm

1: Input biometric image I
2: **for** each *random* A_i where $1 \leq i \leq p$ **do**
3: $A_i = U_i S_i V_i^T$ {**First SVD Transform**}
{Collect singular vectors corresponding to the largest singular value}
4: $\overrightarrow{u_i}$ = first left singular vector
5: $\overrightarrow{v_i}$ = first right singular vector
6: **end for**
7: $\Gamma = \{\overrightarrow{u_1}, \ldots, \overrightarrow{u_p}, \overrightarrow{v_1}, \ldots, \overrightarrow{v_p}\}$
8: *Randomly* create $J[m, 2p]$ from Γ {**Second SVD Transform**}
9: $J = U_J S_J V_J^T$ {Collect singular vectors corresponding to the largest singular value}
10: $\overrightarrow{u_J}$ = first left singular vector
11: $\overrightarrow{v_J}$ = first right singular vector
12: $\overrightarrow{H} = \{\overrightarrow{u_J}, \overrightarrow{v_J}\}$

Fuzzy Vault Scheme - Shortcomings

Attacks on Fuzzy Vault

In August 2007, Preda Mihailescu presented a brute force attack in three known implementations of the vault for fingerprints.
The vulnerability cannot be avoided by mere parameter selection in the actual frame of the procedure.

Thoughts on
Personal Identity Platforms

William I. MacGregor

IDTrust 2010

Foreword

This is a thought experiment…

...to show feasibility…

...and is doubtless reinvention.

National Strategy for Secure Online Transactions

"To improve trustworthiness and security of online transactions by … interoperable trust frameworks and … improved authentication technology and processes … across federal, civil, and private sectors."

- SecureIDNews, 1Apr2010, by Zack Martin

- Protect Privacy: secure PII & transaction data
- Defeat Fraud: reduce losses & improve recovery
- Promote Confidence: increase trust in online transactions

Three Questions

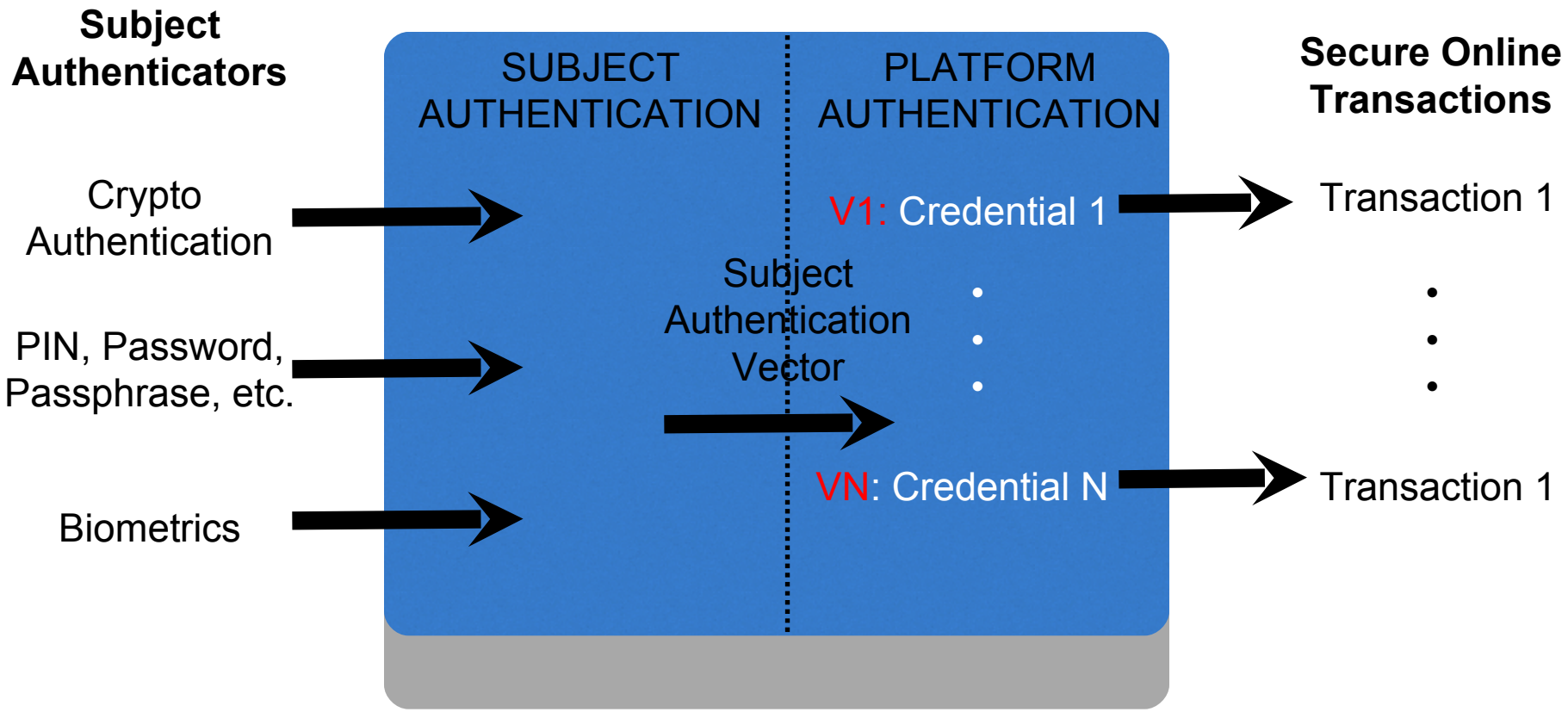
1. Could leakage of subject authenticators be prevented?

2. What are the characteristics of a solution to Question 1?

3. Does strong attribute assurance require strong identity assurance?

Personal Identity Platform

An answer to Question 1



The subject trusts the PIP to present only the selected credential; the relying party trusts the PIP to perform subject authentication first.

Characteristics of PIP

An answer to Question 2

- The PIP is a trust intermediary between the subject and relying party
- Only the Subject Authentication Vector is known to Credentials
- Credentials belong to the subject because they reside on the subject's PIP
- "Platform authentication" is also "SAML generation" or "session key agreement"

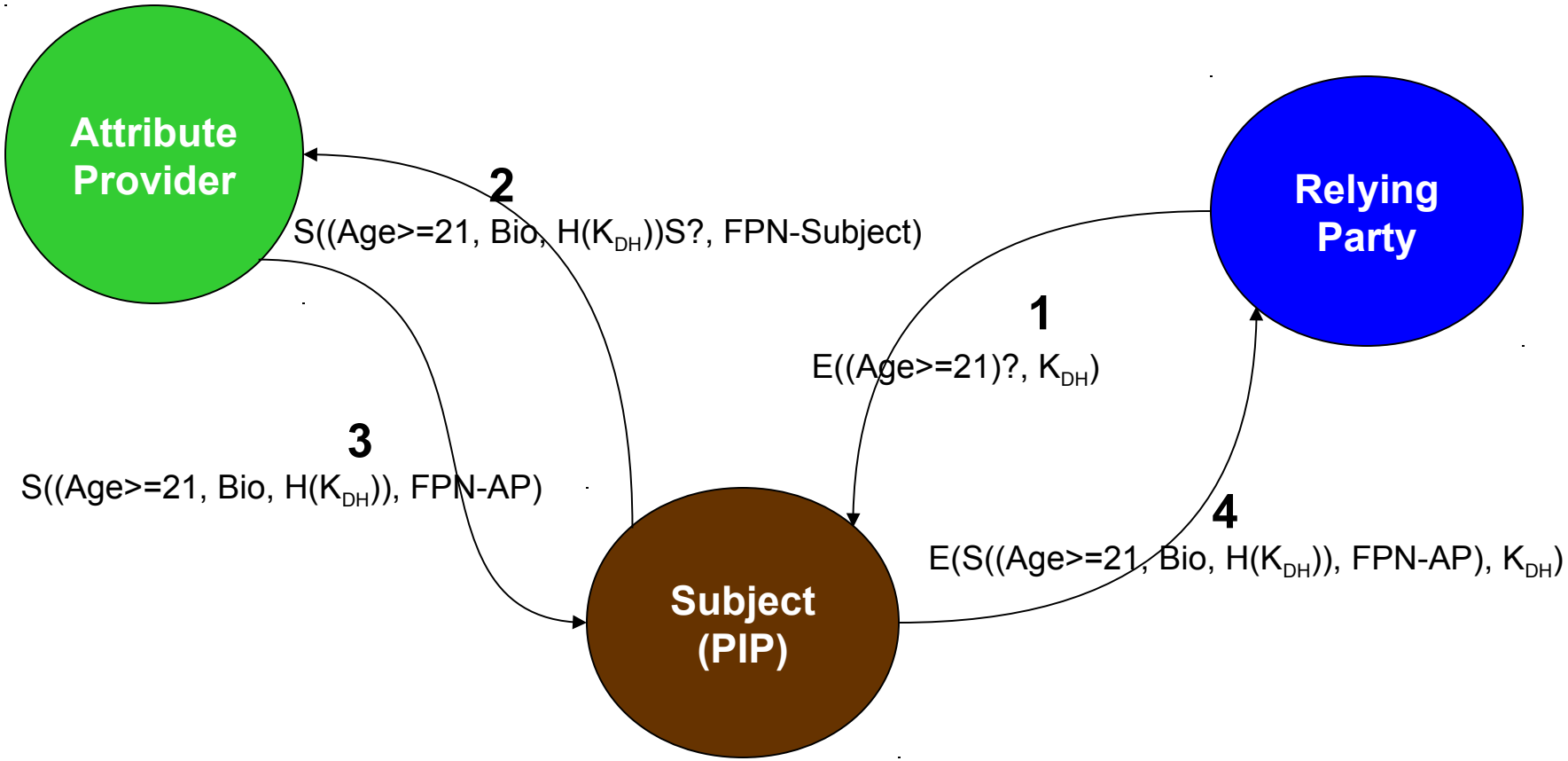
Requirements for a PIP
Another answer to Question 2

- The PIP must be available to, and controlled by, the subject
- The PIP must be a competent computing device or system
  - HIDs, biometrics, crypto, comm, clock, etc.
- The PIP must be coupled into the subject's transaction stream

What have I left out?

Strong Attribute Assurance

An answer to Question 3



The Result

The answer to Question 3: No

- The PIP claims that FPN-Subject is bio authenticated, and the PIP in session $H(K_{DH})$
- The AP claims that subject Age>=21 is bio authenticated, for PIP in session $H(K_{DH})$
- The RP trusts the PIP and AP, so believes the authenticated subject has Age>=21
- The AP does not learn the RP; the RP does not learn any static subject identifier

About Attributes

- Why have Attribute Providers **and** Identity Providers?
  - Go to the source—IDPs aren't all sources
- Why have dynamic attributes?
  - Attributes change—shouldn't be in static credentials
- Examples
  - Conditions of probation
  - Permit to carry
  - EMT certification

Thanks for listening!

Useful references

U-Prove https://connect.microsoft.com/content/content.aspx?contentid=12505&siteid=642

Selective attribute delivery designed to meet privacy objectives.

ISO/IEC 24727 http://csrc.nist.gov/publications/nistir/ir7611/nistir7611_use-of-isoiec24727.pdf

Standard for construction of platforms like PIP.

SASSO http://www.projectliberty.org/liberty/content/download/3960/26523/file/NTT-SASSO%20liberty%20case%20study.pdf

Implementation of a federated IDP provider in a USIM smart card in a mobile phone.

Practical and Secure Trust Anchor Management and Usage

Carl Wallace
Cygnacom Solutions
7925 Jones Branch Drive Suite 5200
McLean, VA 22102

cwallace@cygnacom.com

Geoff Beier
Cygnacom Solutions
7925 Jones Branch Drive Suite 5200
McLean, VA 22102

gbeier@cygnacom.com

ABSTRACT
Public Key Infrastructure (PKI) security depends upon secure management and usage of trust anchors. Unfortunately, widely used mechanisms, management models and usage practices related to trust anchors undermine security and impede flexibility. In this paper, we identify problems with existing mechanisms, discuss emerging standards and describe a solution that integrates with some widely used applications.

Categories and Subject Descriptors
K.6.5 [**Management of Computing and Information Systems**]: Security and Protection - *authentication*.

General Terms
Security

Keywords
Trust anchor management, public key infrastructure (PKI).

1. INTRODUCTION
Trust anchors (TAs) are used for a variety of purposes. For example, trust anchors are used when a web browser authenticates a web server, when an email client verifies a signature on an email message or prepares an encrypted email message and when a domain controller authenticates a user logging in with a smart card. In short, a TA is used whenever a PKI is securely used. Trust Anchor Management Requirements [6] provides the following definition for a TA:

"A Trust Anchor is a public key and associated data used by a relying party to validate a signature on a signed object where the object is either:

- *a public key certificate that begins a certification path terminated by a signature certificate or encryption certificate*
- *an object, other than a public key certificate or certificate revocation list (CRL), that cannot be validated via use of a certification path."*

Trust Anchor Management Requirements [6] also provides a definition for a trust anchor store:

"A trust anchor store is a set of one or more trust anchors stored in a device. A trust anchor store may be managed by one or more trust anchor managers. A device may have more than one trust anchor store, each of which may be used by one or more applications."

In current practice, a trust anchor is a (typically self-signed) certificate that resides in a trust anchor store. Despite their importance, trust anchor stores are usually managed, to a large extent, by software vendors. Trust anchor store users have few or no enforceable constraints available to limit the extent of trust accorded to the trust anchors in the trust anchor store or to the software vendor managing the trust anchor store.

This paper briefly describes current trust anchor management tools and practices, identifies some problems with the status quo and describes an implementation that provides alternative trust anchor management mechanisms for applications that use the Microsoft Crypto API (CAPI) certification path processing interfaces.

2. Current Trust Anchor Management and Usage
In most common scenarios, trust anchors are distributed and managed by operating system and application vendors. TA stores are initialized during software installation and are often are changed by software updates. Proprietary operating system-specific or application-specific tools are used to customize trust anchor store contents. These actions may be undone, however, by automated trust anchor store updates or routine software updates.

IDtrust '10, April 13–15, 2010, Gaithersburg, Maryland, U.S.A.

Synchronization of trust anchor stores from different vendors (or even the same vendor) requires manual steps using proprietary tools. Comparison of trust anchor store contents is a similarly manual affair.

Most operating systems and applications use certificates to represent trust anchor information. In some cases, a collection of trust anchors may be represented using a "certificates only" Cryptographic Message Syntax (CMS) SignedData message. Some applications may require distinguished encoding rules (DER) encoded certificates or privacy enhanced mail (PEM) encoded certificates, but this is a fairly minor problem as conversion tools are readily available.

The following sections provide an overview of some widely used mechanisms and discuss the primary problems with these mechanisms.

2.1 Overview of selected current mechanisms

2.1.1 Microsoft Windows

Many applications that operate on Microsoft Windows platforms use the trust anchor stores built into the operating system. A variety of interfaces are available for adding trust anchors to a trust anchor store, including the following:

- Right-clicking a certificate file, choosing "Install Certificate" from the resulting menu and selecting a trust anchor store destination,
- Installing a certificate into a trust anchor store using the Microsoft Management Console (MMC),
- Installing a certificate into a trust anchor store using an application-provided interface, such as Internet Explorer (IE),
- Installing a certificate into a trust anchor store using group policy or System Center Configuration Manager (SCCM).

The MMC interface to the trust anchor store is shown below.

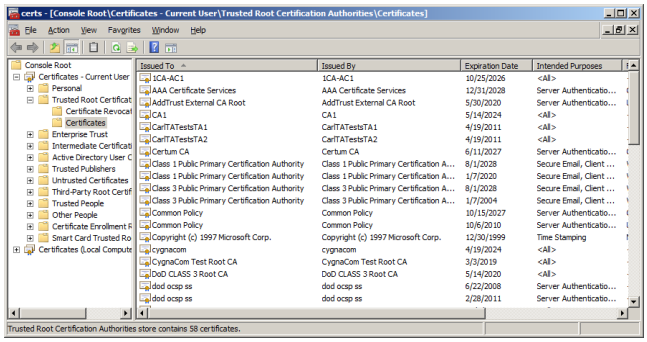


Figure 1 MMC view of a trust anchor store

Some options, such as MMC and Internet Explorer, allow for the specification of certain trust anchor constraints, which are referred to in the user interface as properties or purposes. Constraints are configured using a dialog like the one shown below in Figure 2. The constraint options are very similar to extended key usage values, with a difference being that extended key usage extensions are not processed across a certification path but the constraints configured here appear to be. On Windows Vista SP 2 systems, there are 38 purposes available for selection. When a trust anchor is manually installed, all purposes are enabled by default.

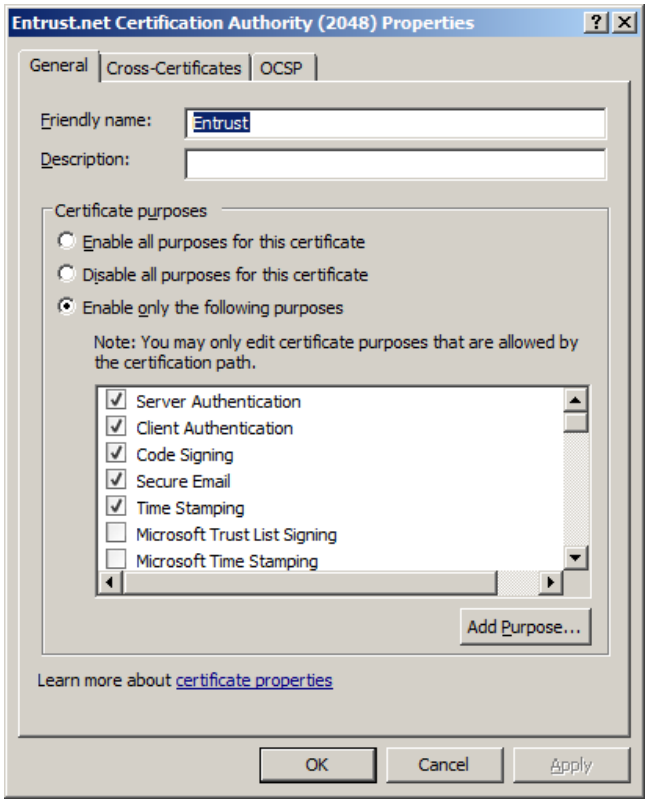


Figure 2 Microsoft trust anchor constraints dialog

In addition to manual trust anchor installation, Windows provides automatic trust anchor store update mechanisms, with different versions of Windows providing somewhat different capabilities. When these features are enabled, a trust anchor may be automatically installed with no visual cue provided to the operator, for example, when a certificate file subordinate to that trust anchor is simply inspected using the Windows certificate viewer a corresponding trust anchor may be downloaded and installed. Trust anchors installed automatically do not necessarily have all purposes enabled.

Trust anchor stores are maintained in the system registry. Trust anchors are imported and exported as certificates. The certificates are stored in the registry along with property information. When trust anchors are exported, the user-configured constraints are not conveyed along with the exported certificates.

2.1.2 *Firefox*

Firefox does not use Microsoft Windows trust anchor stores. Trust anchors are added to the Firefox trust anchor store using the Certificate Manager dialog shown below in Figure 3. This dialog is accessed by invoking the Tools->Options menu and selection the Encryption tab from the Advanced options.

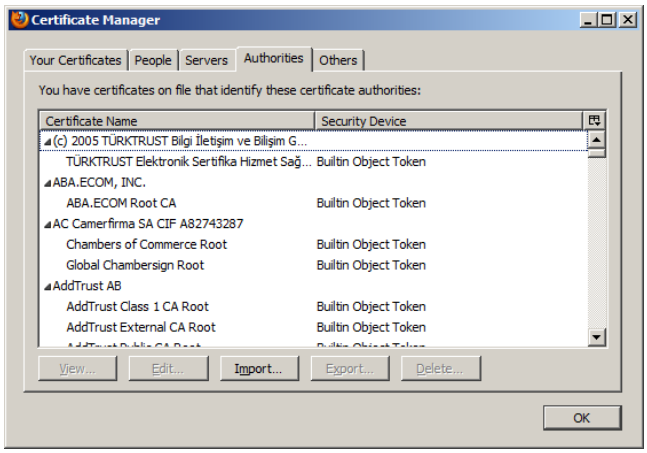


Figure 3 Firefox trust anchor store

Trust anchor constraints can be configured by clicking the Edit button and selecting the desired properties in a dialog like the one shown below, which allows three properties to be enabled. As with Microsoft Windows, the properties are similar to values that are typically expressed via an extended key usage extension.

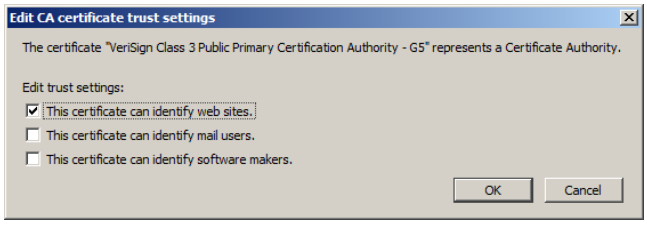


Figure 4 Firefox trust anchor constraints

Firefox trust anchors are maintained in a database that resides in the profile of a Firefox user. Trust anchors are imported and exported as certificates.

2.1.3 *Mac OS X*

Mac OS X maintains trust anchors in the key chain. Trust anchors are added to the trust anchor by invoking the Keychain Access application, as shown below.

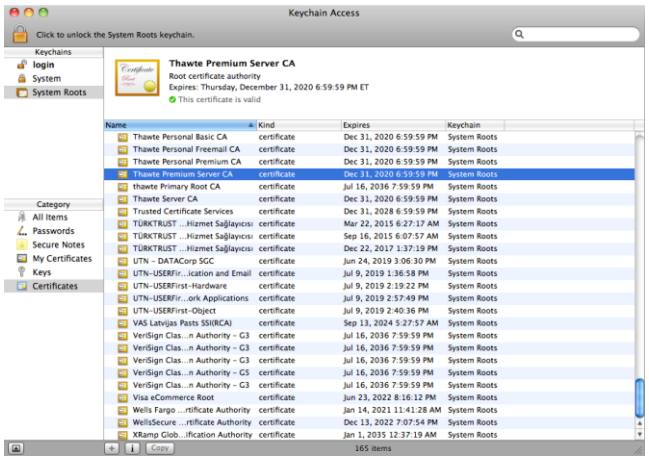


Figure 5 Mac OS X version 10.6 trust anchor store

Trust anchor information, including usage constraints, can be viewed by right-clicking a trust anchor in the Keychain Access application and choosing Get Info. Nine properties are available. As with Microsoft Windows and Firefox, the properties are very similar to values typically expressed via an extended key usage extension.

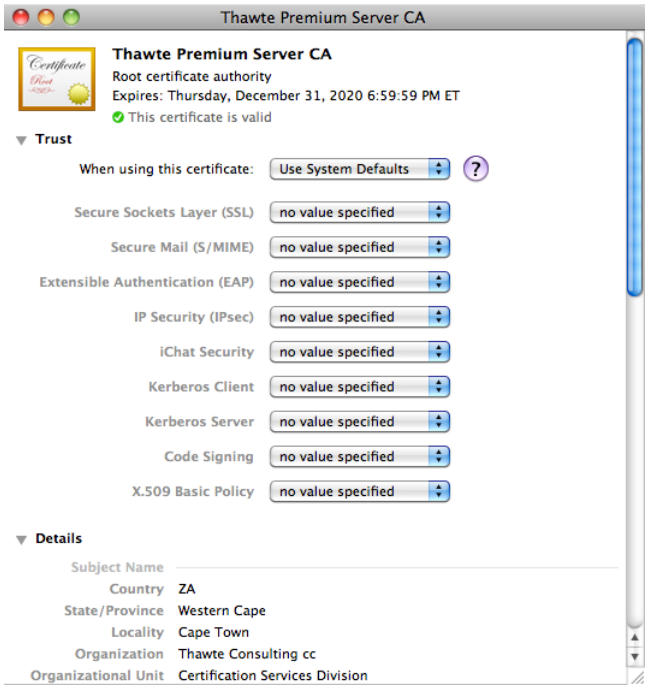


Figure 6 Mac OS X version 10.6 trust anchor constraints

As with Microsoft Windows and Mozilla trust anchor stores, trust anchors are exported as files containing X.509 certificates, and no user-specified constraints are conveyed along with these certificate files.

2.2 Primary problems with current mechanisms

This paper does not aim to catalog problems with existing trust anchor management mechanisms. However, this section discusses

some problems in the areas of trust anchor store management and trust anchor constraints enforcement.

2.2.1 Trust anchor store management

Management of trust anchor stores requires usage of proprietary tools. Where necessary, system administrators must take care to synchronize the contents of multiple trust anchor stores. This requires configuration of trust anchor constraints as well as ensuring trust anchors are installed in (or removed from) the necessary trust anchor stores.

Maintenance of trust anchor store contents is complicated by the fact that software updates frequently adjust trust anchor store contents (sometimes undoing changes made by the system administrator). Automatic trust anchor update mechanisms can create de facto trust anchor stores that contain more trust anchors than are visible to administrators using the available tools.

Trust anchors do not offer any integrity protection or "in-band" security mechanisms. Confirmation that the correct trust anchor is being installed typically requires manual checks.

2.2.2 Constraint representation

As shown in Section 2.1, different trust anchor stores enable the usage of different, non-standard trust anchor constraints. These constraints are stored using a proprietary format. When trust anchors are exported from the trust anchor store the constraint information is lost.

The certification path validation algorithm described in RFC 5280 [1] only makes use of the public key and name of a trust anchor. Implementations are free to perform processing beyond that required by RFC 5280 [1], such as to impose name constraints or certificate policy requirements on a trust anchor. However, there is no standardized process for doing so. This lack of standardization has resulted in inconsistent means of specifying constraints and poor interoperability. Complicating matters is the fact that trust anchors are almost always represented as certificates. Though the signature on the trust anchor's certificate provides little security value, it interferes the editing of certificate contents.

2.2.3 Constraint enforcement

Enterprise PKI operators use cross-certificates to establish trust between enterprises and employ a variety of constraints, i.e., extensions, to limit the degree of trust accorded to the cross-certified PKI. However, cross-certificates are not always a viable option. In some cases, however, a trust relationship may only be appropriate for a small subset of subscribers to an Enterprise PKI. In these cases, directly trusting a trust anchor is an alternative. Unfortunately, existing trust anchor constraint mechanisms do not provide a set of constraint options comparable to those available when using a cross-certificate, making direct trust difficult to use.

For an example of problems caused by lack of trust anchor constraints, consider the community surrounding the Federal Bridge CA (FBCA). Each CA that has issued a cross-certificate to the FBCA creates a large number of potential certification paths that traverse that cross-certificate. Some enterprises, such as the Department of Defense, have adopted an approach to cross-certifying with the FBCA that allows application owners to opt out of the cross-certification by recognizing alternative trust anchors that are not connected to the FBCA. A problem arises when entities who have "opted out" need to establish a trust relationship with another CA that is cross-certified with the FBCA. Simply recognizing the CA as a trust anchor will establish the trust relationship but causes the entire FBCA community to be recognized as well. This could be avoided if it were possible to constrain a trust anchor using similar mechanisms as those used in cross-certificates.

3. Next Generation Specifications

The Internet Engineering Task Force (IETF) is presently working on several specifications related to trust anchor management and usage, including: Trust Anchor Management Protocol (TAMP) [4], Trust Anchor Format (TAF) [3], CMS Content Constraints (CCC) [2], Using Trust Anchor Constraints during Certification Path Processing (UTAC) [5]. These specifications provide complementary features, but subsets of features can be implemented where the full feature set is not required.

The following subsection briefly introduce each of these specifications, which were used in the implementation described in Section 4.

3.1 Trust Anchor Format

TAF [3] provides syntax for representing trust anchors. The primary structure is TrustAnchorChoice:

```
TrustAnchorChoice ::= CHOICE {
    cert    Certificate,
    tbsCert [1] EXPLICIT TBSCertificate,
    taInfo  [2] EXPLICIT TrustAnchorInfo
}
```

This structure provides support for existing trust anchors represented as certificates and provides two mechanisms that allow relying parties to customize the definition of a trust anchor: TBSCertificate and TrustAnchorInfo. Using the TBSCertificate option, the signature is simply removed from a Certificate structure allowing the contents to be edited. Using TrustAnchorInfo, a Certificate can be wrapped, with additional or alternative constraints defined in the wrapper or a name and public key can be used with or without additional information.

3.2 Trust Anchor Management Protocol

TAMP [4] defines eleven message formats and a set of processing rules that can be used to manage trust anchor store contents. Each of these message formats, or content types, can be encapsulated using a CMS SignedData structure to provide source authentication and message integrity. The eleven messages consist of five request/response pairs and a generic error message:

- TAMPStatusRequest
- TAMPStatusResponse
- TAMPUpdate
- TAMPUpdateConfirm

- TAMPApexUpdate
- TAMPApexUpdateConfirm
- TAMPCommunityUpdate
- TAMPCommunityUpdateConfirm
- SequenceNumberAdjust
- SequenceNumberAdjustConfirm
- TAMPError

3.2.1 Reviewing TA store contents

TAMPStatusResponse messages provide a means of representing trust anchor store contents. As with most TAMP response/confirm messages, the message can be either verbose or terse. A verbose TAMPStatusResponse message provides a comprehensive set of information regarding a trust anchor store, including a list of all trust anchors, an indication of which TA is the apex trust anchor (if any) and information on TAMP sequence numbers and TAMP communities. A terse TAMPStatusResponse provides only trust anchor key ids along with communities of which the store is a member. A TAMPStatusRequest simply asks a trust anchor store to provide its contents in the requested message format, i.e., verbose or terse. Use of TAMPStatusRequest and TAMPStatusResponse can reduce reliance on proprietary tools for TA store management and simplify comparison of TA store contents.

3.2.2 Editing TA store contents

TAMPUpdate messages allow new trust anchors to be added to a trust anchor store, existing trust anchors to be changed or existing trust anchors to be removed. Each TAMPUpdate message contains a set of one or more commands (i.e., add, change, remove). Since TAMPUpdate messages are signed, in-band integrity and source authentication checking is enabled.

3.2.2.1 Subordination rules

TAMP defines a strict set of subordination rules that apply when a TAMPUpdate message is processed. These rules allow limits to be placed on TA store managers. These rules could be used to place constraints on automated updates, such as to ensure an undesirable trust anchor is not restored after it has been removed by a local management action, or to ensure that a trust anchor rekey operation does not exceed locally-imposed constraints on the old key.

3.2.3 Replacing the Apex TA

TAMP [4] introduces the concept of the Apex TA, which is defined as being the single trust anchor within a trust anchor store that is superior to all other trust anchors. This concept is primarily used as a disaster recovery technique. Essentially, a trust anchor store is created with a single Apex TA in place. Authority over various management operations is then delegated to other trust anchors that are added to the trust anchor store or to certificate holders. Management operations are conducted by the delegates with the Apex TA private key maintained in secure storage. As an extra safeguard, a contingency public key can be included in the definition of the Apex TA. The contingency public key corresponds to a private key that is intended to be used once to replace the Apex TA in the event of loss or compromise of the operational Apex TA private key.

3.2.4 Managing TAMP community membership

TAMP messages can be created such that all TA stores that recognize the TA store manager will accept the message, a group of TA stores will accept the message or a specific TA store will accept the message. Community identifiers are one means for addressing a group of trust anchor stores. TAMP-enabled trust anchor stores should have the ability to store a list of community identifiers. TA store managers can use these identifiers to create arbitrary groups of trust anchor stores for future management purposes.

TAMPCommunityUpdate messages are used to add or remove community identifiers from a trust anchor store. TAMPCommunityUpdateConfirm is used to report the results of processing a TAMPCommunityUpdate message.

3.2.5 Managing TAMP sequence numbers

TAMP uses sequence numbers to detect attempts to process old TAMP messages. Each TAMP-enabled trust anchor store maintains a sequence number for each trust anchor authorized for TAMP (and may maintain a sequence number for certificate holders who have been authorized for TAMP). A SequenceNumberAdjust message can be used to convey the current sequence number to a trust anchor store to reduce the likelihood of replay. A SequenceNumberConfirm message is used to indicate the results of processing the SequenceNumberAdjust message.

3.3 CMS Content Constraints

A basic problem for any trust anchor management protocol is authorization of management operations. Certification authorities are authorized to issue cross-certificates using constraints expressed as certificate extensions, e.g., basicConstraints, certificatePolicies, etc. CCC [2] defines an authorization mechanism that can be used with TAMP.

CCC is a generic mechanism for authorizing public key certificate holders to originate specific types of information protected using the Cryptographic Message Syntax (CMS). A set of content types is expressed in the CCC extension. When a CMS-protected message is processed, the originator is authenticated and the CCC extension associated with the originator is inspected to ensure the given content type is permitted.

For TAMP, this mechanism can be used to authorize some entities to manage trust anchor stores and others to review the contents of trust anchor stores while leaving other entities with no privileges at all. To authorize an entity to manage trust anchor stores, include, in either the entity's certificate or trust anchor, a CCC extension with the TAMPUpdate, CommunityUpdate, SequenceNumberAdjust and TAMPStatusQuery content types permitted. To authorize an entity to review the contents of trust anchor stores, include a CCC extension in the entity's trust anchor or certificate with the TAMPStatusQuery content type permitted.

3.4 Using Trust Anchor Constraints during Certification Path Processing

UTAC [5] augments the certification path processing algorithm specified in RFC 5280 [1] by describing how to use constraints contained in a trust anchor during certification path processing. Essentially, the constraints contained in a trust anchor are intersected with those provided by a user. The results of this intersection are used as the inputs to the RFC 5280 [1] certification path validation algorithm. This allows a trust anchor store manager (i.e., an enterprise) to establish a minimum set of restrictions on the usage of a trust anchor without removing the ability of an application (i.e., a user) to provide inputs to the path validation algorithm.

UTAC [5] describes rules for using constraints in a TrustAnchorInfo wrapper relative to constraints resident in a certificate that is wrapped, i.e., the wrapper takes precedence. UTAC processing can be integrated directly into an RFC 5280 path validation implementation or as pre or post processing.

4. Integrating Trust Anchor Management with CAPI

The goal of the implementation effort described in this paper was to enable the usage of emerging trust anchor management specifications with commonly deployed commercial off-the-shelf (COTS) products which have been public key-enabled using Microsoft Crypto API (CAPI). This integration aims to enforce constraints associated with a trust anchor. To achieve this, the software must be able to influence the outcome of a certification path validation operation performed by CAPI.

Since there is no publicly documented set of APIs intended for this purpose, existing APIs intended for other purposes were evaluated to determine suitability for integration of trust anchor management functionality. The following interfaces were analyzed: revocation status provider, validation policy provider and certificate store provider.

4.1 Revocation Status Provider

The initial approach that was considered was to use the revocation status provider interface. Revocation status providers are typically used to provide support for Online Certificate Status Protocol (OCSP). A revocation status provider is a dynamic link library (DLL) that implements the CertVerifyRevocation API. The provider is registered with the operating system. The registration information consists of the full path and filename of the revocation status provider and is stored in a registry key containing a list of string values. The list of providers can be ordered according to system administrator preference. Providers are invoked in turn until a one is found that can provide revocation status information for the certificate in question.

When an application validates a certification path, the provider is loaded by CAPI and invoked once for the end entity certificate and each intermediate CA certificate contained in a certification path validated by CertGetCertificateChain or WinVerifyTrust. The provider can cause a path validation operation to fail by indicating the given certificate is revoked.

This approach was not implemented for two reasons. First, the interface is invoked for each certificate in a path, not for an entire certification path. This means the provider would need to maintain state across multiple invocations in order to get a view of the entire path. Second, while this could effectively cause a certification path that violates trust anchor constraints to fail, the error indicated by the provider creates a misimpression that a certificate is revoked. This kind of misreported failure leads to a poor user experience in the desktop applications that are targeted in this effort.

4.2 Validation Policy Provider

Next, the validation policy provider interface was explored. This interface is not as comprehensively documented and less widely used than the revocation status provider interface. Like the revocation status provider interface, a validation policy provider is registered with the operating system and loaded by CAPI during certification path processing. Unlike the revocation status provider interface, providers do not failover from one to another. Providers can be registered for a specific validation policy. However, the processing performed by default policy providers is not documented and replacing the default providers is not recommended. No way could be found to invoke the default providers from a third party provider.

We implemented policy providers for several of the default policies but abandoned the effort due to inconsistent invocation of the installed replacement policy provider. For example, within Microsoft Outlook, the replacement policy provider was invoked when no certification path was found for a message signer but not when a certification path was found.

4.3 Certificate Store Provider

While performing the analysis of the validation policy provider API, we used code interception to inspect and log parameter values. After discarding the revocation status provider and validation policy provider efforts, we focused on finding a means of using code interception as the basis for performing the integration. This required identifying opportunities where code could be loaded prior to the CertGetCertificateChain API and unloaded afterwards, enabling the CertGetCertificateChain to be intercepted. For most applications, the certificate store API provides such an opportunity.

We implemented a certificate store provider that is registered with the operating system as a CA store provider in the HKEY_LOCAL_MACHINE registry hive. When the certificate store is loaded, hooks are created for the CertGetCertificateChain API. No certificate store functionality is actually provided.

To limit the scope of the provider, configuration information can be saved on a per-application basis. When an application that does not require the trust anchor management services implemented by the provider loads it, no hooks are set. The

certificate store provider is loaded into memory but performs no code interception.

A side benefit of this integration approach is the ability to fully replace CAPI certification path processing instead of simply enforcing trust anchor constraints following discovery of a certification path. This enables the usage of the Server-based Certificate Validation Protocol (SCVP) or alternative local certification path processing engines for both path discovery and validation. Though the software described below supports this option, it is not discussed further in this paper. Nor are issues associated exclusively with the provision of SCVP support.

As noted above, integration via the certificate store API proved workable for most applications that were tested but not all. For Internet Explorer, it was necessary to build a browser add-on that causes the certificate store to be loaded before the browser can be used to access SSL/TLS-protected websites. The browser add-on simply forces CertOpenStore to be called by validating a path to a hard-coded trust anchor, which was selected from the list of required trust anchors defined in Microsoft knowledge base article number 293781.

5. CAPI Trust Anchor Guard (CAPI TAG)

CAPI Trust Anchor Guard (CAPI TAG) is a set of software tools that enable management of a local or remote trust anchor store using TAMP and enforcement of trust anchor-based constraints for applications that use CAPI for certification path processing.

5.1 Overview

CAPI TAG consists of eight primary components: PKIFTAM, CAPI TAG Store Creator, Store Manager, mod_tam, Process TAMP Message, CAPI TAG, CAPI TAG Config and CAPI TAG Customization Wizard.

5.1.1 PKIFTAM

PKIFTAM.dll provides basic encoding and decoding functionality for structures defined in TAF [3], TAMP [4] and CCC [2]. Additionally, it provides classes that can be integrated with the PKIF library (www.pkiframework.com) to enforce TA constraints using a TA store managed with TAMP.

5.1.2 CAPI TAG Store Creator

CapiTagStoreCreator.exe is used to initialize a CAPI TAG trust anchor store. A trust anchor store can be created using trust anchors from a CAPI trust anchor store or a file folder.

5.1.3 Store Manager

StoreManager.exe is the primary trust anchor management tool. It can be used to manage local trust anchor stores, remote trust anchor stores accessed via HTTP or remote trust anchor stores via a file containing a TAMPStatusResponse message generated by the target trust anchor store. The user interface in Store Manager is mostly driven by TAMP messages, and all operations are possible regardless of access method, provided the operator possesses an authorized signing key. The primary interface to manage trust anchors using Store Manager is shown below.

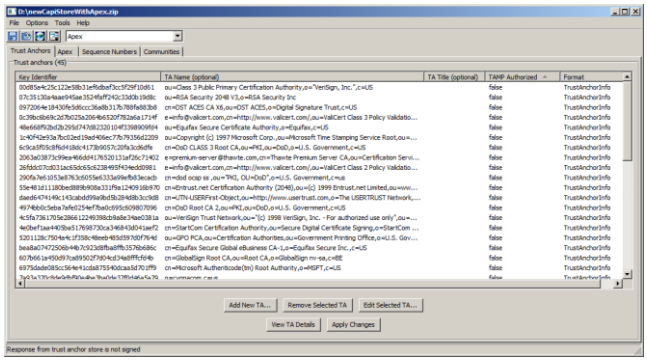


Figure 7 Store Manager trust anchor list

Using Store Manager, trust anchors can be added to a TA store, removed from a TA store or edited. When a trust anchor is added, its format can be changed from certificate to TBSCertificate or TrustAnchorInfo, enabling the expression or alteration of constraints.

Trust anchor constraints are edited using dialogs provided with the PKIF library. These allow the expression of constraints that align with the standard path validation algorithm inputs as defined in RFC 5280 [1]. The constraints editing dialog is shown below.

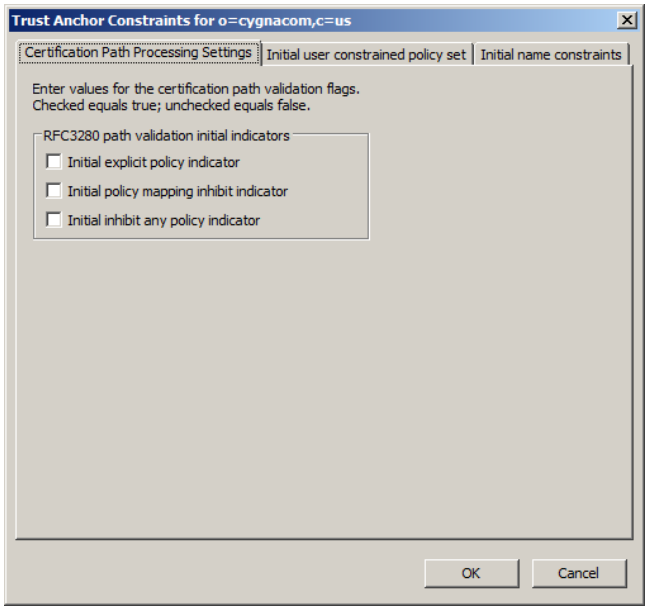


Figure 8 Editing trust anchor constaints in Store Manager

5.1.4 mod_tam

mod_tam is an Apache module that serves either or both of the following purposes:

- Routes TAMP messages received via a particular URI to a TA store file for processing
- Periodically check specified URIs for TAMP messages, which are downloaded and presented to a TA store file for processing.

This enables the suite to support either push or pull for TA management. mod_tam is accompanied by an optional system tray notification applet that allows the user to see desktop alerts as TAMP messages are processed.

5.1.5 Process TAMP Message

ProcessTampMessage.exe allows a file containing a TAMP message to be presented to a CAPI TAG trust anchor store for processing. The store can be addressed either as a local file or using an HTTP URI. Unlike Store Manager, the operator of Process TAMP Message need not have any TAMP privileges (or even possess a private key).

5.1.6 CAPI TAG

CapiTag.dll integrates with Microsoft Windows operating systems to provide trust anchor constraints enforcement or alternative certification path processing.

5.1.7 CAPI TAG Config

CapiTagConfig.exe is the primary means for configuring CapiTag.dll for use. It enables the configuration of default settings and application-specific settings. All configuration information is stored in the system registry.

5.1.8 CAPI TAG Customization Wizard

CapiTagCustomizationWizard.exe is used to create transform files (.mst) that can be used to customize the CapiTag.msi installation package for use in a particular environment. The wizard allows customization of the following aspects of a CAPI TAG deployment:

- Inclusion of one or more CAPI TAG trust anchor stores
- Customization of Store Manager PKI settings (i.e., used when validating TAMP messages generated by a CAPI TAG TA store)
- Customization of CAPI TAG trust anchor store PKI settings (i.e., used when validating TAMP messages generated by Store Manager)
- Customization of CAPI TAG PKI settings (i.e., used when enforcing TA constraints or to configure alternative certification path processing)
- Customization of CAPI TAG settings (i.e., default or per-application settings)
- Specification of a customized mod_tam configuration file

5.2 Trust Anchor Management

Using CAPI TAG, several trust anchor management models are possible. As shown in Figure 9, the management models considered here are: local management, online remote management, indirect remote management and remote pull. The terms local and remote refer to the relative positions of the trust anchor store and the trust anchor manager's private key. For local management scenarios, the TA store and TA store manager's private key are collocated[1]. For remote management scenarios, the TA store and TA store manager's private key need not be collocated.

Given that CAPI TAG trust anchor stores are files, the contents could be prepared in one location and distributed using means like group policy. With minor additions to the current specification suite, additional models including usage of a subjectInformationAccess-based pointer or trust anchor store-initiated client/server exchange are possible.

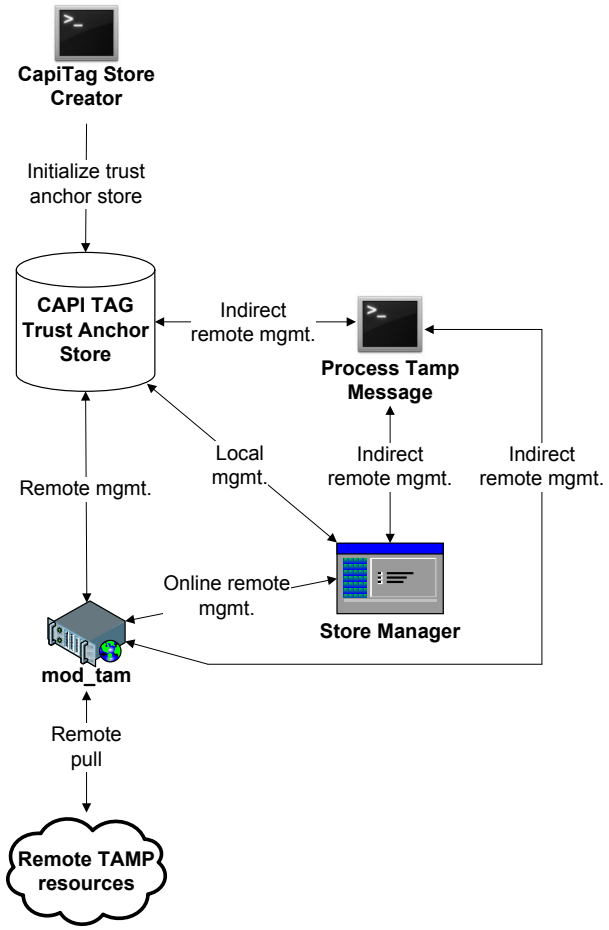


Figure 9 Management models

[1] This taxonomy is quite loose. Files accessed over a local area network are considered "local" despite the fact that the TA store resides on a different physical machine. Similarly, files accessed over HTTP to a local mod_tam service are considered "remote".

5.2.1 Local management

Using the Store Manager application, a CAPI TAG trust anchor store file can be opened and queried using a TAMPStatusQuery message. The Store Manager operator's private key must be available and the target trust anchor store must recognize the operator as authorized to originate TAMPStatusQuery messages (no other permissions are required to simply review the contents of a TA store).

If the operator is authorized to edit TA store contents, changes can be made and saved using Store Manager.

5.2.2 Online remote management

Using the Store Manager application, a CAPI TAG trust anchor store can be managed via HTTP by entering the URI corresponding to the desired trust anchor store. This will establish a connection to a mod_tam service, which will route TAMP messages to/from trust anchor stores collocated with the mod_tam service per the httpd.conf file.

As with local management, the operator may be authorized to edit the TA store or simply to review the TA store contents.

5.2.3 Indirect remote management

TA stores can be managed remotely through exchange of files containing TAMP messages. An entity with at least TAMPStatusQuery privileges can generate a TAMPStatusResponse message using Store Manager. The file containing the response can be provided to another entity with full TAMP privileges, who can then open the file using Store Manager and generate one or more TAMP messages to edit the TA store. These messages can be returned to the requesting entity for processing using the Process TAMP Message utility. To ensure security, the TA store should sign the TAMPStatusResponse.

5.2.4 Remote pull

A TA store manager can prepare TAMP messages using Store Manager for distribution via HTTP. The mod_tam service can be configured to periodically retrieve TAMP messages for zero or more URIs for processing by the indicated trust anchor store associated with the mod_tam instance.

In CAPI TAG, automated remote pull is not available without the mod_tam service. TAMP messages can be manually collected and processed using either ProcessTAMPMessage or the process externally generated TAMP message feature of Store Manager.

5.3 Trust Anchor Constraints Enforcement

CAPI TAG can be configured to enforce trust anchor constraints on a per-machine, per-user or per-application basis. When an application loads CAPI TAG, the most specific available configuration is used. The order of preference is as follows:

- Current user – application
- Local machine – application
- Current user – default
- Local machine – default

This allows a high degree of configurability for trust anchor stores and application PKI settings. Some applications can be configured to enforce trust anchor constraints, others can be configured to use an SCVP responder (or alternative local certification path processing implementation) and other applications can be configured to use native processing without TA constraints enforcement. This degree of configurability makes it easy to enforce constraints for key applications without impacting any legacy incompatible applications that need to run on the same system.

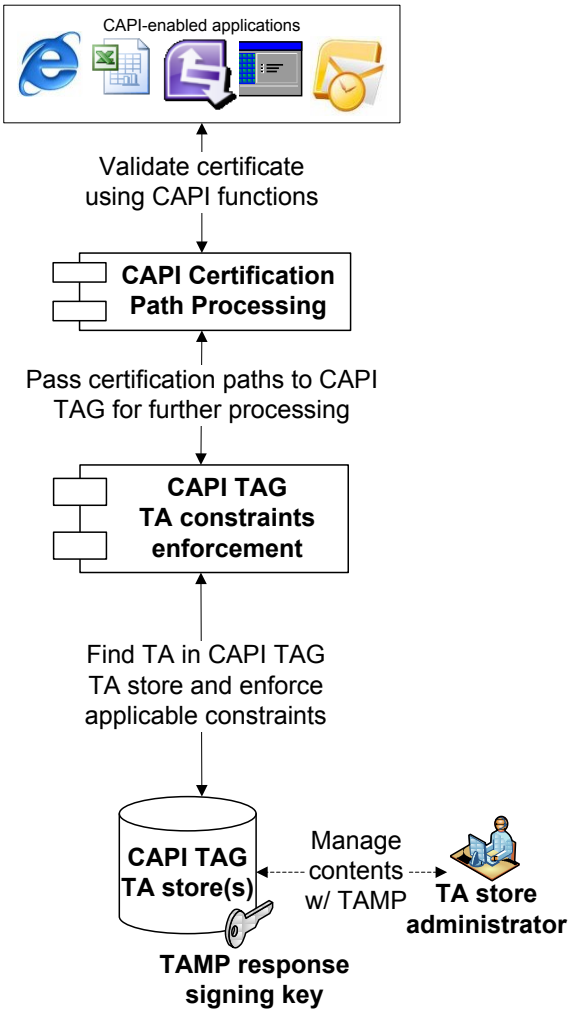


Figure 10 TA constraints enforcement with CAPI TAG

Since CAPI TAG uses a trust anchor store that is separate from CAPI trust anchor stores, the CAPI TAG trust anchor store manager's actions are not affected by changes made to the CAPI trust anchor store via automated trust anchor store updates or software upgrades. CAPI TAG can be configured to accept trust anchors from CAPI when a path is validated to a trust anchor not present in the CAPI TAG trust anchor store and can be configured to write trust anchors to a file folder, enabling the trust anchor store manager to adjust the contents of the CAPI TAG trust

anchor store as necessary. This feature reduces the difficulty of determining which trust anchors must be present and trusted to ensure that an application continues to function as users expect.

CAPI TAG can also be configured to not act for certain types of operations. For example, CAPI TAG can be configured to use only native functionality when a certificate is validated in support of a CAPI trust root list validation operation.

6. Summary

CAPI TAG demonstrates the effectiveness of the emerging IETF trust anchor management specifications in a typical, commercial software environment. CAPI TAG is intended to generate interest and discussion in trust anchor management and usage practices that ensure relying party interests can be satisfied. This section describes some challenges encountered while developing the CAPI TAG products and identifies some areas where additional standardization is potentially required.

6.1 Implementation experience

A primary challenge encountered during the development of the software was the lack of a proper interface for integrating enhanced trust anchor management capabilities and enforcement of trust anchor constraints. Not surprisingly, once an approach was identified it was also proved suitable for implementing an SCVP client. Most of the problems associated with the selected integration mechanism could easily be addressed if a means of utilizing alternative certification path processing implementations similar to that used for installing alternative revocation status providers were available.

Integration of non-certificate formats into a trust anchor store posed another challenge. This was solved by using a CAPI TAG-specific trust anchor store file format. Several challenges prevented the usage of existing mechanisms. The interfaces to existing trust anchor stores accept (usually self-signed) certificates. Trust anchor management messages were the desired format to support in-band integrity checks, authorization, subordination checks, etc. While it may have been possible to have overloaded the CertAddEncodedCertificateToStore to handle TAMP messages, this was not explored. For these reasons, TA store management was implemented as wholly independent of CAPI.

Read/write access to the trust anchor store file is managed by the operating system. CAPI TAG trust anchor store usage only requires read access. Write access can be limited to the mod_tam service, if desired. By default, though, system administrators have write access to CAPI TAG trust anchor store files. Authorization to manage trust anchor store contents via a TAMP interface is enforced using CCC.

Trust anchor constraints enforcement was integrated with an existing public key enablement library (PKIF). Integration of support for alternative formats [3] required a number of changes to the library. These were addressed primarily through the use of abstract interfaces that captured the common elements of the various formats, i.e., all featured a subject name, a public key and extension values.

The SCVP-client mode of operation in CAPI TAG required the availability of certificates in order to use existing structures that could not be changed. For CAPI TAG purposes, trust anchors are always represented as either a certificate or a TrustAnchorInfo containing a certificate. It may have been possible to recast trust anchors stored as TBSCertificate or TrustAnchorInfo objects as Certificates with bogus signatures, but this was not explored.

Integration of trust anchor constraints enforcement [5] with the PKIF library was straightforward. Initially support was integrated as wrapper code that resided in an application, but this was moved into the library itself and exposed as an optional feature of the path validation implementation. Constraints enforcement [5] can be implemented independent of other trust anchor management specifications [2][3][4] using extensions expressed in self-signed certificates. This would be of limited utility at present given the fact that most self-signed certificates do not include constraints of any sort.

At a high level, the implementation of support for the trust anchor management specifications and integration of that support into existing products consisted of the following activities:

- Define trust anchor store format
- Define and implement trust anchor store interface and access control mechanisms
- Identify code that uses trust anchors and make adjustments to accommodate new formats, where necessary
- Implement trust anchor constraints enforcement as pre/post processing of path validation or integrate with path validation code

Following the implementation of support for trust anchor management and trust anchor constraints enforcement, deployment of the capabilities consisted of the following activities:

- Identify the applications of interest (i.e., web browsers, email clients, etc.)
- Identify the trust anchors required by these applications
- Identify entities authorized to manage trust anchor stores
- Initialize trust anchor stores to include desired trust anchors (including constraints) and trust anchor store managers
- Distribute trust anchor stores and enable trust anchor constraint enforcement capabilities
- Manage trust anchor stores using appropriate local, remote, direct or indirect means

6.2 Potential additional standardization needs

Most existing trust anchor constraints mechanisms provide a capability similar to the extended key usage extension. Unfortunately, extended key usage values included in a trust anchor are not processed during certification path validation [1]. Defining an extension and an augmentation of the standard path validation algorithm would be simple and straightforward and

potentially valuable in terms of promoting interoperability. However, the utility of this extension is not entirely clear given that most enterprises do not operate certificate authorities, let alone root certification authorities, on a per extended key usage basis.

The usage of the existing name constraints extensions in trust anchors is effective in enterprise environments where naming conventions are rigorously controlled and are generally hierarchically related. The name constraints mechanism is less suited to internet use, where distinguished names vary greatly within a single certification path and server names are often conveyed as a terminal relative distinguished name (RDN) value. Addressing this issue may be more easily accomplished by refining naming practices to enable the usage of existing name constraints mechanisms than defining alternative constraint mechanisms.

Another name constraints-related issue is the observation that to effectively use name constraints, most or all trust anchors in a given trust anchor store must have an associated name constraint value. To ensure that a particular namespace can only be issued by a given trust anchor all other trust anchors must be defined to either have an alternative permitted namespace or to exclude the namespace of interest.

Though no in-depth investigation of the utility of trust anchor management tools to counter phishing attacks was conducted, it is possible that better use of existing constraints or definition and adoption of additional constraints could provide useful countermeasures.

7. REFERENCES

[1] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, RFC 5280, May 2008.

[2] Housley, R., Wallace, C., and S. Ashmore, “Cryptographic Message Syntax (CMS) Content Constraints Extension”, in progress.

[3] Housley, R., Wallace, C., and S. Ashmore, “Trust Anchor Format”, in progress.

[4] Housley, R., Wallace, C., and S. Ashmore, “Trust Anchor Management Protocol (TAMP)”, in progress.

[5] Wallace, C. and S. Ashmore, "Using Trust Anchor Constraints During Certification Path Processing", in progress.

[6] Wallace, C. and R. Reddy, “Trust Anchor Management Requirements", in progress.

Practical and Secure Trust Anchor Management and Usage

Symposium on Identity and Trust on the Internet

April 15, 2010

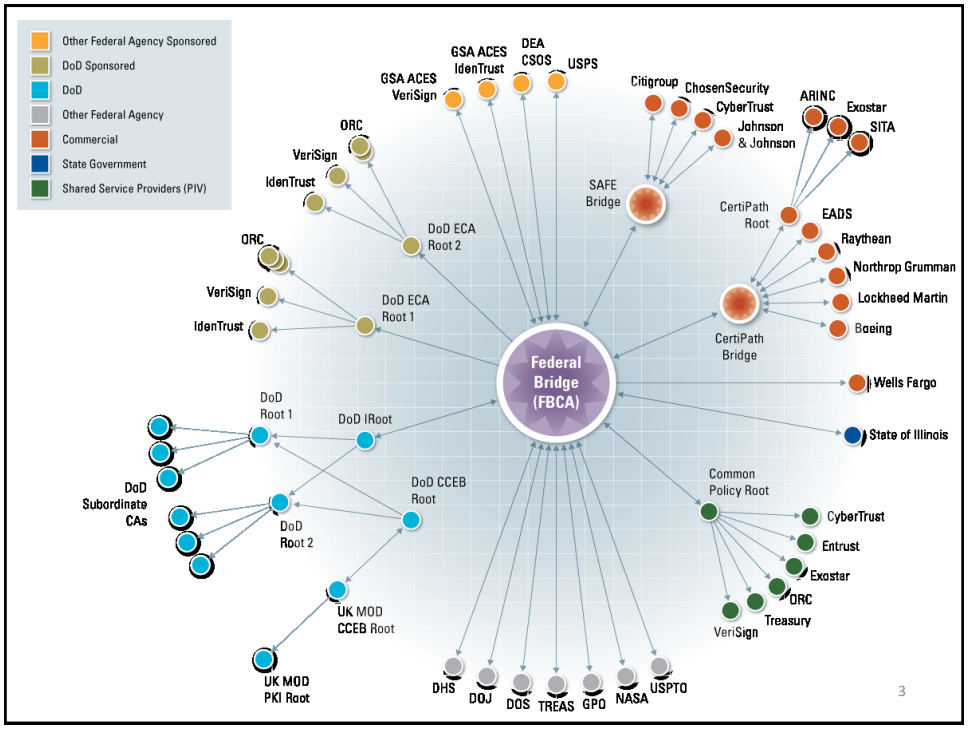
Carl Wallace
Cygnacom Solutions
cwallace@cygnacom.com

Geoff Beier
Cygnacom Solutions
gbeier@cygnacom.com

Scenario #1

- DOD is cross-certified with the Federal Bridge CA (FBCA) via the Interoperability Root CA (IRCA)
  - Enables optional recognition of the FBCA community, i.e., those who install the IRCA as a trust anchor (TA) can interoperate with the FBCA
- In some cases, entities who do not use the IRCA need to communicate with enterprises who are cross-certified with the FBCA
  - For example, Department of State (DOS)
- How can interoperability be enabled without using the IRCA?
  - Installing the DOS trust anchor results in similar (and worse) level of exposure to the FBCA community that installing the IRCA does

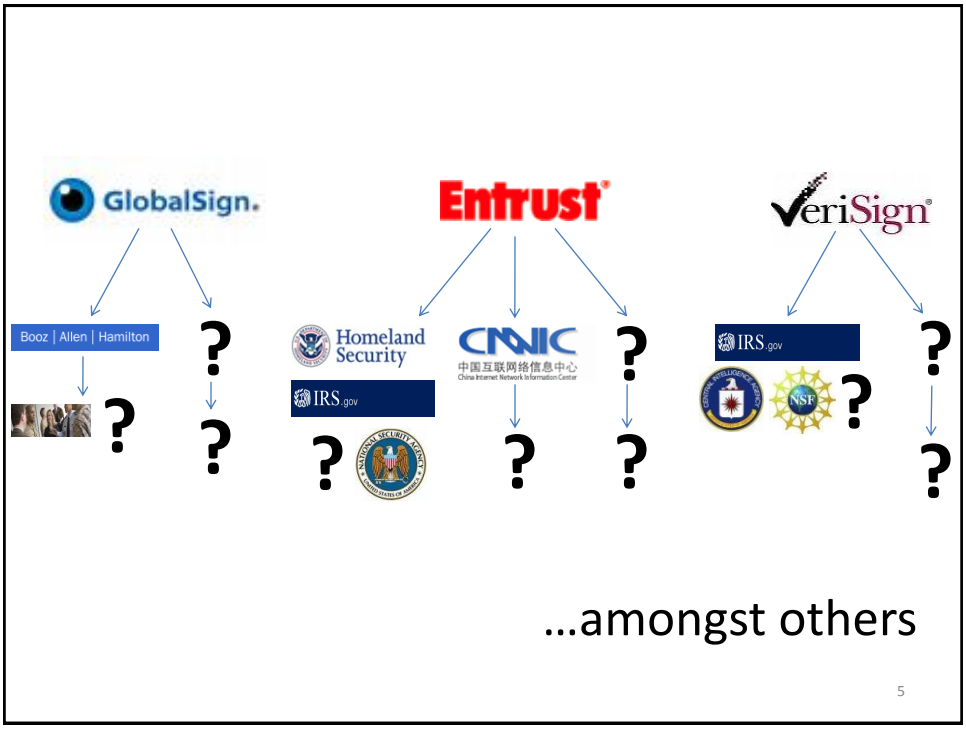
2

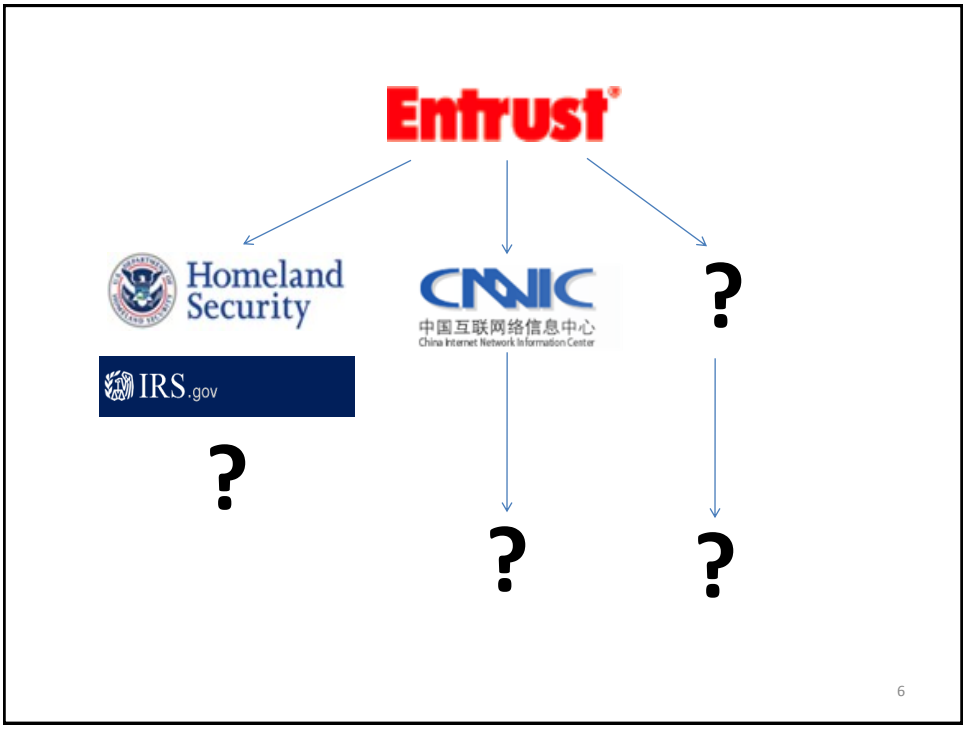


Scenario #2

- Operating systems and applications are pre-loaded with a variety of default TAs
  - Trust graph is unknown (or at least unpublished)
  - Limited options for users or administrators to add constraints
- How can an enterprise ensure CAs validated using a default TA are not issuing certificates that assert names or policies managed by the enterprise?

4


GlobalSign.
Booz | Allen | Hamilton
Entrust
Homeland Security
IRS.gov
CNNIC
中国互联网络信息中心
China Internet Network Information Center
VeriSign
IRS.gov
NSF
?
…amongst others
5


Entrust
Homeland Security
IRS.gov
CNNIC
中国互联网络信息中心
China Internet Network Information Center
?
6

Common problem

- The challenge in both scenarios stems from an inability to constrain TAs in useful ways

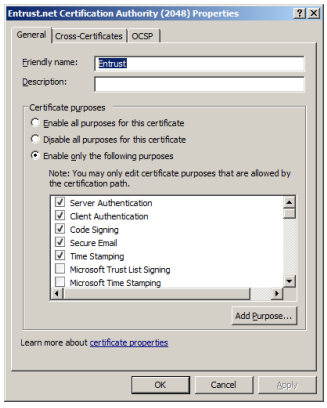
7

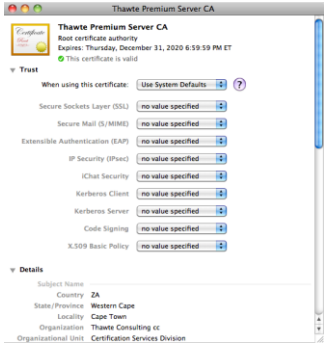
Establishing Trust Relationships Between Enterprises Using a PKI

- Direct bi-lateral cross-certification
  - **Mechanics**: Enterprise A's root CA issues a certificate to Enterprise B's root CA and vice versa
  - **Constraints**: Each certificate includes desired name constraints, policy constraints, path length constraint, policy mapping, etc.
  - **Scope**: enterprise-wide
- Indirect bi-lateral cross-certification (i.e., Bridge CA)
  - **Mechanics** : Both Enterprise A and B root CAs issue certificates to a Bridge CA and vice versa
  - **Constraints**: Each certificate includes desired name constraints, policy constraints, path length constraint, policy mapping, etc.
  - **Scope**: enterprise-wide
- Direct trust/implicit unilateral cross-certification
  - **Mechanics** : Enterprise A installs Enterprise B's root certificate as a trust anchor and vice versa
  - **Constraints**: Unconstrained or limited by the extended key usage-like constraints options supported by the trust anchor store
  - **Scope**: local

8

Current TA constraint mechanisms





9

Emerging standards

- Several specifications are progressing through the IETF to address trust anchor management and trust anchor usage issues
  - Trust Anchor Format (TAF)
    - http://tools.ietf.org/html/draft-ietf-pkix-ta-format-04
  - Trust Anchor Management Protocol (TAMP)
    - http://tools.ietf.org/html/draft-ietf-pkix-tamp-07
  - CMS Content Constraints (CCC)
    - http://tools.ietf.org/html/draft-housley-cms-content-constraints-extn-04
  - Using Trust Anchor Constraints During Certification Path Processing (UTAC)
    - http://tools.ietf.org/html/draft-wallace-using-ta-constraints-02.html
- Requirements for the specifications listed above are defined in an informational draft
  - Trust Anchor Management Requirements
    - http://tools.ietf.org/html/draft-ietf-pkix-ta-mgmt-reqs-05

10

Trust Anchor Format

- Self-signed certificates are de facto standard for representing trust anchors
  - Security requires out-of-band establishment of trust
  - Format does not lend itself to association of constraints by relying parties
- TAF defines three formats for representing a trust anchor
  - Certificate
    - Self-signed or otherwise
  - TBSCertificate
    - i.e., a Certificate structure without signature
  - TrustAnchorInfo
    - Can be as small as name and key (or name and key plus constraints)
    - Can wrap a certificate to add constraints

11

Trust Anchor Management Protocol

- Primary aim is to reduce need for out-of-band trust decisions
  - Enables trust anchor stores to be initialized once (in a secure environment) and managed thereafter using TAMP
- Provides support for disaster recovery
  - Via Apex TA with a contingency key
- Management operations are subject to strict subordination rules

12

TAMP Message Types

- Eleven TAMP message types
  - Content types, in CMS parlance
- Five pairs of request/response messages plus TAMPError

Request (TA Manager-generated)	Response (TA Store-generated)
TAMPUpdate	TAMPUpdateConfirm
TAMPApexUpdate	TAMPApexUpdateConfirm
TAMPCommunityUpdate	TAMPCommunityUpdateConfirm
SequenceNumberAdjust	SequenceNumberAdjustConfirm
TAMPStatusQuery	TAMPStatusResponse
	TAMPError

13

CMS Content Constraints

- Certificate or trust anchor extension that describes the types of content that can be validated using a given public key
  - Content is described in terms of CMS content types and CMS attributes
- Can be used as an authorization mechanism with TAMP

14

Using Trust Anchor Constraints During Certification Path Processing

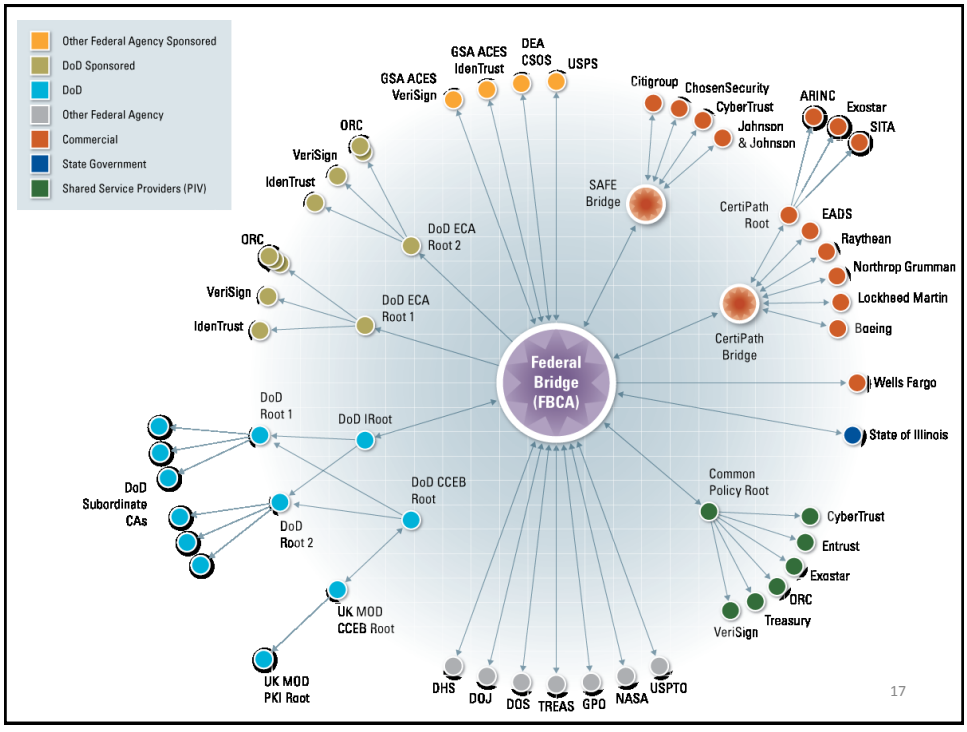
- Describes how to use constraints expressed in a trust anchor during certification path processing
  - Essentially describes how to combine values from trust anchor extensions with standard user-supplied path validation inputs
- Processing can be
  - Incorporated into an RFC 5280 compliant implementation
  - Implemented as pre-processing of RFC 5280 inputs and post-processing of RFC 5280 outputs

15

Using emerging standards to address Scenarios #1 and #2

- Wrap each desired self-signed root certificate in a TrustAnchorInfo structure and associate necessary constraints, i.e., permitted namespaces, excluded namespaces, policies, etc.
  - Uses TAF
- Install new trust anchor definitions in TA store
  - Uses TAMP and CCC
- Enforce the constraints contained in the TA during certification path processing
  - Uses UTAC

16



CAPI Trust Anchor Guard (CAPI TAG)

- Enables applications that using Microsoft CAPI for certification path processing to use trust anchor constraints
- Uses secondary trust anchor store that provides constraints for trust anchors stored in native CAPI root store
  - Managed via CAPI TAG tools using TAMP/CCC

18

Integration with Microsoft CAPI

- Several avenues were explored while searching for a means of providing support for trust anchor constraints to applications enabled using Microsoft CAPI
  - These efforts are described in IDTrust paper
  - Revocation status provider and validation policy provider interfaces were explored but not used
- The certificate store API was selected for use
  - Serves as a point of entry for intercepting calls to the native certification path processing function
    - Enables support for trust anchor constraints, delegated certification path processing (SCVP), etc.

19

CAPI TAG components

- CAPI TAG StoreCreator
  - Used to initialize a CAPI TAG trust anchor store
- Store Manager
  - Manages local CAPI TAG trust anchor stores via local file access or HTTP
  - Manages remote CAPI TAG trust anchor stores via HTTP
  - Manages remote CAPI TAG trust anchor stores via a file containing a TAMPStatusResponse message
- mod_tam
  - Routes TAMP messages received via a particular URI to a TA store file for processing
  - Periodically checks specified URIs for TAMP messages, which are downloaded and presented to a TA store file for processing
- Process TAMP Message
  - Enables a file containing a TAMP messages to be presented to a CAPI TAG trust anchor store (via HTTP or local file access) for processing

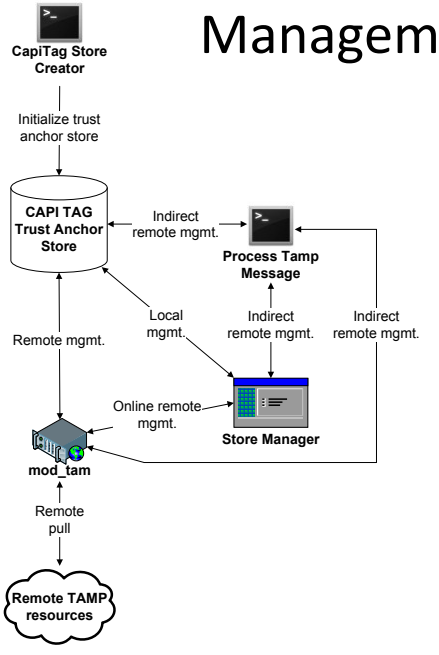
20

CAPI TAG components

- CAPI TAG
  - Integrates with Microsoft Windows to provide trust anchor constraints enforcement (or alternative certification path processing, e.g., SCVP)
- CAPI TAG Configuration Utility
  - Primary means for configuring CAPI TAG for use
- CAPI TAG Customization Wizard
  - Deployment utility used to create MST files
- PKIFTAM
  - C++ library that provides support for TAF, TAM and CCC
  - UTAC support is available in base PKIF library

21

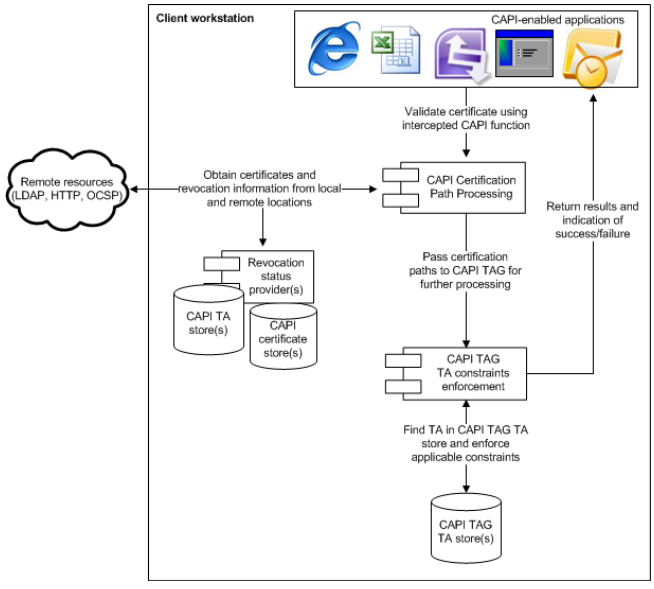
Management models



- Local management
  - File-based
- Online remote management
  - TAMP over HTTP
- Remote pull
  - TAMP messages via HTTP, RSS, LDAP, FTP, etc.
- Indirect remote management
  - TAMP over sneaker-net, email, etc.
- Additional possibilities
  - Status quo w/ RFC 5280 inputs and/or TA constraints in certificates
  - Delegation via SCVP
- Potential models
  - Interactive pull (submit response, receive update)
  - Remote pull (per SIA or AIA)

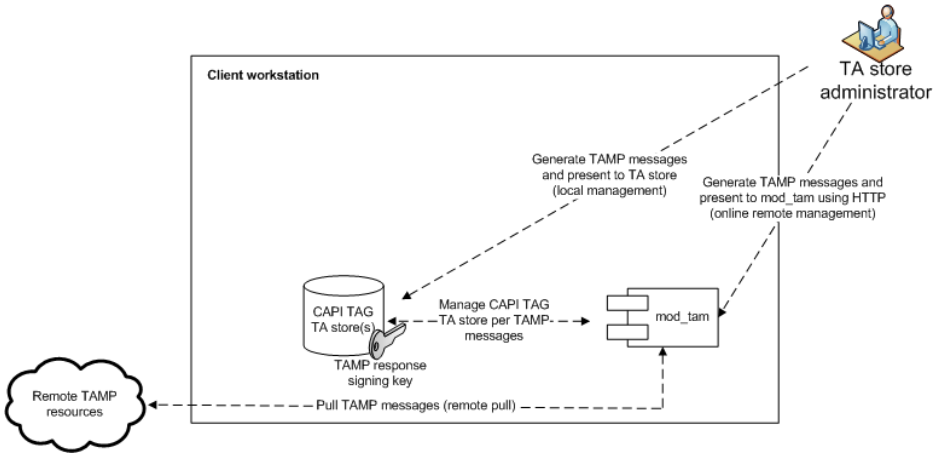
22

Process flow abstraction



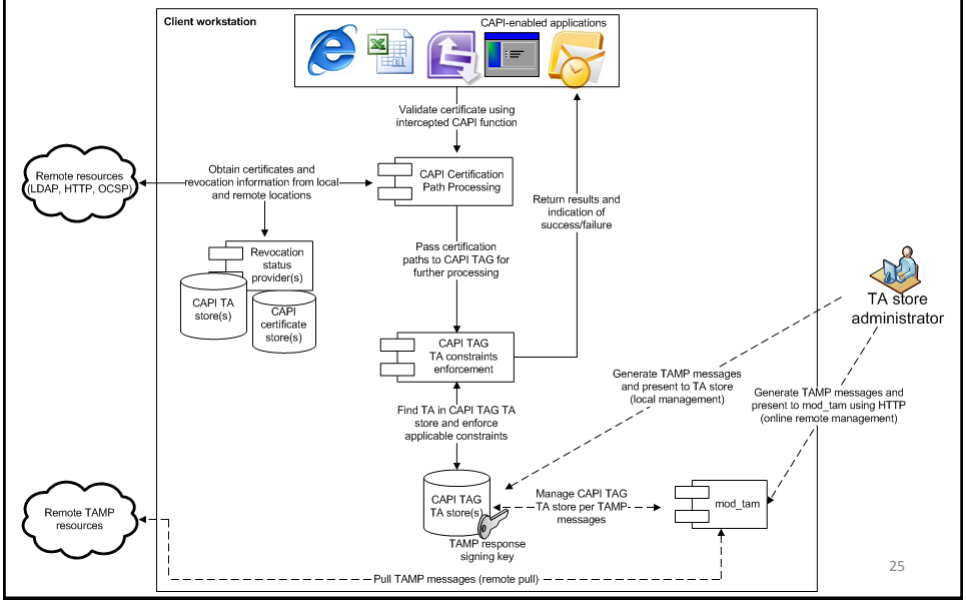
23

Process flow abstraction



24

Process flow abstraction



25

Functional Roles

- Trust anchor store manager
  - Controls private key corresponding to trust anchor or certificate with CCC extension authorizing generation of all TAMP request messages
    - TAMPUpdate, TAMPCommunityUpdate, SequenceNumberAdjust and TAMPStatusQuery messages
- Trust anchor store viewer
  - Controls private key corresponding to trust anchor or certificate with CCC extension authorizing generation of TAMPStatusQuery messages
- Trust anchor store user
  - Is not authorized to generate any TAMP messages but may use a CAPI TAG trust anchor store
  - May present a TAMP message from an authorized source to a CAPI TAG trust anchor store for processing
- System administrator
  - Authorized to edit system registry and filesystem (i.e., to install or configure CAPI TAG)

26

Demonstration

A Proposal for Collaborative Internet-scale trust infrastructures deployment: the Public Key System (PKS)

Massimiliano Pala
Department of Computer Science
Dartmouth College, Hanover, NH
pala@cs.dartmouth.edu

ABSTRACT

Public Key technology is about multiple parties across different domains making assertions that can be chained together to make trust judgments. Today, the need for more interoperable and usable trust infrastructures is urgent in order to fulfill the security needs of computer and mobile devices. Developing, deploying, and maintaining information technology that provides effective and usable solutions has yet to be achieved. In this paper, we propose a new framework for a distributed support system for trust infrastructure deployment: the Public Key System (PKS). We describe the general architecture based on *Distributed Hash Tables (DHTs)*, how it simplifies the deployment and usability of federated identities, and how existing infrastructures can be integrated into our system. This paper lays down the basis for the deployment of collaborative Internet-scale trust infrastructures.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*authentication*

General Terms

Security, Design, Standardization

Keywords

PKI, Federated Identities, Distributed Systems, Peer-to-peer

1. INTRODUCTION AND MOTIVATIONS

Public Key Infrastructures are fundamental building blocks of the Internet. We rely on *Public Key (PK)* technology for many important activities—e.g. eCommerce, email protection, and website authentication. Effective use of PK requires the relying parties to access the information and resources that enable them to verify (1) the identity of the participating entities, (2) the validity of their credentials (e.g., certificates) and (3) the context in which the credentials may be trusted. In on-line environments, relying on information that is not properly validated can lead to fraud, unauthorized access to classified data, or misuse of computing resources. Public Key cryptography offers the possibility to authenticate the identity of a remote party by verifying one's capability to use a private key associated with a known public key. Although the link between the public and the private keys can be easily established through cryptographic algorithms, the link between public key and user's identity requires an additional component: an infrastructure for identity and key management.

Unfortunately, when leaving closed and controlled environments (like proprietary OSes), the complexity and variety of real-world trust infrastructures impacts on the interoperability of trust infrastructures. Solving today's deployment issues will provide the required building block for secure communication and authentication in many environments (e.g., Trusted Computing, Computing Grids, wireless and wired network access). We identify the following as the most important issues related to the deployment of Internet-scale trust infrastructures in open environments.

Problem 1. Unlike the Domain Name System (DNS), which provides a world-wide single Internet host naming infrastructure, PK technology does not rely on a globally-authoritative infrastructure. In order to correctly use the services offered by a Certification Authority (CA), applications need to be able to "discover" them and take informed trust decisions. Regrettably, there is no support system for trust infrastructures deployment, nor a standardized protocol (besides PRQP [21] which is capable of providing the discovery properties for PKI resources) that will allow applications to easily interact with different PKIs. For example, discovering the address and supported protocol for certificate renewal from a Certification Authority (CA) is almost impossible for an application. In this paper we propose and analyze the design of a Public Key System (PKS) that allows PK-enabled applications to discover resources offered by different CAs. Trust decisions regarding a particular CA can then be facilitated by discovering which trust communities or other organizations already rely upon them (see also Problem 4 below).

Problem 2. Interaction among different parts of a PKI is often difficult. Current PKIs require applications to interact with many different services, which are provided through disparate transport protocols. Although many popular ap-

IDTrust '10, April 13–15, Gaithersburg, MD USA

plications (e.g., browsers and Mail User Agents) are capable of using Public Key Certificates (PKCs), even the simplest tasks related to the use of PK technology (e.g., requesting a certificate, renewing a certificate and/or checking the validity of a certificate) requires the application to support a variety of different protocols. *On the developers side*, the problem with PKIs is the complexity of certificate processing and the need to support a wide variety of transport protocols. For example, according to [8] the minimum set of required protocols to be supported is composed by FTP [3] and HTTP [9, 16]. Both of these protocols have been relied upon because of historical reasons (e.g., FTP) or because of their wide deployment (HTTP), the availability of many related services (e.g., HTTP Proxy, HTTP Caching services, etc.) or default access properties (e.g., traffic being allowed through firewalls by default). On top of these, most of CAs use HTTPS, LDAP [26], LDAPS to publish certificates and CRLs. All of these protocols, require the application developer to either rely on existing libraries (when these are available) or to provide her own implementation. In fact, implementing a full HTTP library that is capable of managing all the possible HTTP commands, codes, and configurations could require a lot of additional development time and costs. When it comes to small devices, the need to reduce the size of libraries and memory usage is well known. Therefore, our proposal provides a simple transport protocol for PKI messages. The protocol is easy to implement and flexible enough to support current and future needs for communications between different PKI actors. An analysis of all current PKIX protocols (eg., OCSP, CMC, CMM, etc.) showed that supporting a request—response model in PKS allows to integrate them with the proposed PKS. As described in 3.1, we developed a simple challenge—response protocol that allows for re-utilization of most of the already deployed software.

Problem 3. It is impossible for users and applications to specify the class of PK services they want to trust. The possibility of identifying a set of service providers based on a classes of services (e.g., local, eCommerce, eBanking, eMail, organizational, and Internet) will allow better trust management in applications. Since people carry many small personal devices for everyday use, they might want to exchange information directly and securely (e.g., beam it or radio it). In order for people to interact efficiently with different certification authorities for different purposes, we need to rethink today's infrastructures to allow for globally and locally available trust infrastructure networks. In fact, real-world trust infrastructures demand a simple and interoperable way to federate identities. Today, many PKIs are in place to serve a specific purpose. The deployment of PKIs for providing identities to access resources within federations (e.g., computing grid policy bodies like TAGPMA [25] or IGTF [13]) is an example of such specializations. Another example is the presence of many CAs in the commercial sector dedicated to provide only SSL certificates. We should introduce a mechanism to support contextual trust. For example, when setting up a mobile device to access the home network, it should be easy to discover and utilize *local* PK services; however, when it comes to accessing services on the Internet, we might want to validate certificates/services by using trust anchors associated with specific bodies: government services, Internet services, and on-line banking services. Given the possibility to easily discover PK services, our PKS provides the ability to group them according to specific environments to help users to manage (or delegate) trust settings.

Problem 4. The lack of a standardized method to identify the federation that a trust anchor is a part of impacts on the capability of users to select the context in which the trust anchor should be used. For example, when using a browser to interact with a Federal Agency website, the user is unable to trust only a subset of the trust anchors present in the application's (or Operating System's) certificate store (e.g., certificates that are part of the Federal Bridge PKI), instead all trust anchors are treated equally. The user should be provided with the possibility of trusting a specific set of trust anchors by using the familiar concept of *federation* instead of Policy Identifiers embedded in the digital certificates. By facilitating a method for disseminating information about which organizations or federation use/include/trust a particular trust anchor, our system allows for easy deployment of federated identities.

In this paper we present a support system for trust infrastructures based on Distributed Hash Tables (DHTs) that is suitable for Internet-scale deployment and provides dynamic federation management. Moreover, our work can be easily integrated with existing infrastructures allowing for a smooth roll over between isolated PKI islands to globally available and locally configurable PKI services.

The rest of the paper is organized as follows. Section 2 presents the background and related work. Section 3 describes the basic principles of PKS, the overlay network design and the message format. How to deploy federated identities within PKS is explained in section 4, while Section 5 details how to integrate existing PKIs with our infrastructure. Section 6 contains our conclusions and future work.

2. RELATED WORK

An important part of PKIs is the "I"–nfrastructure that is needed to manage the trust relationships between entities.

We investigated existing trust infrastructure deployment systems and collaborative approaches to provide federated identities. In this section, we provide a description of the previous work and related technologies.

2.1 "I" for Infrastructure

Throughout the years, research has offered many different technologies like PGP [6], SDSI-SPKI [7] and identity-based encryption (IBE) [4] to authenticate users. Although each of them have their own strenghts and weaknesses, an *infrastructure* of some sort is needed in order to provide support for trust building. An example of a widely deployed infrastructure is represented by the *web of trust* used in Pretty Good Privacy (PGP) [6]. Similar to traditional X.509 PKIs, PGP uses signed statements (certificates) to establish the link between a public key and a user's identity. PGP identities are unique in that normal users can endorse them by digitally signing other users' keys. Although this approach may work in small and well-defined communities where out-of-band (e.g., face to face) identity verification is feasible, its decentralized authentication scheme would not work for large-scale, widely distributed deployments (e.g., for the In-

ternet community), automated infrastructure environments (e.g., Trusted Computing), or in high-security environments (e.g., Federal Agencies).

In X.509 infrastructures, well-defined liabilities and certificate policies have been defined to provide the flexibility and the scalability required by Internet-scale trust infrastructures. Researchers and standardization bodies, working at both local and global scales, have defined a set of minimal requirements (or profiles) that can be used as guidelines when deploying these authentication infrastructures [8, 11]. As a consequence, X.509 PKIs now provide the most widely deployed technology for Internet authentication (e.g., WAN and Interdomain). Regardless of the fact that today identity providers need to participate in federations, no standardized infrastructure exists to support federated identities and to help applications and users to correctly manage their trust settings.

To address the need for a trust infrastructure for the Internet, early approaches envisioned the establishment of an Internet Policy Registration Authority [15]. Its failure due to political, rather than technical, issues showed the impossibility to centrally manage Internet-wide trust infrastructures. The absence of a globally-available infrastructure (like the DNS in the case for Internet host naming) led to the establishment of many different and poorly interoperable trust infrastructures. Researchers and Internet working groups have tried to address this problem by studying more distributed trust models that use cross-certification and/or bridge CAs. Unfortunately, the difficulties related to path validation in these more complex trust infrastructures have slowed down their adoption in the real world. Moreover, the need to accept a common certification policy is an obstacle to their deployment in open environment.

Today, the need to provide services beyond the borders of a single organization demands for more interoperable environments. Government Agencies, Grid Computing Communities and Trust Computing environments provide a clear example of organizations (or Virtual Organizations in some cases) where the need for a globally-available trust infrastructure is compelling.

The lack of a standardized method to provide federated identities has pushed many communities to provide solutions based on weak credentials (like passwords). For example, the Federal Government CIO Council established the Identity, Credential, and Access Management (ICAM) Subcommittee i.e. ICAMSC, with the charter to foster effective ICAM policies and enable trust across organizational, operational, physical, and network boundaries [12]. The release of an ICAM Trust Framework (TF) has led to several key federations seeking accreditation with the ICAM TF (e.g. InCommon Federation [14], OpenID Federation [20]). However, these federations are primarily focused on the lower levels of authentication as defined in NIST Special Publication 800-63 [5] (i.e. levels 1 and 2) which do not require strong credentials. Hence, they do not have the identity binding necessary for providing high level of assurance (LoA) credentials.

To avoid political issues that led to the failure of IPRA, the globally-available infrastructure should be designed in such a way that would (a) provide support for *federated identities* under well defined authorities, (b) define a deployment framework that helps *infrastructure management*, and (c) facilitates *trust decisions* for the user.

2.2 Peer-to-peer systems

In this paper we introduce a novel approach to deliver a cooperative system to enable interoperable trust infrastructures deployment at the Internet scale based on peer-to-peer technologies. Since we envision a Peer-to-peer approach in the PKS design, we provide a summary of current Peer-to-peer technologies relevant to our work.

In the first-generation P2P systems (e.g., Gnutella, Kazaa, Napster, etc.) all nodes are both clients and server: any node can provide and consume data. Some of these systems, like Napster, implemented a centralized search service where a single server keeps track of the location of the shared data. On the opposite side is Gnutella; in this type of network, search is implemented by recursively asking the neighbors for files of interest. The search goes on till a *Time To Live (TTL)* limit is reached. Systems like Kazaa or Skype use a hybrid model where super-peers act as local search hubs. Super-nodes are automatically chosen by the system based on their capacities in terms of storage, bandwidth and availability.

Because of the introduction of Distributed Hash Tables (DHTs), the *second generation of P2P overlay networks* provides major advantages over the first generation by implementing a predictable (maximum) number of hops needed to answer a query. DHTs are a distributed version of a hash table data structure. The combination of $(key, value)$ is used to look-up,retrieve, store, and delete shared data across peers. The key idea behind the usage of DHTs is to provide each peer with a unique identifier and assign a sub-set of the general $(key, space)$ to it. There are several routing protocols based on DHTs often referred to as P2P routing substrates or P2P overlay networks. The first usable approach of a DHT-based routing substrate is found in Chord [24] where a circular address space is used to map nodes and the key space.

Several P2P routing substrates followed after Chord. These systems introduced more sophisticated (and sometimes quite complex) design to minimize the maximum number of hops and the overhead introduced by the P2P routing infrastructure. For example Pastry [1] considers the network locality when routing messages through its network. In Pastry, in addition to the leaf nodes a neighborhood list is maintained where the M closest peers, in terms of the routing metrics, are listed. Although it is not used directly in the routing algorithm, the neighborhood list is used for maintaining locality principals in the routing table. A more complex topology is implemented in Content Addressable Network (CAN) [23]. It uses a "d-dimensional" cartesian coordinate space mapped on a d-torus. In CAN, a node is responsible for a specific value if the corresponding key hashes in the sub-space "owned" by the node itself.

Other examples of advanced DHT-based overlay networks are Tapestry [27], Kademila [18], and P-Grid [2]. Tapestry uses two identifiers: the NodeID and the Application specific endpoints or GUID. The main focus of Tapestry is efficiency. In particular, it minimizes message latency by constructing locally optimal routing tables from initialization and by maintaining them in order to reduce routing stretch. Simi-

lar to Tapestry, Kademlia algorithm uses a special notion of locality based on the calculation of the "distance" between two nodes. This distance is computed as the `Exclusive Or` of the two node IDs. Kademila uses the `Exclusive Or` because it shares some properties with the geometric distance formula: the distance between a node and itself is zero, it is symmetric, and it supports the triangle inequality. Kademlia routing tables consist of a list for each bit of the node id: nodes that can go in the n^{th} list must have a differing n^{th} bit from the node's own id. Node look-ups proceed by querying to the k nodes in its own k-buckets that are the closest ones to the desired key. These nodes will send back the k closes entries they know. The iterations continue until no nodes are returned that are closer than the best previous results. Different from any of the previously discussed protocols, P-Grid uses a bit-level approach to provide efficient node look-ups by resolving queries based on prefix matching. Instead of using a DHT, P-Grid uses a *t*rie [10], or prefix tree, which is an ordered tree data structure. P-Grid partitions the key-space in a granularity adaptive to the load at that part of the key-space. Unlike DHTs that perform efficiently only for uniform load-distributions, an overlay network based on P-Grid presents peers with similar storage load even for non-uniform load distributions.

3. THE PUBLIC KEY SYSTEM (PKS)

The PKI System (PKS) we propose in this paper is composed of three main components: the DHT-based overlay network design and routing properties, the message format, and the support for federated identities.

The PKS uses a Peer-to-peer overlay network to route messages to the target CAs and federation authorities. In particular, we use a simplified version of the Chord protocol based on the PEACH [22] system. All of the different types of overlay networks discussed in the previous section provide a large number of options (e.g., storing keys and values, retrieving values, and providing support for multicast traffic). We selected the PEACH routing algorithm for two reasons. First, it already provides support for node identifiers based on public key certificates. Second, the PEACH protocol is easy to support from the developers point of view: other protocols like Kademilia or P-Grid might provide additional features that are not required by our system. In particular, it does not support many of the operations traditionally implemented over peer-to-peer networks (i.e., *get*(), *put*(), *delete*()).

Ultimately, the PKS could use any of the peer-to-peer overlay networks discussed in Section 2 provided that changes to support identity-based node identifiers are in place.

3.1 The PKS Network

In our previous work, we designed and prototyped a scalable system for PKI resources look-up. In [22], we introduced a new peer-to-peer overlay network that makes use of a Distributed Hash Table routing protocol (namely, *Peach*). Results from this work have demonstrated that PKIs can make effective use of peer-to-peer technologies and have laid the path for the next steps in this new field. In this paper we build on our previous work and extend this approach to provide a support system for Public Key trust infrastructures deployment. In particular, we enhance the peer-to-peer protocol to support (1) interoperable PKI message exchange among CAs, and (2) usable federated identities deployment.

Similar to PEACH, we leverage the possibility to *join*() the network by using multiple identity-based node identifiers. Different from our previous work, we support two different type of nodes: the PKS responders and the PK Federation Authorities.

The PKS responders act as a PKI proxy for applications. They are capable of (a) answering clients about PKI requests as described in Section 3.2, and (b) forward PKI requests on the PKS and send back responses to the client application. The PK Federation Authorities, instead, provide information about the deployed federations by indicating if a particular entity is part of the authorized federation.

In order to locate available CAs efficiently on the PKS network, we use unique node identifiers for each CA. We leverage the availability of the CAs' digital certificates by deriving the node's identifier from the fingerprint of the CA certificate itself. For example, if CA_1 wants to participate in the PKS network, it will setup a PKS node and issue a certificate that identifies it as the authoritative PKS responder. When joining the PKS network, the PKI gateway will present its own certificate together with the CA_1's certificate. The node identifier, that is the identifier that will enable the node to be found on the network, is calculated by using the fingerprint of the CA_1's certificate. To validate the identity of the joining node, a simple validation of the presented certificate chain will guarantee that the joining node has been authorized as a PKS responder for that particular CA. Let n be the PKS responder for CA_1, the trust chain:

$$Certificate(n) \leftarrow Certificate(CA_1)$$

guarantees the authorization of the node to respond as the PKS responder (a specific extension in the PKS responder's certificate might be required). Moreover, because the node identifier is the hash of the CA's certificate, it enables the PKS responder only for that particular CA. This approach guarantees high scalability and provides a simple approach to PKS responders deployment.

It is important to notice that the PKS network can support *any type of public key identifiers*. This feature stems from the use of the output of the hash function to link a node on the PKS network to an identity (e.g., a CA or a PK-FA). Although our work primarily focuses on X.509 certificates, PKS is capable of supporting multiple type of public key based identifiers.

Applications such as browsers or email clients, access the PKS by querying the local PKS server. By looking at the target responder in the PKS network, the local PKS responder discovers if a responder for the target CA is available and, if so, forwards the application's request to the target node. The response is then routed back to the client. As described in Section 3.2, applications use only one simple transport protocol for all PKI-related queries (e.g., OCSP, CMM, SCEP, etc.) and do not need to implement any of the overlay network operations (e.g., *join*() or *lookup*()). If a local PKS responder is not available, one of the pre-configured servers can be used instead (same approach as in DNS where applications and operating systems are provided with the list of root DNSs). We envisage that local PKS responders (or PKI gateways)—as in the case of caching servers for DNS—

will regularly be deployed in LANs to facilitate access to PKS for applications.

Although we provide an overview of all of the main features of the PKS network, because many operations are similar to the ones described in PEACH we refer to our previous work for a more exhaustive description of the protocol and its performances.

The PKS Local Routing Table

Our system uses a DHT table together with a hash function (i.e., SHA-256) to implement efficient routing in PKS. Each participating node is provided with an identifier that is derived by calculating the hash of the responder's CA certificate or, in case of a Federation Authority, the authority's certificate. To support efficient nodes lookup, each node stores a *local routing table*. Each table carries m entries where m is equal to the number of bits in the node identifiers, that is the size of the output of the selected hash function. As we use the same algorithms as identified in [17] to build and update the local routing table in PKS, we do not report the full description here. However, we describe the basic structure of the routing table to provide a clear view of the network properties. Moreover, as the routing algorithm is derived from Chord, all formal proofs still hold for the PKS.

The local routing table correlates the nodeIDs to the node's network address. To optimize *lookup* operations, the routing table is kept ordered by nodeID. Let id_n be the node identifier for node n, and m be the size (in bits) of the node identifiers, then the stored values of the local routing table range from:

$$x_0^n \quad = \quad (id_n + 2^0) \mod 2^m$$

to:

$$x_m^n \quad = \quad (id_n + 2^{m-1}) \mod 2^m$$

In general, the value for the i—th entry in the local routing table can be expressed as:

$$\forall i \in [1, 2, \ \ldots \ , m], \quad \Rightarrow \quad x_i^n \quad = \quad (id_n + 2^i) \mod 2^m$$

therefore, the node-identifier space related to the i—th entry is:

$$\gamma_i^n \quad = \quad [x_i^n, x_{i+1}^n)$$

Let k be the target node for a query. By looking at the local routing table, the node n can find the closest node whose identifier *is equal* or *precedes* k. By iterating this approach it is possible to find the requested node in $O(\log(m))$ operations.

Multiple Identifiers

A PKS responder might need to be identified on the PKS network by multiple node—IDs. This happens, for example, when the PKS responder is authoritative for multiple CAs. Moreover, the same node can serve as an authority for one or more federations at once (see section 4).

To be assigned multiple node identifiers, the joining PKS responder performs multiple $join()$ operations on the network. Let n be the number of certificates the PKS responder possesses. The set of the CA certificates related to the PKS responder (ϕ) can be expressed as:

$$\phi \quad = \quad \{x_1, x_2, \ldots, x_n\}$$

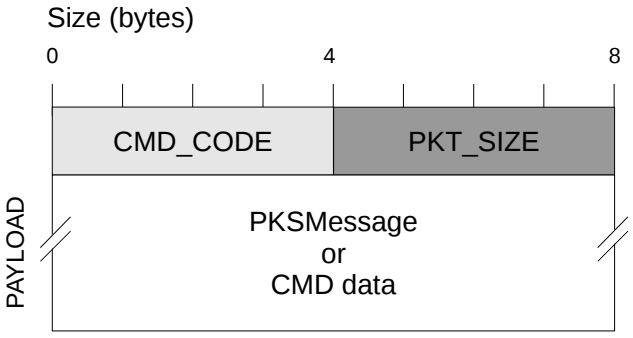


Figure 1: Structure of PKS messages.

If the PKS responder also acts as one or multiple federated authorities, the set of certificates associated with this role can be expressed as:

$$\eta \quad = \quad \{x'_1, x'_2, \ldots, x'_m\}$$

Let θ be the set of network identifiers related to the joining PKS responder:

$$\theta = \quad \{y_1, y_2, \ldots, y_n\}$$

and let ψ be the set of network identifiers related to the federation authority role:

$$\psi = \quad \{y'_1, y'_2, \ldots, y'_m\}$$

where:

$$\forall i \in [1, 2, \ \ldots \ , n], \qquad \exists x_i, y_i \ : \ x_i \in \phi \ \ \vee \ \ y_i \in \theta \\ \Rightarrow y_i = H(x_i)$$

and:

$$\forall k \in [1, 2, \ \ldots \ , m], \qquad \exists x'_k, y'_k \ : \ x'_k \in \eta \ \ \vee \ \ y'_k \in \psi \\ \Rightarrow y'_k = H(x'_i)$$

For each x_i the responder is authoritative for, the PKS responder has a different network identifier y_i, which is based on the CA's certificate fingerprint. For each x'_k the responder is the federation authority for, the PKS responder has a different network identifier y'_k, which is based on the federation's authority certificate (not on the federation authority's certificate issuer). For each of these identifiers, the joining peer performs $findnode()$ to find its successor in the network ring and proceeds to register itself in the right position. This approach enables the responder to provide PKS services for different CAs and federation authorities and potentially facilitate the deployment of existing CAs in the PKS network.

3.2 The PKS Message Format

The simplicity of the PKS message format constitutes one of the core features of our system. To minimize the impact of the message format and support the integration of existing PKIX protocols, we opted for a simple binary format. In PKS, each message is composed by a header and a body. The header carries two integers (`uint32`) that indicate the type of the message (command code) and the size of the body (message length) in bytes. The payload of the message is a DER encoding of a `PKSMessage` that acts as a wrapper

around the PKIX message (e.g., CMS) to be dispatched to the target node.

The *command code* is 4 bytes long and it specifies the action to be performed on the target node or the return code. The *packet length* is used to identify the length of the payload and is also 4 bytes long (type uint32_t). When the control code identifies a network-related operation (e.g., *lookup*(), *join*(), or *leave*()), the payload carries the control data. For example, when a lookup is requested, the payload carries the node identifier of the searched PKS responder (the certificate's hash).

A special case is represented by the `CMD_PKI_MESSAGE` command code. In this case, instead of the `CMD Data`, the payload content is a `PKSMessage`. The PKS message structure is depicted in Figure 1. The PKS command codes are reported in Table 1.

The `PKSMessage` is defined as follows (ASN.1 notation):

```
PKSMessage ::= {
    protocol    OBJECT IDENTIFIER,
         --- Identifier for the data protocol
    targetNode    OCTET STRING,
         --- Target Node Identifier (hash)
    rawBytes      OCTET STRING
         --- Binary data (e.g., CMS message)
}
```

The PKSMessage is composed of three fields: protocol, targetNode, and rawBytes. The `protocol` field carries the object identifier for the data format used. For example, if the body of the message carries an OCSP response, the `protocol` will carry the `id-ad-ocsp` object identifier. The `targetNode` bears the node identifier of the target node. This helps the receiving node to correctly process the request in case the node is assigned with multiple nodeIDs. Last but not least, the `rawBytes` field encapsulates the contents of the original PKI message in DER format. The chosen approach simplifies the routing of PKI protocol messages in PKS without requiring any change in the published standards. Moreover, the `rawBytes` can encapsulate and form of data, thus providing support for future PKIX (and non-PKIX) protocols.

4. FEDERATION AUTHORITIES

Along with PKS responders, we introduce special kind of nodes, namely *PKS Federation Authorities (PK-FA)*. These special nodes serve as responders for determining if a particular entity is part of a specific set, also called Federation. PK-FA nodes use identifiers similar to the PKS responders' ones. However, different from the latter, the Federation Authority identifiers are calculated by using the fingerprint of the PK-FA certificate instead of its issuer's one. For instance, when a node k joins the PKS network as a responder for CA_1, its assigned nodeID is:

$$id_k = hash(x_{CA_1})$$

where x_{CA_1} is the certificate of CA_1. Instead, when a node j joins the PKS network as a federation authority, the assigned node identifier is:

$$id_j = hash(x_j)$$

where x_j is the certificate of the responder (j) itself. This approach relieves federation authorities from deploying ad-hoc certification authorities (as in the case of bridge CAs), and allows them to use end-entity certificates provided by any third-party CA. By trusting the PK-FA to be authoritative for a specific federation, users and applications are be able to query for the participation of an entity in a specific federation.

As an example, let's consider the impact of PK-FA nodes on browsers' trust store. Today, CAs undergo specific audits and certification processes to be included in browsers and operating systems. By leveraging the PKS features, the number of certificates embedded into applications could drop substantially. In fact, let company χ be a certification/auditing provider and x its PK-FA node in PKS. If the CA has positively passed the certification process, the node x will report that CA as being part of its federation (CAs certified by company χ). By embedding the certificate of node x in the trust store, the application can verify that the certificate presented by a third party has been issued by a CA certified by the company χ. The CA certificate does not need to be embedded as a trust anchor in the application's store. This approach would hold, for example, to verify extended validation certificates (the PK-FA node could be maintained by the CA/Browser forum authority).

The introduced federation authority nodes allow for a dynamic approach to trust anchor management, smaller trust stores size, and the possibility for policy management bodies and virtual organization to be easily deployed and supported into applications.

Moreover, applications can leverage the presence of federation support built into PKS and provide more usable interfaces to the user. In fact, users could be provided with the possibility to choose which (set of) PK-FA is to be trusted for a particular session. For example, when shopping online a user could enable α and β credit card federation authorities only, thus providing the application with a *trust context* based on the familiar concept of federation/organization. On top of knowing that a merchant's website is responding to a verified URL, the user can discover if her credit card company has an established trust relationship with the merchant. It is worth noticing that queries to a PK-FA can be related to CAs or to End Entities (e.g., a website's certificate or even a user's certificate). If the appropriate authoritative PK-FA is deployed, the system can provide answers to queries like "Is this user's certificate part of the help desk (federation) of organization's γ ?", "Is this CA part of TAGPMA ?", or "Is this CA part of the US Higher Educational Authority?"

4.1 Federation Authority Queries

An important feature of the PKS is the possibility to easily federate identities under well-defined federation authorities. PK-FA nodes provide authoritative answers to the question "Is this entity part of your federation ?".

In particular, when an application wants to know if a certification authority is part of a federation, it routes a PKS message with the `CMD_LOOKUP_FEDERATION` code. The payload of the message is a `PKIAuthRequest`.

The data structure of the the federation lookup command is as follows:

```
PKSAuthRequest ::= {
   targetAuthority    OCTET STRING,
        --- Target Authority Identifier (hash)
```

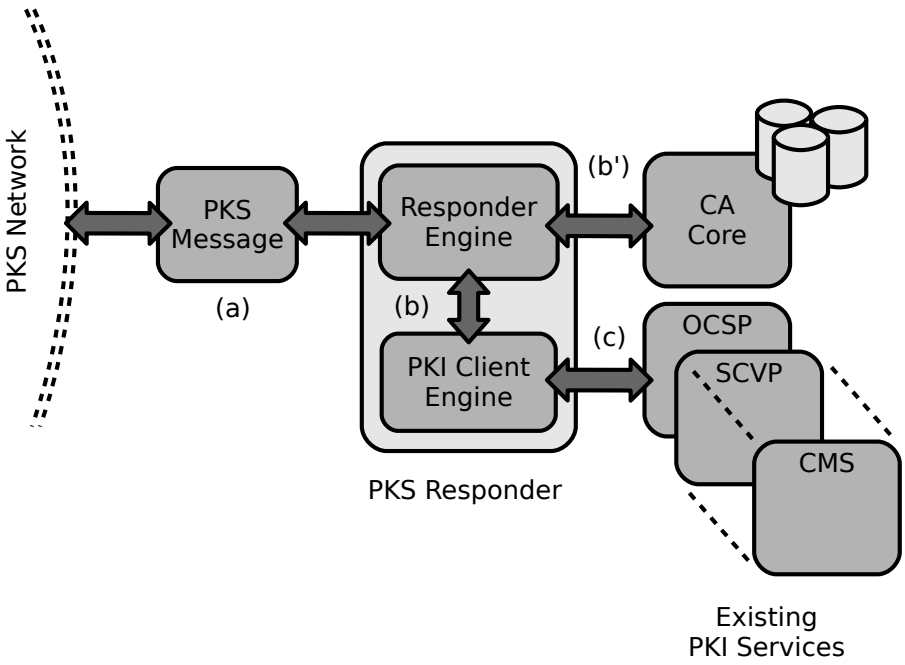


Figure 2: Integration of current PKI services. The PKS responder can act as a PKS/PKI communication gateway.

```
    targetEntity       CertIdentifier
         --- Certificate Identifier (hash)
}
```

To populate the fields of the `PKIAuthRequest` structure, the application derives the target nodeID from the federation authority's certificate (`targetAuthority` field). Then, it calculates the entity's certificate identifier (`targetEntity` field). The message is then routed to the target PK-FA node through the PKS network.

When a `CMD_LOOKUP_FEDERATION` message is received by a PK-FA node, it responds with a `CMD_SUCCESS` in case the target entity is part of its federation, otherwise a `CMD_ERROR` followed by the appropriate error code is returned.

We notice that the possibility to support federated identities in PKS provide a technical mean to provide contextual trust in applications, thus demanding for well defined policies.

4.2 Classes of Federation Authorities

In order to provide a more flexible authority management, we use different classes of PK-FA such as *Local*, *Internet*, *Network*, *Organizational*, and *Application*. Different classes have different characteristics. For example, authorities can freely join the PKS network in the *Organizational* class, allowing for private organizations to easily deploy their own federation authorities. Other PKS classes might require tighten control over who can join the PKS. For example, the set of participants in the *Internet* class—which will comprise authorities that secure the Internet infrastructure (e.g., S-BGB [19])—can be constrained depending on well identified properties. To support classes of federations, we introduce a special type of PK-FA nodes, namely Class Federation Authorities. This special set of PK-FA nodes are used to support hierarchical authorities deployment.

These authorities use the same message protocol as PK-FA nodes. The most noticeable difference is the usage of the hash of the Public Key associated with the Class Federation Authority instead of the hash of its certificate. This choice is based on the consideration that these types of nodes can be deployed (but further work is needed in this area) to provide a distributed support system for secure DNS.

In order to discover if a federation authority is part of a specific class of federations, the application sends a PKSMessage with a `CMD_LOOKUP_FEDERATION` command code. By using the hash of the public key associated with the authoritative node for the requested class (e.g., Internet or Trusted Computing classes), the application is capable of recognizing which class the PK-FA is part of.

This type of system could be used to deploy trusted keys for primary DNS domains (e.g., ".", ".net", ".edu"). We envisage that well identified authorities like ICANN will run the class federation authorities.

5. PKS DEPLOYMENT

PKS provides an overlay network that supports the deployment of trust infrastructures at the Internet scale.

An important feature of PKS is the possibility to integrate existing PKI services. Figure 2 depicts the design of a PKS responder that allows for integration of existing infrastructures in PKS. In particular, to deploy the services offered by a CA, a PKS responder can act as a bridge between the PKS and the deployed PKI services. The control flow is as follows:

(a) The Responder Engine subsystem is responsible for providing PKS network services. Besides the overlay network operations, the Responder Engine is in charge

Table 1: PKS opt codes values and description.

Command Name	Code	Description
CMD_ERROR	0x200 + 0	General Error
CMD_SUCCESS	0x200 + 1	Cmd Successful
CMD_GET_NODE_INFO	0x500 + 0	Get node information
CMD_GET_NODE_SUCCESSOR	0x500 + 1	Get node successor
CMD_GET_NODE_PREDECESSOR	0x500 + 2	Get node predecessor
CMD_UPDATE_PREDECESSOR	0x600 + 1	Update predecessor info
CMD_UPDATE_SUCCESSOR	0x600 + 2	Update successor info
CMD_LOOKUP_NODE	0x600 + 2	Perform a lookup
CMD_LOOKUP_FEDERATION	0x600 + 2	Federation participation
CMD_PKI_MESSAGE	0x800 + 0	PKI Data Packet

of processing PKS messages. In particular, when a message is received via the PKS network, the Responder Engine unwraps the PKI message embedded in the `rawBytes` field. On the other hand, when a response is ready to be sent over the PKS Network, the PKS responder builds up the PKSMessage by including the generated PKI response (e.g., the OCSP or SCVP response) in the message and routes it to the requesting PKS node. In case the service is not available from the CA, an error message is returned instead.

(b) If no integration with the CA core component is possible, the Responder Engine passes the contents of the `rawBytes` on to the PKI Client Engine which is in charge of the communication between the PKS responder and the provided PKI services. The response received from the PKI client engine is then returned to the requesting PKS node.

(b') If the CA provides some sort of integration with the core service (e.g., via a plug-in infrastructure), the PKS responder can leverage this tight integration with the CA software in order to efficiently build the PKS response. In this case, the CA core service must provide APIs capable of parsing the PKI request, accessing the data needed to build the response, and building the PKI response. The development costs of providing such an interface can be justified by the the faster response times and easier application debugging as the interaction with the CA core component is not achieved via a client/server approach (as in (c)).

(c) In case PKI services are available only though standard protocols (e.g., HTTP), the PKI client sends the PKI request (extracted from the PKS message) via normal network communication. If a valid response is returned, it is sent back to the Responder Engine. An error message is returned in case the requested service is non responsive, unknown, or not available.

Where the integration with the CA's infrastructure is not possible, and the path:

$$b \rightarrow c$$

is used to generate the PKS response, a communication overhead is introduced that can negatively impact the response time of the PKS responder.

We envisage the deployment of Internet PKS to happen in three phases. In particular, we think that initial participation in PKS will be driven by policy bodies and their communities (Phase I). For example, computing grids communities have already expressed interest in our work. These communities can freely deploy their own Federation Authorities. After an experimental deployment, we plan to work closely with Internet communities (Phase II) to identify and deploy the root Class Federation Authorities. As the success of the PKS will depend on the availability of software to support it, we plan to working closely with certification authorities, software vendors, and certificate service providers to stimulate the adoption of PKS on a large scale (Phase III).

6. CONCLUSIONS AND FUTURE WORK

The need for an homogeneous PKI System capable of addressing current problems in trust infrastructure deployments is evident. This work outlines major problems related to current approaches and lists the limitations that come from the lack of a support system for public key infrastructures.

Our future work will be focused in three different areas. First, we will build a *PKS protocol simulator* to evaluate the performance of the PKS routing protocol. This will help us to measure routing overhead in the PKS network and serve as a validation tool for the correctness of the developed algorithm. The simulation tracks will provide valuable information about the developed model and an overview of the scalability properties of PKS infrastructure. Secondly, after setting up the test-bed environment, we will build and deploy a PKS prototype in collaboration with our peers and domain experts (e.g., members of organizations like IGTF, TAGPMA, TACAR) to keep our work tied to real-world requirements and constraints. Finally, we will promote PKS within IRTF and IETF working groups by writing a PKS Internet Draft (I-Ds) and encouraging PKIX and PKNG participants to provide feedback on our proposed system.

Ultimately, our proposal provides initial steps toward an Internet-scale trust system that will enable new opportunities for research in federated identities deployment, trust infrastructure deployment, and usability of digital identities.

7. REFERENCES

[1] Pastry.

[2] K. Aberer, P. Cudrï£¡-Mauroux, A. Datta, Z. Despotovic, M. Hauswirth, M. Punceva, and R. Schmidt. P-Grid: A Self-organizing Structured P2P System. *SIGMOD Record*, 32(3), September 2003. `http://lsirpeople.epfl.ch/rschmidt/papers/Aberer03P-GridSelfOrganizing.pdf`.

[3] A. K. Bhushan. File transfer protocol, 1971.

[4] D. Boneh and M. Franklin. Identity Based Encryption from the Weil Pairing. *SIAM Journal of Computing*, 32(3):586–615, 2003.

[5] W. E. Burr, D. F. Dodson, and W. T. Polk. Electronic authentication guideline. OnLine.

[6] J. Callas, L. Donnerhacke, H. Finney, and D. Shaw. OpenPGP Message Format. Internet Engineering Task Force: RFC-4880, November 2007.

[7] D. Clark, J. Elien, C. Ellison, M. Fredette, A. Morcos, and R. Rivest. Certificate Chain Discovery in SPKI/SDSI. *Journal of Computer Security*, 9(4):285–322, 2001.

[8] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, May 2008.

[9] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1, 1999.

[10] E. Fredkin. Trie memory. *Commun. ACM*, 3(9):490–499, 1960.

[11] R. Housley, W. Polk, W. Ford, and D. Solo. Certificate and Certificate Revocation List (CRL) Profile. Internet Engineering Task Force: RFC 3280, 2002.

[12] ICAM. Identity, credential, and access management. OnLine.

[13] IGTF. The International Grid Trust Federation. OnLine.

[14] InCommon. InCommon Federation Homepage. OnLine.

[15] S. Kent. Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management. Internet Engineering Task Force: RFC-1422, February 1993.

[16] R. Khare and S. Lawrence. Upgrading to tls within http/1.1, 2000.

[17] Massimiliano Pala and Sean W. Smith. PEACHES and Peers. In 5^{th} *European PKI Workshop: Theory and Practice*, volume 5057, pages 223—238. Lecture Notes in Computer Science, Springer Verlag, June EuroPKI 2008.

[18] P. Maymounkov and D. Mazières. Kademlia: A peer-to-peer information system based on the xor metric. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 53–65, London, UK, 2002. Springer-Verlag.

[19] D. Meyer and K. Patel. Bgp-4 protocol analysis. Internet Engineering Task Force: RFC 4274, 2006.

[20] OpenID. Open identity homepage. OnLine.

[21] M. Pala. The PKI Resource Query Protocol (PRQP). Internet Engineering Task Force: Internet-Draft, November 2009.

[22] M. Pala and S. W. Smith. PEACHES and Peers. *Proceedings of the 5th European PKI Workshop: Theory and Practice*, 5057:223–238, June 2008.

[23] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In *SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, volume 31, pages 161–172. ACM Press, October 2001.

[24] I. Stoica, R. Morris, D. Karger, F. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, October 2001.

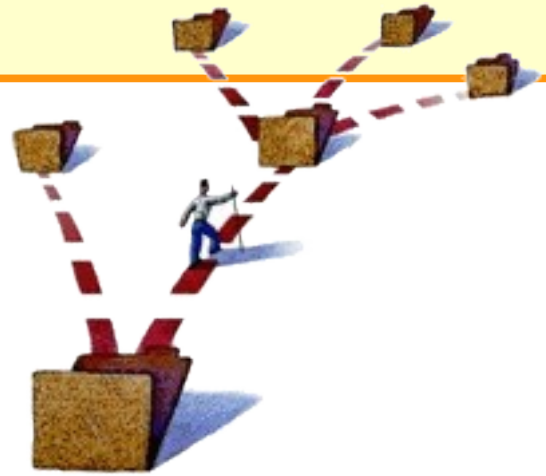
[25] TAGPMA. The Americas Grid Policy Management Authority. OnLine.

[26] M. Wahl, T. Howes, and S. Kille. Lightweight directory access protocol (v3), 1997.

[27] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical Report UCB/CSD-01-1141, UC Berkeley, # apr # 2001.

A proposal for

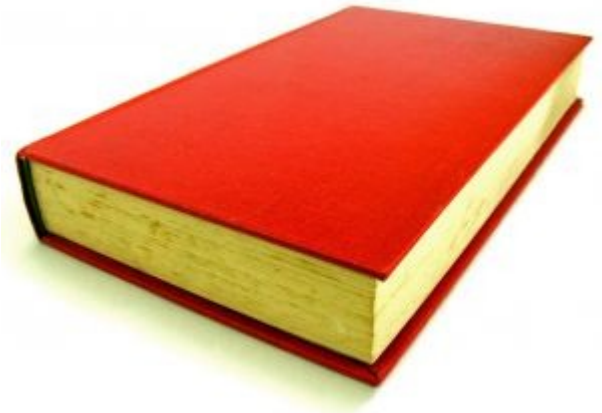
Collaborative Internet-Scale Trust Infrastructures Deployment: The Public Key System



Massimiliano Pala

Outline

- Motivations
- Model Description
- Message Definition
- The PKS Node
- Federated Identities
- Considerations
- Future Work



The Objective

- Ease deployment of Trust Infrastructures based on Public Key technology in the Internet
  - X.509 PKIs
  - DNSSEC

Motivations-1

- Heterogeneous deployment environment
  - How easy is it to interact with your PKI ?
- The Need for Federated Identities
  - FBPKI, HEBCA, 4BF, TACAR, IGTF, etc.
- Many different Protocols (X509)
  - SCVP, CMP, OCSP, TAMP...
- Other Public Key Infrastructures (DNSSEC)
  - Future Infrastructures (?)

Motivations-2

- Each day we rely on Public Key technologies for online authentication
  - Web Authentication
  - Physical Authentication

No support for
Trust Infrastructures Deployment
in the Internet

Current Needs Demand Solutions...

- DNSSEC to distribute certificates
  - Trust does not follow DNS hierarchies
  - Organizational Problems (DNS vs CA)
- Computing Grids TA distribution
  - Ad-Hoc TA distribution
  - No interoperability

Message to take away...

We need a **standardized**, **scalable** and **interoperable** system for PK support for the Internet

So far... It's been a Bumpy Ride!

Problem-1

- No Globally Authoritative Infrastructure
- No easy Interaction with different Infrastructures
  - PKI Resource Query Protocol (PRQP)

A Public Key System is needed to allow PK-enabled applications to discover and easily use resources offered by different Authorities

Problem-2

- Interaction with different parts of a PKI is difficult
  - Many Different PKI Protocols
  - Many Different Transport Protocols
    - HTTP, HTTPS, FTP, etc.
- Applications and Certificates
  - renewal, revokation

A Public Key System that mandates for a simple transport protocol capable of routing all current and future PKI messages

Problem-3

- Lack of ***contextual trust***
  - Classes of trust (eCommerce, eBanking, eMail)
  - Easy Trust Anchor Management
- Mobile devices
  - Local trust in home environment

A Public Key System that provides the ability to group TA according to specific environments to help users manage (or delegate) trust settings.

Problem-4

- Lack of support for federated identities
- Need to know if a CA/PK is part of a federation
  - Computing Grids, DNSSEC, etc.

A Public Key System that eases the deployment of federated identities by facilitating a method for disseminating information about which organization or federation use/include/trust a specific TA.

The Challenge

To provide a flexible support system for Trust Infrastructures deployment

.... SO ...

(very dramatic pause...)

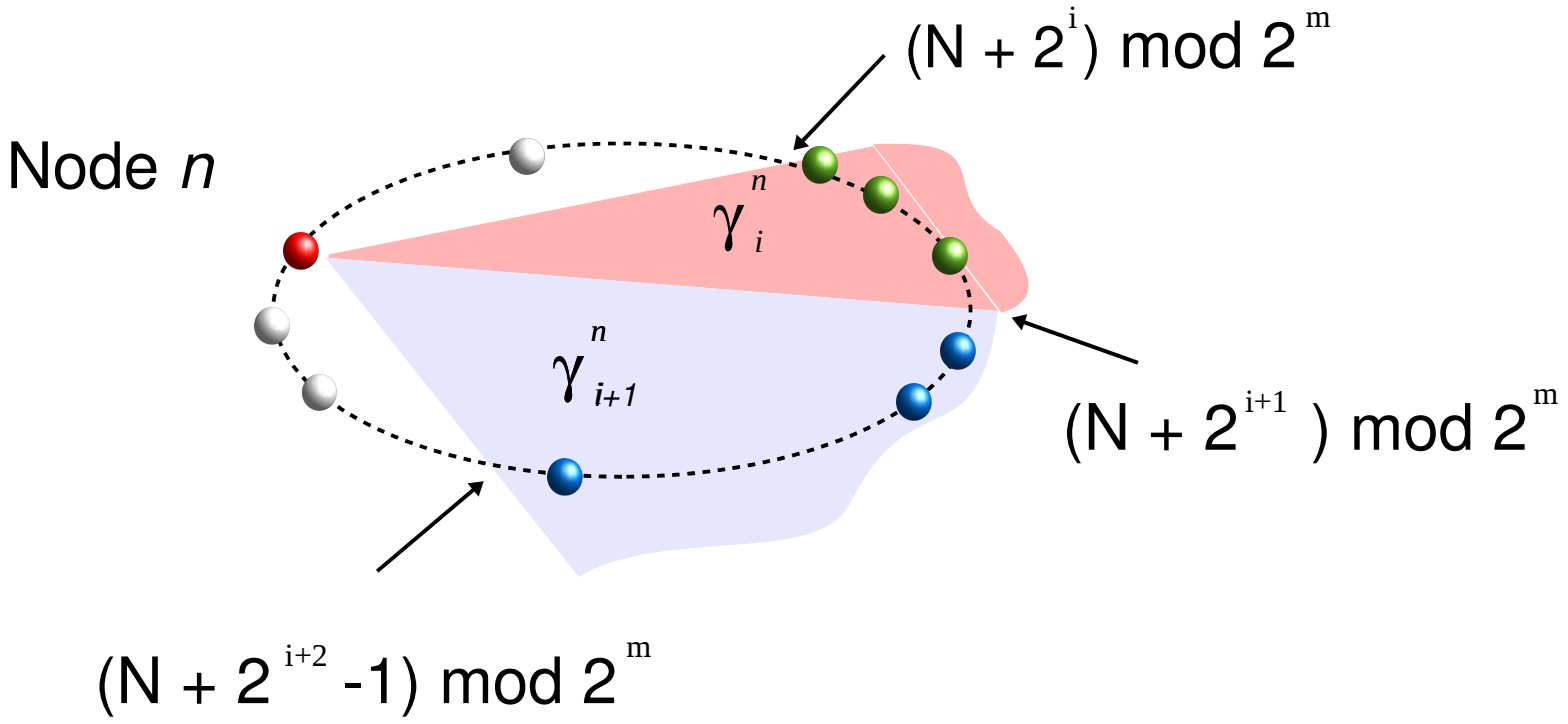
The Public Key System

- A system to support current needs for Trust Infrastructures (TI) deployment
  - Addresses the aforementioned problems
  - Increases Interoperability among TI
- Supports Public Key systems
  - Algorithm(s) agile
  - Backward compatible with deployed TI
- Internet Oriented
  - Scalability

The Public Key System (PKS)

- Peer-to-peer system based on DHT [Chord]
- Simple Operations
  - lookup()
  - join()
- Identifiers based on hash(PK) [PEACH]
  - m = bits hash function
- Each node keeps a lookup table
  - m entries

The Public Key System (PKS)



DHT Basics

- ID space hash(x)

$$\forall i \in [1, 2, \ \ldots \ , m], \quad \Rightarrow \quad x_i^n \quad = \quad (id_n + 2^i) \mod 2^m$$

- Lookup table n => $id_n < x_i^k$

$$\gamma_i^n \quad = \quad [x_i^n, x_{i+1}^n)$$

- Lookup in O(log(m))

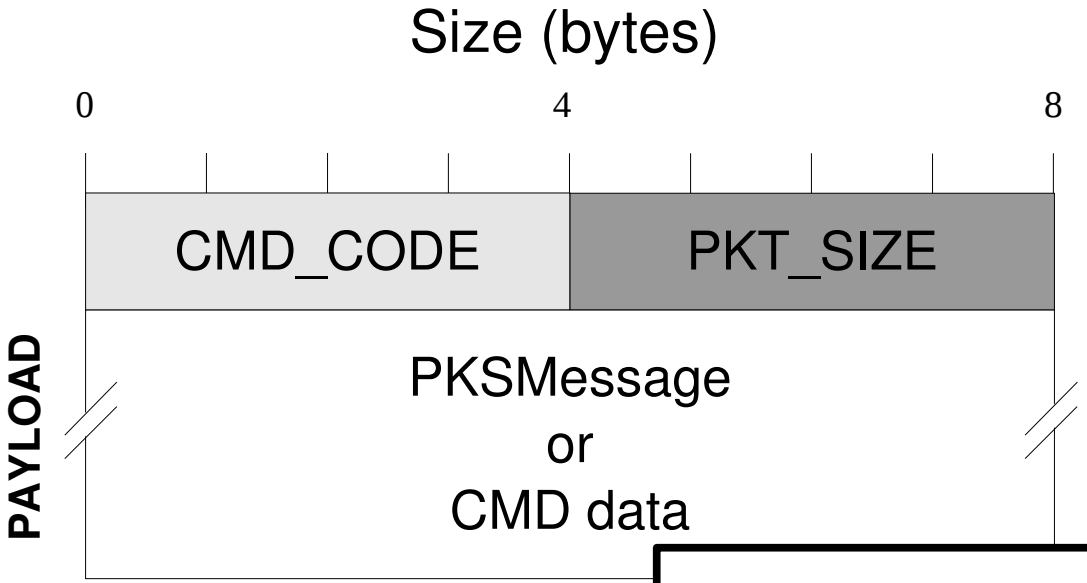
The Public Key System (PKS)

- n-th node lookup table

$$
\begin{aligned}
x_0^n &= (id_n + 2^0) \mod 2^m \\
&\vdots \\
x_m^n &= (id_n + 2^{m-1}) \mod 2^m
\end{aligned}
$$

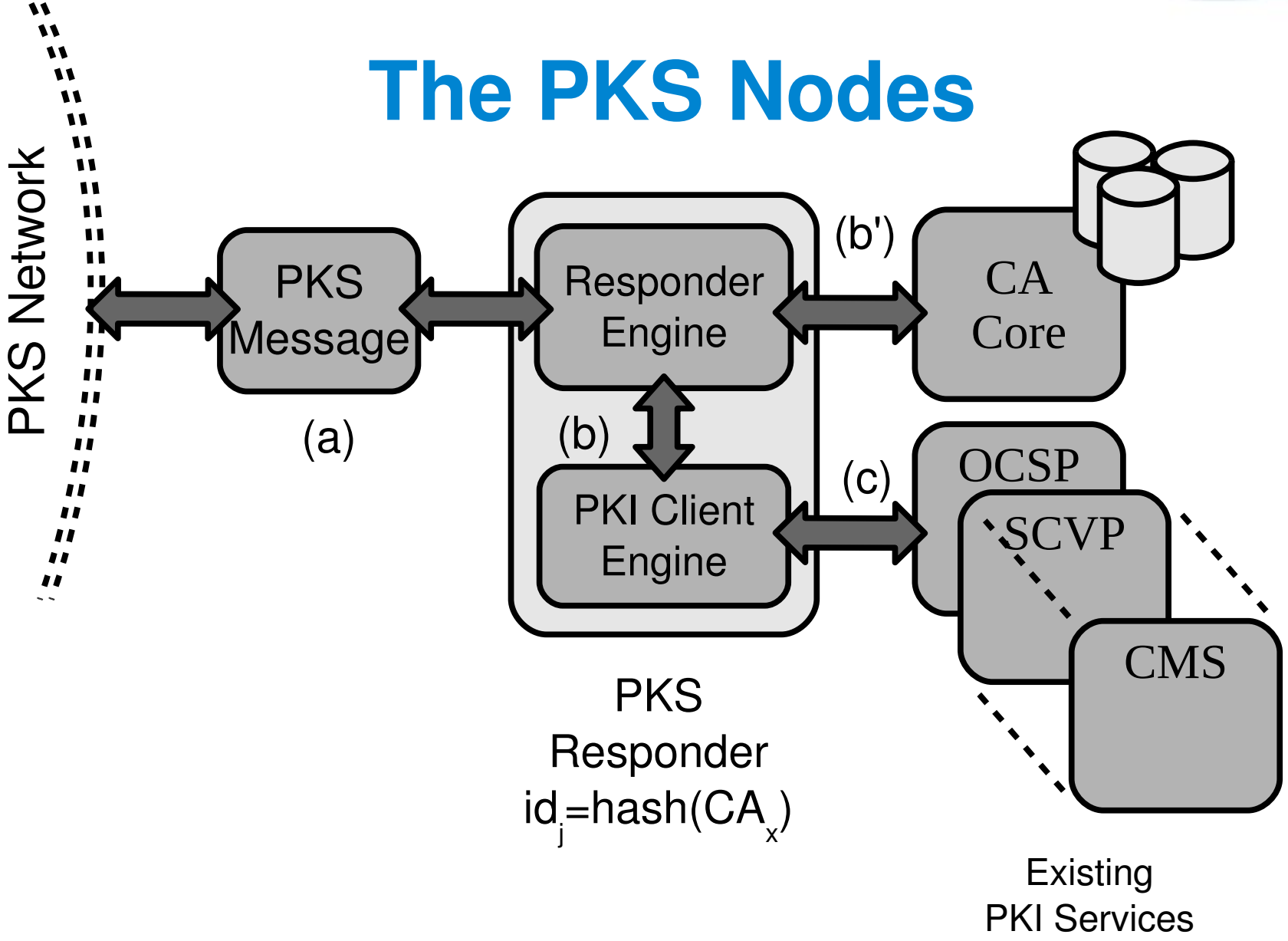
The PKS Message Format

Simplified Message System




```
PKSMessage ::= {
    protocol    OBJECT IDENTIFIER,
        --- Identifier for the data protocol
    targetNode    OCTET STRING,
        --- Target Node Identifier (hash)
    rawBytes      OCTET STRING
        --- Binary data (e.g., CMS message)
}
```

The PKS Nodes



Federated Identities

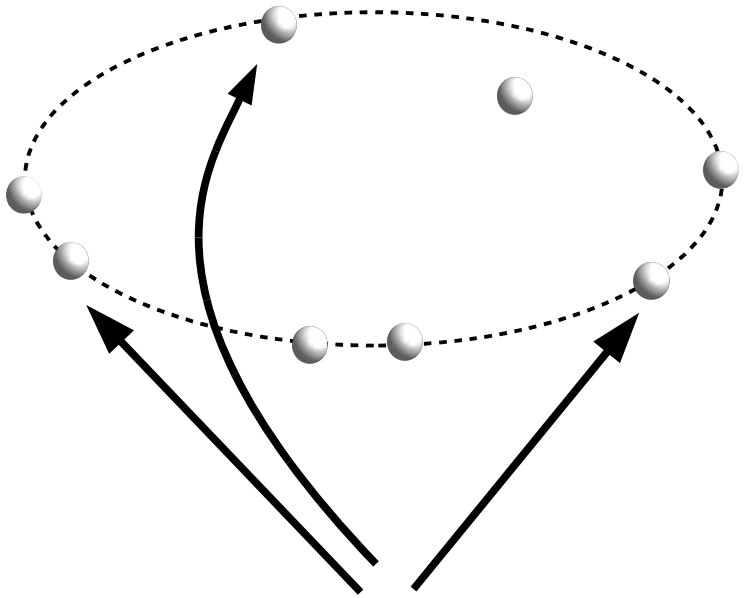
- PKS Federation Authorities (PK-FA)

$$id_j = hash(x_j)$$

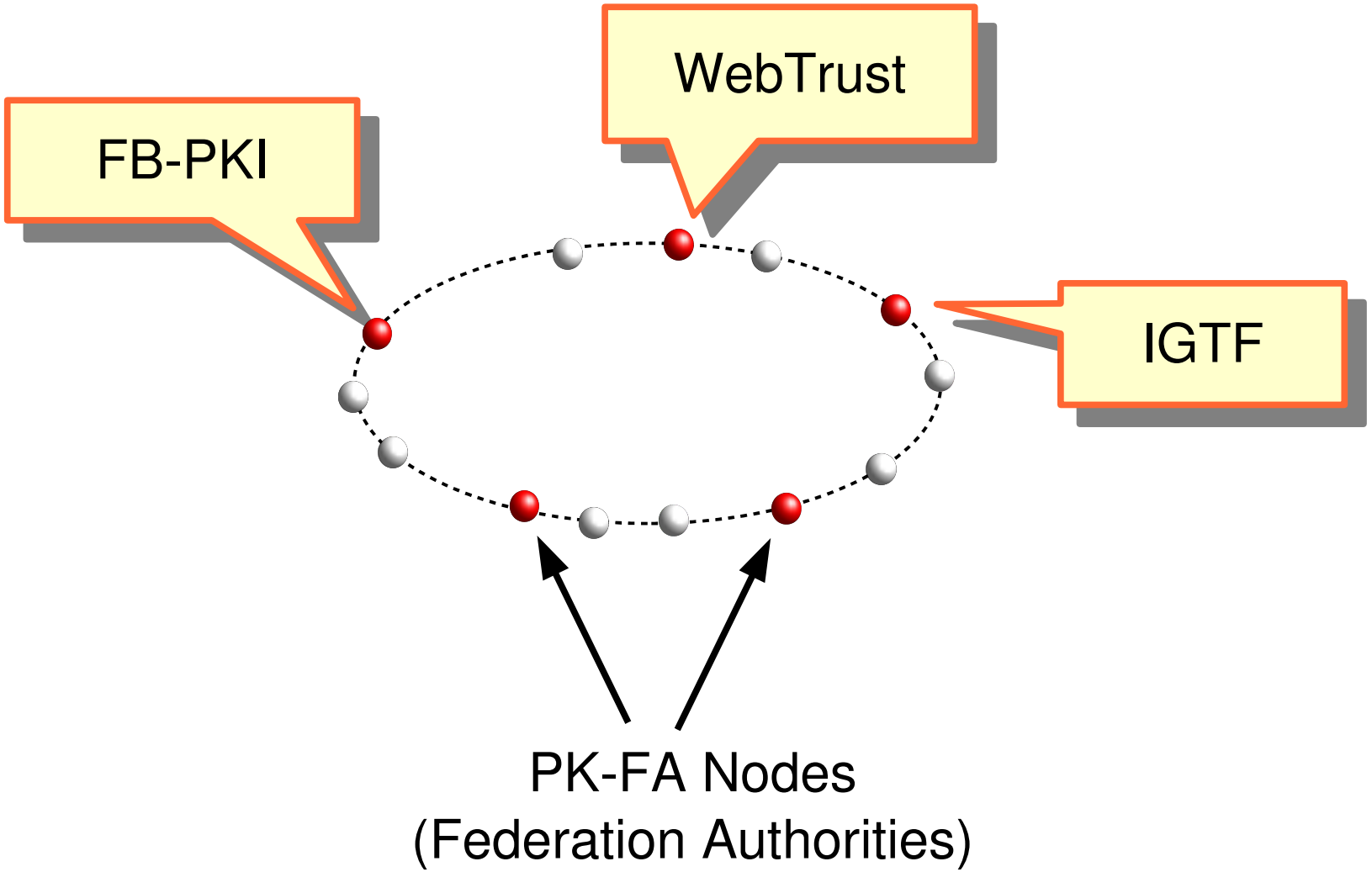
- PK-FA provides responses to client about a CA being part of a federation
  - Is this CA part of the Federal Government ?
  - Is this user's certificate part of TACAR ?
  - Is this certificate for an Internet DNS server ?

Extending the PKS Network

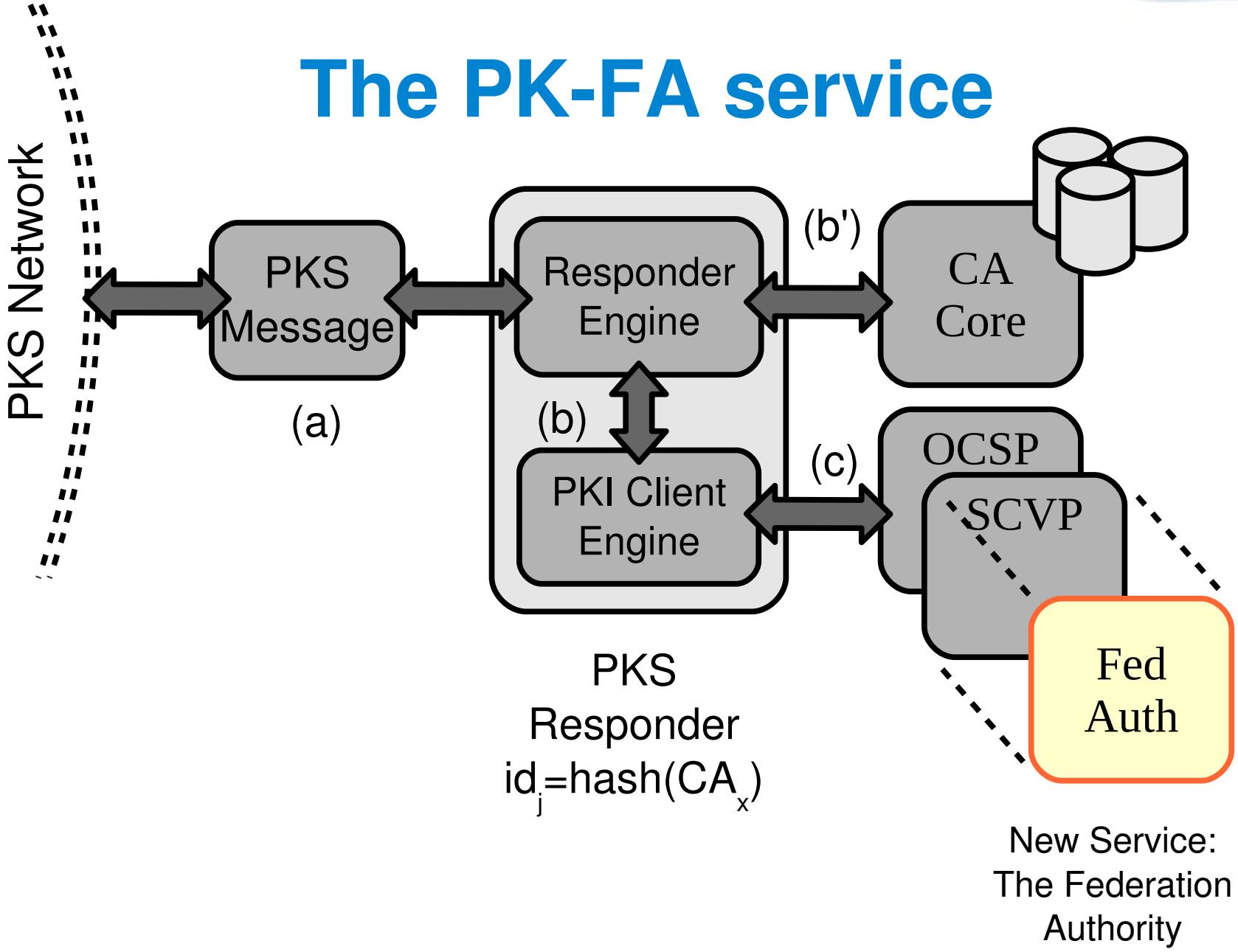
Let's add a new class of Nodes to the PKS network



Extending the PKS Network: Federation Authorities



The PK-FA service



Classes of Federations

- Hierarchical Federation Infrastructure
- Class Federation Authorities
  - Identifiers based on PK (not certs)
  - Local, Internet, Network, Organization, and Application
- Deployment of Trusted Keys for primary DNS domains
  - “.”, “.edu”, “.net”, “.org”, “.com”, etc.
  - Keys for “.” can be used/revoked/replaced

Conclusions

- We rely on PK technology
  - Digital IDs
  - Passports
  - DNSSec
- We need a Public Key System capable of supporting the use of PK on the Internet
- We proposed a PKS and a possible deployment design based on a collaborative approach

Future Work

- Deploy the system in a test bed
- Study attacks to the PK network
  - Malicious nodes, etc.
- Define an API for providing access to the PKS for:
  - Easy integration with existing OSes and Apps
- Publishing an I-D at IETF for consideration within the PK-NG WG (IRTF)

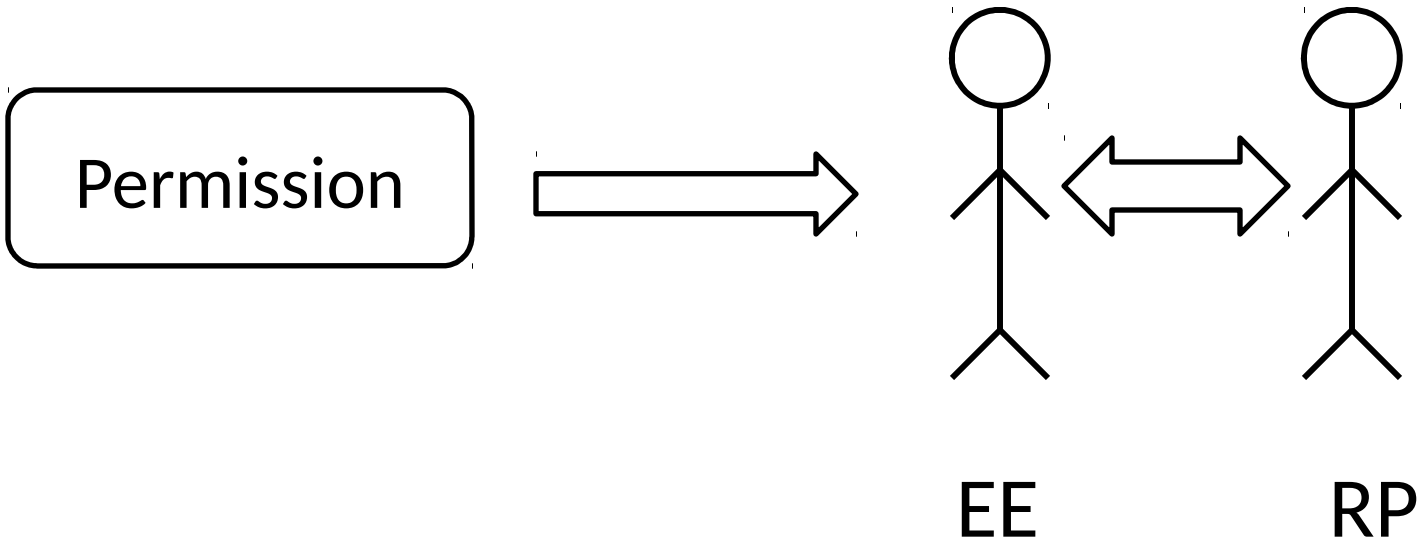
Contacts, Questions, etc.

- Email:
  - Massimiliano Pala <pala@cs.dartmouth.edu>
- Website:
  - http://www.openca.org/projects/ng/

LOA Panel

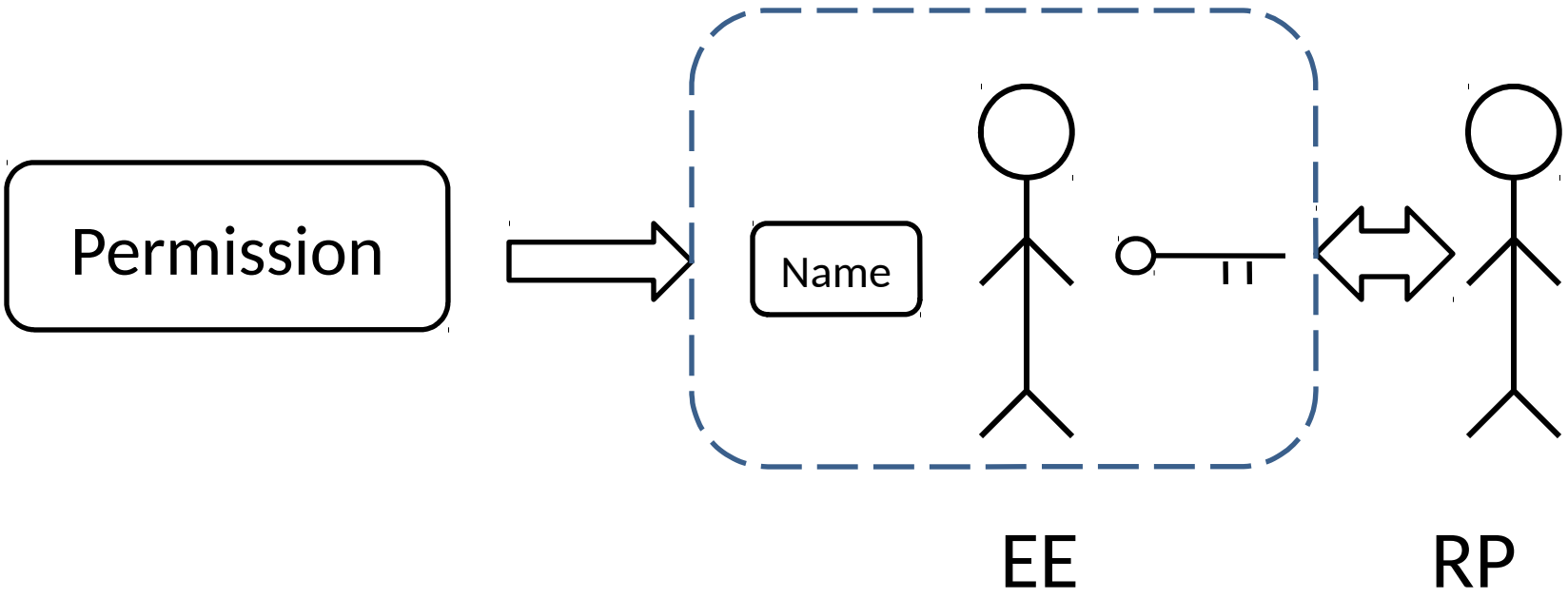
15 April 2008

Pre-computer model

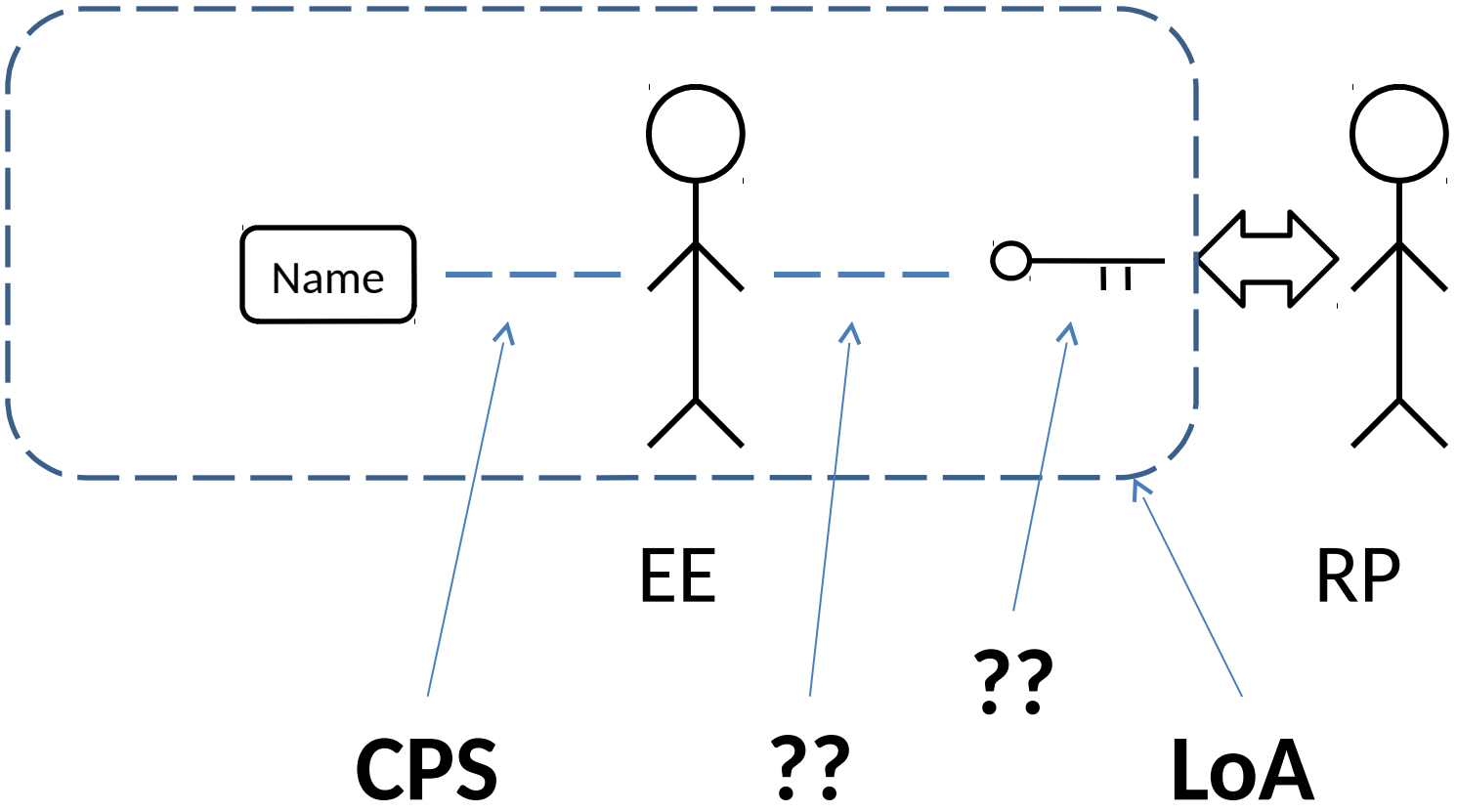


RP: Relying Party or Resource Provider

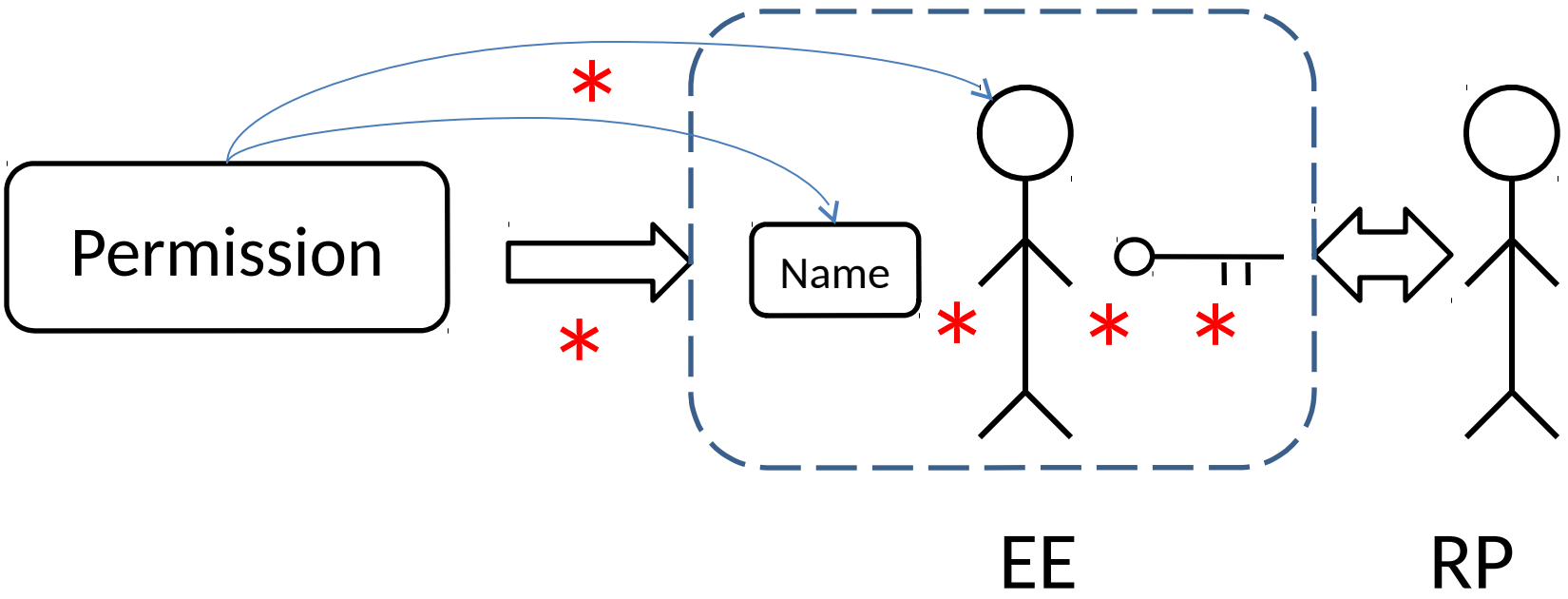
Computer-age model



PKI model



Partial Threat Model → Full LOA



* = Point of vulnerability

Expression of multiple permissions

LoA for Attributes

David Chadwick
University of Kent

Acknowledgements

- This research has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 216287 (TAS³ - Trusted Architecture for Securely Shared Services).

Current NIST 800-63 LoA Model

- Guidelines “to remotely authenticate a user’s identity to a Federal IT system”.
- Two components
  - Identity Proofing and Registration of applicant
  - Authentication mechanism used
- Combined into one LoA value in range 1 (lowest) to 4 (highest)
- Designed for a single system that both registers the user and authenticates the user and provides the identity of the user to the Federal IT system (as an identifier and optional attributes)

Deficiencies in NIST Model (1)

- What if a user has multiple authentication mechanisms provided by an IdP e.g. un/pw and a hardware PKI token?
  - Different LoAs should be provided per login session
- Leads to concept of Session LoA, which is dynamically computed from Registration LoA (fixed) and Authentication Mechanism LoA (variable)

Deficiencies in NIST Model (2)

- What if the system is distributed and the user's identity attributes are provided by multiple authorities? Authorisation is what is actually required, not just authentication
- So, you are David Chadwick? But what are you entitled to do?
- In federated identity management, a user's identity is now recognised as being a set of possibly distributed identity attributes, rather than an identifier and optional local attributes (which is assumed by NIST)
- E.g. "the user is a student of university X". This may be sufficient to authorise access to a resource (typical Shibboleth scenario). The resource does not need to know that the user is David Chadwick so the identifier is not needed.

The Way Forward Today (for a single IdP)

- In RBAC/ABAC systems access is granted based on the attributes of the user (one of which may, but need not, be a unique identifier)
- We can supplement the set of user attributes with the existing NIST LoA value assigned to the current session in order to provide finer grained access controls
  - E.g. Students with Session LOA 1 can read the module syllabus. Students with Session LOA 2 can upload their assignments
- We have had this implemented for several years in our open source software (PERMIS)

A Way Forward Today (for multiple IdPs)

- Users typically have accounts at multiple IdPs and need to provide attributes from several IdPs in order to gain access. The user configures a linking service to know (some of) these accounts
- When the user logs in, a Session LoA is dynamically computed by the authenticating IdP

Session LoA = Authentication LoA (if no attributes are asserted)

Session LoA = Lowest of Authentication LoA and Registration LoA (if at least one registered attribute is asserted)

- The linking service coordinates attribute assertion collection from the multiple IdPs
- Each of these attribute assertions need their own LoA but currently we have to munge these to fit the single session LoA by excluding those assertions with a lower LoA

And for Tomorrow - A Model

- A user registers with each IDP and is assigned a Registration LoA (according to the procedure that is used) which is attached to the user's registered attributes.
- The user is given one or more authentication tokens/mechanisms by the IdP each with its own Authentication LoA
- When the user logs in to an IdP, a Session LoA is dynamically computed for the session according to the formula

Session LoA = Authentication LoA (if no attributes are asserted to service provider)

Session LoA = Lowest of Authentication LoA and Registration LoA (if at least one registered attribute is asserted to service provider)

- All other linked IdPs create their own attribute assertions for this session and include their own LoA in the attribute assertion

Assertion LoA = Lowest of Session LoA and Registration LoA

- Service Provider has a fine grained ABAC policy in which each identity attribute in a rule has a required LoA. For the rule to be passed the assertion LoA must be GE to the required LoA

Example Use Case

- **Case: American Medical Schools (AAMC)**
- **Scenario:** The American Medicals Schools (AAMC) administer a test for admission into accredited US medical schools. Accounts are primarily given to users via e-mail verification to allow for the application process, but full identity proofing is then undertaken (fingerprinting and photo) when the students come to take the test. Campuses could benefit from capturing the value of the AAMC identity-proofing process.
- **LoA Details:** The initial Registration LoA is low (1) due to email verification only, which means that the Session LoA will remain low no matter how good the authentication mechanism is. After the students have taken the test, the Registration LoA is now high (say 3) due to fingerprinting etc., so the Session LoA can rise to the lower of the Authentication LoA and Registration LoA.

Example Use Case

- **Case: Students Using External Identities**
- **Scenario:** User creates an OpenID for a username you do not know and Provider does no checks as to who user is in the real world. However it has a good authentication mechanism (LoA 2). Any RP accepting the OpenID has reasonable assurance it is the same user each time (but not who the user is.) User then turns up as a student at University X. The university can do all its normal checks on the person e.g. have the right school exam results, have paid fees, am entitled to be in the UK, etc. (Registration LoA >2) but it does not need to issue its own authentication credentials. Instead it checks the technical quality of OpenID Provider, and that its processes are sufficiently robust to qualify as LoA 2, and then it can assert the student's identity attributes to service providers with a Session LoA of 2, even though the OpenID Provider doesn't know them.
- **LoA Details:** Although the OpenID Registration LoA is the lowest, since no attributes are asserted the Session LoA is 2 due to its good authentication procedures. Once the user registers at the University and is verified she can continue to use the OpenID and the university asserts its own attributes with a Session LoA of 2 since its Registration LoA >2.

Example Use Case

- **Case: E Commerce Site**
- **Scenario:** Online shopping at Amazon you provide a self assertion of your name and postal address (for delivery), a signed assertion from Visa that you have a credit card, and a signed assertion from IEEE that you are a member and thus eligible for a discount. Visa has provided you with a smart card and PIN for authentication
- **LoA Details:** Your session LoA is relatively high (say 3) due to the smartcard authentication mechanism, but your name and address card is self asserted so this has the lowest LoA (1). Your credit card attribute is sent by the issuer with a high LoA (3) due to the rigorous registration checks the bank undertook before issuing the card, whereas the IEEE membership attribute has an LoA of 2 due to the limited amount of registration checking they did.

Conclusions

- Federated Identity Management systems recognise that users will need to provide attributes from multiple IdPs within a single session, but only need to authenticate once
- The session LoA should be dynamically computed based on the authentication mechanism used, the IdP used, and what it asserts
- Each IdP should be able provide its own Assertion LoA along with the attributes it asserts
- This allows the SP to have a fine grained authorisation policy which places a LoA requirement on each identity attribute

LOA of Attributes: A Community-Based Approach

Dr Ken Klingenstein
Director, Internet2 Middleware and Security

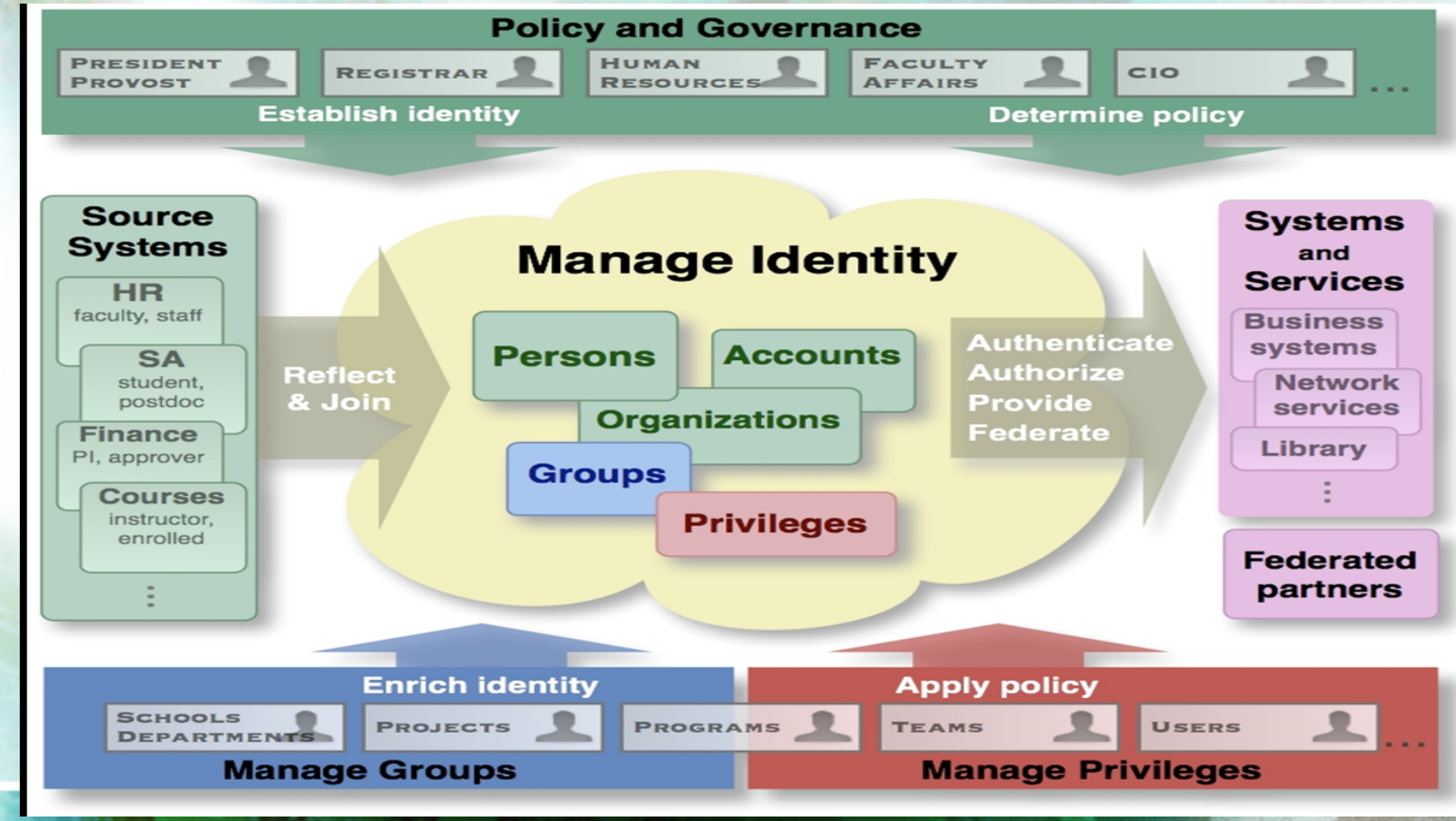
INTERNET®

Topics

- The larger picture – the Tao of Attributes
- The theory of LOA of attributes
  - Parameters, mathematics, contracts, audits
- The practice of LOA of attributes
  - Common community practices
  - Common software and systems
  - Common relying parties
- Early lessons learned

kjk@internet2.edu

Enterprise IdM middleware plumbing



The Attribute Ecosystem

- Authentication is very important, but identity is just one of many attributes
- And attributes provide scalable access control, privacy, customization, linked identities, federated roles and more
- We now have our first transport mechanisms to move attributes around – SAML and federations
- There will be many sources of attributes, many consumers of attributes, query languages and other transport mechanisms
- Together, this attribute ecosystem is the “access control” layer of infrastructure

kjk@internet2.edu

Attribute use cases are rapidly emerging

Disaster “first responders” attributes and qualifications dynamically

Access-ability use cases

Public input processes – anonymous but qualified respondents

Grid relying parties aggregating VO and campus attributes

The “IEEE” problem

The “over legal age” and the difference in legal ages use cases

Self-asserted attributes – friend, interests, preferences, etc

kjk@internet2.edu

The Tao of Attributes workshop 属性之道

- Purpose of workshop was to start to explore the federal use case requirements for attributes, aggregation, sources of authority, delegation, query languages, etc.
- Participants were the best and brightest – the folks who invented LDAP, SAML, OpenId, etc.
- Webcast at http://videocast.nih.gov/PastEvents.asp
- Twittered at TAOA
- http://middleware.internet2.edu/tao-of-attributes/

Categories of attributes

- Self-asserted
- Enterprise and organizationally asserted
  - Values assigned by business processes
- Third party asserted
  - Citizenship by SEVIS
  - “Verified by Verisign”
  - “Gleaned by Google”

kjk@internet2.edu

Attribute aggregation at the RP

- From where - Gathering attributes from multiple sources
  - From IdP or several IdP
  - From other sources of authority
  - From intermediaries such as portals
- When - static and dynamic acquisition
  - Some attributes are volatile (group memberships); others are static (Date of Birth)
  - Some should be acquired per assertion; some once in a boarding process
- Will require a variety of standardized mechanisms –
  - Bulk feeds, user activated links, triggers

Principles of the Tao

- Least privilege/minimal release
- Using data “closest” to source of authority
- Late and dynamic bindings where possible
- Dynamic identity data increases in value the shorter the exposure.
- How much meaning is encoded in the attribute versus context, metadata?
- How much flat attribute proliferation can be managed through a structured data space?

kjk@internet2.edu

- “In theory, there is no difference between theory and practice. But, in practice, there is.”
  - Jan L. A. van de Snepscheut/Yogi Berra

kjk@internet2.edu

The Theory of LOA of attributes

- Parameters
  - LOA of authn, integrity of the source systems, integrity of the attribute transports, etc.
- Mathematics - unknown
- Contracts –
  - Explicitly defined business processes for assigning values to attributes
  - Managing risk
- Audits
  - Establishing compliance with the contract

kjk@internet2.edu

Before we practice…

- The limits of 800-63
- Attributes without identity are “creepy”
- The many possible issuers of “over 21”
- Role of identity proofing in LOA of attributes and step-up identity

kjk@internet2.edu

The Practice of Attributes in R&E

- There exists a set of widely shared attributes that work with consistent LOA for the applications that use them.
  - eduPersonaffiliation (the relationship of the subject to the institution)
  - epTID (the binding of a persistent, opaque identifier to an individual)
- Who relies on them today?
  - MS to distribute software
  - Elsevier to distribute content
  - Student travel to provide discount travel passes
  - Many, many others

LOA, attributes and collaboration

- VO's are the heart of science, research and collaboration
- Roles and attributes scoped by collaboration; the "systems of record" are the PI's

Lessons learned

- Commonality drives rough consensus and working attributes
  - E.g. student-ness, .edu-ness
  - Provide a few common base attributes
    - E.g. epTID, member of the IdP
- Extensible attributes entitlements – establishes syntax and hint of semantics
- Control the vocabulary
  - Principle of parsimony –more value -> more complexity
  - Create new schema rather than enlarge vocabulary

kjk@internet2.edu

Powerful Insights.
Proven Delivery.™

Levels of Assurance for Attributes

NIST IDtrust
4/15/2010

Chris Louden

protiviti®
Risk & Business Consulting.
Internal Audit.

Agenda

Perspectives:

- Levels of Assurance (LOA)
- Sources of Authority

Disclaimer:

- Views presented are not necessarily the views of my employer or my clients

protiviti®

Levels of Assurance

Two sides to usable Levels Of Assurance:

- Assurance needed by the RP
- Assurance provided by the attribute authority

protiviti®

Levels of Assurance

Assurance Needed by the RP

- Some uses require more assurance than others:
  - Convenience for the user… "Welcome John"
  - Basis of Access Control
    - Is a Privilege an Attribute?
  - Attributes are often more important than identity…
    - All police officers can carry a gun
    - All "John Smith" can carry a gun
- How sure does the RP need to be in this situation?
  - Generally risk based
  - Specifically the risk of a false positive
    - This person is not really a police officer

protiviti®

Levels of Assurance

Is M-04-04 Adoptable for Attributes?

	Assurance Level Impact Profiles			
Potential Impact Categories for Authentication Errors	**1**	**2**	**3**	**4**
Inconvenience, distress or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or agency liability	Low	Mod	Mod	High
Harm to agency programs or public interests	N/A	Low	Mod	High
Unauthorized release of sensitive information	N/A	Low	Mod	High
Personal Safety	N/A	N/A	Low	Mod High
Civil or criminal violations	N/A	Low	Mod	High

protiviti®

Levels of Assurance

Assurance provided by the Authority

- Practices used to establish the attribute values
- Practices used to maintain the values
- Proper controls to protect the attribute database
  - Basic security controls for data integrity
  - Access by subject?
- Trustworthiness of the bindings
  - Attribute bound to a common name?
  - Attribute bound to a session context?
- Type of Authority
  - Different assurance for different types…
  - Different key practices for different types…

protiviti®

Sources of Authority

Types of Authority

- "Natural Authority"
  - Employer for employment
  - SSA for SSN
  - Department of Motor Vehicles for drivers license number
- "Proper Diligence"
  - Service provider checked appropriate sources, gathered appropriate evidence, etc
- "Trusted Administrator"
  - Administrator sets the role & "they ought to know"
  - Often used for delegation
  - "Superuser" grants access to Administrators, they set up others

protiviti®

Sources of Authority

Issues

- What role does the subject have?
  - What if the SSA says you're dead?
  - Does the subject always reconcile with the source?
  - Can the subject reconcile with "Proper Diligence" authority?
- What do authorities bind attributes to?
  - Common name?
  - Authenticated Session?
  - Credential Identifier?
- Can Authorities delegate?
  - Do delegates necessarily inherit authority?

Sources of Authority

Needs & Tools

- How do you anchor attribute trust?
  - Common trust anchor for attributes and identity?
  - Different anchors for different namespaces?
- Do standards allow different authorities for attributes & identity?
  - Can the products do that?
    - Verify the identity claim & that the IDP is trusted
    - Find the attribute authority & request an attribute claim
    - Verify the attribute claim & that the authority is trusted for this claim
    - Verify these claims are bound to this session?
    - Verify this attribute is bound to this identity?

protiviti®

Levels of Assurance

Needs & Tools

- Common Identifiers
  - Is jsmith the policy officer or is smithj?
- Do we need an 800-63 equivalent?
  - Maybe just best practices for Natural Authorities?
- Do we need an *AuthN Context* equivalent?

protiviti®

Agenda

Perspectives:

- ✓ Levels of Assurance
- ✓ Sources of Authority

Chris.Louden@pgs.protiviti.com

protiviti®

LOA of Attributes: An Examination

Peter Alterman, Ph.D.
Senior Advisor for Strategic Initiatives
National Institutes of Health

Center for Information Technology

Fundamentals

- Attributes are consumed by relying party applications for AuthZ and/or provisioning;
- Attributes may be assigned by many issuers, including relying party apps, and these issuers are authoritative for them;
- It doesn't look like there will be consensus on the form of the attributes any time soon;

Basic Principles

- The issuer of attributes is authoritative for the validity of those attributes;
- Any useful user credential is likely to include attributes from more than one issuer;
- Attributes may be stored or aggregated anywhere;
- Relying party applications are likely to be both consumers of attributes and issuer of attributes.

Existing Models

- **X.500ish**: local repositories hold attributes (assumes all attributes are issued locally) and some are exposed;
- **Shibboleth**: user proxy service holds attributes (punts the question of issuer/reliability)
- **Silo-Land**: each relying party application assigns attributes – usually roles and AuthZ – and stores them locally (since the app is issuing and storing them, they are authoritative for them)

Key Shortcomings of Existing Models

- Transaction protocols are technology-specific – requires intermediate functionality;
- Attribute exchange is pairwise today – will not scale – includes discovery and validation – see above;
- No trust infrastructure for attributes that is comparable to that for identity.

The Million Dollar Question

- In a federated world, how can a relying party application know it can trust an attribute issued by another entity?

Proposed Solutions

- Keep the siloed approach, where each application issues and manages attributes locally;
- Local Back-End Attribute Exchanges (BAE) store attributes and pointers to issuing entity data stores;
- Wait for Government to issue attribute policies comparable to identity policies;
- Select an industry entity (Internet Society, OASIS, ISO, etc.) to host the design, development and construction of a global attribute management infrastructure, such as an uber-BAE.

Why LOA of Attributes Is More Trouble Than It's Worth

- Any separation of attribute validation from issuer introduces trust and security threats which rapidly degrade the utility of attributes;
- Proxied attribute validation requiring LOA also requires a common body of policy, an authoritative source for policy and a high assurance assessment infrastructure;
- Informal agreements don't scale – reintroduces the pairwise model and there is no way to mediate among multiple pairwise models.

Attribute LOA Should Be Binary (but no solution is without its issues)

- Let the issuer validate attributes. Then the answer is either Y or N (yes, it's like the X.509 model)
- Requires attributes to include a pointer to the issuer and would require the issuer to maintain a repository

Caboose

- Because of our experience and the general culture of our business, we are inclined to find elegant, complex solutions to issues. That should be avoided like the plague in this case.
- *Contact info: peter.alterman@nih.gov*

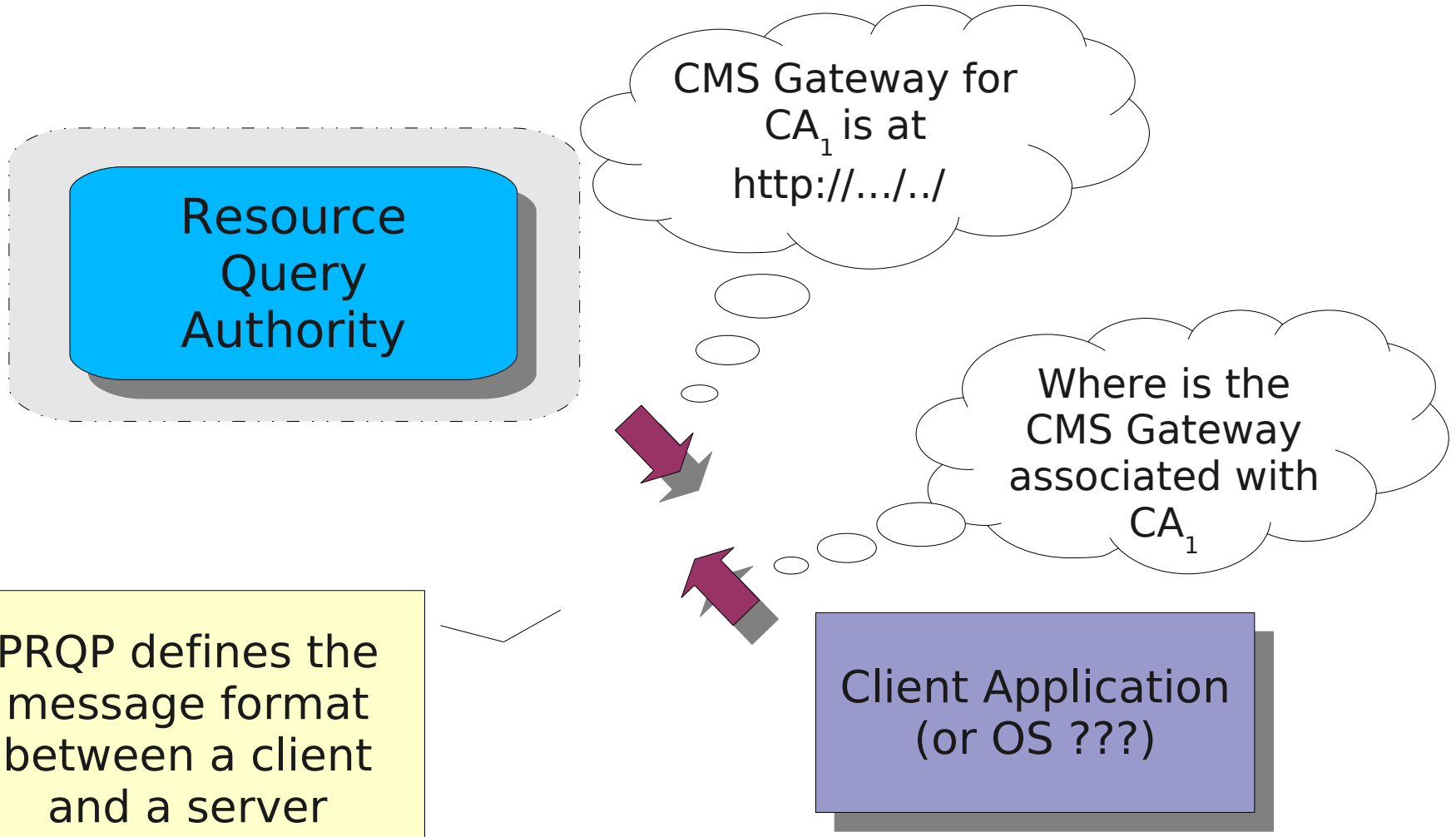
PKI Resources Query Protocol Deployment

Massimiliano Pala <pala@cs.dartmouth.edu>
OpenCA Project Manager <project.manager@openca.org>

PKI Resources Discovery

- Pointers to Resources
  - **Extensions in Certificate**
  - **Ad-Hoc Configurations in Apps**
  - **Advertise them on the CA's web pages**
- The PKI Resource Query Protocol
  - **Working Item at PKIX WG**
  - **Experimental Track**

PKI Resource Discovery Protocol



PRQP & Document Status

- Simple client-server protocol
- Defines two type of messages
  - **PRQP Request**
  - **PRQP Response**
- Updated beginning of 2010 (v04)
  - **Small Fixes**
  - **Addition of new OIDs for Grid Services**

Updated OIDs

	OID	Text	Description
PKIX	id-ad 1	ocsp	OCSP Service
	id-ad 2	calssuers	CA Information
	id-ad 3	timeStamping	TimeStamping Service
	id-ad 10	dvcs	DVCS Service
	id-ad 11	scvp	SCVP Service
General PKI Operations	id-ad 50	certPolicy	Certificate Policy (CP) URL
	id-ad 51	certPracticesStatement	Certification Practices Statement (CPS) URL
	id-ad 60	httpRevokeCertificate	HTTP Based (Browsers) Certificate Revocation Service
	id-ad 61	httpRequestCertificate	HTTP Based (Browsers) Certificate Request Service
	id-ad 62	httpRenewCertificate	HTTP Based (Browsers) Certificate Renewal Service
	id-ad 63	httpSuspendCertificate	Certificate Suspension Service
	id-ad 40	cmsGateway	CMS Gateway
	id-ad 41	scepGateway	SCEP Gateway
	id-ad 42	xkmsGateway	XKMS Gateway
	eng-ltd 3344810 10 2	webdavCert	Webdav Certificate Validation Service
	eng-ltd 3344810 10 3	webdavRev	Webdav Certificate Revocation Service
Grid	id-ad 90	accreditationBody	Accreditation Body URL
	id-ad 91	accreditationPolicy	Accreditation Policy
	id-ad 92	accreditationStatus	Accreditation Status Document
	id-ad 95	commonDistributionUpdate	Grid Distribution Package
	id-ad 96	accreditedCACertificates	Certificates of Currently Accredited CAs

Deployment in TACAR

- TACAR Project
  - **TERENA Academic CA Repository**
  - **Identification/authorisation procedures**
  - **Most of the EuGridPMA root CAs**
  - **National Research and Education Networks**
- PRQP Management included in the new CA Management Panel
- Server hosted at Dartmouth College
  - **Certificate Issued by TERENA's CA**
  - **Responder for all TACAR's CA**

Deployment in FBPKI

- Initial Deployment in ICAM test lab
  - **Open Source Software**
  - **Evaluation for deploying the protocol within the FBPKI architecture**
- Just Started!
  - **Expect some news in the next few months**

Available Software

- Open Source implementation (PRQPD) available
  - **OpenCA Labs**
  - **OpenCA PKI support for PRQP build in v1.1.0+**
  - **UNIX operating system(s)**
  - **Based on LibPKI library**
    - Ease-to-use PKI Library
- New release available (v0.5.0)
- Client implemented (?) in PKIF

Conclusions

- Move PRQP from Experimental to Standard Track
  - **Move to standard-track I-D**
- Extend support for major clients
  - **Firefox**
  - **Operating Systems**
- Continue the development of the PRQP Server
  - **OpenCA Labs**

Questions & Contacts

- Dartmouth College
  pala@cs.dartmouth.edu
- OpenCA
  madwolf@openca.org

- Website
  http://www.openca.org/projects/prqpd
  http://www.openca.org/wiki/