

IDtrust 2009

8th Symposium on Identity and Trust on the Internet

Program

Notes

Transportation

There will be a shuttle leaving the Gaithersburg Holiday Inn at 8:00 a.m. Tuesday morning to travel to NIST. The shuttle will leave at 8:15 a.m. Wednesday and Thursday. The shuttle will return to the hotel at the end of the sessions on Tuesday and Wednesday. There will not be shuttle service the afternoon of Thursday.

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 8th Symposium on Identity and Trust on the Internet.

Blogging

Participants and observers are encouraged to use the tag "idtrust2009" when blogging and tweeting about the symposium.

Tuesday, April 14, 2009 - 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: Kent Seamons, *Brigham Young University* (Slides: ppt)

9:10 - 10:00 Keynote Talk I

Federal Authentication and Identity Programs are Making Progress and Impacting Industry, But Much Work Remains

(Presentation slides: ppt)

Dan Blum, *Burton Group*

10:00 - 10:15 Break

10:15 - 11:40 Session 1 - Panel - Comparative Identity Systems

Panel Moderator: Kent Seamons, *Brigham Young University* (Slides: ppt)

Radia Perlman, *Sun* (Slides: ppt)

Ken Klingenstein, *Internet2* (Slides: ppt)

Paul Trevithick, *Higgins Project* (Slides: pdf)

George Fletcher, *AOL* (Slides: pdf)

11:40 - 12:00 Break

12:00 - 1:00 Session 2: Panel - Can Federations Scale?

Panel Moderator: Dan Blum, *Burton Group*

Peter Alterman, *General Services Administration*

Ken Klingenstein, *Internet2*

Roger Lambert, *Covisint*

1:00 - 2:00 Lunch

2:00 - 3:30 Session 3 - Technical Papers - Identity Management

Session Moderator: David Chadwick, *University of Kent*

Identity, Credential, and Access Management at NASA, from Zachman to Attributes

(Presentation slides: ppt)

Corinne Irwin, *NASA*

Dennis Taylor, *NASA (INDUS Corp.)*

Personal Identity Verification (PIV) Cards as Federated Identities - Challenges and Opportunities

(Presentation slides: pdf)

Sarbari Gupta, *Electrosoft*

A Calculus of Trust and Its Applications to PKI and Identity Management

(Presentation slides: pdf)

Jingwei Huang, *University of Illinois*

David Nicol, *University of Illinois*

3:30 - 4:00 Break

4:00 - 5:30 Session 4 - Panel - What is Special About My Application

Panel Moderator: Tim Polk, *NIST*

Walter G. Suarez, *Institute for HIPAA/HIT Education and Research* (Slides: ppt)
Andrew Regenscheid, *NIST* (Slides: ppt)
Barry Leiba, *Internet Messaging Technology* (Slides: pdf)
Bob Sunday, *Government of Canada* (Slides: ppt)
Stephen Whitlock, *Boeing* (Slides: ppt)

5:30 Bus Departs for Gaithersburg Holiday Inn

6:00 Social Gathering and Dinner Buffet - Gaithersburg Holiday Inn

Wednesday, April 15, 2009 - Full Day

8:15 Bus Departs from Gaithersburg Holiday Inn for NIST

8:30 - 9:00

Registration and Continental Breakfast

9:00 - 9:50 Keynote Talk II

Identity and Trust in Context

(Presentation slides: pdf)

Peter G. Neumann, *SRI*

9:50 - 10:10 Break

10:10-11:40 Session 5 - Technical papers - Federations and Virtual Organizations

Session Moderator: Scott Rea, *Dartmouth College*

Palantir: A Framework for Collaborative Incident Response and Investigation

(Presentation slides: ppt)

Himanshu Khurana, *University of Illinois*

Jim Basney, *NCSA, University of Illinois*

Mehedi Bakht, *NCSA, University of Illinois*

Mike Freemon, *NCSA, University of Illinois*

Von Welch, *NCSA, University of Illinois*

Randy Butler, *NCSA, University of Illinois*

Safeguarding Digital Identity: The SPICI (Sharing Policy, Identity, and Control Information) Approach to Negotiating Identity Federation and Sharing Agreements

(Presentation slides: pdf)

Deborah Bodeau, *The MITRE Corporation*

Usable Trust Anchor Management

(Presentation slides: pdf)

Massimiliano Pala, *Dartmouth College*

Scott Rea, *Dartmouth College*

11:40 - 12:00 Break

12:00 - 1:00 Session 6: Special Session: Browser Security

Latest advances in browser security- a report card and forthcoming W3C WSC specification

(Presentation slides: pdf)

Anil Saldhana, *Red Hat*

Which Browsers handle SSL certificates in a standard way?

David Chadwick, *University of Kent*

1:00 - 2:00 Lunch

2:00 - 3:30 Session 7: Technical Papers - Applied Cryptography

Session Moderator: Nelson Hastings, *NIST*

Privacy-Preserving Management of Transactions' Receipts for Mobile Environments

(Presentation slides: pdf)

Federica Paci, *Purdue University*

Ning Shang, *Purdue University*

Sam Kerr, *Purdue University*

Kevin Steuer, Jr, *Purdue University*

Jungha Woo, *Purdue University*

Elisa Bertino, *CERIAS, Purdue University*

Quantum Resistant Public Key Cryptography: A Survey

(Presentation slides: ppt)

Ray Perlner, *NIST*

David Cooper, *NIST*

3:00 - 3:30 Session 8: Technical Papers - Information Cards

Session Moderator: Peter Alterman, *General Services Administration*

FileSpace - An Alternative to CardSpace that supports Multiple Token Authorisation and Portability Between Devices

(Presentation slides: ppt)

David Chadwick, *University of Kent*

3:30 - 4:00 Break

4:00 - 5:30 Session 9: Panel - Comparative Authorization Models

Panel Moderator: John Sabo, *CA, Inc.*

Radia Perlman, *Sun* (Slides: ppt)

Rakesh Radhakrishnan, *Sun* (Slides: pdf)

Dr. Ramaswamy (Mouli) Chandramouli, *NIST* (Slides: ppt)

Tim Brown, *CA, Inc.* (Slides: ppt)

5:30 Bus Departs for Gaithersburg Holiday Inn

Dinner (on your own)

Thursday April 16, 2009 - Half Day

8:15 Bus Departs from Gaithersburg Holiday Inn for NIST

8:30 - 9:00

Registration and Continental Breakfast

9:00-10:00 Session 10 - Panel - Defensive PKI - What Happens when PKI Fails?

Panel Moderator: Carl Ellison, *Microsoft* (Slides: pptx pdf)

Tim Polk, *NIST*

Stephen Whitlock, *Boeing*

Kelvin Yiu, *Microsoft*

10:00 - 10:30 Session 11 - Technical Paper - Usability

Session Moderator: Peter Alterman, *General Services Administration*

Usable Secure Mailing Lists with Untrusted Servers

(Presentation slides: pptx)

Rakesh Bobba, *NCSA, University of Illinois*

Joe Muggli, *NCSA, University of Illinois*

Meenal Pant, *NCSA, University of Illinois*

Jim Basney, *NCSA, University of Illinois*

Himanshu Khurana, *University of Illinois*

10:30 - 11:00 Break

11:00 - 12:00 Session 12: RUMP Session (Work in Progress)

Session Chair: Neal McBurnett, *Internet2*

Why Create new ID-related Standards?

(Presentation slides: ppt)

Shivaram Mysore, *Key Pair Technologies*

DNSSEC Update

Tim Polk, *NIST*

Digital Identity

(Presentation slides: ppt)

Bill MacGregor, *NIST*

Group signature with selective disclosure for privacy enhanced ID management

(Presentation slides: pdf)

Kazue Sako and Jun Furukawa, *NEC*

Identity Quality Assurance

(Presentation slides: ppt odp)

Wes Kussmaul, *Reliable Identities*

'Break the Glass' Obligation Policies Demo

(Presentation slides: pdf)

David Chadwick, *University of Kent*

Easy-To-Use Secure Email

(Presentation slides: ppt)

Kent Seamons, *Brigham Young University*

Delivering Anonymous Certificates

(Presentation slides: ppt)

James Fisher, *Noblis*

12:00-12:30 Wrap up

See Also

This workshop is part of the IDtrust Symposium Series

- 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

IDtrust2009

April 14-16, 2009
National Institute of
Standards and Technology
Gaithersburg, MD

8th Symposium on Identity and Trust on the Internet

Kent Seamons
Brigham Young University
Program Chair

Program Committee

- Gail-Joon Ahn, *Arizona State*
- Peter Alterman, *GSA*
- Abbie Barbir, *Nortel*
- John Bradley, *ooTao*
- David Chadwick, *University of Kent*
- Carl Ellison, *Microsoft*
- Stephen Farrell, *Trinity College Dublin*
- Peter Gutmann, *University of Auckland*
- Adam J. Lee, *University of Pittsburgh*
- June Leung, *FundSERV*
- Simson Garfinkel, *Naval Postgraduate School*
- Eve Maler, *Sun Microsystems*
- Neal McBurnett, *Internet2*
- Clifford Neuman, *USC*
- Arshad Noor, *StrongAuth*
- Eric Norman, *University of Wisconsin*
- Radia Perlman, *Sun Microsystems*
- Tim Polk, *NIST*
- Scott Rea, *Dartmouth College*
- Andrew Regenscheid, *NIST*
- John Sabo, *Computer Associates*
- Anil Saldhana, *Red Hat*
- Krishna Sankar, *Cisco Systems*
- Frank Siebenlist, *Argonne National Laboratory*
- Sean Smith, *Dartmouth College*
- Jon Solworth, *Univ. of Illinois - Chicago*
- Anna Squicciarini, *Penn State*
- Von Welch, *NCSA*
- Stephen Whitlock, *Boeing*
- Michael Wiener, *Cryptographic Clarity*

Thank You!

Special Thanks

- Steering Committee
Chair: Neal McBurnett, Internet2
- Local Arrangements Chair - Sara Caswell, NIST
- Registration - Teresa Vicente, NIST
- Panels Chair - Radia Perlman, Sun Microsystems
- General Chair - Ken Klingenstein, Internet2

Technical Program

- Technical Paper sessions (peer reviewed)
 - Accepted 10 out of 30 total submissions
 - Each paper received 4 reviews on average
 - Some papers received shepherding
 - Thank you authors and PC members
 - Published in the ACM Digital Library as part of the ACM International Conference Proceedings Series
- Keynote talks
 - Dan Blum, Burton Group
 - Peter Neumann, SRI
- Panels sessions (6)

RUMP Session

- Short Work-In-Progress Talks
 - Thursday morning
 - Submit an abstract
 - 5 minute presentations (subject to change)
 - Contact: Neal McBurnett
 - neal@mcburnett.org



Courtesy: http://www.flaminghotideas.co.uk/library_travel.htm

Social Gathering and Dinner Buffet

- Tuesday, Gaithersburg Holiday Inn, 6 PM



Last Minute Instructions - Speakers

- Speakers please contact your session chairs in advance
 - At the beginning of the break before your session
- An electronic copy of each presentation should be given to Neal for the web site

Looking to the Future

- Please make plans now to submit a technical paper for next year
 - Submission deadline will be in the fall (October)
- Complete a survey at the conclusion of the workshop – your feedback is important to us!

Enjoy the Workshop

- The success of the workshop is in your hands
 - Participate!
 - Ask compelling questions





U.S. Federal Authentication and Identity Programs are Making Progress and Impacting Industry, But Much Work Remains

Dan Blum
April 14, 2009
dblum@burtongroup.com

Presented for NIST IDTrust 2009

U.S. Federal Authentication Programs



Agenda

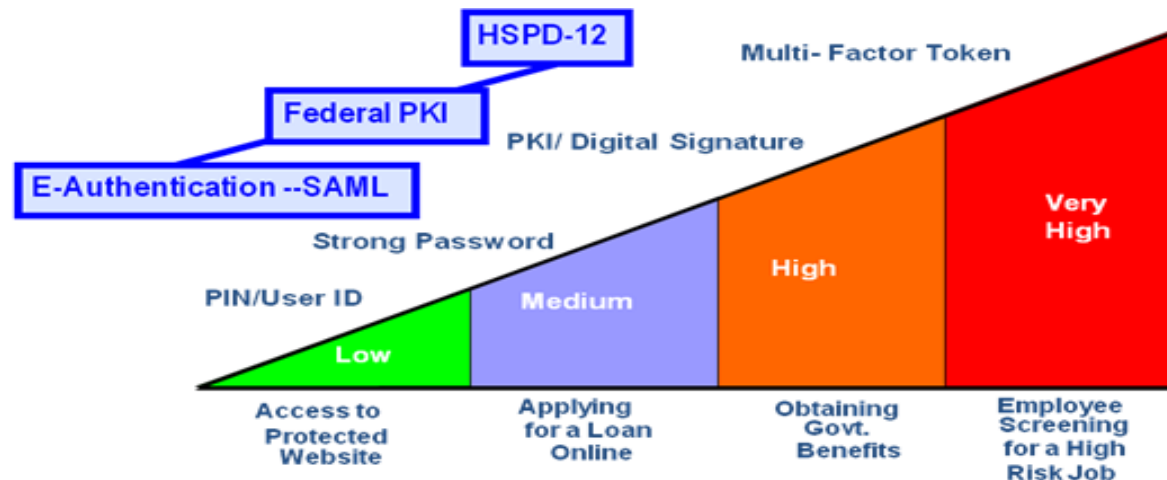
- *Overview of HSPD-12, FPKI, and E-Authentication*
- Taking the next steps in federated identity
- Recommendations

U.S. Federal Authentication Programs



Three main federal authentication initiatives

- Cross-certified federal public key infrastructure (FPKI) bridge is starting to gain traction through agency and shared service provider support
- E-Authentication Initiative for common policy, federated identity
- Homeland Security Presidential Directive 12 (HSPD-12) mandated Personal Identification Verification (PIV) cards for government employees and contractors



Source: GSA

U.S. Federal Authentication Programs



Multiple missions for multi-level, interoperable authentication

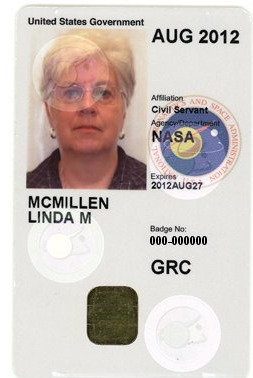
- Protect government information and facilities (cybersecurity)
- Improve internal efficiency and effectiveness
- Extend e-government services to citizens with protection appropriate to the risks involved

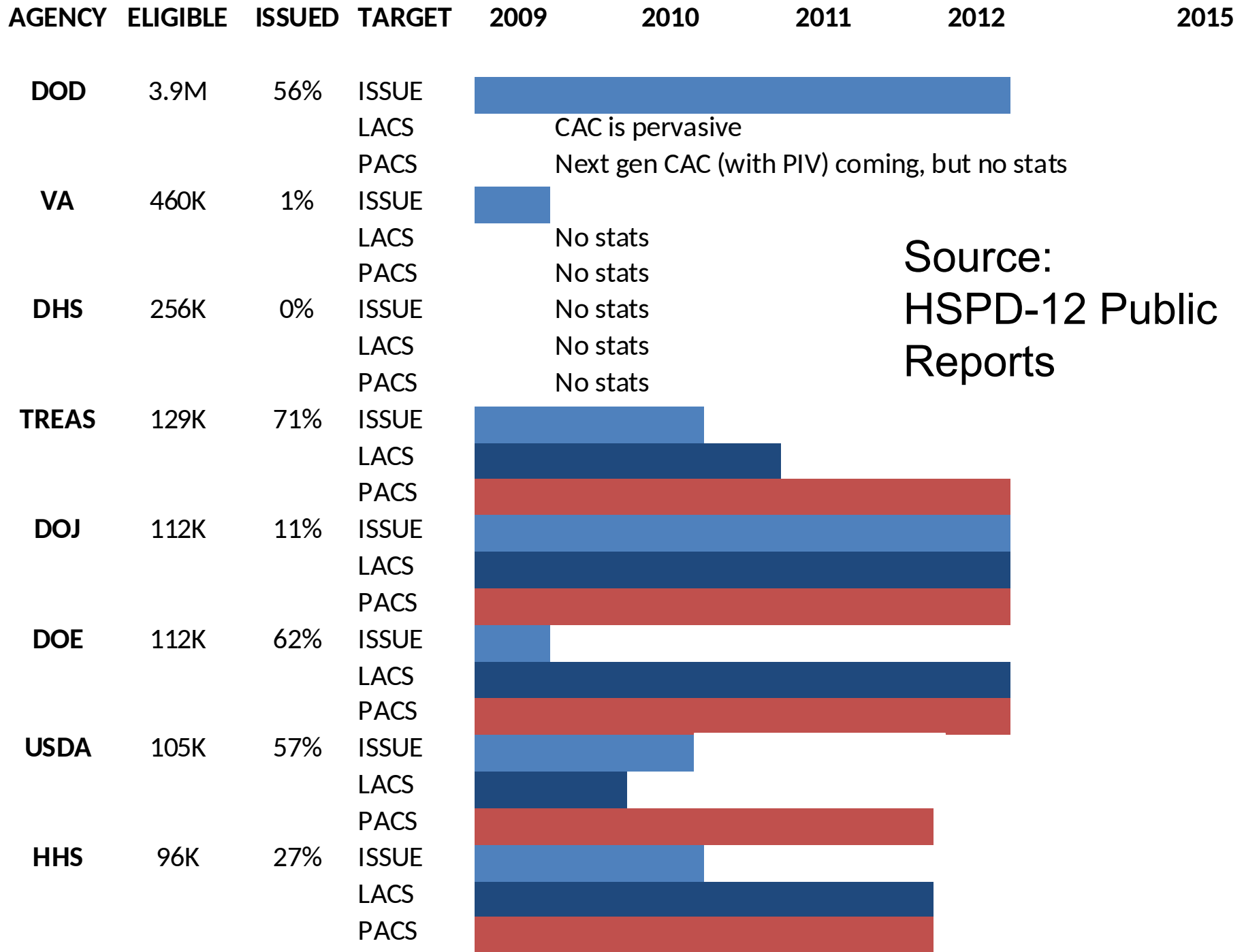
U.S. Federal Authentication Programs



Smartcards

- Card issuance continues
 - As of March 2009, 48% of the PIV-eligible workforce have been issued cards
- Emphasis is shifting from issuance to putting the PIV smartcard to use
- PIV card usage and deployment challenges include
 - Integrating card issuance with back-end directory and provisioning functions
 - Integrating smartcards with desktops, applications, and building access
 - Life cycle card management

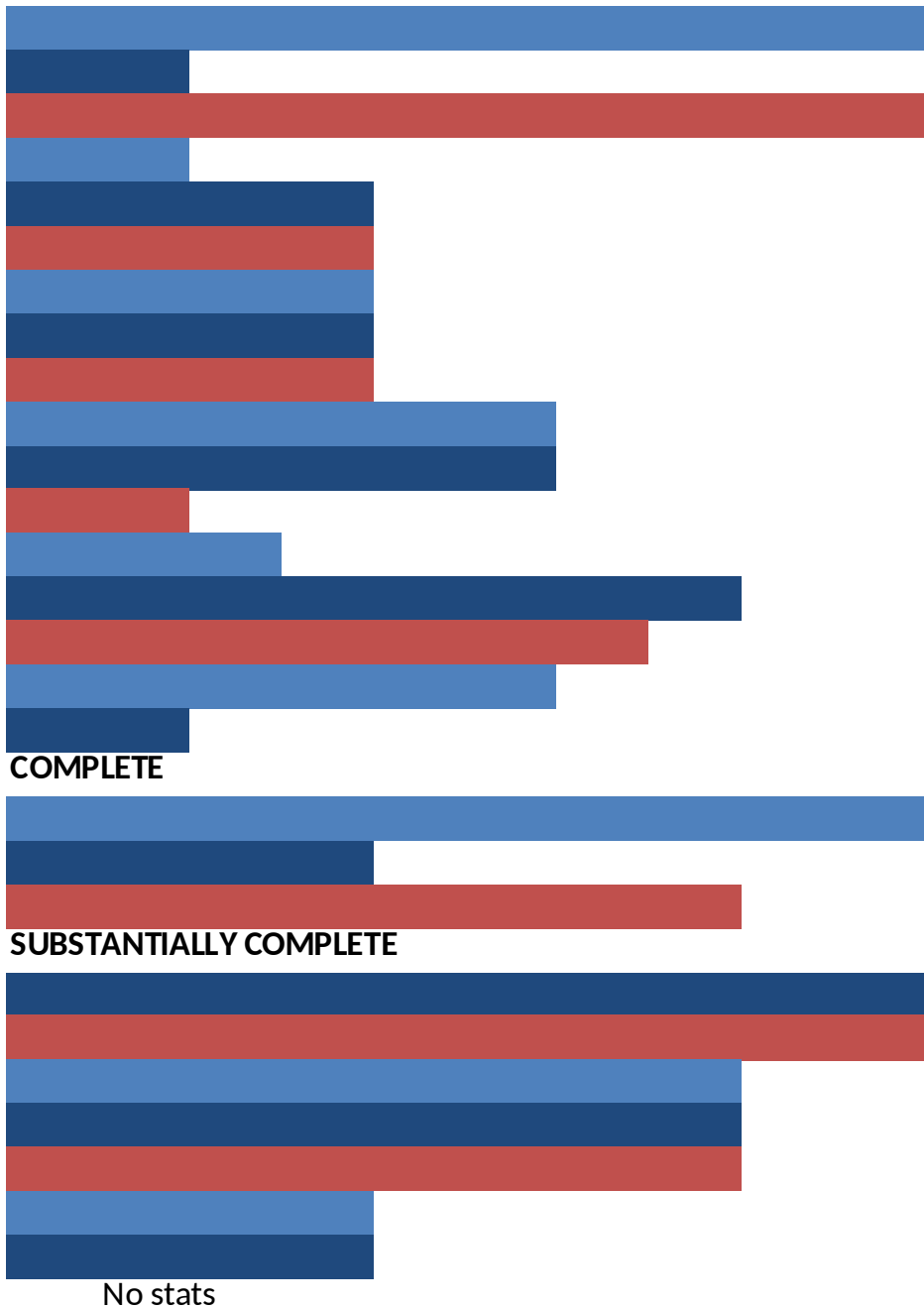




Source:
HSPD-12 Public
Reports

AGENCY	ELIGIBLE	ISSUED	TARGET	2009	2010	2011	2012	2015
--------	----------	--------	--------	------	------	------	------	------

DOT	92K	13%	ISSUE
			LACS
			PACS
DOI	83K	26%	ISSUE
			LACS
			PACS
SSA	81K	92%	ISSUE
			LACS
			PACS
NASA	78K	92%	ISSUE
			LACS
			PACS
DOC	49K	38%	ISSUE
			LACS
			PACS
STATE	27K	89%	ISSUE
			LACS
			PACS
GSA	22K	71%	ISSUE
			LACS
			PACS
EPA	19K	91%	ISSUE
			LACS
			PACS
DOL	18K	89%	ISSUE
			LACS
			PACS
HUD	11K	100%	ISSUE
			LACS
			PACS



COMPLETE

SUBSTANTIALLY COMPLETE

No stats

U.S. Federal Authentication Programs



Other large federal smartcard initiatives

- The Department of Defense (DoD) Common Access Cards (CAC) program is migrating towards FIPS 201 compliance
- Department of Homeland Security (DHS) programs will ultimately outfit populations that dwarf PIV's
 - Transportation Workers Identification Cards (TWIC) have been issued to more than 1 million workers
 - First Responder Authentication Credential (FRAC) cards will be issued by states and local governments as well as accredited service providers
 - Airport Credential Interoperability Solution (ACIS) is in the planning stages
- While other programs are not 100% PIV-interoperable, newer specifications are converging

U.S. Federal Authentication Programs



The significance of all the smartcard work...

"Although there are 6 million probes of Defense Department networks a day, successful intrusions have declined 46 percent in the past year because of a requirement that all DoD personnel log on to unclassified networks using Common Access Cards."

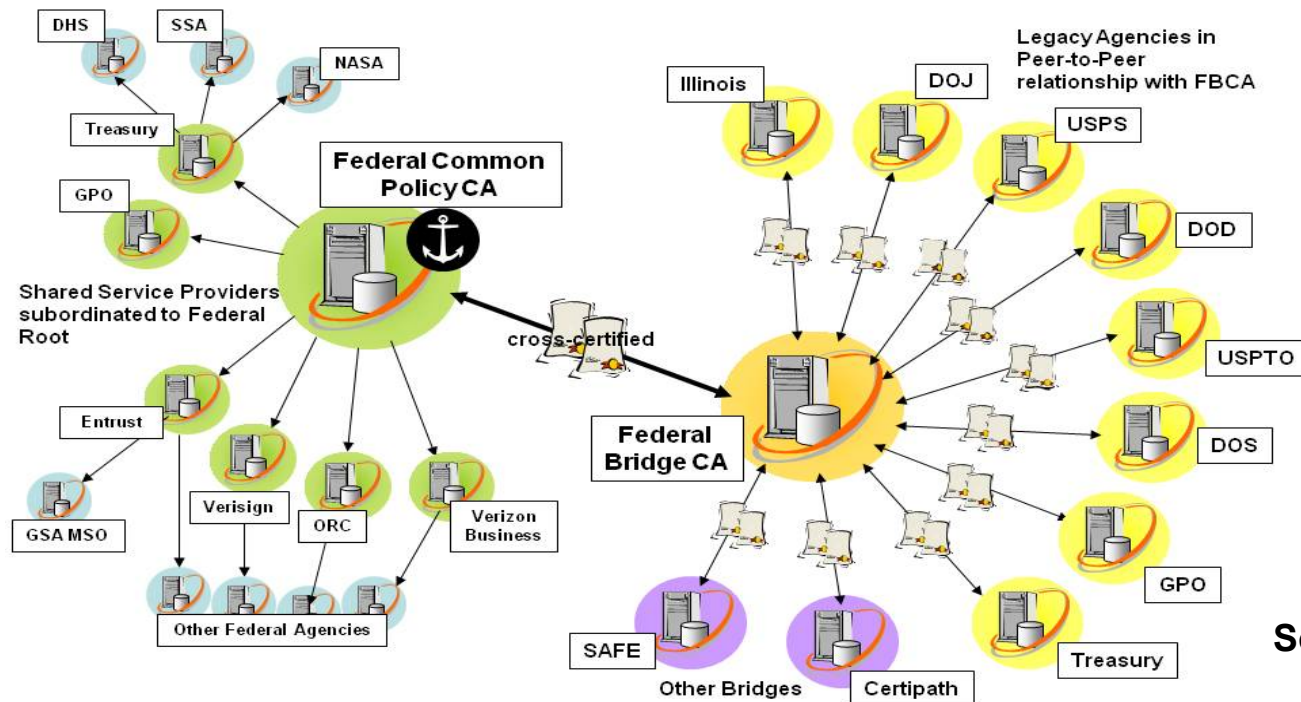
Lt. Gen. Charles Croom, at the AFCEA SpaceComm 2007 Conference

U.S. Federal Authentication Programs



Federal PKI

- FPKI is starting to gain traction through agency and shared service providers support
- Certification Authorities (CAs) from CertiPath, SAFE BioPharma, and Internet 2 (for higher education) are in place



Source: GSA



U.S. Federal Authentication Programs

FPKI: The evolution continues

- Revocation and path validation
 - CRL -> Client-side OCSP support
 - Client side path validation -> infrastructure path validation
 - OCSP -> SCVP
- Risk aggregation (in the breadth of trust, and concentration of CAs)
 - Migrate to stronger algorithms, longer keys over time

Related issues: Authorization and audit compensate for the breadth of the authenticable population

- Authentication is NOT authorization
- Beef up authorization, entitlements management
- Audit: the most important complementary and compensating control



U.S. Federal Authentication Programs

Organizational questions

ICAM: Identity, Credentials, and Access Management

- Subcommittee reports up to Federal CIO council has jurisdiction over smartcard, PKI, and federation programs
- OMB, GSA, NIST coordinate fairly closely
- ICAM also coordinates with DHS, defense and intelligence community

Hopes for authentication continuity with the new administration

- ICAM, agency community generally optimistic that there will be continuity on cybersecurity matters, such as HSPD-12 and FPKI
- Tension anticipated between cybersecurity, e-government, and social media camps

U.S. Federal Authentication Programs



Federated identity

- E-Authentication Initiative successful on a policy level
- Federated identity is in place at multiple agencies and is used for
 - Many Level 1 and 2 government applications
 - Reducing identity management silos
 - Reducing sign-on requirements
- Business and citizen-facing government applications didn't take off
- GSA has shut down the E-Authentication Portal and now encourages agencies to deploy locally using E-Authentication guidance



U.S. Federal Authentication Programs

Industry impact: a large market for products and services

- Specifications for PIV-interoperable smartcards will soon be formally released, providing standards for commercial service providers, federal contractors, states, municipalities, and others
- DoD/government contracting practices and regulations may come to require PIV support
- GSA has approved more than 400 products for various FIPS 201 smartcard-related functions, and 18 products for SAML
- Higher education is in the forefront of federating with NIH and other agencies for research and grant-based applications
- Federations for law enforcement and medical information sharing are in pilot stages at DHS, DOJ, VA

U.S. Federal Authentication Programs



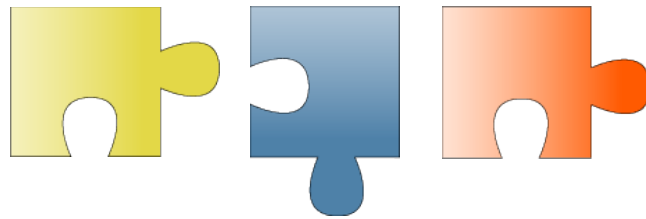
Agenda

- Overview of HSPD-12, FPKI, and E-Authentication
- ***Taking the next steps in federated identity***
- Recommendations

U.S. Federal Authentication Programs



The puzzle of how to create identity interoperability for the masses remains unsolved...

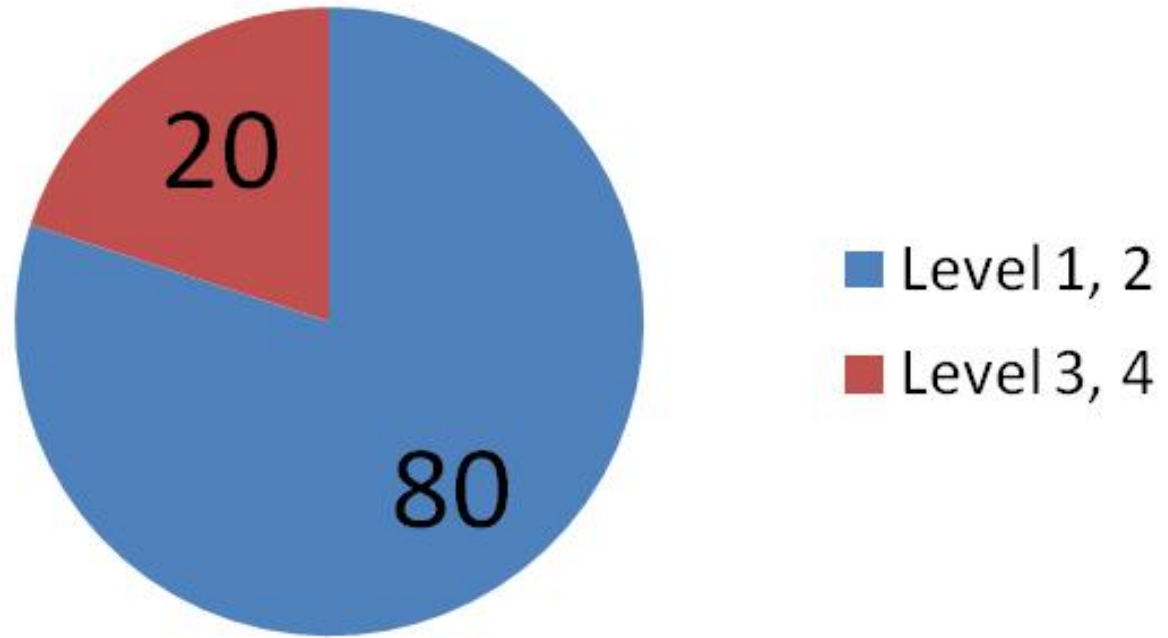


U.S. Federal Authentication Programs



Where does federated identity apply?

A survey once said applications were...



But is that correct?

The answer depends not just on risk, but also on the mix of compensating controls...

U.S. Federal Authentication Programs



Change we can believe in?

New user-centric schemes for identity interoperability

- Information Cards have promising implementations, but few sites support them
- OpenID is popular but has trust, security, and usability problems



"Click-in", not login!



U.S. Federal Authentication Programs



Smartphones with OTP software

- Reported in the New York Times March 31st
- A strong authenticator in every pocket
- Planned for BlackBerry too



VeriSign

U.S. Federal Authentication Programs



In our humble opinion...

- The user is not always center
- The organization is not always center
- Interoperability takes on the context of relationships
- Use cases exist for
 - 1st party (user centric) federation
 - 2nd party (e.g. today's typical SAML) federation
 - 3rd party (e.g. SAML, WS-*) federation



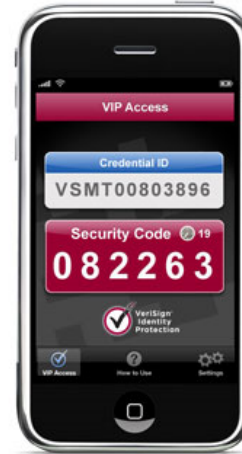


U.S. Federal Authentication Programs

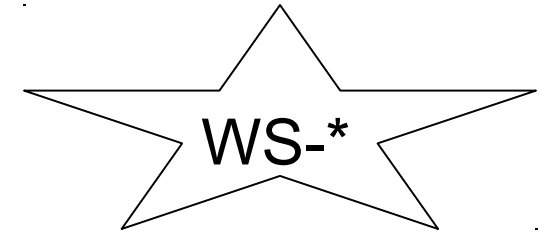
They may ALL make sense somewhere...



SAML



VeriSign



- Pick the technologies that are simplest and most apt to the problem
- And find business models, partnerships to solve human and organizational problems

U.S. Federal Authentication Programs



Agenda

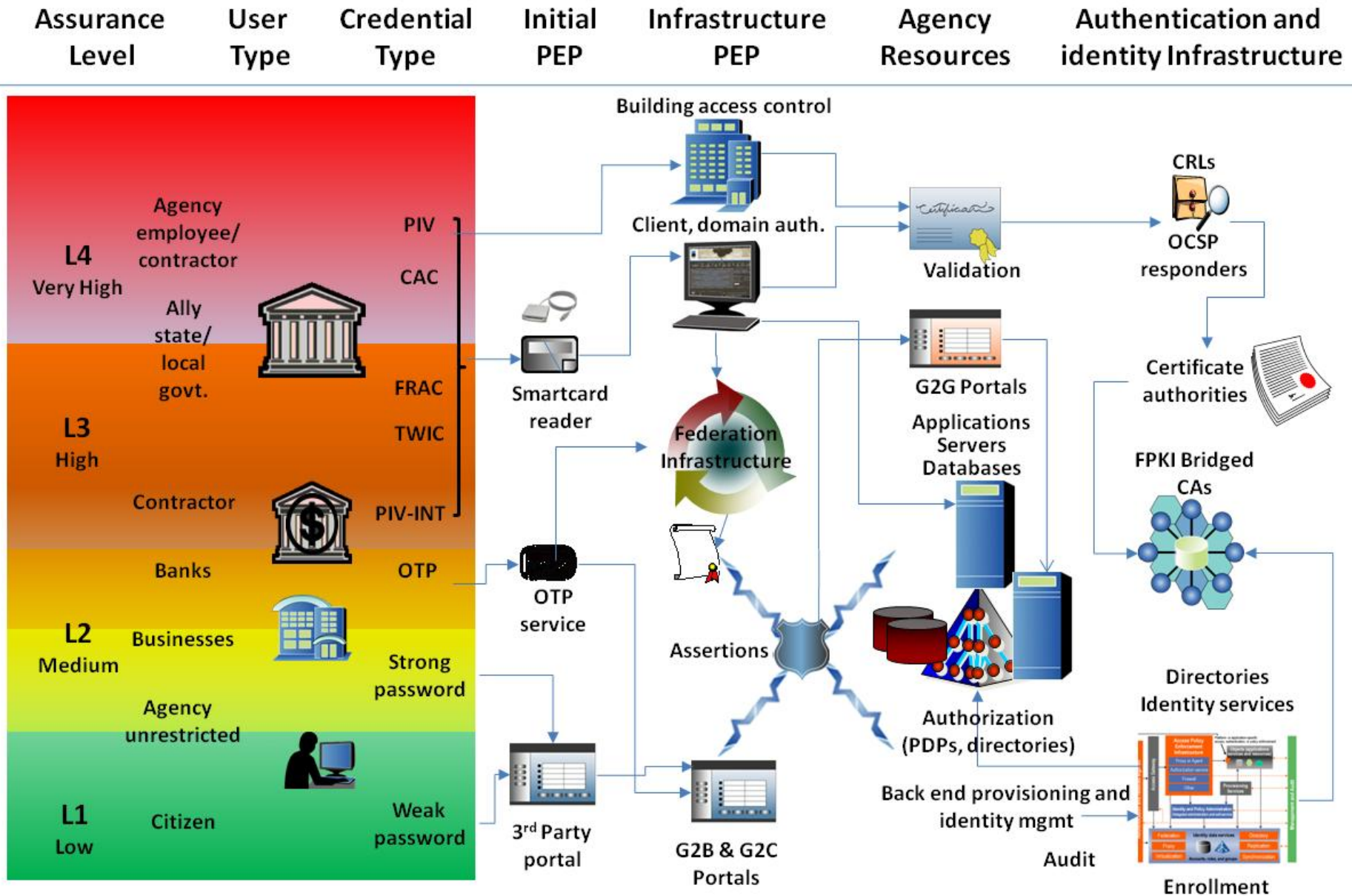
- Overview of HSPD-12, FPKI, and E-Authentication
- Taking the next steps in federated identity
- **Recommendations**

U.S. Federal Authentication Programs



Recommendations for agencies

- Put newly issued PIV cards to use as required per HSPD-12
- Build consensus on overall agency architecture (e.g., see next slide) among physical facilities, information security, and other groups
- Integrate smartcard management and issuance with agency identity management and provisioning systems
- Integrate smartcard login with client operating systems
- Dovetail PIV rollout with desktop and physical access control system upgrades
- Implement robust recovery and emergency access procedures
- Enable applications to consume PIV through reduced sign-on approaches (assertions, web access management, Kerberos...)
- Enhance identity federation and authorization capabilities



KEY:

BIO: Biometric	FPKI: Federal public key infrastructure	PEP: Policy enforcement point
CAC: Common Access Card	G2C: Government-to-citizen;	PDP: Policy decision point
CRLs: Certificate revocation lists	OCSP: Online certificate status protocol	PIV: Personal identity verification
FRAC: First responder authentication credential	OTP: One time password (token)	PIV-INT: PIV-Interoperable
G2B: Government-to-business	PIN: Personal identification number	TWIC: Transportation Workers Identity Card

U.S. Federal Authentication Programs



Recommendations for federal authentication governance

- Enhance policy and implementation guidance to agencies
- Strongly promote federated identity for externally facing e-government applications at assurance levels 1, 2, and 3
 - Stay the course with SAML 2.0
 - Promote additional federation “schemes” where advantageous
- Continue outreach programs to gain industry consensus on federated and user-centric identity

U.S. Federal Authentication Programs



Recommendations for other organizations in the industry

Study Federal authentication initiatives for opportunities to

- Gain competitive advantage
- Comply with future DoD or civilian agency contract stipulations
- Learn how to improve their own IT and security infrastructure

References



General references

- Government's identity management site
- HSPD-12 Public Reports
- Technical Comparison: OpenID and SAML - Draft 06

Burton Group documents

- U.S. Federal Authentication and Identity Programs are Making Progress and Impacting Industry, But Much Work Remains
- Let's Get Logical: The Convergence of Physical Access Control and Identity Systems
- Entitlement Management: Ready to Enter the IdM Mainstream
- Federation's Future in the Balance: Teetering Between Ubiquity and Mediocrity
- The Information Card Landscape
- In Search of the Internet Identity System: Contrasting the Federation Approaches of SAML, WS-Trust, and OpenID
- A Relationship Layer for the Web . . . and for Enterprises, Too
- Web Access Management Market 2007: Expanding Boundaries
- Federation Products 2008

U.S. Federal Authentication Programs



Conclusion

- The U.S. federal authentication programs for issuing smartcards and extending PKI are gaining traction
- Government and industry now have a roadmap for high assurance authentication interoperability
- Agencies must now enhance identity management and security infrastructure to leverage strong authentication for logical and physical access control
- Meanwhile, federated identity schemes could support many e-government applications
 - E-Authentication identity federation initiatives must be revitalized to build public/private and federal/state partnerships

Session 1 Panel: Comparative Identity Systems

Panelists:

George Fletcher, *AOL*

Ken Klingenstein, *Internet2*

Radia Perlman, *Sun*

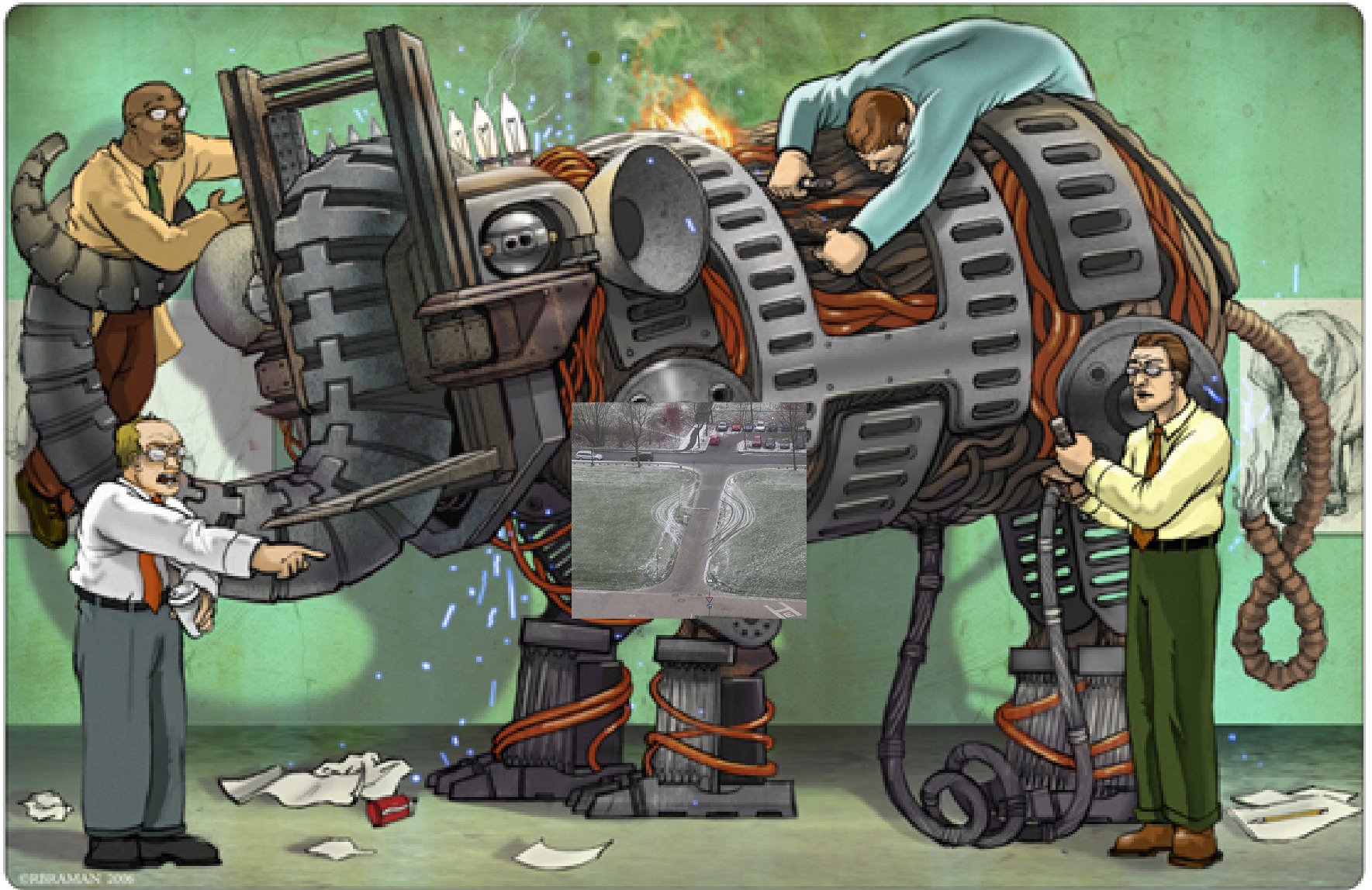
Paul Trevithick, *Higgins Project*

Moderator: Kent Seamons, *Brigham Young University*



April 14-16, 2009

IDtrust 2009



April 14-16, 2009

IDtrust 2009



April 14-16, 2009

IDtrust 2009

PKI-based authentication

Radia Perlman
Radia.Pperlman@sun.com

I don't care about formats

- “Certificate” is a signed thing, asserting by some trusted entity, a mapping between things such as a name and a key

PKI-based Authentication

Alice

Bob

→ ["Alice", key=342872]CA

← ["Bob", key=8294781]CA

← Auth, encryption, etc. →

Yes there are issues

- How does a user get her private key?
- How does a user know the CA's public key?
- How does a user get a certificate?
- Revocation..

Within an organization

- Should be trivial, single CA
- To create an account
 - Sysadmin told username and initial pwd
 - Types that into “create new account” tool
 - Tool generates key pair, certifies public key, encrypts private key with pwd, stores cert in dir
- User logs in
 - Types name and pwd, retrieves private key
 - Accesses resource: authenticates with public key

Better with smart cards

- Badges have smart cards. What's the problem?

How about individuals?

- Think of this as just doing what we do with username/pwd, but more securely, and without torturing the user
- Assume first the user has a smart card with a secret (private key, or secret key)

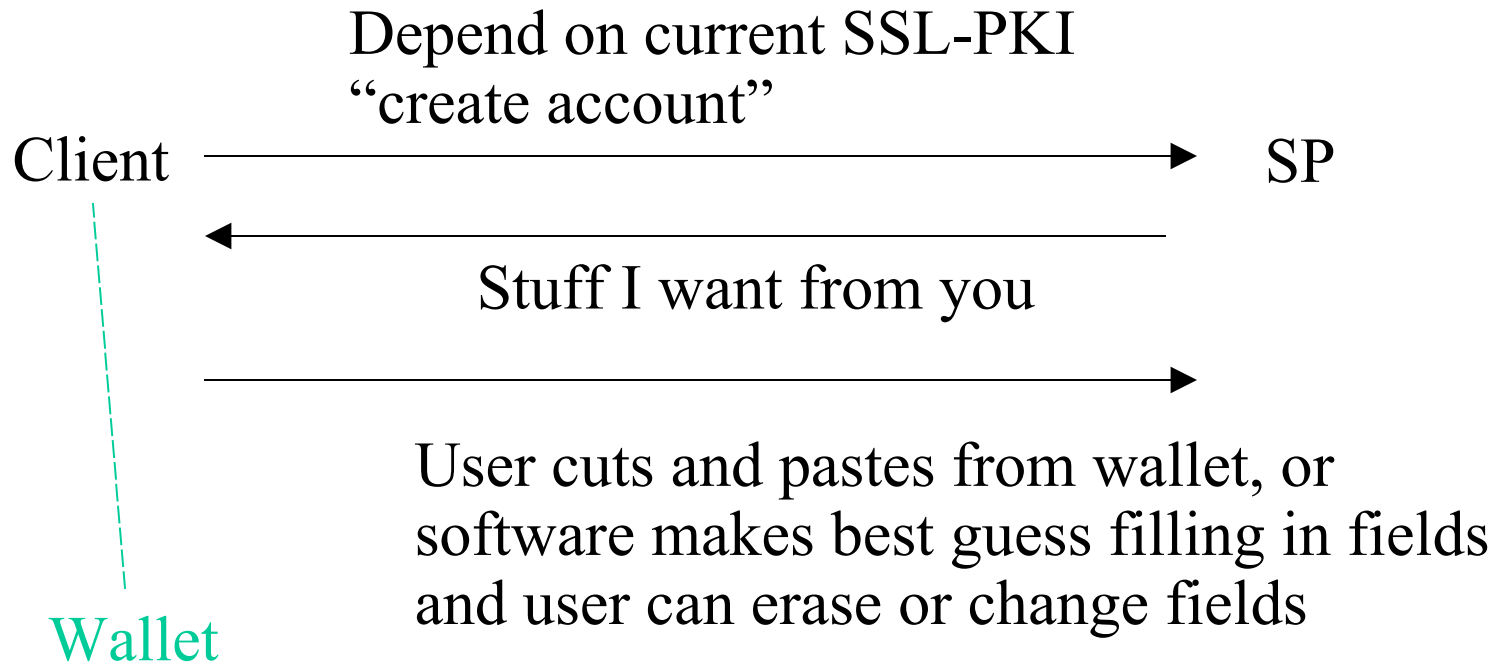
“Wallet”

- A bunch of data cryptographically protected with the user’s smart card secret
- Downloadable from one or more places
- Contains, for instance, public keys of various merchants, perhaps private keys to use with that merchant, information such as passport number and credit card numbers

Enrolling at a site

- Just like today, except username/pwd is replaced by “public key”
- The wallet information (such as address) can be filled into the form, to save the user typing, or the user could drag info she wants into the form
- The SP sends the user its public key

Enrolling



Wallet

{addresses}

{credit cards}

{telephone numbers}

Passport number

Per site info (its public key, your key pair for that site)

Note

- Instead of enrolling with a username and password, your account name is your public key, and you authenticate with your private key
- And by saving the SP's public key (a la SSH), you can do mutual authentication, knowing you are again reaching the same site as before

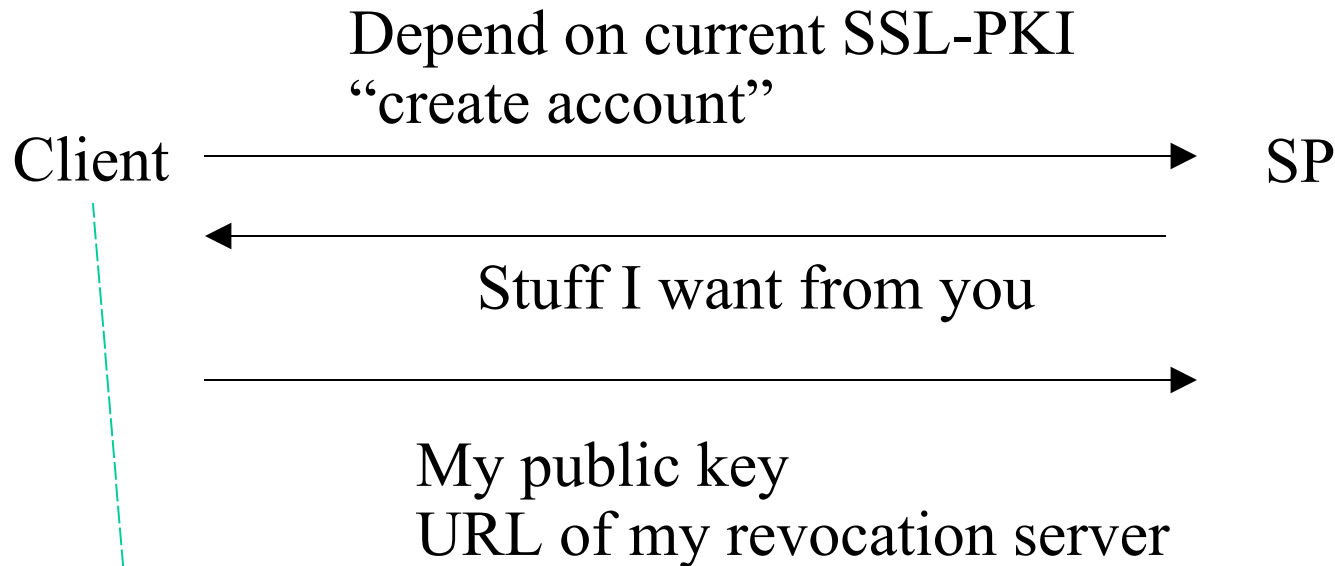
Revisiting the site

- Mutual authentication using public keys (e.g., SSL with client certs)

One-step revocation

- Suppose you are using your public key at lots of sites
 - (not sure how useful different keys for each site is)
- And someone steals it
- Use “revocation service”

Enrolling



Wallet

- {addresses}
- {credit cards}
- {telephone numbers}
- Passport number
- Per site info (its public key, your key pair for that site)

Revocation service

- SP learns user's revocation server along with the user's public key
- SP can “enroll” with that revocation service, to be notified in case of revocation
- Or SP can check periodically
- User has to have some sort of out-of-band mechanism to authenticate and revoke the key
- User can store {next keys} signed by current key, and escrow the future private keys

Authenticated attributes

- User can have, in wallet, certs signed by whoever is trusted to assert the attribute, that a public key associated with the user is over 18, a citizen, whatever
- Can send such certs to SP when needed, along with proof of knowledge of the private key

Yes, things can go wrong

- Establish trust, then after increasingly large purchases, skip town
- Credit cards today somehow work “well enough” – certainly could be improved, but banks seem to think it’s not worth the bother

My view of federated things

- Microsoft created the “Passport” vision, with Microsoft the center of the world
- Others said, “Hey, let’s not anoint one organization to be an eternal monopoly
- So, the notion of lots of IDPs, and a federation is the set of SPs that trust that IDP

If there is just one IDP

- User authenticates to that IDP
- That IDP vouches for the user at all the affiliated sites

But what if there are hundreds?

- And what if the SPs the user wants to use affiliate with different subsets of them?

And what value does the IDP give,
anyway?

Online vs offline trusted box

- Online box solution less secure
 - Can impersonate all users
 - More likely to be compromised than an offline box
 - Knows who is talking to who
 - May have database that if stolen, can compromise users

Also

- More expensive (must be high performance, replicated for availability and performance)
- Less robust (more boxes have to be up)
- Slower (have to talk to 3rd box before Alice can talk to Bob)

Online intermediaries vs PKI

- Performance
- Availability
- Security
- Privacy

**Federated Identity:
It's the attributes,
urn:mace:incommon:entitlement:clue:zero**

Ken Klingenstein
Internet2

Topics

- Federated identity basics
- Integration, not differentiation
- Attributes

Internet identity

- Federated identity
 - Enterprise centric, exponentially growing, privacy preserving, rich attribute mechanisms
 - Requires lawyers, infrastructure, etc
- User centric identity
 - P2P, rapidly growing, light-weight
 - Marketplace is fractured; products are getting heavier to deal with privacy, attributes, etc.
- Unifying layers emerging – Cardspace, Higgins, OAuth

Federated identity

- Convergence around SAML 2.0 – even MS
- Exponential growth in national and international R&E sectors
- Emerging verticals in the automobile industry, real-estate, government, medical
- Policy convergence for LOA, basic attributes (eduPerson), but much else, including interfederation and the user experience, remains to be developed
- Application use growing rapidly
- Visibility is about to increase significantly through end-user interactions with a privacy manager

Trust, Identity and the Internet

- ISOC initiative to introduce trust and identity-leveraged capabilities to many RFC's and protocols
- Acknowledges the assumptions of the original protocols about the fine nature of our friends on the Internet and the subsequent realities
- <http://www.isoc.org/isoc/mission/initiative/trust.shtml>
- First target area is DKIM; subsequent targets include SIP, federated calendaring and sharing, firewall traversal

Federation Update

- R&E federations sprouting at national, state, regional, university system, medical and library alliances, and elsewhere
- Federated identity growing in business
 - Many bilateral outsourced relationships
 - Hub and spoke
 - Multilateral relationships growing in some verticals

Federating Software

- Move from bilateral to multilateral critical for scaling, and thus making metadata standards much more important
 - Metadata can include signing keys, attribute release policies, attribute consumption policies, DKIM signing keys and so much more....
- Shibboleth does this; vendor SAML products are starting to consume metadata better
- MS Geneva will be configurable for InCommon, done right

R&E Federation Killer Apps

- Content access – Elsevier, OCLC, JSTOR, iTunes
- Government access – NIH, NSF and research.gov
- Access to collaboration tools – wikis, moodle, drupal, foodle; soon federated calendaring
- Roaming network access
- Outsourced services – National Student Clearing House, student travel, plagiarism testing, travel accounting
- MS Dreamspark
- Google Apps for Education

International R&E federations

- More than 25 national federations
- Several countries at 100% coverage, including Norway, Switzerland, Finland; communities served varies somewhat by country, but all are multi-application and include HE
- UK intends a single federation for HE and Further Education ~ tens of millions of users
- EU-wide identity effort now rolling out - IDABC and the Stork Project (www.eid-stork.eu)
- Key issues around EU Privacy and the EPTID
- Some early interfederation – Kalmar Union and US-UK

InCommon

- Over 150 members now
- Almost three million “users”
- Most of the major research institutions
- Other types of members
 - Non usual suspects – Lafayette, NITEL, Univ of Mary Washington, etc.
 - National Institute of Health, NSF and research.gov
 - Energy Labs, ESnet, TeraGrid
 - MS, Apple, Elsevier, etc.
 - Student service providers
- Growth is quite strong; doubled in size for the fifth year straight
- Silver profile approved

NIH

- Driving agency for much of our government activity
- Several types of applications, spanning two levels of LOA and a number of attributes
 - Wikis, access to genome databases, etc
 - CTSA
 - Electronic grants administration
- “Why should external users have internal NIH accounts?”
- Easier stuff – technology, clue at NIH, user interest
- Harder stuff – attributes (e.g. “organization”), dynamically supplied versus statically-supplied info

Principles to be established by InCommon futures process

- Community served
- Business models
- Service and business opportunities
- Governance and representation
- Pricing and packaging principles – membership models, working with soup, etc.
- -----
- The relationship between InCommon and Internet2

Federation Soup

- Within the US, federations happening in many ways – state, university system, library, regional, etc
- Until we do interfederation, and probably afterwards, federations will form among enterprises that need to collaborate, regardless of their sector
- Common issues include business models, legal models, LOA and attributes, sustainability of soup
- Overlapping memberships and policy differences creates lots of complexity in user experience, membership models, business models, etc.
- One workshop in, so far...
- <https://spaces.internet2.edu/display/FederationSoup/Home>

Interfederation

- Necessary for cross-vertical interactions, global scaling, etc.
- Technical issues are tractable – dynamic metadata and metadata tagging, user discovery, etc – as long as basics (LOA, attributes, etc) are kept consistent
- Policy issues are interesting – liability, adjudication, privacy, federation operator practices, etc.
- Liberty and R&E and ISOC to work together on it

The consumer marketplace

- For federated identity, it hasn't happened yet; existing IdP's service current application needs.
- GSA to engage with multiple federations of IdP's, including InCommon, for government access
- Several natural consumer IdP providers: banks, ISP's, governments...
- Costs are low; lock-in potential high

Integration

- Different forms of Internet identity will exist, serving different purposes, arising from different constituencies
- The trick is the intelligent integration of the technologies, at user and application level
- Cross-overs are happening
 - Shib and Openid
 - SAML and high assurance PKI – holder of key
 - Infocard/Higgins as an overarching user experience
 - Federation and portal integration

Unifying the application developer experience

- Discussions in IETF around OAuth could address API's
- Discussions in OASIS around Shib profiles
- Discussions with Quali/Rice about services
- Discussions with portal people about portals

Unifying the user experience

- Among various identity providers, including P2P, self-issued, federated
- Need to manage discovery, authentication, and attribute release
- Cardspace, Higgins, uApprove, etc.
- Consistent metaphors, different technical approaches
- Starting to deploy

Privacy management

- Two approaches emerging
 - uApprove
 - <http://www.switch.ch/aai/support/tools/uApprove.html>
 - InfoCard/Higgins
- Who sets attribute release policies? Who overrides the settings? What logs are kept?

This is the Digital ID Card to be sent to 'https://aai-demo.switch.ch':

Digital ID Card	
Surname	SWITCHaai
Given name	Demouser
Unique ID	234567@example.org
User ID	demouser
Home organization	example.org
Home organization type	other
Affiliation	staff
Entitlement	http://example.org/res/99999 http://publisher-xy.com/e-journals

Don't show me this page again. I agree that my Digital ID Card (possibly including more data than shown above) will be sent automatically in the future.

Cancel

Confirm

Attributes

- Now that we have a transport system, attention is turning to the cargo...
- Standard schema (e.g. eduPerson) have proven very valuable; new attribute needs are emerging (over legal age, citizenship, disabilities, etc.)
- Semantics need to be addressed more rigorously in a federated environment
- Workshops are beginning to look at these issues
- Attributes can conceal identity and preserve privacy, preserve secrecy, generate revenue

The Art of the attribute

- Proliferation of attributes – see http://wiki.idcommons.net/Identity_Schemas
- Attribute aggregation approaches are beginning
- No real understanding of sources of authority, delegation, audit, etc
- Mappings and other evils lurk
- All of which needs to work with humans as users, authorities, etc.

GSA Attribute Workshop

- Begin exploring the attribute issues
- Using federal use cases, including
 - Citizenship, voting residency
 - Access-abilities
 - First responder capabilities
 - PI-person
- Motivate the larger requirements, drive privacy policies
- Explore rich query languages, etc.
- All-star cast at the end of September at NIH

Information Cards

IDtrust 2009, NIST

Paul Trevithick

paul@azigo.com

Selector-based Model



- Each identity displayed as a card
- The selector is the wallet
- A selector on every device
- Cards can roam between devices
- Card issuer defines claim set
- Issuer defines auth method
- User authenticates to card (not to RP/SP)

Card Types



Managed

What some other entity says about you
(Manual Push)



Personal

What you say about you
(Manual Push)

Coming Soon

Relationship

What we say about you
(Automatic Pull & Push)

Multiple Token Types + Multi-Protocol*

Protocols

- InfoCard
- SAML & ID-WSF
- OpenID / AX
- Username/password (!)

Token Types

- SAML
- Kerberos
- Zero Knowledge Proofs
- Proprietary...

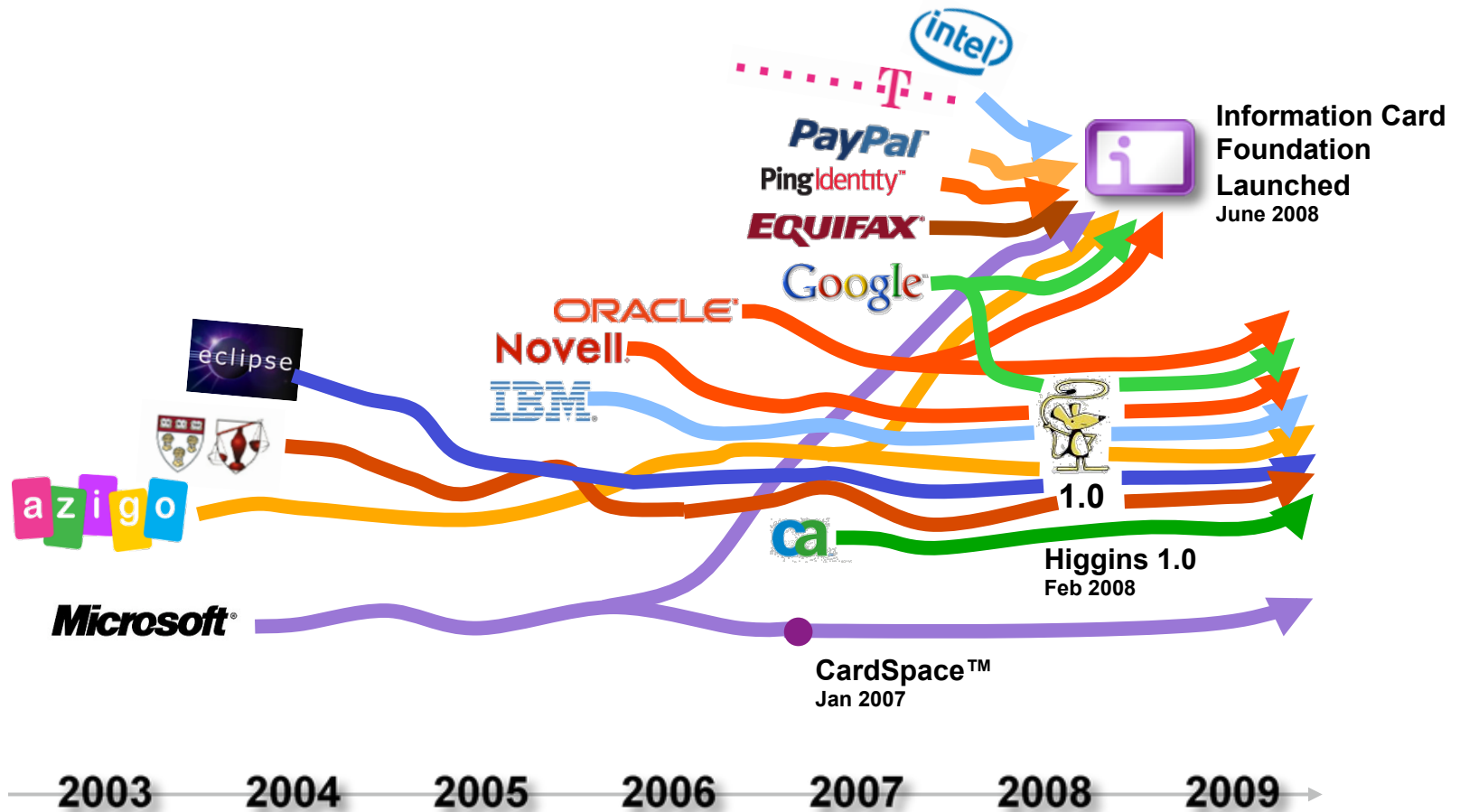
...all hidden behind a the same card metaphor

*Higgins developers are integrating SAML Circles of Trust, WS-Trust STS/IdPs, and ID-WSF

Selector Implementations

- Mac, Windows, Linux available now
- Mobile devices coming soon
- Open source and closed source
- Microsoft CardSpace™
- Azigo (Higgins-based, Adobe AIR-based)
- Novel DigitalMe™ (Higgins-based, native code)
- OpenInfoCard

Development Timeline



Community: <http://informationcard.net>**

ORACLE

Google

PayPal

Novell

■ ■ ■ **T** Deutsche
Telekom

intel

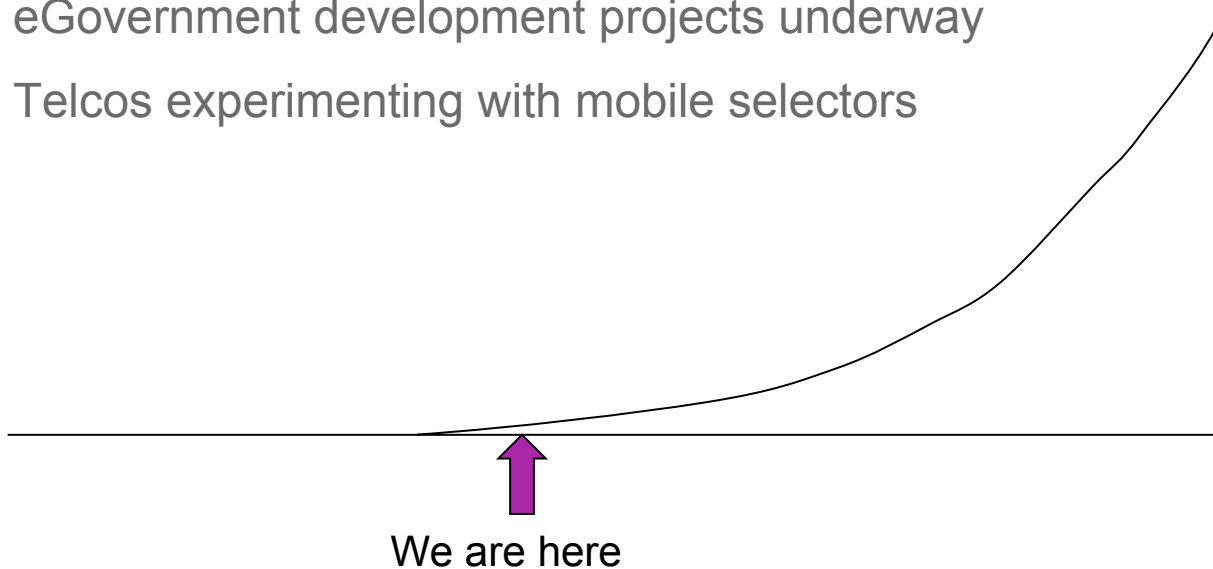
Microsoft

EQUIFAX

- Information Card Foundation (**ICF**) publicly launched in the New York Times June 24th 2008
- **New Website will launch at RSA 2009
- ICF Board member sponsors are shown at left
- Eclipse Higgins project is a collaboration led by Azigo and including IBM, Google, Oracle, Novell, CA and others

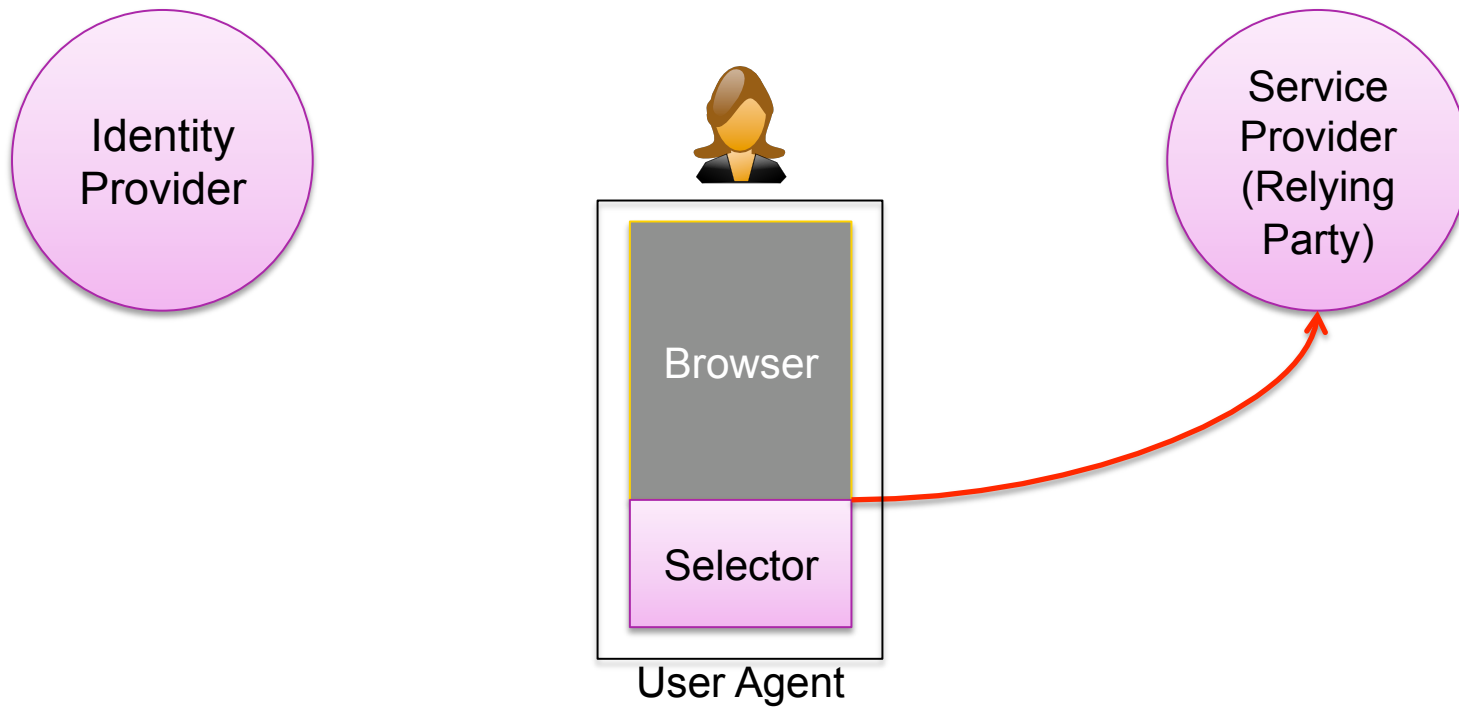
Adoption is just beginning

- Equifax Over 18 card
- Top 10 Website will demonstrate at RSA 2009
- AAA “RemindMe” discount/loyalty card launching in May
- eGovernment development projects underway
- Telcos experimenting with mobile selectors

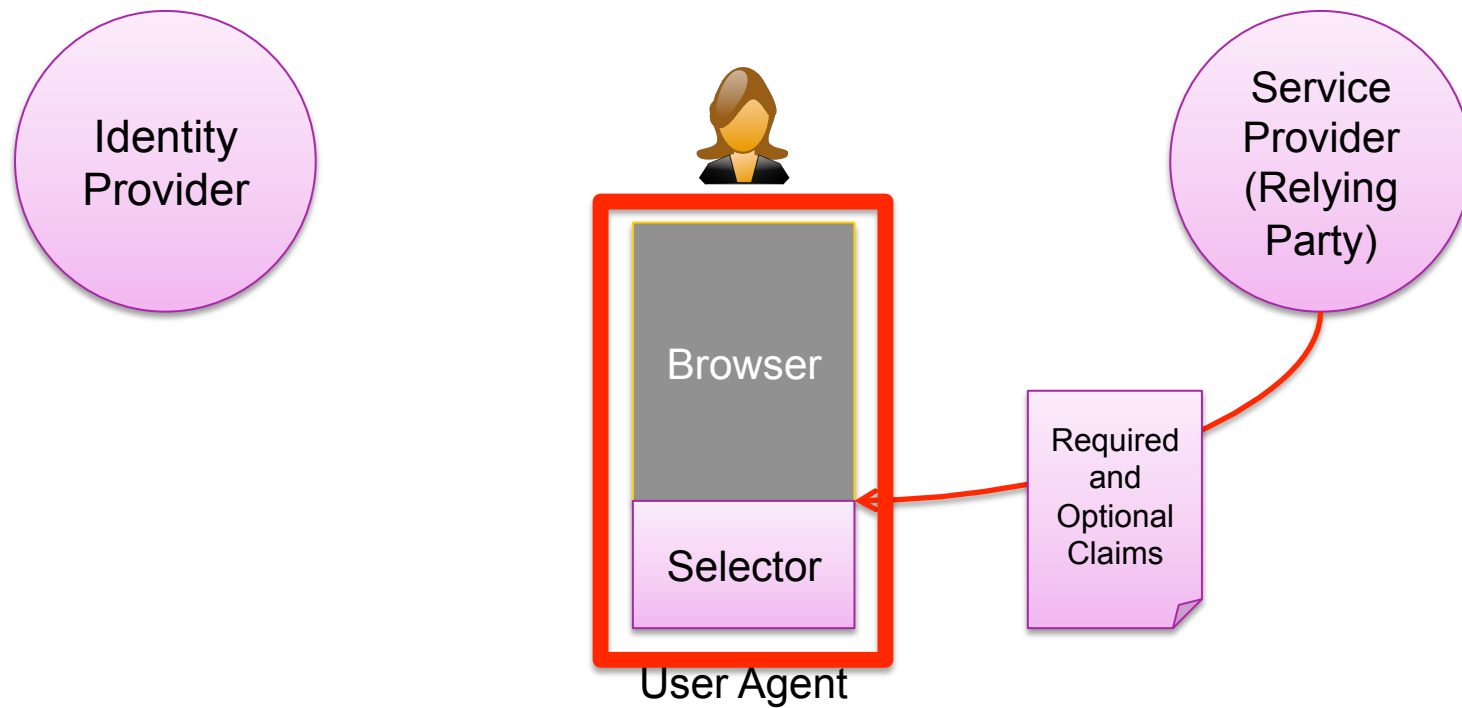


Appendix: Login Flow

1: Alice goes to site (or app)

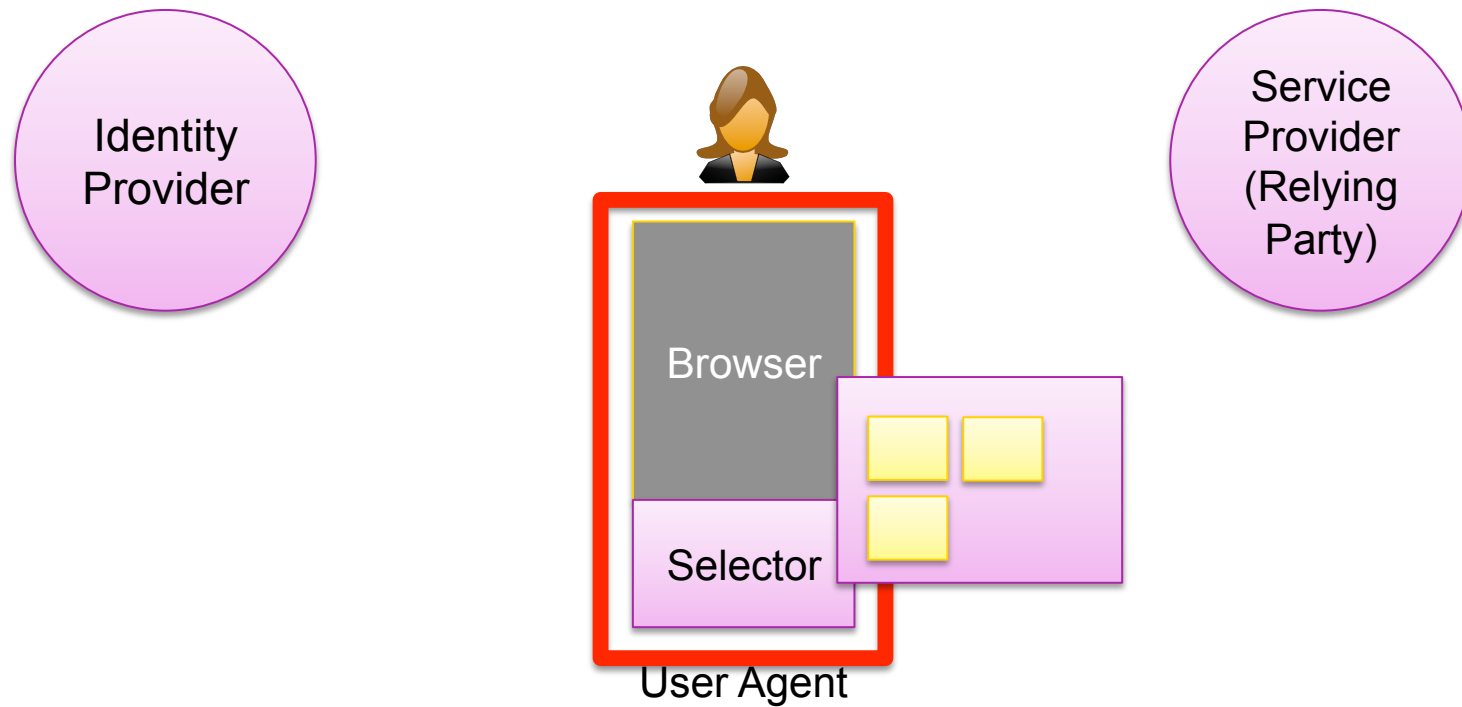


2: Selector retrieves policy

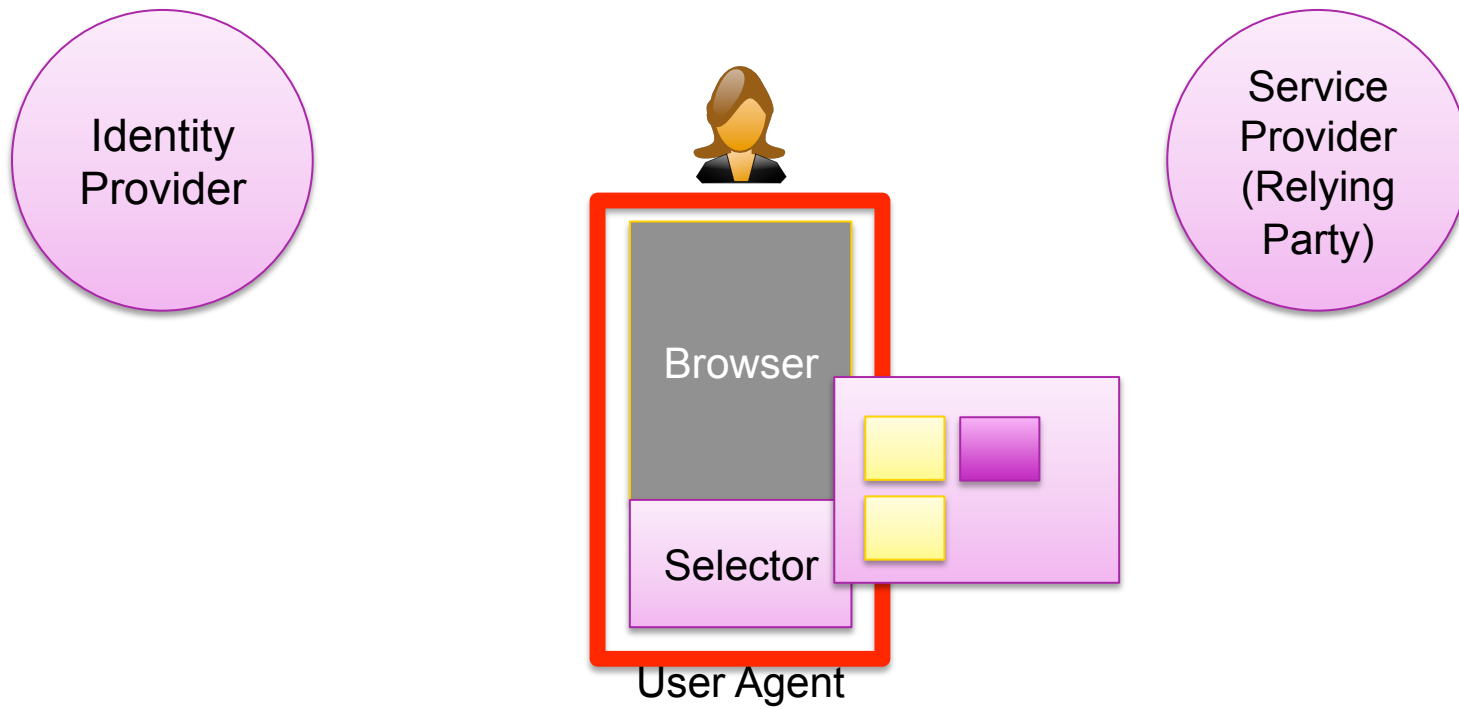


3: Display cards that match policy

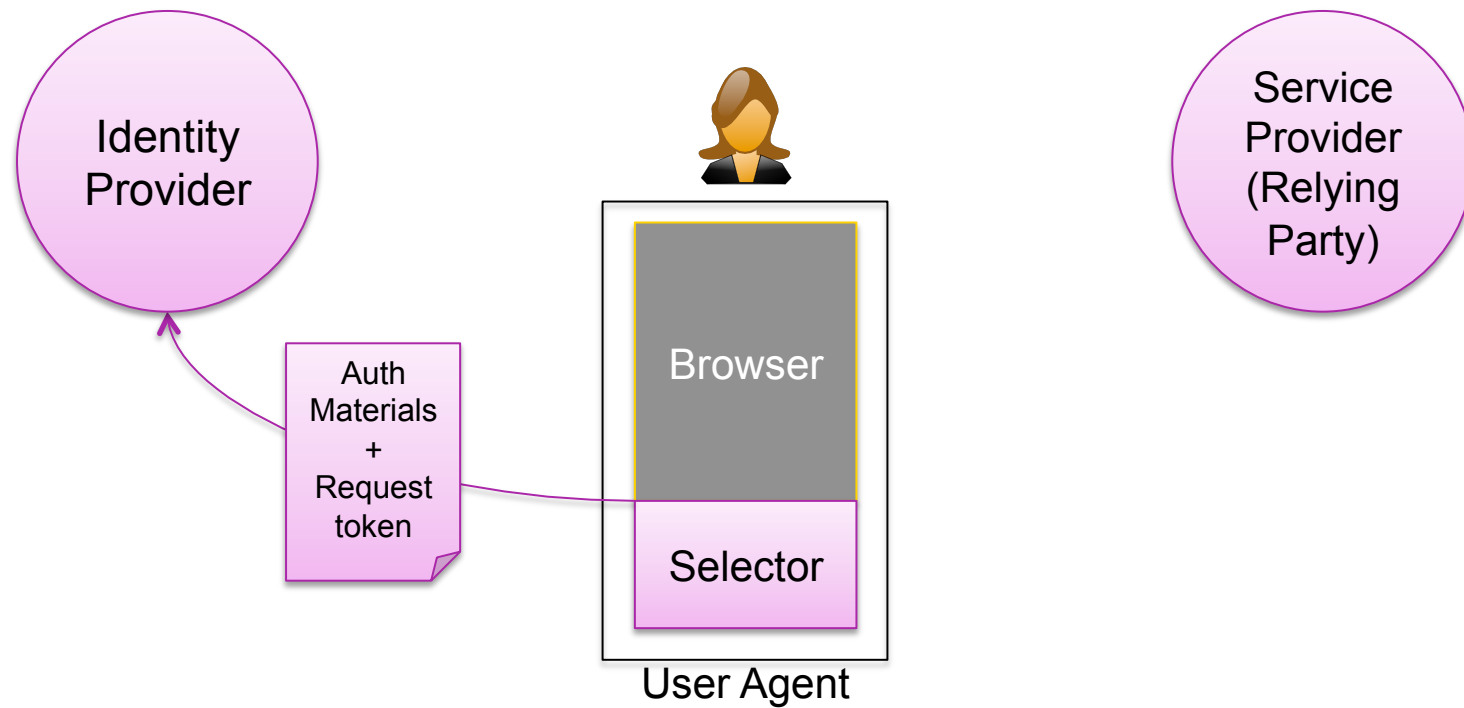
(each card points to a different IdP or is self-issued)



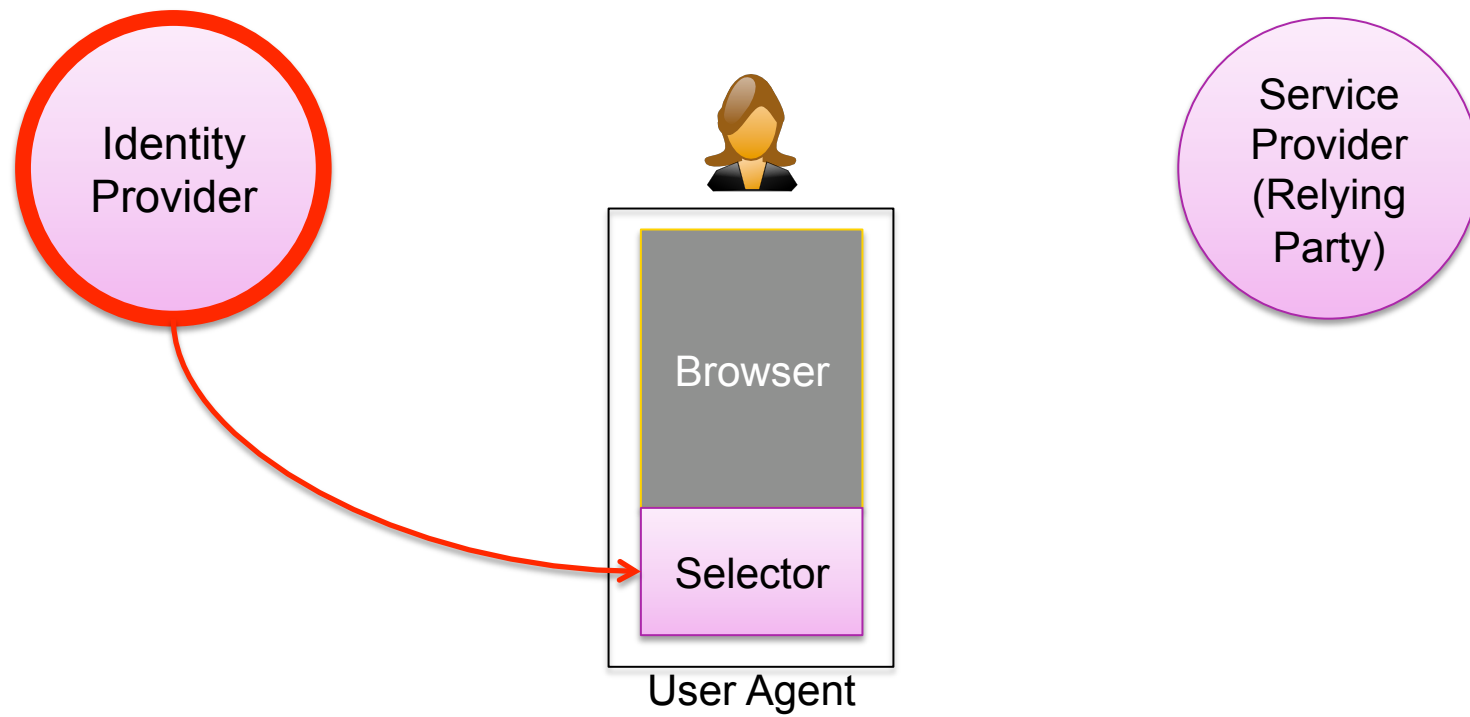
4: Select a card



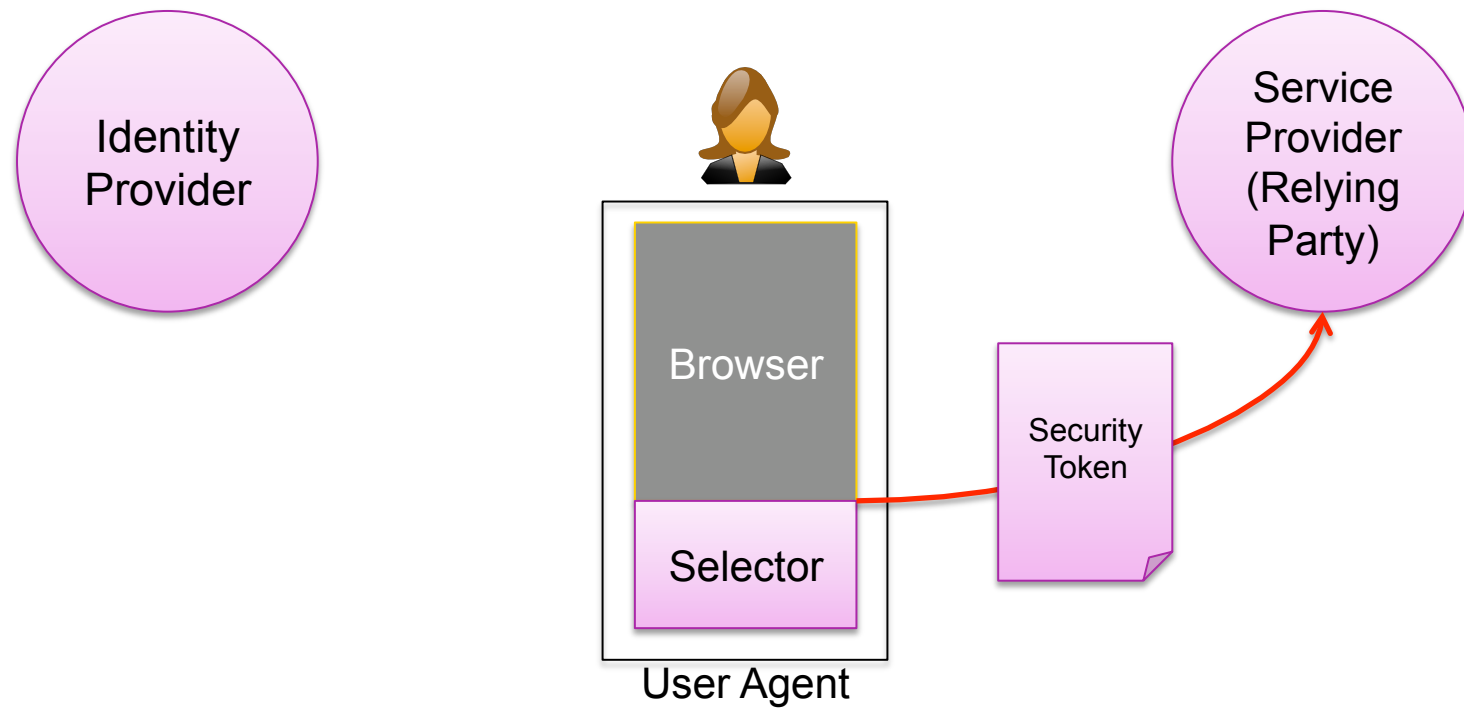
5: Auth to IdP & Request Token



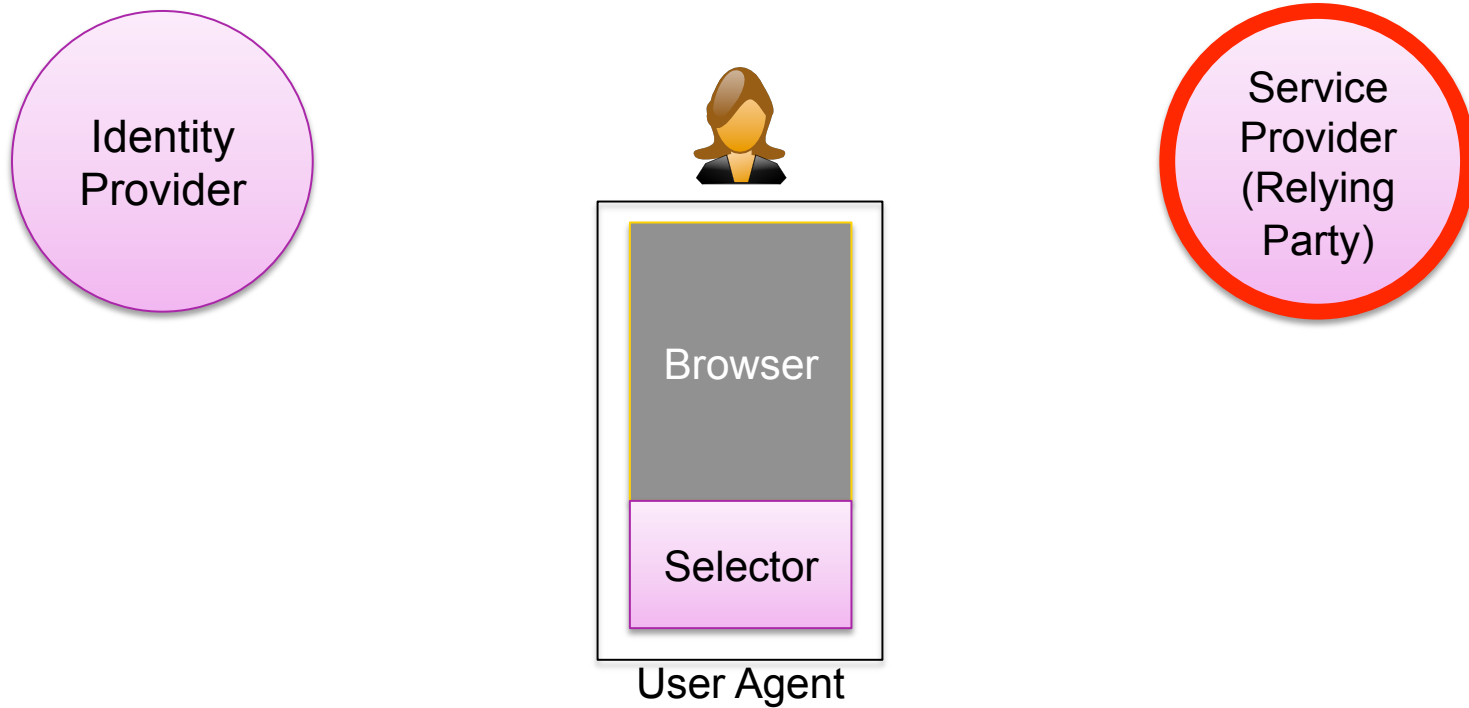
6: Generate token



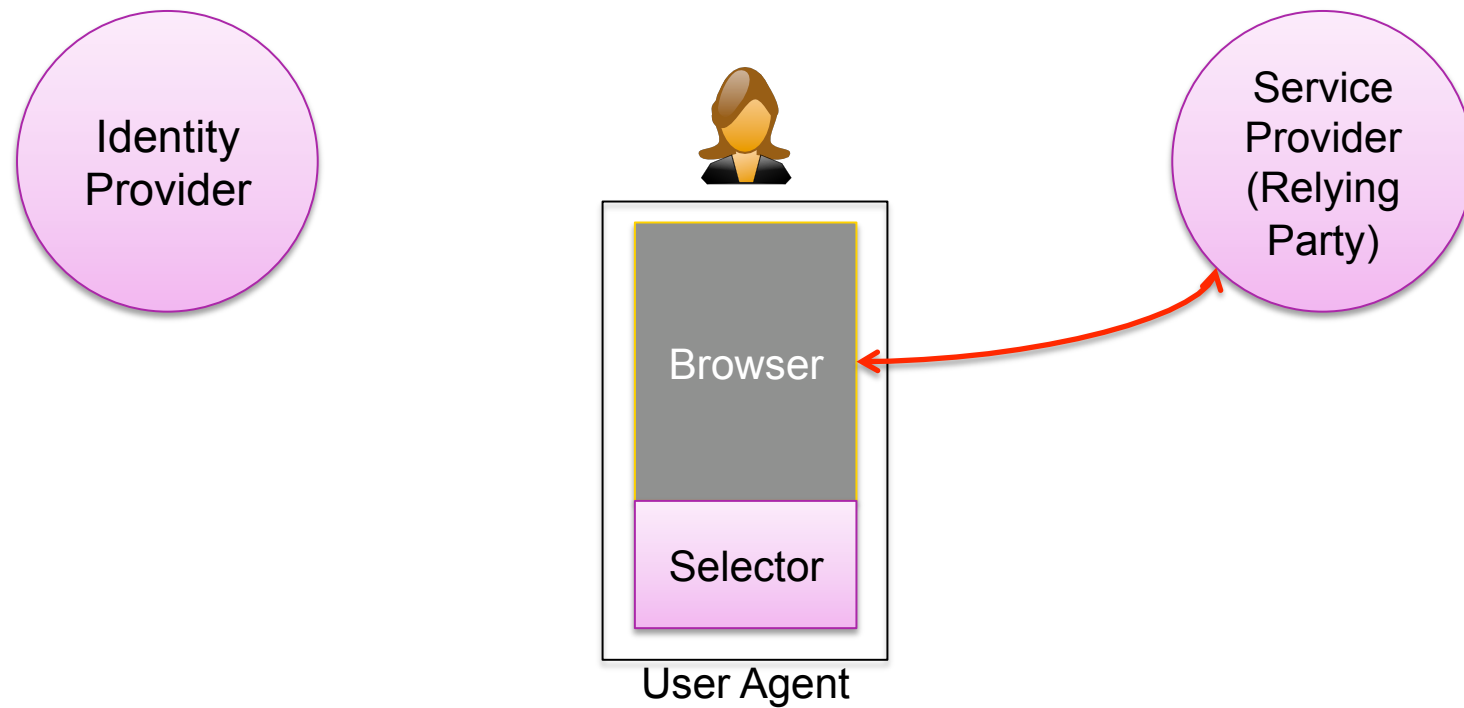
7: Forward token



8: Validate & assess token



9: Alice uses services





David Recordon
OpenID Foundation

George Fletcher, AOL LLC.
Chief Architect
Identity Services

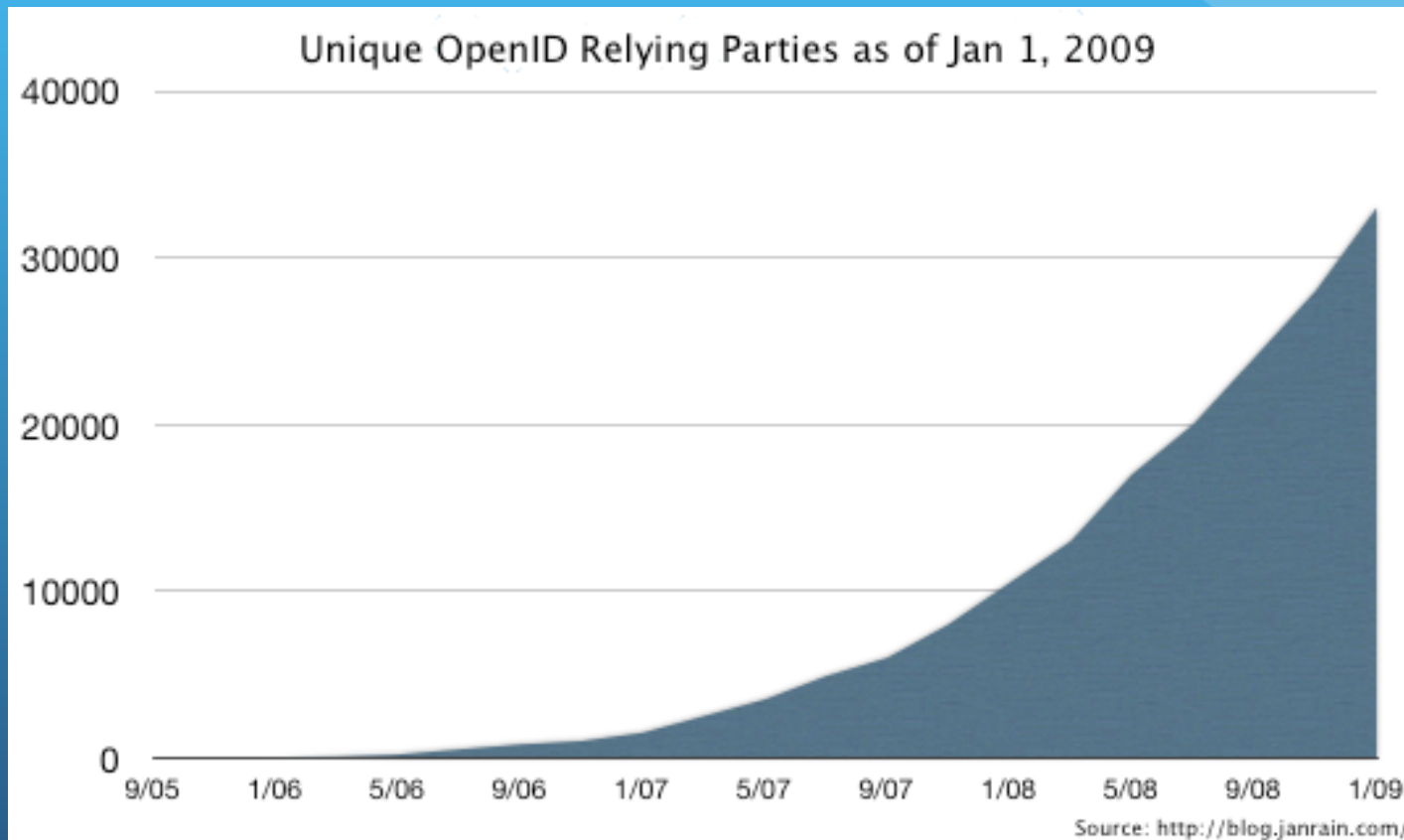
What is OpenID?

OpenID is an easy to implement and decentralized identity protocol designed to be used across the Internet.

<http://openid.net/>



Over 35,000 OpenID Relying Parties



Nearly One Billion OpenIDs



OpenID is a great way to engage citizens via social media.

Original OpenID “Design Goals”

- Empowers people in a user-centric fashion
- De-centralized with no single enforced trust model
- Simple specifications: “small pieces loosely joined”
- Easily deployable in a vast array of environments
- Allows people to choose to invest in portable reputation based upon their OpenID identity
- Privacy protecting use cases supported via “Directed Identity”

OpenID Brand



OpenID Identifiers

- URLs directly issued by OpenID Providers
 - <http://openid.aol.com/gffletch>
 - <https://recordond.pip.verisignlabs.com/>
 - <https://me.yahoo.com/a/69Ely0U13vQcyKjQsmRCurcBZX0glvM->
- User owned URLs via Delegation
 - <http://www.davidrecordon.com/>
 - <http://practicalid.blogspot.com/>
- OpenID Provider URLs for Directed Identity
 - <http://openid.yahoo.com/>
 - <http://api.myspace.com/openid>







OpenID Specifications

- OpenID Authentication 2.0
 - http://openid.net/specs/openid-authentication-2_0.html
- Extensions
 - Provider Authentication Policy Extension (PAPE)
 - http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html
 - Attribute Exchange (AX)
 - http://openid.net/specs/openid-attribute-exchange-1_0.html
 - Simple Registration (SREG)
 - http://openid.net/specs/openid-attribute-exchange-1_0.html



Two Basic User Interactions

Sign In X

Pick an OpenID provider

Or pick another third-party account

How does it work?



OpenID
Provider

OpenID
Relying
Party

How does it work?

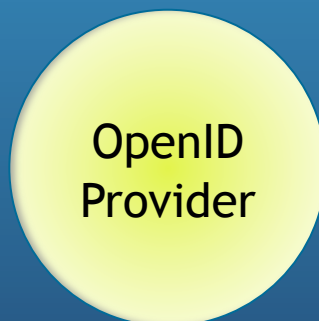


Present
OpenID

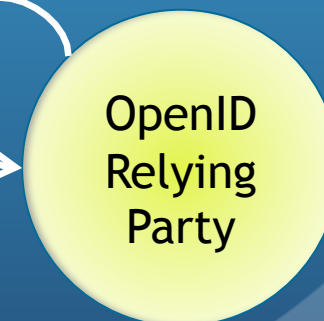
OpenID
Provider

OpenID
Relying
Party

How does it work?



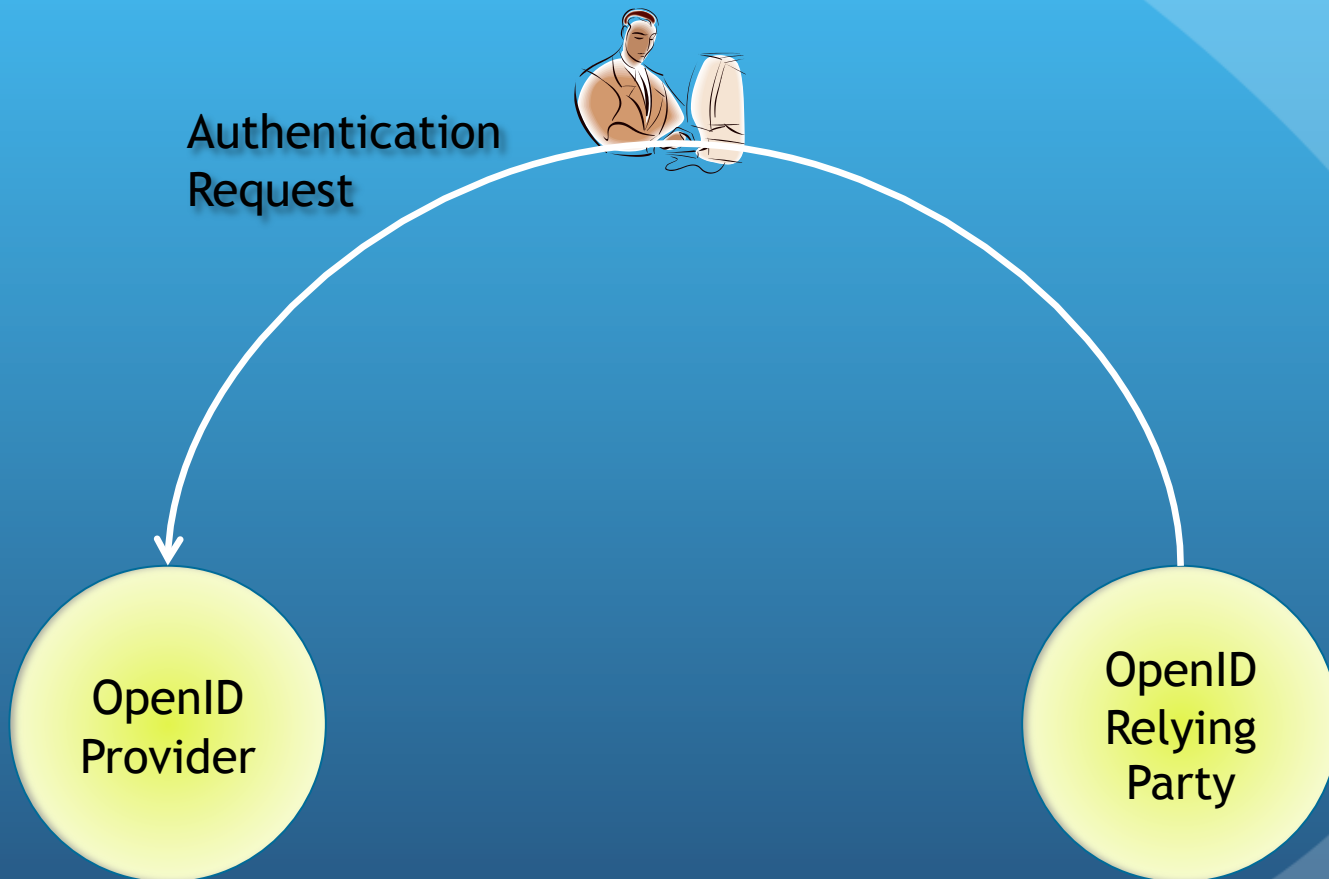
Discover
OP



How does it work?



How does it work?



How does it work?



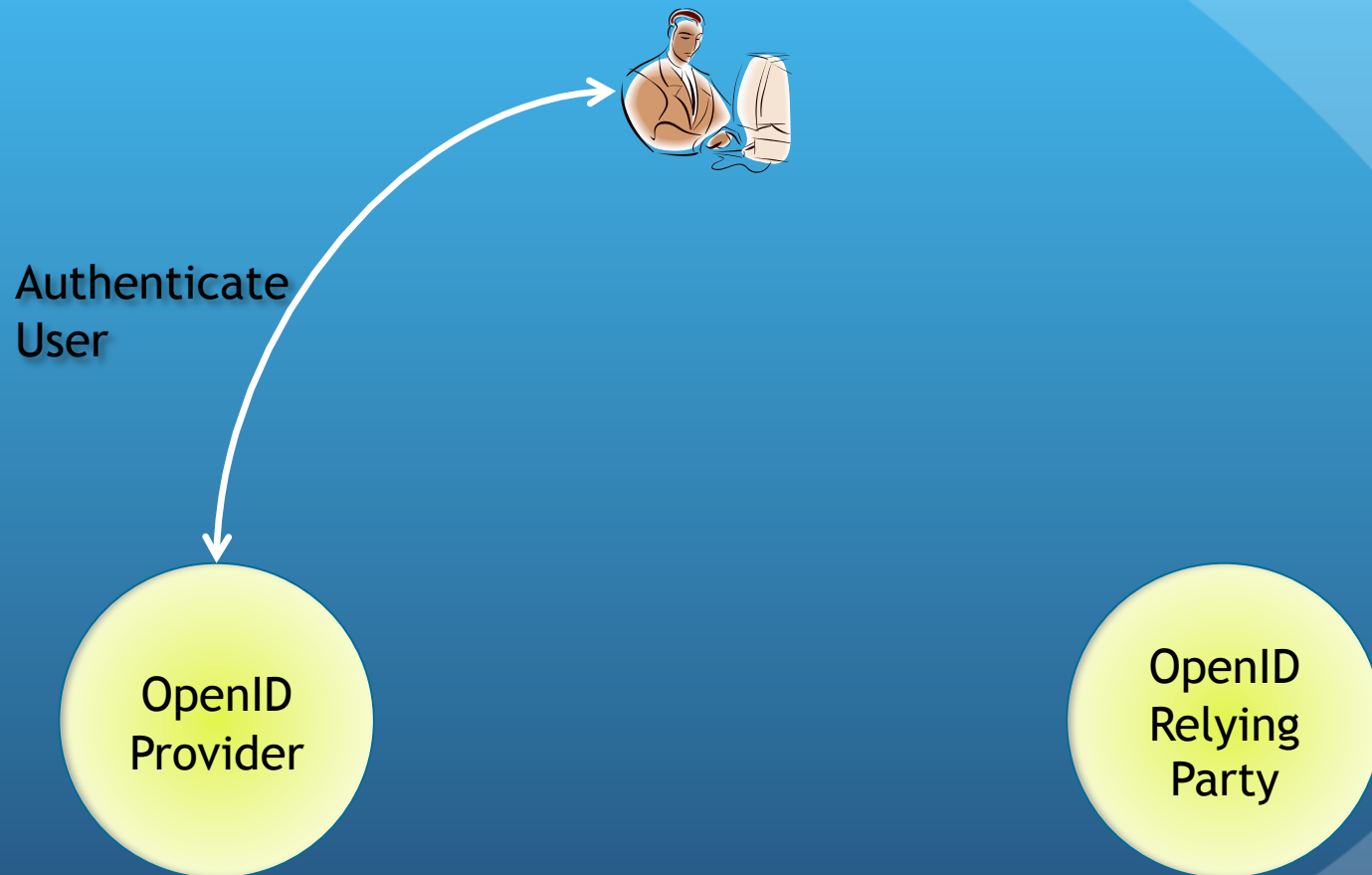
Realm/RP
Validation



OpenID
Provider

OpenID
Relying
Party

How does it work?



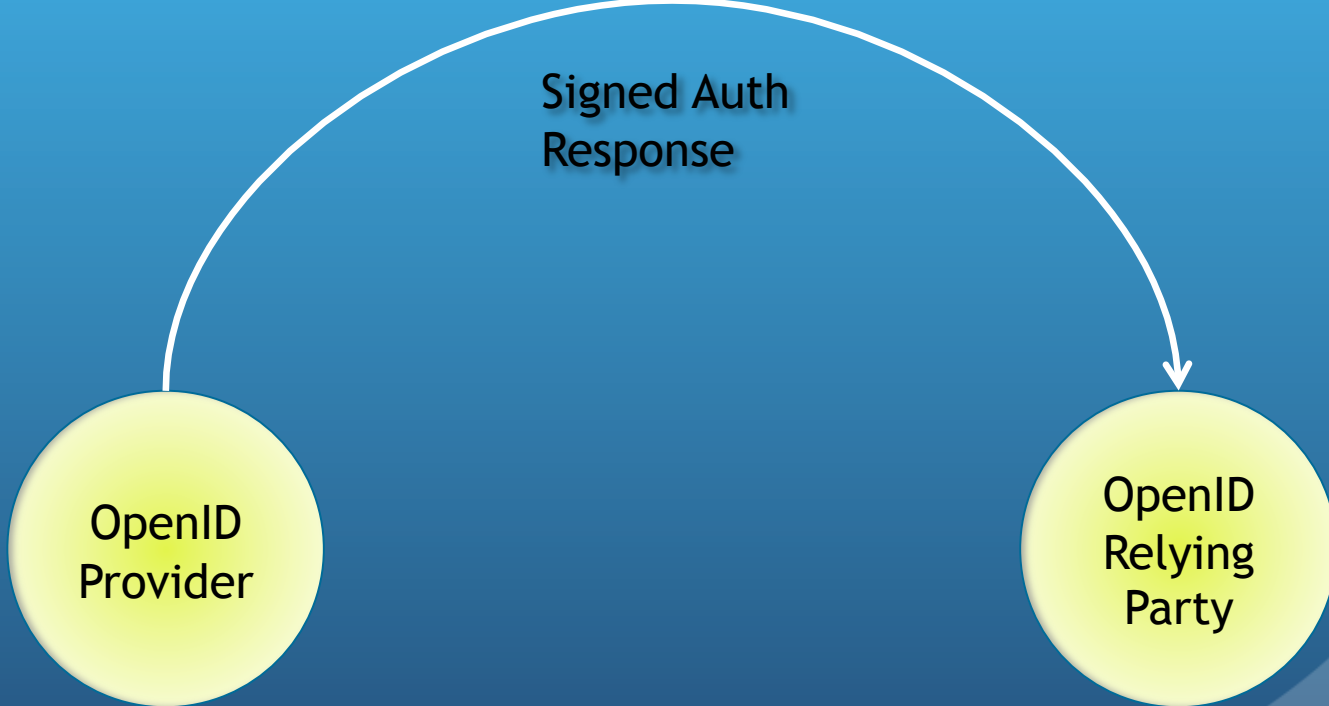
How does it work?



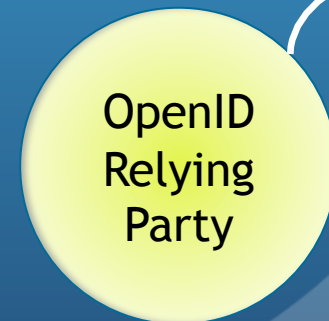
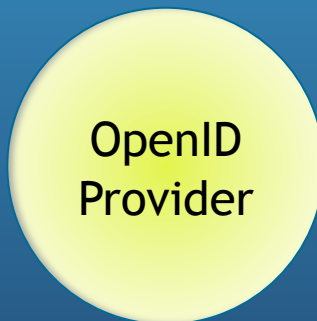
Signed Auth
Response

OpenID
Provider

OpenID
Relying
Party



How does it work?



Verify
Signature



How does it work?



Personalized
Content

OpenID
Provider

OpenID
Relying
Party

Appendix

Security Characteristics

- SSL enabled OpenID URLs and Provider endpoints
 - SSL must be used to protect MITM attacks
- Crypto separated from trust and “identity proofing”
- Providers can authenticate the user however they please and describe the authentication via PAPE
- Obviates the need for the “password anti-pattern”
- Core data path is through the browser
 - Except for establishing the shared secret used to verify authentication assertions
- Provider responses are not encrypted; just signed

Attribute Exchange Extension

- Purpose
 - Allow the relying party to request/require user attribute data by the OpenID Provider
 - Attribute specification is extensible
 - Supports push notifications on attribute data change
- Verified Attributes
 - Proof of concept demos of exchanging signed SAML assertions via OpenID Attribute Exchange as a tie to PKI systems

PAPE Extension

- Purpose
 - Allow the Relying Party to request/require certain authentication policies of the OpenID Provider
 - Allows the OpenID Provider to describe characteristics to the Relying Party of how the user authenticated
 - Supports extension to other policies (such as those defined in NIST SP 800-63) via the auth_level parameter
- Default Supported Policies
 - Phishing-Resistant Authentication
 - Multi-Factor Authentication
 - Physical Multi-Factor Authentication

Identity, Credential, and Access Management at NASA, from Zachman to Attributes

Corinne S. Irwin
NASA
Code JD, 300 E Street, SW
Washington, DC 20546
1-202-358-0653

Corinne.S.Irwin@nasa.gov

Dennis C. Taylor
NASA (INDUS Corp.)
Code 720, NASA GSFC
Greenbelt, MD 20771
1-301-286-4290

Dennis.C.Taylor@nasa.gov

ABSTRACT

To achieve the ultimate goal of attribute-based access control (ABAC), a robust architecture for Identity, Credential, and Access Management must first be established. The National Aeronautics and Space Administration (NASA) began formal development of its Identity, Credential, and Access Management Architecture using the Zachman Framework for Enterprise Architecture in June 2006. The Architecture provided the necessary structure to meet aggressive deadlines for issuance and use of the PIV smartcard. It also led to the development of NASA's Logical Access Control infrastructure to support not only PIV smartcards, but all authentication credentials in use at NASA.

Use of the Zachman Framework has transformed the way that NASA looks at Logical Access Control, and has positioned NASA to provide robust attributed-based access control in the future. In this paper, we will discuss the Logical Access Control System (LACS) we are implementing at NASA, changes in the way NASA views Identity Trust and Level of Assurance, technical challenges to implementation, and our future vision for Identity, Credential, and Access Management.

Categories and Subject Descriptors

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

© 2009 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

IDtrust '09, April 14-16, 2009, Gaithersburg, MD
Copyright 2009 ACM 978-1-60558-474-4...\$5.00

General Terms

Security, Design

Keywords

Attribute-based Access Control (ABAC), Logical Access Control System (LACS), Level of Assurance (LoA)

1. INTRODUCTION

On August 27, 2004, President George W. Bush signed Homeland Security Presidential Directive 12, *Policy for a Common Identification Standard for Federal Employees and Contractors* [HSPD12]. At that time, NASA was within months of deploying its first smartcards for physical access to its facilities. NASA had also initiated projects for Identity Management and Access Management prior to HSPD12. NASA responded to the directive by moving its existing projects for smartcard badging and physical access control, identity management, and account management under a single umbrella program. The projects individually re-examined and adjusted their requirements to meet HSPD12 and its supporting documents. However, it became clear by early 2006 that the projects were still being developed and implemented in a fairly stovepiped fashion. Integration requirements were not well understood or managed.

In June 2006, NASA began the development of the business architecture for Identity, Credential, and Access Management, while we continued system implementation to meet aggressive deadlines for issuance and use of the PIV smartcard.

Developing the business and system architecture not only helped to identify integration points among the projects, but also highlighted areas that were not being addressed by existing projects. One of the missing elements was the Logical Access Control System (LACS). The Logical Access Control Integration Team (LACIT) was chartered in October 2006 to fill this gap.

In this paper, we will discuss how we implemented the LACS at NASA to meet the requirements of HSPD12 and

prepare NASA for comprehensive ABAC. We will explain how we used the Zachman Framework for Enterprise Architecture [Zachman] to develop an integrated enterprise architecture for Identity, Credential, and Access Management. We will then provide an overview of NASA's LACS requirements and use cases. We will examine how NASA's view of Identity Trust has changed due to HSPD12 and National Institute of Standards and Technology (NIST), Special Publication 800-63, Electronic Authentication Guideline [SP80063], and explain our need for Level of Assurance (LoA) attributes to authorize access based on the credential presented. Finally, we will look at NASA's future plans to implement a robust ABAC architecture.

2. THE NASA ENTERPRISE

NASA is comprised of 10 major field centers, plus additional facilities. The NASA workforce includes 20,000 civil servants and 80,000 permanent, on-site contractors. NASA systems are accessed by tens of thousands of additional partners at universities, corporations, and other US and foreign governmental entities throughout the world. Unlike most federal agencies, NASA requires a large number of remote and foreign users to access its systems in order to meet its mission. The scope of our implementation is the entire NASA enterprise.

NASA has historically operated in a highly decentralized IT environment. Each field center, and often each project, would develop its own technical infrastructure to provide access to its systems. The result is that, just prior to implementation of our consolidation efforts, NASA had:

- 13 different identity management systems
- 12 different X.500 systems fed from the identity management systems
- 28 RSA token infrastructures
- Hundreds of Active Directory domains

Account management and authentication were also highly decentralized. NASA had at least 7 different account management systems. Most access, however, was granted based on approvals on paper forms.

Of the approximately 3,000 applications in use at NASA, about 1,000 used Active Directory to authenticate users. The remaining applications used local user tables and custom authentication routines on an application-by-application basis.

Over the years, several attempts to consolidate and centralize NASA's Active Directory infrastructure failed due to the lack of political will to consolidate. HSPD12 provided the regulatory impetus that NASA needed to consolidate its IT infrastructure, increase security, and

improve the user experience. The requirements for Identity, Credential, and Access Management provided a derived requirement for a single Identity and Credential management system, single AD forest, and single directory infrastructure.

When NASA merged its projects to meet HSPD12 requirements, a project management compliance audit was conducted to determine the project organization and implementation changes that were needed to more closely integrate the identity, credential, and access management projects. A major recommendation from the review was to apply the Zachman Enterprise Architecture (EA) framework to ensure successful implementation of HSPD12 requirements.

Designing, implementing, and transitioning to a completely new infrastructure that impacts every NASA worker on a daily basis was no simple task. We were, in fact, changing the basic enterprise architecture of the Agency.

3. USE OF THE ZACHMAN FRAMEWORK

The Zachman Framework for Enterprise Architecture [Zachman] is a methodology for developing large, complex systems starting with scope, then working through layers for the Business, System, and Technology models, and finally providing detailed representations of the system. Each layer addresses Data, Function, Network, People, Time, and Motivation.

The Clinger-Cohen Act was passed by Congress in 1996, and required Federal Agency Chief Information Officers to develop, maintain, and facilitate integrated systems architectures. Since that time, a series of laws, requirements and guidance issued by Congress, OMB, Treasury, NIST, GAO and the CIO Council have established the Federal Enterprise Architecture Framework. Agencies are required to include certified Enterprise Architects on their staffs, and report EA metrics on an annual basis to OMB.

The Zachman framework is recognized as the standard for classification of Business, System, and Technology layers of the Enterprise. Other architecture frameworks, including the Federal Enterprise Architecture Framework and The Open Group Architecture Framework utilize Zachman's classification and differentiate themselves by providing methodologies for development of the framework.

In June 2006, we commenced a series of workshops with a small group of subject matter experts to develop the business model. The business model includes business processes, entity relationships, and definition of actors to perform the various tasks in the process.

As NASA builds out the Identity, Credential, and Access Management architecture, the Zachman framework continues to figure prominently in our development. The detailed business model is the precursor to any system enhancement. While the initial effort to develop the business model was difficult and time-consuming, we now find that system enhancements move very quickly from the business process design through the system and technology layers. Release reviews are efficient, and there is little re-work during the system design and implementation phases.

3.1 NASA's Identity, Credential, and Access Management Business Architecture

To communicate NASA's business architecture to the greater NASA community, we created Figure 1, "The Really Big Picture." Figure 1 was designed by consolidating and simplifying a number of entity/relationship diagrams, state models, and business processes. The figure is a simplification of the business architecture; therefore, some relationships are left out or simplified. It provides a high-level explanation of how our architecture works.

A Position is created to perform some work for NASA. A Position is assigned to a Worker. Based on the

requirements of the Position, the Worker is subject to an Investigation, and may require a Clearance. The Worker is issued a Credential once the Investigation is successfully adjudicated.

At the same time, either the Position or the Worker can be granted Membership in a Community. A Worker or Community can be granted Access Permission to an Asset or Asset Group.

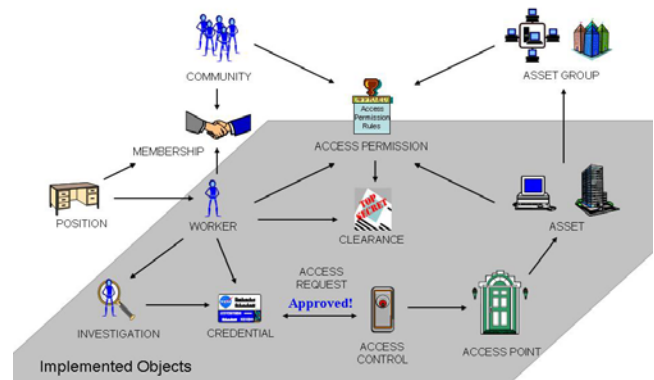


Figure 1: NASA's Really Big Picture

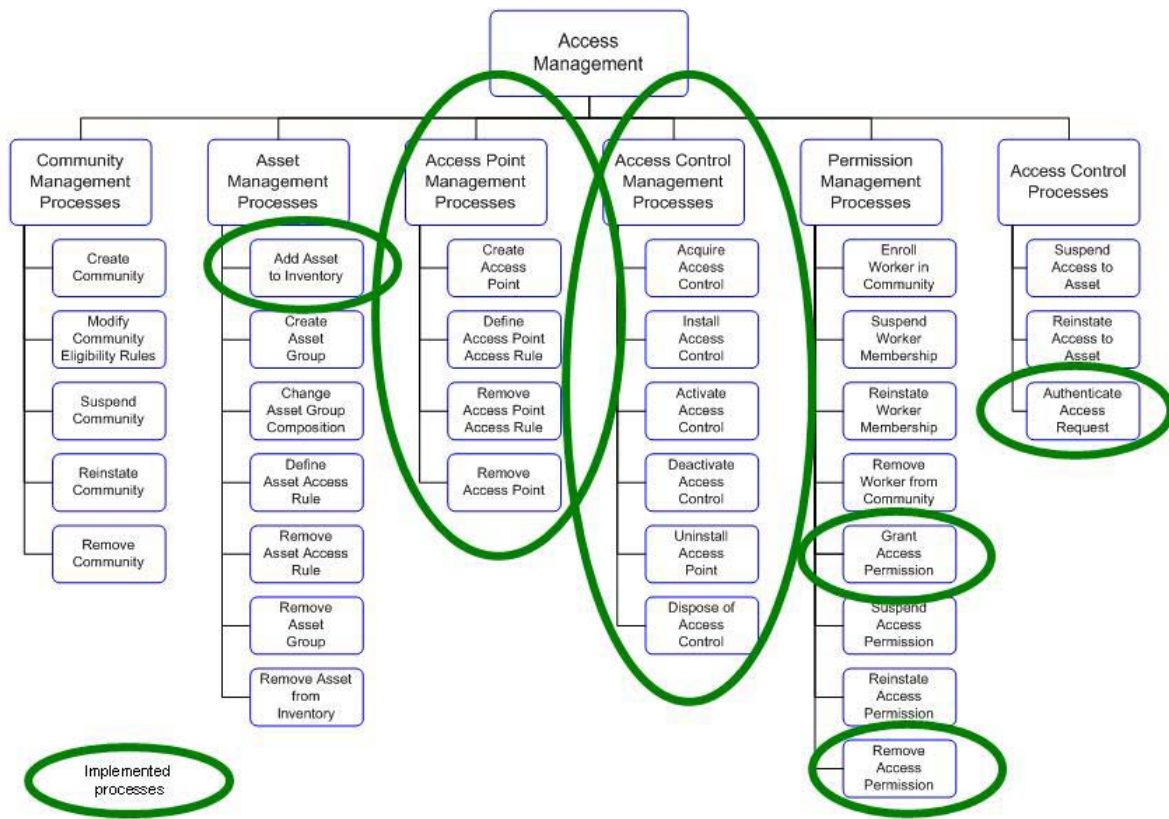


Figure 2: Access Management Business Processes

When a Worker wants to access an Asset, s/he presents his/her Credential to the Access Control device in order to gain access through the Access Point to the Asset.

3.2 Access Management Business and System Models

Figure 2 shows the list of Access Management business processes. (There are separate lists for Identity and Credential Management.)

The circled processes are those that have been implemented at NASA. Notably, the entire set of Community Management processes is unimplemented at this time. With no community management, the permission management processes are also unimplemented with regard to communities.

As shown in Figure 2, NASA today is only able to assign an access permission from a worker to an asset. While some Basic Levels of Entitlement have been implemented using standard attributes, NASA does not have a system in place today to register the access permissions granted on the basis of attributes. For example, we allow any NASA civil servant access to our Human Capital Information Environment (HCIE), based on the identity attribute (employer=NASA) found in our directory. However, there is no system that defines the Community of NASA civil servants, and no registry that shows that the Community, NASA Civil Servants, has been granted an Access

Permission to the Asset Group HCIE. One would have to delve into the code of the HCIE itself to discover this relationship.

Less obvious, but no less important, we do not have well-defined asset management processes to support access management. NASA does perform asset management per se: we have inventories of computer equipment and management systems around them. However, those systems are designed to manage the acquisition, ownership, and disposal of assets. They are not designed to support access management to those assets. In the realm of logical assets such as applications, NASA is in the process of building its application asset inventory.

These to-be-implemented objects are the key to NASA moving beyond simple communities such as NASA civil servants, into more complex, approval-based communities such as projects. NASA's core business is conducted through programs and projects, with multidisciplinary members matrixed from across the Agency. Project membership cuts across organizational and geographic boundaries, and changes as projects are initiated, implemented, and completed. Community Management is needed as a precursor to providing ABAC to projects' IT assets.

Figure 3, Authenticate Access Request, is shown as an example of a detailed business process.

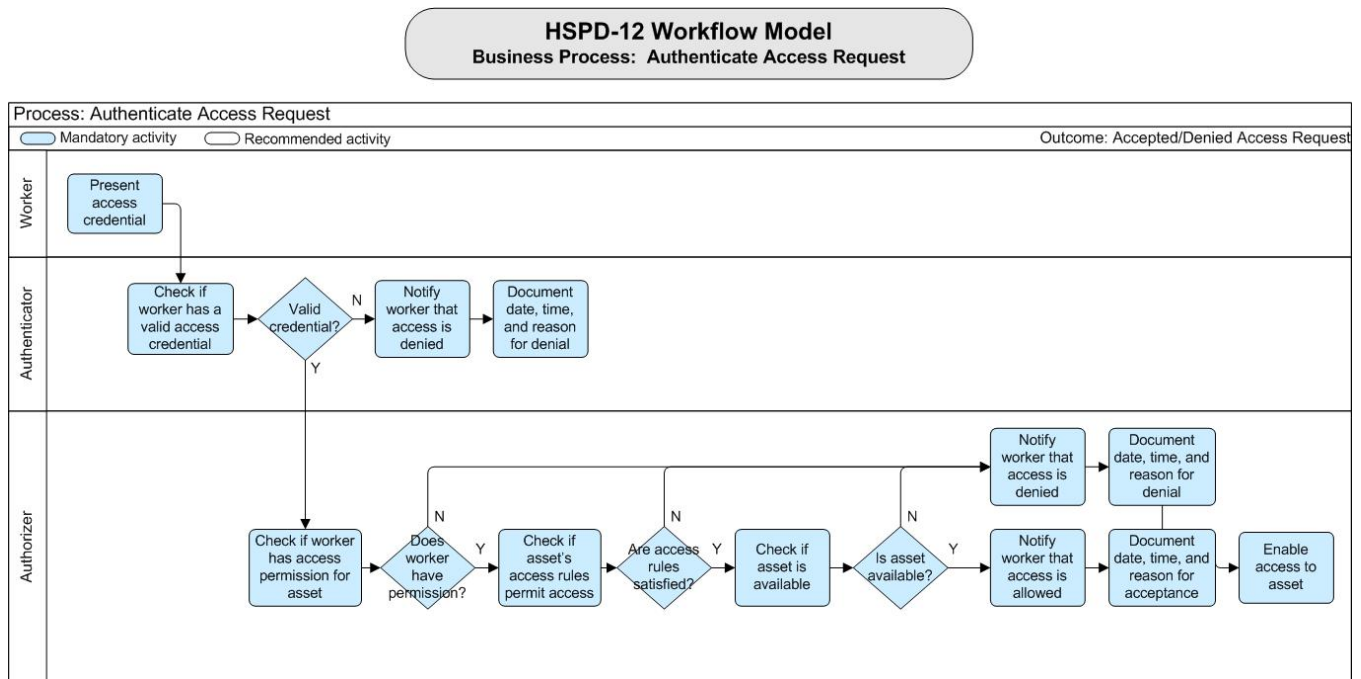


Figure 3: Business Process: Authenticate Access Request

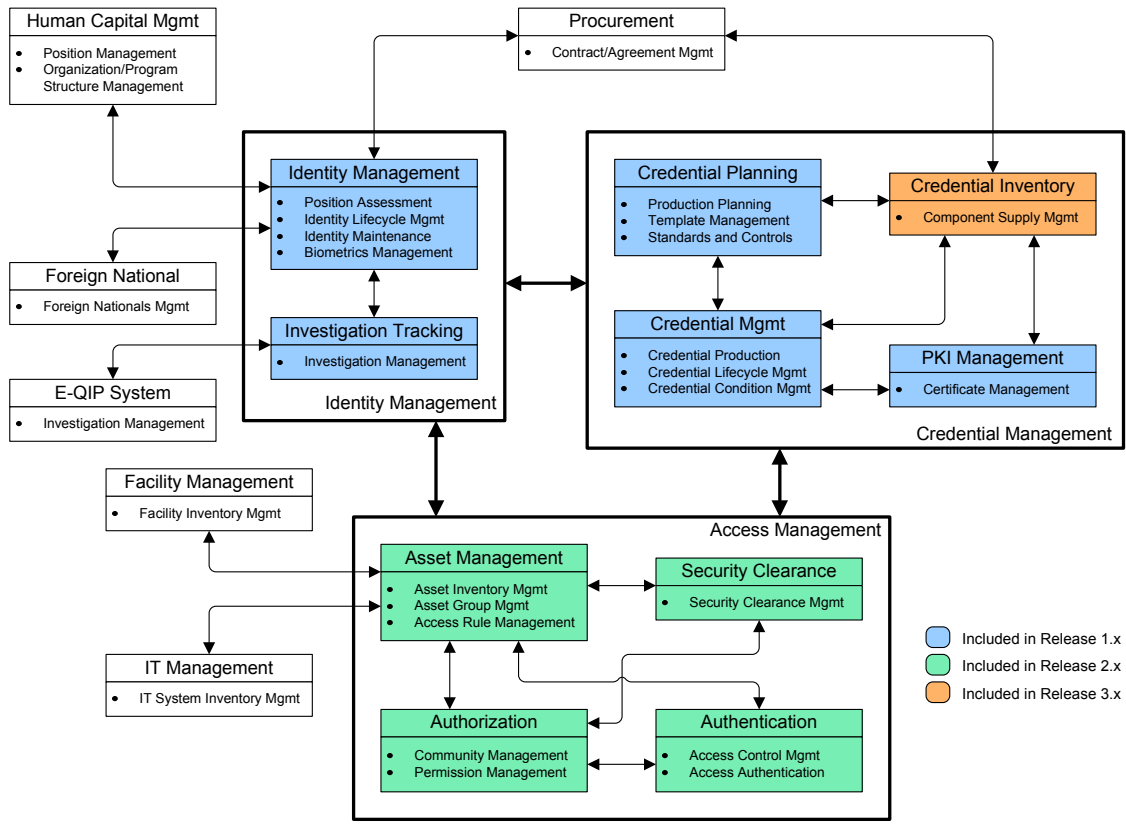


Figure 4: System Model

When the Business Model was completed, the System Model was derived from the Business Model. The System model sets boundaries for identity, credential, and business management, both internally and externally.

Once the System Model is derived, assignment of responsibility can take place. What we found at NASA was that several systems had overlap of responsibility, and other areas had not been assigned to any project. The authentication and authorization systems were assigned to three separate projects: NASA’s Consolidated AD project, the NASA Enterprise Directory project, and eAuthentication. There was overlap, but no comprehensive plan to ensure that these three worked as a system to meet NASA’s authentication and authorization requirements.

Therefore, use of Zachman led directly to the establishment of the Logical Access Control Integration Team (LACIT), which was chartered to address single- and multi-factor authentication and authorization across the Agency. LACIT’s scope was derived from the System Model depicted in Figure 4, to include the Authentication and Authorization modules.

4. TECHNICAL MODEL OF LOGICAL ACCESS CONTROL

In the federal arena, use of the PIV smartcard is generally divided into Physical Access Control Systems (PACS) and Logical Access Control Systems (LACS). LACS includes any logical access including desktop authentication and access to systems and applications.

When HSPD12 was signed, NASA was nearing operational rollout of its own smartcard infrastructure for Physical Access Control. However, use of the smartcard for Logical Access Control had not yet been addressed. NASA’s IT infrastructure has historically been highly decentralized; systems tend to be implemented at the project or program level. Authentication is implemented, in most cases, on an application-by-application basis. Early attempts to manage the project to smartcard-enable NASA applications to meet HSPD12 compliance requirements were unsuccessful. The major reason for this lack of success was that we could not clearly articulate how application owners should make their applications “HSPD12 compliant,” because we did not have a well-developed strategy for compliance, nor an infrastructure that application owners could use to meet those compliance requirements.

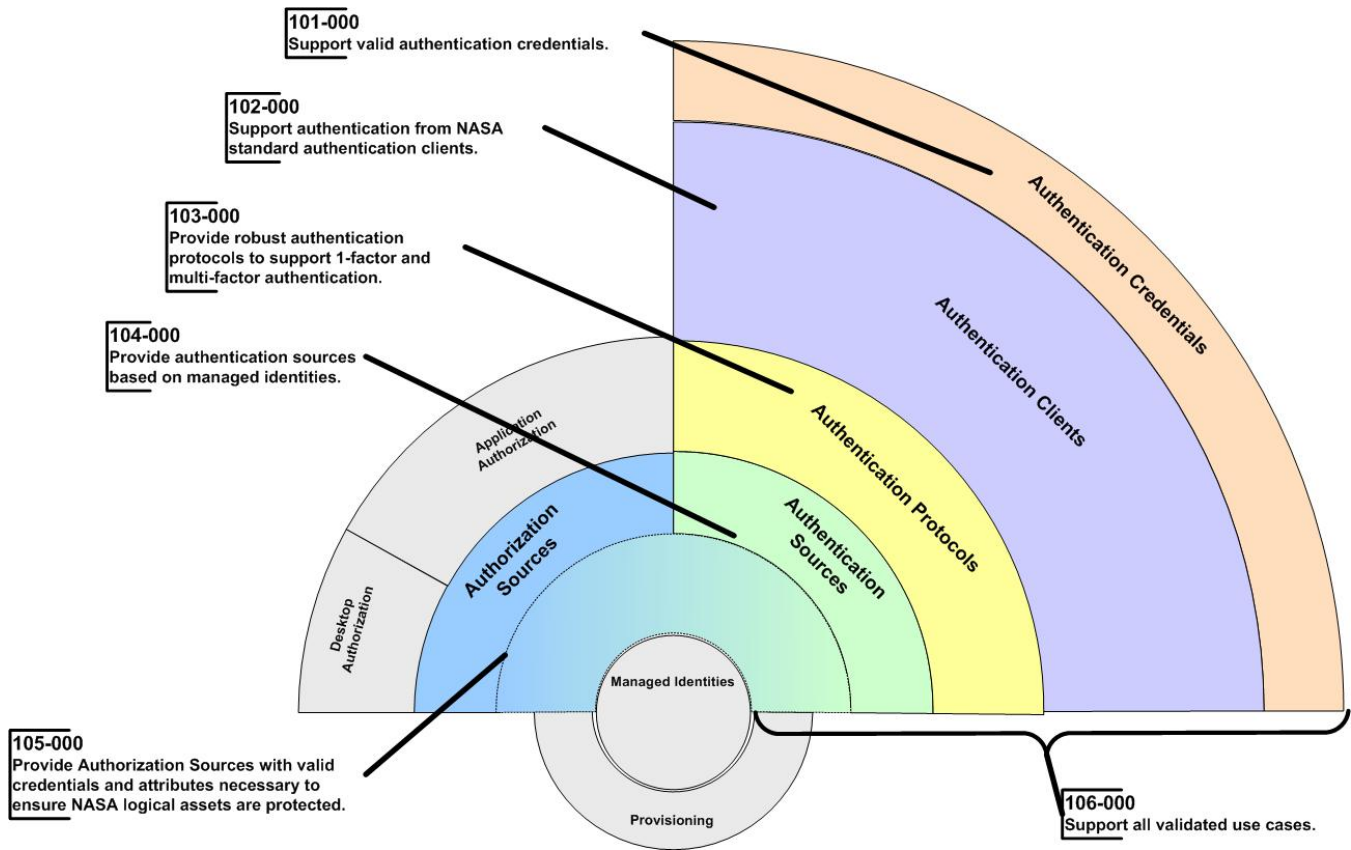


Figure 5: Logical Access Control Framework

LACIT was chartered to develop the high-level requirements for enterprise-wide one-factor and two-factor authentication. The team was chartered in part because the work on the business and system models for Access Management revealed that smartcard-enablement of applications could not be addressed in isolation; rather, smartcard use within the entire spectrum of Logical Access Control at NASA had to be implemented. Smartcard enablement includes a pre-requisite to utilize central authentication sources closely tied to vetted identities. The real effort on the part of application owners, therefore, is not smartcard-enablement per se. It is migrating from local authentication to central authentication.

4.1 LACS Framework and Use Cases

LACIT developed a Logical Access Control Framework (Figure 5), and aligned its requirements with the Framework layers. The framework and its requirements are applied to all enterprise authentication services at NASA. It assures that all enterprise authentication services provide the same level of robust authentication at each layer of the framework.

The framework includes the catchall requirement that the LACS must support all validated use cases.

We identified the use cases by breaking down the components that are used in an access control transaction, and then listing the types of components we might have. The components are:

- The credential being presented
- The device being used to present the credential
- The network location of the device being used
- The system/device being accessed
- The time conditions under which the access is attempted

We turned these into a natural-language construct. Taking the first elements from each component provides an example use case: *A worker with a NASA PIV Card using a NASA-managed PC on the Center Institutional Network to access a resource on the device being used during normal operations.* We ran a program to derive all of the permutations of the use cases. The current raw result is 76,000 use cases, although not all are valid. For example,

while a worker should be able to access resources on the device being used when the network service is unavailable, it is not reasonable to expect to access a remote application when the network service is not available. Removing systematically those use cases we know are invalid, just over 60,000 use cases remain.

The addition or subtraction of a single element results in an order-of-magnitude change in the number of use cases that must be addressed. This recognition has driven NASA to attempt to remove elements from the list of valid use case options wherever possible.

The Use Case model (Figure 6) has helped to scope phases of implementation in a way that is logical and manageable. For example, we are currently focused on implementing those use cases that support institutional and administrative systems being accessed from the institutional network. Use cases supporting our Mission and specialized systems are more unique and carry more risk, so they will be implemented in later phases.

It also underscores that the complexity of the service is increased each time we add an item to one of our component lists. We are therefore motivated to reduce the

items to the minimum necessary to achieve NASA's strategic goals.

4.2 LACS Technical Implementation

Several systems comprise the LACS at NASA. Figure 7 shows these systems and their relationship with our Identity and Credential management systems.

The Identity Management and Account eXchange (IdMAX) system is the Authoritative distribution source for all NASA identities, including civil servants, contractors, foreign nationals, and other affiliates. IdMAX feeds our Common Badging and Access Control System (CBACS) with identity verification data needed to issue a Personal Identity Verification (PIV) compliant smartcard credential. IdMAX also contains the NASA Account Management System (NAMS) workflows needed to issue other NASA Credentials, including Agency AD accounts, RSA tokens, and PKI encryption and signing certificates. NAMS supports access management to NASA applications as well. NASA has identified over 3,000 applications across the Agency. At this time, about 500 of those applications are integrated into NAMS for access management. All applications are required to use NAMS for account management by the end of Fiscal Year 2010.

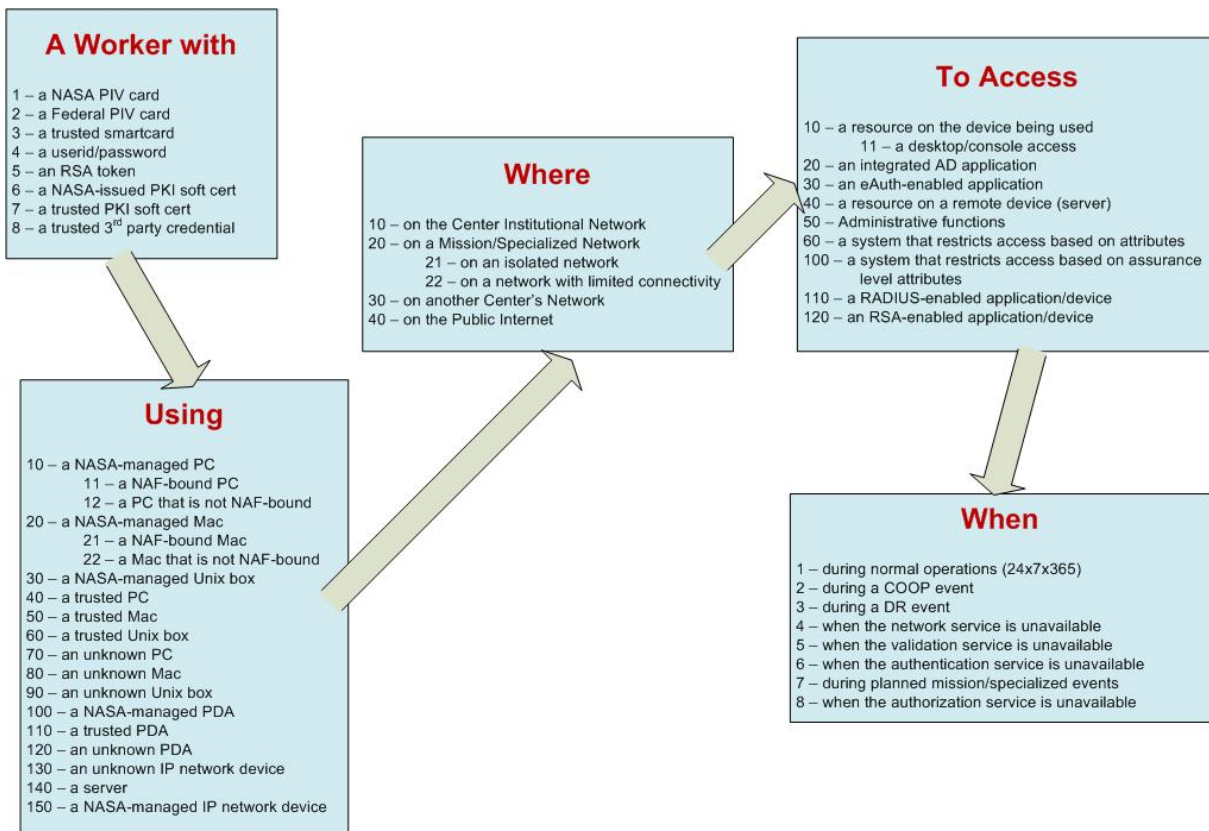


Figure 6: Authentication Use Cases

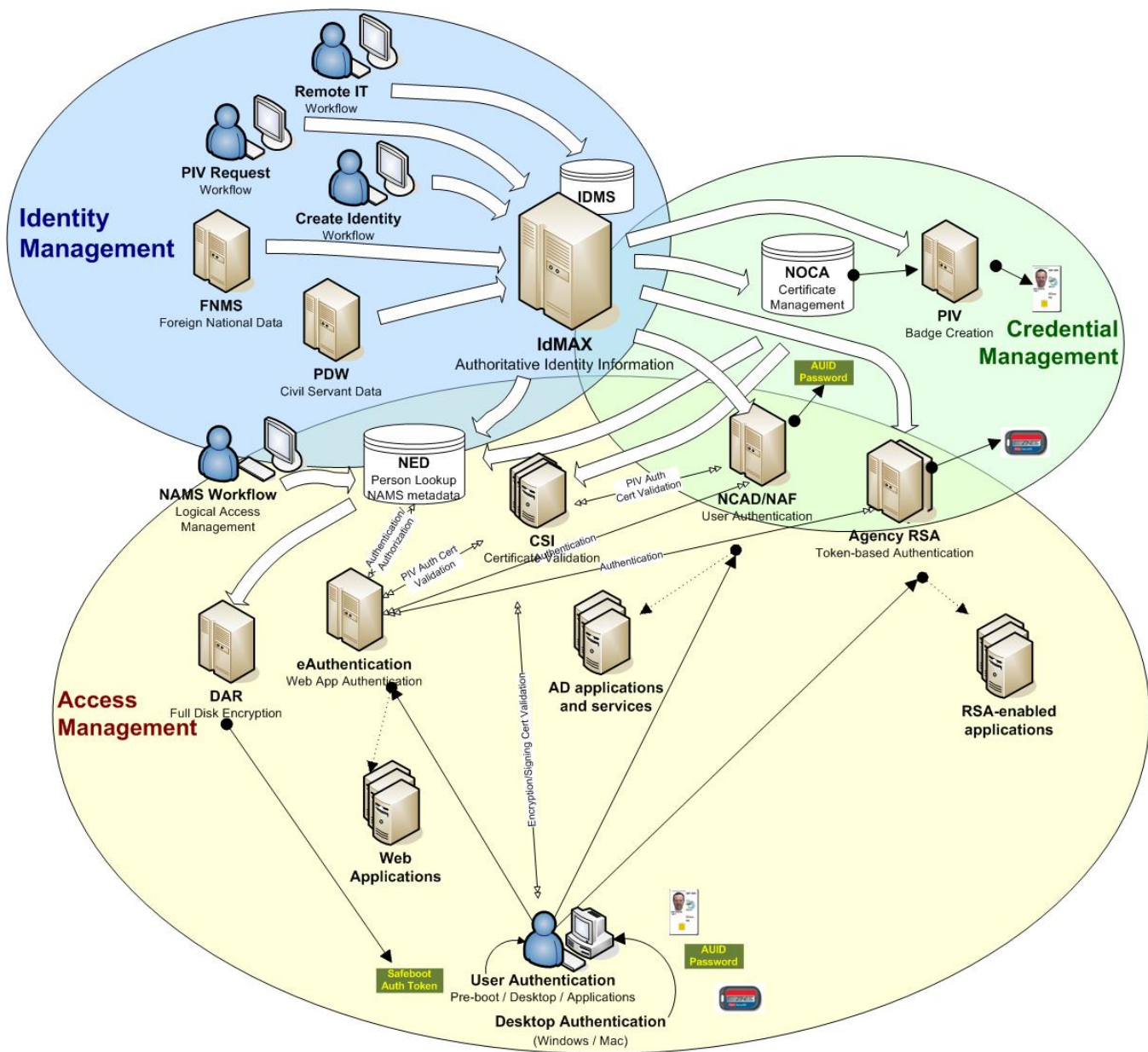


Figure 7: Technical Architecture

The NASA Enterprise Directory (NED) provides both web-based lookup and LDAP-supported search of identity data for NASA workers.

The NASA Agency Forest (NAF, NASA's enterprise AD), eAuthentication (built on Sun Access Manager), and Agency RSA provide authentication and authorization services based on the identities in IdMAX, and the authorization attributes created through the NAMS process. The authorization attributes are contained in either the

NED (for eAuthentication), or the NAF (for AD). Both the NAF and eAuthentication will be smartcard-enabled in FY 2009.

The Agency RSA service provides an alternate two-factor authentication service for use cases where a smartcard cannot be used. For example, IT remote users who will never receive a smartcard, and access Moderate risk systems at NASA, will require an alternate such as RSA. We expect the use of RSA tokens to decline over time, as

PIV smartcard use becomes ubiquitous, and smartcards from other issuers are certified and accepted.

Primarily because of firewall segmentation between Centers, the authentication systems of NAF, eAuthentication, and RSA are implemented in a distributed model. In the example of the NAF, in order to meet or exceed all previous performance and reliability requirements, the NAF domain controllers are instantiated on a one-to-one mapping to match legacy Center domain controllers. The motivation here is that newer equipment with higher-performance CPUs and robust storage systems located on the same network segments as previous equipment will allow us to guarantee performance requirements even if all of the exigencies of the previous environment are not known.

HSPD12 directly motivated the change to centralized Active Directory services. Centralization had been considered for many years, but Centers were unwilling to surrender autonomy due the perceived risk inherent in the centralized structure. One of the arguments against centralization included the classic “risk in putting all your eggs in one basket”. NASA mitigated this concern by implementing a strong commercial Security Information and Event Management (SIEM) system to monitor the NAF. Ultimately the prospect of solving the relatively complex issue of integration of smart card logon once centrally versus many times over in each AD instance was finally enough to sway the argument to consolidate to one Active Directory infrastructure.

4.3 If You Build It, Will They Come?

NASA’s LACS is largely built. Migration to the Agency AD infrastructure is underway, and all NASA centers will be migrated to the Agency AD infrastructure by the end of 2009. Over 700 applications that are integrated with local AD domains today will become part of the Agency AD infrastructure as part of this migration.

NASA’s eAuthentication service has been available for about a year, and in late 2008, additional servers were distributed throughout the Agency to improve availability of this re-constructed service. A few applications have been integrated with eAuthentication already, and several more are in development and test at the time of this writing.

The Agency RSA infrastructure will be implemented in FY 2009 to consolidate NASA’s 28 RSA implementations and ensure a consistent process for RSA token issuance, tied to vetted identities in IdMAX.

The work of LACIT culminated in the publication of a NASA Enterprise Architecture Standard, *Standard for Integrating Applications into the NASA Access Management, Authentication, and Authorization Infrastructure*, in August 2008. The standard includes

compliance deadlines for all NASA applications that stretch from 2008 – 2011. The standard requires that all applications utilize NAMS for access management by 2010. It requires that all applications utilize either the NAF or eAuthentication by 2011, or have an approved deviation to utilize an alternate authentication method. (Agency RSA is one of the possible deviations.)

By the end of FY 2011, virtually all NASA applications will be integrated into the Central Authentication and Authorization Infrastructure, and when this integration is complete, attribute-based authorization based on communities becomes a real possibility.

These assertions belie the enormous culture change imposed on the Agency to achieve these goals. Centers and projects must give up control of identities, credentials, and access management and control to a central authority. While there are significant technical challenges that must be met to implement the infrastructure, the change management that we have had to implement to convince our customers to actually use the infrastructure is far more difficult.

The benefits to the Agency are clear: improved security and cost containment benefits, brand-new opportunities for inter-center collaboration, and the ability to access all services from any NASA center as well as remote locations. It is more difficult to demonstrate the benefits to centers and projects until some critical mass of applications have integrated with the new infrastructure. Yet those benefits are real: improved user experience through the use of single sign-on, faster and more convenient access to a suite of applications through a central, on-line access management system tied directly to the authorization source for the application, and improved mobility, supporting users as they move from center to center, travel, or telecommute.

5. IDENTITY TRUST

HSPD12 and FIPS Publication 201-1: *Personal Identity Verification (PIV) of Federal Employees and Contractors* [FIPS201] established the requirements for a federally-interoperable credential for use in federal systems. On the surface, this implies that a non-NASA federal worker need only present his or her valid federal credential in order to be granted access to the NASA system s/he needs.

This premise is sound as far as authentication goes; however, it ignores the necessary trust path for authorization to NASA systems. While NASA trusts other federal agencies to properly identify, vet, and credential their employees, only NASA has the right to determine access to a particular NASA asset.

These authorization requirements provide a derived requirement for NASA to create a “NASA Identity” for other federal workers, which references the credential issued by his/her home Agency. An identity record for each non-NASA federal worker that has a business relationship with NASA is therefore included in NASA’s IdMAX system. Through NAMS, this NASA identity is granted an AD account as well as an entry in NED which references the federal credential (UPN, subject alternate name, etc.) NAMS is used to grant access to the NASA applications to which access should be granted.

Because eAuthentication was only recently added as an Enterprise authentication service, it was implemented consistent with the new concept of identity trust. Therefore, the discussion in this section focuses on the impact to the AD infrastructure at NASA.

5.1 Previous View of Trusts within the Agency

The advent of client-server and distributed systems beginning in the 1980’s fostered work segregation by workgroups and departments. Systems were also deployed on application boundaries. These approaches collectively led to classic stovepiped systems. Each stovepiped system contained an island of identities. When business needs required interoperability the solution was built bottom-up. The interoperability solution would often be case specific and often relatively novel.

Even single vendor solutions such as Microsoft NT domains and later AD installations were grown from the bottom-up. The number of Microsoft domains grew into the hundreds across the Agency. Interoperability among these Microsoft domains could be obtained with the creation of domain trusts. But, trusts in this model led to haphazard partial meshes. The trust maintenance model was complex due to the n times ($n-1$) number of possible trusts required. Even with a large number of trusts the mesh was never complete and the majority of users were not interoperable. When Agency-wide applications were deployed AD authentication could not be relied upon to be ubiquitously available. As a result, even large, Agency-wide applications relied on their own authentication service accessing an identity store they created for the application itself.

5.2 The Effect of HSPD12 on Internal Trusts

Per HSPD12 credentials must be “issued based on sound criteria for verifying an individual employee’s identity”. The process also mandates in person registration for the PIV authentication credential, the smartcard. The PIV credential itself is unique per person—there is only one PIV credential per person in the Agency. There is a

corresponding one-to-one binding from the credential to the person. In total, this mandates a central directory as a source of authority for identities. All authorization decisions are ultimately bound to this central directory. For NASA, the central directory is the IdMAX system identities, and the authorization binding system is NAMS (see section 3.2).

Since there is now a single authoritative source of identities it provides an impetus to consolidate authentication sources. NASA has chosen to have just two primary authentication sources. The first is Microsoft Active Directory as implemented in a NASA Agency Forest (NAF), a single domain housing accounts for all users who have a requirement for IT access. The second is Sun Access Manager as implemented in the eAuthentication system. These two systems provide all necessary process and protocols for authentication of most applications within the Agency. (The RSA infrastructure is provided for applications as a “deviation” from the standard.)

[M0524] provides implementation guidance for usage of the PIV credential. This memo provides definitions for the terms “Federally Controlled Facilities” and “Federally Controlled Information Systems”. Using this definitive guide NASA has logically drawn a perimeter for inside systems that must use one of the NASA authentication sources, NAF or eAuth. Any of the thousands of NASA systems or applications that do not use the two sources are in deviation of a prescriptive NASA Enterprise Architecture standard (see section 4.3). These systems can only continue to operate if they have an accepted Deviation Transition Plan. The process for filing a deviation is defined in Enterprise Architecture Standard Operating Procedures (SOP). Restating, all NASA systems must use one of the two NASA authentication sources or have an accepted Deviation Transition Plan in order to continue to operate with the NASA environment.

Thus, since all systems and applications must rely upon one of the two authentication sources, and these sources both rely on the same directory of identities, all internal trust requirements are removed. The only internal trusts remaining at NASA are continuing to exist as a deviation.

5.3 Previous View of Trusts with External Partners

NASA has historically had some number of trusts with contractors, a business-to-government (B2G) relationship. The most significant of these historical trusts are used operationally in support of the Space Shuttle program. The major support contractors for Shuttle, such as Boeing and United Space Alliance, have established trusts with NASA. These trusts arrangements are manifested as NT domain or AD trusts. The newer AD forest style trust is implemented

as a Kerberos cross realm trust. At this point in time these trusts can only continue to operate as a deviation.

Vendors have been approaching NASA for some years with interest in developing PKI, AD, or credential trust relationships—credential trust in this case referring to smartcard and underlying PKI trusts. It is possible to exploit PKI trust through the Federal Bridge Certification Authority (FBCA), now FPKI. This PKI trust arrangement generally has not been utilized for relatively classic reasons, such as lack of PKI software maturity, lack of a critical mass of client deployments, problems with certificate status checking availability across disparate networks, and perhaps mostly because PKI was a solution awaiting the right high-value problem. AD trusts have flourished when the AD forest is a subset of the contractor's total population and where most of the identities in the forest are dedicated to the contract. Full scale AD trust relationships with a vendor's AD system housing the complete set of identities, possibly numbering several hundred thousand, has generated security concerns and not been implemented. The concerns generally have to do with limiting authorization only to those individuals with a specific contract relationship with NASA. AD trusts can imply a basic level of entitlement whereby the entire community of a trusted domain has some level of access.

5.4 The Effect of HSPD12 on External Trusts

As referenced in Section 5.2, [M0524] defines the perimeter of systems and locations which NASA must protect under [HPSD12]. NASA must use a PIV credential for authentication and access to these systems and locations. NAMS provides a workflow approval process for defining the authorization of each worker to an application. The NAMS process is implemented based upon the directory of Agency identities, IdMAX. For NASA workers with NASA PIV credential the process is complete for provisioning, authorization, and access for NASA resources.

NASA is also directed to provide interoperability for other Federal PIV credentials. The credentials themselves are expected to be portable and interoperable because of the firm specifications for the PIV token per NIST documentation. The credential will also be interoperable because the PKI is rooted in trust through the Shared Services Program (SSP). However, problems arise if the individual with an external PIV credential is not housed in IdMAX. Without an existent identity in IdMAX there can be no approval process for defining authorizations. Also, without an IdMAX identity there can not be an account created in AD (in the NAF). Another technical issue for AD is that the PIV authentication certificate's subject alternative name is used for identity binding; at least for Windows 2003 (Windows 2008 offers other options). The

subject alternative name must contain a User Principal Name (UPN) value. The suffix portion of this UPN must be predefined in AD as an alternate UPN suffix. There is added complexity in that within a collection of AD forest trusts there can only be one forest which claims ownership of a UPN suffix. So, there is not yet an approach to provide authentication and authorization based on a worker provisioned with an external PIV credential.

Problems also arise with the legacy trusts as referenced in 5.3 above. By definition, contractor identities housed in contractor directories will not be HSPD12 based identities—that is unless they are replicas of NASA or other Federal identities. Therefore NASA can not allow access control decisions to be based upon these identities (again, per [M0524]). Even if by policy these identities could be used they would not be in IdMAX, and thus are subject to the same problems as discussed with external PIV credentials.

NASA's current tactics for relationships are defined here:

G2G: We will accept external PIV credentials for authentication and will implement changes to IdMAX and authentication sources as necessary to create a NASA identity in IdMAX that is bound to the PIV credential issued from another agency.

B2G: We will continue Active Directory trusts only as a deviation. These trusts will be one way whereby the contractor consumes NASA identities for authentication purposes. NASA will not consume contractor identities for authentication. eAuthentication authentication has less limitation than AD, but is still reliant on the IdMAX based identity.

NASA will consider Federation as the architecture and technology matures. A primary problem NASA will face with Federation is our NAMS requirement for a workflow approval process for authorization policy based upon an individual. In other words, any external user requires a static one-to-one mapping to a user object in IdMAX, and that user must be expressly granted access to any system.

C2G: We have no firm requirements yet. NASA does not usually interact with citizens, except to provide public data available to anyone.

6. LEVEL OF ASSURANCE

According to [M0524], the scope of HSPD12 extends to persons on Federal facilities accessing systems on Federal facilities. For use cases out of the scope of HSPD12, NIST guidance, especially [SP80063], applies.

[SP80063] provides guidance for the authentication credential's level of assurance (LoA) which agencies should require to access systems of differing security categorizations. (See NIST, FIPS Publication 199:

Standards for Security Categorization of Federal Information and Information Systems. [FIPS199])

NASA has about 20,000 civil servants and about 80,000 affiliates, ranging from contractors under formal contract to University partners under loose agreements. Partners are world-wide, including citizens of designated countries. The issues regarding identity vetting of foreign nationals are extensive and will not be addressed in this paper. Of note is that export control rules can restrict the types of credentials that can be issued to our foreign partners, depending on their citizenship and the country from which they are accessing NASA systems.

According to [SP80063], a [FIPS199] Low system can accept a level 2 credential (userid/password), whereas a Moderate system requires a two-factor credential. [M0524] requires PIV credentials for all systems accessed by federal workers on federal facilities. This puts us in the odd position of requiring a higher LoA for the people we trust the most, e.g. NASA civil servants accessing applications from NASA-managed devices while on a NASA network, than the people we know the least about, e.g. IT Remote users with claimed identities accessing applications from unknown devices across the open Internet.

Given the extensive external partners at NASA, LACIT developed a framework for the credentials that could be used by Remote IT Users. The Application Integration Standard addresses the considerations for Application Owners in determining which credentials to require for access to their applications.

What is important to note is that NASA will be accepting a variety of credentials for the foreseeable future: userIDs/passwords, RSA tokens, and PIV smartcards. As stated earlier, NASA has two primary LACS services, the NAF and eAuthentication, and one major alternate service: Agency RSA. Agency RSA accepts RSA tokens; the NAF accepts either UserId/password or Smartcards; and eAuthentication accepts any of the three credentials. Each of these LACS services will support both low- and moderate-risk systems, including those accessed remotely. This situation drives NASA's requirement to be able to determine the LoA of the credential used as part of ABAC. We want to grant access to an application not only based on who the person is and the community to which s/he belongs, but also based on the credential used to gain access -- we sometimes refer to this requirement as Authorization based on the strength of Authentication.

The Security Access Markup Language (SAML) protocol used by NASA's eAuthentication implementation supports authorization based on the LoA of the credential presented. This provides NASA the flexibility to provide access to applications at all FIPS risk levels through a single LACS, with assurance level restrictions set on a per-application

basis. A user can log in to an eAuthentication-enabled application using a userID/Password to a Low risk application, and be passed using the SAML token to any other low risk application. Upon attempting to access the first moderate application, the user will be prompted that a stronger credential is required.

Within AD, level of assurance is not so easy to determine. Once a user has authenticated, it is the Kerberos ticket that is exchanged for access to AD-enabled systems. The ticket does not provide information about the type of credential used to authenticate. This leaves NASA with a predicament: either we presume that all AD-based authentication is only at Assurance level 2, or we lock AD down to smartcard-only authentication in order to ensure Assurance Level 4 authentication. Neither of these options is practical.

Presuming that all AD-based authentication is only at Assurance level 2 is problematic, since the Worker with an AD-provided Kerberos ticket most likely got it through smartcard login to the desktop. Yet, locking down AD to smartcard-only authentication is also impractical, since there are multiple use cases we must support for workers who either do not have a smartcard, or do not have access to a desktop with the required reader and middleware to use the smartcard.

As we move forward, we will also need to distinguish between PIV smartcards and other smartcards. Since a PIV smartcard has an assurance level of 4, and other smartcards may only have an assurance level of 2 or 3, this distinction becomes important for access to higher-risk systems. NASA expects to issue non-PIV smartcards in the future to NASA temporary employees including summer interns, visiting scientists, and short-term contractors. As stated before, NASA also plans to accept smartcards issued by contractors and other entities, once a process for certification and acceptance is established. It is therefore a requirement that the authentication ticket, either SAML or Kerberos, be able to provide information about credentials, certificate types, and possibly even certificate issuer information.

NASA has advocated with Microsoft and the MIT Kerberos consortium to include the ability to determine LoA in the Kerberos ticket presented by the user.

7. NASA'S IDENTITY AND ACCESS MANAGEMENT FUTURE

Fiscal Year 2009 will see implementation of many components of NASA's identity, credential, and access management architecture, particularly in the LACS arena. As of October 27, 2008, NASA had issued PIV smartcards to over 90 percent of its civil servant and contractor workforce. NASA has begun migration to the Agency AD

service, the NAF, and migration will be completed at the end of FY 2009. All NASA [FIPS199] High and Moderate systems will be integrated with NAMS by the end of FY 2009, and LACS integration with the NAF and eAuthentication will be well underway. Both the NAF and eAuthentication will be smartcard-enabled in FY 2009. Smartcard-enablement of desktops will follow closely behind migration to the NAF, and will be largely complete at the end of the fiscal year. The Agency RSA service will be implemented as well, consolidating the existing RSA installations, and tying token credentials more closely to vetted identities in IdMAX.

We plan to continue implementing modules of our Business Architecture. As part of this effort, we plan to implement an initial registry of simple communities, and register existing Basic Levels of Entitlement (registering access permissions from established communities to assets). We also intend to design and implement the process for ingesting PIV credential information from NASA partners who are issued PIV smartcards from other Agencies into our IdMAX and LACS.

We have referenced our need to perform authorization based on strength of authentication. We have also pointed out that this information is not available in current implementations of Microsoft Kerberos. We, along with several other large organizations, have asked Microsoft to implement this feature. Specifically, we would like to have dynamic binding to a security group per the level of assurance at time of authentication. This information would then be passed to policy decision points in the Kerberos ticket—in the Privilege Attribute Certificate (PAC) portion of the ticket. We would like the ability to AND access control conditions to include both the normal static group membership evaluation and this level of assurance attribute. Further, we are asking for granularity in the authentication process to include group membership assertion mapped to assertion of particular OIDs in the certificate policy extension. Of prime importance, the PIV Authentication Certificate as specified in [COMMON] asserts OID id-fpki-common-authentication. This OID satisfies level of assurance 4, per [RELYING].

We have also referenced the need for NASA to ingest identities into IdMAX as a prerequisite to accepting credentials from other organizations. Though there is prescriptive guidance for NASA to accept external PIV credentials for authentication, we still have a requirement to approve of the individual's relationship with NASA prior to granting any form of access to NASA resources. NAMS provides the workflow to define this relationship to NASA. The workflow includes the roles of requester, sponsor, and approver. Role holders are subject to rules enforcing separation of duty. For external workers there is also a need for more data to be captured as part of this

process. We need information from the individual's PIV authentication certificate to include Subject Alternate Name—UPN. The UPN may contain any arbitrary user name. This user name and/or the PIV certificate Subject DN will have to be retained within IdMAX and be made available for evaluation during the authentication process. If a UPN is to be consumed then, as previously motioned, the suffix matter of the UPN will also have to be retained. All PIV credentials are issued by a PKI Shared Service Provider and are anchored under the FPKI Common Policy Root, and thus will be usable. Any future requirement to accept non-PIV credentials will require full evaluation of trust criteria prior to accepting any other anchors.

NASA's acceptance of other non-PIV smartcards issued by industry is dependent on federal policy and process for certifying and accepting such credentials. Our industry partners are very interested in NASA accepting their credentials. NASA would prefer not to create its own policy in this area; this policy should be consistent across the federal government.

As NASA has worked to implement robust ABAC, we have found that the technology, while non-trivial to implement, is less complex than the policy. HSPD12, perhaps unwittingly, was a catalyst for NASA to implement an architecture at the business and system levels that can support ABAC in the near future. And, we could not hope to solve our policy and technology needs for ABAC without our Zachman framework.

8. ACKNOWLEDGEMENTS

We would like to thank Dale S. Mietla, Synfusion, Inc. for his extensive work with NASA to understand and implement the Zachman framework for this project.

9. REFERENCES

[HSPD12] Homeland Security Presidential Directive (HSPD)-12, : Policy for a Common Identification Standard for Federal Employees and Contractors. Available at <http://www.whitehouse.gov/news/releases/2004/08/20040827-8.html>

[Clinger-Cohen] Clinger-Cohen Act, Public Law 104-106, section 5125, 110 Stat. 684 (1996).

[Zachman] J. Zachman et al, Zachman International Enterprise Architecture. Multiple publications available at: <http://www.zachmaninternational.com/index.php>.

[M0524] Office of Management and Budget, Memo M-05-24, Implementation of Homeland Security Presidential Directive (HSPD) 12 – Policy for a Common

Identification Standard for Federal Employees and Contractors. Available at <http://www.whitehouse.gov/omb/memoranda/fy2005/m05-24.pdf>

[SP80063] National Institute of Standards and Technology (NIST), Special Publication 800-63, Electronic Authentication Guideline. Available at <http://csrc.nist.gov/publications/PubsSPs.html>

[FIPS201] NIST, Federal Information Processing Standards (FIPS) Publication 201-1: Personal Identity Verification (PIV) of Federal Employees and Contractors. Available at <http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf>

[FIPS199] NIST, FIPS Publication 199: Standards for Security Categorization of Federal Information and Information Systems. Available at <http://csrc.nist.gov/publications/fips/fips199/FIPS-PUB-199-final.pdf>

[COMMON] X.509 Certificate Policy for the U.S. Federal PKI Common Policy Framework, Version 3647 - 1.4, August 13, 2008. Available at <http://www.cio.gov/fpkipa/documents/CommonPolicy.pdf>.

[RELYING] Implementation Guidance for Relying Parties Using the Common Policy Root, February, 2007. Available at <http://tingwww.cio.gov/fpkia/documents/RPguidance0207.pdf>



Identity, Credential, and Access Management at NASA, from Zachman to Attributes

Office of the Chief Information Officer

Corinne Irwin
Dennis Taylor

VISION: Integrated, secure, and efficient information
technology and solutions that support NASA



Agenda

Office of the Chief Information Officer

- Introduction
- Zachman Framework
- Identity Trust
- Level of Assurance
- Conclusions



Introduction

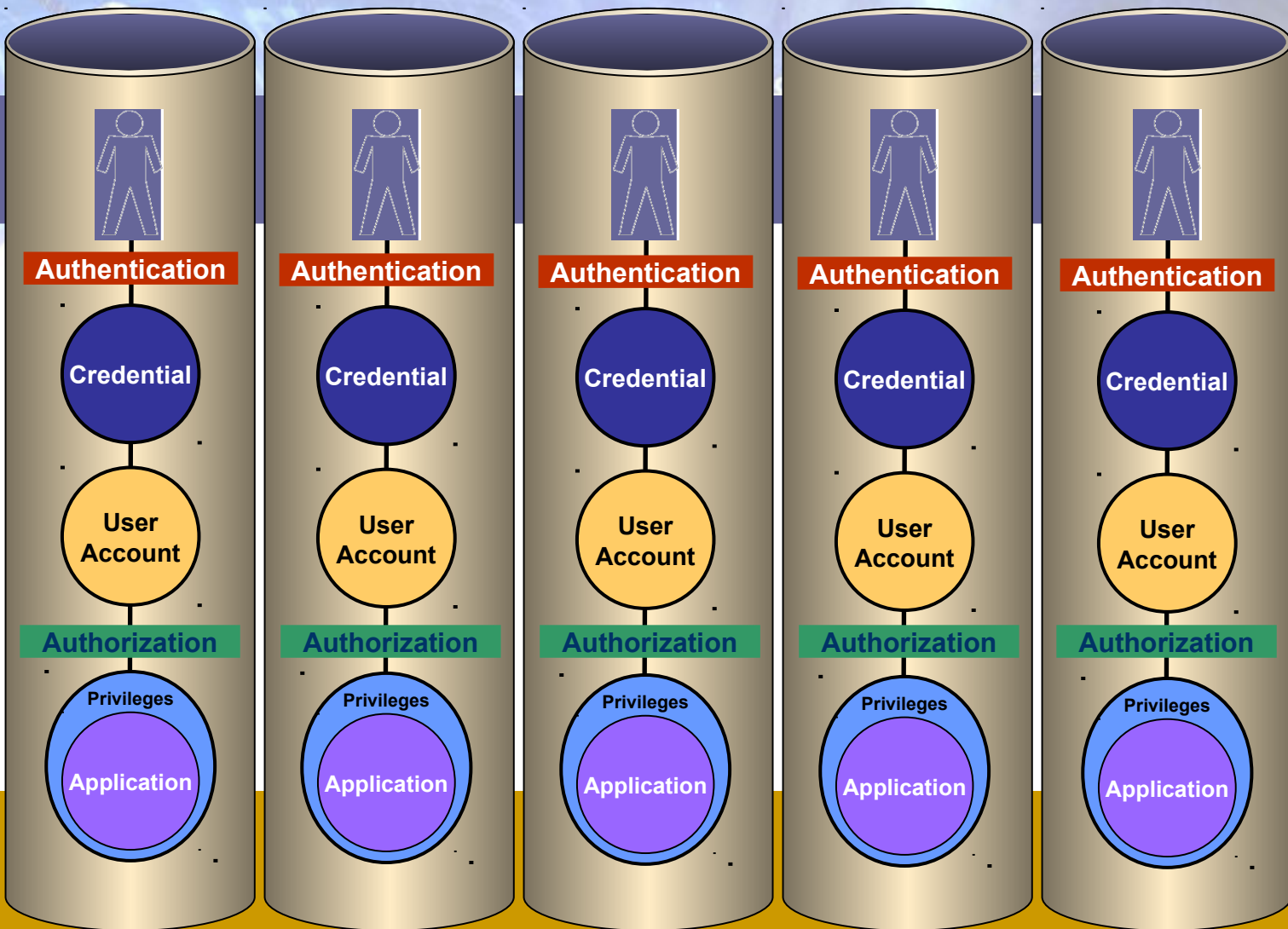
Office of the Chief Information Officer

- NASA includes:
 - 20,000 civil servant employees
 - 80,000 on-site contractors
 - Additional partners world-wide
- NASA's system/application landscape includes:
 - 3,000 applications, most built in-house
 - Mission control, research labs, product fabrication, more
 - Every flavor of every operating system, hardware, software....
- Historically, NASA has been:
 - Highly decentralized
 - Autonomous Centers with a B-to-B network infrastructure
 - Characterized by weak CIO governance
- HSPD-12 helped us:
 - Implement a robust Identity, Credential, and Access Management Architecture
 - Position NASA for use of ABAC and RBAC



Traditional ICAM - Parallel Environments

User Identity



fficer

Credit: Owen Unangst, USDA



Future ICAM – Enterprise Single-SignOn

(User Perspective)

Office of the Chief Information Officer

User
Identity



Authentication



Authorization



Privileges

Credit: Owen Unangst, USDA



Enterprise Architecture

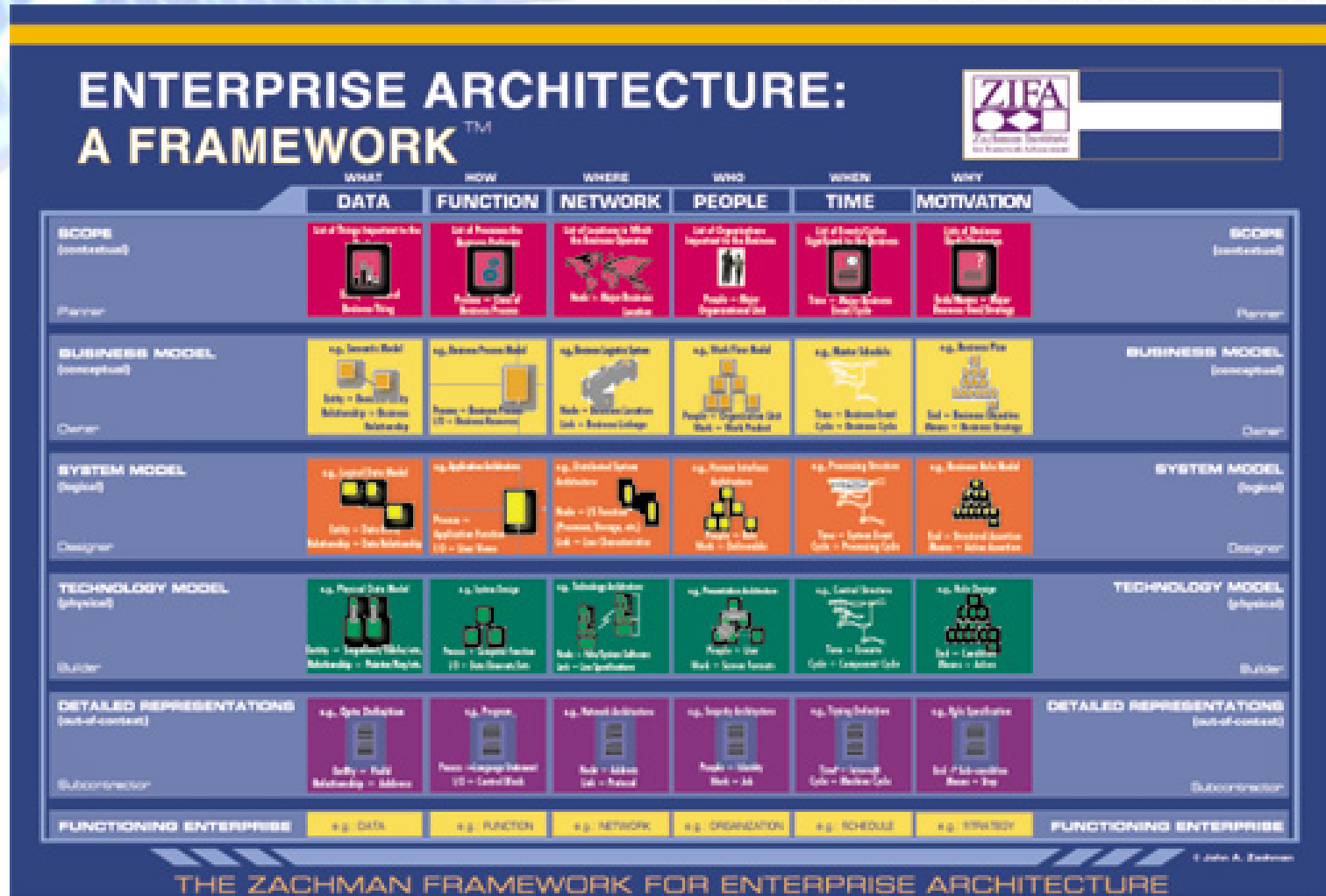
Office of the Chief Information Officer

- Enterprise Architecture (EA) frameworks provide structure for developing complex, integrated systems
- Ideally, one:
 - Develops an As-Is architecture
 - Develops a To-Be architecture
 - Performs gap analysis
 - Develops plan to move toward the To-Be architecture
- NASA used Zachman to develop its ICAM architecture starting in 2006
- At a federal level, the ICAM Sub-committee was tasked in FY2009 with developing the segment architecture for Federal ICAM.



Zachman Framework

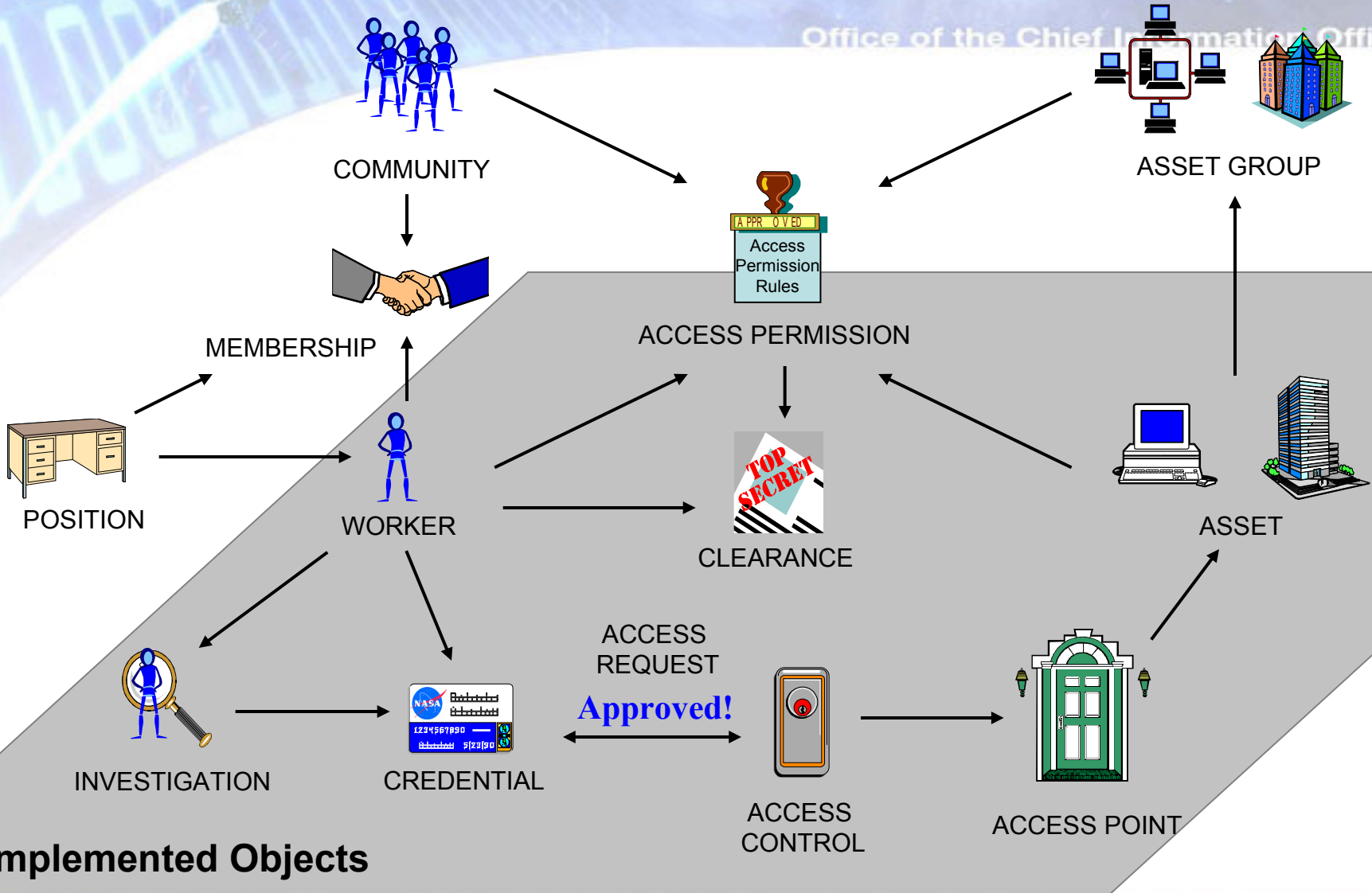
Office of the Chief Information Officer





The Really Big Picture

Office of the Chief Information Officer

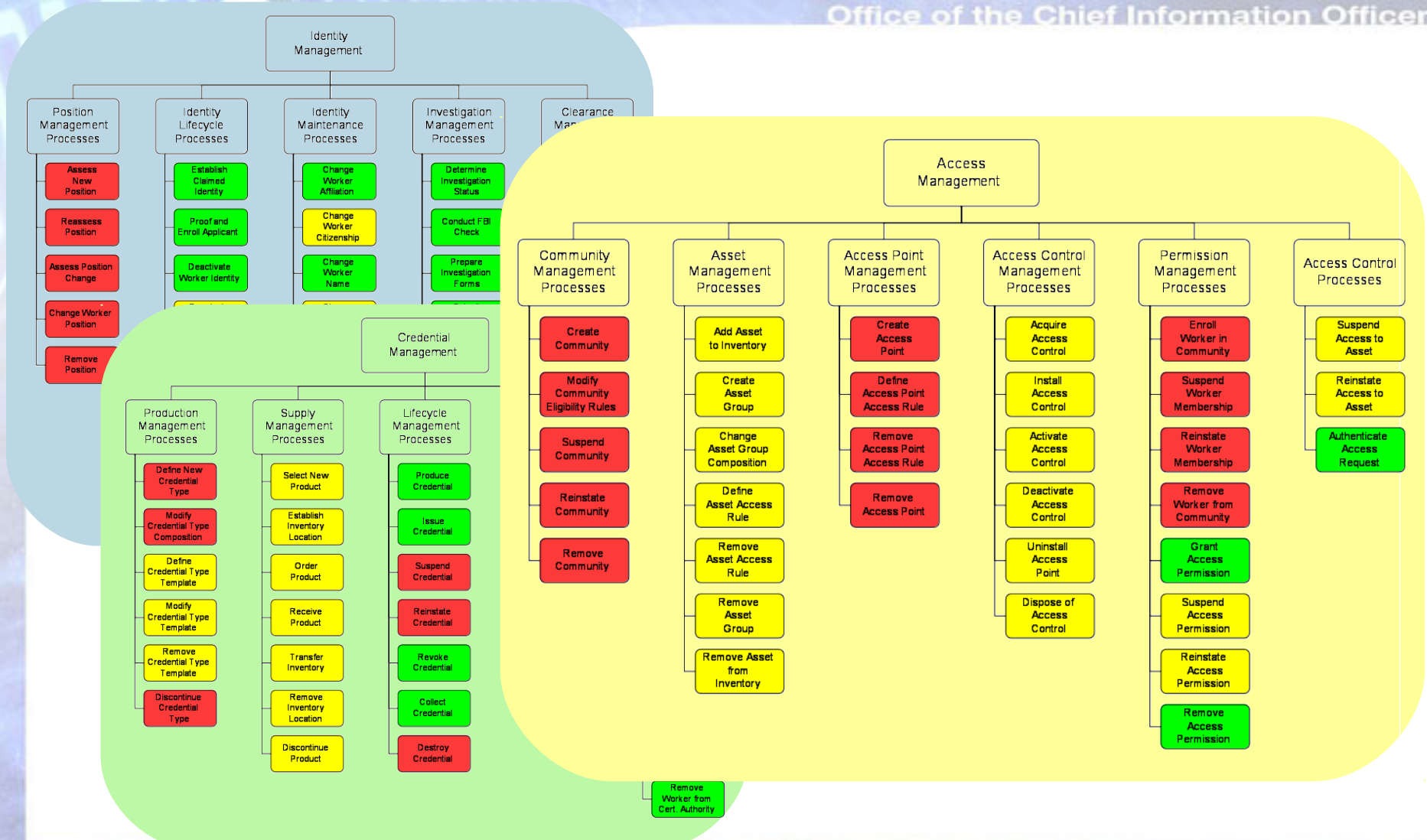


Implemented Objects



ICAM Business Processes

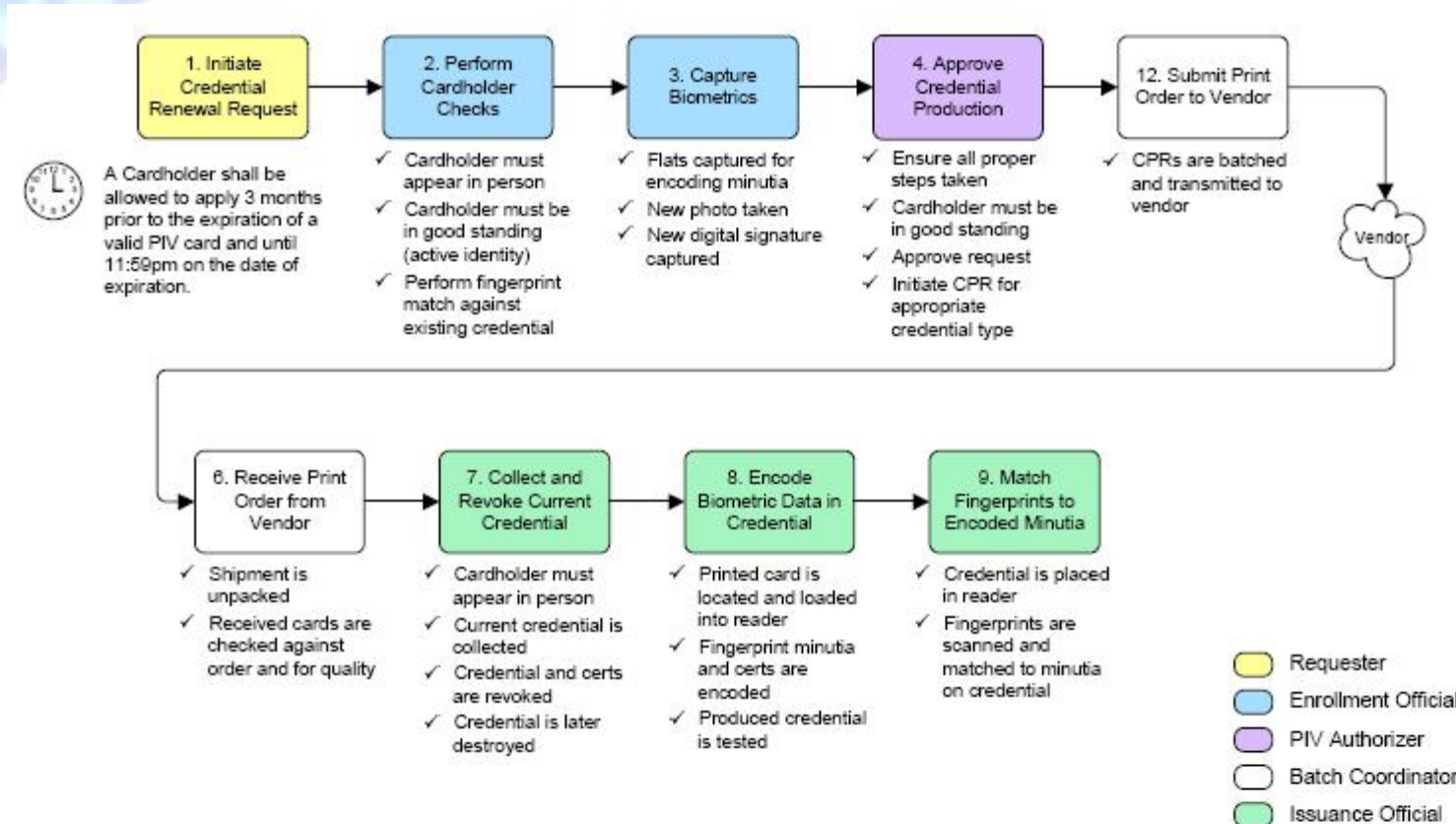
Office of the Chief Information Officer





Example Business Workflow: Badge Renewal

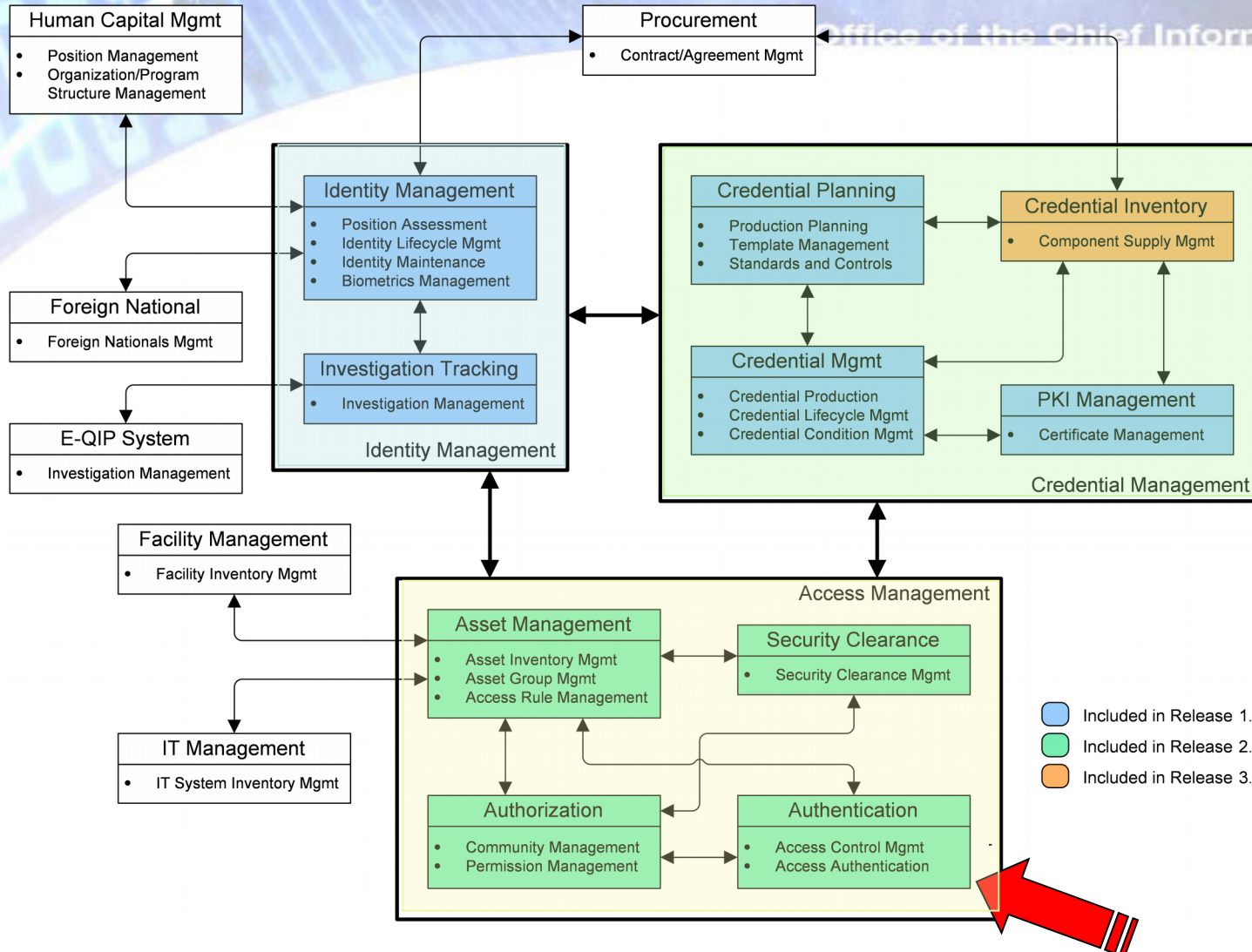
Office of the Chief Information Officer





ICAM Systems Model

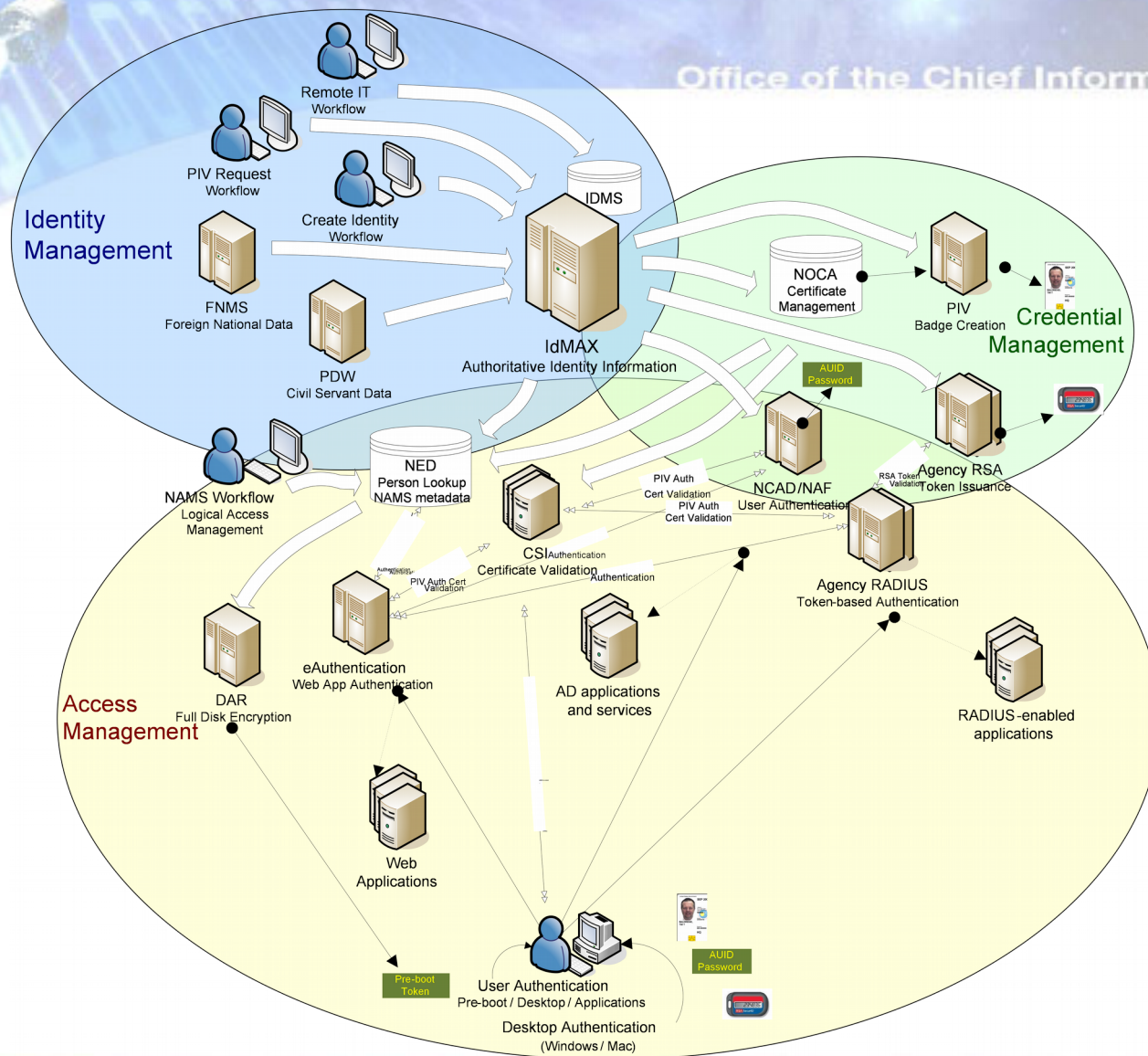
Office of the Chief Information Officer





Technology Model

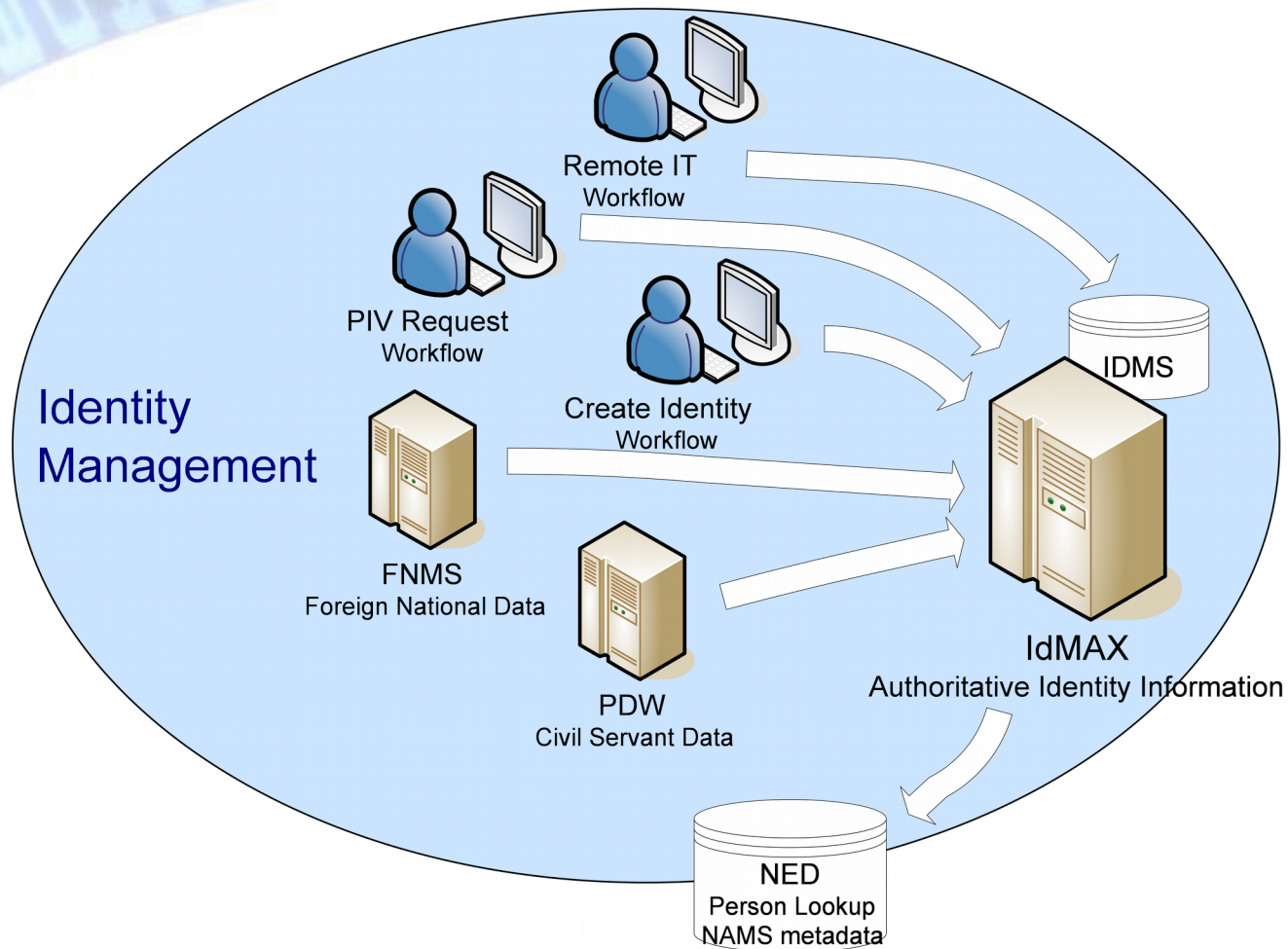
Office of the Chief Information Officer





Identity Management

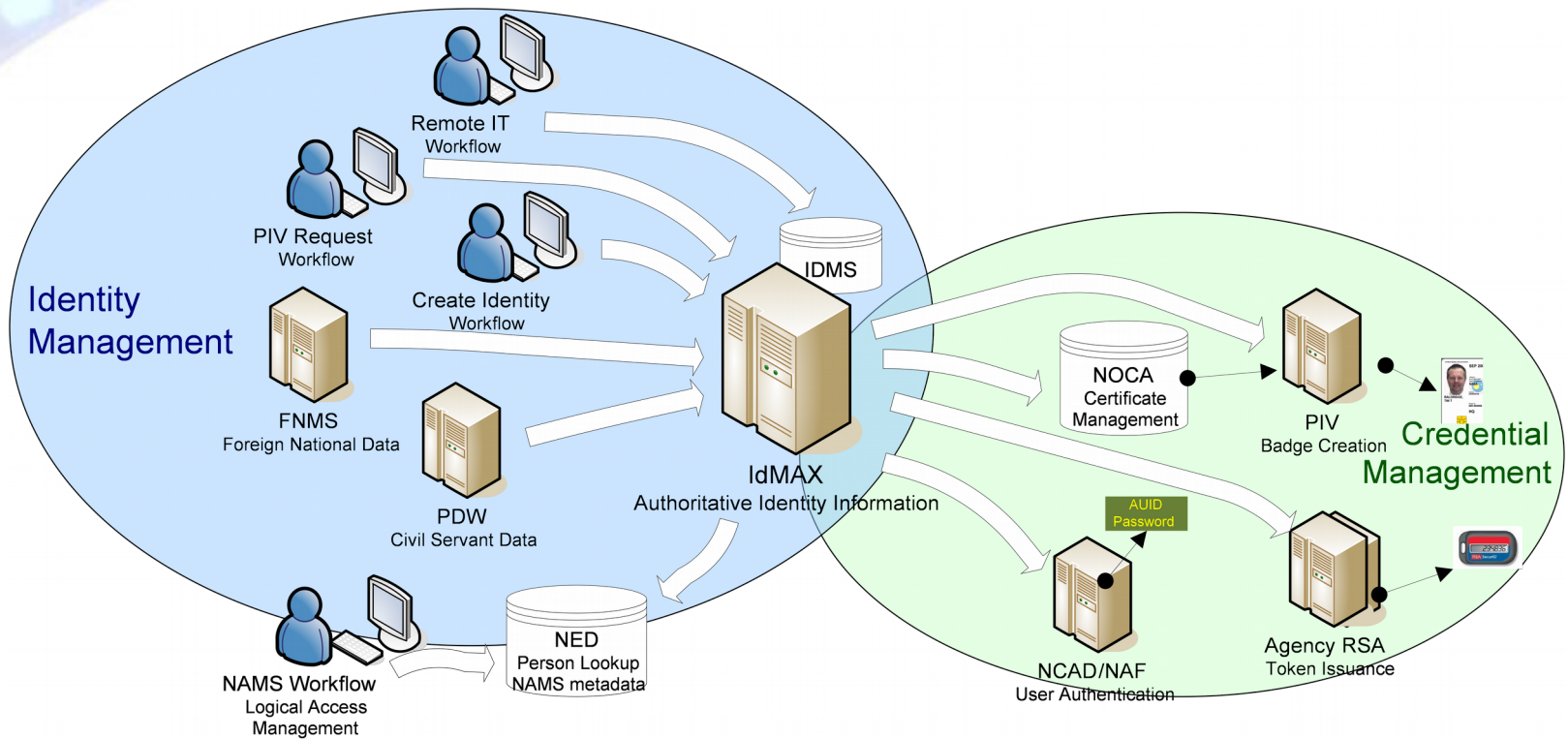
Office of the Chief Information Officer





Identity and Credential

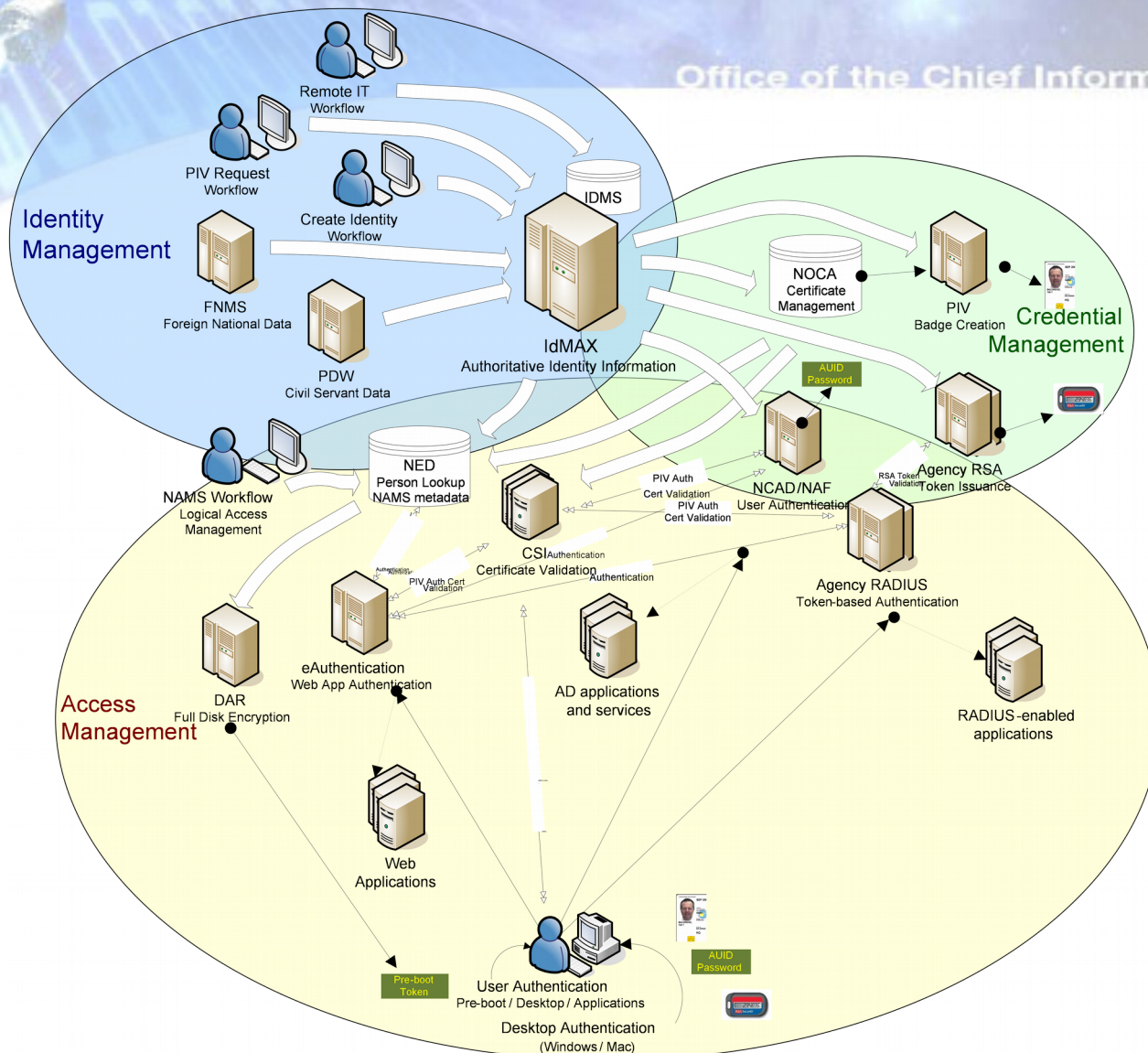
Office of the Chief Information Officer





Full ICAM Model

Office of the Chief Information Officer

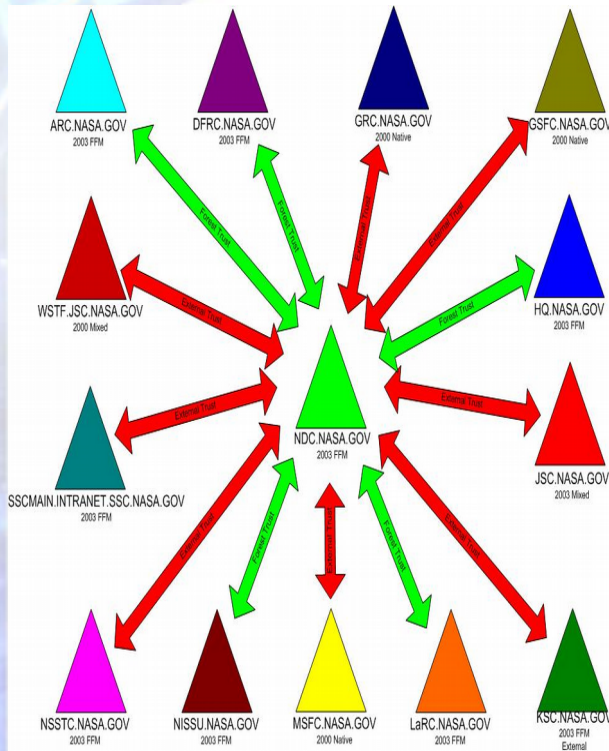




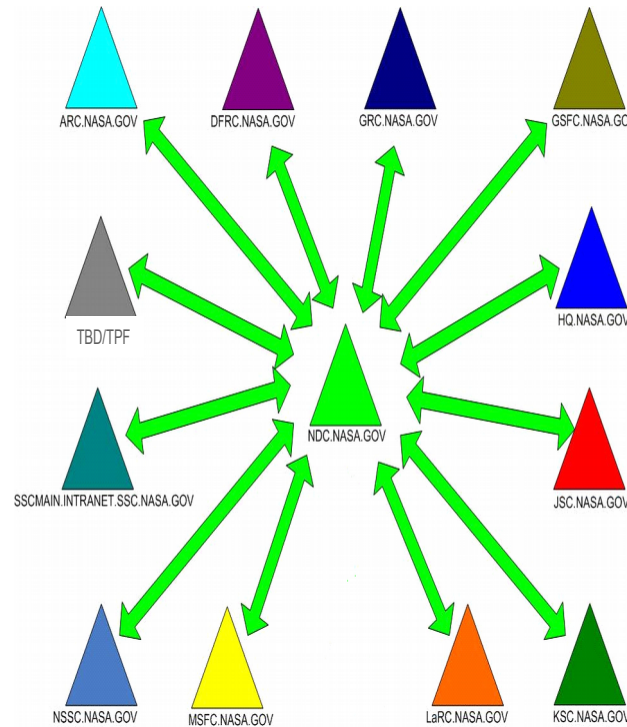
NCAD—Active Directory Forest and Domain Structure

Office of the Chief Information Officer

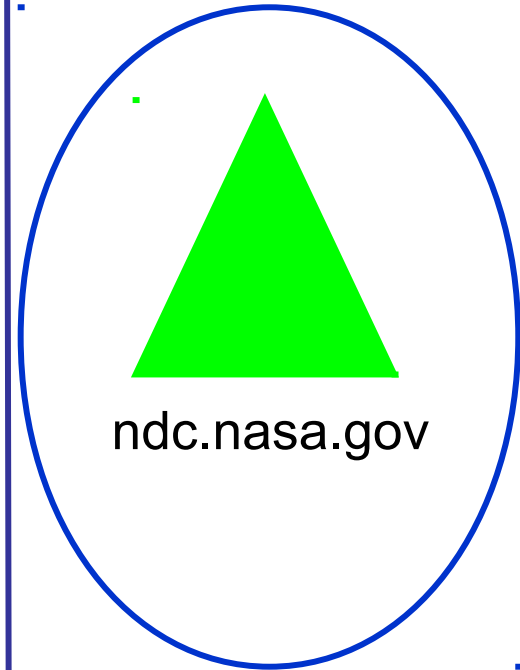
As-Is Structure



To-Be CDR Structure Supports Migration Activities



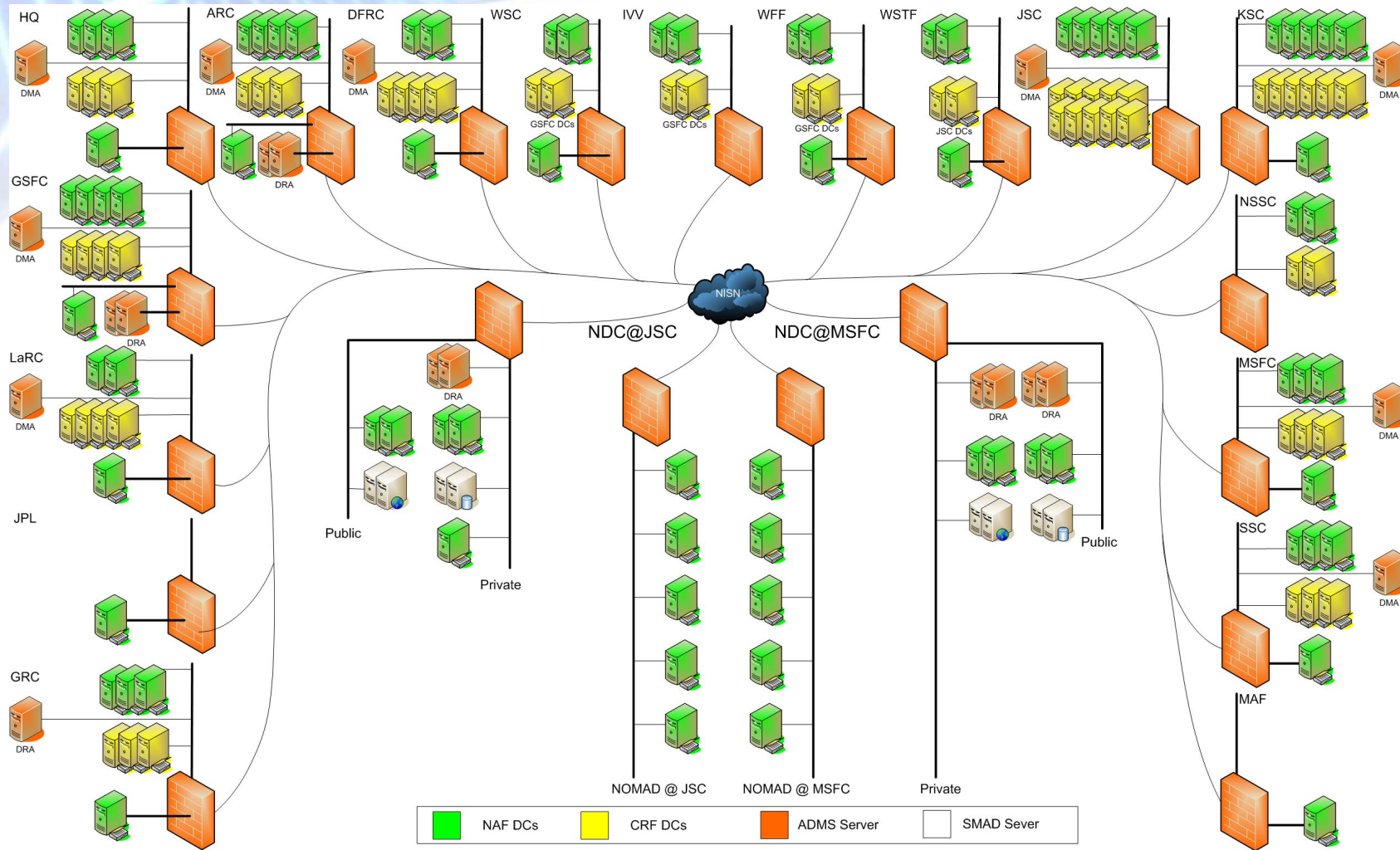
To-Be Structure: One Forest One Domain





NCAD—Interim/Current Topology

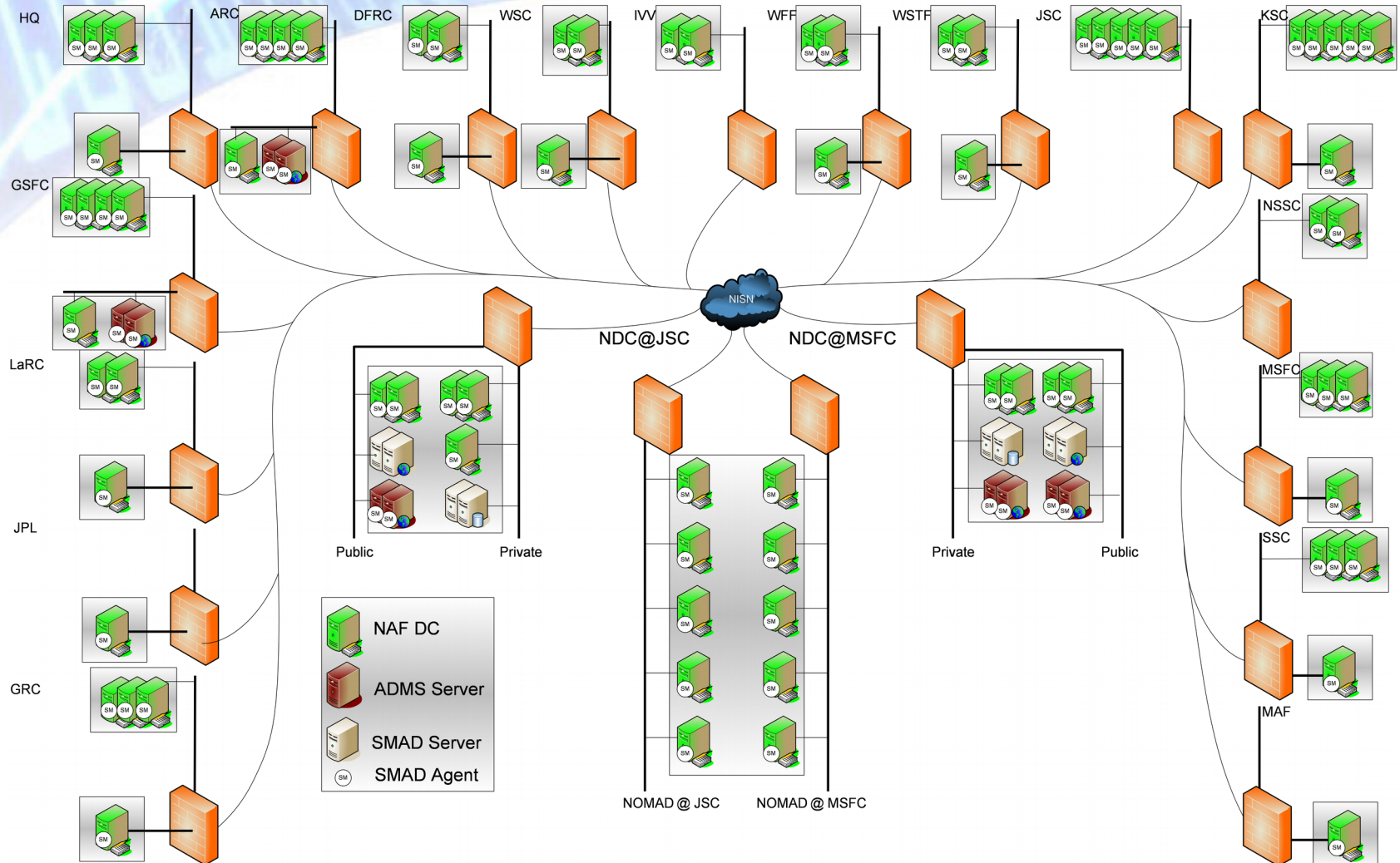
Office of the Chief Information Officer





NCAD TO-BE Topology

Office of the Chief Information Officer





AD Consolidation Summary

Office of the Chief Information Officer

- Finally top-down versus grass-roots
- Formal project methodology
 - System Engineering Methodology per NASA NPR 7123
 - Project Management Lifecycle per NASA NPR 7120.7
- Detailed large project plan with linked tasks
 - Project plan maintained by an experienced project scheduler
- Formality in test-set development
 - SIR-TP, SATS, ORTS, all with traceability
- Project Manager experienced in large engineering development; experienced program managers for two major contractors leading effort
- Brought in personnel with experience in similar consolidation efforts at Army, AF, and Navy-Marines
- All eggs in one basket argument...SIEM



Identity Trust

Office of the Chief Information Officer

- FIPS 201 tells NASA to accept any valid PIV card (from any Federal Issuer) for authentication
- Simple enough, except for technology and policy
- Only NASA has the right to determine particular access rights —and we have a heavy duty system (NAMS) for this purpose
- Internally, was a peck of AD trusts (many hundreds), but still wasn't enough to be ubiquitous
- Now, with NAF and eAuth, we are a single entity on the inside...with ubiquitous authentication service
- No trusts necessary on the inside (NASA trusts itself)



Identity Trust

Office of the Chief Information Officer

External Trusts (G2G)

- We will accept (To-Be) the PIV token from anywhere
 - Need to capture token information for NAF and NED
 - Need to provision IdMAX
 - UPN suffix provisioning in NAF
 - PKI trusts paths and revocation checks
- We agree with concept of federated identities
 - Federation allows us to proxy In-person verification
 - Need to link identity to relationship with NASA
 - Need further Federal Guidance
 - Being worked within FPKI/ISIMC/ICAM subcommittee

External Trust (B2G)

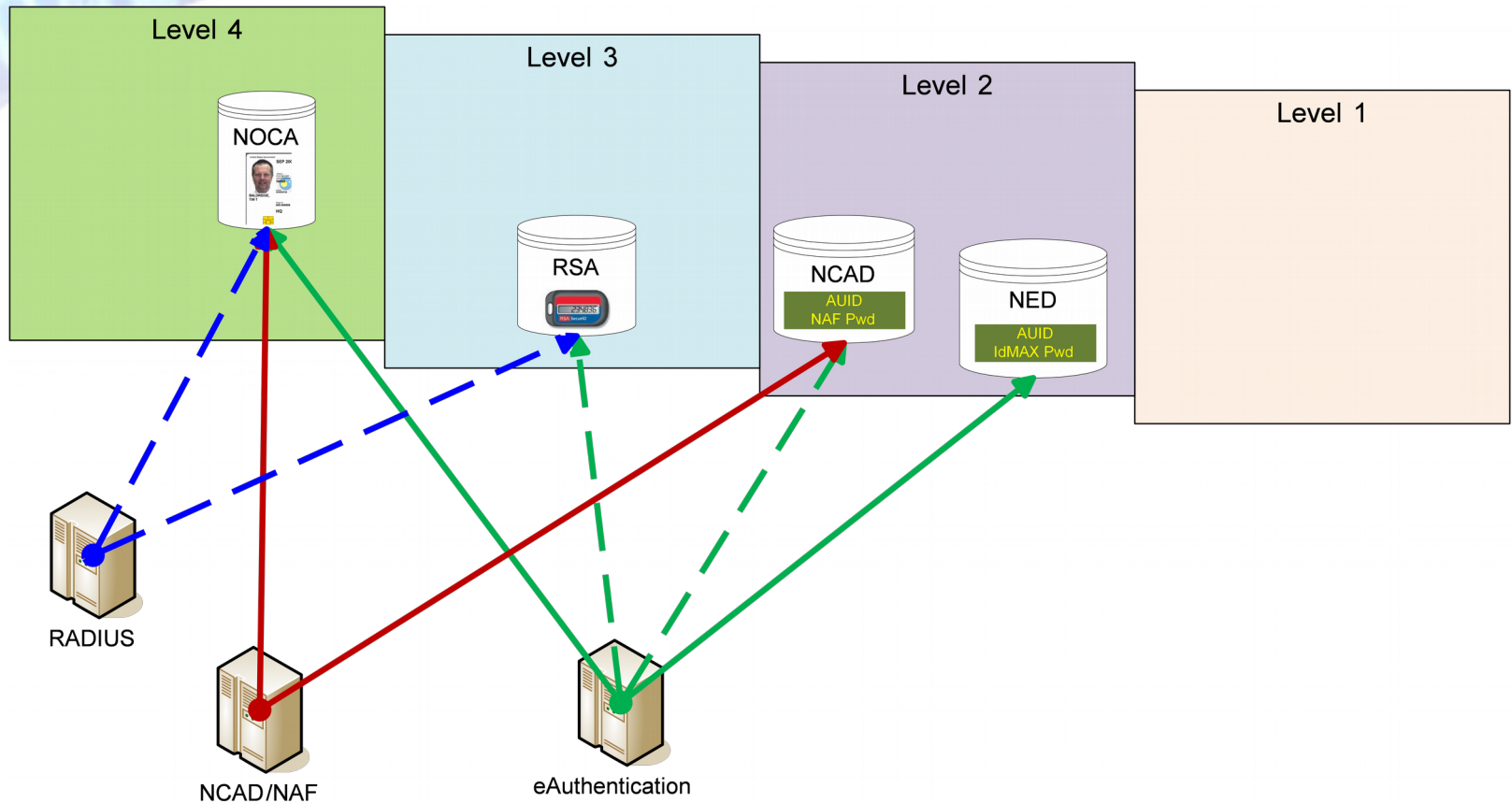
- To Be worked sometime after G2G

- More work needed on this



LoA Introduction: Tokens

Office of the Chief Information Officer





LoA and HSPD-12 (To-Be)

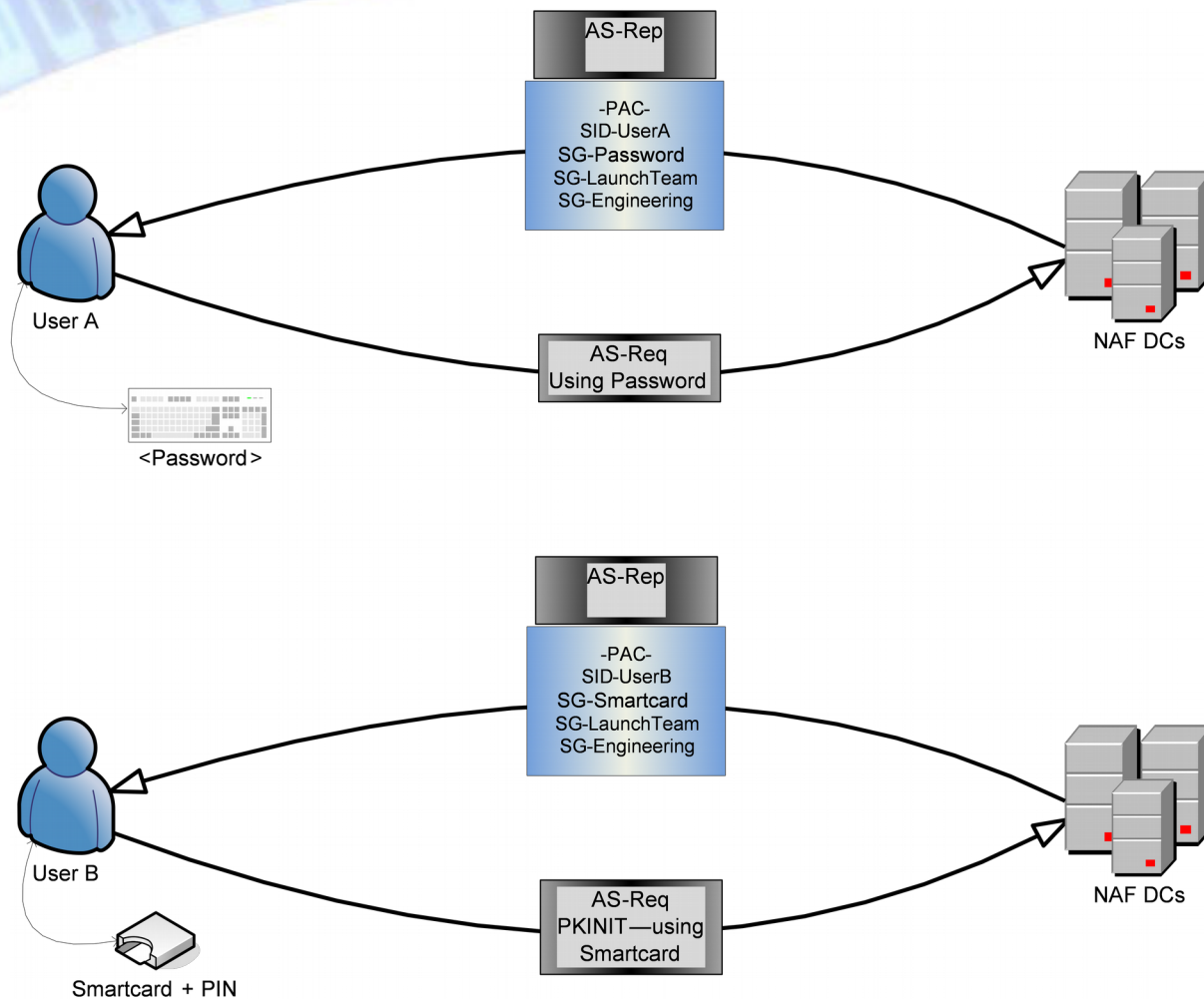
Office of the Chief Information Officer

NIST 800-63 Requirements with M-05-24 and M-07-16 Overlays					
Accessed From	Application Type	Minimum Acceptable Credentials	Acceptable Identity Check		
			Self proclaimed identity	ID Check	FIPS 201 PIV process
Non-Federally controlled facility	Public	Anonymous UserID/Password	Acceptable	Acceptable	Acceptable
	FIPS 199 Low	UserID/Password		Required	Acceptable
	FIPS 199 Moderate	Two-factor, such as: PKI soft certificate RSA Token		Required	Acceptable
	Access to PII, FIPS 199 High	Hard-crypto token, such as: RSA Token PIV Auth Certificate Other Smartcard		Required	Acceptable
Federally controlled facility	Any Application	PIV Auth Certificate (Smartcard)			Required



Missing—Capture of LoA on Logon

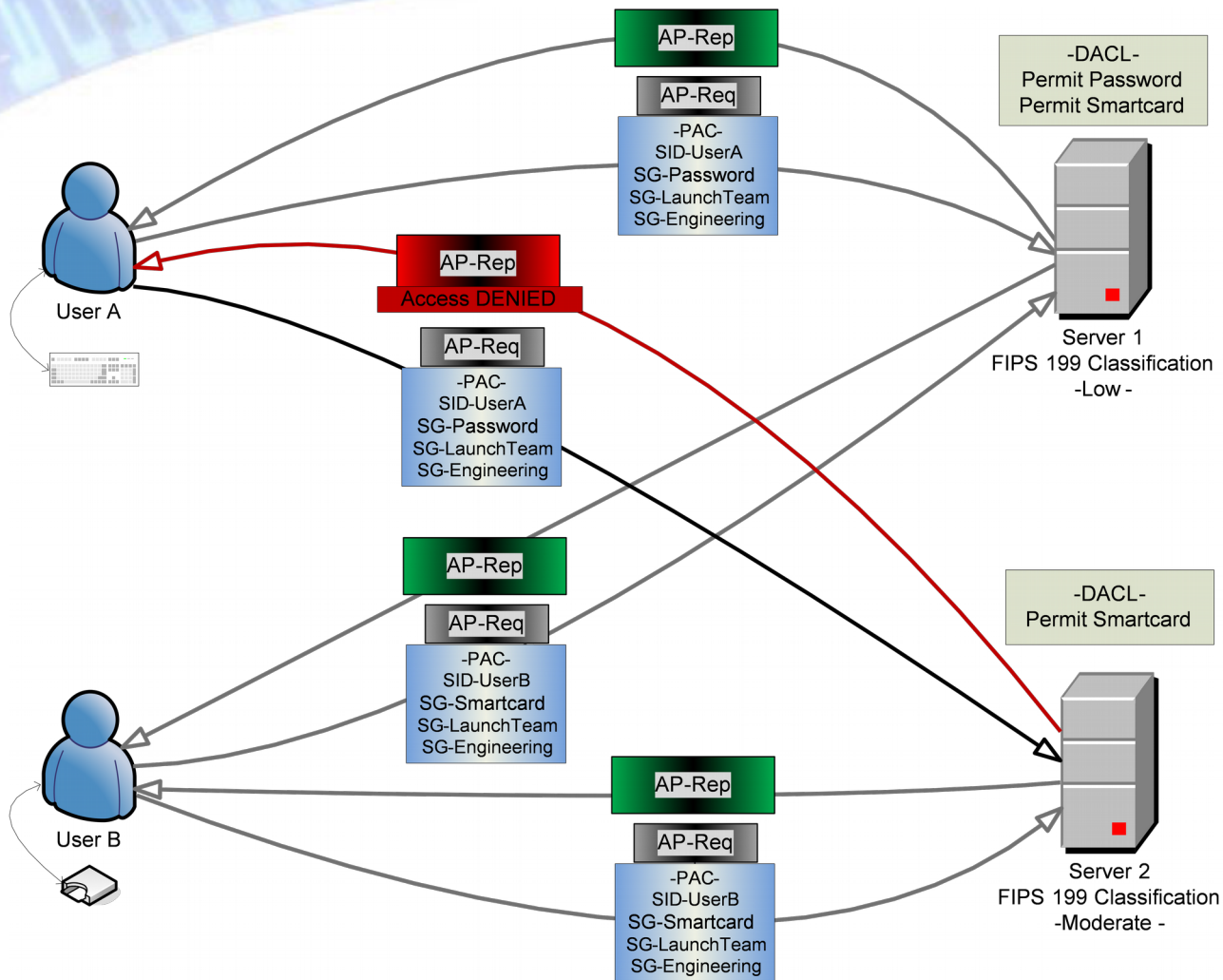
Office of the Chief Information Officer





Missing—AuthZ based upon LoA

Office of the Chief Information Officer





LoA Summary

Office of the Chief Information Officer

- We are going to be using a mix of primarily passwords and smartcards for a long time
- We need our authentication service to provide an LoA attribute to our authorization mechanism
 - Authorization based upon strength of authentication
- Our eAuth service (based upon Sun Access Manager) can provide this attribute through SAML like structures
- We need Microsoft Active Directory to provide a similar functionality in their logon (KINIT, PKINIT) and resultant PAC authorization data
- We need capability to map particular policy OID to security group
 - id-fpki-common-authentication means PIV card (only real measure)



Conclusions

Office of the Chief Information Officer

- A well-developed Enterprise Architecture is essential to ICAM implementation
- NASA must implement Position and Community Management modules in order to support robust ABAC
- Integrated data flow means data is only authoritative at the source, and changes can only occur at the source
- Identity federation and LoA require additional maturity in the market
- Technology is sometimes tricky, but politics is harder!
- Single sign-on is a strong motivator for migration



Backup

Office of the Chief Information Officer

VISION: Integrated, secure, and efficient information technology and solutions that support NASA



Use Cases

Office of the Chief Information Officer

A Worker with

- 1 - a NASA PIV Card
- 2 - a Federal PIV Card
- 3 - a trusted smartcard
- 4 - a userid/password
- 5 - an RSA token
- 6 - a NASA-issued PKI soft cert
- 7 - a trusted PKI soft cert
- 8 - a trusted 3rd party credential

To Access

- 10 -- a resource on the device being used
- 11 -- a desktop/console access
- 20 -- an integrated AD application
- 30 -- an eAuth-enabled application
- 40 -- a resource on a remote device (server)
- 50 -- Administrative functions
- 60 -- a system that restricts access based on attributes
- 100 -- a system that restricts access based on assurance level attributes
- 110 -- a RADIUS-enabled application/device
- 120 -- an RSA-enabled application/device

Where

- 10 - on the Center Institutional Network
- 20 - on a Mission/Specialized Network
- 21 -- on an isolated network
- 22 -- on a network with limited connectivity
- 30 - on another Center's network
- 40 - on the Public Internet

Using

- 10 - a NASA-Managed PC
- 11 -- a NAF-bound PC
- 12 -- a PC that is not NAF-bound
- 20 - a NASA-Managed Mac
- 21 -- a NAF-bound Mac
- 22 -- a Mac that is not NAF-bound
- 30 - a NASA-Managed Unix Box
- 40 - a trusted PC
- 50 - a trusted Mac
- 60 - a trusted Unix Box
- 70 - an unknown PC
- 80 - an unknown Mac
- 90 - an unknown Unix Box
- 100 - a NASA-managed PDA
- 110 - a trusted PDA
- 120 - an unknown PDA
- 130 - an unknown IP network device
- 140 - a server
- 150 -- a NASA-managed IP network device

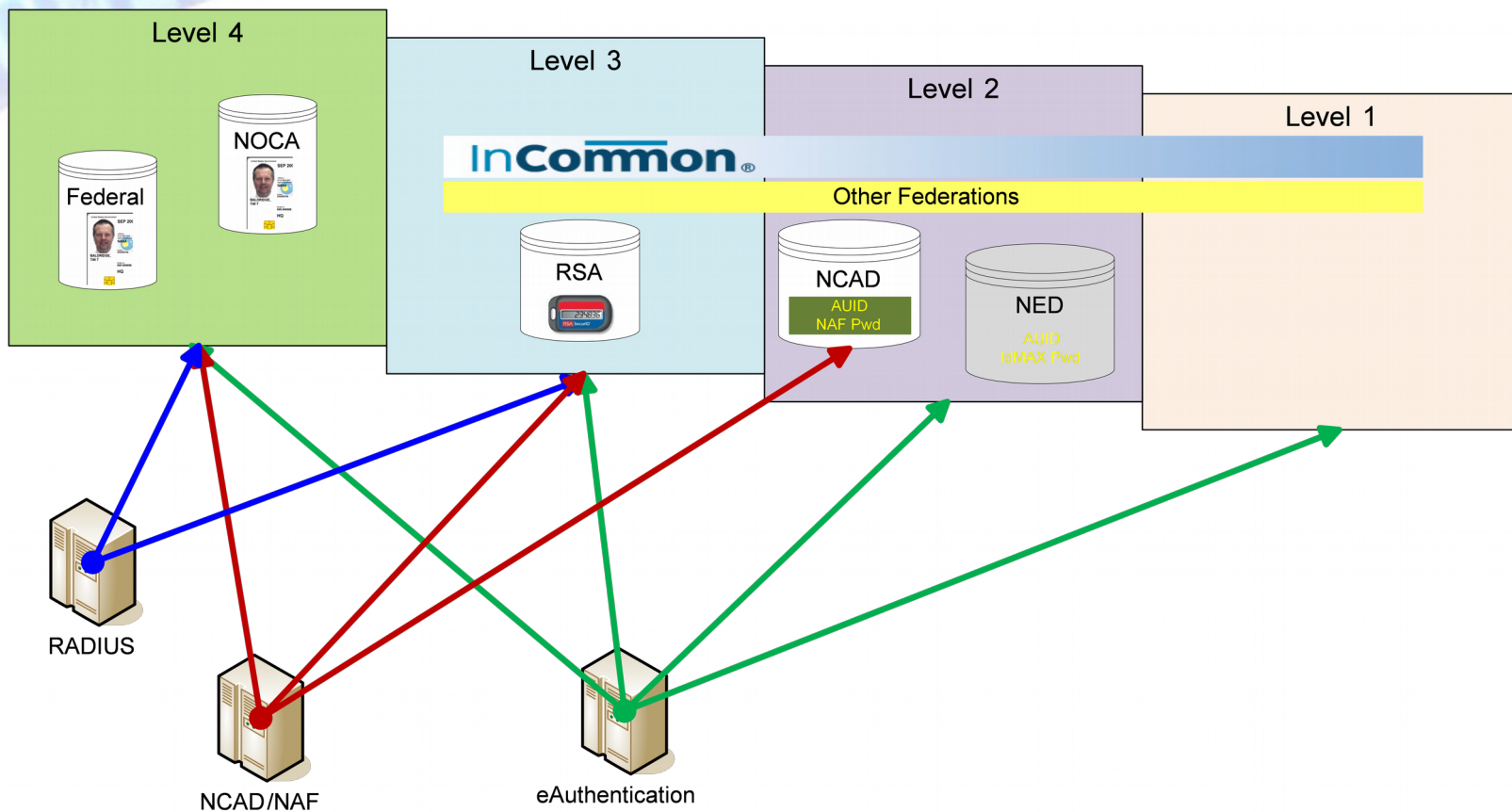
When

- 1 -- during normal operations (24x7x365)
- 2 -- during a COOP event
- 3 -- during a DR event
- 4 -- when the network service is unavailable
- 5 -- when the validation service is unavailable
- 6 -- when the authentication service is unavailable
- 7 -- during planned mission/specialized events
- 8 -- when the authorization service is unavailable



Future LoA Tokens

Office of the Chief Information Officer



Personal Identity Verification (PIV) Cards as Federated Identities – Challenges and Opportunities

Sarbari Gupta

Electrosoft

11417 Sunset Hills Road, Ste 228

Reston, VA 20190

703-437-9451x12

sarbari@electrosoft-inc.com

ABSTRACT

In this paper, we describe the challenges in using Personal Identity Verification (PIV) cards and PIV-like cards as federated identities to authenticate to US Federal government facilities and systems. The current set of specifications and policies related to the implementation and use of PIV cards leave a number of gaps in terms of trust and assurance. This paper identifies these gaps and proposes approaches to address them towards making the PIV card the standardized, interoperable, federated identity credential envisioned within Homeland Security Presidential Directive 12 (HSPD-12).

Categories and Subject Descriptors

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

General Terms

Management, Security, Standardization.

Keywords

Authentication, Smart cards, PKI, Assurance, Federal Bridge Certification Authority, Authorization.

1. BACKGROUND

Homeland Security Presidential Directive 12 (HSPD-12) entitled “*Policy for a Common Identification Standard for Federal Employees and Contractors*” was issued in 2004 to enhance security, increase Government efficiency, reduce identity fraud, and protect personal privacy by establishing a mandatory, Government-wide standard for secure and reliable forms of identification issued by the Federal Government to its employees and contractors [1] that:

+ Is issued based on sound criteria for verifying an individual employee’s identity

+ Is strongly resistant to identity fraud, tampering, counterfeiting, and terrorist exploitation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDtrust '09, April 14-16, 2009, Gaithersburg, MD

Copyright 2009 ACM 978-1-60558-474-4...\$5.00.

+ Can be rapidly authenticated electronically

+ Is issued only by providers whose reliability has been established by an official accreditation process.”

In response, the National Institute of Standards and Technology (NIST) published Federal Information Processing Standard (FIPS) 201 – “*Personal Identity Verification (PIV) for Federal Employees and Contractors*” [2] and several related Special Publications (found at <http://csrc.nist.gov/piv-program>) with detailed specifications on issuance and deployment of PIV cards to their personnel. The latest version of this standard is FIPS 201-1 published in March, 2006.

The goal of this standard is to support an appropriate level of assurance in conjunction with efficient verification of the claimed identity of an individual seeking physical access to Federal facilities and electronic access to government information systems. The PIV card is a smart card based digital identity container with a collection of identity credentials that provide graduated levels of assurance regarding the identity of the holder of the card.

When implemented and deployed by Federal agencies, the PIV card is envisioned to provide the attributes of security, authentication, trust and privacy using this commonly accepted identification credential.

1.1 PIV Documentation

NIST has published a suite of documents in support of PIV. These are identified below.

FIPS 201-1: Personal Identity Verification (PIV) of Federal Employees and Contractors. This document specifies the physical card characteristics, storage media, and data elements that make up the identity credentials resident on the PIV card.

SP 800-73-2: Interfaces for Personal Identity Verification (4 parts). This document specifies the interfaces and card architecture for storing and retrieving identity credentials from a smart card.

SP 800-76-1: Biometric Data Specification for Personal Identity Verification. This document describes technical acquisition and formatting specifications for the biometric credentials of the PIV system.

SP 800-78-1: Cryptographic Algorithms and Key Sizes for Personal Identity Verification. This recommendation identifies acceptable symmetric and asymmetric encryption algorithms, digital signature algorithms, and message digest algorithms, and

specifies mechanisms to identify the algorithms associated with PIV keys or digital signatures.

SP 800-79-1: Guidelines for the Accreditation of Personal Identity Verification (PIV) Card Issuers. This document provides guidelines for accrediting the reliability of issuers of Personal Identity Verification cards that collect, store, and disseminate personal identity credentials and issue smart cards.

SP 800-87-1: Codes for the Identification of Federal and Federally-Assisted Organizations. This document provides the organizational codes necessary to establish the PIV Federal Agency Smart Credential Number (PIV FASC-N) that is required to be included in the FIPS 201 Card Holder Unique Identifier (CHUID).

SP 800-104: A Scheme for PIV Visual Card Topography. This document provides additional recommendations on the Personal Identity Verification (PIV) Card color-coding for designating employee affiliation.

SP 800-116: A Recommendation for the Use of PIV Credentials in Physical Access Control. This document describes a risk-based approach for selecting appropriate PIV authentication mechanisms to manage physical access to Federal government facilities and assets.

1.2 PIV CREDENTIALS

The PIV card contains a number of mandatory and optional data elements that serve as identity credentials with varying levels of strength and assurance. These credentials are used singly or in sets to authenticate the holder of the PIV card to achieve the level of assurance required for a particular activity or transaction. A Personal Identification Number (PIN) is required to activate the card for privileged operations.

The mandatory credentials on the PIV card are:

- Cardholder Unique Identifier (CHUID)
- PIV Authentication Private Key and X.509 Certificate
- Biometric Object with cardholder fingerprints

The optional elements on the PIV card are:

- PIV Card Authentication Key (CAK) and X.509 Certificate (if CAK is asymmetric)
- PIV Digital Signature Private Key and X.509 Certificate
- PIV Key Management Private Key and X.509 Certificate
- Cardholder Facial Image

The reader is directed to [2] for further details on any or all of these credentials.

2. U.S. FEDERAL PKI and FIPS 201

In this section, we present a brief overview of the related Federal PKI policies to aid the understanding of the core thoughts presented in this paper.

The “X.509 Certificate Policy For The Federal Bridge Certification Authority (FBCA)” defines seven certificate policies to facilitate interoperability between the FBCA and other Entity PKI domains. The policies represent different assurance levels indicating the strength of the binding between the public key and the individual whose subject name is cited in the certificate, the mechanisms used to control the use of the private key, and the security provided by the PKI itself. Of these, the *Medium-HW* policy is of relevance to this paper.

The “X.509 Certificate Policy for the U.S. Federal PKI (FPKI) Common Policy Framework” governs the public key infrastructure component of the Federal Enterprise Architecture. It incorporates six specific certificate policies of which two are of direct relevance to this paper: *id-CommonAuth* or *id-CommonHW*.

FIPS 201-1 requires the PIV authentication certificate loaded on a PIV card to be issued under the *id-CommonAuth* or *id-CommonHW* policies or under a policy that is equivalent to the FBCA *Medium-HW* policy.

FIPS 201-1 includes a detailed set of requirements related to identity proofing, registration processes and security controls required to securely store, process, and retrieve identity credentials from the card. In many cases, the requirements levied by FIPS 201-1 are more stringent than the requirements stemming from one or both of the FPKI policies mentioned above. For the purposes of this paper, it is important to recognize the elements where the requirements of FIPS 201 differ from the policy requirements of these two FPKI policies. These are summarized in the table below:

Table 1 - Differences in Requirements

<u>FIPS 201-1</u>	<u><i>id-CommonAuth</i> or <i>id-CommonHW</i> policies</u>	<u>FBCA <i>Medium-HW</i> policy</u>
NACI has to be initiated for Interim PIV card. NACI has to be completed for full scope PIV card.	NACI not required for regular applicants. Only CA personnel are required to undergo background checks.	NACI not required for regular applicants. Only CA personnel are required to undergo background checks.
FBI fingerprint check required.	Fingerprint check not required. Biometric collected for potential dispute resolution purposes.	Fingerprint check not required.
Facial image collected at registration.	Facial image not collected if some other biometric is collected.	Facial image not collected.
The applicant must appear in person at Registrar at least once prior to issuance.	Remote registration allowable; applicant may avoid in-person encounter prior to issuance.	Remote registration of applicant possible through existing subscriber with a valid certificate at the same level; applicant may avoid in-person encounter prior to issuance.

<u>FIPS 201-1</u>	<u>id-CommonAuth or id-CommonHW policies</u>	<u>FBCA Medium-HW policy</u>
Two forms of original identity source documents from list in Form I-9 presented in original form. At least one must be a government issued picture ID.	One government issued identification document which includes or can be linked with biometric data of applicant.	One Federal government issued picture ID or two non-Federal IDs one of which is a picture ID.
Only designated sponsors can submit request for PIV card for an applicant.	Anyone with a valid credential issued under id-CommonAuth policy can act as a sponsor.	No requirement for a sponsor for an applicant.
Role separation implies that at least two authorized individuals need to be involved prior to issuance of card to applicant.	Only one authorized individual involved prior to issuance of credential to applicant.	Only one authorized individual involved prior to issuance of credential to applicant.
Identity proofing and registration process self-accredited by head of agency.	Third party audit required for authorization to operate CA.	Third party audit required for authorization to operate CA.
Card activated via PIN.	Card activated by passphrase, PIN or biometric.	Card activated by passphrase, PIN or biometric.

3. PIV AUTHENTICATION MECHANISMS

Chapter 6 of FIPS 201-1 provides a series of authentication use cases that can be supported using the electronic credentials resident on a PIV card. They are presented here to facilitate the reader's understanding of subsequent sections of this paper.

- CHUID – The cardholder is authenticated using the signed CHUID data element on the card. The PIN is not required. This mechanism is useful in environments where a low level of assurance is acceptable and rapid contactless authentication is necessary.
- CAK – The PIV card is authenticated using the Card Authentication Key in a challenge response protocol. The PIN is not required. This mechanism allows contact or contactless authentication of the PIV card without the holder's active participation, and provides a low level of assurance.
- BIO – The cardholder is authenticated by matching his or her fingerprint sample(s) to the signed biometric data element in an environment without a human attendant in view. The PIN is required to activate the card. This mechanism achieves a high level of assurance and

requires the cardholder's active participation is submitting the PIN as well as the biometric sample.

- BIO-A – The cardholder is authenticated by matching his or her fingerprint sample(s) to the signed biometric data element in an environment with a human attendant in view. The PIN is required to activate the card. This mechanism achieves a very high level of assurance when coupled with full trust validation of the biometric template retrieved from the card, and requires the cardholder's active participation is submitting the PIN as well as the biometric sample.
- PKI – The cardholder is authenticated by demonstrating control of the PIV authentication private key in a challenge response protocol that can be validated using the PIV authentication certificate. The PIN is required to activate the card. This mechanism achieves a very high level of identity assurance and requires the cardholder's knowledge of the PIN.

In each of the above use cases, except the symmetric CAK use case, the source and the integrity of the corresponding PIV credential is validated by verifying the digital signature on the credential. The entity signing the credential objects resident on a PIV card is called a PIV Signer. The PIV Signer has a special certificate under the Common Policy Framework; however, in legacy and cross-certified PKIs under the Federal Bridge environment, the PIV Signer can use a digital signature certificate issued under policies equivalent to the Federal Bridge CA (FBCA) Medium-HW and High policies.

3.1 Decomposition of PIV Authentication and Authorization

Identity credentials issued to conform to the PIV standard and related specifications can support a number of mechanisms for authentication of the user as described above. Assuming that technical interoperability have been achieved, the authentication of the holder of a PIV card can be decomposed into a series of activities as described below:

- Credential Integrity Validation – the relying party (RP) needs assurance that the identity credential is not tampered
- Credential Source Authentication – the RP needs to determine the identity and trustworthiness of the issuer of the credential
- Issuer Authority Verification – the RP needs to verify that the issuer of the credential has the authority to issue PIV credentials
- Credential Status Check – the RP may need to check that the identity credential is currently valid and not revoked
- Proof-of-Possession Check – the RP may require the user presenting the PIV card to prove that he or she is the rightful owner of the PIV card

The table below illustrates how each of the credentials present on a PIV card support the above decomposition steps.

Table 2 - CHUID Authentication

<u>Activity</u>	<u>Details of execution</u>
Integrity Validation	CHUID signature validated
Source Authentication	CHUID Signer certificate trust path validated to trust anchor
Issuer Authority Check	<i>id-PIV-content-signing</i> asserted within <i>extendedKeyUsage</i> extension of Signer certificate, or, explicit trust of CHUID Signer certificate/key
Status Check	Revocation check of PIV Authentication certificate (if practical)
Proof-of-Possession	-

Table 3 - CAK Authentication

<u>Activity</u>	<u>Details of execution</u>
Integrity Validation	CHUID contents used in CAK derivation (possibly ¹)
Source Authentication	Issuer key used in CAK derivation (possibly ¹)
Issuer Authority Check	Explicit trust of PIV card issuer as authoritative (possibly ¹)
Status Check	Backend channel status queries (if practical)
Proof-of-Possession	PIV card presented can perform challenge response to prove control of a CAK that matches derived/registered CAK

Table 4 - Biometric Authentication

<u>Activity</u>	<u>Details of execution</u>
Integrity Validation	Biometric object signature validated
Source Authentication	Biometric Signer certificate trust path validated to trust anchor
Issuer Authority Check	<i>id-PIV-content-signing</i> asserted within <i>extendedKeyUsage</i> extension of Signer certificate, or explicit trust of CHUID Signer certificate/key
Status Check	Revocation check of PIV Authentication certificate (if practical)
Proof-of-Possession	User provides PIN to activate PIV card; provides biometric sample which is matched to biometric object on card

¹ A possible symmetric CAK implementation could use the CHUID and Issuer key as inputs to derive a unique CAK for each PIV card.

Table 5 - PKI Authentication

<u>Activity</u>	<u>Details of execution</u>
Integrity Validation	PIV Authentication certificate signature validated
Source Authentication	PIV authentication certificate trust path validated to trust anchor
Issuer Authority Check	Certificate issuer asserts <i>id-Common-HW</i> policy, or, explicit trust of certificate issuer certificate/key
Status Check	Revocation check of PIV Authentication certificate
Proof-of-Possession	User provides PIN to activate PIV card ; uses private key on card in challenge response scheme to match PIV Authentication certificate

Following successful completion of some or all of the steps above, the RP knows the identity and a set of attributes of the PIV cardholder with varying degrees of certainty and assurance. The next step is to determine whether the cardholder can be granted access to the requested physical or logical resource. This access control decision is typically based on one of the following models:

- Identity-based access – the identity of the authenticated subscriber determines the authorization that may be granted. This model is appropriate when very fine-grained access provisioning and access revocation is required. For example, a specific Federal employee who is on detail to another agency for an extended period may be provisioned access based on their FASC-N.
- Role- or Group-based access – authorization is determined based on whether the identity is part of a broader group or set of individuals. This model is useful for rapid access provisioning and de-provisioning of groups of users. For example, all users from a particular agency may be provisioned access rapidly by allowing access to anyone whose PIV agency code matches the target agency.
- Attribute-based access - various other attributes (or combinations thereof) are evaluated to determine the authorization for the PIV cardholder. These attributes may be retrieved from the PIV card or from attribute authorities through backend channels. This model is useful to establish specific criteria for access without limiting access to specific individuals or groups. For example, users who are from a particular agency and whose NACI has been completed successfully may be granted access to a resource.

4. PIV COMPATIBLE AND PIV INTEROPERABLE CARDS

As the Federal government rolls out PIV cards for Federal employees and contractors, various other segments of government (e.g., state and local) and industry are also adopting the standards specified for PIV cards. These organizations desire to interoperate with Federal agencies. To this end, the Federal Identity Credentialing Committee (FICC) defined two new categories of identity credentials that are functionally and technically similar to

PIV cards, and may be accepted for access to Federal facilities and systems [4].

The primary challenges in making these non-Federally issued identity credentials interoperable are that non-Federal organizations cannot:

- 1) Satisfy the requirement to conduct a National Agency Check with Inquiries (NACI) on Subscribers
- 2) Issue digital certificates under the Common Policy Framework
- 3) Create Federal Agency Smart Credential Numbers (FASC-N) since these numbers include an Agency Code that is only capable of supporting Federal agencies.

PIV-Compatible cards conform to the technical specifications for PIV but do not support the trust and assurance of PIV cards.

PIV-Interoperable cards conform to the technical specifications for PIV and additionally have been issued in a manner that supports trust by Federal relying parties. Specifically, these cards must include an authentication certificate issued by a provider cross-certified with the Federal Bridge certification authority (FBCA) at *Medium-HW* policy and require subscriber registration through an identity proofing process that satisfies NIST SP 800-63 Level 4 requirements.

5. PIV CREDENTIALS AS FEDERATED IDENTITIES - CHALLENGES

A federated identity supports portability of identity information across disparate security domains. This allows users of one security domain to obtain services from a second security domain without the need for each domain to administer redundant identities for the same user. In promoting a “Government-wide standard for secure and reliable forms of identification”, HSPD-12 inherently envisions the use of the PIV card for access to various Federally controlled facilities and information systems. Thus, an implicit goal of HSPD-12 is to facilitate the use of the PIV card as a federated identity across the Federal government.

When an agency accepts credentials on PIV cards or PIV-like cards issued by organizations outside of their own agency, it constitutes a use case of “federated identity”. [Note that using local agency PIV cards for authentication and authorization is not considered federated use.] There are at least three scenarios of federated use of PIV or PIV-like cards as described below.

- Non-local Agency PIV cards – An agency allows the use of PIV cards issued by other Federal agencies as a means of authentication and subsequent authorization to agency controlled facilities and systems.
- PIV-Interoperable cards – An agency allows the use of PIV-Interoperable cards as defined by the FICC for authentication and authorization to agency controlled facilities and systems.
- PIV-Compatible cards - An agency allows the use of PIV-Compatible cards as defined by the FICC for authentication and authorization to agency controlled facilities and systems.

The challenges in accepting identity credentials as federated identities in each of the above scenarios are described the sections below.

5.1 Non-Local Agency PIV Cards

In accordance with HSPD-12 and FIPS 201, only Federal agencies can issue PIV cards to Federal employees and contractors. HSPD-12 requires that agencies “require the use of identification by Federal employees and contractors that meets the Standard in gaining physical access to Federally controlled facilities and logical access to Federally controlled information systems.” As agencies deploy PIV-enabled authentication mechanisms for physical and logical access, they need to evaluate the risks posed by acceptance of PIV cards issued by other agencies.

HSPD-12 requires that PIV cards be “issued only by providers whose reliability has been established by an official accreditation process.” NIST has published SP 800-79-1, “Guidelines for the Accreditation of Personal Identity Verification Card Issuers” to serve as a framework for accreditation [3]. However, accreditation is essentially an agency’s internal risk-based decision to authorize operation of a system. In the context of HSPD-12, accreditation is the subjective process of determining whether a PIV card issuer is compliant with FIPS 201-1 and related specifications. Each agency applies its own level of rigor to the compliance checking to determine whether their PIV card issuer can be accredited. FIPS 201-1 does not require an independent audit of the issuance and management processes for PIV cards. While the PKI credentials resident on the PIV card are issued through an infrastructure that mandates an independent annual audit, the additional requirements that pertain to a PIV card are never subjected to an independent audit.

The decision to accept PIV cards issued by other Federal agencies becomes even more complicated because HSPD-12 does not apply uniformly to all Federal agencies. HSPD-12 states that only “executive departments and agencies” need to implement a program in compliance with the directive. Effectively, this implies that Federal government organizations that are outside the executive branch are not mandated to implement HSPD-12 compliant programs. Although not required to do so, many of these non-executive branch agencies have decided to implement identity credentials technically equivalent to PIV cards (including PKI certificates issued through the Common Policy Framework for Subscribers as well as PIV Signers) – however, many of the process-oriented requirements of FIPS 201-1 are not being followed by these agencies since they are not required to comply with HSPD-12. Typically, these same agencies have decided not to accredit their issuance systems using the framework in NIST SP 800-79-1. As a result, while the PIV cards from these agencies are technically indistinguishable from PIV cards issued by executive branch agencies that have followed all required processes, they are, in essence, inferior in terms of the vetting and due diligence and hence do not have the same level of assurance.

Another concern in using the CHUID and biometric objects on the PIV card as a basis for authentication is that the integrity and source of these objects have to be verified through validation of the signature on the CHUID and biometric objects as described earlier. When the PIV card is issued by an agency that obtains PKI certificates through the Common Policy Framework, the PIV content signing certificate is clearly distinguishable through the

presence of the *id-CommonHW* policy identifier and an extended key usage of *id-PIV-content-signing*. However, when the PIV card issuer is using a legacy branch of the Federal PKI (e.g., one that is directly cross-certified with the FBCA) there is no obvious way to differentiate a PIV content signing certificate from a regular signature certificate issued under a policy equivalent to the FBCA *Medium-HW* policy.

In essence, it is entirely possible that a regular user who has a digital signature certificate asserting the equivalent of *Medium-HW* policy within the FBCA trust environment, can create PIV-like cards with digitally signed fictitious CHUIDs – a Federal relying party that verifies the signature on this type of CHUID would typically consider the CHUID to be trustworthy since the signer’s certificate can be validated through the FBCA; yet, this is clearly a scenario that needs to be detected by the relying party to prevent fraudulent CHUIDs being used to gain access. It may be noted that only trusted Certification Authorities (CAs) can issue the PIV authentication certificate, so this credential is not vulnerable to the same type of weakness as the signed CHUID and biometric objects.

The above concerns are summarized below.

Table 6 - Risks of Non-local Agency PIV Cards

Scenario	Risk
Independent audit of compliance not required by HSPD-12; only internal risk based accreditation using SP 800-79-1.	Agencies accepting non-local PIV cards don’t have assurance about the rigor of the SP 800-79-1 accreditation.
HSPD-12 mandate does not apply to non-Executive branch agencies.	Agencies accepting PIV cards from non-Executive branch agencies have little assurance of compliance with HSPD-12
Agencies with legacy PKI don’t have a mechanism to indicate authorized PIV object signers	Agencies accepting PIV cards from Issuers that use legacy PKI certificates have a low level of assurance in the integrity of the CHUID and BIO objects on the card.

The mitigation strategies to address the identified concerns are as follows:

- A relying party agency may analyze the issuing agency’s NIST SP 800-79-1 accreditation process and assessment results. The former may additionally require targeted assessments of the latter agency’s PIV issuance activities to more adequately identify the risks of accepting the issuing agency’s PIV cards.
- A relying party agency that wants to allow CHUID and BIO authentication for PIV cards issued by another Federal agency, can import the PIV Signer certificates from the second agency as trusted certificates (after careful vetting of the second agency’s processes related to issuance of the CHUID and biometric objects); this would ensure that only signed PIV objects from verified non-local PIV Signers are accepted for identity authentication purposes.

- A relying party agency may only accept PKI based authentication for holders of non-local PIV cards.

5.2 PIV-Interoperable Cards

As mentioned earlier, PIV-Interoperable cards are required to include an authentication private key and certificate that can be validated through the FBCA under *Medium-HW* policy. Additionally, NIST SP 800-63 Level 4 registration requirements need to be met by PIV-Interoperable cards.

Since the authentication certificate on the PIV-Interoperable card is issued under a policy equivalent to the *Medium-HW* policy of the FBCA, the assurance provided by this certificate (and corresponding private key) is very high. However, if the relying party desires to use the CHUID, biometric or CAK credentials loaded on the PIV-Interoperable card, the assurance level quickly drops off to nearly nothing. This is because the *Medium-HW* policy of the FBCA or requirements for Level 4 identity proofing under NIST SP 800-63 do not include the collection of biometrics during subscriber registration, nor do they include any form of background checking or role separation during registration and issuance.

Additionally, for the same reasons described in the previous section on PIV cards issued through legacy PKIs, there is no way to distinguish that the signer of the CHUID or biometric is an authoritative signer rather than just another user with a digital signature certificate within the FBCA environment. In summary, the CHUID and biometric credentials on a PIV-Interoperable card have little or no assured association to the identity asserted within the authentication certificate on the same card. Relying party agencies deciding to utilize PIV-Interoperable cards need to exercise the utmost discretion in choosing to use the CHUID, BIO and BIO-A authentication mechanisms with PIV-Interoperable cards.

The above concerns are summarized below.

Table 7 - Risks of PIV-Interoperable Cards

Scenario	Risk
No independent audit or SP 800-79-1 accreditation required for PIV-Interoperable cards	Agencies accepting PIV-Interoperable cards have little assurance of compliance with HSPD-12.
No mechanism to identify authorized signers of data objects on PIV-Interoperable cards.	Agencies accepting PIV-Interoperable cards have a low level of assurance in the integrity of the CHUID and BIO objects on the card.

The mitigation strategies to address the identified concerns are as follows:

- A relying party agency may require that the issuer of PIV-Interoperable cards demonstrates that it has performed a thorough assessment of their issuance facility and processes based on the NIST SP 800-79-1 guideline and are willing to make the results of the assessment available for review.
- A relying party may wish to include the certificate of the PIV Signer for each approved PIV-Interoperable

card issuer as an explicit trust anchor rather than accepting any Medium-HW signing certificate through the FBCA – this limits the acceptable signers of CHUID and biometric objects.

- A relying party agency may wish to perform background checking (such as NACI) on the subjects of PIV-Interoperable cards prior to allowing them access to federal facilities and systems.
- A relying party agency may only accept PKI based authentication for holders of PIV-Interoperable cards.

While these techniques definitely hinder interoperability, an agency with a low risk tolerance level may wish to employ one or more of these to allow the controlled acceptance of PIV-Interoperable cards as federated identities.

5.3 PIV-Compatible Cards

PIV-Compatible cards suffer from all of the assurance related drawbacks of PIV-Interoperable cards. In addition, there is no basis for trusting any of the digitally signed credentials on the card. Relying party agencies wishing to accept PIV-Compatible cards for access to facilities and systems should exercise the utmost caution and perform out of band due diligence of issuance processes and trustworthiness of the credentials on the PIV compatible card.

6. STRATEGIES TO IMPROVE ASSURANCE IN FEDERATED IDENTITY USING PIV AND PIV-LIKE CARDS

In Section 5, we discussed assurance related challenges in using PIV and PIV-like cards issued by external organizations and related mitigation options. This section offers some additional strategies to promote the use of PIV and PIV-Interoperable cards as federated identities.

In the near term, we recommend that the Office of Management and Budget (OMB) establish a clear policy that requires Executive branch agencies to conduct a thorough accreditation of their PIV card issuers prior to issuance of PIV cards; agencies should also be required to report their PCI accreditation activities to the OMB on a yearly basis. Likewise, we recommend that OMB establish policy that PIV and PIV-like cards that are accepted as a basis for allowing access to Federal facilities and resources, are issued by accredited issuers (in accordance with SP 800-79-1). This creates an environment where non-Executive branch agencies and commercial PIV-Interoperable card issuers would undergo SP 800-79-1 accreditation if they wish their cards to be accepted by other federal agencies.

In the long-term, it may be worth investigating whether the cost of implementing a third-party audit and compliance regime for issuers of PIV, PIV-Interoperable and PIV-Compatible cards can be balanced against the improved security and ease of federation between the digital identities of government and commercial organizations. This would be very similar to the work being done by the Liberty Alliance Identity Assurance Expert Group in the context of the assurance levels for electronic authentication.

7. STRATEGIES FOR RAPID ELECTRONIC AUTHENTICATION OF NON-LOCAL PIV AND PIV-LIKE CARDS

HSPD-12 establishes policy for secure and reliable forms of identification that can be “rapidly authenticated electronically.” When using non-local PIV or PIV-like cards, this becomes difficult since the types of authentication mechanisms that allow for rapid authentication – namely, CHUID, CAK, BIO, BIO-A – have little or no assurance. The PKI authentication mechanism is the only one that provides a reasonable level of assurance, however, this requires contact readers, PIN use, and possible fetching of online revocation lists. In this section, we describe a novel approach to rapid electronic authentication of non-local PIV and PIV-like cards.

Consider the scenario where an employee of Federal Agency A needs to work at the facility of Agency B for six or more months. This scenario occurs very often when agency employees are on detail to another agency. One very effective way to allow this non-local person rapid but secure authenticated access to Agency B’s physical facilities may be use a hybrid PKI-CAK scheme. In a “Visitor Enrollment” step at Agency B, the employee of Agency A can present their PIV card to the physical security group. The latter employs tools (like the PIV Trust Validation Tool being developed by NIST) to perform a thorough validation of all of the credentials on the non-locally issued PIV card, including the CHUID, biometric object and PKI credentials. The tool performs full path validation and revocation checking of all digital certificates needed to validate the credentials on this PIV card. The cardholder validates that they know the correct PIN to activate the PIV card, and his or her biometric samples match those stored on their PIV card. At the end of the Visitor Enrollment step, Agency B has a high degree of assurance that the cardholder is the genuine owner of the PIV card presented and that the credentials on the card are trustworthy and unmodified. As the last step of the Visitor Enrollment step, a series of random challenge strings (perhaps five to ten) are issued to the PIV card and the CAK is invoked to generate responses to each challenge string. The challenge-response pairs are stored along with the cardholder’s unique FASC-N as a part of the physical access control database (PACS-DB).

Following the Visitor Enrollment step, when this non-local individual needs to enter Agency A’s facilities, the contactless reader at the entry point will likely detect that the CHUID is not for a local subscriber. In this case, the PACS-DB record for that CHUID will be retrieved, and one of the stored challenges (selected randomly) will be issued to the visitor’s PIV card and the CAK invoked to respond. The received response will be compared to the stored response for that challenge string, and on a successful match, the visitor will be considered adequately authenticated. The FASC-N associated with that PACS-DB challenge-response pair will then be used for the authorization decision for the targeted facility. Since this CAK based challenge response scheme can be performed with a contactless reader without PIN submission, it allows for painless, rapid and secure authentication of the visitor. The assurance of this scheme can be further raised through additional mechanisms such as:

- Periodic revocation checking of all registered visitors to eliminate the need to do revocation checking in real-time

- Adding biometric authentication of the cardholder to match stored biometric objects (collected during the registration step)

The above scheme is most rapid when a symmetric CAK is present on the external PIV card, but works with a asymmetric CAK as well. Certificate path development and validation in real-time is eliminated in the scheme since it is done during the Visitor Enrollment step – occasional revocation checking is done in the background to validate the current status of the certificates within the PACS-DB. When the visitor presents their PIV card for authentication and access to a facility, the CAK is invoked with known challenge response pairs to establish the identity of the cardholder; additional assurance can be achieved by requiring cardholder biometric matching with the enrollment record.

Let's consider the use of PIV-Interoperable and PIV-Compatible cards by non-local individuals that need access to Agency A facilities for longer than six months. A similar Visitor Enrollment step can be followed which validates all of the credentials on the card and records the unique GUID of the card, biometric objects, and challenge-response pairs generated by invoking the CAK on the card. Additionally, a background check on the visitor may be performed if needed. Once the Visitor Enrollment record is completed, the visitor can use their PIV-like card for rapid but secure authentication for access to Agency A facilities.

8. CONCLUSION

In this paper, we discussed a number of trust and assurance issues related to the use of non-local PIV cards and PIV-like cards as federated identity credentials. We presented a number of

strategies to improve the assurance in the credentials carried in these non-local cards. We also presented a novel approach to higher assurance authentication of long-term visitors to a Federal facility through the use of a thorough Visitor Enrollment step that records challenge-response pairs for the CAK on the card.

9. ACKNOWLEDGMENTS

My thanks to Nabil Ghadiali and Dennis Bailey on supporting the refinement of the thoughts presented in this paper through lengthy discussions.

10. REFERENCES

- [1] The White House. August 2004. Policy for a Common Identification Standard for Federal Employees and Contractors. Homeland Security Presidential Directive/Hspd-12. DOI=<http://www.whitehouse.gov/news/releases/2004/08/20040827-8.html>
- [2] National Institute of Standards and Technology, March 2006. Federal Information Processing Standard (FIPS) Publication 201-1, Personal Identity Verification (PIV) of Federal Employees and Contractors.
- [3] National Institute of Standards and Technology, June 2008. NIST Special Publication 800-79-1, Guidelines for the Accreditation of Personal Identity Verification Card Issuers.
- [4] Spencer, J. 2008. Beyond HSPD-12: Interoperability with non-PIV Credentials. Office of Governmentwide Policy, Federal Identity Credentialing Committee. Presentation at Federal Information Assurance Conference 2008.



Personal Identity Verification (PIV) Cards as Federated Identities – Challenges and Opportunities

Dr. Sarbari Gupta

sarbari@electrosoft-inc.com

703-437-9451 ext 12

8th Symposium on Identity and Trust on the Internet (IDtrust 2009)

April 14-16, 2009

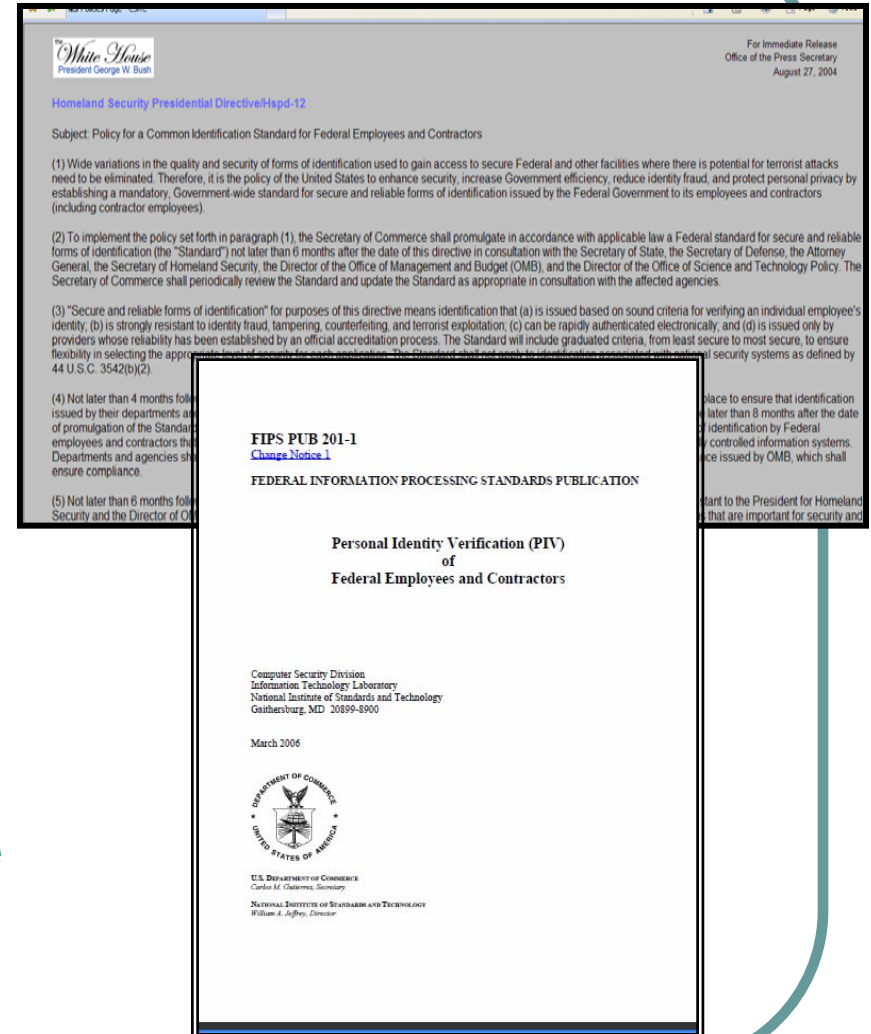


Overview

- **HSPD-12 requires Federal agencies to issue PIV Cards to all employees and contractors**
- **State/Local governments and commercial entities are issuing PIV-like Cards**
- **Immense opportunity to use PIV Cards (and PIV-like cards) as federated identities**
 - **Challenges**
 - **Strategies to promote federation**

HSPD-12 Background

- **Homeland Security Presidential Directive 12**
 - **Issued August 2004**
 - **Mandates Federal Agencies to issue common form of identification to Federal employees & contractors**
- **FIPS 201 - Personal Identity Verification (PIV) of Federal Employees and Contractors**
 - **PIV Card: Smart Card based digital identity container with a set of identity credentials**
- **PIV Card Issuers are required to be accredited by Agency Official**
 - **SP 800-79-1 – Accreditation Guide**



PIV Card Credentials

- **Mandatory Credentials:**
 - Cardholder Unique Identifier (CHUID)
 - PIV Authentication Private Key and X.509 Certificate (PKI)
 - Cardholder Fingerprints in Biometric Object (BIO)

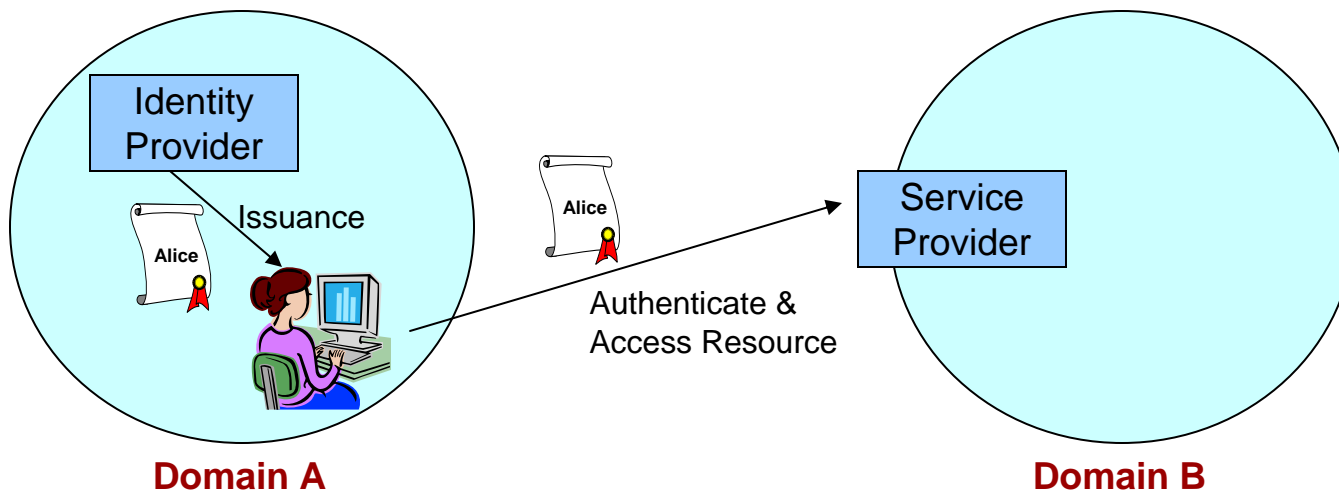
- **Optional Credentials:**
 - PIV Card Authentication Key (CAK)
 - PIV Digital Signature Private Key & X.509 Certificate
 - PIV Key Management Private Key & X.509 Certificate
 - Cardholder Facial Image



Pictures courtesy www.fedidcard.gov

Digital Identity Federation

- **Identity federation can be defined as ‘the agreements, standards and technologies that make identity and entitlements portable’ across otherwise autonomous security domains [Burton Group]**
- **Goal: Enable users of one domain to securely access data or services of another domain**



PIV-Interoperable & PIV-Compatible Cards

- Defined to promote *identity federation* between Federal and non-Federal Organizations
- Issued to personnel not eligible for PIV Cards
 - State and Local Government
 - Commercial Organizations
- **PIV Compatible:**
 - Meets technical specifications for PIV Card
 - Issuance process does not assure trust by federal relying parties
- **PIV Interoperable:**
 - Meets technical specifications for PIV Card
 - Issuance process assures trust in PKI Certificate
 - E-Authentication Level 4 Registration Requirements
 - PKI certificate issued under policy mapped to FBCA *Medium-HW* policy

FIPS 201 & FBCA Medium-HW Policy

FIPS 201-1

FBCA *Medium-HW* policy

NACI has to be completed for full scope PIV card.

NACI not required for regular applicants.

FBI fingerprint check required.

Fingerprint check not required.

Facial image collected at registration.

Facial image not collected.

The applicant must appear in person at Registrar at least once prior to issuance.

Remote registration of applicant possible; applicant may avoid in-person encounter prior to issuance.

Two forms of original identity source documents. At least one must be a government issued picture ID.

One Federal government issued picture ID or two non-Federal IDs one of which is a picture ID.

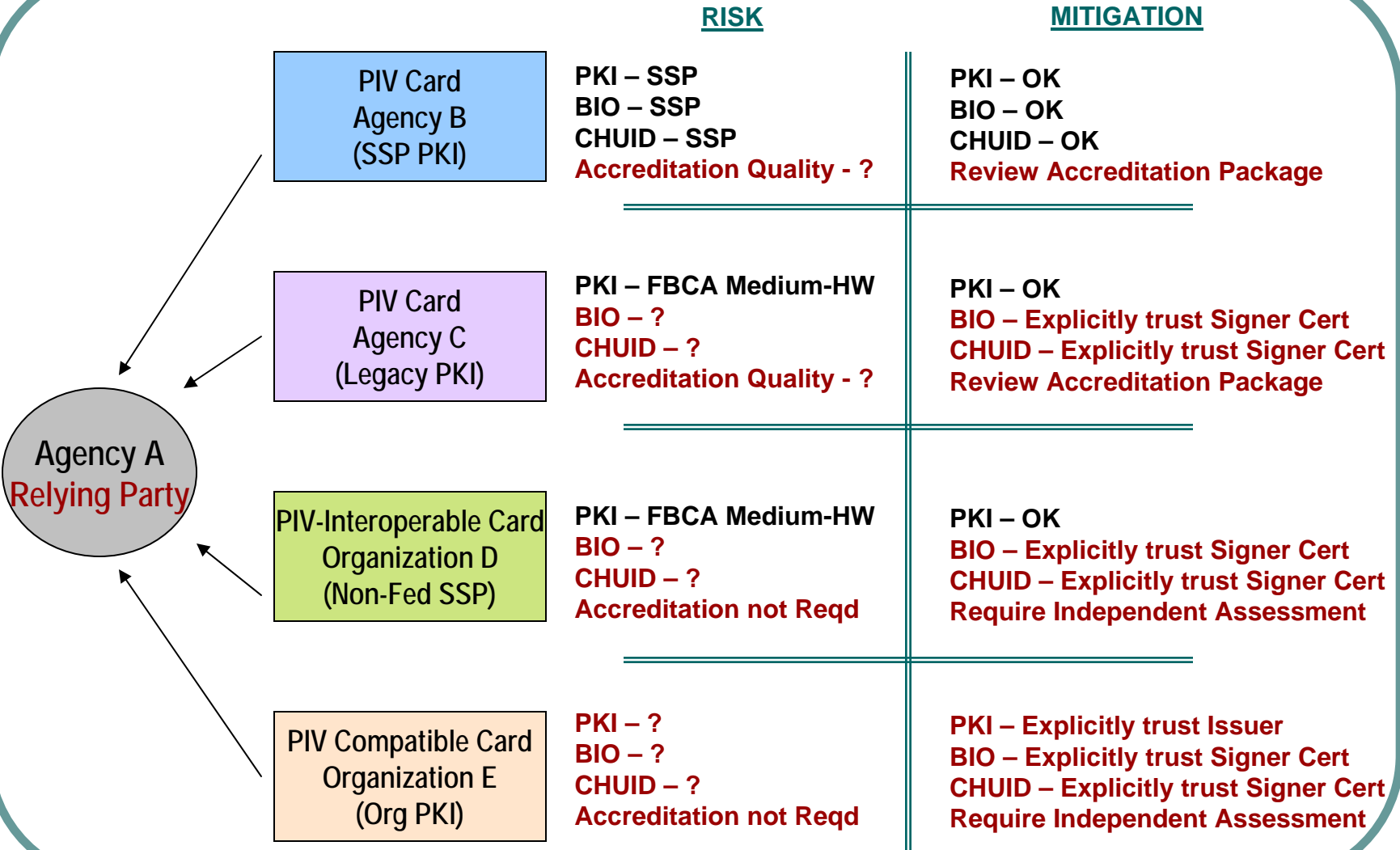
Only designated sponsors can submit request for PIV card for an applicant.

No requirement for a sponsor for an applicant.

Identity proofing and registration process approved by head of agency.

Third party audit required for authorization to operate CA.

PIV Federation - Risks and Mitigations



Fostering ID Federation with PIV-like Cards

■ **Suggestions:**

- **OMB Memo: Federal Relying Party can accept PIV-Interoperable Cards only from Issuers that are accredited/assessed using SP 800-79-1**
- **Update Certificate Profiles for FBCA Medium-HW policy to indicate authority of PIV Object Signers**
 - **E.g., Common Policy supports *id-PIV-content-signing* certificate extension**
- **Align the requirements of FIPS 201, Common Policy and FBCA Medium-HW Policy**
- **Establish 3rd party audit regime for compliance with FIPS 201 requirements**



Summary

- **Immense opportunity to use PIV Cards (and PIV-like cards) as federated identities**
- **QUESTIONS??**

A Calculus of Trust and Its Application to PKI and Identity Management

Jingwei Huang
Information Trust Institute
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801, USA
jingwei@iti.illinois.edu

David Nicol
Information Trust Institute
Dept. of Electrical & Computer Engineering
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801, USA
nicol@iti.illinois.edu

ABSTRACT

We introduce a formal semantics based calculus of trust that explicitly represents trust and quantifies the risk associated with trust in public key infrastructure (PKI) and identity management (IdM). We then show by example how to formally represent trust relationships and quantitatively evaluate the risk associated with trust in public key certificate chains. In the context of choosing a certificate chain, our research shows that the shortest chain need not be the most trustworthy, and that it may make sense to compare the trustworthiness of a potential chain against a threshold to govern acceptance, changing the problem to finding a chain with sufficiently high trustworthiness. Our calculus also shows how quantified trust relationships among CAs can be combined to achieve an overall trust assessment of an offered certificate.

Categories and Subject Descriptors

K.6.5[Management of Computing and Information Systems] [Security and Protection]; I.2.11 [Distributed Artificial Intelligence]

General Terms

Theory, Measurement, Security

Keywords

Trust modeling, PKI, Identity management, Risk assessment, Uncertainty, Semantics of trust, Social networks

1. INTRODUCTION

Trust plays a crucial role and is recognized as a major risk factor in public key infrastructure (PKI) and identity management (IdM). This paper explicitly represents trust with well defined semantics, and quantifies the risk associated with trust in PKI and IdM.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust '09, April 14-16, 2009, Gaithersburg, MD, USA
Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

Trust issues exist throughout identity management mechanisms. As identified in [2], organizations have concern about the business rules and mechanisms of their IdM partners with respect to the use of shared identity and credential information, how their partners protect the information, and the quality of identity information they provide; individuals (identity owners) are concerned whether their identity information in an organization is secure (not being stolen or revealed), how the information is used, and with whom it is shared. These trust concerns are beyond technologies, but are tightly associated with organizational and human behaviors. They are most difficult factors in identity management.

Digital signature and certification, facilitated by PKI, is considered to provide secure communication in the Internet, and as a fundamental tool to support IdM. However, many risks exist in PKI. The number one risk identified by Ellison and Schneier [8], "Who do we trust, and for what", reveals the risk of "imprecise use of the word 'trust' "; to avoid such risk, the use of trust relationships in digital certification and validation need to be precise and to be specific. An incident [10] in which VeriSign issued an impostor two digital certificates associated with Microsoft still reminds people to think whether a certification authority (CA) can be safely trusted regarding the validity of the issued certificates.

A number of PKI trust models have been proposed[33] [23] [30] [37]. However, these studies focus on the relationships among certification authorities (or the structure of PKI), the certification path construction methods, and the performance of different PKI structures and path build methods. Left missing is the explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths.

In PKI, a certification path corresponds to a chain of trust relationships. In distributed IdM such as federated IDM systems, a credential chain also corresponds to a chain of trust relationships. What do these trust relationships mean, exactly? On what precisely does one entity trust another in a credential chain? What is the specific context of each trust? How do we quantify the risk associated with trust in a credential chain?

In a typical public key validation model using PKI, at least one certification path with shortest length needs to be found and validated. When multiple paths exist, typically a short path is chosen in order to minimize the work involved. From the risk point of view this implicitly assumes that each certificate has the same level of risk, so the longer a certifica-

tion path is, the higher the risk is. However, when quantified evaluation of risk is introduced to PKI and different risks become associated with different certificates, some very interesting issues emerge. There are multiple certification paths, but which certification path should be chosen? Should more than one certification path be considered? Should all certificate paths be considered?

This paper explores some answers to these questions by introducing and applying a formal semantics based calculus of trust [17] to explicitly represent trust, and to quantify risk associated with trust in PKI and IdM.

This paper is organized as follows. Section 2 introduces related research; section 3 introduces a motivating scenario; section 4 discusses the semantics of trust; from a probabilistic perspective, section 5 discusses the measurement of uncertain trust; section 6 discusses sequence trust aggregation and parallel trust aggregation models; section 7 applies the trust calculus to formally represent trust relationships and quantitatively evaluate the risk associated with trust in public key certificate chains. Section 8 summarizes the research and discusses future directions.

2. RELATED RESEARCH

In this section, we review the research related to trust models for PKI, trust formalization and quantification.

Trust in PKI.

A number of PKI trust models have been studied [33] [23] [30] [37]. These studies focus on the relation among certification authorities (the structure of PKI), the certification path construction methods, and the performance analysis of different structures and path build methods. However, explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths is not a consideration of that research.

Most of public key certification path construction methods [9] [40] [29] implicitly assume that a certificate has the same level of risk, so a certification path with the shortest length has the least risk of being invalid. However, different certification authorities have different levels of rigor in the identity verification they apply before issuing a certificate, so that different certificates have different levels of risk.

Reiter and Stubblebine [36] work towards a quantitative evaluation of risk by studying the metrics for authentication in PKI. Based on their model of “resilient authentication”, the authors suggested two metrics to measure the risk in public key certification: the maximum number of independent paths with bounded length, and the maximum number of nodes which have to be removed (compromised) to break all certification paths. The proposed metrics are distinguished by their resiliency to malicious behaviors, but still follow the implicit assumption that each certificate has the same level of risk.

Maurer [27] explicitly represent certificate, trust, and recommendation in a certificate chain in PKI, and quantified uncertainty of the validity of a statement as “confidence level” in $[0,1]$. The limitation of his representation is that a single number between 0 and 1 can represent the uncertainty regarding the validity of a statement, but cannot represent the uncertainty due to incomplete knowledge, which is necessary in modeling trust in an open environment.

Trust Formalization and Quantification.

There are two major streams of research on trust formalization: logical approach and quantitative approach. The logical stream mainly focuses on the semantic structure of trust, the logical conditions and effects of trust. Examples include [4] [6] [16].

The quantitative stream mainly focuses on the uncertainty of trust, trust quantification, and the models & algorithms of trust computing. Trust has been quantified in several ways, at least including: linguistic values, graded levels, subjective probability, and probability distribution. In Marsh’s formalism [25], trust is quantified as a number in interval $[-1, +1]$; $+1$ represents complete trust, -1 represents completely distrust, and 0 represents “no trust” (untrust). In this way, Marsh clearly discerns untrust and distrust; but the relation between trust, untrust and distrust is somewhat oversimplified. A trust value is either between trust and untrust or between untrust and distrust. Based on a thorough examination of the concepts of trust developed in social sciences, Marsh designed a set of formulas to express the relations in trust; while Marsh’s model basically is a heuristic formalism. Most quantitative trust models define trust degree as a real value in interval $[0,1]$, e.g. [27] [31] [20]. As addressed in [12], it is a problem regarding how to interpret the meaning of 0 , which could be untrust or distrust. Ding et al. [7] defined 1 as fully trust, 0 as fully distrust, and 0.5 as fully ignorant (untrust). This is a simple approach to discern untrust and distrust. Most models in this school of research are heuristic, the measurement of trust is subjective, and the semantics of trust is not formally defined.

Josang [18] addressed uncertain belief representation by using subjective logic, in which an opinion regarding belief is represented as a triple (b, d, u) , where b , d , and u denote the degrees of belief, disbelief, and uncertainty, respectively, and $b+d+u = 1$. Later, Josang et al. [19] applied the subjective logic to represent uncertain trust. Explicit inclusion of uncertainty u enables to express and explain degrees of trust, distrust, and untrust, which takes into account incomplete knowledge about a trustee. We adopt this trust measure triple, but our model is different from Josang’s model [19]. In their model, although belief opinion is formally modeled, but the semantics of trust is not formally defined, and the relation between belief and trust is not modeled; hence, their trust model was subjectively defined as a heuristic formulation. Although they introduced degree of distrust, their trust derivation failed to clearly discern the semantics of untrust and distrust. We formally define each degree (of trust, distrust, and undecidable) in terms of formal semantics of trust, and based on the formal semantics, we derive trust calculation formulas rather than subjectively define the model.

Social Networks based Trust.

The Web has become an open dynamic and decentralized information/knowledge repository, global electronic markets, and a distributed computing platform. In such a cyberspace, people, organizations and software agents need to interact with others with unknown identity. To meet the needs, social networks based trust has attracted numerous researchers. We discuss this field in three categories: (1) **Reputation-based**, to infer the trustworthiness of an entity by considering its reputation in social networks, e.g. [32] [20] [38]; (2) **trust relationships based**, to infer indirect trust through

trusted friends, e.g. [11][7] [19]; (3) **vulnerability analysis based**, to evaluate trust indirectly by evaluating how vulnerable is the trust relation network which the trust relies on, e.g. [36] and Advogato [22].

Even though tightly related, trust and reputation are essentially different concepts. The reputation of an entity is the aggregated opinion that a community of people have about how good this entity is. Those people may give their opinions about that entity just based on a single encounter, and may not trust that entity at all; whereas trust is between two entities. Trust means that trustor believes his expectation on trustee to be fulfilled and he is willing to take the risk for that belief.

The rationale for reputation based trust computing is that an agent having high reputation in a domain usually is trustworthy in that domain. So, a reputation metric is frequently used as a substitute of a trust metric. Classical reputation systems (e.g. those used in eBay and Amazon) have been developed in e-commerce[5], which have a central trusted authority to aggregate opinions of users/partners. Some major limitations exist. A rating is usually given by a “stranger” who knows little about the evaluated subject and is limited to just a single interaction; users of reputation metrics don’t know the raters; unfair ratings exist [39]; a very large number of transactions are needed for statistical significance.

From network perspective, Kleinberg [21] proposed a profound model using eigenvector to discover “authorities” and “hubs” in a network. This work has wide influence in succeeded research; Page et al [32] adopted this thought in well known PageRank algorithm (used by Google) to calculate the reputation of a webpage; in the same vein, Kamvar et al [20] developed EigenTrust algorithm using eigenvector to calculate the global trust (actually reputation) from local trust in a P2P network.

Trust relationships based trust computing models infer or calculate indirect trust by using direct trust relationships in social networks. Most of them have two basic trust aggregation operators (or functions): sequence and parallel aggregation. The logic basis of sequence aggregation is transitivity and/or transferability of trust. Generally speaking, if agents share trust data with their trusted friends, within a specific context, *trust in belief* (trust in what trustee believes) is transitive, *trust in performance* (trust in what trustee performs) is not, but can propagates through *trust in belief*[15]. What is the basis for parallel aggregation still remains unclear. Parallel aggregation is an opinion aggregation problem. Ding et al [7] used entropy based aggregation. Many quantitative trust models use various weighted average, appearing as heuristics designed from intuition, for example, the more a trusted friend is, the more the weight of this friend’s opinion is in the aggregation.

Our model is a trust relationships based trust model. We believe, formal semantics is important, because without strictly defined semantics, the meanings of trust and trust degree are vague, the conditions and contexts to apply trust are not well defined, and the implication of trust are unclear, so that trust may be misused; for probability interpretation of trust degrees, it is critical to explicitly define the sample space of that probability. Different from other models in this category, our model has explicitly and formally defined semantics of trust; based on the formal semantics and probability theory, we derive sequence and parallel aggregation operators, rather than define them as heuristics. In this way,

our model is based on a solid formalism foundation.

3. MOTIVATING SCENARIOS

The technology and standards of digital signature and certification with PKI provide a fundamental and secure approach to authentication. An authenticated identity plus a set of authenticated credentials (assertions that the entity has a set of attributes) may lead to authorizing the authenticated entity access to controlled resources such as information, services, or goods, subject to predefined authorization policies. An authorization policy usually is based on either the rights of the authenticated entity or a trust relationship from the resource controller to that entity.

We illustrate the underlying concepts of our research in the following scenarios.

In Chicago, Alice, a physician, needs helps for treating the disease of a patient; in the electronic medical messaging system connecting to her clinic, she creates a message to ask for help, which is sent to a set of doctors with an attribute-based messaging system; soon, Alice receives a message with digital signature from Dan, a specialist in epidemiology in Philadelphia; the message says that Dan could help and he needs further information about the patient; but Alice does not know Dan; can Alice trust Dan regarding Dan’s professional performance and regarding whether Dan follows the terms of use of the privacy data she provides?

First of all, Alice needs to ensure that the message is sent by the person as claimed. To authenticate Dan’s identity, Alice needs to validate Dan’s public key. This is accomplished by discovering a certificate chain from an Alice’s trusted certificate authority (CA) (called trust anchor in PKI literatures) to the CA who issues Dan’s public key, and then by validating each public key in the certificate chain from the public key of Alice’s trust anchor to Dan’s public key. The certification path construction and validation is a standard PKI function. However, any CA in the chain may make mistakes in its certificate such as issuing the certificate to a wrong entity as [10], key compromised for its limited “cryptographic lifetime” or “theft lifetime” [8] before the expired date stated in the certificate, failure in maintaining CRL (Certification Revocation Lists), and so forth. With the growth of the length of a certificate chain, the risk gets higher and higher. How large is this the risk and can it be quantified? If the risk in each certificate chain is quantified, how can this quantified risk be used for certification path construction?

However, the authentication of Dan’s identity is not sufficient for Alice to trust Dan as an expert in epidemiology. This chain of public key certificates is easily misunderstood as a chain of references to Dan’s knowledge of epidemiology, especially when PGP is used for public key certification. Actually, Dan’s public key certificate consists only of (1) Dan’s public key; (2) Dan’s ID information; (3) certification information such as expiration date; (4) a digital signatures signed by a certification authority who verified that the signed public key belongs to Dan, and maintains the validity of this public key. The relationship in a public key certificate is only limited to trust in public key validation.

These scenarios reveal and motivate us to consider a number of relevant issues:

- There are many risks associated with trust in pub-

lic key certificate chains and more general credential chains. These trust relationships are largely dependent on the organizational or human behaviors, which is beyond PKI technology;

- Trust is inherently uncertain, being dependent on behavior of organizations or humans. To support better authentication authorization decisions, there is a need to quantify the risks associated with trust in credential chains;
- The quantified risk associated with trust may be useful in certification path construction and selection;
- Trust is subject to a specific context. Alice may trust a CA regarding issuing and maintaining Dan’s public key, but not regarding whether Dan is a specialist in epidemiology;
- Trust is uncertain. An interesting question here is how to measure the uncertainty in trust in an open environment in which each entity may subjectively give his own trust?
- There may be multiple trust paths between Alice and the CA issuing Dan’s public key. The connection between them may be even more complex as a network. How should the risk associated with a trust network be evaluated?

4. SEMANTICS OF TRUST

4.1 Concept of Trust

Trust is a complex social phenomenon. The concepts developed in social sciences provide an important foundation for trust formalization. A large body of research has contributed to the conceptualization of trust [24] [28][1].

In this paper, we use the following definition of trust [16][15]: *Trust is the psychological state comprising (1) **expectancy**: the trustor expects a specific behavior of the trustee such as providing valid information or effectively performing cooperative actions; (2) **belief**: the trustor believes that the expected behavior occurs, based on the evidence of the trustee’s competence and goodwill; (3) **willingness to be vulnerable**: the trustor is willing to be vulnerable to that belief in a specific context, where the specific behavior of the trustee is expected.*

According to the types of the expectancy in trust, there are two types of trust: *trust in performance* and *trust in belief*. The former is the trust in what trustee performs; the later is the trust in what trustee believes. These two types of trust play important roles in our trust modeling.

4.2 Semantics of Trust and Distrust

Based on the formal semantics of trust defined in the [15], we give a simpler version of the semantics of trust in First Order Logic as follows.

Definition (trust in performance): That trustor d trusts trustee e regarding e ’s performance (represented by x) in context k means that if information x is made by entity e , then entity d believes x in context k .

$$trust_p(d, e, x, k) \equiv madeBy(x, e, k) \supset believe(d, k \dot{\supset} x) \quad (1)$$

In the above definition, information x is a reified proposition¹ representing either an assertion made by e or a commitment made by e to perform (or not to perform) an action; context k is a reified proposition representing the conjunction of a set of “propositions” to characterize a context. A dot over a logical operator is a function to mimic that logical operator, e.g. $\dot{\supset}$ is a function to mimic logical implication.

Definition (trust in belief): That trustor d trusts trustee e regarding e ’s belief (represented by x) in context k means that if information x is believed by entity e , then entity d believes x in context k .

$$trust_b(d, e, x, k) \equiv believe(e, k \dot{\supset} x) \supset believe(d, k \dot{\supset} x) \quad (2)$$

In order to represent uncertainty in trust, the concept of distrust needs to be introduced.

In Marsh’s review on the concepts of trust, untrust, distrust, and mistrust [26], distrust is regarded as the negative form of trust; untrust is a status where the degree of confidence is not enough to trust; mistrust is misplaced trust.

More specifically, in our discussion, *distrust* means trustor d believes the expectancy not to be true, in other words, for *distrust in performance*, the trustor believes that the expected information created by trustee e is false, the expected performance of e does not come true, or unexpected behavior comes true; for *distrust in belief*, the trustor believes what trustee believes is false. Formally,

$$distrust_p(d, e, x, k) \equiv madeBy(x, e, k) \supset believe(d, k \dot{\supset} \neg x) \quad (3)$$

$$distrust_b(d, e, x, k) \equiv believe(e, k \dot{\supset} x) \supset believe(d, k \dot{\supset} \neg x) \quad (4)$$

Here, corresponding to the formal notation of that entity a believes a proposition x , $believe(a, x)$, the formal notation of disbelief is represented by $believe(a, \neg x)$.

This view of distrust is grounded by the theory of Luhmann [24](chapter 10). The literature addresses that trust and distrust are qualitatively different but functionally equivalent; both of them are based on familiarity; both of them reduce social complexity. In other words, both trust and distrust correspond to certainty, but in different directions.

Finally, we define the semantics of a more general form of trust relationships with a specific context but without a specific expectancy.

$$\begin{aligned} trust_b(d, e, k) &\equiv (\forall x, trust_b(d, e, x, k)); \\ trust_p(d, e, k) &\equiv (\forall x, trust_p(d, e, x, k)). \end{aligned} \quad (5)$$

The straightforward meaning of them is that d trust e regarding e ’s every performance or belief in context k . This more general form of trust relationships is more often used in the real world.

Similarly, we define

$$\begin{aligned} distrust_b(d, e, k) &\equiv (\forall x, distrust_b(d, e, x, k)); \\ distrust_p(d, e, k) &\equiv (\forall x, distrust_p(d, e, x, k)). \end{aligned} \quad (6)$$

The general form of distrust is not often in the real world, but logically it is the extreme end of distrust, which will be

¹A reified proposition is a relation representing a “proposition” but it is data rather than a proposition in the representation language.

used in uncertain trust model in the next section.

4.3 Trust Reasoning

The above semantics of trust can be used for trust reasoning. Generally, trust is placed on an entity (or agent), which behaves autonomously. In other words, the behaviors of the trusted entity is out of control of the trustor. On the other hand, belief is placed on information (or more exactly, a proposition). The purposes of formalizing trust are to accurately define what trust means when we use trust, and to use the semantics of trust to infer whether the information created by the trusted entity or the information representing an expected behavior from the trusted entity to be believed to be true.

In the following, we briefly introduce the logical rules for trust reasoning based on the formal semantics of trust.

By applying the formal semantics of trust in performance as well as modus ponens, we have

Rule 1:

$$madeBy(x, e, k) \wedge trust_p(d, e, x, k) \supset believe(d, k \dot{\supset} x) \quad (7)$$

Similarly, for trust in belief, we have:

Rule 2:

$$believe(e, k \dot{\supset} x) \wedge trust_b(d, e, x, k) \supset believe(d, k \dot{\supset} x) \quad (8)$$

By *trust in belief*, *trust in performance* can propagate in a network; in other words, for a given context k , if entity a trusts b on b 's belief in other entities' performance, b trusts c in c 's performance, then a indirectly trusts c regarding c 's performance.

However, when entity a trusts b on belief in a context, and b trusts c on performance in a *different* context, from these two trust relationships, we cannot derive that a trusts c .

The rules for trust propagation are given as follows.

Rule 3:

$$trust_b(a, b, x, k) \wedge trust_p(b, c, x, k) \supset trust_p(a, c, x, k) \quad (9)$$

$$trust_b(a, b, x, k) \wedge trust_b(b, c, x, k) \supset trust_b(a, c, x, k) \quad (10)$$

This rule requires the expectancy (x) of trust to be the same in the trust from a to b and the trust from b to c . Actually, this rule can be extended to a more general form without a specific expectancy, as follows.

Rule 4:

$$trust_b(a, b, k) \wedge trust_p(b, c, k) \supset trust_p(a, c, k) \quad (11)$$

$$trust_b(a, b, k) \wedge trust_b(b, c, k) \supset trust_b(a, c, k) \quad (12)$$

The proof of these trust propagation rules can be found in [15] and is omitted here.

5. MEASUREMENT OF UNCERTAIN TRUST

Because trust is placed on another organization, another person, or a group of persons, the features of human behaviors make trust inherently uncertain. In this section, we discuss the measurement and representation of uncertain trust. We use probability to measure uncertainty in trust, so that each entity in a distributed computing environment could give his degrees representing uncertainty in trust, based on a common understanding regarding what the numbers mean.

5.1 Formal Definition of Trust Degree

Based on the formal semantics of trust presented in the previous section, as well as the connections of probability and conditionals [13] studied in philosophical logic, the degree of trust is defined as follows.

$$td^p(d, e, x, k) = pr(believe(d, x) | madeBy(x, e, k) \wedge beTrue(k)) \quad (13)$$

for trust in performance;

$$td^b(d, e, x, k) = pr(believe(d, x) | believe(e, x) \wedge beTrue(k)) \quad (14)$$

for trust in belief.

When the type of trust is not concerned, we omit the superscript p/b .

Similar to trust degree, based on the formal semantics of distrust, the degree of distrust is defined as:

$$dtd^p(d, e, x, k) = pr(believe(d, \dot{\supset}x) | madeBy(x, e, k) \wedge beTrue(k)) \quad (15)$$

for distrust in performance;

$$dtd^b(d, e, x, k) = pr(believe(d, \dot{\supset}x) | believe(e, x) \wedge beTrue(k)) \quad (16)$$

for distrust in belief.

The sample space of the probability representing trust degree could be any event set that contains the events in which the conditions are true. The minimal sample space is exactly the set of events in which the conditions in the conditional probability are true.

5.2 Measurement of Trust Degree

The previous subsection provides formal definitions of trust /distrust degrees in probability theory. This subsection gives a frequency interpretation to the probabilities defining trust /distrust degrees. The formal definitions and the frequency interpretation provide a formal interpretation about the semantics of the numbers representing trust /distrust degrees, and puts the calculus of trust in a firm theoretic basis. The latter point is important—in practice, this frequency interpretation of trust /distrust degree can be used as a practical method to calculate trust /distrust degrees, by using the data accumulated in the interaction between a trustor and a trustee. However, when the data of interactions are not available or not used, a trust /distrust degree may be given as a subjective probability but with the assurance that the calculus itself is nevertheless correct.

In the following, we describe measurement of trust /distrust degrees and a frequency interpretation of probability.

Trust can be divided into two categories: (1) *direct trust*, the trust coming from direct interaction between two parties, and (2) *indirect trust*, the trust derived from a social network. The derivation of indirect trust will be discussed in the next two sections; we only need to discuss how to count direct trust.

The degree of trust is measured with the frequency rate of trustor's positive experience among all encounters with the trustee. That is,

$$td(d, e, x, k) = n/m, \quad (17)$$

where, m is the total number of encounters regarding an instanced expectancy x , and n is the number of trustor's

positive experience. For example, x is an assertion about the authentication of a specific customer, John, who signs on to request a service; d is the service provider; e is the identity provider who makes authentication assertion x ; k is a context such as “sign in for online shopping”; m is the total times of which e informed d about the authentication of John’s Id in d ’s historic data set; n is the number of correct authentication about John; this rate of n/m reflects the probability by which e makes correct authentication about John’s Id.

similarly, we have

$$dtd(d, e, x, k) = l/m, \quad (18)$$

where l is the number of trustor’s negative experience. $n + l$ is not necessarily equivalent to m , if some encounters are hard to say being positive or negative.

For the degree of general trust without a specific expectancy,

$$\begin{aligned} td(d, e, k) &= n'/m', \\ dtd(d, e, k) &= l'/m' \end{aligned} \quad (19)$$

where, m' is the number of all encounters between trustor and trustee regarding all instanced expectancy ($\forall x$). n' is the number of positive experience in those encounters. l' is the number of negative experience in those encounters. In the earlier example, m becomes the total times of which e informed d about the authentication of the identity of any signed custom in d ’s historic data set; n becomes the number of correct authentication.

In practice, people use specific information for a specific problem solving, but when the specific information is not available, the general inform may be applied. So, for the case of lack of information about a specific expectancy x , $td(d, e, k)$ may be used as an estimated value of $td(d, e, x, k)$.

Similar to uncertain belief and uncertain trust, a trustor may evaluate each encounter as positive (or satisfied, succeed) to a certain extent, as negative (or unsatisfied, failed) to a certain extent, and as undecidable (or hard to say positive or negative) in a certain degree. In such case, trust /distrust degrees can be refined as:

$$td(d, e, x, k) = \frac{\sum_{i=1}^m e_p(i)}{m}, \quad (20)$$

where m is the same as defined earlier;

$$e_p(i) \in [0, 1]$$

represents the degree of encounter i being positive, $e_p(i) = 1$ represents completely positive, and $e_p(i) = 0$ represents completely not positive.

Similarly, for distrust degree,

$$dtd(d, e, x, k) = \frac{\sum_{i=1}^m e_n(i)}{m}, \quad (21)$$

where,

$$e_n(i) \in [0, 1],$$

which is the degree of encounter i being negative, and

$$e_p(i) + e_n(i) \leq 1. \quad (22)$$

The difference,

$$1 - e_p(i) - e_n(i) \quad (23)$$

represents the degree of uncertainty in trustor’s evaluation of encounter i due to lack of sufficient information for judgment.

5.3 Notation of Uncertain Trust Relationships

A trust relationship can be represented by the degree of trust and the degree of distrust.

The sum of degrees of trust and distrust actually represents the degree of certainty, denoted as cd , e.g.

$$cd(d, e, x, k) = td(d, e, x, k) + dtd(d, e, x, k). \quad (24)$$

The degree of uncertainty, denoted as ud , is defined as

$$\begin{aligned} ud(d, e, x, k) &= 1 - cd(d, e, x, k) \\ &= 1 - td(d, e, x, k) - dtd(d, e, x, k). \end{aligned} \quad (25)$$

This uncertainty comes from the unfamiliarity of trustor to trustee, or from a trustor’s lack of sufficient information to evaluate trust of the trustee.

We have $td + dtd + ud = 1$.

When $ud = 0$, $td + dtd = 1$, which corresponds to the most certain situation in which the trustor is sufficiently familiar with the trustee, so the trustor can surely make decision to either trust or distrust the trustee.

When $0 < ud < 1$, $td + dtd < 1$, which corresponds to a typical uncertain situation in which the trustor is not sufficiently familiar with the trustee, so there is uncertainty regarding trust decision.

When $ud = 1$, $td + dtd = 0$, which corresponds to the most uncertain situation in which the trustor is completely not know about the trustee, so the trustor cannot give the degrees of trust or distrust at all. This is the typical case of “untrust”

Another interesting case is $td = dtd = 0.5$, which is the most uncertain situation when $ud = 0$.

A situation easy to be confused is when $td = 0$. Some people may think it means untrust; some may think it means distrust; some others may think it’s not sure which case is. If a trust relationship is defined sole by td , it is hard to distinguish between these two possibilities. In our notation, when $td = 0$, $ud + dtd = 1$. So, depending on what ud and dtd are, there is a probability distribution over distrust and untrust.

Therefore, a trust relationship can be formally represented as

$$tr(d, e, x, k) = \langle td(d, e, x, k), dtd(d, e, x, k) \rangle, \quad (26)$$

or

$$tr(d, e, x, k) = \langle td(d, e, x, k), dtd(d, e, x, k), ud(d, e, x, k) \rangle, \quad (27)$$

when the value of ud need to appear explicitly.

For a general trust relationship without a specific expectancy, we have

$$tr(d, e, k) = \langle td(d, e, x, k), dtd(d, e, k) \rangle. \quad (28)$$

This formal representation of trust relationships has strictly defined semantics, so that it can help to avoid mistakes as we develop a calculus for trust.

6. TRUST CALCULATION

In the following, we discuss how to calculate the degree of trust when trust propagates in trust networks. Basically, there are two types of trust aggregations: sequence aggregation and parallel aggregation. Sequence aggregation describes aggregation of trust degrees along a trust path; parallel aggregation is about how to aggregate trust degrees in several parallel trust paths.

6.1 Sequence Trust Aggregation

The sequence aggregation problem is this: given that a trusts b (on belief) with a probability distribution trust, distrust, and undecidable, b trusts c (either on belief or on performance) with another probability distribution, what is the probability distribution for a to trust c , that is, what are $td(a, c, x, k)$ and $td(a, c, x, k)$?

The following theorem answers this question.

Theorem UT-1: (1) assume that agent a has a trust in belief relationship with b ,

$$tr(a, b, x, k) = \langle td^b(a, b, x, k), dtd^b(a, b, x, k) \rangle, \quad (29)$$

b has trust in performance relationship with c ,

$$tr(b, c, x, k) = \langle td^p(b, c, x, k), dtd^p(b, c, x, k) \rangle, \quad (30)$$

and the belief of a in x is conditionally independent to the provenance of x (or the belief of c in x) given the belief of b in x , then the trust relationship from a to c can be derived as follows:

$$tr(a, c, x, k) = \langle td^p(a, c, x, k), dtd^p(a, c, x, k) \rangle, \quad (31)$$

and

$$td^p(a, c, x, k) = td^b(a, b, x, k) \cdot td^p(b, c, x, k) + dtd^b(a, b, x, k) \cdot dtd^p(b, c, x, k), \quad (32)$$

$$dtd^p(a, c, x, k) = dtd^b(a, b, x, k) \cdot td^p(b, c, x, k) + td^b(a, b, x, k) \cdot dtd^p(b, c, x, k); \quad (33)$$

The certainty degree in this derived trust relationship satisfies

$$cd(a, c, x, k) = cd(a, b, x, k) \cdot cd(b, c, x, k). \quad (34)$$

(2) if b has trust in belief relationship with c , and the conditional independent condition is – the belief of a in x is conditionally independent to the belief of c in x given the belief of b in x , then the trust relationship from a to c can also be derived, but the derived trust is trust in belief.

<end of theorem>

For reasons of space, the proof of this theorem is omitted here but can be found in [17].

The above assumed conditional independent condition is similar to the assumption in belief networks and Markov chains, which assumes an event is only directly dependent on its parents.

For general trust relationships without a specific expectancy, the above theorem also true by removing all expectancy x , and revising the conditional independent condition as follows: the belief of a in any x is conditionally independent to the provenance of any x (or the belief of c in any x) given the belief of b in any x .

This sequence aggregation operator has some interesting properties.

Property 1: with the growth of the length of a trust path, the certainty degree of the aggregated trust multiplicatively decreases.

For simplicity, we use subscript i, j represents that a trust relationship is from entity i to entity j .

Assume the length of a trust path is n . By theorem UT-1, we have

$$cd_{1,n} = cd_{1,2} \cdot cd_{2,3} \cdot \dots \cdot cd_{n-1,n}, \quad (35)$$

Therefore, we have property-1.

This property is coincident with people’s intuition regarding trust decreasing quickly in propagation along a trust path.

Property 2: sequence aggregation is associative.

By this property, the outcome of the sequence aggregation is independent to the order of sequence aggregation for each pair of trust relationships in a trust path. This property is important when applying sequence aggregation in the algorithm for trust aggregation in a network.

Property 3: A trust relationship with $ud = 1$ is a “zero” element in trust aggregation.

This property says that if the trust of a in b or the trust of b in c is untrust with $ud = 1$, the derived trust of a in c is also “untrust” with $ud = 1$. In other words, in a trust network, a trust relationship with $ud = 1$ is the same as there is no trust relation between the two entities.

The implication of this property is obvious. In a trust network, if there is only one trust path from a to c , then any “untrust” in the path will make the trust path “broken” which is equivalent to the case that there is no trust path from a to c . As a result, a will “untrust” c .

This property reveals that trust evaluation will not change by adding or cutting off a trust relationship with $td = dtd = 0$. Therefore, cutting off all trust relationships with td and dtd near 0 will effectively reduce the complexity of a trust network and the associated computation complexity.

Sequence aggregation is a basis for trust aggregation in a trust network. Sequence aggregation also can be applied independently. For example, in identity management, a credential chain is a trust path, and sequence aggregation can be used for analyzing trust related risk in a credential chain. In PGP, a sequence of public key introducers forms a trust path, and sequence aggregation can be used to calculate a numeric value of the overall trust along that trust path.

A key issue now is aggregation of trust relationships (opinions) in parallel trust paths, which is difficult due to the lack of the commonly recognized logic to synthesize different opinions. In the following, we discuss how to make parallel trust aggregation first, then we discuss how to make trust evaluation in a trust network by using sequence aggregation and parallel aggregation.

6.2 Parallel Trust Aggregation

In general, as shown in figure 1, parallel trust aggregation needs to answer the following question: given that entity a directly or indirectly trusts (in belief) $b_1, \dots, b_n, b_1, \dots, b_n$ trust c (either in belief or in performance), and a may also directly trusts c , what is the aggregated trust from a to c ?

We assume that in a trust network each direct trust relationship (described by trust degree and distrust degree) and the number of samples used to determine that trust relationship are given.

For simplicity, we use $td(e_i, e_j)$ ($dtd(e_i, e_j)$) to denote $td(e_i, e_j, x, k)$ ($dtd(e_i, e_j, x, k)$), by omitting x, k , because they are the same²; we use $s(e_i, e_j)$ to denote the number of samples used in assessing $td(e_i, e_j)$ and $dtd(e_i, e_j)$; furthermore, we use a superscript $*$ to denote an aggregated trust relationship, e.g. $td^*(e_i, e_j)$ ($dtd^*(e_i, e_j)$). In addition, we omit superscripts p and b when the type of trust is not

²Actually, for a specific trust evaluation regarding the information x in context k , a specific sub-network with the same x and k is selected from a real world trust network. See detail in subsection 6.3.

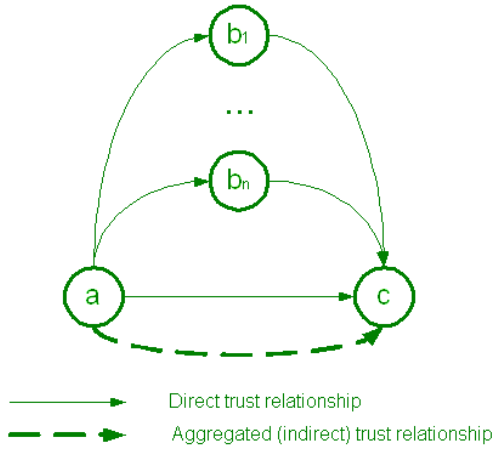


Figure 1: Parallel trust aggregation in multiple trust paths

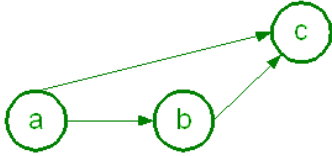


Figure 2: A simple example of parallel trust aggregation

concerned.

We start from the simplest case as shown in figure 2. Here entity a has two trust paths to entity c : the direct trust from a to c , and the indirect trust via b .

We define parallel aggregation based on the interpretation of a trust degree as a frequency rate of successful interaction. From the view of entity a , the total number of encounters with c regarding the information x in context k is the sum of the encounters of both entity a 's direct interaction and indirect interaction via b with c , that is,

$$s^*(a, c) = s(a, c) + s(a, b, c), \quad (36)$$

where,

$$s(a, b, c) = s(b, c); \quad (37)$$

among these encounters, the total number of successful encounters with c is the sum of (1) the successful encounters that a directly interacts with c , which is, by frequency rate definition of trust degree,

$$s(a, c) \cdot td(a, c); \quad (38)$$

and (2) the successful encounters that a indirectly interact via b with c , which is, by sequence trust aggregation,

$$s(a, b, c) \cdot (td^b(a, b) \cdot td(b, c) + dtd^b(a, b) \cdot dtd(b, c)). \quad (39)$$

A natural interpretation is that a reviews each direct interaction that b had with c , and evaluates each certain (i.e., positive or negative) interaction (from a 's point of view) as being the same as that from b 's point of view with probability $td^b(a, b)$, and being the opposite of b 's with probability $dtd^b(a, b)$. In a sense b 's direct interactions with c are being interpreted as a 's interactions with c , and are given the same

weight (after normalization by trust-in-belief measures) as direct interactions that a has with c .

So, the aggregated degree of trust from entity a to c , $td^*(a, c)$, is:

$$\begin{aligned} td^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot td(a, c) \\ &+ (s(a, b, c)/s^*(a, c)) \cdot \\ &(td^b(a, b) \cdot td(b, c) \\ &+ dtd^b(a, b) \cdot dtd(b, c)) \end{aligned} \quad (40)$$

The type of $td^*(a, c)$ depends on the type of $td(b, c)$. For $td^p(b, c)$, $td^*(a, c)$ will be trust in performance; for $td^b(b, c)$, $td^*(a, c)$ will be trust in belief.

Similarly, we have the aggregated degree of distrust as follows.

$$\begin{aligned} dtd^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot dtd(a, c) \\ &+ (s(a, b, c)/s^*(a, c)) \cdot \\ &(td^b(a, b) \cdot td(b, c) \\ &+ dtd^b(a, b) \cdot dtd(b, c)) \end{aligned} \quad (41)$$

For the general case shown in figure 3, the aggregated trust degree can be calculated as

$$\begin{aligned} td^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot td(a, c) \\ &+ \sum_{i=1, \dots, n} (s(a, b_i, c)/s^*(a, c)) \cdot \\ &(td^b(a, b_i) \cdot td(b_i, c) \\ &+ dtd^b(a, b_i) \cdot dtd(b_i, c)), \end{aligned} \quad (42)$$

where

$$s^*(a, c) = s(a, c) + \sum_{i=1, \dots, n} s(a, b_i, c), \quad (43)$$

and

$$s(a, b_i, c) = s(b_i, c); \quad (44)$$

Similarly, the aggregated distrust degree can be calculated as

$$\begin{aligned} dtd^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot dtd(a, c) \\ &+ \sum_{i=1, \dots, n} (s(a, b_i, c)/s^*(a, c)) \cdot \\ &(td^b(a, b_i) \cdot dtd(b_i, c) \\ &+ dtd^b(a, b_i) \cdot td(b_i, c)); \end{aligned} \quad (45)$$

From the frequency definition of trust degree, a parallel aggregation is derived, which appears in the form of weighted average of all parallel trust paths, and the weight of a path is the rate of the number of the samples in this path to the total number of samples.

Generally, the more samples are used in a trust path, the more accurate the trust degree is in that trust path, and the more weight that trust path will have in aggregation. This is also coincident with people's intuition regarding opinion aggregation.

Parallel aggregation is associative, so that the aggregated trust relationship will not change with the order of aggregation. This property, together with the associativity of sequence aggregation, is important to make the calculated result unique in different algorithm implementations which may aggregate trust paths in different order.

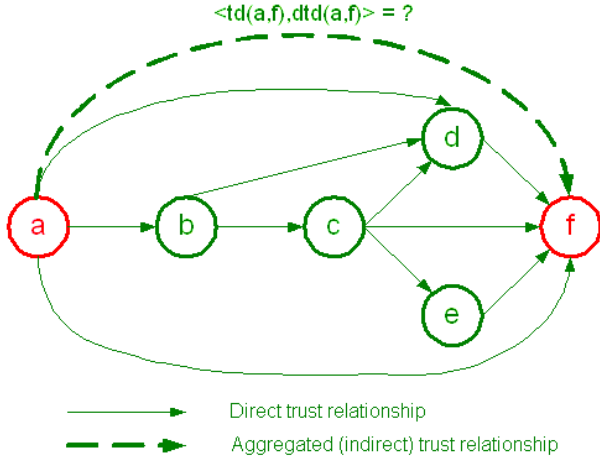


Figure 3: Example: trust aggregation in trust networks – What is the aggregated trust relationship between a and f ?

Finally, it is interesting to interpret this parallel aggregation from the view of trust revision with new information. $td(a, c)$ can be regarded as the initial opinion of entity a regarding trust in entity c , based on a 's direct interaction with c ; later, a learns new information from her/his trusted friends b_1, \dots, b_n regarding their opinion about c , and then a revises her/his opinion as $td^*(a, c)$, by synthesize her/his friends' opinions. If a 's opinion is based on a small number of direct interaction, those friends' opinions may have large influences on her/his revised opinion; on the other hand, if her/his original opinion is based on a big number of samples, her/his friends opinions may have smaller influences.

6.3 Trust Evaluation in a Network

We now show how to use sequence aggregation and parallel aggregation to aggregate trust in a network.

A *trust network* is a subgraph of a social network. A *social network* can be regarded as a directed graph with labeled arcs, where the vertices are entities such individuals and organizations in society, and the arcs are various social relationships, typically acquaintance relationships. In the context of trust, we only concern a special type of subgraphs of social networks, called *trust networks*, in which arcs represent inter-individual (or direct) trust relationships. An arc from vertex a to vertex b represents that a has direct trust relationship with b which is described as $\langle td(a, b, x, k), dtd(a, b, x, k) \rangle$. In general, all direct trust relationships in a social network form a directed graph usually with circles. For the purpose of deriving a new trust relationship by a trust network, trust circle should be eliminated. For this reason, we assume a concerned trust network is a directed acyclic graph (DAG).

More specifically, for a specific trust evaluation from trustor a to trustee z regarding information x in context k , we only consider a sub-network (of a real world trust network) with source node a and sink node z and the arcs in which trust relationships have the same x and k . Because in this specific subset of a trust network, all trust relationships have the same x and k , they are omitted.

From this point of view a trust network with source a and

sink z is a DAG (directed acyclic graph), represented as

$$TN = (E, A); \quad (46)$$

E is the set of entities, and $a, z \in E$; $A \in E \times E$, and each arc (e_i, e_j) ($e_j \neq z$) is labeled by trust (in belief) relationship

$$\langle td^b(e_i, e_j), dtd^b(e_i, e_j) \rangle; \quad (47)$$

each arc to the sink, (e_i, z) , is labeled by trust relationship

$$\langle td^x(e_i, e_j), dtd^x(e_i, e_j) \rangle \quad (48)$$

x is either b or p , that is, all arcs to the sink are either trust in belief relationship or trust in performance relationship.

An algorithm is designed to make trust aggregation in a network, which recursively simplifies a more complex network to a simpler one, by replacing multiple parallel paths into a single arc. Each replacement is made by using sequence or parallel aggregation.

$aggregate(a, z, TN)\{$

if (a, z) *is the only path from a to z in TN, stop;*
else if a has and only has one path to z, then {

use sequence aggregation to aggregate;
remove the last arc in this path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

} *else if a has multiple disintersected paths to z,*
then {

use parallel aggregation to aggregate all paths from a to z;
remove the last arc in each path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

} *else* {

calculate $N = neighbors(z)$;
for each $n_i \neq a$ in N do $aggregate(a, n_i, TN)$;
use parallel aggregation to aggregate all paths from a to z;
remove the last arc in each path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

}

}

The example shown in figures 3 to 7 demonstrates the trust aggregation process in a trust network.

In figure 3, the set of f 's neighbors is $\{a, c, d, e\}$. Apply the algorithm to aggregate trust in each neighbor first.

Figure 4 shows the process to aggregate trust between a and d . Since the subgraph with a as source and d as sink is still a network, apply the algorithm again. The neighbors of d are b and c ; a has direct trust in b , so no aggregation is applied; a has a single path to c , apply sequence aggregation, then remove the last arc (b, c) in the path, and add arc (a, c) (bold dash arc) corresponding to the aggregated trust relationship $\langle td^*(a, c), dtd^*(a, c) \rangle$ by the sequence aggregation. The result is shown in in figure 4 (b); now, return to the aggregation between a and d . a has three parallel paths to d , (a, d) , (a, b, d) and (a, c, d) . Apply parallel aggregation, then remove the last arc in each path, i.e. (a, d) , (b, d) and (c, d) , add a new arc, (a, d) (bold dash arc), which is corresponding to the aggregated trust relationship $\langle td^*(a, d), dtd^*(a, d) \rangle$ by the parallel aggregation. The result is shown in figure 4 (c).

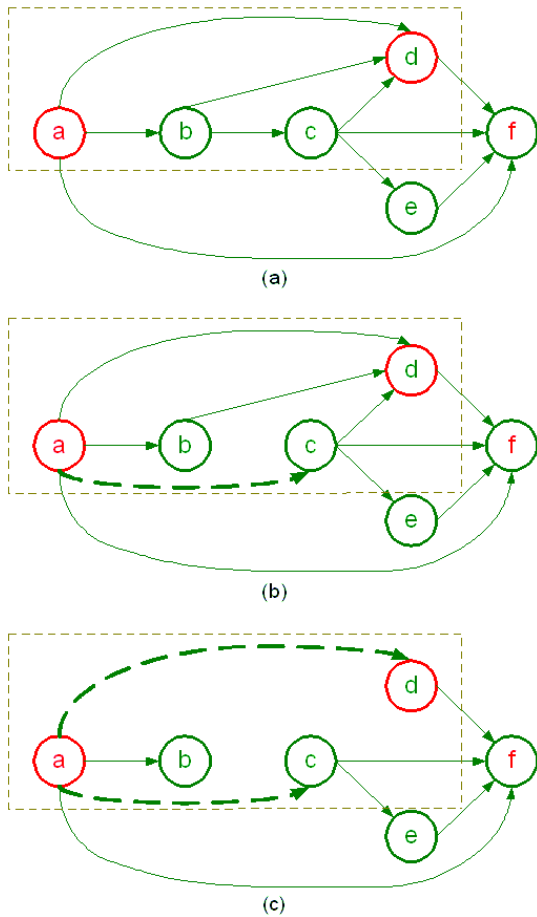


Figure 4: Example: trust aggregation in trust networks – aggregation between a and d

In figure 5, to aggregate trust between a and c , a has an arc to c , so the algorithm do nothing, and return to upper level of the process to aggregate trust between a and f .

Figure 6 shows the process to aggregate trust between a and e . There is a single path (a, c, e) , so apply sequence aggregation, then remove arc (c, e) and add arc (a, e) (bold dash arc) corresponding to the aggregated trust relationship by the sequence aggregation.

Returning to the process of aggregating trust between a and f , as shown in figure 7(a), there are four parallel trust paths, (a, f) , (a, d, f) , (a, c, f) and (a, e, f) , as shown in (a). Apply parallel aggregation, remove the last arc of each path, i.e. (a, f) , (d, f) , (c, f) and (e, f) , and add the new arc (a, f) (red bold dash arc) corresponding to the aggregated trust relationship by the parallel aggregation. Thus, we obtain the aggregated trust between a and f , as shown in 7(b).

7. TRUST QUANTIFICATION IN CERTIFICATE CHAINS

As discussed in section 1, a number of PKI trust models have been proposed [33] [23] [30] [37]. However, the explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths are missing.

In PKI, a certification path actually corresponds to a chain

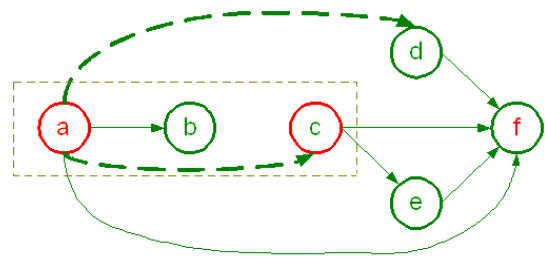


Figure 5: Example: trust aggregation in trust networks – no aggregation needed between a and c

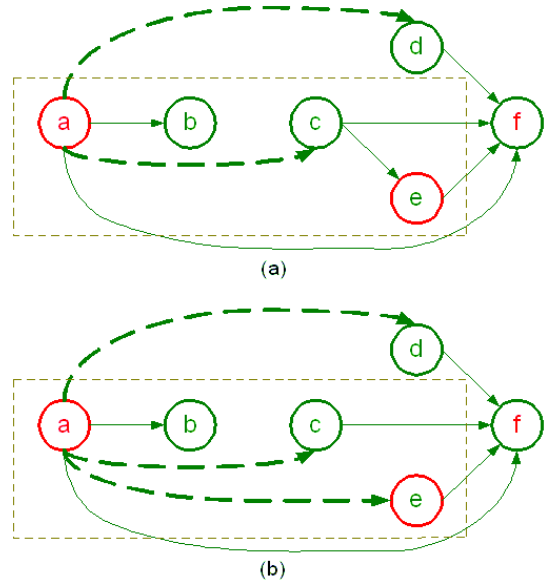


Figure 6: Example: trust aggregation in trust networks – Sequence aggregation between a and e

of trust relationships. What do these trust relationships exactly mean? On what things does each entity trust another in a credential chain? What is the specific context of each trust? In order to avoid misuse trust in PKI, we need to answer these questions and to explicitly and accurately represent trust in certificate chains.

In a typical public key validation model, a single certification path with shortest length is discovered and validated. An implicit risk evaluation criterion here is that a longer certification path has higher risk. This criterion actually assumes each certificate has the same level of risk. However, different certificates have different levels of risk, as they are produced by different organizations, with different identity standards. In order to make better decisions on authentication and authorization, it is important to quantify the risk associated with trust in certificate chains. When quantified evaluation of risk is introduced to certification paths, some very interesting issues emerge. There are multiple available certification paths, but which certification path should be chosen? The shortest path? or the one most trusted? Should more than one certification paths be considered? Should all certificate paths be considered? In this section, we explore the answers to these questions.

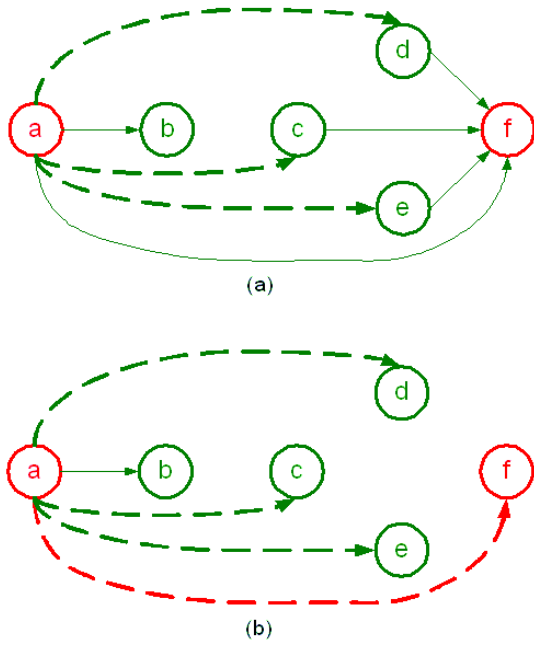


Figure 7: Example: trust aggregation in trust networks – parallel aggregation between a and f

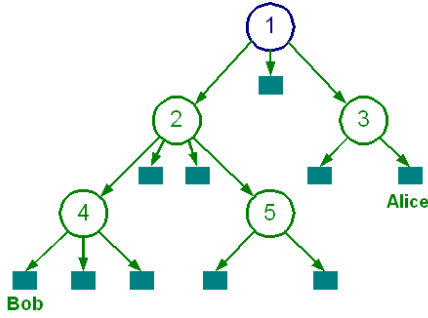


Figure 8: An example of hierarchical PKI (cited from Burr (1998))

In the following, we discuss the semantics of trust in PKI, the formal representation of trust relationships in PKI, and the quantified evaluation of risk associated with trust in certificate chains, by using our calculus of trust. There are different types of PKI architectures [3] [34][14] such as single-CA structure, hierarchical structure, mesh structure, bridge structure, and hybrid structure. We mainly discuss two representative types: hierarchical and mesh.

7.1 Trust in Hierarchical Structure PKI

An example of hierarchical PKI structure is shown in figure 8.

Alice needs to validate Bob’s public key. CA3 is Alice’s trust anchor, the CA Alice trusts; CA1 is root CA that everyone including Alice knows its public key; CA4 issues Bob’s public key certificate. In this case, the certificate chain will be CA1 - CA2 - CA4.

The semantics of the trust relationships in this certificate chain are as follows. The trust from CA2 to CA4 is that CA2

trusts CA4 regarding the validity of the certificates created and maintained by CA4. The trust from CA1 to CA2 is that CA1 trusts CA2 regarding CA2’s performance in the digital certification business, including (1) issuing and maintaining certificates to CA2’s clients, (2) auditing CA2’s subordinate CAs. The latter implies that CA1 trusts CA2 regarding what CA2 believes about the validity of the certificates created and maintained by CA2’s subordinate CAs. Similarly, the trust from Alice to CA1 is that Alice trusts what CA1 believes about the validity of the certificates created and maintained by CA1’s subordinate CAs and all descendants.

In the terminology of our formal trust model, CA2 trusts CA1 on performance in the context of “issuing and maintaining certificates”; CA1 trusts CA2 on belief in the context of “issuing and maintaining certificates”; Alice trusts CA1 in the same context. Depending the application, the context of trust could be more specific issues such as “accurate validation of key holder’s identity” and “good maintenance of CRL”.

Using our formal notation, the above trust relationships can be formally represented and calculated. The expectancy of Alice on Bob is that “public key pk(Bob) is Bob’s”. So, let x be this proposition. In this example, context k is set as “issuing and maintaining certificates”. Assume each of the above trust relationships has a different level of trust. They are formally represented as follows.

$$\begin{aligned}
 tr^b(Alice, CA1, k) &= \langle td^b(Alice, CA1, k), dtd^b(Alice, CA1, k) \rangle \\
 td^b(Alice, CA1, k) &= 0.98; \\
 dtd^b(Alice, CA1, k) &= 0.01; \\
 ud^b(Alice, CA1, k) &= 0.01;
 \end{aligned} \tag{49}$$

$$\begin{aligned}
 tr^b(CA1, CA2, k) &= \langle td^b(CA1, CA2, k), dtd^b(CA1, CA2, k) \rangle \\
 td^b(CA1, CA2, k) &= 0.92; \\
 dtd^b(CA1, CA2, k) &= 0.02; \\
 ud^b(CA1, CA2, k) &= 0.06;
 \end{aligned} \tag{50}$$

$$\begin{aligned}
 tr^p(CA2, CA4, k) &= \langle td^p(CA2, CA4, k), dtd^p(CA2, CA4, k) \rangle \\
 td^p(CA2, CA4, k) &= 0.96; \\
 dtd^p(CA2, CA4, k) &= 0.01; \\
 ud^p(CA2, CA4, k) &= 0.03;
 \end{aligned} \tag{51}$$

For hierarchical PKI the certification path is unique so we can directly apply sequence trust aggregation to evaluate the overall risk in the certificate path.

Using sequence aggregation, the aggregated trust relationship from CA1 to CA4 is calculated as

$$\begin{aligned}
 tr^p(CA1, CA4, k) &= \langle td^p(CA1, CA4, k), dtd^p(CA1, CA4, k) \rangle \\
 td^p(CA1, CA4, k) &= 0.883; \\
 dtd^p(CA1, CA4, k) &= 0.028; \\
 ud^p(CA1, CA4, k) &= 0.09;
 \end{aligned} \tag{52}$$

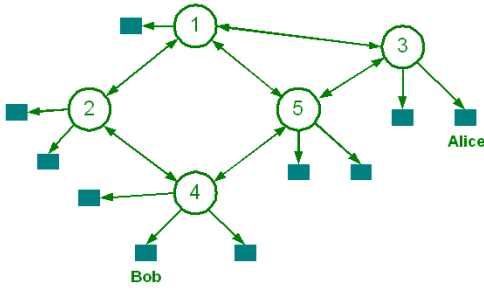


Figure 9: An example of mesh PKI, cited from Burr (1998)

The aggregated trust relationship from Alice to CA4 is calculated as

$$\begin{aligned}
 tr^P(Alice, CA4, k) &= \langle td^P(Alice, CA4, k), dtd^P(Alice, CA4, k) \rangle \\
 td^P(Alice, CA4, k) &= 0.866; \\
 dtd^P(Alice, CA4, k) &= 0.037; \\
 ud^P(Alice, CA4, k) &= 0.097;
 \end{aligned} \tag{53}$$

The single-CA PKI is a special case of hierarchical structure, with only a root CA. The semantics of trust, formal representation, and calculation are the same as in hierarchical PKI.

7.2 Trust in Mesh Structure PKI

An example of mesh PKI is illustrated in figure 9. In this case, CA3 is Alice's trust anchor, i.e. an CA Alice trusts, and by this CA, Alice finds a certificate chain to CA4, the issuer of Bob's public key certificate.

For different structures, while the certification path (certificate chain) construction methods [9][40] differ, from the view of a relying party (verifier or recipient), the trust relationships in a certificate chain have the same semantics.

The semantics of the trust relationships in this example are as follows.

In the following pairs of CAs, CA1 and CA2, CA2 and CA4, CA1 and CA5, CA4 and CA5, CA1 and CA3, as well as CA3 and CA5, each pair of CAs trust each other regarding the validity of the certificates created and maintained by the other; and each pair of CAs trust each other regarding what the other believes about the validity of the certificates created and maintained by a third CA. In the terminology of our formal trust model, each pair of CAs trust each other on both performance and belief in the context of "issuing and maintaining certificates"; Alice trusts CA3 on both performance and belief in the context of "issuing and maintaining certificates".

The trust relationships in a bridge PKI are similar to the ones in mesh PKI. The difference is that bridge CAs play the role of gateway and issue certificates only to CAs, so all trust relationships to bridge CAs are *trust in belief* type. A hybrid PKI consists of many CA groups of different types, so some gateway CAs have trust relationships cross groups; and some others has trust relationships within their groups.

The formal representation of the above trust relationships are similar to the ones given in hierarchical structure. In the following discussion, we give only that part of them needed for the trust calculation examples.

As discussed earlier, when introducing quantified values of trust in certificate chains, we need to answer a number of interesting questions. We discuss those questions in two representative cases.

7.2.1 Using One-path certification

By a traditional certification path construction method, the shortest path, CA3->CA5->CA4, will be used to validate Bob's public key. Assume that the trust relationships are as follows.

Alice highly trust her trust anchor, CA3.

$$\begin{aligned}
 tr^b(Alice, CA3, k) &= \langle td^b(Alice, CA3, k), dtd^b(Alice, CA3, k) \rangle \\
 td^b(Alice, CA3, k) &= 0.99; \\
 dtd^b(Alice, CA3, k) &= 0.0; \\
 ud^b(Alice, CA3, k) &= 0.01;
 \end{aligned} \tag{54}$$

Trust from CA3 to CA5 is not high, even somewhat negative.

$$\begin{aligned}
 tr^b(CA3, CA5, k) &= \langle td^b(CA3, CA5, k), dtd^b(CA3, CA5, k) \rangle \\
 td^b(CA3, CA5, k) &= 0.65; \\
 dtd^b(CA3, CA5, k) &= 0.25; \\
 ud^b(CA3, CA5, k) &= 0.1;
 \end{aligned} \tag{55}$$

The trust from CA5 to CA4 is fairly uncertain.

$$\begin{aligned}
 tr^P(CA5, CA4, k) &= \langle td^P(CA5, CA4, k), dtd^P(CA5, CA4, k) \rangle \\
 td^P(CA5, CA4, k) &= 0.75; \\
 dtd^P(CA5, CA4, k) &= 0.0; \\
 ud^P(CA5, CA4, k) &= 0.25;
 \end{aligned} \tag{56}$$

By sequence aggregation, the derived trust from CA3 to CA4 is

$$\begin{aligned}
 tr^P(CA3, CA4, k) &= \langle td^P(CA3, CA4, k), dtd^P(CA3, CA4, k) \rangle \\
 td^P(CA3, CA4, k) &= 0.488; \\
 dtd^P(CA3, CA4, k) &= 0.188; \\
 ud^P(CA3, CA4, k) &= 0.324;
 \end{aligned} \tag{57}$$

the derived trust from Alice to CA4 is

$$\begin{aligned}
 tr^P(Alice, CA4, k) &= \langle td^P(Alice, CA4, k), dtd^P(Alice, CA4, k) \rangle \\
 td^P(Alice, CA4, k) &= 0.483; \\
 dtd^P(Alice, CA4, k) &= 0.186; \\
 ud^P(Alice, CA4, k) &= 0.331,
 \end{aligned} \tag{58}$$

which shows a weak trust relationship from Alice to CA4, so even though certification is verified successfully along the certificate chain, Alice may still not trust the validity of Bob's public key.

However, a longer path, CA3 -> CA1 -> CA2 -> CA4, with a higher level of trust, may make the derived trust from Alice to CA4 in a acceptable level.

Assume trust relationship from CA3 to CA1 being

$$\begin{aligned}
& tr^b(CA3, CA1, k) \\
& = \langle td^b(CA3, CA1, k), dtd^b(CA3, CA1, k) \rangle \\
& td^b(CA3, CA1, k) = 0.98; \\
& dtd^b(CA3, CA1, k) = 0.01; \\
& ud^b(CA3, CA1, k) = 0.01;
\end{aligned} \tag{59}$$

other trust relationships are as assumed earlier.

By sequence aggregation along this longer path, the derived trust relationship from CA3 to CA4 is:

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.866; \\
& dtd^p(CA3, CA4, k) = 0.037; \\
& ud^p(CA3, CA4, k) = 0.097;
\end{aligned} \tag{60}$$

the derived trust relationship from Alice to CA4 is:

$$\begin{aligned}
& tr^p(Alice, CA4, k) \\
& = \langle td^p(Alice, CA4, k), dtd^p(Alice, CA4, k) \rangle \\
& td^p(Alice, CA4, k) = 0.849; \\
& dtd^p(Alice, CA4, k) = 0.045; \\
& ud^p(Alice, CA4, k) = 0.106,
\end{aligned} \tag{61}$$

and Alice may accept this level of trust to Bob's public key.

This example shows when quantified risk is introduced in certificate chains, the most trustworthy certification path with respect to "issuing and maintaining certificates", need not be the shortest path.

Practical application of the calculus here might be to provide a framework for accepting or rejection a validated. The risk is that one or more CA's in the chain may have erroneously bound an identity and public key, allowing for subversion of the binding of Bob's identity and the public key in his certificate. If some chain chosen (e.g. the shortest) yields an unacceptably low level of trust, then another chain may be sought and validated, in an effort to find a chain with a high enough trust value. It is important to note here that the different paths through CAs are disjoint and hence statistically independent. In particular, knowledge that all the signatures on one path "checked out" in no way influences the probability that the signatures on another path will, because only one CA signs Bob's certificate. The situation changes significantly though when multiple CAs sign a certificate.

7.2.2 Using multiple-paths certification

Inspired by network reliability, Reiter and Stubblebine's research [35] [36] proposed "resilient authentication" by using redundant multiple independent paths to increase assurance or reliability. The authors suggest two types of independence: (1) node-disjoint paths with bounded length; and (2) k -connective paths with bounded length, in which to break all path, k nodes have to be removed. By their approach, one misbehaving node (CA) will compromise at most one path. So, in the context of public key certification validation, multiple certification paths will make certificate validation more reliable. The drawback is that there is no quantified trust evaluation on certificates or certification authorities, and there is an implicit assumption that the risk

level of each certificate is the same. By combining this approach with our calculus of trust, a better risk evaluation can be made. We discuss this method as follows.

First, use Reiter and Stubblebine's BDP algorithm [35] to get a set of node-disjoint paths of bounded length. Assume that the totally number of such node-disjoint paths is k .

Second, for each path, use sequence aggregation to calculate the aggregated trust in each path.

The aggregated result is a triple $\langle td, dtd, ud \rangle$. Consider the semantics of these degrees. td represents the conditional probability of the trust anchor believing that the certificate made by the target, given fact that the certificate is made by the target; dtd is the conditional probability of disbelief in the certificate; and ud is the degree of uncertainty in current information status. When more relevant information becomes available and the uncertainty is resolved, the ud will partly go to belief, and partly go to disbelief. In the extreme cases, ud completely goes to belief or disbelief. Consequently this triple can be regarded as a probability interval $[td, td + ud]$, (and now, $td + dtd = 1$.)

Assume that the i^{th} path has probability of p_i being valid, and the aggregated trust in the i^{th} path is $\langle td_i, dtd_i, ud_i \rangle$. Then, we have

$$td_i \leq p_i \leq td_i + ud_i. \tag{62}$$

Since those paths are node-disjoint, they are statistically independent. So, the probability of this public key, certificated by k multiple parallel certification paths, being valid is,

$$p = 1 - \prod_{i=1}^n (1 - p_i) \tag{63}$$

Because each p_i has a range, that probability also has a lower bound and upper bound, i.e.

$$1 - \prod_{i=1}^n (1 - td_i) \leq p \leq 1 - \prod_{i=1}^n (1 - (td_i + ud_i)) \tag{64}$$

Returning to the example, BDP algorithm outputs two paths: CA3 -> CA5 -> CA4, and CA3 -> CA1 -> CA2 -> CA4. For each of them, the trust calculated through sequence aggregation is as follows.

For path 1: CA3 -> CA5 -> CA4,

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.488; \\
& dtd^p(CA3, CA4, k) = 0.188; \\
& ud^p(CA3, CA4, k) = 0.324;
\end{aligned} \tag{65}$$

the interval of the probability that CA4's certificate for Bob's public key being valid is [0.488, 0.812], obviously, which is very uncertain; for path 2: CA3 -> CA1 -> CA2 -> CA4,

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.866; \\
& dtd^p(CA3, CA4, k) = 0.037; \\
& ud^p(CA3, CA4, k) = 0.097;
\end{aligned} \tag{66}$$

the interval of the probability that CA4's certificate for Bob's public key being valid is [0.866, 0.963].

So, by using these two paths, the probability that CA4's certificate for Bob's public key being valid has a lower bound of

$$1 - (1 - 0.488) \cdot (1 - 0.866) = 0.931, \quad (67)$$

and has an upper bound of

$$1 - (1 - 0.812) \cdot (1 - 0.963) = 0.993. \quad (68)$$

So, by using multiple paths, the interval of the probability of the certificate being valid is [0.931, 0.993].

This example shows that using multiple paths for certification is much more certain and more reliable than using single certification path for validation.

The above multiple paths certification has a drawback that to derive the trust in the target, for each CA in a path, only the trust from the proceed CA to this CA is taken into account, and other CAs' trust relationships to this CA are not considered. For example, in path CA3 -> CA5 -> CA4, to evaluate trust in CA5, only CA3's trust in CA5 is considered; CA1's opinion on CA5 is completely neglected. However, CA3's opinion may be based on a small number of encounters between them; CA1's opinion may be based on a much greater number of encounters.

From the perspective of trust in social networks, to avoid the above possible bias in trust evaluation, using the trust relationships in a network of certificates will make the trust anchor get more opinions about a CA from this CA's neighbor CAs. In this way, the trust anchor can be more objectively evaluate this CA.

Return to the example story. Assume that the trust from CA1 to CA5 is

$$\begin{aligned} tr^b(CA1, CA5, k) &= \langle td^b(CA1, CA5, k), dtd^b(CA1, CA5, k) \rangle \\ td^b(CA1, CA5, k) &= 0.91; \\ dtd^b(CA1, CA5, k) &= 0.02; \\ ud^b(CA1, CA5, k) &= 0.07, \end{aligned} \quad (69)$$

and the opinion is based on 5,000 encounters; CA3's direct trust in CA5 is based on just 100 encounters. Then by using parallel aggregation, the aggregated trust from CA3 to CA5 will be

$$\begin{aligned} tr^b(CA3, CA5, k) &= \langle td^b(CA3, CA5, k), dtd^b(CA3, CA5, k) \rangle \\ td^b(CA3, CA5, k) &= 0.887; \\ dtd^b(CA3, CA5, k) &= 0.033; \\ ud^b(CA3, CA5, k) &= 0.08, \end{aligned} \quad (70)$$

which is significantly different from CA3's direct trust in CA5 (formula 55).

In the following, we briefly discuss how to use the trust relationships in a network of certificates, to evaluate the trust in a CA; and leave the detailed algorithm design and analysis for future research.

For each concerned CA, the trust from the trust anchor can be evaluated by using the algorithm given in section 6.3; then use the derived trust as heuristic information in Reiter and stubblebine's multiple independent certification paths discovery.

Ideally, for trust evaluation in a social network, the more information is used, the more accurate the evaluation is.

It would be perfect to use complete information (all relevant trust relationships) in the network. However in the real world, a social network usually is huge. Consider the cost for computing, it is unreal to use all information. By Simon's theory of *bounded rationality*, a decision making process in the real world is limited by bounded rationality i.e. the "rational choice that takes into account the cognitive limitations of the decision maker - limitations of both knowledge and computational capacity". Thus, it is acceptable to use just partial but most relevant information (trust relationships) to make trust evaluation in a huge social network.

8. CONCLUDING REMARKS

In order to explicitly represent trust and to quantify the risk associated with trust in public key infrastructure (PKI) and identity management (IdM), we introduced a formal semantics based calculus of trust, and demonstrated how to apply the trust calculus, to formally represent trust relationships and to quantitatively evaluate the risk associated with trust in public key certificate chains. This research shows that after introducing formal representation and quantification of trust in certificate chains, for using one-path certification, the shortest certification path need not be the most trustworthy certification path, and that a chain with an acceptably high level of trust should be constructed for validation; for using multi-path certification, multiple independent certification paths provides much more reliable and certain public key certification validation.

To continue the work presented in this paper, the future work can go further in several directions. First, knowledge that a certificate has been validated by some path (or some set of paths) clearly impacts the probability that the certificate will be validated by another. This is a feature of the analysis yet to be developed. Other future work is to develop effective and efficient trust aggregation algorithms in huge size social networks; use trust calculus as heuristic information in public key certification path building, and other applications of the calculus of trust, for example, modeling trust in privacy protection in healthcare.

9. ACKNOWLEDGMENTS

The US Department of Homeland Security, through grant award number 2006-CS-001-000001 under the auspices of the Institute for Information Infrastructure Protection (I3P) research program, partly supported this work. 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 the US Department of Homeland Security, the I3P, or Dartmouth College, which manages the I3P program.

10. REFERENCES

- [1] K. Blomqvist. The many faces of trust. *Scandinavian Journal of Management*, 13(3):271-286, 1997.
- [2] D. Bodeau. *Sharing Protected Identity and Credential Information (SPICI) Framework for Assessable Identity and Privacy Protection*. The MITRE Corporation, 2008.
- [3] W. Burr. Public key infrastructure (pki) technical specifications: Part a - technical concept of operations. 1998.

- [4] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.
- [5] C. Dellarocas and P. Resnick. Online reputation mechanisms a roadmap for future research. In *First Interdisciplinary Symposium on Online Reputation Mechanisms*, 2003.
- [6] R. Demolombe. To trust information sources: a proposal for a modal logical framework. In C. Castelfranchi and Y.-H. Tan, editors, *Trust and Deception in virtual societies*, pages 111–124. Kluwer Academic Publishers, 2001.
- [7] L. Ding, P. Kolari, S. G. Finin, A. Joshi, Y. Peng, and Y. Yesha. Modeling and evaluating trust network inference. In *The Seventh International Workshop on Trust in Agent Societies, at AAMAS 2004*, 2005.
- [8] C. Ellison and B. Schneier. Ten risks of pki: what you're not being told about public key infrastructure. *Computer Security Journal*, XVI(1), 2000.
- [9] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman, and S. Proctor. Building certification paths: Forward vs. reverse. In *The 10th Annual Network and Distributed System Security Symposium*, 2001.
- [10] R. Forno and W. Feinbloom. Pki: A question of trust and value. *Communication of the ACM*, 44(6), June 2001.
- [11] J. A. Golbeck. *Computing and Applying Trust in Web-Based Social Networks*. Ph.D. Thesis, University of Maryland, College Park, 2005.
- [12] R. Guha and R. Kumar. Propagation of trust and distrust. In *WWW2004*, 2004.
- [13] A. Hajek. Probability, logic, and probability logic. In L. Goble, editor, *Philosophical Logic*. Blackwell Publishing, 2001.
- [14] S. Hanna and P. Hesse. Approaches to certificate path discovery (slides). In *PKI'2004*, 2004.
- [15] J. Huang. *Knowledge Provenance: An Approach to Modeling and Maintaining The Evolution and Validity of Knowledge*. PhD Thesis, University of Toronto, <http://hdl.handle.net/1807/11112>, December 2007.
- [16] J. Huang and M. S. Fox. An ontology of trust – formal semantics and transitivity. In *Proceedings of The Eighth International Conference on Electronic Commerce*, pages 259–270. ACM, 2006.
- [17] J. Huang and D. Nicol. A formal semantics based calculus of trust. In *ITI Research Report, (to be submitted for journal publication)*, University of Illinois at Urbana-Champaign, 2008.
- [18] A. Josang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 9(3):279–311, 2001.
- [19] A. Josang, E. Gray, and M. Kinatader. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems Journal*, 4(2):139–161, 2006.
- [20] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.
- [21] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [22] R. Levien. *Attack-resistant trust metrics*. PhD Thesis, University of California, Berkeley, USA, 2002.
- [23] J. Linn. Trust models and management in public key infrastructures. *RSA Laboratories*, 2000.
- [24] N. Luhmann. *Trust and Power*. John Wiley & Sons Ltd, 1979.
- [25] S. P. Marsh. *Formalising Trust as a Computational Concept*. Ph.D. Thesis, University of Stirling, 1994.
- [26] S. P. Marsh and M. R. Debben. Trust, untrust, distrust and mistrust - an exploration of the dark(er) side. In *Proceedings of iTrust2005, LNCS 3477*, pages 17–33, 2005.
- [27] U. M. Maurer. Modelling a public-key infrastructure. In *ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security*, pages 325–350, London, UK, 1996. Springer-Verlag.
- [28] R. Mayer, J. Davis, and F. Schoorman. An integrative model of organizational trust. *Academic of Management Review*, 20(3):709–734, 1995.
- [29] Microsoft. Pki trust models (slides). 2004.
- [30] T. Moses. Pki trust models.
- [31] L. Mui and A. Halberstadt. A computational model of trust and reputation. In *Proc. 35th Hawaii Int. Conf. on System Sciences*, 2002.
- [32] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford Digital Library Technologies Project*, 1998.
- [33] R. Perlman. An overview of pki trust models. *IEEE Network*, 13:38–43, 1999.
- [34] W. T. Polk and N. E. Hastings. Bridge certification authorities: Connecting b2b public key infrastructures. 2000.
- [35] M. K. Reiter and S. G. Stubblebine. Resilient authentication using path independence. *IEEE Trans. Comput.*, 47(12):1351–1362, 1998.
- [36] M. K. Reiter and S. G. Stubblebine. Authentication metric analysis and design. *ACM Trans. Inf. Syst. Secur.*, 2(2):138–158, 1999.
- [37] C. Satizabal, R. Paez, and J. Forne. Pki trust relationships: from a hybrid architecture to a hierarchical model. 2006.
- [38] B. Yu and M. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165, 2000.
- [39] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *Proceedings of The Eighth International Conference on Electronic Commerce*, pages 225–234. ACM, 2006.
- [40] M. Zhao and S. W. Smith. Modeling and evaluation of certification path discovery in the emerging global pki. In *EuroPKI2006*, 2006.

IdTrust 2009, NIST, April 14-16, 2009

A Calculus of Trust and Its Application to PKI and Identity Management

Jingwei Huang & David M. Nicol*

Information Trust Institute

*Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

{jingwei,nicol}@iti.uiuc.edu



1

Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



2

Specific Motivation (to IdM &PKI)

- ❑ Trust is a foundation for IdM and PKI
 - ❑ Ten risks in PKI (Ellison&Schneier,2000)
 - ❑ Incident: VeriSign, cert for Microsoft
 - ❑ “Who do we trust, and for what?” [Ellison&Schneier,2000]
- ❑ Current PKI trust models
 - Assume -- each certificate has the same level of risk
 - Evaluate risk -- the longer a certification path is, the higher risk is
 - Focus on:
 - **Structure of PKI (e.g. hierarchical, mesh, bridge)**
 - **Certification path discovery (to find shortest one)**
- ❑ Question: **How to quantify the risk associated with trust in PKI?**



3

General Motivation

- ❑ On the Web, people need to interact with “strangers”.
- Trust becomes a crucial problem!
- **How can we make trust judgment on the entities we don't know (or are not familiar with)?**



4

Methodology

- Our approach of trust modeling
 - Abstract concepts of trust from social studies
 - Formalize in logic
 - Extend logical model of trust to uncertainty model
 - Apply in real domain and make further improvement

- Principles to follow:
 - Semantics consistency
 - Common sense consistency
 - simplicity



5

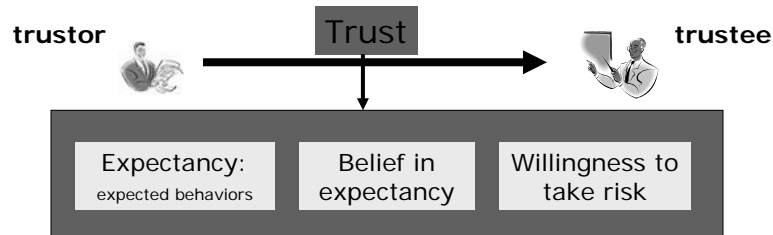
Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



6

Our View of Trust



- Trust is a **mental state** comprising:
 - (1) **expectancy**
 - (2) **belief** – expected behaviours to be true
 - (3) **willingness to take risk** for that belief



7

Trust in Belief / Performance

- By different expectancy, two fundamental types of trust can be identified:
 - Trust in performance
 - **trust what trustee performs** in a context
e.g. trust ftd.com to deliver a bouquet as ordered.
 - Trust in belief
 - **trust what trustee believes** in a context
e.g. trust the opinion of a wine expert regarding the quality of wine products
- Trust is context-dependent
e.g. trust a physician in healthcare but not in finance



8

Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



9

Formal Semantics of Trust

- Uses a logical language of situation calculus.
- We develop uncertain trust model, based on a simplified version
 - Simplifies notation
 - The obtained results remain true for the full version of logic model.



10

Two Types of Trust

- $trust_p(d,e,x,k)$ (*trust in performance*)
--- “Trustor d trusts trustee e on a thing x made by e in context k ”

- $trust_b(d,e,x,k)$ (*trust in belief*)
--- “Trustor d trusts trustee e on trustee’s belief x in context k ”



11

Other Notation

- Distrust
 - $distrust_p(d,e,x,k) \iff$
($madeBy(x,e,k) \rightarrow believe(d, k \sim \rightarrow neg(x))$)

 - $distrust_b(d,e,x,k) \iff$
($believe(e,k \sim \rightarrow x) \rightarrow believe(d, k \sim \rightarrow neg(x))$)



12

Trust Reasoning

- **Trust in belief is transitive**

$trust_b(a,b,x,k) \ \& \ trust_b(b,c,x,k) \ \rightarrow$
 $trust_b(a,c,x,k)$

- **Propagation of trust in performance via trust in belief**

$trust_b(a,b,x,k) \ \& \ trust_p(b,c,x,k) \ \rightarrow$
 $trust_p(a,c,x,k)$



13

Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



14

Formal Semantics of Uncertainty in Trust

- Trust is not binary
- Using probability logic [Hajek, 2001], we define:
 - **Degree of trust in performance**
 $td_p(d,e,x,k) = pr(believe(d,x) | madeBy(x,e,k) \& beTrue(k))$
The sample space based on history of interactions
 - **Degree of trust in belief**
 $td_b(d,e,x,k) = pr(believe(d,x) | believe(e,x) \& beTrue(k))$
 - Degree of distrust defined similarly



15

Measurement of Uncertainty

Trust degree is measured by the fraction of successful encounters

$$td = n/m, \quad dtd = l/m; \quad n + l \leq m$$

m – total encounters

n – successful encounters;

l – negative encounters.

- General form
 $td = \sum_{i=1, \dots, m} ep(i)/m,$
 $dtd = \sum_{i=1, \dots, m} en(i)/m$



16

More on Uncertainty

- ❑ Not all encounters need to yield 'positive' or 'negative' as result
- ❑ Cognitively there are three mental states:
 - believed
 - disbelieved
 - undecidable.
- ❑ We model multiple sources of uncertainty:
 - **Randomness**, inaccuracy, complexity, **incomplete information**
- ❑ Uncertainty is represented as probability distribution (*td, dtd, ud*) or simply (*td, dtd*).



17

Trust Calculation in Trust Networks

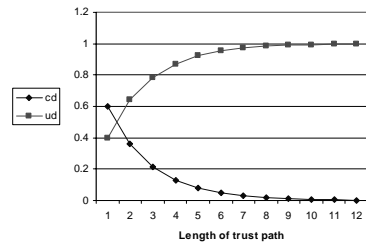
- ❑ A trust network is a directed graph, comprising a set of nodes – entities, and edges – trust relationships
 - A subset of a social network
- ❑ Calculation of trust from a trustor to a trustee though trusted friends in a network?
- ❑ Two basic operators:
 - **Sequence aggregation**: to aggregate trust in a chain
 - **Parallel aggregation**: to aggregate trust in parallel structure



18

Sequence Aggregation

- When a trusts b (in belief), and b trusts c (in either belief or performance), how much does a trust c ?
 - For simpler notation, we omit subscripts b (for trust in belief) and p (for trust in performance)
- From the formal definitions, we derived and proved a theorem defining td , dtd , and $cd = td+dtd$
 - cd is “degree of certainty”



19

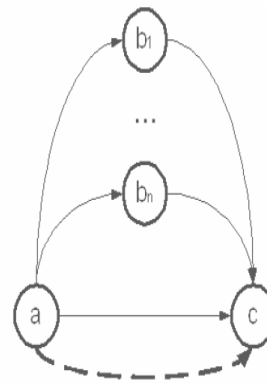


Parallel Aggregation

- Combine independent trust paths.
- Use sequence aggregation on paths. e.g.

$$td(a,bi,c) = td(a,bi) * td(bi,c) + dtd(a,bi) * dtd(bi,c)$$
- Aggregated trust degree of trust weighted average
 e.g. aggregated trust from a to c ,

$$td(a,c)' = \frac{[m(a,c) * td(a,c) + m(b1,c) * td(a,b1,c) + \dots + m(bn,c) * td(a,bn,c)]}{[m(a,c) + m(b1,c) + \dots + m(bn,c)]}$$



- Direct trust relationship
- - -→ Aggregated (indirect) trust relationship



- Path weight proportional to # encounters

20

Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



21

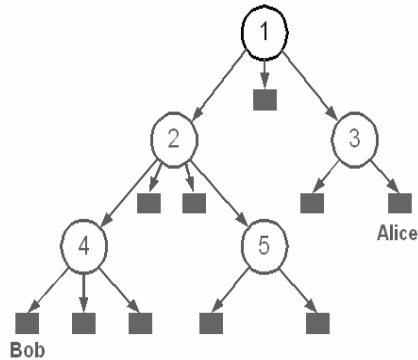
Trust in PKI

- Motivating question:
 - How to quantify the risk associated with trust in PKI?
- **uncertainty** is represented as **probability distribution** on <believed, disbelieved, unknown>
- Apply the calculus of trust to quantifying risk associated with trust in PKI



22

Trust Evaluation in Hierarchical PKI

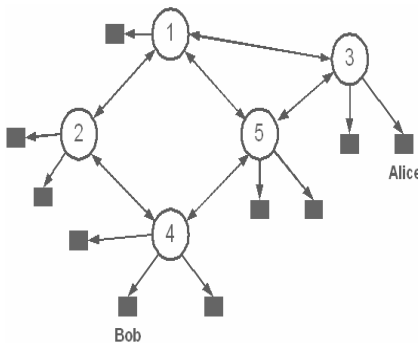


- Chain of trust:
 Alice – CA3 – CA1 – CA2 - CA4
 $tr^b(A, CA3, pk.validity) = (1, 0, 0)$
 $tr^b(CA3, CA1, pk.validity) = (0.98, 0.01, 0.01)$
 $tr^b(CA1, CA2, pk.validity) = (0.92, 0.02, 0.06)$
 $tr^p(CA2, CA4, pk.validity) = (0.96, 0.01, 0.03)$
- By sequence aggregation
 $tr^b(A, CA4, pk.validity) = (0.866, 0.037, 0.097)$



23

Trust Evaluation in Web PKI



- Multiple chains of trust exist
 1. Alice-CA3-CA1-CA2-CA4
 2. Alice-CA3-CA5-CA4
- Assume path 1 the same as before
 $tr^b(A, CA4, pk.validity) = (0.866, 0.037, 0.097)$
- Assume path 2:
 $tr^b(CA3, CA5, pk.validity) = (0.65, 0.35, 0.1)$
 $tr^b(CA5, CA4, pk.validity) = (0.75, 0.00, 0.25)$
 then
 $tr^b(A, CA4, pk.validity) = (0.488, 0.188, 0.324)$

- For using **one-path certification**, the shortest certification path may not be the most trustworthy path;
- In practice, if the shortest path has unacceptable level of trust, another path with high enough level of trust needs to be found



24

Risk in Multiple Independent Trust Paths

- If use multiple independent paths for certification, What is the risk level ?
- Assume path i having aggregated trust level (td_i, dtd_i, ud_i)
- Let p_i be the probability of certification path i being valid, then

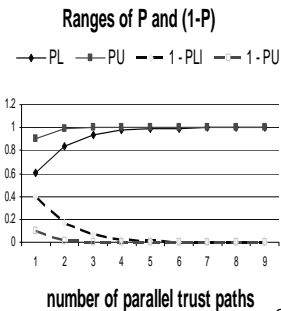
$$td_i \leq p_i \leq td_i + ud_i.$$

- The probability of at least one of n paths being valid will be:

$$p = 1 - \prod_{i=1}^n (1 - p_i)$$

$$1 - \prod_{i=1}^n (1 - td_i) \leq p \leq 1 - \prod_{i=1}^n (1 - (td_i + ud_i))$$

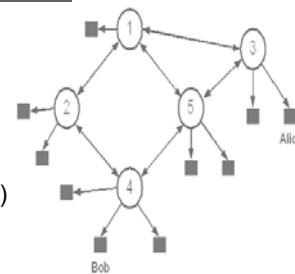
- So, the probability of multiple independent certification paths being compromised, $1-p$, decreases exponentially
- In general, multiple independent trust paths significantly increase trustworthiness and certainty



25

Example

- By path-1: CA3-CA1-CA2-CA4
 $\text{tr}^b(\text{CA3}, \text{CA4}, \text{pk. validity}) = (0.866, 0.037, 0.097)$
- The probability of path-1 being valid, p_1 in $[0.866, 0.963]$
 $0.963 = td + ud = 0.866 + 0.097$
- By path-2: CA3-CA5-CA4 $\text{tr}^b(\text{CA3}, \text{CA4}, \text{pk. validity}) = (0.488, 0.188, 0.324)$
- The probability of path-2 being valid, p_2 in $[0.488, 0.812]$
- Evaluate the probability (p) of at least one path being valid:
 lower bound: $1 - (1 - 0.866)(1 - 0.488) = 0.931$
 upper bound: $1 - (1 - 0.963)(1 - 0.812) = 0.993$
 so, p in **[0.931, 0.993]**,
 which is **much more certain and trustworthy than any single-path validation**,
 $[0.866, 0.963]$ and $[0.488, 0.812]$.



26

Outline

1. Motivation
2. Trust conceptualization
3. Trust formalization / Formal semantics
4. A formal semantics based calculus of trust
5. Quantifying risk associated with trust in PKI
6. Discussion and Concluding remarks



27

Concluding Remarks

- **The semantics of trust needs to be defined explicitly and accurately.**
 - To avoid misuse of trust
 - To understand trust deeper
 - To answer questions about trust clearer and more accurate
 - To make model design clearer
- Our research shows:
 - ***Trust in belief* is transitive; *trust in performance* is not, but through trust in belief it can propagate in a social network.**
 - **With the growth of the length of a trust path, trust along the path decreases multiplicatively;**
 - **Multiple independent trust paths significantly increase the trustworthiness and certainty.**



28

Next...

- Use quantified risk as heuristics for certificate path discovery
- We are looking for industrial partners to put it into practice :)



29

Thank you !

&

Questions ?



30

IDtrust2009

Session 4 – Panel

What Is Special About My Application?

A Health and Healthcare Perspective

April 14, 2009 | 4:00 – 5:30 pm

Presenter

Walter G. Suarez, MD, MPH

President and CEO, Institute for HIPAA/HIT Education and Research

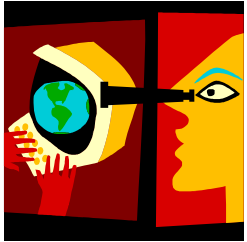
President, Public Health Data Standards Consortium

Co-Chair, HITSP Security, Privacy and Infrastructure Technical Committee

Basic Concepts

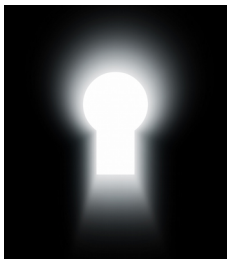
- What is Privacy (of health information)?

- An individual's (or organization's) right to determine whether, what, when, by whom and for what purpose their personal health information is collected, accessed, used or disclosed



- What is Security (of health information)?

- A defined set of administrative, physical and technical actions used or taken to protect the confidentiality, availability and integrity of health information



Source: HITSP Vocabulary – modified and expanded from 45 CFR 164.304

Basic Concepts

■ Confidentiality

- The property that data or information is not made available or disclosed to unauthorized persons or processes

■ Integrity

- The property that data or information has not been altered or destroyed in an unauthorized manner

■ Availability

- The property that data or information is accessible and usable upon demand by an authorized person

Source: 45 CFR 164.304

Privacy and Security Scenarios

- Patient with sensitive conditions (AIDS, mental health)
- Patient's ability to control granular levels of health information (who can access what, when, for what purpose; selective restriction of access; opt-in/opt-out)
- Patient asks for accounting of disclosures
- Patient that retracts/changes an existing consent
- Need to allow access on emergency situations ('Break the Glass')
- VIP (politician, movie star, sports figure)
- Domestic violence victims
- Daughter with sensitive tests hidden from Parent

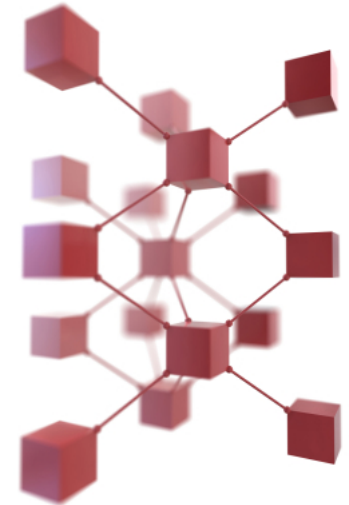
What is Special about Health and Healthcare (1)



- Medical records among the most sensitive information about a person
- Health care is an information-driven field
 - Everything about the health care system involves information
 - Information is much more complex than other industries (amount, type, frequency)
- Health information is central to the doctor-patient relationship
- Privacy and security of health information are central to the doctor-patient relationship

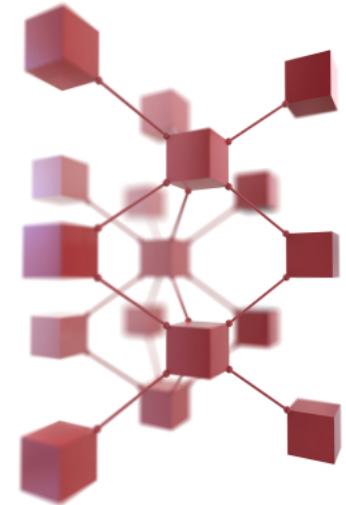
What is Special about Health and Healthcare (2a)

- Health care is a complex system, when it comes to health information
 - Many actors (patient, provider, health plan, employer, government, public health, researchers, vendor, etc)
 - Various types of information (demographic, clinical, financial)
 - Many processes related to health information (collection, creation, maintenance, access, use, disclosure)
 - Many devices associated with, and used in the care of patients (hospital/medical devices, home monitoring devices, others)
 - Various ways of delivering care (in person, remotely/telemedicine, interactively)



What is Special about Health and Healthcare (2b)

- Health care is a complex system, when it comes to health information (cont)
 - Different purposes (treatment, payment, operations, public health, research, judicial, legal, etc)
 - Many places where health information reside
 - Lack of common identifiers and other standards
 - Patient IDs (each provider, each payer)
 - Provider IDs (although being simplified with the implementation of the National Provider Identifier)
 - Payer IDs
 - Vendor IDs
 - Medical Device IDs



What is Special about Health and Healthcare (3a)



- Many laws
 - Federal laws, including HIPAA, Privacy Act, Education Records Law, Mental Health Records Laws, Public Health information laws
 - State laws – patchwork of varying types and levels of state privacy laws, few addressing health privacy and security in a comprehensive fashion
- Different policies and practices created and used by organizations
 - Many go above and beyond what federal/state laws require

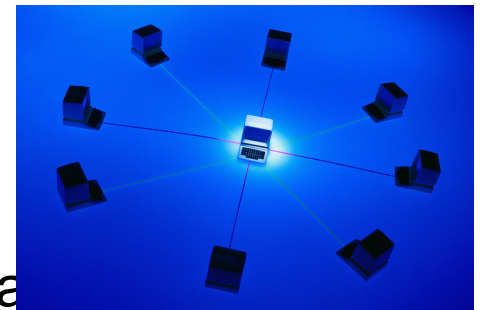
What is Special about Health and Healthcare (3b)



- Laws provide rights to consumers to control their information (through Consumer Consent and Patient Authorization)
- Laws provide for boundaries/restrictions on what entities that collect, access, use and disclose health information can do with it
- Laws also required certain security protections be implemented by entities on the health information they collect, maintain, use or disclose

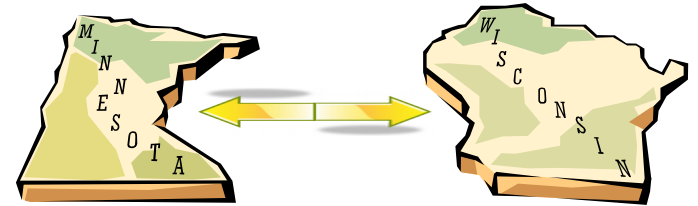
What is Special about Health and Healthcare (4)

- Increasing complexities
 - Expanded use of electronic health records
 - Increased electronic communications between patients and the health care system (i.e., websites, email)
 - Electronic networks (Regional Health Information Exchanges, NHIN)
 - Evolving personal health records
 - Different levels of 'sensitive' health information



What is Special about Health and Healthcare (5)

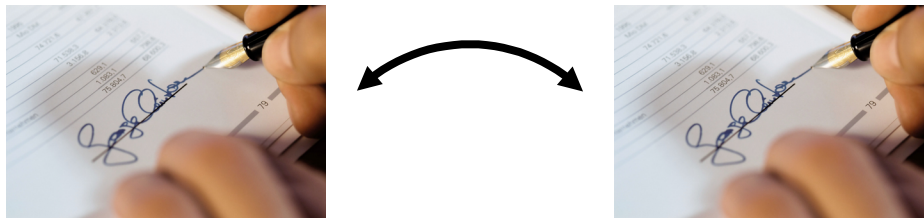
- Inter-jurisdictional Portability



- Consumer privacy consent laws and requirements, and consumer privacy desires and directives in one jurisdiction may not be legally applicable/enforceable in another jurisdiction
 - An entity operating in one jurisdiction uses and discloses health information based on its own policies and procedures, created to meet consent requirements under that jurisdiction
 - When information is disclosed to a different entity in another jurisdiction, the receiving entity applies its own policies and procedures to the received data, which were created to meet consent requirements under the receiving entity's jurisdiction

What is Special about Health and Healthcare (6)

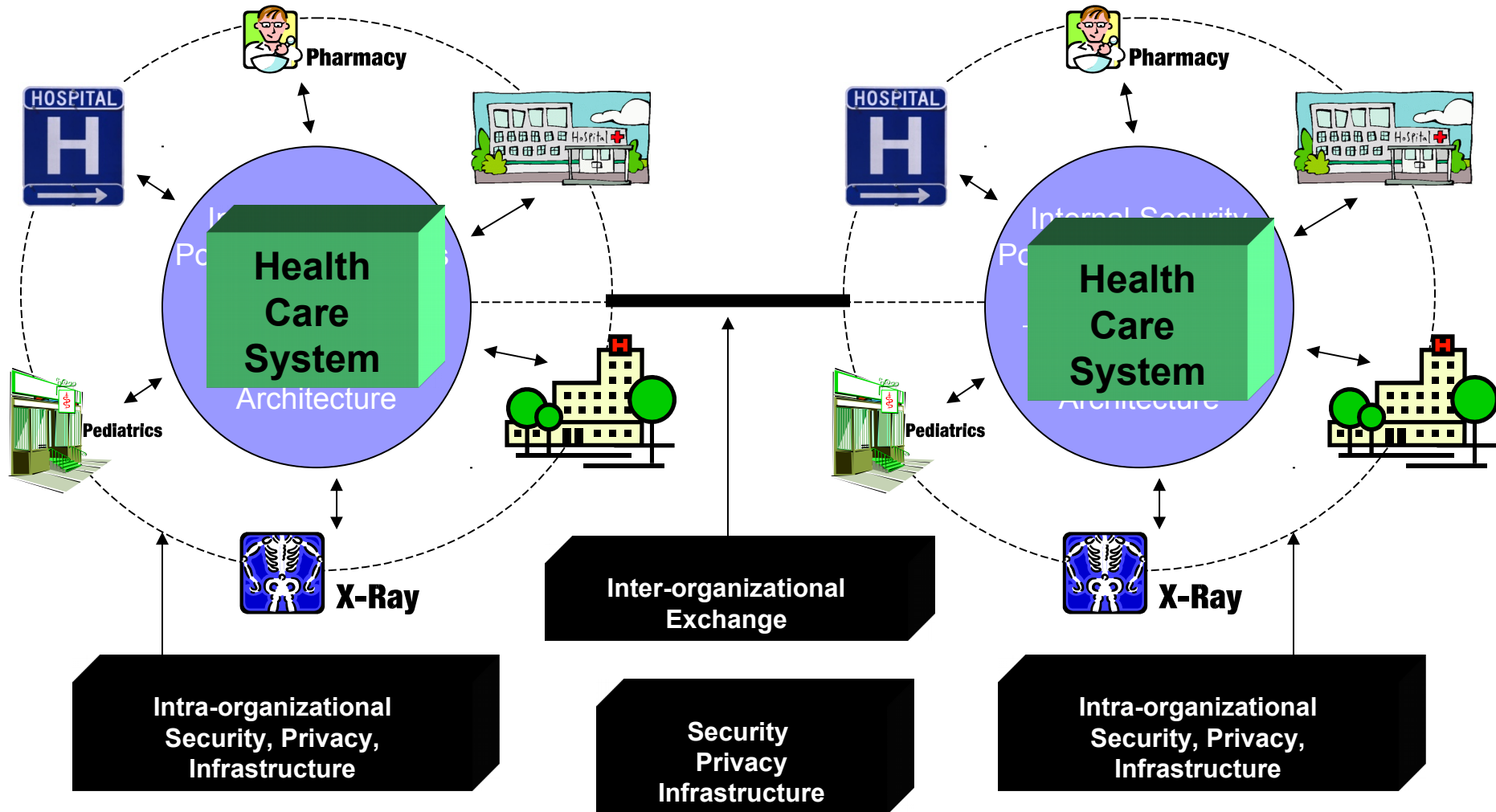
- Cross-validation and verification of conflicting consents
 - What is the most recent/latest consent from a patient?
 - Does that override other consents for specific data, specific purpose?
 - Where can I find the various consents issued by a consumer to perform cross-validation and verification?



What is Special about Health and Healthcare (7)

- Security Requirements
 - Identification, Authentication
 - Various actors and systems
 - Patient, Providers, Payers, Others
 - Authorization, Access Controls
 - Who can collect, access, use, disclose what
 - Audit
 - Account for access, edit, delete, and other actions, by actor
 - Account for security threats
 - Secure data transport, non-repudiation, message encryption
 - Time-stamp

Privacy and Security Interoperability – The Next Challenge





Contact Information

Walter G. Suarez, MD, MPH

President and CEO

Institute for HIPAA/HIT Education and Research

(703) 354-0042

walter.suarez@sga.us.com

An Overview of E-Voting Security Challenges

IDTrust
April 14, 2009

Andrew Regenscheid
Computer Security Division
National Institute of Standards and Technology

Overview

- Background
- Security Challenges in E-voting
 - Strong authentication and Voter privacy
 - Transparency and Auditability
 - Usability and Accessibility
 - Difficulty of making good security decisions
- Research Areas in E-voting

NIST Voting Efforts

- NIST provides technical support to the EAC in the development of the voting guidelines
 - VVSG
 - Technical research items
 - UOCAVA voting
- Topic Areas
 - Security
 - Usability and Accessibility
 - Hardware & software reliability

(Nearly) Conflicting Goals

- Need to **identify and authenticate** voters to ensure only eligible people vote
- Need to protect voter **privacy** to prevent coercion
 - Protect privacy even from insiders
 - Protect voters from themselves (vote selling)
- This is why voting is an interesting crypto problem

I&A for E-voting

- I&A works differently for different systems
- Polling place e-voting
 - I&A performed by officials separately from voting machines
 - Voters receive a token to vote after checking in
 - Authentication information varies
- Internet voting
 - Voting systems authenticate voters
 - Typically, PINs are used

Transparency and Auditing

- Many systems must provide evidence of correct behavior
- It's mostly a matter of:
 - Who can do the auditing?
 - What information do they need?
- Often owners/operators need assurance of correct behavior by equipment
- Auditing can be difficult on voting systems
- The **general public** needs assurance of fair & honest elections

Usability and Accessibility

- These are goals for many systems
- Accessibility is mandated by law
- Usability hampered by:
 - Limited opportunity for training
 - Systems seldom used
- Expectation that any voter can walk up to a voting machine and easily vote without assistance
- These issues **limit acceptable technical solutions** to security challenges.

Decision Making

- Goal is cost-effective, risk-based security
- This is difficult to do with voting
 - There are no risk assessments on voting systems
 - It can be difficult to detect security violations
 - Difficult to monetarily quantify loss

Current Research

- Auditable Voting Systems
- Split-Process Architectures
 - Spread out trust over several pieces of equipment
 - Detect fraud when at least one device functions properly
- End-to-End Voting Systems
 - Cryptographic schemes
 - Voters can verify integrity of their own votes
 - Anyone can verify vote tabulation

Improving U.S. Voting Systems

- NIST activities supporting the Help America Vote Act

NIST

National Institute of
Standards and Technology

Thank you

Security and Social Networking

Barry Leiba
Internet Messaging Technology

Aspects of Computer Security

- ◆ Authentication: Who am I? Prove it.
- ◆ Authorization: What am I allowed to do?
- ◆ Access Control: What can I allow others to do?
- ◆ Privacy: Am I safe from unauthorized viewing?
- ◆ Integrity: Am I safe from undetected changes?
- ◆ Non-repudiability: Can I or others deny what they said or did?

The Social Networking Model

- ◆ Everything is shared
- ◆ You make “friends”
- ◆ All friends are equal
- ◆ Some systems allow categorizing friends
 - ◆ It's not convenient
 - ◆ It's not really part of the model
- ◆ “Friending” is often fairly promiscuous
- ◆ Friends post public communiques to you

Some Problems

- ◆ IRL, not all friends are equal
 - ◆ You don't usually share everything with everyone
 - ◆ Close friends, work friends, ...
- ◆ IRL, it may not be easy to categorize friends
 - ◆ One friend belongs to multiple categories
 - ◆ The categories overlap in odd ways
 - ◆ Category combinations are unworkable

Some Problems

- ◆ IRL, you choose friends more carefully
 - ◆ Face-to-face information is more certain
 - ◆ Face-to-face interaction provides many cues
- ◆ IRL, your friends have more limited access
 - ◆ ...to you
 - ◆ ...to what you have to share
 - ◆ ...to your other friends
 - ◆ ...to what your other friends say

Example

- ◆ Joe has a party IRL
 - ◆ Some of Joe's friends are invited
 - ◆ Some are not
- ◆ The next day, friends post to Joe's "wall"
 - ◆ Thanks for the wonderful party!
 - ◆ What a great time we had!
 - ◆ Check out this pic from the party!
- ◆ Joe's uninvited friends see that too

Example

- ◆ *Is Britney Spears Spam?*
- ◆ Automated filtering in social networking?
- ◆ Much more a value judgment than with email
- ◆ You want to be her “friend”; I don't
- ◆ Is it really her?
- ◆ What are the tradeoffs?

Some Problems

- ◆ Online presence is vulnerable to malware
- ◆ Accepting certain things from false “friends” can be dangerous
- ◆ Once infected, you will infect real “friends”
 - ◆ ...even if they trust you

Example

- ◆ *Social Honeypots: Making Friends With A Spammer Near You*
- ◆ Researchers created MySpace identities
 - ◆ Waited for “friend” requests; stored and rejected
- ◆ 1570 requests, most within 2 months
 - ◆ Click traps, Friend infiltrators, Pornography, Pills
 - ◆ 1245 contained links
 - ◆ 1048 links worked
 - ◆ Only 6 unique clusters

Some Problems

- ◆ “Anonymized” data is aggregated for analysis
- ◆ Aggregated data is vulnerable (AOL problem)
- ◆ Anonymization isn't sufficient

Example

- ◆ *De-anonymizing Social Networks*
- ◆ Researchers looked at Flickr and Twitter
 - ◆ Anonymized network graph of Twitter
 - ◆ Identified network information from Flickr
- ◆ Relatively small overlap between the two
- ◆ Very successful at identifying Twitter users

Authentication & Access Control

- ◆ They like to use other services
 - ◆ Import your address book (from Gmail)
 - ◆ Access photos from Flickr (Yahoo!)
 - ◆ Print your friends' photos (Kodak Gallery)
- ◆ OpenID
 - ◆ Shared authentication service, no access control
- ◆ Oauth
 - ◆ Distributed access control
 - ◆ IETF chartering a working group

Legal Questions...

Is anonymization of data sufficient to protect our privacy?

As we live our lives more publicly, do we give up a legal sense of *expectation of privacy*?

References

- ♦ *Web Searchers' Identities Traced on AOL*; Barbaro, M. and T. Zeller Jr.; New York Times article; 9 August 2006;
<http://www.nytimes.com/2006/08/09/technology/08cnd-aol.html>
- ♦ *Is Britney Spears Spam?*; Zinman, A. & J. Donath; 4th Conference on Email and AntiSpam; August 2007; <http://www.ceas.cc/2007/papers/paper-82.pdf>
- ♦ *Social Honeypots: Making Friends With A Spammer Near You*; Webb, S., J. Caverlee, C. Pu; 5th Conference on Email and AntiSpam; August 2008;
<http://www.ceas.cc/2008/papers/ceas2008-paper-50.pdf>
- ♦ *De-anonymizing Social Networks*; Narayanan, A. and V. Shmatikov; IEEE Symposium on Security and Privacy; May 2009;
http://www.cs.utexas.edu/~shmat/shmat_oak09.pdf

Respect

Excellence

Integrity

Leadership



What is Special About My Application?

for the NIST Idtrust 2009 Symposium

14th April, 2009

For information, please contact:

Bob.Sunday@pwgsc.gc.ca



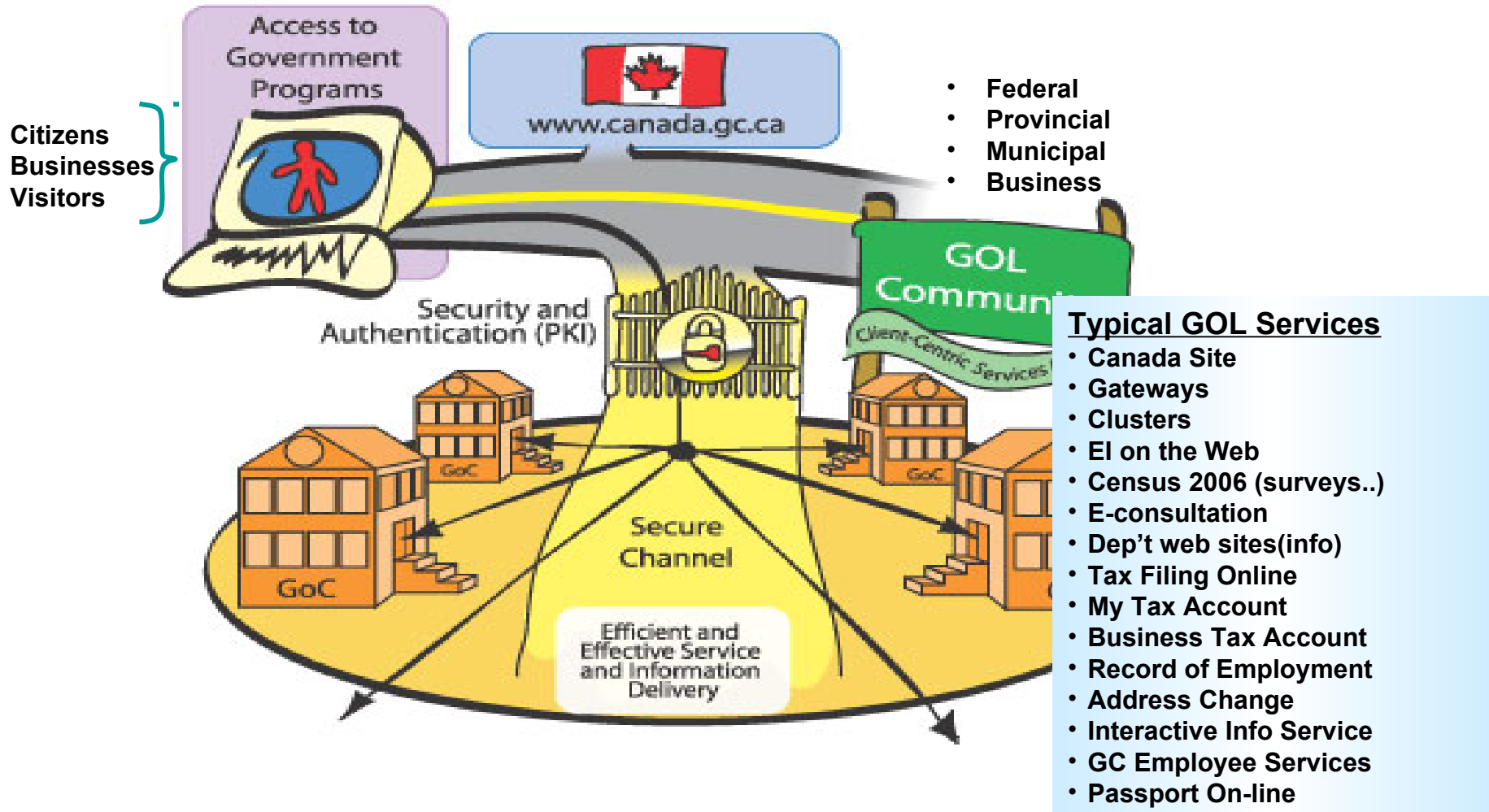
Public Works and
Government Services
Canada

Travaux publics et
Services gouvernementaux
Canada

Canada

Secure Channel: The Enabler for Government On-Line

Secure Channel: Meeting The Challenges of Government On-line



epass Canada (the Common Registration Service)



epass Canada Log In or Register

Already have an epass? [Log In](#)

Your password contains one upper case letter, one lower case letter and one digit.

User ID:

Password:

[Log In](#)

Did you **Forget Your Password?** To change your User ID or Password, or revoke your epass, you must first log in.

Need an epass?

If you do not have an epass and would like one, please click Register.

[Register](#)

You can access all [epass Enabled Services](#) with one epass.

Note: Due to the secure nature of our system, the use of your browser's [Bookmarks/Favorites], [Back] button and [Refresh/Reload] buttons are not supported.

For more information on how your privacy is protected, please refer to our [Privacy Statement](#).

[Exit](#)

Last updated: 2008-12-01

[Top of Page](#)

[Important Notices](#)

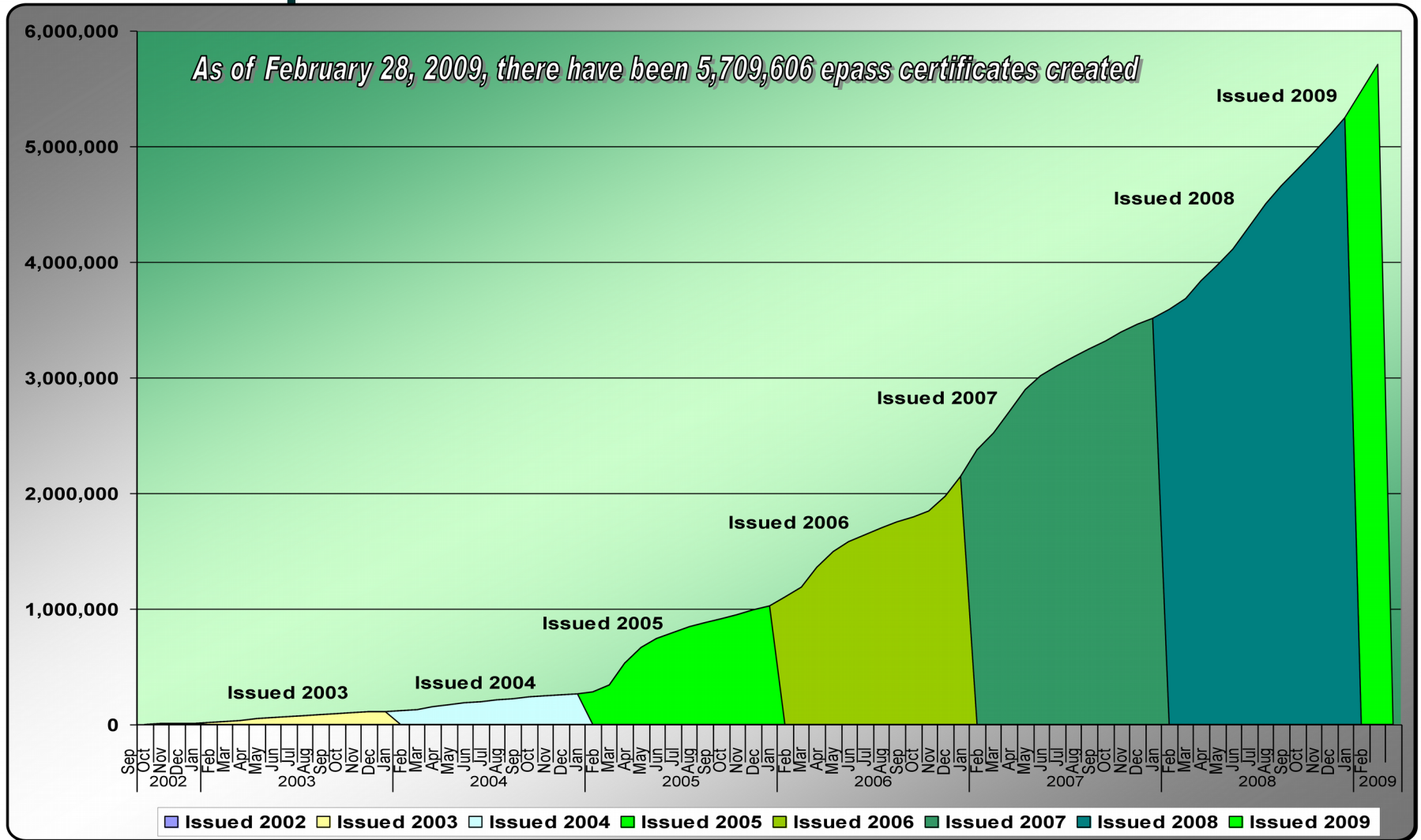
- Single Sign-on
- Two-way encryption (user <--> department)
- Signature Verification
- On-line Registration
- On-line or In-Person ID Proving
- Out-of-Band Secret option
- Time Stamping
- Non-Repudiation support
- User managed

Secure Channel Enabled Applications in Production

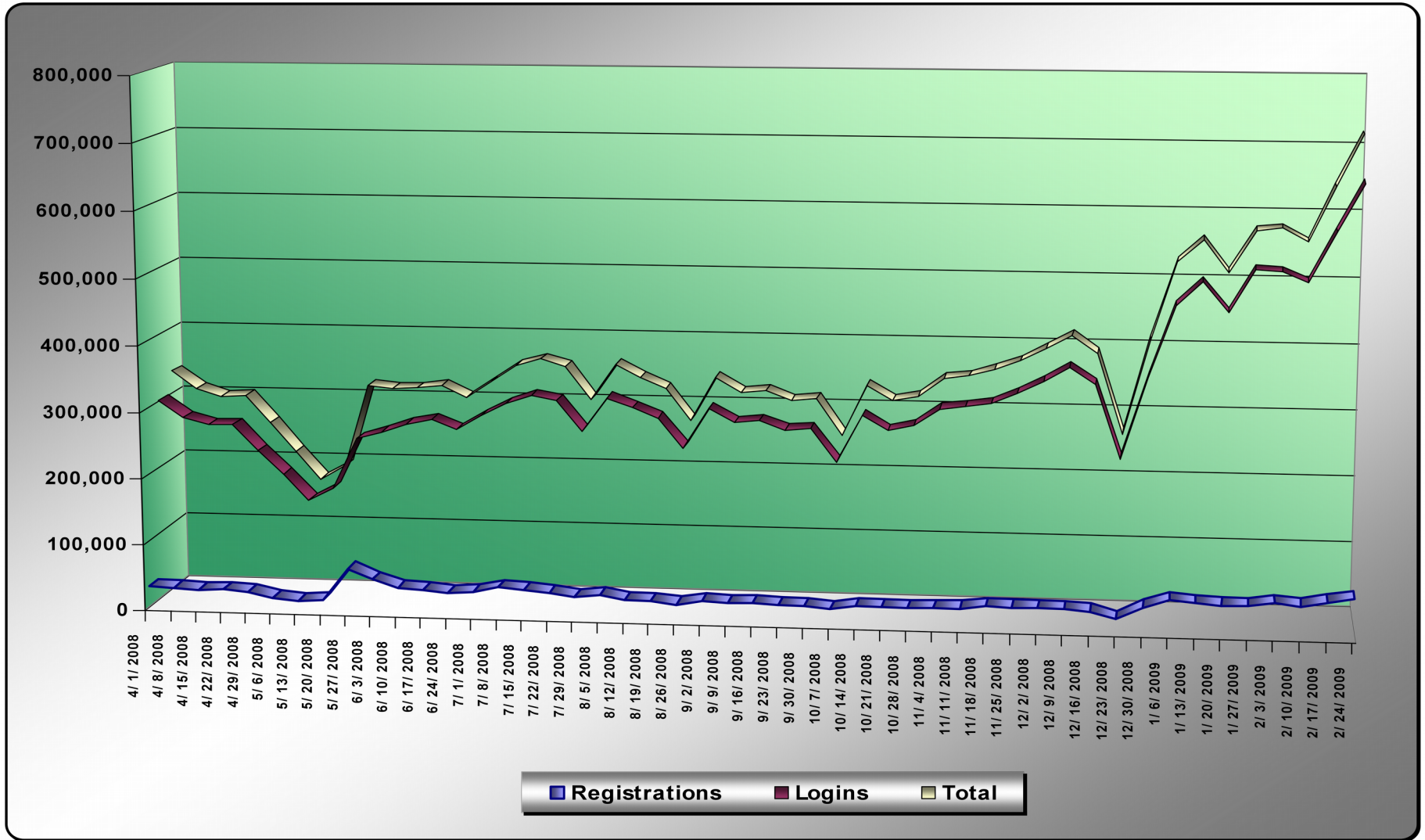
Department or Agency	First Implementation	Number of Programs
Canada Revenue Agency (CRA)	09/01/2002	40
Service Canada	04/01/2003	9
CRTC	08/25/2004	5
Atlantic Canada Opportunities Agency	08/30/2004	3
Health Canada	09/15/2004	1
Veterans' Affairs Canada (VAC)	11/12/2004	2
Téléfilm	12/06/2004	2
Foreign Affairs Canada	01/12/2005	1
Enterprise Cape Breton Corporation	02/01/2005	3
Environment Canada (EC)	03/30/2005	1
Immigration and Refugee Board	04/22/2005	1
Competition Tribunal	06/06/2005	1
Department of National Defence	10/06/2005	1
National Energy Board	10/07/2005	1
Transport Canada (TC)	11/21/2005	4
International Trade (ITCan)	11/30/2005	1
Bank of Canada	03/06/2006	1
Agriculture and Agri-Food Canada	03/31/2006	1
Canadian Nuclear Safety Commission (CNSC)	05/19/2006	1
Canadian International Trade Tribunal (CITT)	05/30/2006	1
Natural Sciences & Engineering Research Council (NSERC)	05/14/2007	1
Public Works and Government Services Canada / Travaux publics et Services gouvernementaux Canada	06/25/2008	1
Citizenship and Immigration Canada	06/25/2008	1
Bank of Montreal (BMO)	12/01/2008	1

83 Programs in 23 Departments

Issued epass Certificates (since Sept 2002)



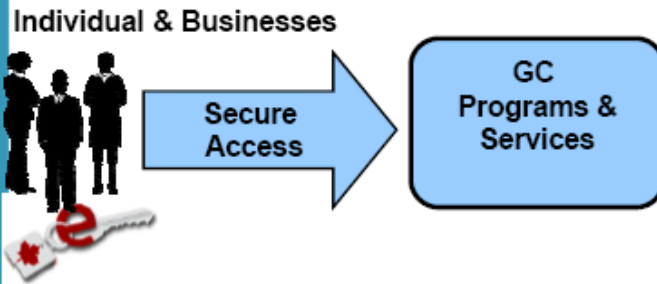
Weekly Logins and Registrations (April 1, 2008 to February 28, 2009)



New Approach for Cyber-Authentication

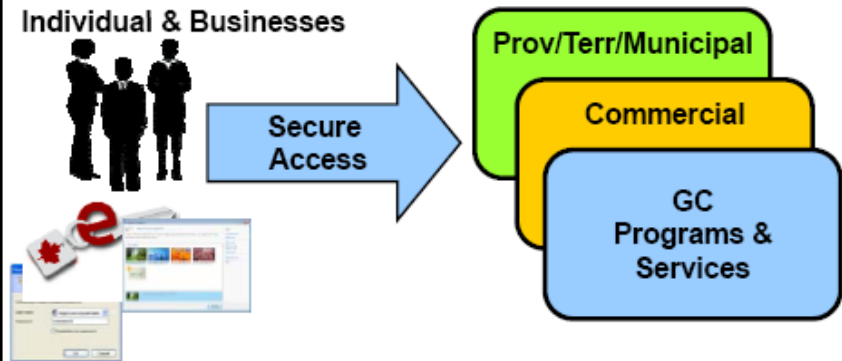
GC Today.... *GC Tomorrow....*

Current Solution



- Single credential
- Single provider
- One level of assurance
- Bundled with other services
- Federal Only

Federated Model



- Technology neutral
- Accepts credentials from other providers (GC, Provinces, Territories, Commercial)
- Multiple providers
- Multiple levels of assurance
- Loosely-coupled Services
- Multi-jurisdictional
- Multi-channel

3



Treasury Board of Canada
Secrétariat

Secrétariat du Conseil du Trésor
du Canada

Canada



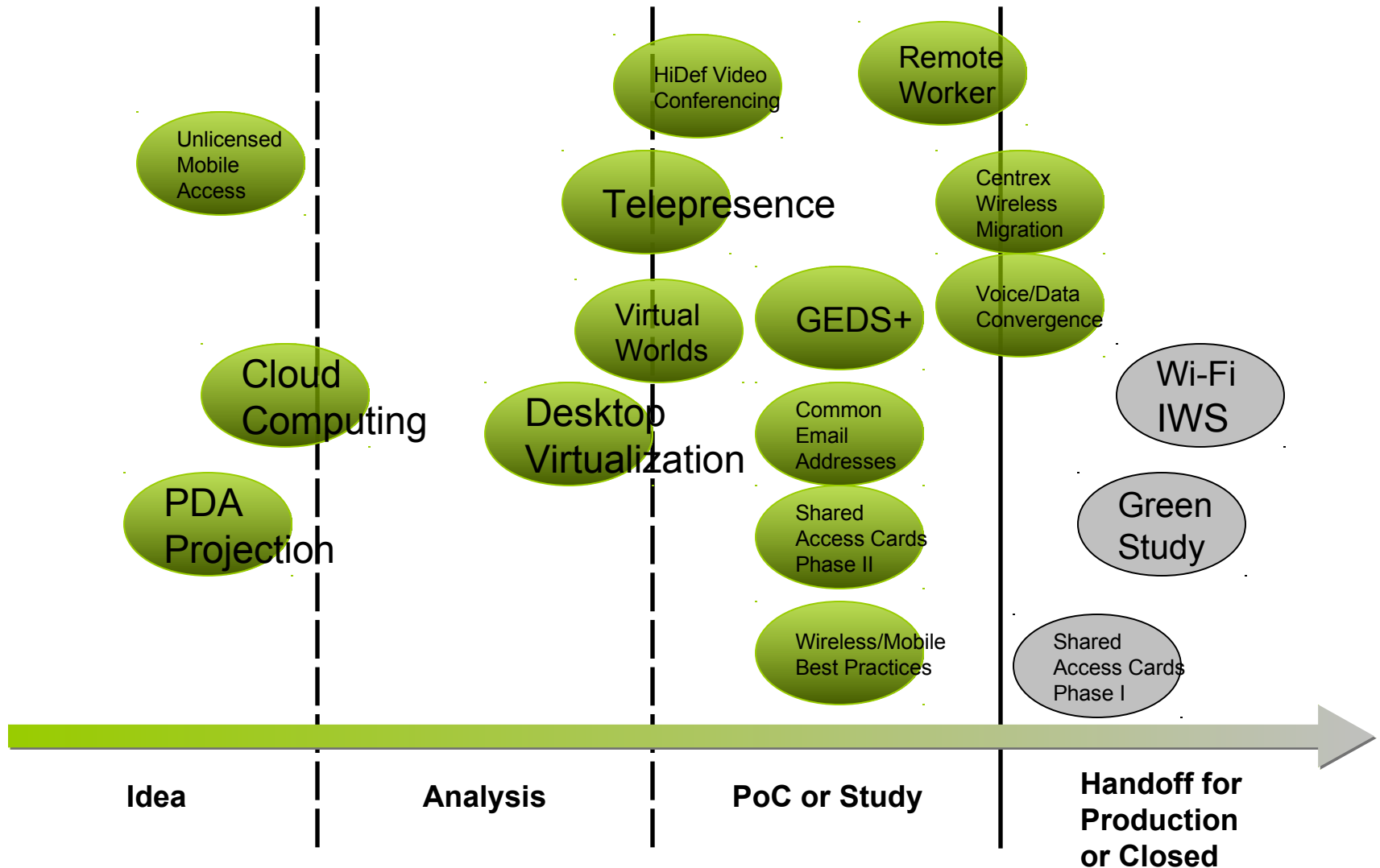
Public Works and
Government Services
Canada

Travaux publics et
Services gouvernementaux
Canada

So why does GC need to change?

- \$\$\$\$
 - Decentralized funding
 - Expense of PKI
 - Custom GC code
- Risk based Assurance Model
- Multi-jurisdiction environment
 - Provincial, municipal
- Changing policy requirements
 - Digital signature
 - Positioning for future identity possibilities

Innovation Activities (Research Agenda)



Some thoughts

1. Stable interfaces over the long term are still necessary
 - We must continue to interoperate
 - Return on investment
 - ♦ Expense of changing
2. Movement of the interface boundary to the human interface will cause new interoperability challenges
 - Loss of choice
 - ♦ Vendor/Provider lock-in
 - Dependence on visual environment
 - ♦ Accessibility?
 - ♦ Language choice?
 - Weakest link in the security chain
 - If we must go here, then do we standardize the people?































Cloud Computing Service Layers

	Services	Description
Application Focused	Services	Services – Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
	Application	Application – Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
	Development	Development – Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as Salesforce
Infrastructure Focused	Platform	Platform – Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
	Storage	Storage – Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
	Hosting	Hosting – Physical data centers such as those run by IBM, HP, NaviSite, etc.

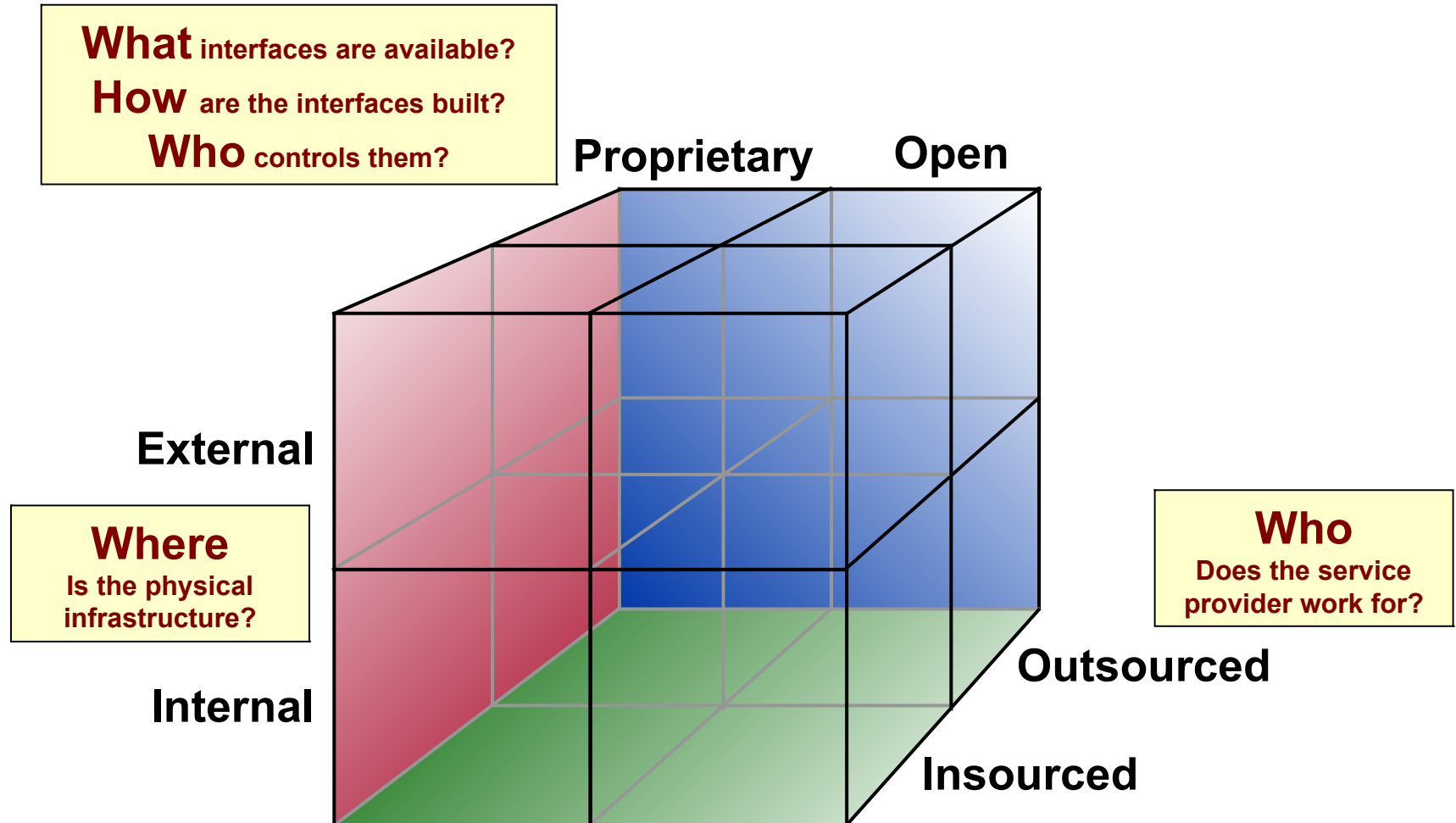
General Characteristics by Service Layer

Boeing Technology | Information Technology

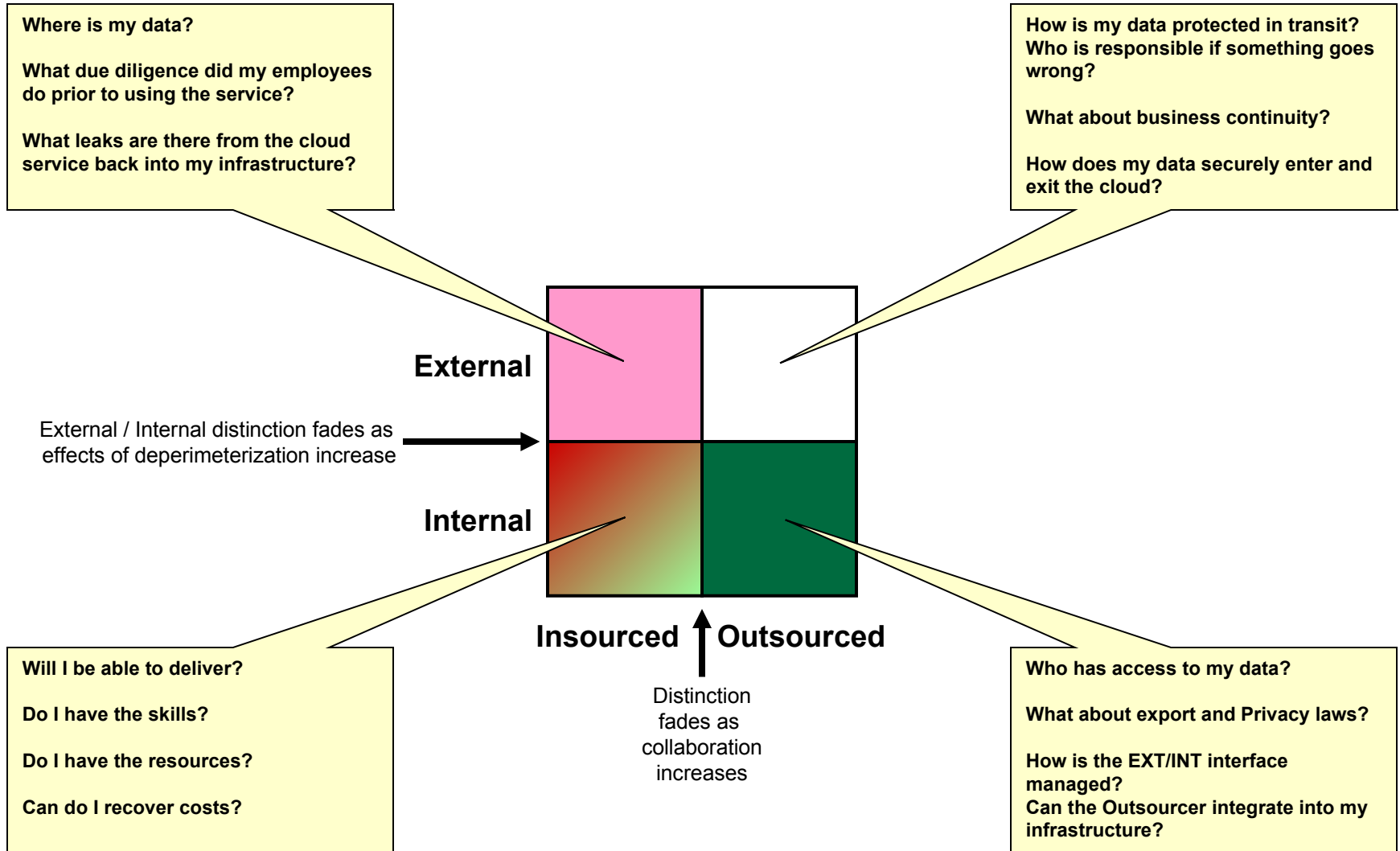
Information Security

Services	Capability Maturity	Information Risk	Relative effectiveness of technical controls	Inter-operability Risk	Difficulty of enterprise integration	
Services						
Application						
Development						
Platform						
Storage						
Hosting						

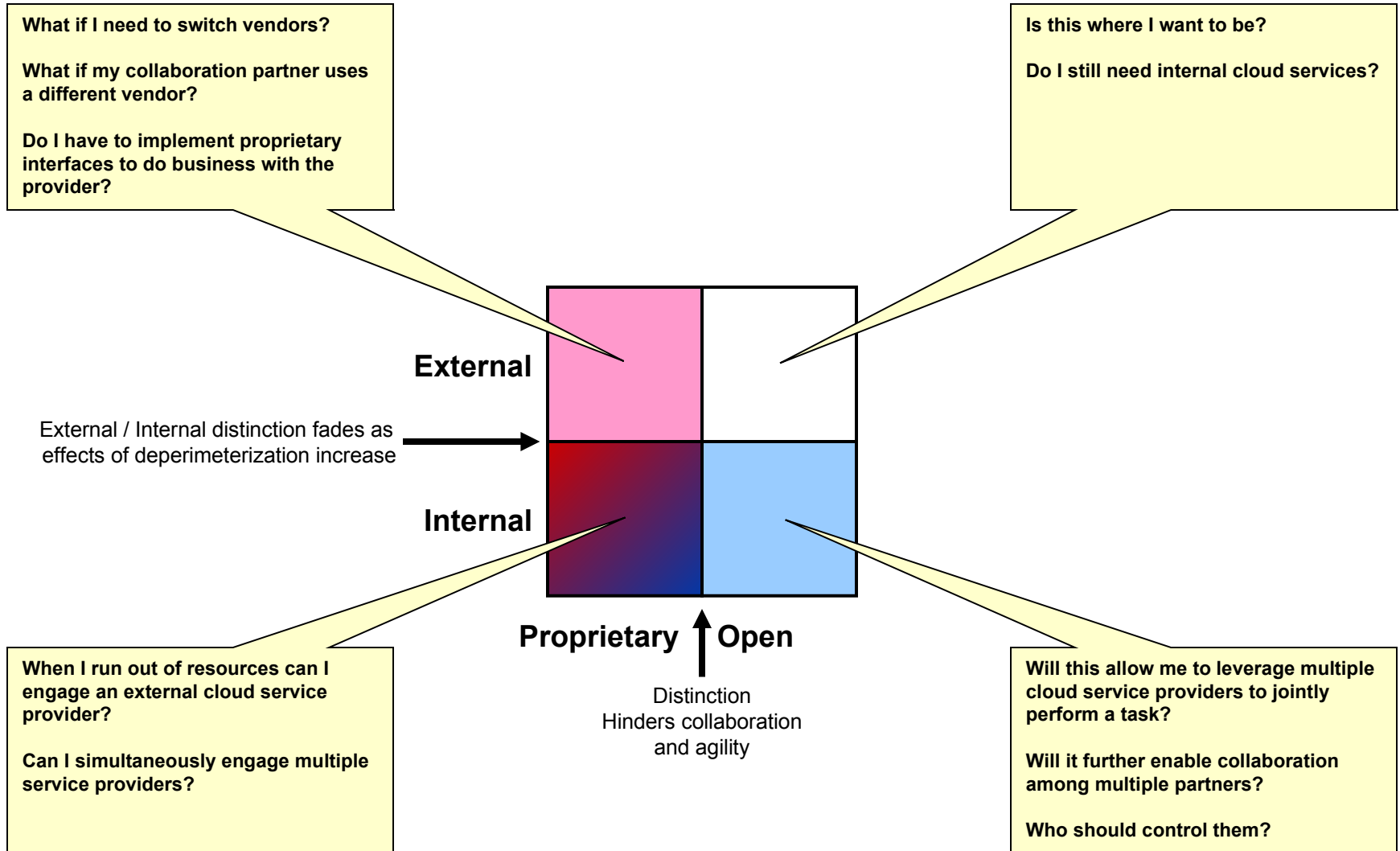
Cloud Property Model



Example Security Questions



Example Interoperability Questions



Identity and Trust in Context

Peter G. Neumann
Principal Scientist
SRI International ComputerSciLab
Menlo Park, CA 94025-3493
Neumann@CSL.sri.com
<http://www.csl.sri.com/neumann>
Telephone 1-650-859-2375
IDes+2 of April 2009, NIST

1

Outline of This Talk

- The overall context
- Global-scale ID management
- Assessable ID/privacy protection
- Health-care as an example
- Attribute-based encryption
- Attribute-based messaging
- Lattice-based cryptography
- References, URLs, search words

2

Identity

Something that identifies a person (or class of persons, or process, or piece of hardware, or other computer-related entity) – but not necessarily uniquely.

3

IDtrust and Trustworthiness

'IDtrust' begs the question: Why trust authentication systems/IDs? Because it's easy? Alternatives are not user-friendly? I'm not paranoid?

Trustworthiness of mechanisms, systems, and people on which you must depend is very important, but difficult to ensure. Risks: errors, ID fraud, spoofing, ... [References]

4

IDEals and IDylls

In the myth of perfect security, our beliefs are often misplaced. Perfect security does not exist. Instead, we tend to have:

- IDEology: Faith-based security
- IDolatry: Worship of physical security rather than systemic and operational security.

5

IDiomatic?

Psychoanalytic terms (IDentity types)

- ID: completely unconscious division of the psyche (users!)
- EGO: organized conscious mediation (administrators!)
- SUPEREGO: partially conscious morality/conscience/guilt/... (privacy advocates!)

6

The Basic IDEa

We need holistic total-system trustworthy identity management. IDEally, IDentities should relate to strong system authentication, fine-grained authorization, nonsubvertible accountability, real-time and post-hoc analysis, remediation, revocation, and more.

7

IDealization

We need to mask underlying complexity to make IDs and ID management more usable – with abstraction, encapsulation, invisible encryption/hash functions, virtualization, sensible interfaces, judicious use of anonymization, and much more.

8

IDIOSYNCRASIES

Characteristic peculiarities must be accommodated. “One size fits all” is not practical, with many special cases: long names, hyphenated names, foreign languages, alternative spellings, non-ASCII characters, ambiguities, false positives/negatives, and much more. Beware of oversimplification!

9

IDIOTS AND IDLENESS

- **IDiot:** Typically, an attribute associated with someone blamed for misusing a dysfunctional human interface or who is dysfunctional.
- **IDleness:** Inaction that may result in serious risks, typified by laziness with respect to security practices.

10

IDEOGRAMS

IDEograms are symbolic but not literal representations, useful for identification (candidate or party icons in elections), CAPTCHAs (for confirmations), authentication. Caveats: dyslexia, prosopagnosia (face-blindness), other disabilities, user unfriendliness, ...

11

GLOBAL-SCALE ID MANAGEMENT

ID management, authentication, authorization, accountability must

- adapt to continual change
- transcend local identities
- transcend centralized control
- transcend untrustworthy systems
- transcend untrustworthy people
- avoid conflicts and ambiguities
- scale to large heterogeneity

12

Roadmap for Global ID Management

Doug Maughan's R&D roadmap for cybersecurity addresses GIDM as one of 11 hard problems, holistically synergistic with the other 10: scalable trustworthiness, metrics, evaluation life-cycles, insider threats, malware, system survivability, situational awareness, provenance, privacy-aware security, usability.

13

Assessable IDentity and Privacy Protection

Dartmouth-I3P-funded joint project:

- MITRE (PI Bruce Bakis) *
 - Cornell University
 - Georgia Tech
 - Purdue University *
 - SRI International
 - University of Illinois Urbana *
- [* => project paper presented here.]

14

Some Health-Care Challenges

Patient and personnel identification, authentication, authorization, accountability; correct up-to-date medical histories; network/system/data security, integrity, privacy; controlled data access for insurance, medication, research, and analysis.

15

Health-Care Risks

- System and information misuse; wrong IDs, privacy violations, malpractice, ... (<http://www.risks.org>).
- Computer-centric doctors may cause patient depersonalization. (See *The Computer Will See You Now*, Anne Armstrong-Cohen, *The New York Times*, March 6, 2009, risks-25.60).

16

Health-Care Risk Avoidance

- Trustworthy systems are essential, but privacy is largely extrinsic. They demand pervasive oversight.
- Well-defined enforceable policies are essential.
- Attribute-based encryption might provide natural mappings between identities and role-based applications.

17

Attribute-Based Encryption (ABE)

ABE (Brent Waters et al.) involves IDs, role-based-like authorization with expressive access controls, practical usability, collusion resistance, simplifies key management, and is holistically well-suited to applications such as health care (21 papers since 2007). Search: `functional encryption Waters`

18

Attribute-Based Messaging (ABM)

- UIUC's ABM (Carl Gunter et al.) uses ABE. The messaging system constructively uses access-control attributes that can be systematically derived and automatically managed (10 recent papers). Search: `attribute messaging Gunter`

19

Lattice-Based Cryptography (LBC)

- LBC (Chris Peikert et al.), based on a problem other than factoring or discrete logs, seems resistant to quantum computing. Uses include strong public-key cryptography and a hash function SWIFFTX with provable properties: a NIST SHA-3 candidate (11 recent papers). Search: `Peikert`

20

Conclusion

- Local and global IDentities need trustworthy systems and networks with authentication, authorization, accountability, and much more. Enterprise architectures, system engineering, sound operational practices, usability, and people tolerance are all vital to reducing risks.

21

CSTB Trustworthiness/ID Reports

- NatlResCouncil, www.nap.edu:
 - * Toward a Safer and More Secure Cyberspace, 2007
 - * IDs Not That Easy: Questions About Nationwide Identity Systems, 2002
 - * Trust in Cyberspace, 1998
 - * Computers at Risk: Safe Computing in the Information Age, 1990

22

PGN IDentity Reference

- PGN, Security and Privacy in the Employment Eligibility Verification System (EEVS) ..., House Ways and Means Committee Subcommittee on Social Security, 7 Jun 2007.

<http://www.csl.sri.com/neumann/house07.pdf>

23

Other Relevant PGN References

- Reflections on System Trustworthiness, Advances in Computing, volume 70, Academic Press, Elsevier, 269–310, 2007
- Principled Assuredly Trustworthy Composable Architectures, 2004: <http://www.CSL.sri.com/neumann/chats4.html>, .pdf, .ps

24

More PGN References

- Holistic Systems, *ACM SIGSOFT Softw.Eng.Notes*, Nov. 2006
<http://www.csl.sri.com/neumann/holistic.pdf>
- Computer-Related Risks, Addison-Wesley, 1995
- www.CSL.sri.com/neumann
- ACM Risks Forum, www.risks.org

25

IDIographic Summary

Identity IDEals, offset by
kIDstuff fIDelity and
epIDemic avIDity slowed by
accIDental antIDotes with
consIDerable fastIDiousness and
indivIDual coincIDences but with
improvIDent backslIDing, result in
self-evIDent nonconfIDence or else
unconsolIDated overconfIDence!

26

Palantir: A Framework for Collaborative Incident Response and Investigation

Himanshu Khurana, Jim Basney, Mehedi Bakht, Mike Freemon, Von Welch, Randy Butler
NCSA, University of Illinois
1205 W. Clark St., Urbana IL 61801, USA
{hkhurana, mbakht2}@illinois.edu, {jbasney, mfreemon, vwelch, rbutler}@ncsa.uiuc.edu

ABSTRACT

Organizations owning cyber-infrastructure assets face large scale distributed attacks on a regular basis. In the face of increasing complexity and frequency of such attacks, we argue that it is insufficient to rely on organizational incident response teams or even trusted coordinating response teams. Instead, there is need to develop a framework that enables responders to establish trust and achieve an effective collaborative response and investigation process across multiple organizations and legal entities to track the adversary, eliminate the threat and pursue prosecution of the perpetrators. In this work we develop such a framework for effective collaboration. Our approach is motivated by our experiences in dealing with a large-scale distributed attack that took place in 2004 known as Incident 216. Based on our approach we present the *Palantir* system that comprises conceptual and technological capabilities to adequately respond to such attacks. To the best of our knowledge this is the first work proposing a system model and implementation for a collaborative multi-site incident response and investigation effort.

Categories and Subject Descriptors

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

General Terms

Security

Keywords

incident response, digital investigation, multi-site collaboration

1. INTRODUCTION

Increasing awareness of cyber-security incidents in terms of their prevalence, impact on productivity and financial

loss have motivated organizations to ramp-up their security stance and better prepare for dealing with such incidents, for example, by establishing Computer Security Incident Response Teams (CSIRTs) [8] and setting up digital investigation procedures [26]. Such capabilities allow for incident response that results in full recovery and patching to prevent relapse as well as for working with law enforcement when appropriate to pursue criminal prosecution. While measuring the success of these capabilities is not easy, anecdotal evidence and an increasing deployment rate indicates their effectiveness. However, a new breed of large-scale distributed cyber-attacks is emerging that is characterized by a set of motivated, dedicated and resourceful adversaries that attack a number of hosts, sites and organizations that span multiple countries. In these attacks adversarial motivations range from demonstrating hacking skills to criminal intent for financial gain, and specific targets range from sensitive data theft and public image maligning to network disruptions (e.g., via denial-of-service). These attacks can be overwhelming to individual organizations responding on their own.

A prime example of such a large-scale distributed attack and our motivating use case is a series of cyber attacks known as *Incident 216* [32]. This incident took place in 2004 and involved an attacker from a foreign country who compromised the integrity of a large number of hosts in U.S. government, higher education, and commercial institutions and similar institutions abroad. The incident response and investigation process for Incident 216 brought to fore new requirements and challenges for dealing with large-scale multi-site attacks in a collaborative manner. That is, there is a need to develop a framework for effective collaboration on incident response and investigation tasks by sharing information and resources.

Establishing trust is a major challenge for these collaborations. First, the affected organizations are chosen by the attacker(s), rather than the organizations themselves, so there may be no existing relationships in place between the organizations. Second, since the collaborations would typically need to take place only after an incident occurs, involve many organizations and last for the duration of the response/investigation, they need to be short term and dynamic in nature. Third, the collaborations need to deal with data and information that is sensitive and private in nature. This includes 1) sharing of logs across institutional boundaries with user information in them that faces issues of security and privacy, 2) interaction with law enforcement and 3) interaction with the media.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust '09, April 14-16, 2009, Gaithersburg, MD
Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

These aspects of the collaboration lead to several challenges that must be addressed when designing a collaboration framework. First, the framework must provide a means for managing the tasks and processes for response and investigation undertaken by the CSIRTs of the collaborating organizations; i.e., determine who should do what and when. Second, in order to manage the tasks the organizations must place trust in each other and provide a means to share information and resources. Third, the framework must provide trustworthy information and data management with effective access control given the sensitive nature of the collaborations.

In this work, we review lessons learned from Incident 216 and propose an effective collaboration framework, that comprises a system model as well as a system design and prototype implementation that allows multiple organizations and legal entities to actively collaborate for investigating and responding to cyber-attacks. While the proposed response and investigation system is distributed in nature, it is centrally managed by a trusted entity, which we call an Independent Center for Incident Management (ICIM). The system model for the response and investigation process defines the roles, responsibilities and processes undertaken by multiple organizations (including law enforcement) to achieve full recovery and prosecution. The system design carefully addresses security and privacy of the data (e.g., security and network logs) and messages (e.g., emails, instant messages, web boards) exchanged across organizations during the response and investigation process. The security architecture provides identity-based and role-based authorization to facilitate sharing and collaboration according to organizational policies and trust relationships. The prototype system implements roles and processes for responding to and investigating an incident, incorporates tools for the collaborative response and digital investigation process, and provides adequate security and privacy.

Our approach builds on several well-known principles for effective collaboration. For trust establishment we adopt a mutual incentives based approach where organizations participate so they can learn more information and can get access to additional resources in order to respond to and recover from the attacks in their organization. Furthermore, we use a collaborative access policy enforcement approach so that organizations providing leadership in the response process can collectively define access policies. For managing tasks and processes we focus on identifying specific tasks that warrant collaboration and integrate them in a well-defined process workflow for each organization. Finally, for managing data and information we use role based access control with the least privilege principle in mind. We use these principles to design a framework that addresses this important problem of large-scale cyber-attacks.

Dealing with multi-site attacks has long been an important issue for the security community. For example, Computer Emergency Response Teams (CERTs) and Information Sharing and Analysis Centers (ISACs) have been setup around the world for vulnerability and exploit tracking as well as facilitating coordination between CSIRTs. We believe that institutions like CERTs and ISACs could potentially serve as ICIMs in our system model. By doing so they would extend their current capabilities to support more effective multi-lateral *collaboration* between the sites, providing significantly improved incident response and investiga-

tion.

Our work is the first to develop a framework for supporting multi-site collaborative digital investigation and incident response. We integrate the two areas of prior work, namely, digital investigation and incident response, by developing models for Roles and Responsibilities as well as Processes that identify their interaction. Furthermore, we propose to extend the scope of CERTs/ISACs to support effective *collaborations* involving multiple organizations by additionally becoming ICIMs.

The rest of this paper is organized as follows. In the next Section we present lessons learned from Incident 216. In Section 3 we discuss the requirements, challenges and approach. In Sections 4 and 5 we specify the system model. In Section 6 we discuss the security architecture. In Section 7 we discuss the challenge of trust establishment. In Section 8 we describe the prototype implementation, and we provide an evaluation of our approach in Section 9. In Section 10 we discuss related work and we conclude in Section 11.

2. INCIDENT 216: LESSONS LEARNED

The scenario motivating our work is an attack by an individual or group against hosts, sites, and organizations across multiple countries. A prime example and our motivating use case is a series of cyber attacks known as *Incident 216*. This incident took place in 2004 and involved an attacker from a foreign country who compromised the integrity of a large number of hosts in U.S. government, higher education, and commercial institutions and similar institutions abroad. While the ultimate motivations of the attacker remain unknown, he seemed to be primarily interested in building this network of compromised hosts for his own personal interests.

The attacker behind Incident 216 used a well-organized process for compromising a large number of hosts and then harvesting user passwords to continue to expand his set of compromised hosts. The attacker initially compromised some number of hosts using known exploits. He then installed trojan secure shell (SSH) clients on these systems that harvested host, username and password tuples as users used the trojan SSH clients to logon to other systems. The attacker then used those stolen credentials to logon to those systems and then gained administrative privileges via known exploits for privilege escalation. Once administrative privileges were gained, the attacker would then install a rootkit to hide himself and trojan the SSH clients to use the new system to gather further account information to repeat the process and grow the base of compromised systems. As discussed by [32], he used the SSH “known hosts” file to find new attack targets.

Besides the fact that the attacker’s collection of compromised systems was spread across multiple domains, the attacker also had supporting infrastructure that was also spread over multiple domains. Figure 1 shows these supporting systems, which included:

- **A Password Collector.** Every time a trojan SSH client captured a hostname, username and password tuple, it sent this information over the network to the Password Collector host. The Password Collector host was one of the compromised hosts where the attacker installed a service to collect and record these tuples for latter use.
- **A Dynamic DNS Service.** The trojan SSH clients

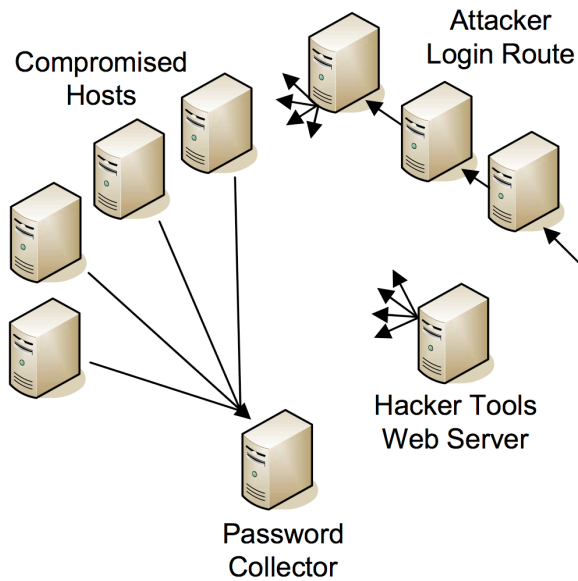


Figure 1: Topology of Incident 216 Attacker Network

used a statically configured hostname to address their network traffic with the captured tuples. This hostname was managed by the attacker through a public dynamic DNS site that allowed him to manage the mapping of the hostname to an IP address anonymously via a web form. This allowed him to move the Password Collector several times during the investigation when he felt it was potentially discovered and being monitored.

- **Hacker Tools Repository.** On one of the hosts the attacker compromised, he installed a set of exploits that he used for privilege escalation. These tools were made available via a web server already installed on the host. After gaining access to a new host, he would download these tools and use them to gain privileged access.
- **Login Route.** Instead of logging in directly from his local system to compromised systems, the attacker always went through a series of distributed intermediate systems. Presumably this was done to make the task of tracking a session back to the attacker difficult.

Investigation of Incident 216 was a difficult task because it required data acquisition across a wide range of distributed systems. Within a week of the initial discovery of the attacker, the investigation spanned a dozen sites. Eventually, the attacks spanned tens of sites in multiple countries. Many of the system administrators at the various sites were willing and even eager to help in the investigation, but often lacked the skills or time to assist, even with just understanding the events at their own site.

The result was a highly manual investigation process with the lead investigators walking sites through data gathering on their local systems, and then collecting, managing and analyzing this data. Communication between the various investigators was ad hoc, with a combination of telephone and email. At one point in the investigation it became clear

that the intruder was monitoring the email of a security administrator, motivating the use of email encryption, which was cumbersome for group messaging.

NCSA staff worked side-by-side with FBI investigators to investigate and solve these attacks. One of the hardest challenges investigators faced during the investigation was the lack of knowledge in the field to identify the attacks at each of the attacked sites and hosts, without any existing coordination between sites that had been attacked. Further complicating this was a void of automated analysis of the security log information that was available, leaving the investigation up to few individuals who analyzed all of the data by hand. The NCSA investigation team committed over 3000 hours in the pursuit of this investigation. In addition to the forensic investigation from security log analysis, considerable effort was undertaken by legal teams in multiple countries to identify the perpetrator and build sufficient evidence against the perpetrator to hold up in court. This aspect of the investigation also faced hurdles in effective collaboration between prosecutors in multiple countries as well as collaboration among law enforcement personnel and system administrators. The total duration of the investigation lasted over nine months with extensive delays caused by the repeated time-consuming tasks of establishing trust between the attacked sites as well as in dealing with the complexity of the attack and the tasks required for both incident response and forensic investigation.

3. REQUIREMENTS, CHALLENGES AND APPROACH

In this section, we outline the challenges that need to be overcome and the requirements that need to be met for effective collaborative response and investigation of large scale distributed attacks. We then outline our approach, which is then detailed over the next three sections.

Requirements and Challenges. In dealing with large-scale attacks with Incident 216 being an example, the incident response and investigation process faces three kinds of challenges.

First, it is hard to establish adequate levels of trust between the involved institutions and personnel. Institutions are reluctant to share information and communicate over such matters while effective response to such attacks requires them to share information, data (e.g., logs) and communicate regularly. The core issues behind the reluctance are security, privacy and financial concerns. For example, logs contain user data that needs to be protected by law, leakage of information to media and competitors can harm the institution's image and lead to financial losses, leakage of information to the adversaries can worsen the ongoing attacks causing further delays in recovery, and investment of personnel time towards regular communication without a clear view of benefits can be perceived as a waste of resources.

Second, even after establishing adequate levels of trust, managing all the tasks and processes in the response and investigation processes is hard. There are a myriad of tasks and activities that need to be executed and managed. This includes, for example, detecting the attack, evidence gathering and storage, forensics and discovery of the attack, restoring services, eradicating vulnerabilities and flaws, sharing data and logs, collaborative decision making, information sharing and analysis, and legal prosecution. Typically, such

a complex set of tasks and activities are organized into intuitive phases such as preparation, analysis, recovery, etc. However, in large-scale attacks different institutions can be in different phases and, furthermore, depending on the attack sequence and evidence discovery, institutions can have multiple phases active. Management of tasks and activities is further complicated by the duration of the response and investigation process, which lasted several months in the case of Incident 216. Such long durations make ad-hoc approaches insufficient.

Third, at the core of the response and investigation process is analysis of the digital system that includes logs and alerts gathered by various system components such as IDSs, server logs, and network logs. These logs can be large in size (100s of MBs or several GBs per day is not uncommon) and have varying formats across different organizations. Consequently, the tools needed to analyze the digital systems as well as personnel skills required to do so are not always available with all organizations that are part of a large-scale attack. Furthermore, in a collaborative response and investigation effort all of this data will need to be managed for the duration of the effort.

Approach. In this work we take a comprehensive approach of defining a system model, specifying the security architecture and describing the system implementation to address all of these requirements. The proposed system model comprises two components: 1) a *Roles and Responsibilities Model* that defines the entities involved in the response and investigation, their responsibilities and their interactions and 2) a *Process Model* that defines the various phases of the response and investigation process as well as the execution of responsibilities in these phases. Combining together these components will ensure that the response and investigation team members will be able to effectively manage the required tasks. In particular, the system model effectively integrates the technical incident response and the legal investigation and prosecution process in a multi-site collaborative manner. The following risks are minimized by this system model: missed or unassigned responsibilities, overlapping responsibilities, unclear reporting functions in a site as well as in the collaborative effort, inability to track global progress and ineffective management of tasks and phases between a site and the collaborative effort.

At the core of the system implementation is a collaborative workspace hosted by the ICIM that is accessible by all team members for managing and analyzing data (such as logs) and communications. While it is possible to implement this workspace in a distributed manner (e.g., using peer-to-peer systems) we chose a more centralized approach based on our model of central management and also to be able to provide better security. In doing so we assume the risks of a single point of failure but benefit from greater security and management assurances. The workspace is equipped with a default set of tools specifically geared towards addressing the above requirements. This includes tools for secure email, instant and web messaging, log and data anonymization, data and evidence storage, and data and log analysis and forensics. For broad adoption we have composed the workspace using open-source tools.

The proposed security architecture is designed to address a large number of threats from both passive and active adversaries. We enumerate these threats in Section 6. Threats from active adversaries are an important concern as we are

dealing with active adversaries who specifically attack communications between administrators to disrupt the response process (as observed in Incident 216). An analysis of such threats led to the design of the security architecture that includes strong two-factor authentication, Role Based Access Control authorization, and a secured network perimeter around the servers implementing the workspace. In analyzing the interplay between implementing a flexible workspace and meeting the security requirements, we chose a centralized workspace environment for simplicity but we believe that a distributed workspace implementation is also feasible though perhaps with higher costs.

Collectively, the workspace along with its default set of tools and the security architecture address the remaining requirements. The presence of such a secured workspace with a plethora of useful tools will make it significantly easier for organizations to establish trust and collaborate on the investigation and response process by committing resources and personnel. Knowing that their data is well protected and can be anonymized, if needed, will encourage them to share data and logs. The workspace also allows the collaborative process to be managed for a long duration, if needed. Lastly, the specific tools in the workspace allow for effective data management and analysis.

4. ROLES AND RESPONSIBILITIES

At the core of any collaborative multi-site response to a large-scale attack is a dedicated team of personnel staffed by the sites and by law enforcement. In this section we identify roles played by these personnel and the responsibilities associated with each role. To ensure that these roles and responsibilities are comprehensive but not significantly overlapping we use the following approach. First, we distinguish between *site roles* and *collaboration roles*. While the same individual may be assigned to both site and collaboration roles, distinguishing the roles allows for contextualization of responsibilities (i.e., site versus collaborative) and supports multiple reporting hierarchies to allow for effective team management. Second, we identify roles that cover technical, managerial, public relations and legal responsibilities. These broad set of roles and responsibilities allow for the specification of comprehensive policies and procedures in dealing with large scale attacks effectively. Specifically, the roles in our proposed model can be divided into the following five categories: 1) Site Technical Roles, 2) Collaboration Technical Roles, 3) Site Legal Roles, 4) Law Enforcement Roles and 5) Other Roles. Third, we place the responsibilities of each role in the context of the response and investigation process, as described in Section 5. Next we describe each role and its associated responsibilities.

The **Site Technical Roles** are responsible for local investigative activities at the site. The **Site Lead** is the person who leads the investigation in a particular site. He/she is also the point of contact for that site in the collaborative investigation process. The **Site Incident Investigator** assists the *Site Lead* with the local investigation, as well as containment, eradication, and recovery activities. The **Site Digital Forensics Specialist** collects, extracts and stores digital evidence locally based on the investigation strategy determined by the *Site Lead*. This role requires expertise with digital forensic tools and adequate training/knowledge to follow the right procedures so that collection and handling of the evidence meets all the legal requirements. The **Secu-**

rity/System Administrator is in charge of maintaining the site Information Technology (IT) system. He/she issues necessary authorizations for evidence collection and investigation. The **Security/System Architect** assists the investigation by sharing his/her knowledge of the IT system and the security design of the system.

The **Collaboration Technical Roles** are responsible for managing and supporting the collaboration. The **Collaboration Incident Lead** leads the investigation into the large-scale attack and also acts as a moderator/coordinator for the entire collaborative investigation process. Typically an experienced investigator in the ICIM is assigned to this role. In the CSIRT model, this is the “incident coordinator” for the designated lead CSIRT [21]. The **Collaboration Investigator** helps the *Collaboration Incident Lead* in investigating the incident(s). This role will be populated by investigators from the sites as well as from the ICIM. The **Collaboration Digital Forensics Analyst** is responsible for extracting relevant data from the evidence collected from individual sites. He/she uses different tools available for the collaborative investigation to perform cross-site analysis and construct a global timeline of the events. This role may also be populated by investigators from the sites as well as from the ICIM. The **Collaboration Workspace Administrator** is the person responsible for maintaining the collaborative environment, which supports exchange of data and messages between sites for the response and investigation process. Since the workspace is hosted by the ICIM, this role should typically be assigned to an administrator from that ICIM.

The **Site Legal Roles** are filled by lawyers, law enforcement, and security personnel local to the site. The **Site Legal Adviser** is a law practitioner associated with a particular site and responsible for advising the *Site Lead* on legal matters. This includes advice on legal and regulatory constraints on what action can be taken, reputation protection and publication relation issues, when/if to advise partners, customers and investors, etc [30]. The legal adviser also plays a crucial role in formulating and checking organizational policies to ensure that there is provision for using forensic tools to collect necessary evidence. The **Site Liaison with Law Enforcement** initiates contact with the appropriate law enforcement agency when decided by the *Site Executive*. He/she acts as the point of contact for all reporting and communication between the site incident response team and law enforcement.

The **Law Enforcement Roles** are filled by government personnel. The **Legal Prosecutor** determines when and how the litigation process should proceed. He/she advises the *Site Lead* and/or the *Collaborative Incident Lead* about what legal recourse may be taken against the perpetrator(s) and the appropriate actions to take for building a strong legal case. Finally, when the investigation is successfully over, it is the *Legal Prosecutor* who takes charge and takes appropriate legal steps for prosecution of the perpetrator(s). The **Legal Investigator** is a member of law enforcement who conducts the investigation with the goal of prosecution. This role exists for both a site and the collaboration. In a collaborative environment, a *Legal Investigator* might have to coordinate the investigation with *Legal Investigator(s)* belonging to other agencies and other jurisdictions.

Finally, the following two roles are also crucial in the investigation process. The **Site Executive** is the person

having overall administrative or supervisory authority of a particular site. The *Site Lead* must keep the *Site Executive* briefed on the investigation process and follow the *Site Executive’s* direction. The **Media Liason** performs the important job of interacting with the media and briefing them about progress of the incident response and the investigation. Utmost care needs to be taken to ensure that no sensitive information gets revealed that might be against the interest of the affected site(s) or might hamper the investigation process.

5. PROCESS MODEL

In this section, we propose a process for the multi-site collaborative incident response and investigation approach advocated in this paper. We describe in detail a four-phase model that represents the process that each site goes through locally for incident response and investigation (which leverages the Incident Response Life Cycle presented in [17]), a four-phase model that represents the collaborative process and the interactions between the site and collaborative processes. These phases are illustrated in Figure 2. The collaborative process is assumed to be executed at the ICIM.

Before going to the description of each phase, it needs to be mentioned that the division of the response and investigation process into phases is not a rigid one. If the process enters a particular phase, it does not mean that only activities that are part of that phase are permitted at that point. Rather, the implication here is that at least some part of the process has progressed up to that specific phase but there is every possibility of revisiting a step belonging to an earlier phase if the need arises. Furthermore, different sites might be in different phases as compared to the collaboration depending on the progress of the investigation.

Preparation. The primary goal of the preparation phase is to develop the capability of handling incidents based on risk assessment and lessons learned from prior experience. In a given site the *Site Executive* leads the effort by establishing an Incident Response Team. Regular training of all concerned individuals is arranged to keep pace with the latest security threats and security tools. The site *System Administrator* also plays an important role in the preparation phase by acquiring tools and resources necessary for incident response and effective investigation. Intrusion detection systems (IDSs), centralized logging, and forensic software are some examples of software tools that are deployed for detection of an incident and also for evidence gathering in subsequent phases as part of forensic readiness. Detailed policy and procedure documents are formulated that specify who should be contacted inside and outside the organization when an incident occurs. They also contain information about how that contact can be made and how much information can be shared especially with outside parties; e.g., law enforcement and other incident response teams. Taking steps to prevent incidents from occurring in the first place is also an important part of the preparation phase. *System Administrators* follow a set of recommendend practices (e.g., those given in [17]) to ensure the security of network, systems and applications that includes patch management, malicious code prevention, training to increase user awareness, host security, etc. The site *Security/System Architect* prepares proper documentation about the site’s network/system designs to aide incident responders.

To prepare for the legal aspects of incident response and

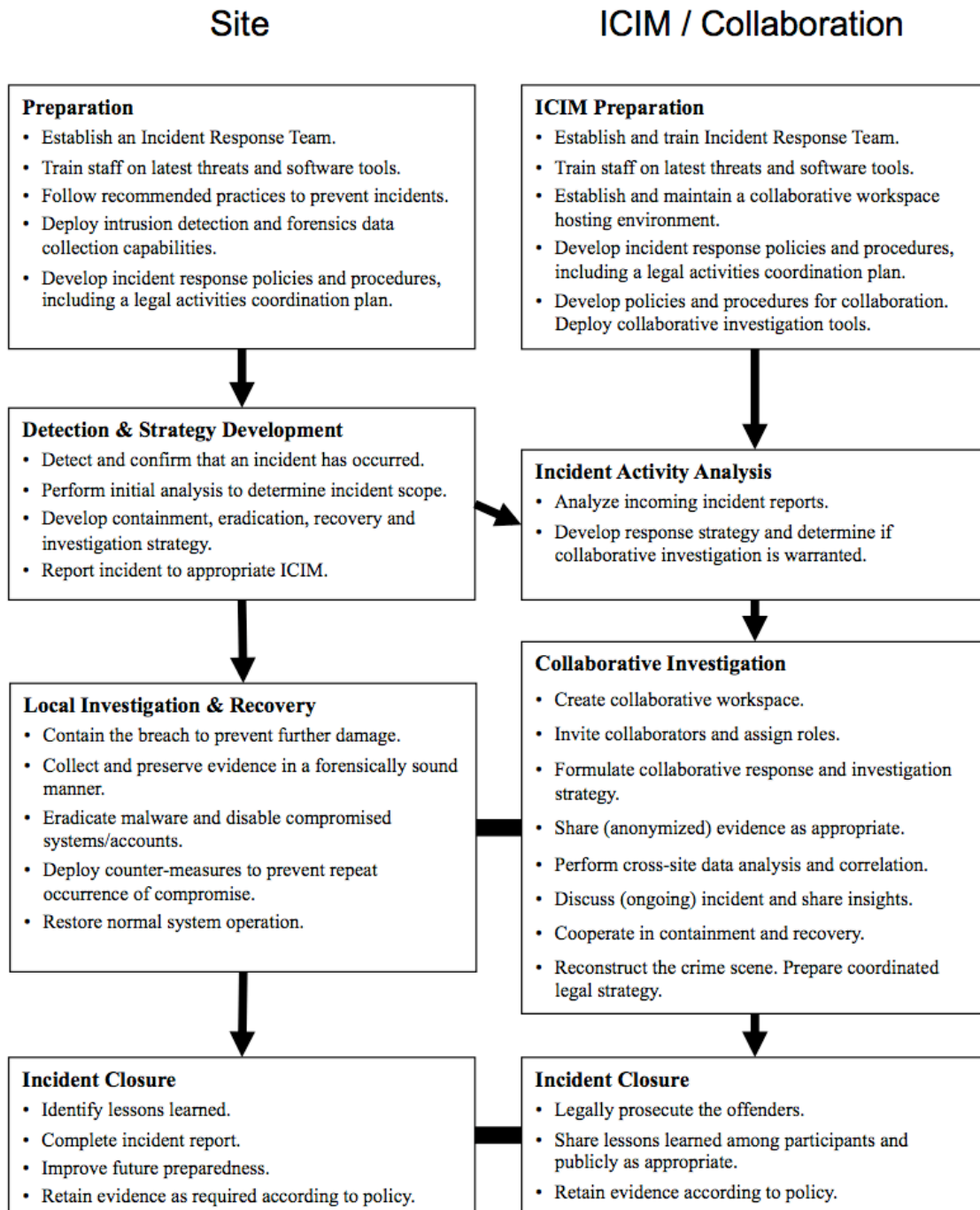


Figure 2: Process Model

investigation the *Site Executive* devises a legal activities coordination plan in consultation with the *Site Legal Adviser*. This plan provides the *Site Liason with Law Enforcement* basic guidance in coordinating the activities of the local Incident Response Team with that of law enforcement agencies.

Detection & Strategy Development. The main focus of this phase is to accurately detect and confirm that

an incident has indeed occurred. Installation of Intrusion Detection System (IDS), antivirus software and other monitoring mechanisms in the preparation phase helps the *System Administrator* identify signs that an incident may have occurred or may be occurring. After getting confirmation about the detection of the incident, an investigation team comprising of a *Site Lead* and one or more *Site Incident Investigators* is created. The *Site Lead* carries out an initial

analysis to determine the category, scope and magnitude of the incident as it is vital in choosing the next steps of the response process.

A strategy regarding containment, eradication, recovery and investigation is developed at this phase by the Site Lead. The *Security/System Architect* of the site and the *Site Legal Adviser* play important roles in formulating this strategy by sharing their knowledge about the technical and legal factors respectively. The *Site Lead* then informs the ICIM about the incident. Depending on the perceived scale and scope of the attack the ICIM personnel are invited to play an active role in developing the strategy.

Local Investigation & Recovery. After validating an incident, containing the scope and impact of the attack to minimal level becomes a major concern for the *Site Lead*. Actions regarding containment may include shutting down system(s), segregating a compromised component from the rest of the network, suspension of accounts that are suspected to be compromised, etc. At the same time, the *Site Digital Forensics Specialist* starts the important task of evidence collection. Using forensic software and toolkits, he/she obtains and extracts evidence from various sources while ensuring their integrity and authenticity. Comprehensive documentation, particularly that related to chain of custody of digital evidence, is of utmost importance in this phase. In addition, eradication, for example, malware removal and disabling of breached user accounts (if any), is undertaken to ensure that the site is no longer vulnerable to that attack. Finally, *System Administrators* restore systems to normal operation. Recovery may involve such actions as using backups to restore systems when possible, performing clean installations, etc [17].

Per-Site Incident Closure. Once the incident is over and the system recovery is complete, it is important to identify the lessons that can be learned from the handling of the incident. A report containing a critical review of the entire process is placed before management. Based on that report, the *Site Executive* may take necessary steps for better preparedness that may include modifying the policy and procedures, making changes to the personnel of the incident response team, etc. The *Security/System Architect* may decide to modify the design of the system for better security. Additional software and hardware may be deployed by the *System Administrator* to bolster the defense of the system against future threats. Based on organizational policy, the *Incident Lead* decides on whether to store evidence and in what form. It should depend on factors like whether the prosecution is finished or not, the laws regarding data retention, hardware cost, etc.

ICIM Preparation. Like any site, the ICIM develops capabilities for handling incidents in this phase. This includes training of a response team and developing policies and procedures including a legal activities coordination plan. In addition to developing these capabilities for assisting a single site with an incident, the ICIM develops these capabilities for leading a collaborative effort in responding to a large-scale multi-site attack. This includes training of personnel to lead such collaborative teams and developing collaboration-specific policies, procedures, and legal activities coordination plans.

The cornerstone of preparing for a collaborative response is setup of a **workspace** hosting environment for multi-site collaborative investigation of large-scale cyber-attacks.

This environment allows *Collaboration Lead Investigators* to create workspaces and invite site and ICIM personnel to join the collaborative response. Each workspace corresponds to one incident and provides the collaboration access to tools, data and messages for executing the response and investigation process whereby each collaborator lives up to his/her responsibility as per the assigned role. A *Collaboration Workspace Administrator* is assigned for maintenance of this environment.

Incident Activity Analysis. As part of its day-to-day operations the ICIM receives reports about incidents at various sites in its purview. In this phase the ICIM undertakes an analysis of these reports to determine the level of response needed and the role that it needs to play in that response. When the analysis indicates a large-scale attack the ICIM may decide that a collaborative response is warranted. Examples include evidence indicating a growing or active botnet, zero-day exploit that affects multiple sites, website vandalism at multiple sites, and a request to do so from multiple *Site Leads*.

Collaborative Investigation. Once the ICIM decides on a collaborative response a *Collaboration Incident Lead* is identified who proceeds to set up a collaborative workspace for the incident with the assistance of the *Collaboration Workspace Administrator*. The *Collaborative Incident Lead* notifies other sites about the workspace and invites them to join. The *Collaborative Incident Lead* also performs different bootstrapping activities for the workspace including, but not limited to, assignment of *Collaborative Investigators* and *Collaborative Digital Forensics Analyst(s)* from the ICIM and other sites. In addition, depending on local laws and the nature and scale of the attack law enforcement is invited to participate in the collaboration and investigators are assigned appropriate roles.

An initial task of the collaboration is to formulate a strategy regarding containment, eradication, recovery and investigation. This strategy is documented within the workspace and often reviewed and updated as the process progresses. Data analysis is a crucial part of the investigation process. Availability of data from multiple sites opens up the possibility of performing cross-site analysis to establish links among events happening at individual sites. This analysis is conducted by *Collaboration Investigators*, *Collaboration Digital Forensic Analysts* and *Legal Investigators* and requires member sites to share data and communicate regularly. Based on the analysis the collaboration provides support to all sites for containment, eradication and recovery. While this analysis is being conducted, *Collaboration Digital Forensic Analysts* extract and store forensic evidence accumulated by the collaboration for legal prosecution. Based on the evidence, *Collaborative Investigators*, with the help from *Collaborative Digital Forensics Analysts*, reconstruct the digital crime scene/incident [11] and *Legal Prosecutors* and *Legal Investigators* formulate a legal prosecution strategy. As needed *Collaborative Investigators* interact with *Site Incident Investigators* in this phase to assist the latter with local investigation and recovery. One of the benefits of a collaborative effort is that the analysis in this phase can assist local site investigators to come up with strategies for local investigation and recovery. This includes assistance or guidance for evidence gathering and preservation, forensic tool usage, recovery, etc.

Collaboration Incident Closure. Once the investiga-

tion is over, appropriate legal steps are taken by the *Legal Prosecutor(s)* for prosecution. The evidence and theory developed in the analysis and reconstruction stage is presented to the appropriate authority. Dissemination of information [13] is another critical task in this final phase. Depending on the policy, the information may be shared with organizations that participated in the incident response or it may be added to a global knowledge repository. Finally, like participating sites, the ICIM should also have a policy on evidence retention for collaborative responses.

6. SECURITY ARCHITECTURE

At the core of our approach for collaborative response and investigation is the workspace environment that allows sites to instantiate incident workspaces and collaborate. Given the sensitive nature of this collaboration security for the workspace environment is crucial. In this section we discuss threats against the environment and our approach for addressing them. In our system design, we have worked to unify our security and system models [14] by modeling system threats and desired security properties.

Threat Model. We consider both insider and outsider threats to the workspace environment. Insiders (i.e., investigators with valid system logins) must obtain access to sensitive forensics data only as deemed necessary for the investigation. When multiple incident investigations are hosted inside the workspace environment, investigators’ access must be restricted to only the incidents they are investigating. Furthermore, access to forensics data within an incident investigation must be controlled to minimize disclosure of sensitive site information. In our experience, it is typical for site personnel to share forensics data only with a small number of trusted collaborators. The workspace environment must enable multiple sites to participate in the collaboration while limiting data disclosure between the sites inside the system, including disclosure of identifying information about the participants.

Outsiders include the suspects under investigation and others who would desire to obtain sensitive information from the workspaces or otherwise abuse system resources. Suspects under investigation must not be able to use information from the workspaces to help to cover their tracks or otherwise adjust their attack strategy. Furthermore, we must limit a suspect’s ability to disrupt the investigation via denial of service attacks against the workspace environment. Sensitive information that must not be disclosed to outsiders includes digital forensics data (containing sensitive site information), personal details about investigators (such as names, phone numbers, or IP/email addresses), or information about the capabilities and methods of investigators.

Authentication and Access Control. To limit workspace access to valid site and ICIM investigators we use strong two-factor authentication and to limit access to authorized data and resources between and within incident workspaces we use Role-Based Access Control. The security architecture is illustrated in Figure 3.

Role-Based Access Control (RBAC) is a natural choice for meeting the access control requirements of the collaborative environment. We map authenticated system users to per-incident roles following the approach presented earlier in Section 4. Authorized users have permission to create new incident workspaces and manage per-incident role-based permissions within the workspaces they create. Authenti-

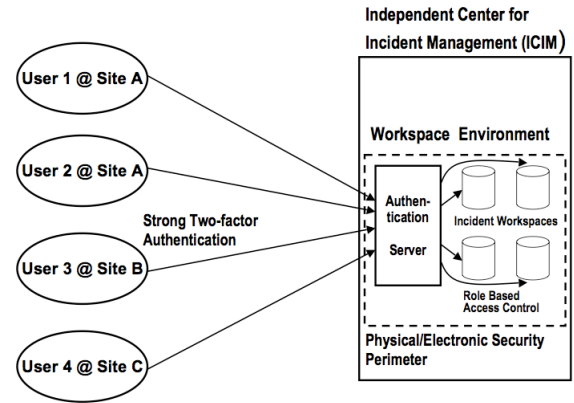


Figure 3: Security Architecture

cated users have no access to incident workspaces by default. They must be granted per-incident roles by the owner of the workspace.

Network Security. To protect the workspace environment from network-based attack, we establish a physical and electronic security perimeter around the environment that minimizes exposure via firewalls and private networks, requires encryption for all external network traffic, and employs network- and host-based intrusion detection. Database and data analysis services that support the user interface are deployed on a dedicated private network with no direct external network access. Leveraging a small number of standard protocols via well-known open source software enables system administrators to apply standard, best practice network security measures to address common attacks. For greater network security, the environment can be deployed inside a virtual private network to limit exposure to attacks from external networks.

Data Privacy. The collaborative environment must provide tools to enable collaborative incident response while respecting each site’s information disclosure policies. The disclosure of sensitive incident data is subject to privacy policies and laws, public relations, the desire to avoid disclosure to competitors and adversaries, and the desire to avoid negatively impacting an ongoing criminal investigation [8, 9, 15]. The ICIM plays a central role in overseeing and directing data disclosure among participants. This is crucial for enabling the collaboration as otherwise the involved organizations may not end up sharing the necessary information. The ICIM may use both technical and business means in supporting data disclosure. Technical means include the use of anonymization techniques that can hide sensitive information where appropriate; e.g., [33]. However, in many cases data may not be suitable for anonymization or may be rendered useless after doing so. In which case, the ICIM can utilize established procedures for obtaining approval from the organizations on each *type* of data such that the investigation team only needs to incur occasional overhead for data sharing. The collaborative environment must support flexible access control policies to facilitate data sharing according to the different information disclosure policies of the different participants. For example, some data may be provided to ICIM personnel only while other data may be shared among all collaborators. Thus, it is not necessary for all participants to agree on a common data sharing pol-

icy; instead, participants can specify and implement their desired access control policies on the data they provide to the collaboration.

7. ESTABLISHING TRUST

The collaborative incident response approach that we describe relies heavily on establishing trust between the responders from the affected organizations. As described in the Introduction, trust establishment is a major challenge because the affected organizations are chosen by the attacker, the collaboration may be formed only after the incident occurs, and the collaboration involves sensitive information. We have observed that some organizations have a strict policy against disclosure and cooperation during incident response, so they would be unwilling to participate in a collaborative approach under any circumstances. However, NCSA staff had very positive experiences collaborating with other organizations during the response to Incident 216 and other incidents, which indicates that many organizations see the value in working together to address large-scale distributed attacks. In this section, we discuss three methods for establishing trust during a collaborative response: (1) leveraging pre-existing collaborations, (2) utilizing trusted introducer groups and services, and (3) sharing incident information of interest to the participants.

Leveraging pre-existing collaborations. In today’s world of collaborative computing there is an opportunity to create ICIMs with the ability to quickly manage incidents that span these environments. For example, grid computing environments for scientific research, such as TeraGrid, Open Science Grid, the Enabling Grids for E-science (EGEE), and the Worldwide LHC Computing Grid (WLCG), have relatively stable member organizations that trust each other for resource sharing. Due to their common environments and user communities, security incidents can spread between these organizations, which has motivated them to share incident response contact information and establish processes for coordinated response that are primarily email-based. These existing collaborations could directly apply our proposed workspace mechanisms to enhance their existing coordinated processes.

Utilizing trusted introducer groups and services. While we can leverage pre-existing collaborations, we have seen that attackers do not respect their boundaries, and large-scale attacks often affect organizations with no prior working relationships. The incident response community has established groups and services to facilitate trust establishment in these cases. For example, the Trusted Introducer¹ network provides vetted contact information for CSIRTs in Europe and facilitates trusted information sharing among accredited response teams. Other groups such as the Forum of Incident Response and Security Teams (FIRST)² as well as CERTs and ISACs act as trusted introducers between organizations impacted by distributed attacks.

Sharing incident information of interest to the participants. Finally, responders can use information about the incident to establish trust with new organizations being invited to join the collaboration. An overview of the incident for new collaborators can be maintained in the incident workspace, including timelines, attack vectors, and recom-

mended mitigation techniques. When new collaborators see that the details in the incident overview match what they are seeing inside their organization, and they benefit from the recommended mitigation techniques listed in the overview, they are more inclined to join the collaborative response effort. Furthermore, specific details about the attack can be very effective at gaining the interest of new collaborators, as illustrated by the following anecdote. During Incident 216, one of the responders needed timely assistance from a new organization and was having difficulty getting a response. He noticed that the attacker had collected the password of one of the CSIRT members from the organization, so he asked him, “Is this your password?” When the CSIRT member recognized his password, he responded, “Now you’ve got my attention!”

8. DESIGN AND IMPLEMENTATION

We have developed the “Palantir” prototype system to provide a software environment that supports the collaborative response and investigation process. The Palantir system provides the collaborative workspace for discussions and data sharing among incident investigators, as seen in Figure 4. Collaboration mechanisms in the workspace include a *data repository* for log files, network traces, and other forensic data, a *wiki* for providing an overview of the incident for new members, documenting incident details, and keeping a timestamped incident notebook, secure *instant messaging* for real-time discussions, secure *email lists* [7, 19, 20] for ongoing discussions, *anonymization tools* [33] for sanitizing data before it is shared, *analysis tools* [10], and *visualization tools* [36].

Our implementation is a web application built on open source web software that can be accessed by standard web browsers. We use the Liferay Portal³ platform, running in the Apache Tomcat⁴ container, connected to the Apache HTTP⁵ server. Building on open source software enables independent verification of software security through source code reviews and scanning, as performed for the Apache HTTP server by the Scan Project⁶ and for Liferay and Tomcat by the Java Open Review Project⁷.

Liferay supports secure chat services via the standard XMPP (Jabber) protocol using the open source Openfire⁸ Jabber server. Responders can chat via a portlet within Liferay (over HTTPS) or via desktop Jabber chat clients running the Jabber protocol over TLS.

For strong two-factor authentication, we support both one-time password (OTP) hardware tokens and PKI-based smartcards. The OTP tokens and smartcards both require a PIN to unlock, providing both “something you have” and “something you know” authentication factors. When the OTP token is unlocked, it displays a one-time use password that the Palantir user enters at the login prompt. When the smartcard is unlocked, it authenticates to the server via the TLS protocol using private cryptographic data residing on the card. While smartcards save the user from manually entering a one-time password, they require hardware and

³<http://liferay.com/>

⁴<http://tomcat.apache.org/>

⁵<http://httpd.apache.org/>

⁶<http://scan.coverity.com/>

⁷<http://opensource.fortifysoftware.com/>

⁸<http://www.igniterealtime.org/projects/openfire/>

¹<http://www.trusted-introducer.nl/>

²<http://www.first.org/>



Wiki Display

Incident Log » FrontPage

2007-05-01 3:50pm
Trojan SSH daemon discovered on login node 5.

2007-05-01 4:10pm
Login node 5 network flows uploaded.



Calendar

Summary							Day	Week
Month		Year		Events				
October 1, 2007								
S	M	T	W	T	F	S		
	1	2	3	4	5	6		
7	8	9	10	11	12	13		
14	15	16	17	18	19	20		
21	22	23	24	25	26	27		
28	29	30	31					

Add Event

Time	Title	Type
There are no events on this day.		

Chat

- Bill Baker
- Joe Muggli
- Mike Freeman
- Von Welch

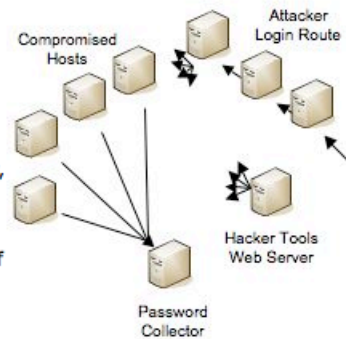


Wiki Display

Incident Overview » FrontPage

Incident ID: abcd-20070501-22

The attacker behind this incident used a well-organized process for compromising a large number of hosts and then harvesting user passwords to continue to expand his set of compromised hosts. The attacker initially compromised some number of hosts using known exploits. He then installed Trojan secure shell (SSH) clients on these systems that harvested host, username and password tuples as users used the Trojan SSH clients to logon to other systems. The attacker then used those stolen credentials to logon to those systems and then used a number of exploits to gain administrative privileges using known exploits for privilege escalation. Once administrative privileges were gained, the attacker would then install a rootkit to hide himself and Trojan the SSH clients to use the new system to gather further account information to repeat the process and grow the base of compromised systems. Besides the fact that the attacker's collection of distributed systems was spread across multiple domains, the attacker also had supporting infrastructure that was also spread over multiple domains. The figure above shows these supporting systems, which include:



A Password Collector. Every time a Trojan SSH client captured a hostname, username and password tuple, it sent this information over the network to the Password Collector host. The Password Collector host was one of the compromised hosts where the attacker installed a service to collect and record these tuples for latter use.

A Dynamic DNS Service. The Trojan SSH clients used a statically configured hostname to address their network traffic with the captured tuples. This hostname was managed by the attacker through a public dynamic DNS site that allowed him to manage the mapping of the hostname to an IP address anonymously via a web form. This allowed him to move the Password Collector several times during the investigation when he felt it was potentially discovered and being monitored.

Hacker Tools Repository. On one of the hosts the attacker compromised, he installed a set of exploits that he used for privilege escalation. These tools were made available via a web server already installed on the host. After gaining access to a new host, he would download these tools and use them to gain privileged access.

Login Route. Instead of logging in directly from his local system to compromised system, the attacker always went through a series of distributed intermediate systems. Presumably this was done to make the task of tracking a session back to the attacker difficult.



Figure 4: A Palantir Workspace

software support (i.e., readers and drivers) to interface with the user's desktop.

The open source Secure Email List Services (SELS)⁹ software provides support for email-based group discussions in Palantir. SELS uses the OpenPGP standard for compatibility with commonly available email client plugins from the Gnu Privacy Guard (GnuPG) project¹⁰. SELS uniquely

⁹<http://sels.ncsa.uiuc.edu/>

¹⁰<http://gnupg.org/>

provides end-to-end privacy for email discussion lists using proxy cryptography, whereby messages are protected both on the network and the mailing list server.

The open source Framework for Log Anonymization and Information Management (FLAIM)¹¹ supports anonymization of log files on the responder's desktop before upload into the collaborative environment, as well as anonymization by the Palantir server during file upload and prior to export.

¹¹<http://flaim.ncsa.uiuc.edu/>

Supported log types include pcap, netfilter, NetFlows, and Unix process accounting.

Roles and Responsibilities. We now describe how the roles and responsibilities from Section 4 map to the Palantir system’s capabilities.

The *Collaboration Incident Lead* is responsible for creating and managing the incident workspace, with the assistance of the *Collaboration Workspace Administrator*. The *Collaboration Incident Lead* adds *Collaboration Investigators* to the workspace, where they can coordinate their efforts via wiki pages and discussions over instant messaging and email. He also maintains a primary wiki page for the incident with an incident overview, current status, and technical information to be shared among all participants.

Collaboration Investigators learn information about the investigation that informs their local site’s response, as well as contribute their knowledge to the collaborative effort. If an investigator obtains information relevant to other sites, he can share it via the workspace. *Collaborative Investigators* can upload evidence for analysis by other *Collaborative Investigators* and *Collaboration Digital Forensics Analysts*. The workspace provides anonymization tools that the *Collaborative Investigators* can apply to their site’s data before sharing it.

The *Collaboration Digital Forensics Analysts* apply forensics tools available in the workspace to the forensics data provided by the *Collaboration Investigators*. The analysts publish requests for evidence, guidelines for evidence collection, and analytical results to wiki pages to inform the other collaborative participants.

Other areas of the workspace, established by the *Collaboration Incident Lead* as needed, provide forums for collaboration among *Legal Advisors*, *Media Liasons*, and *Law Enforcement*. For example, *Media Liasons* can draft joint press statements in the workspace.

Process Model. The Palantir system does not enforce a specific process model on participants. Instead, the collaborators can use the available tools in the workspace as they see fit according to the response and investigation strategy they have developed. The Palantir system allows subgroups to form and collaborate privately within the investigation workspace, before sharing their results with the larger group. The *Collaboration Incident Lead* can use the incident’s secure mailing list to direct and track the group’s work. Wiki pages can document current tasks and milestones in the investigation, updated as they are completed. Upon further experience, we may augment the Palantir system with forms and dialogs that facilitate common incident workflows based on best practices.

As we see in Figure 2, coordination is required between the local site incident response and the collaborative process. A simple but important practice for facilitating this coordination is for each site to record their local tracking number for the incident in the Palantir incident wiki. Palantir creates a unique tracking number for each workspace, following the recommendations in [8].

Workspace Template. In order to realize the Roles and Responsibilities Model and the Process Model in incident workspaces, Palantir provides a Workspace Template that is instantiated for every incident. The template provides ready-to-use containers for each workspace where users can be assigned to roles and automatically get access to an authorized set of resources. The template currently imple-



Figure 5: Palantir Workspace Template

mented in Palantir is described in Figure 5 and is easily customizable. For each role identified in Section 4 the template specifies: 1) the role that can assign users to this role, 2) the default view (interface layout) when this role is activated, 3) set of tools (via Liferay portlets) that this role can access, and 4) a default set of resources (objects such as files) that this role can access via each tool. A factory within Liferay generates the necessary resources and access policies based on the specified template.

9. EVALUATION

To evaluate the Palantir approach, we describe how the Palantir system would have assisted in the collaborative investigation effort conducted for Incident 216. Collaboration played a very important role in this investigation for tracking the attacker’s widely-distributed activities, understanding the attacker’s methods, and finally locating and apprehending the attacker.

Compromise Tracking and Notification. Notifying sites that they had been compromised was one of the most time-consuming activities in the Incident 216 investigation. Network traffic and server logs from the attacker’s password collectors and web servers provided information about compromised systems to the incident investigators. Investigators analyzed the latest information each day and notified personnel at newly compromised sites. Network, security, and system administrators at different sites gathered and provided this data to the investigators. The attacker moved the password collector several times, requiring the investigators to contact new administrators to re-establish their monitoring capabilities.

Managing the network and server logs for daily analysis was a manual process. Palantir’s data repository provides the capability for administrators to directly upload their

data to the investigators via the secure web interface. Investigators can use wiki pages to track which logs have been analyzed and which sites have been contacted. Using Palantir tools, the investigators can automate the daily analysis process.

When contacting newly compromised sites, it is helpful for the investigators to have a standard incident overview to share with site personnel. During Incident 216, this overview was maintained by a single investigator, but Palantir’s secure wiki would allow it to be written and updated collaboratively by multiple investigators. Additionally, new sites can obtain logins to the Palantir system to read the wiki pages and participate in the investigation.

Collaborative Analysis. Incident 216 investigators were hindered by their inability to read the encrypted network traffic from the attacker’s rootkit. From analyzing network logs, an administrator at one site identified the encryption protocol being used but needed the encryption key. Investigators asked for help from colleagues skilled in reverse engineering, who were able to analyze the rootkit binary to locate the encryption key. With the key, the administrator developed a decryption tool that he shared with the other investigators. Investigators were lucky that the administrator at this site had both access to the rootkit logs and the skill to develop a decryption tool. If this had not been the case, the administrator could have used Palantir to upload the logs to be analyzed by another participant.

This is one of many examples in the Incident 216 investigation of system administrators who were highly motivated to contribute to the investigation. By sharing information with them and allowing them to contribute, the investigation benefited greatly from their expertise.

As described in Section 2, the Incident 216 collaborative investigation and analysis was hindered by ad hoc communication methods. Palantir’s secure instant messaging, email lists, and wiki pages provide convenient and trustworthy communication mechanisms for the investigators.

10. RELATED WORK

Starting with work that leverages experiences in dealing with paper evidence [25], considerable effort has been spent in developing models for the digital investigation process. This includes the Digital Forensic Science process [24], the End-to-End Digital Investigation Process [34], an approach for forensics in military settings [16], the Integrated Digital Investigation Model [11], the Digital Crime Scene Investigation process [12], the Enhanced Digital Investigation Model [5], a process for integrating investigations with information flow [13], the FORZA [18] framework that emphasizes legal issues, a two-tier investigation approach [6] and the combined forensics and intelligence gathering framework [31]. Reith *et al.* [28] and Pollitt [26] provide good surveys of these and other related works.

Similarly, models have been developed that deal primarily with how individual sites should respond to incidents, usually with the help of a predesignated incident response team. This includes the Incident Response Life Cycle [17], an incident response methodology [27], guidelines for formation and operation of CSIRTs [8], a study of organization models and their impact on incident response [21], best practices and guidelines [3, 29] and a corporate framework for incident management [23]. Additionally, software systems have been developed to help CSIRTs internally manage incident inves-

tigations including ticket tracking [22] and request tracking [35].

Going beyond CSIRTs, a number of efforts have been launched worldwide to establish institutions that coordinate response to large-scale multi-site attacks. Examples include the Computer Emergency Response Teams (CERTs) and Information Sharing and Analysis Centers (ISACs) currently being operated worldwide.

Our work is significantly different in that we focus on a framework for supporting multi-site collaborative digital investigation and incident response. We integrate the two areas of relevant work, namely, digital investigation and incident response, by developing models for Roles and Responsibilities as well as Processes that identify their interaction. Furthermore, we propose to extend the scope of CERTs/ISACs to support effective *collaborations* involving multiple organizations by additionally becoming ICIMs. In particular, these novel enhancements result in a Process Model (see Figure 2) that allows incident responders and investigators across multiple sites affected by an attack to effectively collaborate in their common goals of investigating and responding to the attacks. To the best of our knowledge this is the first work proposing a system model and implementation for a collaborative multi-site incident response and investigation effort. We believe that this work can help develop capabilities to adequately prepare for large-scale attacks such as Incident 216. The US Department of Homeland Security has conducted two exercises for large-scale cyber attacks, Cyber Storm I (February 2006) and Cyber Storm II (March 2008). From the public report of Cyber Storm I [1] it is clear that even with the presence of CSIRTs, CERTs and ISACs, tools and technologies that provide advanced collaboration capabilities for incident response and investigation are needed.

Many software systems are available to help CSIRTs internally manage incident investigations. Open Source incident ticket tracking systems include Application for Incident Response Teams (AIRT) [22], Request Tracker for Incident Response (RTIR) [35], and System for Incident Response in Operational Security (SIRIOS)¹². The Internet2 Research and Educational Networking Operational Information Retrieval (RENOIR)¹³ project is developing a system for incident reporting to a trusted third-party such as REN-ISAC (Research and Education Networking ISAC)¹⁴. This work is complementary to ours. We assume good internal incident management and incident reporting mechanisms are in place, and we focus on collaborative incident response in reaction to large-scale, distributed incidents. While individual components of our solution, such as secure wikis and instant messaging, are starting to see more widespread use in the incident response community, we believe our work is the first to bring together these components into an integrated environment for collaborative incident response.

11. CONCLUSIONS AND FUTURE WORK

Organizations with cyber-infrastructure assets face large-scale distributed attacks on a regular basis. Based on lessons learned in dealing with such an attack and the realization that the complexity and frequency of such attacks is increas-

¹²<http://sirios.org/>

¹³<http://security.internet2.edu/csi2/>

¹⁴<http://www.ren-isac.net/>

ing in general, we argue that is insufficient to rely on organizational incident response teams or even trusted coordinating response teams. Instead, there is need to develop a framework that allows an effective collaborative response and investigation process that include multiple organization and legal entities to track the adversary, eliminate the threat and pursue prosecution of the perpetrators. To that end we develop a system model and prototype implementation that would provide the ability to execute this collaborative process led by an Independent Center for Incident Management (ICIM). The system model defines an appropriate set of roles and responsibilities as well as the process undertaken by the collaboration. We describe a workspace environment supported by ICIMs and define a security architecture for the environment that leverages the roles in the system model for role based access control. We then describe a prototype implementation of the workspace environment, called the Palantir system, that provides a collaboration access to necessary tools and resources for undertaking the response and investigation while enforcing the security requirements. In addition, we define a workspace template for incident workspaces that supports our system model for roles, responsibilities and processes.

Several directions of future work can greatly benefit the proposed system model and prototype. First, the RBAC model can be enriched with the addition of role hierarchies, delegations and constraints that provide fine-grained access control and advanced policies such as separation-of-duty. Enforcement of constraints will require a reference monitor in the workspace environment, which can be designed by adapting techniques for RBAC in collaborative settings [2]. Second, while the process model does not lend itself to well-defined workflows there exist several tasks that can be combined into workflows for efficiency and correctness; e.g., uploading and analyzing logs. Such workflows can be supported by developing the concept of *wizards* in the workspace environment that allow users to combine tasks into workflows. Third, the usability aspects of the workspace environment can be significantly improved by undertaking usability studies and interface enhancements. Based on comments from early adopters, we are exploring the possibility of supporting command-line and thick-client interfaces to the workspace, using CyberIntegrator [4] to capture data provenance and manage workflows. Fourth, further evaluation of the system in handling real large-scale incidents will help provide useful enhancements and validation.

12. ACKNOWLEDGMENTS

We would like to thank the members of the NCSA Incident Response team, whose efforts in Incident 216 and subsequent discussions led to our understanding of these challenges. Members of that team at the time of Incident 216 included Jim Barlow, Tim Brooks, Aashish Sharma and Jeff Rosendale.

This work was funded by the Office of Naval Research under award number N00014-06-1-1108. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research or the United States Government.

13. REFERENCES

- [1] Cyber Storm Exercise Report. National Cyber Security Division, U.S. Department of Homeland Security, September, 2006, 2006.
- [2] T. Ahmed and A. R. Tripathi. Specification and verification of security requirements in a programming model for decentralized csw systems. *ACM Trans. Inf. Syst. Secur.*, 10(2):7, 2007.
- [3] C. Alberts, A. Dorofee, G. Killcrece, R. Ruefle, and M. Zajicek. Defining Incident Management Processes for CSIRTs: A Work in Progress. Technical Report CMU/SEI-2004-TR-015, Software Engineering Institute, Carnegie Mellon University, 2004.
- [4] P. Bajcsy, R. Kooper, L. Marini, B. Minsker, and J. Myers. CyberIntegrator: A Meta-Workflow System Designed for Solving Complex Scientific Problems using Heterogeneous Tools. In *Proceedings of the Geoinformatics Conference*, May 2006.
- [5] V. Baryamureeba and F. Tushabe. The Enhanced Digital Investigation Process Model. *Process Model Asian Journal of Information Technology*, 2006.
- [6] N. Beebe and J. G. Clark. A hierarchical, objectives-based framework for the digital investigations process. *Digital Investigation*, 2(2):147–167, 2005.
- [7] R. Bobba, J. Muggli, M. Pant, J. Basney, and H. Khurana. Usable secure mailing lists with untrusted servers. In *Symposium on Identity and Trust on the Internet (IDtrust)*, 2009.
- [8] M. J. W. Brown, D. Stikvoort, K. P. Kossakowski, K. P. Kossakowski, G. Killcrece, R. Ruefle, and M. Zajicek. Handbook for Computer Security Incident Response Teams (CSIRTs). CMU/SEI-2003-HB-002, April, 2003, 2003.
- [9] N. Brownlee and E. Guttman. Expectations for Computer Security Incident Response. IETF RFC 2350, June 1998.
- [10] Y. D. Cai, D. Clutter, G. Pape, J. Han, M. Welge, and L. Auvil. Maids: mining alarming incidents from data streams. In *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 919–920, New York, NY, USA, 2004. ACM Press.
- [11] B. Carrier and E. H. Spafford. Getting Physical with the Digital Investigation Process. *International Journal of Digital Evidence*, 2(2), Fall 2003.
- [12] B. Carrier and E. H. Spafford. An Event-Based Digital Forensic Investigation Framework. In *DFWRS'04: Proceedings of the 4th Digital Forensics Research Workshop*, 2004.
- [13] S. Ó. Ciardhuáin. An Extended Model of Cybercrime Investigations. *International Journal of Digital Evidence*, 3(1), Summer 2004.
- [14] P. T. Devanbu and S. Stubblebine. Software engineering for security: a roadmap. In *ICSE '00: Proceedings of the Conference on The Future of Software Engineering*, pages 227–239, New York, NY, USA, 2000. ACM Press.
- [15] B. Fraser. Site Security Handbook. IETF RFC 2196, Sept. 1997.
- [16] J. Giordano and C. Maciag. Cyber Forensics: A Military Operations Perspective. *International Journal*

- of *Digital Evidence*, 1(2), Summer 2002.
- [17] T. Grance, K. Kent, and B. Kim. Computer Security Incident Handling Guide: Recommendations of the National Institute of Standards and Technology. *NIST Special Publication 800-61*, January 2004.
- [18] R. S. C. Ieong. FORZA - Digital forensics investigation framework that incorporate legal issues. *Digital Investigation*, 3(Supplement-1):29–36, 2006.
- [19] H. Khurana, J. Heo, and M. Pant. From proxy encryption primitives to a deployable secure-mailing-list solution. In *ICICS'06: International Conference on Information and Communications Security*, pages 260–281, 2006.
- [20] H. Khurana, A. J. Slagell, and R. Bonilla. SELS: a secure e-mail list service. In *ACM Symposium on Applied Computing (SAC), Security Track*, pages 306–313, 2005.
- [21] G. Killcrece, K.-P. Kossakowsk, R. Ruefle, and M. Zajicek. Organizational Models for Computer Security Incident Response Teams (CSIRTs). Technical Report Report: CMU/SEI-2003-HB-001, Carnegie Mellon University/Software Engineering Institute, 2003.
- [22] K. Leune and S. Tesink. Designing and developing an Application for Incident Response Teams. In *FIRST'06: Forum for Incident Response Teams Conference*, Baltimore, MD, USA, June 2006.
- [23] S. Mitropoulos, D. Patsos, and C. Douligeris. On Incident Handling and Response: A state-of-the-art approach. *Computers & Security*, 25(5):351–370, July 2006.
- [24] G. Palmer. A Road Map for Digital Forensic Research. Technical Report Technical Report DTR-T001-01, Report From the First Digital Forensic Research Workshop (DFRWS), 2001.
- [25] M. Pollitt. Computer Forensics: an Approach to Evidence in Cyberspace. In *Proceedings of the National Information Systems Security Conference*, volume 2, pages 487–491, 1995.
- [26] M. M. Pollitt. An Ad Hoc Review of Digital Forensic Models. In *SADFE '07: Proceedings of the Second International Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 43–54, Washington, DC, USA, 2007.
- [27] C. Prorise, K. Mandia, and M. Pepe. *Incident Response and Computer Forensics, Second Edition*. McGraw-Hill Osborne Media, 2003.
- [28] M. Reith, C. Carr, and G. Gunsch. An Examination of Digital Forensic Models. *International Journal of Digital Evidence*, 1(3), Fall 2002.
- [29] R. L. Rollason-Reese. Incident handling: an orderly response to unexpected events. In *SIGUCCS '03: Proceedings of the 31st annual ACM SIGUCCS conference on User services*, pages 97–102. ACM Press, 2003.
- [30] R. Rowlingson. A Ten Step Process for Forensic Readiness. *International Journal of Digital Evidence*, 2(3), Winter 2004.
- [31] G. Ruijin, C. Kai, Y. Tony, and M. Gaertner. Case-Relevance Information Investigation: Binding Computer Intelligence to the Current Computer Forensic Framework. *International Journal of Digital Evidence*, 4(1), Spring 2005.
- [32] S. Schechter, J. Jung, W. Stockwell, and C. McLain. Inoculating SSH Against Address Harvesting. In *NDSS'06: The 13th Annual Network and Distributed System Security Symposium*, San Diego, CA, February 2006.
- [33] A. Slagell, K. Lakkaraju, and K. Luo. FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs. In *LISA'06: 20th USENIX Large Installation System Administration Conference*, Washington, D.C., Dec. 2006.
- [34] P. Stephenson. Modeling of Post-Incident Root Cause Analysis. *International Journal of Digital Evidence*, 2(2), Fall 2003.
- [35] J. Vincent, R. Spier, D. Rolsky, D. Chamberlain, and R. Foley. *RT Essentials*. O'Reilly Media, Aug. 2005.
- [36] X. Yin, W. Yurcik, and A. Slagell. VisFlowCluster-IP: Connectivity-Based Visual Clustering of Network Hosts. In *21st IFIP TC-11 International Information Security Conference (SEC '06)*, May 2006.

Palantir: A Framework for Collaborative Incident Response and Investigation

Himanshu Khurana, Jim Basney, Mehedi Bakht, Mike Freemon, Von Welch, Randy Butler

IDTrust, April 14 – 16, 2009.

Gaithersburg, MD.



Introduction

- **Computer Security Incident Response Teams (CSIRTs) are becoming common**
 - Digital investigation and forensics
 - Recovery and restoration
 - Working with law enforcement
- **CERTs and ISACs provide information sharing and coordination**
 - Vulnerabilities, exploits, policy documents
- **Recently experienced large-scale cyber attacks require *collaboration* and not just *coordination***

FBI Major Case 216 – Stakkato (2004)

The New York Times

Internet Attack Called Broad and Long Lasting by Investigators

SAN FRANCISCO, May 9 – The incident seemed alarming enough: a breach of a Cisco Systems network in which an intruder seized programming instructions for many of the computers that control the flow of the Internet.

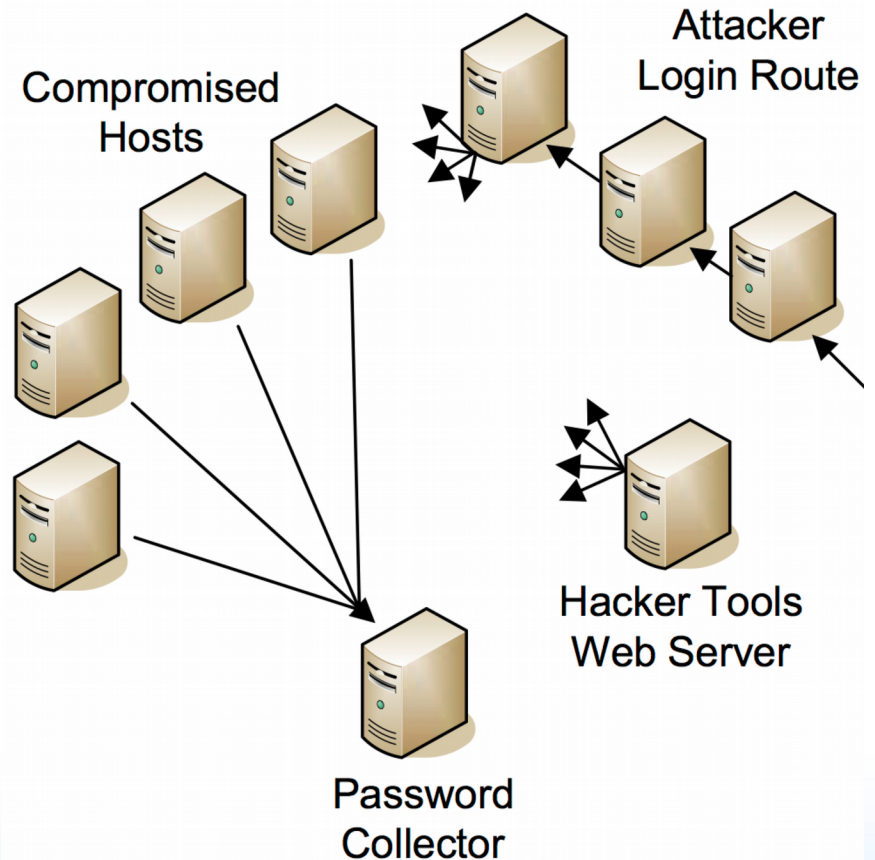
Now federal officials and computer security investigators have acknowledged that the Cisco break-in last year was only part of a more extensive operation – involving a single intruder or a small band, apparently based in Europe – in which thousands of computer systems were similarly penetrated. [...]

Attention is focused on a 16-year-old in Uppsala, Sweden. [...]

As the attacks were first noted in April 2004, a researcher [...] began to receive taunting e-mail messages from someone going by the name Stakkato [...]

FBI Major Case 216 – Stakkato (2004)

- Broad attack against supercomputing centers, DOE labs, Universities and other government and commercial sites internationally.
- Simple attack strategy with a highly complex infrastructure
- Compromise host -> escalate privilege -> install trojan SSH -> capture passwords -> use SSH known hosts file to find next victim
- Lather. Rinse. Repeat



Lessons learned from Major Case 216

- **Multi-site collaboration is hard**
 - Tens of sites in multiple countries
 - Trust establishment
 - Partners not known in advance
 - Data collection and analysis
 - Limited skills at each site
 - Privacy and security issues with data sharing
 - Lack of tools
 - Communication and interactions
 - Adversary was monitoring emails
 - Total process took 9 months
 - Law enforcement finally stopped the attacks

Requirements and Challenges

- **Requirements**

- Trust establishment for data sharing
- Management of tasks and activities
- Data management and analysis

- **Challenges**

- Interactions between site and collaborative activities
 - Effective responsibilities and reporting
 - Integrating technical and legal investigation
 - Tracking global progress

Approach

- **Principles**

- Mutual benefit for trust establishment
- Focused collaboration
- Least privilege for access control

- **Design**

- Propose a system model for collaboration
 - Roles and responsibilities
 - Process model
- Secure architecture
 - ICIM: Independent Center for Incident Management
- Analysis tools (open source)
 - System implementation and prototype

Roles and Responsibilities

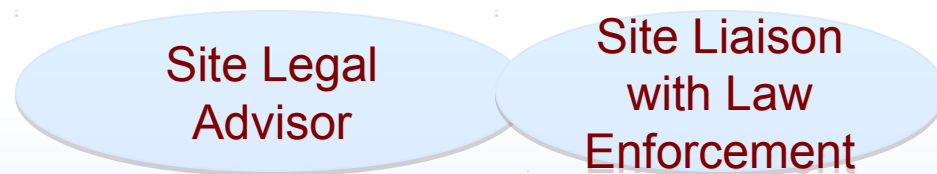
Site Technical Roles



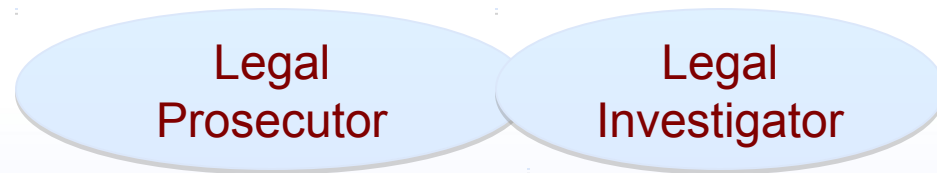
Collaboration Technical Roles



Site Legal Roles



Law Enforcement Roles



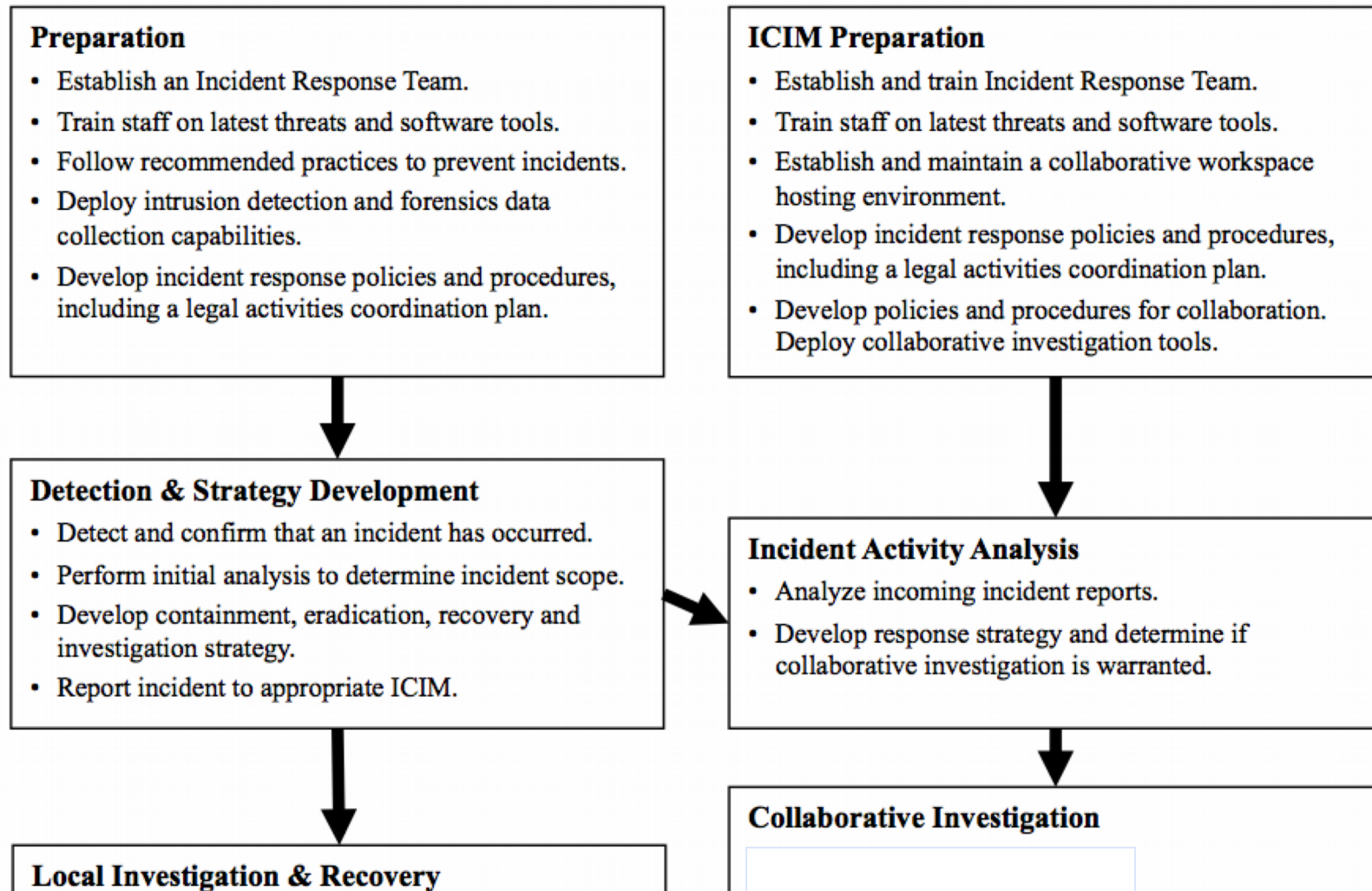
Other Site Roles



Process Model

Site

ICIM / Collaboration



Process Model

Site

Detection & Strategy Development

Local Investigation & Recovery

- Contain the breach to prevent further damage.
- Collect and preserve evidence in a forensically sound manner.
- Eradicate malware and disable compromised systems/accounts.
- Deploy counter-measures to prevent repeat occurrence of compromise.
- Restore normal system operation.

Incident Closure

- Identify lessons learned.
- Complete incident report.
- Improve future preparedness.
- Retain evidence as required according to policy.

ICIM/Collaboration

Incident Activity Analysis

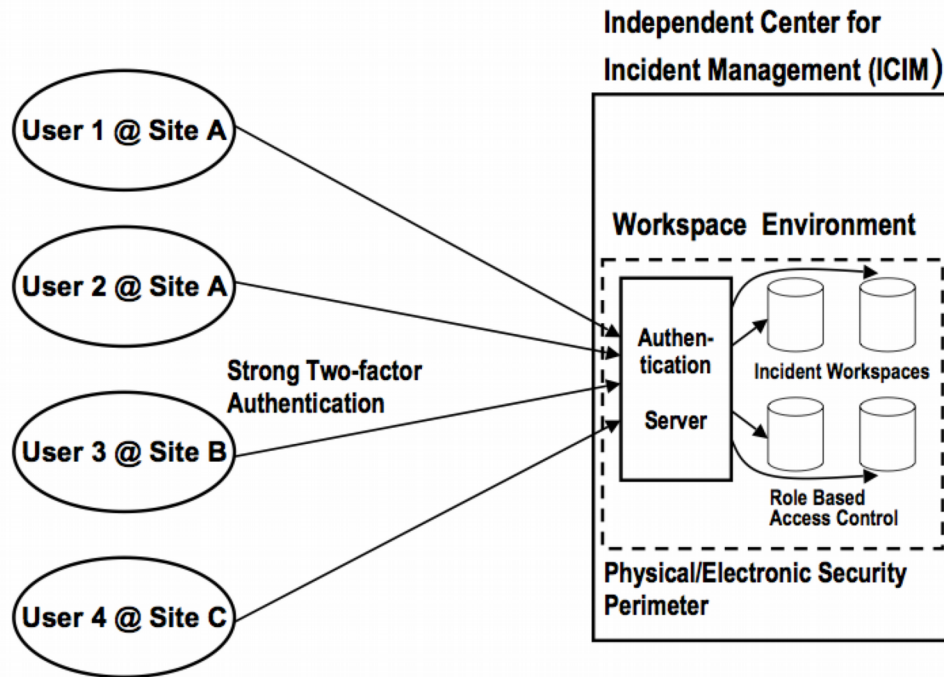
Collaborative Investigation

- Create collaborative workspace.
- Invite collaborators and assign roles.
- Formulate collaborative response and investigation strategy.
- Share (anonymized) evidence as appropriate.
- Perform cross-site data analysis and correlation.
- Discuss (ongoing) incident and share insights.
- Cooperate in containment and recovery.
- Reconstruct the crime scene. Prepare coordinated legal strategy.

Incident Closure

- Legally prosecute the offenders.
- Share lessons learned among participants and publicly as appropriate.
- Retain evidence according to policy.

Architecture and Security



- Address both outsider and insider threats
- Strong authentication and authorization
 - Two-factor, RBAC
- Enforce data privacy
 - Disclosure policies, anonymization

Incident Workspace and User Interface

Incident ID: abcd-20070515-29

Incident ID: abcd-20070510-25

Incident ID: abcd-20070501-22

Incident Lead: HumphreyBogart

Participants:

Investigators: FrankSinatra, DeanMartin

Forensics Analysts: CesarRomero

Legal Advisors: PeterLawford

Law Enforcement: JoeyBishop

Media Liasons: SammyDavisJr

Wiki pages: Incident Overview, Incident Log

Mailing lists:

abcd-20070501-22-announce

abcd-20070501-22-discuss

abcd-20070501-22-forensics

abcd-20070501-22-legal

abcd-20070501-22-media

Views:

Home: Wiki, Chat

Analysis: File Management, Log Analysis,
Anonymization Tools



Home

Analysis

Add Page

Welcome, Jim Basney!



Wiki Display

Incident Log » FrontPage

2007-05-01 3:50pm
Trojan SSH daemon discovered on login node 5.

2007-05-01 4:10pm
Login node 5 network flows uploaded.



Calendar

Summary Day Week

Month Year Events

October 1, 2007

S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Add Event

Time Title Type

There are no events on this day.

Chat

Bill Baker
Joe Muggli
Mike Freeman
Von Welch

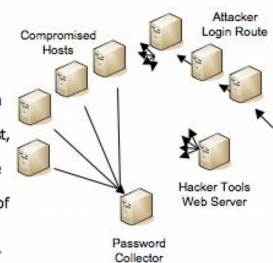


Wiki Display

Incident Overview » FrontPage

Incident ID: abcd-20070501-22

The attacker behind this incident used a well-organized process for compromising a large number of hosts and then harvesting user passwords to continue to expand his set of compromised hosts. The attacker initially compromised some number of hosts using known exploits. He then installed Trojan secure shell (SSH) clients on these systems that harvested host, username and password tuples as users used the Trojan SSH clients to logon to other systems. The attacker then used those stolen credentials to logon to those systems and then used a number of exploits to gain administrative privileges using known exploits for privilege escalation. Once administrative privileges were gained, the attacker would then install a rootkit to hide himself and Trojan the SSH clients to use the new system to gather further account information to repeat the process and grow the base of compromised systems. Besides the fact that the attacker's collection of distributed systems was spread across multiple domains, the attacker also had supporting infrastructure that was also spread over multiple domains. The figure above shows these supporting systems, which include:



A Password Collector. Every time a Trojan SSH client captured a hostname, username and password tuple, it sent this information over the network to the Password Collector host. The Password Collector host was one of the compromised hosts where the attacker installed a service to collect and record these tuples for latter use.

A Dynamic DNS Service. The Trojan SSH clients used a statically configured hostname to address their network traffic with the captured tuples. This hostname was managed by the attacker through a public dynamic DNS site that allowed him to manage the mapping of the hostname to an IP address anonymously via a web form. This allowed him to move the Password Collector several times during the investigation when he felt it was potentially discovered and being monitored.

Hacker Tools Repository. On one of the hosts the attacker compromised, he installed a set of exploits that he used for privilege escalation. These tools were made available via a web server already installed on the host. After gaining access to a new host, he would download these tools and use them to gain privileged access.

Login Route. Instead of logging in directly from his local system to compromised system, the attacker always went through a series of distributed intermediate systems. Presumably this was done to make the task of tracking a session back to the attacker difficult.



Example Integrated Tools



- FLAIM: Framework for Log Anonymization and Information Management



- SELS: Secure Email List Services



- Liferay: Portal and collaboration software



- Openfire: XMPP/Jabber Chat

Future Work

- Software packaging and release
- Evaluation in test environment
- Usability studies
- Explore deployment avenues
 - Explore suitability of integration with CERTs/ISACs
 - Quick deployment by any organization involved in an incident
- **QUESTIONS?**
 - Contact: Himanshu Khurana; hkhurana@illinois.edu, Randy Butler; rbutler@ncsa.uiuc.edu

Backup Slides

Establishing Trust – a people problem

- **Challenges**

- Information privacy (user and organization)
- Publicity concerns
- Lack of clear incentives for information sharing

- **Potential approaches**

- Leverage pre-existing relationships
 - E.g., Computational grid systems
- Utilizing trusted introducer groups and services
 - E.g., CERTs, ISACs, FIRST, Trusted Introducer
- Share information to establish trust
 - Mutual benefit

Safeguarding Digital Identity: The SPICI (Sharing Policy, Identity, and Control Information) Approach to Negotiating Identity Federation and Sharing Agreements

Deborah Bodeau
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730
1-781-271-8436
dbodeau@mitre.org

ABSTRACT

To perform key business functions, organizations in critical infrastructure sectors such as healthcare or finance increasingly need to share identifying and authorization-related information. Such information sharing requires negotiation about identity safeguarding policies and capabilities, as provided by processes, technologies, tools, and models. That negotiation must address the concerns not only of the organizations sharing the information, but also of the individuals whose identity-related information is shared. SPICI (Sharing Policy, Identity, and Control Information) provides a descriptive and analytic framework to structure and support such negotiations, with an emphasis on assurance.

Categories and Subject Descriptors

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

General Terms

Security

Keywords

Identity Management Identity Federation, Information Sharing, Credentials

1. INTRODUCTION

To perform key business functions, organizations in critical infrastructure sectors such as healthcare or finance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust 2009, April 14-16, 2009, Gaithersburg, MD, U.S.A.
Copyright© 2009 ACM 978-1-60558-474-4...\$5.00

increasingly need to share identifying and authorization-related information. Thus, organizations increasingly need to negotiate agreements for identity federations or other sharing of identifying and/or authorization-related information. Such negotiations cover, among other topics, identity safeguarding policies and capabilities required to implement policies, provide agreed-upon functionality, and thus meet business needs while managing risks.¹

Negotiations that lead to contractual or other documented agreements to share identity-related information must address the concerns not only of the organizations sharing the information, but also of the individuals whose identity-related information is shared. A common framework for assessing potential harms – to the partnering organizations that share and rely upon identifying information and to the identified individuals – facilitates agreement on a risk-appropriate level of assurance.

The SPICI (Sharing Policy, Identity, and Control Information) approach is being developed under the Institute for Information Infrastructure Protection (I3P) Safeguarding Digital Identity project [1]. SPICI is intended to help organizations identify the capabilities they need, and to negotiate how they will provide those capabilities via technologies and business processes, so that they can share identity and supporting information in a way that protects individual privacy as well as organizational interests. Thus, SPICI complements, and provides a usage context for,

¹ Identity safeguarding capabilities are organizational abilities to create, protect, share, use, and manage identity and/or authorization-related information in a way that safeguards individual privacy and protects organizational interests, using a combination of processes, technologies, tools, and models. For brevity, the term “capabilities” is used.

automated negotiation systems that can implement organizational agreements [2, 3].

SPICI provides a structure in which

- The concerns of stakeholders, including organizational users of identity information, individuals, and oversight bodies, are expressed as overarching goals and objectives for sharing identity and credential information in an appropriately protected manner.
- Identity safeguarding capabilities that organizations can use to manage and share identity, policy, and control information (particularly as represented by digital credentials) in a protected way are defined. These capabilities are motivated by related to the overarching goals of Unambiguous Identification, Assured Authentication, Accurate Authorization, Privacy Protection, and Accountable Trust and to the more specific objectives that derive from those goals.
- Four levels of assurance are defined for capabilities specifically related to sharing identity and credential information. The recommended identity safeguarding capability assurance level depends on organizational and privacy concerns.

More specifically, SPICI consists of a descriptive framework, an analytic framework, and a process. The SPICI descriptive framework identifies five goals for sharing identifying and authorization-related information (and the policy and control information needed to support that sharing), a set of objectives for technologies and/or business processes used for such sharing, and a set of capabilities which can be implemented in IT products and/or via business processes to achieve those goals. The SPICI descriptive framework also defines four capability levels (weak, basic, strong, and enhanced); these levels can be achieved by business processes and/or technologies (e.g., prototype tools, IT products). The SPICI analytic framework also defines levels of potential harms associated with sharing identifying and authorization-related information, and maps those levels of harm to capability levels. The SPICI process uses the descriptive and analytic frameworks to support negotiation of identity federation agreements, or of agreements for other forms of sharing identifying and authorization-related information.

2. BACKGROUND

Sharing of identity information, particularly in the form of easily propagated digital credentials, raises privacy as well as business concerns. The consequences to individuals of privacy violations range from minor embarrassment or inconvenience to identity theft, misdelivery of medical services leading to injury or death, or misapprehension by law enforcement. Sharing of identity and credential information also raises concerns for the organizations that

handle such information. Consequences to organizations can include damage to reputation or business relationships, as well as legal liability or financial costs associated with identity fraud or error.

To address these concerns, organizations must share more than identity or authorization-related information. They must also communicate their associated policies for using and protecting that information. For example, an organization that provides identity information might communicate its retention policy: how long shared identity information may be retained. To enable policy enforcement, the organization also needs to share control information (e.g., start date for the allowable retention period). Credentials can include policy and control information (e.g., period of validity), or such information can be shared using another mechanism.

Via negotiation, organizations determine what capabilities they will use to share identity- and authorization-related, and supporting policy and control, information. The set of technologies for managing identity information and credentials is large and growing. These are increasingly supported by technical, architectural, or assurance frameworks intended to facilitate specification and assessment of capabilities. However, these frameworks were not designed to facilitate analysis and negotiation. Furthermore, as discussions with stakeholder organizations have repeatedly highlighted, technical problems are frequently overshadowed by the challenges of establishing trust among organizations, by aligning policies and business processes.

SPICI takes into consideration the findings and recommendations of the Identity Theft Prevention and Identity Management Standards Panel (IDSP, [4]). SPICI is informed by existing identity management and identity federation frameworks, in particular the E-Authentication Guidance [5, 6], the Liberty Identity Assurance Framework (LIAF, [7]) and the framework [8] produced by the Focus Group on Identity Management (FG IdM) of the International Telecommunications Union – Telecommunication Standardization Sector (ITU-T).

The IDSP, E-Authentication, Liberty, and FG IdM work all define or rely upon an identity life cycle. The life cycle for identity and credential information defined by SPICI builds upon these high-level life cycles. However, SPICI considers identity- and authorization-related information (and supporting policy and control information) solely in digital form. Hence, SPICI does not consider issuance of foundational documents (e.g., birth certificates) or subsequent credentials in physical form (e.g., drivers licenses).

3. NEGOTIATING IDENTITY FEDERATION AGREEMENTS

Organizations that negotiate identity federation agreements (or other identity information sharing agreements, such as a Circle of Trust in the Liberty model or a simple bilateral agreement) need to address such topics as:²

- **Identity Assurance:** How confident can or should the federation or information sharing partners be that identified individuals actually exist and are distinctively identified; that credentials are correctly and securely managed and delivered; that attributes are meaningful and correct; and that identity and credential information is protected and accountable?
- **Trust Relationships:** How much trust does a given federation or sharing partner have in its credential service provider, in the federation as a trusted third party for the organizations in the federation, in another federation partner, in other federations in which the given organization participates, and in other federations in which other federation partners participate? To what extent do trust relationships depend on the business model and business rules (e.g., rules for compliance monitoring or auditing)?
- **Attributes:** What attributes are needed to make authorization-related decisions about identified individuals? What are the semantics of those attributes? How confident can or should the federation partners be in the quality (e.g., timeliness, correctness) of shared attributes?
- **Privacy and the Use of Personally Identifiable Information (PII):** How are the Fair Information Practice Principles³, as they apply to identifying or authorization-related information, interpreted within the federation or among the partners in identity information sharing? What legal and/or regulatory requirements apply to the identifying or authorization-related information? How confident are federation or sharing partners that these principles and regulatory requirements are interpreted and met consistently?
- **Technologies and Terminology:** What technical standards are used for identity data representation, for

policy representation and enforcement, for control data representation and use, and for communications between partner systems? What terminology is common to federation or other sharing partners, for roles and responsibilities; for identity and authorization-related attributes, policies, and control information; for information sensitivity and criticality; and for potential harms and corresponding risk mitigations?

- **Business Models and Business Rules:** How does the identity information sharing or identity federation relate to the business processes and models of the federation or sharing partners? In the case of a federation, how are the costs of operating the federation recovered? What service level agreements (SLAs) related to shared identifying or authorization-related information are needed to support the business processes of federation or sharing partners? What transparency standards apply, and how are they implemented (e.g., by sharing or review of audit trails)? What liability is accepted or disclaimed, and what enforcement procedures are implemented?

Answers to questions in these areas constrain the choice of technical solutions. The SPICI process uses the SPICI descriptive and analytic frameworks to provide a starting point for negotiating agreements in a way that makes explicit the concerns that drive or constrain the choice of technical solutions.

4. THE SPICI PROCESS

SPICI provides a descriptive and an analytic framework that organizations looking to form, join, or evolve an identity federation can use to identify issues about capabilities that they will need to negotiate, and to frame the discussion of those issues. The following paragraphs sketch the process for using SPICI to support analysis and negotiation. The process takes the form of a facilitated discussion, in five stages.

1. **Describe Business Process Needs.** Organizational representatives – typically the business process owners, and the technologists who support them – briefly describe the *business processes* that require sharing identifying or authorization-related information. They characterize or describe the *identifying or authorization-related* information they need or plan to share with each other. They indicate how they intend to *use* federated or shared identity or credential information. Examples of uses include: to identify an individual, to authenticate an asserted identity, to decide whether to provide a restricted service to an authenticated individual, to personalize the individual's interactions with their services, to market additional services to the individual, and to track an individual's

² This list of topics is derived from the Internet2 Federation Soup report [9], the Liberty Alliance legal framework for a Circle of Trust [10], and published examples of federation agreement templates.

³ Fair Information Practice Principles are “widely-accepted principles regarding the collection, use, and dissemination of personal information” [11]. While many different versions have been articulated, the principles usually address notice, consent or choice, access and redress, and security.

use of their services (for purposes of charging, to provide additional dynamic personalization, for auditing and compliance). They identify the types of other organizations with which they expect to *share* the information. They might also indicate how they plan to *protect* the information as it is sent to them, as they send it to another organization, and/or in storage. They might also indicate their expectations and plans about *accuracy or data quality*. (Thus, to a large extent, they answer the questions that would arise if they were drafting a Privacy Notice for the identity- and/or authorization-related information.)

2. **Assess Concerns or Potential Harms.** Organizational representatives then identify and assess the potential harms associated with the identity- or authorization-related information they provide to and/or get from the federation. Business process owners identify and assess potential harms related to their business processes; other organizational stakeholders identify and assess other potential harms. For example, a Chief Privacy Officer (CPO) might address the harms related to identified individuals, while a representative of the organization's Legal department might consider the harms related to unauthorized disclosure and to possible civil and criminal violations, and an organizational risk manager might consider financial and reputational harms. They might capture those assessments in table form, or using a spreadsheet or database. (Under the I3P project, a spreadsheet tool has been prototyped.) They could structure and normalize their assessments using the sets of threats and harms provided in the SPICI report. Note that harms can be assessed for different groupings of identity or authorization-related information; for example, if only one field in a credential is highly sensitive, it could be described and assessed separately from the rest of the credential. This lays the foundation for selective application of capabilities and technologies.
3. **Review Recommended Capabilities.** Based on these assessments, each organization will have a profile of recommended capabilities and levels. Many of those capabilities are defined by the LIAF. However, some capabilities are related specifically to the *sharing* of identifying or authorization-related information. SPICI defines six such capabilities, and (via the framework of goals, objectives, and capabilities discussed below) explains how they and the LIAF capabilities relate. The non-technical organizational representatives check that the recommendations make sense, and fine-tune the description or assessment as they deem appropriate to reflect their organizations' risk appetite.
4. **Negotiate Capability Implementation.** The technologists among the organizational representatives are now well positioned to start talking about how they

can implement (and assure their partners in the federation that they provide) the recommended capabilities. The definitions of the different assurance levels for the six SPICI-value-added capabilities can help guide those discussions, and help identify areas for more detailed negotiation and planning. In addition, SPICI identifies (but does not define assurance levels for) other capabilities; the negotiation can result in definitions of, and recommendations for, agreed-upon assurance levels of those additional capabilities.

5. **Negotiate Additional Considerations.** The organizational representatives must also define shared or cross-organizational processes (e.g., data correction, dispute resolution), legal and financial risk management, and (in the case of a federation) the federation business model. This negotiation, while crucial, is not the focus of SPICI.

5. SPICI DESCRIPTIVE FRAMEWORK

SPICI defines five overarching goals for sharing policy, identity, and control information:

- **Unambiguous Identification:** A user of identity or credential information (more specifically, a service provider, i.e., an entity that provides a service to individuals, such as an organization, a system, an application, or a Web service) should be able to distinguish one individual from another well enough to make decisions regarding and establish accountability for actual or attempted uses of services.
- **Assured Authentication:** A user of identity or credential information should be able to determine the identity or authorization-relevant attributes of an individual with assurance appropriate to the potential harms of misdelivered services.
- **Accurate Authorization:** A user of identity or credential information should be able to determine whether an individual is authorized to use that service with assurance appropriate to the potential harms of misdelivered services.
- **Privacy Protection:** A user of identity or credential information should handle the information regarding an individual that it maintains or uses with care sufficient to protect the individual's privacy. Privacy protection can be characterized in more detail by using the Fair Information Practice Principles.
- **Accountable Trust.** Identified individuals, and organizational providers and users of identifying and authorization-related information, should be able to explain defensibly (i.e., to give an account of) why they trust one another to the extent that they do.

Based on these goals, and on information life cycle models, SPICI defines a set of objectives, and capabilities

for achieving those objectives. Capabilities defined by other frameworks and initiatives (in particular, the LIAF) fit into this framework. SPICI currently defines six capabilities which are not addressed by other frameworks but are crucial to achieving the objectives.

For ***Unambiguous Identification***, the *objectives* and capabilities are:

- ***Identity Specification***: Within the scope of the identification problem, each individual can be uniquely characterized. Capabilities: Assignment of Subject Identifier / Attributes, Assignment of Service Provider Identifier / Attributes. These capabilities are well addressed by the LIAF.
- ***Identity Resolution*** : The unique characterization of an individual can be constructed (or reconstructed) from identifying and/or authorization-related information. Capabilities: Identity Attribute Correlation. As defined in SPICI, this is the capability to determine, to a specified or calculable degree of confidence, that presented credentials or sets of presented identifying information correspond to the same individual by correlating or matching presented values with known attributes, possibly by relying upon credentials issued using the LIAF.
- ***Initial Identity / Attribute Verification***: The association of identifying and/or authorization-related information with an individual is verified. Capabilities: In-Person Verification, Remote Evidence-Based Verification. These capabilities are well addressed by the LIAF.

For ***Assured Authentication*** (which could be more precisely called Assured Credential Authentication), the *objectives* and capabilities are:

- ***Credential Binding***: The credential is bound to the individual and/or to the transaction that the individual requests. Capabilities: Individual Binding, Transactional Binding.
- ***Credential Property Validation***: The expected properties of the credential are validated. Capabilities: Conditional Property Validation, Procedural Property Validation.
- ***Credential Status Validation***: The status of the credential (in particular, whether or not it has been revoked) can be determined. Capabilities: Conditional Status Validation, Procedural Status Validation.

For ***Accurate Authorization***, the *objectives* and capabilities are:

- ***Authorization Attribute Comprehensibility***: The intended meaning of attributes used to make authorization decisions can be discerned; the attributes cannot easily be misconstrued. Capabilities: Common Vocabulary, Mutual Understanding via Common

Attribute Syntax and Semantics. As defined in SPICI, Mutual Understanding is the capability to use syntactic and semantic rules (e.g., policy interpretation rules, rules for combining attributes) to provide a common understanding of well-founded uses of authorization-related information.

- ***Authorization Attribute Quality***: The quality (e.g., accuracy, currency) of attributes used to make authorization decisions can be determined. Capabilities: Attribute Quality Specification, Attribute Quality Assurance.
- ***Authorization Attribute Validation***: The validity of attributes used to make an authorization decision can be assessed in the context of the decision. Capabilities: Conditional Identity/Attribute Validation, Internal Attribute Validation. As defined in SPICI, Conditional Identity / Attribute Validation is the capability to validate, and make authorization decisions based on, authorization-related information using conditions that involve information not in the credential (e.g., history-based conditions such as Chinese Wall, location-based conditions, time-based conditions, conditions asserted by credential issuers).

For ***Privacy Protection***, the *objectives* and capabilities are:

- ***Notice and Consent***: Identified individuals are provided with notice of the intended and expected uses of identifying information, and are given the opportunity to consent to uses as appropriate. Capabilities: Notice, Consent.
- ***Usage Restriction***: Uses of identity and credential information are restricted to those to which identity information providers and identified individuals consent. Capabilities: Agreement on Terms of Use, Conditional Use. As defined in SPICI, Agreement on Terms of Use is the capability to validate agreement to the terms of use for shared identity and authorization-related information as a precondition for sharing it.⁴
- ***Disclosure / Exposure Restriction***: Identity and credential information is disclosed, or exposed to observation, only in accordance with restrictions to which credential providers and identified individuals consent. Capabilities: Selective Disclosure/Retrieval

⁴ *Terms of use* for information are statements about restrictions and obligations applicable to any individual or organization that handles the information. Terms of use can include how the information may or may not be used (e.g., for what purposes, in combination with what other information, for how long), with whom else the information may or may not be shared, how the information must be protected, and what accountability for using or sharing the information is needed.

Protection, Onward Transfer Restriction, Transmission Protection. As defined in SPICI, Selective Disclosure / Retrieval Protection is the capability to provide in, or derive from, credentials only such identity or authorization-related information as is required for agreed-upon business processes.

- *Retention Restriction*: Retention of identity and credential information is restricted to the duration and conditions to which credential providers and identified individuals consent. Capabilities: Conditional Retention Restriction, Procedural Retention Restriction.

For *Accountable Trust*, the *objectives* and capabilities are:

- *Policy Specification and Enforcement*: The policies related to the collection, use, sharing, retention, maintenance, and destruction of identity-related information are transparent and effective. Capabilities: Policy Specification, Policy Enforcement.
- *Trust Specification*: Users of identity and credential information are able to state the extent to which they can or wish to trust its source. Capabilities: Trust Designation, Trust Assessment, Trust Accreditation.
- *Accountability*: Organizations and individuals are accountable for their handling and use of identity and credential information. Capabilities: Accountability for Creation/Collection, Accountability for Use, and Accountability for Disclosure/Sharing, which is the capability to provide accountability for the onward transfer⁵ of identity or authorization-related information.
- *Recourse*: Organizations that handle or use identity or credential information provide recourse processes to individuals and/or oversight bodies. Capabilities: Access/Correction Process, Violation/Non-Compliance Recourse.

SPICI defines assurance levels for the six capabilities it defines (Identity Attribute Correlation, Mutual Understanding, Conditional Identity/Attribute Validation, Agreement on Terms of Use, Selective Disclosure/Retrieval Protection, and Accountability for Disclosure/Sharing). Higher levels give greater confidence that the corresponding goals and overarching objectives will be achieved. SPICI capability levels are defined consistent with existing frameworks, including the LIAF and the E-

⁵ Onward transfer is the transfer or disclosure of personal information to an additional party that did not collect or create that information and that is not acting on behalf of the party that collected or created that information. [12]

Authentication Guidance.⁶ The SPICI-defined capabilities complement those defined by other frameworks, and thereby fill some gaps in those frameworks related to sharing. The SPICI-defined capabilities do not fill all gaps; however, SPICI is easily extensible to include additional capabilities.

6. SPICI ANALYTIC FRAMEWORK

SPICI identifies potential harms associated with sharing identity or credential information, consistent with the E-Authentication Guidance. Based on the level of harm (minimal, moderate, substantial, or high),⁷ SPICI recommends capability assurance levels. This is illustrated for Identity Attribute Correlation, i.e., for the capability to determine an individual's identity based on identifying or authorization-related attributes, which may come from different credentials. This capability involves either correlation and aggregation of identity and credential information or reliance on a unique identifier [14]. This capability is needed when credential tokens are validated, information from them is extracted, and local stores of shared identity or credential information are updated, so that the individual can be identified unambiguously. The four levels of assurance for this capability are:

- *Weak*: A service provider (i.e., an entity that provides goods or services to an individual) can have little or no confidence that shared identity and/or credential information refers to a specific individual. Shared identity and/or credential information is associated with information that the service provider maintains based on a single attribute that can frequently be confused or conflated, e.g., name.
- *Basic*: A service provider can have some confidence that shared identity and/or credential information refers to a specific individual. Shared identity and/or

⁶ The E-Authentication Guidance and the LIAF define assurance levels that apply to both Unambiguous Identification and Assured Authentication, and maps potential harm levels to assurance levels. Community acceptance of this lack of granularity is possible largely because of a relatively large common experience in the use of credential processes. However, less experience has been captured, and less consensus can be expected, regarding the set of capabilities that enable *sharing* of identity or credential information. Thus, SPICI provides more granularity: levels are defined for different capabilities, and potential harms are mapped not to a bundle of capabilities but to each capability.

⁷ These levels of harm are largely identical to those in the E-Authentication Guidance. SPICI provides additional detail for harms to individuals. When those harms are associated with unauthorized disclosure of personal information, the SPICI definitions of moderate, substantial, and high levels of harm are consistent with the examples of low, moderate, and high confidentiality impacts in the draft NIST guide [13].

credential information is associated with information that the service provider maintains based on multiple attributes (e.g., name plus phone number, name plus account identifier), or on a single attribute that is intended to be unique (e.g., SSN, driver’s license number).

- Strong: A service provider can have high confidence that shared identity and/or credential information refers to a specific individual. Shared identity and/or credential information is associated with information that the service provider maintains based on multiple attributes that comply with an agreed-upon specification, or on a single attribute that the service provider and the entity that shared the information accept as unique (e.g., identity credential supplied by an agreed-upon Assurance Level 3 Credential Service Provider).
- Enhanced: A service provider can have very high confidence that shared identity and/or credential information refers to a specific individual. Shared identity and/or credential information is associated with information that the service provider maintains based on selective retrieval or evaluation of multiple attributes that comply with an agreed-upon specification (e.g., age range rather than date of birth), based on rigorous methods (e.g., statistical methods), or on a single attribute that the service provider and the entity that shared the information have very high confidence in as unique (e.g., identity credential supplied by an agreed-upon Assurance Level 4 Credential Service Provider).

Table 1 presents the mapping from levels of potential harm to the recommended level for Identity Attribute Correlation. For example, if an organization’s potential harm from unauthorized release of sensitive information, due to misidentifying an individual and granting them more privileges than they are entitled to, is high, then the recommended level of Identity Attribute Correlation is Strong. If the potential physical harm to an individual, for example due to medical mistreatment based on misidentification associated with an incorrect attribute, is substantial, then the recommended level is Enhanced. The overall recommended level is the maximum of the recommendations across all stakeholders.

Identity attribute correlation or matching that does not rely upon a unique identifier is an active research area at the Enhanced level [15, 16, 17].

7. CONCLUSION

Organizations that share identity or credential information can use SPICI as part of their process of negotiating identity federation or other identity information sharing agreements. The organizations can determine which capabilities are relevant to their joint and separate

business processes, assess their respective concerns, and determine what capability levels they require or can achieve. The organizations can thus reach agreement on how they each provide the relevant capabilities – on the processes and mechanisms they will use to achieve the five objectives. Under the I3P project, a spreadsheet tool has been prototyped.

Table 1. Recommended Level of Identity Attribute Correlation

Potential Harm to Service Provider	Level of Harm				
	<i>N/A</i>	<i>Min</i>	<i>Mod</i>	<i>Sub</i>	<i>High</i>
Inconvenience, distress or damage to standing or reputation	Weak	Weak	Basic	Strong	Enh.
Financial loss or liability	Weak	Weak	Basic	Strong	Enh.
Harm to organizational programs or interests	Weak	Weak	Basic	Strong	Enh.
Sensitive information breach	Weak	Weak	Weak	Basic	Strong
Civil or criminal violations	Weak	Weak	Basic	Strong	Enh.
Potential Harm to Individual	<i>N/A</i>	<i>Min</i>	<i>Mod</i>	<i>Sub</i>	<i>High</i>
Social harms	Weak	Weak	Basic	Strong	Enh.
Physical harm or distress	Weak	Basic	Strong	Enh.	Enh.
Financial harms	Weak	Weak	Weak	Basic	Strong

In addition, an organization that handles identity or credential information can use SPICI to manage risks, by identifying gaps in current or planned capabilities vis-à-vis recommended assurance levels. An identity management or federation solution provider can use SPICI to profile product capabilities. Finally, researchers can use SPICI to identify capability gaps as research areas. The work being performed or leveraged as part of the I3P Safeguarding Digital Identity project aligns with the six capabilities currently defined in SPICI, and in general will produce enhanced capabilities:

- A service—VeryIDX [15]—that facilitates trust negotiations across organizations that wish to share digital identities, and an attribute trust framework [18], which address Identity Attribute Correlation and Conditional Identity / Attribute Validation.
- Minimum-Disclosure Credentials [19], Attribute-Based Messaging [20], Attribute-Based Encryption [21, 22],

Zero-Knowledge Identity Federation [23] and Privacy-Preserving Distributed Queries address Selective Disclosure / Retrieval Protection in a variety of contexts.

- Enabling Web Services for Federated Digital Identities (integrated into a demonstration being developed under the I3P project) address Identity Attribute Correlation and Mutual Understanding via Common Attribute Syntax and Semantics.

The SPICI analytic framework is extensible; future versions of SPICI could include definitions of, and recommendations for, additional capabilities. For example, Trust Calculus [24], also being developed under the I3P project, addresses Trust Assessment.

8. ACKNOWLEDGMENTS

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. The views and conclusions contained in this document are those of the author and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

9. BIBLIOGRAPHY

- [1] I3P, Safeguarding Digital Identities Overview, 2008, <http://www.thei3p.org/docs/research/idmgmtoverview.pdf>
- [2] Bhargav-Spantzel, Abhilasha, Squicciarini, Anna C., Bertino, Elisa, Trust Negotiation in Identity Management, *IEEE Security and Privacy*, March/April 2007.
- [3] Gateau, B., Feltus, C., Aubert, J., Incolt, C., An agent-based framework for identity management: The unsuspected relation with ISO/IEC 15504, in *Research Challenges in Information Science, 2008*. ISBN: 978-1-4244-1677-6. DOI: 10.1109/RCIS.2008.4632091
- [4] American National Standards Institute-Better Business Bureau (ANSI-BBB) Identity Theft Prevention and Identity Management Standards Panel (IDSP), Final Report Volume I: Findings and Recommendations, 31 January 2008, <http://publicaa.ansi.org/sites/apdl/ID%20Theft%20Prevention%20and%20ID%20Management%20Standards%20Pa/IDSP%20Final%20Report%20-%20Volume%20I%20Findings%20and%20Recommendations.pdf>
- [5] Office of Management and Budget (OMB), E-Authentication Guidance for Federal Agencies, OMB Memorandum 04-04, 13 December 2003, <http://www.whitehouse.gov/omb/memoranda/fy04/m04-04.pdf>
- [6] National Institute of Standards and Technology (NIST), Electronic Authentication Guideline, NIST Special Publication (SP) 800-63, Version 1.0.2, April 2006, http://csrc.nist.gov/publications/nistpubs/800-63/SP800-63V1_0_2.pdf and Draft Revision 1, December 2008, <http://csrc.nist.gov/publications/PubsDrafts.html#SP-800-63-Rev.%201>
- [7] Liberty Alliance Project, Liberty Identity Assurance Framework, Version 1.1, June 2008, <http://www.projectliberty.org/liberty/content/download/4315/28869/file/liberty-identity-assurance-framework-v1.1.pdf>
- [8] ITU-T FG IdM, Report on Identity Management Framework for Global Interoperability, Study Group 17 Temporary Document 0297 (FG IdM Doc 196), <http://wftp3.itu.int/fgidm/Deliverables/0297-att-1.doc>
- [9] Internet2, Federation Soup: An Assembly of Ingredients, Proceedings of the Federation Soup Workshop, held 2-4 June 2008 in Seattle, WA, 7 September 2008, http://middleware.internet2.edu/fedsoup/docs/internet2-fed_soup_report-200809.pdf
- [10] Liberty Alliance Project, Liberty Alliance Contractual Framework Outline for Circles of Trust, March 2007, <http://www.projectliberty.org/liberty/content/download/2962/19808/file/Liberty%20Legal%20Frameworks.pdf>
- [11] Federal Trade Commission (FTC), Privacy Online: Fair Information Practices in the Electronic Marketplace: A Federal Trade Commission Report to Congress, May 2000, <http://www.ftc.gov/reports/privacy2000/privacy2000.pdf>
- [12] Connolly, K. *Law of Internet Security and Privacy (2004 Edition)*, Aspen Publishers Online, ISBN 0735542732, 9780735542730
- [13] NIST, Guide to Protecting the Confidentiality of Personally Identifying Information (PII) (DRAFT), NIST SP 800-122 (Draft), <http://csrc.nist.gov/publications/drafts/800-122/Draft-SP800-122.pdf>
- [14] National Alliance for Health Information Technology (NAHIT), Safety in Numbers: Resolving shortcomings in the matching of patients with their electronic records, Point of View Paper #1, December 2007, <http://www.nahit.org/images/pdfs/PatientIdentifierPointofView.pdf>

- [15] Bhargav-Spantzel, A., Jungha Woo, Bertino, E. Receipt Management- Transaction history based receipt management, *Workshop On Digital Identity Management, Proceedings of the 2007 ACM workshop on Digital identity management*, November 2007, pages: 82 – 91, <http://homes.cerias.purdue.edu/~bhargav/pdf/ReceiptDIM07.pdf>
- [16] Language Resources and Evaluation Conference, *Proceedings of the 2008 Workshop on Resources and Evaluation for Identity Matching, Entity Resolution and Entity Management*, 31 May 2008, <http://www.lrec-conf.org/proceedings/lrec2008/>
- [17] Hatakeyama, Makoto, Shima, Shigeyoshi. Privilege federation between different user profiles for service federation. *Proceedings of the 4th ACM workshop on Digital identity management*, November 2008. DOI=<http://doi.acm.org/10.1145/1456424.1456432>
- [18] Mohan, A., and Blough, D. "AttributeTrust: A Framework for Evaluating Trust in Aggregated Attributes via a Reputation System," *Proceedings of the Conference on Privacy, Security, and Trust*, 2008.
- [19] Bauer, D., Blough, D., and Cash, D. "Minimal Information Disclosure with Efficiently Verifiable Credentials." *Proceedings of the 4th ACM Workshop on Digital Identity Management*, November 2008.
- [20] Bobba, R., Fatemeh, O., Khan, F., Gunter, C., and Khurana, H. Using Attribute-Based Access Control to Enable Attribute-Based Messaging, *IEEE Annual Computer Security Applications Conference (ACSAC '06)*, Miami, FL, December 2006.
- [21] Goyal, V., Pandey, O., Sahaiz, A., and Waters, B. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. *Proceedings of the ACM conference on Computer and Communications Security*, 2006.
- [22] Bethencourt, J., Sahai, A., and Waters, B. 2007. Ciphertext-Policy Attribute-Based Encryption. In *Proceedings of the 2007 IEEE Symposium on Security and Privacy* (May 20 - 23, 2007). SP. IEEE Computer Society, Washington, DC, 321-334. DOI=<http://dx.doi.org/10.1109/SP.2007.11>
- [23] Bhargav-Spantzel, A., Squicciarini, A. C., and Bertino, E. 2006. Establishing and protecting digital identity in federation systems. *J. Comput. Secur.* 14, 3 (May 2006), 269-300.
- [24] Huang, J. and Nicol, D. 2009. A Calculus of Trust and Its Applications to PKI and Identity Management. *Proceedings of IDtrust 2009*, April 2009.



I3P
Institute for Information
Infrastructure Protection

Safeguarding Digital Identity: The SPICI (Sharing Policy, Identity, and Control Information) Approach to Negotiating Identity Federation and Sharing Agreements

Deb Bodeau
The MITRE Corporation
dbodeau@mitre.org
Presented at IDtrust 2009

This material is based upon work supported by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. 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 the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

<http://www.thei3p.org>




© 2009 The MITRE Corporation; all rights reserved
Approved for Public Release; Distribution Unlimited
Case Number 09-1056

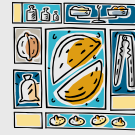
Outline

- Introduction
 - SPICI in a Nutshell
 - Intended Use
 - Descriptive Framework
- Related Work
- SPICI Process
- Risk Functions
- Conclusion

2

I3P 

SPICI in a Nutshell



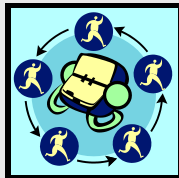
- SPICI provides a structure for clarifying business needs and supporting negotiation on risk-appropriate solutions when organizations share identity (and supporting policy and control) information
 - Includes a descriptive framework of goals, objectives, and capabilities
 - Includes an analytic framework which maps levels of stakeholder concerns to recommended levels of capabilities
 - Includes a high-level description of the negotiation process
- SPICI complements technical frameworks and reference implementations for identity federation
 - Technical frameworks identify “who, what, and where”
 - Reference implementations demonstrate “how”
 - SPICI helps organizations analyze “when, why, and how much”

3

I3P

MITRE

SPICI Intended Use



Our organizations need to share identity-related information to support our interrelated business processes



How do we think about this?

Key Concepts and Definitions
 Information Sensitivity
 Terms of Use
 Information Life-Cycle

What could go wrong?

Potential Harms / Concerns
 Damage to reputation or relationships
 Financial loss or liability
 Damage to business or organization
 Unauthorized release of sensitive information
 Civil or criminal violations
 Impacts on individuals – social, financial, physical

How do we decide which controls need to be in place?

Risk Functions
 General Risk Model (aligned with Liberty)
 Control- or Practice-Specific Functions
 Presented as Simple Tables and in Spreadsheet Form to Facilitate Discussion

SPICI gives us a basis for negotiating about technical and procedural controls ... avoiding overkill, but managing our risks

4

I3P

MITRE



- ## Related Work
- Identity Assurance
 - Liberty Identity Assurance Framework (LIAF)
 - E-Authentication Guidance (OMB 04-04)
 - NIST SP 800-63, Electronic Authentication Guideline (v. 1.0.2, April 2006; draft revision, December 2008)
 - NIST SP 800-103 (Draft), An Ontology of Identity Credentials
 - Identity Framework for Global Interoperability, Focus Group on Identity Management of the ITU
 - Identity Federation
 - Liberty Framework Outline for Circles of Trust
 - Internet2 Federation Soup
 - Privacy
 - NIST SP 800-122 (Draft), Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)
 - Identity Management Technology Research & Development
 - Technical Frameworks (architectures, specifications, reference implementations)
 - Specific Technologies
- 6
- I3P MITRE

SPICI Process

- Goal: Enable organizations that need to share credential or other identity- or authorization-related information to agree on the processes, procedures, and technologies they will use
 - Organizations need to protect their business interests
 - Identified individuals need assurance that their privacy is protected
- To share the credential information, organizations also need to share policy and control information
 - Example of policy information: Use this credential for no more than 6 months after it was issued
 - Example of control information: This credential was issued on 14 April 2009

I3P

MITRE

SPICI Process

1. Describe business process needs
 - What credential, identity, or authorization-related information will be shared?
 - How is credential, identity, or authorization-related information used in business processes?
 - What supporting policy and/or control information is needed?
 2. Assess concerns or potential harms
 - To the organizations
 - To individuals
 3. Evaluate risk functions to obtain recommendations for appropriate capability levels
 - Liberty Identity Assurance Level (includes multiple capabilities)
 - SPICI-defined capabilities
 4. Review recommendations and negotiate mutually acceptable implementation mechanisms
 5. Negotiate additional considerations
- Results of process
 - Agreements on business rules
 - Agreements on standards and technologies to be used

I3P

MITRE

8

Assessing Potential Harms

Harms to Organizational Providers and Users of Shared Identity / Credential Information

- Inconvenience, Reputation Damage
- Financial Loss or Liability
- Organizational Harms
- Unauthorized Release of Sensitive Information
- Criminal or Civil Violations

Definitions of types and levels of harms consistent with LIAF and E-Authentication Guidance

Harms to Individuals

- Social Harms Due to Misuse, Unauthorized Disclosure, Inaccuracy
- Physical Harms Due to Misuse, Unauthorized Disclosure, Inaccuracy
- Financial Loss or Liability due to Identity Theft

Definitions of types and levels of harms consistent with FIPS 199 and draft NIST SP 800-122

Harms are assessed as N/A, Minimal, Moderate, Substantial, or High

9

I3P

MITRE

SPICI-Defined Capabilities

- Identity Attribute Matching
 - Objective: Resolve Identities
- Conditional Identity / Attribute Validation
 - Objective : Validate Attributes
- Mutual Understanding
 - Objective : Make Attributes Understandable
- Selective Disclosure / Retrieval Protection
 - Objective : Restrict Disclosure / Exposure
- Assertion of Terms of Use
 - Objective : Restrict Uses
- Accountable Disclosure / Sharing
 - Objective : Provide Accountability

For each capability, SPICI defines levels (weak, basic, strong, and enhanced) which can be achieved via technical and/or procedural means

10

I3P

MITRE

SPICI Risk Functions

- For each type of potential harm, the level of harm due to not achieving the overarching goals (which has been assessed for the identity information to be shared) maps to a risk-appropriate capability level
- The recommended capability level is the maximum of the levels appropriate to the different types of harms
- Example: Risk function to determine recommended level of Identity Attribute Correlation

Potential Harm to Service Provider	Level of Harm				
	N/A	Min	Mod	Sub	High
Inconvenience, distress or damage to standing or reputation	Weak	Weak	Basic	Strong	Enh.
Financial loss or liability	Weak	Weak	Basic	Strong	Enh.
Harm to organizational programs or interests	Weak	Weak	Basic	Strong	Enh.
Unauthorized release of sensitive information	Weak	Weak	Weak	Basic	Strong
Civil or criminal violations	Weak	Weak	Basic	Strong	Enh.
Potential Harm to Individual	N/A	Min	Mod	Sub	High
Social harms	Weak	Weak	Basic	Strong	Enh.
Physical harm or distress	Weak	Basic	Strong	Enh.	Enh.
Financial harms	Weak	Weak	Weak	Basic	Strong

11

I3P

MITRE

Conclusion

- SPICI provides
 - An extensible framework which clarifies the role of specific technologies or processes in meeting objectives by providing capabilities
 - A process for negotiating identity information sharing agreements
 - A set of risk functions to support that process
- Investigation into venues for application and standardization is ongoing

I3P Work in the SPICI Framework

- VeryIDX addresses Identity Attribute Correlation and Conditional Identity / Attribute Validation
- Minimum-Disclosure Credentials, Attribute-Based Messaging, Attribute-Based Encryption, Zero-Knowledge Identity Federation and Privacy-Preserving Distributed Queries address Selective Disclosure / Retrieval Protection in a variety of contexts
- Enabling Web Services for Federated Digital Identities address Identity Attribute Correlation and Mutual Understanding via Common Attribute Syntax and Semantics
- Trust Calculus addresses Trust Assessment

12

I3P

MITRE

Backup

13

I3P

MITRE

SPICI Overview

SPICI Potential Harms and Levels

- Harms to Organizational Providers and Users of Shared Identity / Credential Information
 - Inconvenience, Reputation Damage
 - Financial Loss or Liability
 - Organizational Harms
 - Unauthorized Release of Sensitive Information
 - Criminal or Civil Violations
- Harms to Individuals
 - Social and Physical Harms Due to Misuse, Unauthorized Disclosure, Inaccuracy
 - Financial Loss or Liability due to Identity Theft

Mapping of Harm Levels to Recommended Capability Levels

SPICI Goals

- Unambiguous Identification
- Assured Authentication
- Accurate Authorization
- Privacy Protection
- Accountable Trust

SPICI Objectives, Capabilities, and Capability Levels

SPICI Life-Cycle Models

14

I3P

MITRE

SPICI Goals

Unambiguous Identification

A user of identity or credential information should be able to distinguish one individual from another well enough to make decisions regarding and establish accountability for actual or attempted uses of services.

Assured Authentication

A user of identity or credential information should be able to determine the identity or authorization-relevant attributes of an individual with assurance appropriate to the potential harms of misdelivered services.

Accurate Authorization

A user of identity or credential information should be able to determine whether an individual is authorized to use that service with assurance appropriate to the potential harms of misdelivered services.

Privacy Protection

A user of identity or credential information should handle the information regarding an individual that it maintains or uses with care sufficient to protect the individual's privacy. Privacy protection can be characterized in more detail by using the Fair Information Practices Principles.

A user of identity or credential information, also referred to as a service provider, is an entity that provides a service to individuals, such as an organization, a system, an application, or a Web service.

Accountable Trust

Identified individuals, and organizational providers and users of identifying and authorization-related information, should be able to explain defensibly (i.e., to give an account of) why they trust one another to the extent that they do.

15

I3P

MITRE

SPICI-Defined Capabilities

SPICI Goals & Objectives

- Unambiguous Identification
 - *Specify Identities*
 - Resolve Identities
 - *Verify Initial Assignments*
- Assured Authentication
 - *Bind Credentials*
 - *Validate Credential Properties*
 - *Validate Credential Status*
- Accurate Authorization
 - Make Attributes Understandable
 - Ensure Attribute Quality
 - Validate Attributes
- Privacy Protection
 - Notice and Consent
 - Restrict Uses
 - *Restrict Disclosure / Exposure*
 - *Support Redress*
- Accountable Trust
 - *Policy Specification and Enforcement*
 - *Specify Trust*
 - *Provide Accountability*
 - *Provide Recourse*

SPICI Capabilities

- Identity Attribute Correlation
 - Objective: Resolve Identities
- Conditional Identity / Attribute Validation
 - Objective : Validate Attributes
- Mutual Understanding
 - Objective : Make Attributes Understandable
- Selective Disclosure / Retrieval Protection
 - Objective : Restrict Disclosure / Exposure
- Assertion of Terms of Use
 - Objective : Restrict Uses
- Accountable Disclosure / Sharing
 - Objective : Provide Accountability

Note: Depending on results of project & stakeholder review, capability definitions could change; more capabilities could be added.

Key:

- *Italics + Underline = Capabilities to meet this goal are largely defined by LIAF*
- *Italics = Capabilities to meet this goal are partially defined by LIAF*

16

I3P

MITRE

Example: Motivating Scenario

- Dr. Alpha orders Test X for Patient Beta
- Test X requires use of Drug X-Static (a controlled substance)
- Dr. Alpha practices at Clinic Gamma
- Test X must be performed at Laboratory Delta

Credential:

- Ordering Physician: Hippocrates Alpha
- Physician Medical License Number: mmmmmm
- Physician Affiliation: Group Practice Gamma
- Ordering Physician's DEA Number or National Provider Identifier: CAnnnnnn
- Credential Issuer: Gamma-Cred
- Credential Issuance Date / Time: mm/dd/yy/TOD
- Other Credential Information
 - Could include obligations (e.g., encrypt digitally stored copies of DEA Number or NPI)

17

I3P

MITRE

Example: Assessment of Potential Harms

Potential Harm to User or Provider of Shared Identity / Credential Information	Assessment
	The scenario of concern involves a criminal (possibly an insider at Lab Delta) misusing Alpha's NPI or DEA number to obtain or prescribe vast amounts of X-Static.
Inconvenience, distress or damage to standing or reputation	Moderate: If Lab Delta does not protect this information, the relationship between Delta and Gamma could be damaged
Financial loss or liability	Minimal
Harm to organizational programs or interests	Moderate: Delta and/or Gamma operations could be disrupted during an investigation
Unauthorized release of sensitive information	Minimal: The NPI or DEA # is not used to authorize access to sensitive patient information
Civil or criminal violations	Moderate (or Substantial, if part of a pattern of abuse)
Potential Harm to Individual (Physician)	Assessment
Social harms (Inconvenience, embarrassment, distress, or damage to personal standing or reputation)	Moderate: Dr. Alpha could be perceived as irresponsible
Physical harms (including detention or imprisonment)	Substantial: Dr. Alpha could be arrested or detained
Financial loss or liability due to identity theft	Minimal or N/A

18

I3P

MITRE

Example: Assertion of Terms of Use

Terms of use for information are statements about restrictions and obligations applicable to any individual or organization that handles the information. Terms of use can include how the information may or may not be used (e.g., for what purposes, in combination with what other information, for how long), with whom else the information may or may not be shared, how the information must be protected, and what accountability for using or sharing the information is needed. Terms of use can also include obligations regarding accuracy and correction processes.

19

I3P

MITRE

Example: Assertion of Terms of Use (Continued): Definitions of Capability Levels

Level	Definition
Weak	The individual, and the entity that shares identity and/or credential information, has little or no confidence that the user of the shared information understands and accepts the terms of use for that information. Typically, this lack of confidence is because terms of use for shared identity and/or credential information are not asserted clearly or explicitly to the user of the shared information. While terms of use may be asserted to the individual (e.g., via a Privacy Notice), this assertion is not communicated to users of shared identity and/or credential information.
Basic	The individual, and the entity that shares identity and/or credential information, can have some confidence that the user of the shared information understands and accepts the terms of use for that information. Restrictions and obligations are communicated informally.
Strong	The individual, and the entity that shares identity and/or credential information, can have high confidence that the user of the shared information understands and accepts the terms of use for that information. Restrictions and obligations are stated explicitly and formally agreed (e.g., via a contract or Memorandum of Agreement). Redress processes are defined.
Enhanced	The individual, and the entity that shares identity and/or credential information, can have very high confidence that the user of the shared information understands and accepts the terms of use for that information. Restrictions and obligations are stated explicitly and formally agreed (e.g., via a contract or Memorandum of Agreement). Redress processes and compliance processes or mechanisms (e.g., auditing processes) are agreed. As feasible, enforcement mechanisms (e.g., Digital Rights Management controls, selective disclosure) are agreed.

20

I3P

MITRE

Example: Assertion of Terms of Use (Continued): Recommended Capability Levels

Potential Harm to Service Provider	Level of Harm				
	N/A	Min	Mod	Sub	High
Inconvenience, distress or damage to standing or reputation	Weak	Weak	Weak	Basic	Strong
Financial loss or liability	Weak	Weak	Basic	Strong	Enhanced
Harm to organizational programs or interests	Weak	Weak	Basic	Strong	Enhanced
Unauthorized release of sensitive information	Weak	Weak	Weak	Basic	Strong
Civil or criminal violations	Weak	Weak	Basic	Strong	Enhanced
Potential Harm to Individual	N/A	Min	Mod	Sub	High
Social harms	Weak	Weak	Basic	Strong	Strong
Physical harm or distress	Weak	Weak	Basic	Strong	Enhanced
Financial harms	Weak	Weak	Basic	Strong	Strong

21

I3P

MITRE

SPICI Uses

- Organizations that share identity (and supporting policy and control) information can use SPICI to establish business trust
 - Determine which capabilities are relevant to their joint and separate business processes
 - Reach agreement on how they each provide the relevant capabilities – on the processes and mechanisms they will use to achieve the four objectives
- An organization that handles identity information can use SPICI to manage risks (gaps in current / planned capabilities vis-à-vis recommended levels)
- An identity management / federation solution provider can use SPICI to profile product capabilities
- Researchers can use SPICI to identify capability gaps as research areas

22

I3P

MITRE

Usable Trust Anchor Management*

Massimiliano Pala
Department of Computer Science
Dartmouth College, Hanover, NH
pala@cs.dartmouth.edu

Scott A. Rea
Institute for Security, Technology, and Society
Dartmouth College, Hanover, NH
Scott.A.Rea@dartmouth.edu

ABSTRACT

Security in browsers is based upon users trusting a set of root Certificate Authorities (called Trust Anchors) which they may know little or nothing about. Browser vendors face a difficult challenge to provide an appropriate interface for users. Providing usable Trust Anchor Management (TAM) for users, applications and PKI deployers is a complex task. The PKIX working group at Internet Engineering Task Force (IETF) is working on a new protocol, the Trust Anchor Management Protocol (TAMP), which will provide a standardized method to automatically manage trust anchors in applications and devices. Although promising, this protocol does not go far enough to allow users to gather information about previously unknown trust anchors in an automatic fashion. We have proposed the PKI Resource Query Protocol (PRQP)—which is currently an Internet Draft on Experimental Track with IETF—to provide applications with an automatic discovery system for PKI management. In this paper we describe the basic architecture and capabilities of PRQP that allow Browsers to provide a more complete set of trust anchor management services. We also provide the design of a PRQP enabled infrastructure that uses a trust association mechanism to provide an easy solution for managing Trust Anchors for Virtual Organizations.

Categories and Subject Descriptors

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

General Terms

Security

*This work was supported in part by the NSF (under grant CNS-0448499), the U.S. Department of Homeland Security (under Grant Award Number 2006-CS-001-000001), and Sun. 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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDtrust '09 April 14-16, 2009, Gaithersburg, MD
Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

Keywords

Trust Anchor, PRQP, Discovery System, Digital Certificate, PKI

1. INTRODUCTION AND MOTIVATIONS

Browser—based trust decisions are facilitated by a set of Trust Anchors (TA) that come preloaded in the browser or via an operating system based Trust Anchor Store (TAS) that the browser relies upon. These TAS contain many trust anchors that the average user has no idea in respect to their purpose or community of applicability, yet the browser makes trust decisions on behalf of those users based on the presence and configuration of these elements.

At the NIST PKI Workshop in Gaithersburg, MD in 2007 [10], Kelvin Yiu from Microsoft made the revelation that browsers were struggling with the growing size of the TAS—he indicated that Microsoft's browsers were originally designed with a TAS of approximately 100 elements in mind, with an upper limit of 200. As PKI communities continue to proliferate, it was plainly apparent that the current processes of managing the TAS primarily at bootstrap and relying upon users to tweak thereafter was not going to be sufficient [19]. As of today, a simple analysis of the default TA stores in Operating Systems (OSes) and browsers reveals 132 TA in Firefox, 152 TA in OSX, 286 in XP, and (obviously heading in the right direction) just 30 TA in Vista.

A simple task like path construction and validation in hierarchical PKIs still raise many issues today that are not completely solved thus impacting on the operation and management of large-scale PKIs. Most end-user applications today still rely exclusively on trust lists. For example, some of the most widely deployed network applications such as browsers or Mail User Agents (MUAs) use Trust-Lists as a basis for trust. As described in [15], trust lists build trust by embedding digital certificates locally into applications. This approach leads users to completely rely on the application vendor for management of her own Trust Anchors without being actively involved in deciding what is trustworthy and for what purpose. On the other end, the application vendor is forced to manage a quite large set of trust anchors— i.e. the ones that are embedded into the shipped application.

Another important aspect to consider is the low level of users' awareness about trust management [5]: in general users do not understand protection technologies and poor user interfaces exacerbate the problem. The authors practi-

cal experiences participating on the EuroPKI [1, 14] project as well as being reported in other realities [16] show that users often require specialized help and support in order to correctly add certificates to the ones trusted by default in a given application. Therefore using pre-cooked TA stores encourages people to accept by “faith” whatever is hardwired into applications.

There are also environments, such as Grid Computing communities [2–4], that are very sensitive about TAM. In these communities interoperability is extremely important and PKI administrators and application developers struggle with problems related to the lack of flexibility in trust anchor management.

Our work addresses trust management issues from a practical point of view by providing a model which both helps in removing the reliance upon pre-installed trust lists, and provides a local trust management system by using cross-certification. The purpose of our work is to let a single organization (be it a company or an aggregate such as a research network or a Virtual Organization) to unilaterally create trust for its users of selected foreign PKIs or CAs in a way that is easily supported by current applications.

The rest of the paper is organized as follows. Section 2 presents the related work and current limitations. Section 3 introduces an overview of the possibilities offered by a dynamic TA management system. Section 4 describes our new hybrid trust model. Section 5 explains how to integrate our hybrid trust model with PRQP in order to provide a dynamic TA management system. Section 6 contains our conclusions and future work.

2. RELATED WORK

Since the 1976 paper by Diffie and Hellman [22] and the introduction of a public key cryptosystem [18], the presence of some sort of trusted directory for public key access has been a steady reality in PKIs. The very nature of the trusted directory strictly depends on the trust model used in the infrastructure. In fact the organization of a PKI reflects the trust model required by its constituency. In X.509-based PKIs (or PKIX), there are three different “pure” trust models: hierarchical, cross-certification (e.g. bridge CA), and trust lists. In addition to individual “pure” models, combinations of these models may be adopted, with varying capacities and implications for interoperability. In all cases models require that participants obtain trusted knowledge of one or more public keys to enable them to discover and validate certification paths. This information may be obtained by using special configurations, by out-of-band management and/or by explicit acceptance of keys offered during network exchanges. In PKIX (X.509-based PKIs), trust anchors are usually provided in the form of self-signed certificates. Although this approach is a convenient implementation strategy and allows integrity checking, it does not add any fundamental trust enhancement relative to other representations.

In the web environment, where browsers are preloaded with an high number of trusted root keys, the inclusion of root certificates have become commercially valuable assets. Trust

lists are commonly used in major off-the-shelf applications and provide a very simple solution to the trust management problem for the average user, but they are criticized because the criteria for insertion of a CA into the list are often based more on a commercial rather than a security analysis. Also, the use of the browser as the interface to many internet and local applications means that the TAs trusted for one context do not necessarily mean they should be trusted for a different context (yet this is implied by the way the TAs are used). Therefore we deduce a compelling need for a usable approach to provide Trust Anchor Management services that would allow organizations to dynamically manage TAs for a large number of users.

2.1 The Trust Anchor Management Protocol

In response to this growing demand for bulging TAS, the IETF PKIX Working Group set about defining a new protocol (TAMP) that would help to manage them in an automated standardized way. TAMP is a transport independent protocol that allows an entity designated as a Trust Anchor Manager to query for status and update TAS with TA elements. As TAMP is making its way through the standardization process, it has become apparent through comments on the discussion lists, that the protocol is not broad enough to cover all use cases for TA management. In fact TAMP defines only three types of TA:

- (a.) An Apex TA which is the TA that is superior to every other TA within the TAS and is used to manage the rest of the TAS elements;
- (b.) one or more Management TA that is used to manage cryptographic modules within the TAS;
- (c.) one or more Identity TA that are the traditional X.509 anchors used to validate certificate paths for typical PKI

Obviously there are many scenarios where the choice of Apex TA leads to issues. In particular allowing a user to control their own Apex TA may undermine the possibility for all the browser (and OS) vendors to adopt TAMP in the preferred way to manage their respective TAS (however, the later may simply be an effort by browser vendors to bind users to them by holding onto the responsibility of managing trust for the users, but could also simply be a function of the TAMP standard not being completed and accepted by the community yet).

Moreover the current document which specifies the requirements for the TAM protocol [17] restricts the design to the “push” model. In this model, a centralized service would directly manage the application(s) TA store by pushing the content directly to the application. This approach is specifically adopted because the current version of TAMP is thought to be efficient for enterprises where the control over the clients can be very strict. Future versions of the protocol, however, could extend this approach to a “pull” model which would allow for more interoperable TAM services across an organization’s boundaries.

One potential issue with TAMP is that it specifies that there should be one and only one Apex TA for each TAS. While this makes perfect sense from a management protocol perspective, it has implications for the browser trust model. A

Table 1: Comparison between Trust Models.

	Hierarchies	Cross-Cert	Bridge CA	Trust Lists
Trust Anchor	Hierarchy Root	Local CA	Local CA	Listed CAs
Inter Domain Support	Poor	Good	Very Good	Good
Path Construction	Very Easy	Hard	Easy	Very Easy
Repository Dependency	Low	High	High	Low

single root that controls the trust settings of all other trust anchors in a browser TAS should only be allowed where the Apex TA is under direct control of the user. If the Apex TA was controlled by a single vendor, then that vendor could potentially lock out any other vendors from being accepted for that user, and enforce a restriction of trade. In some sense, this is the situation today with the current browser trust model—the browser vendors determine who is in the trust list i.e. the vendors act as an Apex TA—however, out of band updates by users are permitted in the current browser model, and it is not clear that the equivalent functionality is supported in TAMP. Also, TAMP being an “automated” management protocol, means that updates occur without notifying the user once the initial subscription has been entered into.

2.1.1 Current Limitations

Putting the responsibility for managing their TAS into the hands of an uneducated or inexperienced user also has severe trust implications—which is why the vendors have generally undertaken the current process to make a best effort determination at who should be trusted by default and who should not, and allow the users to manage it from there. The problem with this approach is that almost all average users have no idea how to make a trust decision about a given TA. Sure, there are standards and processes that can be utilized to facilitate the trust decision, and that is what most browser vendors currently do for the TAS default set, but a user can not be expected to take on that responsibility as an individual, when even experts in the field have issues agreeing on trust implications on a regular basis. For instance, just because an organization spends \$120K to get a Web Trust audit, it does not necessarily mean that it runs a benevolent Certification Authority (CA)—nor does getting a Web Trust approval necessarily mean that it is running a secure and trustworthy PKI! Yet often this is the yard stick applied to vendor gated TAS.

Putting aside the question of individuals managing their TAS for the moment, a business or enterprise must also make decisions about what TA they intend to trust and for what purpose. This is based upon some assessment (usually risk-benefit based), and they advise their constituents via policy or other method as to which TA’s should make up a TAS that is acceptable to their enterprise. Individuals within the enterprise then adjust their TAS based on the advice or policy from the enterprise or community to which they belong. TAMP is a fantastic protocol to facilitate this process as long as the individual has some way of establishing trust with the enterprise or community. Since individuals are most often not restricted to having trust relationships with a single enterprise or community, an individual adopting a single Apex

TA from a given enterprise is not a sufficient solution, since this may lead to conflicts for an individual as they interact with multiple organizations, by allowing the TA policy (and Apex TA) of one organization to control the trust settings of another.

Therefore we propose that browsers should support a single Apex TA, instantiated by a user, that they can utilize to subscribe to multiple TAS management services (which may be organization based), and that the browsers support multiple TAS instances that are scoped for a given purpose or application, for a given enterprise or community of interest. That way, a user of enterprise X services could subscribe to the enterprise X TAS Management Service (TMS), which would automatically update a partition of their browser of choice TAS, with those TA’s that enterprise X has determined as trustworthy. The user’s browser would then rely upon those trust settings in that partition of their TAS, for all domains that the user indicates they are applicable to.

This approach would make it possible, for instance, for a user to subscribe to a company TMS, and have a partition of their browser TAS (it could be the entire TAS, or a separated store altogether) be automatically set to have the trust settings that the company recommends. The user could then apply those trust settings globally if they wished to all transactions they undertake, or could restrict it to only those dealing with services from the company domain. This would mean that browser vendors could be relieved from managing users initial TAS (thus reducing costs and effort) or offer it as a separated service. Consequently, users would have a simplified trust decision to make rather than having to evaluate each TA individually, that is “do I want to adopt the recommended TA’s from organization X ?”.

Now in order to make the above feasible, it should be possible to discover what PKIs are trusted by what organization, whether an organization has a TMS, and how to access those services. Until recently, there was no mechanism to discover these details.

The PKI Resource Query Protocol (PRQP) [11,12]—recently adopted as a working item by the PKIX Working Group at IETF—would facilitate users and application vendors by being able to discover appropriate PKI and TMS services.

2.2 The PKI Resource Query Protocol in a Nutshell

Integrating all current protocols and standards to provide an efficient way to discover PKI related services is an open challenge. The PRQP protocol provides a simple approach that changes the current paradigm by allowing for more dy-

dynamic management and configuration of applications and their TAS.

PRQP assumes the presence of a Resource Query Authority (RQA) that would provide applications with the locators of services associated to a particular Certification Authority (CA). In PRQP, the client and a RQA (the server) exchange a single round of messages where the client requests a resource token by sending a request to the server and the server replies back by sending a response to the requesting entity. An RQA can play two roles. First, a CA can directly delegate an RQA as the party who can answer queries about its certificates, by issuing a certificate to the RQA with a unique value set in the *extendedKeyUsage* (i.e. **prqpSigning**). The RQA will provide authoritative responses for requests regarding the CA that issued the RQA certificate. Alternatively, an RQA can act as *PRQP Trusted Authority* (PTA). In this case, the RQA may provide responses about multiple CAs without the need to have been directly certified by them. In this configuration the RQA may be configured to respond for different CAs which may or may not belong to the same PKI as the RQA's one.

3. DYNAMIC TRUST ANCHOR MANAGEMENT: THE NEXT CHALLENGE

PKIs enable trust judgments between distributed users without the need of a direct interaction between them. However, being able to securely identify a user is not sufficient. Indeed in order to make trust judgments, a relying party needs more information than the user's bare certificate. For example, a Web Server or a SSO application must be able to locate critical parameters such as the certificate repositories and certificate validation servers relevant to the trust path under consideration.

Current PKIs often fail to provide such integration, therefore application vendors and users struggle when trying to enable all the supported protocols. This failure is primarily due to the lack of (a.) interoperability between PKIs and (b.) availability of pointers to services. In [11], the authors point out how current CA certificates do not provide a good source of information when it comes to discovering the services that are associated with them. For example, out of the analyzed browser stores¹, almost no service locators were provided in the certificates (besides a few OCSP pointers). This lack of flexibility and service availability is also present in current TA management. Indeed each party involved in TA management deals with it independently without considering all the different points of view of the other parties.

When considering Trust Management issues, one of the most common errors is to restrict the view to a single perspective only. We think that the problem should be analyzed from, at least, three different points of view: the user perspective, the application vendor perspective, and the PKI vendor perspective. In this section we focus on the benefits of adopting a dynamic approach to TAM and how PRQP is a fundamental building block of a viable TAM infrastructure.

The User Perspective. *As far as most users are con-*

¹Firefox, IE, and Konqueror

cerned, the notion of Digital Certificates or Public Key Infrastructure is a mystery. Many papers [5,20] have discussed the issue of the lack of understanding about security and the underlying technologies. Therefore we need a new approach that will allow Users to make informed Trust decisions.

A Recent paper [7] has shown that users understand trust through the concept of Institutions and their reliability more than Digital Certificates and Certification Authorities. In particular the study shows how users tend to rely on well known institutions and their level of reliability more than other users' suggestions (also if well known) or technical details. It is therefore clear how the concept of trust should be presented to the users in terms of what they can really understand: "who already trusts this organization (eg., Certification Authority) ?"

By integrating PRQP with current applications and enabling the automatic discovery of PKI-related resources, the TA management problem can be actively put into the hand of the users. One major hurdle to overcome however, is how the interface for managing these services are presented to, and interacted with by the user. A sample User Interface (UI) from a common application is reported in Figure 1. It is evident that such interface is too complicated for the majority of users. In order to be able to provide an easier UI, the application could dynamically discover all the services provided by the PKI and the trust information about the CA provided by other organizations, and present this additional information to the user in a meaningful way. The PRQP protocol would allow applications to provide such an interface for Trust Management.

The Application Vendor Perspective. *As far as the average developer is concerned, the notion of Digital Certificates or Public Key Infrastructure is also a mystery.* Moreover the complexity of current PKI standards makes it difficult for developers who are not PKI experts to correctly process the information within certificates. Also, because of the number of different services and repositories to access in order to build a simple certificate chain, the average application either does not implement correctly and fully the standards or has dozens of options to configure those services and they pass those decisions onto the user who has no idea how to set them up.

To solve this problem PRQP provides a simple and efficient way to enable applications to easily discover PKI services. We estimate that it would be possible to get rid of most of the current configuration options faced by vendors in preparing UIs, thus allowing for easier UIs and faster development times in applications.

In particular, for TAS Management, PRQP could be used to discover TAMP data sources. Multiple sources could be dynamically discovered for specific domains and a hierarchy of stores can be built according to the user's preferences. The Application Vendor will still be able to dynamically provide its own Trust Anchor Management updates (e.g., for OS management purposes or specific trusted domains) while the user would be able to add its own settings.

The PKI Vendor Perspective. There are many differ-

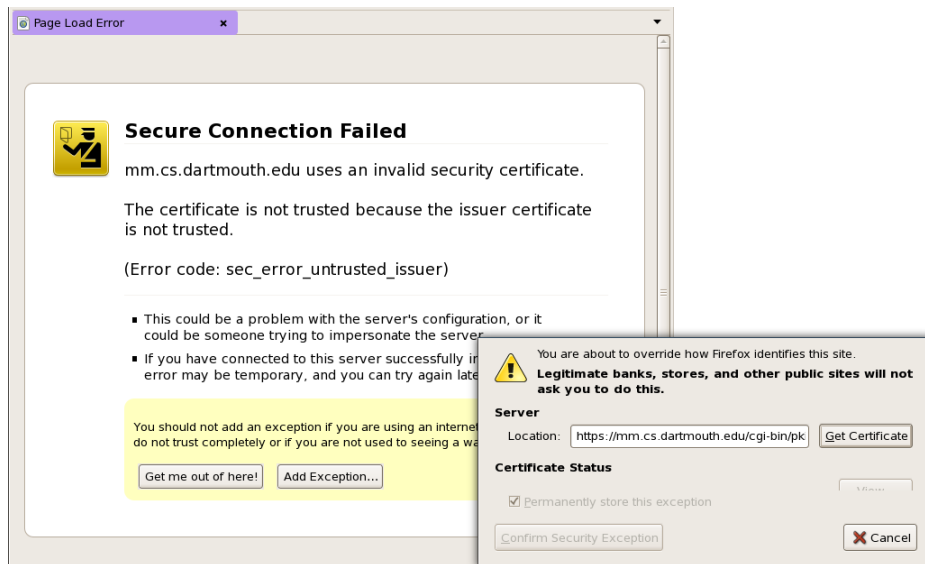


Figure 1: Example UI—Understanding the full consequences of adding a Trust Anchor is beyond the ability of the average users and of many advanced ones. Also, interfaces from other applications suffer from similar problems: either it is misleading and too complex or it provides no useful information to the user.

ent types of PKIs that serve different purposes. Besides SSL vendors, a huge market for PKI related services is today totally ignoring commercial vendors. Environments like Computing Grids where Public Key Certificates have been used for well over a decade are, today, left with little support by application vendors. By enabling organizations to integrate different PKIs and provide dynamic TA management, the occasion for a bigger market is evident. The more usable PKIs are, the more the users will be willing to adopt this technology. There are potentially real world-wide client-side PKIs everywhere you look—e.g, cellular phones, mobile devices, wireless network authentication, home devices, e-commerce application, etc...

In order to provide more flexible support to the users, PKI vendors need the capability to deploy more flexible PKIs. In particular the adoption of PRQP—and eventually its Peer-2-Peer extension [13]—can enable vendors to activate, move, dismiss, and enhance services easily. This provides PKI deployers with the possibility of molding the infrastructure as the needs or profile of the users change thus being able to offer a more usable experience of the offered products. An example of the benefits of adopting PRQP would be the seamless dismissal of an old (or potentially broken) service—like CRLs—in favor of a new one—e.g. OCSP—without the need of re-issuing every certificate because of the usage of static extensions.

As we wait for trust anchor management protocols to wind their way through the standardization process, we still need to manage trust anchors in environments today. In the next section we present a hybrid trust infrastructure that could be deployed in current systems to help manage TAs until TAMP matures.

4. TA MANAGEMENT FOR CURRENT APPLICATIONS

While waiting for standard protocols to be finalized and adopted by applications, the need for a solution is compelling. The trust model we present in this section allows organizations (or Virtual Organizations such as accreditation bodies) to provide users with an easy-to-use solution which is compatible with deployed software and supports the way users make trust decisions.

When already deployed infrastructures—established by different organizations—want to develop a common trust relationship between them, some form of cross-certification must be applied to link them. Table 1 reports a summary of the characteristic of different trust models. It could be desirable to have a model which provides the flexibility typical of cross-certification while not introducing high complexity for certification path building. Our work is based on the integration of two different components: a deployed PRQP infrastructure and a hybrid trust model. Our solution is capable to address trust management issues by providing:

- a method to avoid multiple trust points hardwired into applications (i.e. to avoid embedding large trust stores); our approach requires a single trust anchor (e.g. a Root CA which could, in future, act as an Apex TA under TAMP) for inter-hierarchies trust support
- a simple trust management system for applications based on cross-certificates
- the possibility for PKI operators to provide their users with a set of revocable endorsed TAs

Cross-Certification has the interesting characteristic that it can preserve the organization's ability to enforce constraints within their hierarchies. Annex G of the ISO document [9] discusses specifically this case by presenting three different types of Cross-Certificates:

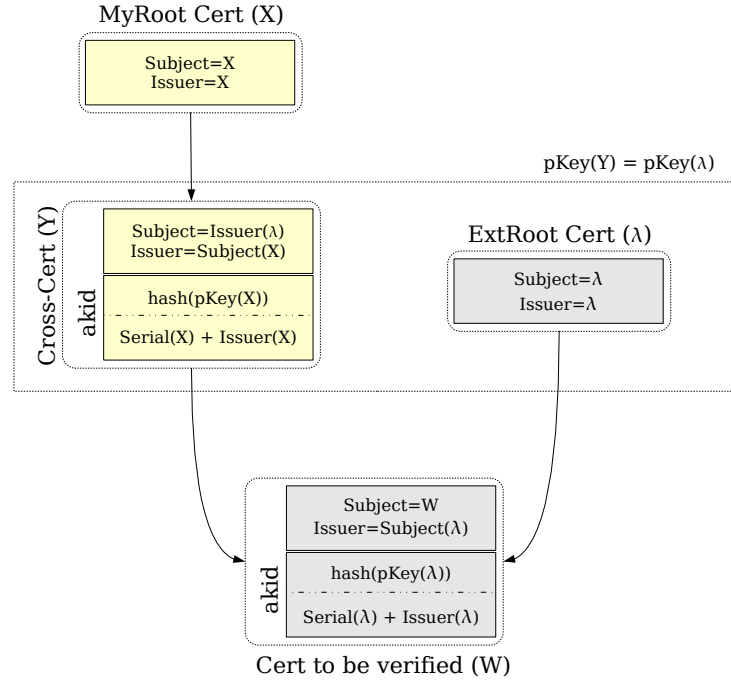


Figure 2: Cross-Certificate of an external Root CA.

- *Hierarchical Cross-Certificates* extend path construction from leaf CAs upwards Root CA, thus allowing relying parties to use their local CA as trust anchor
- *General Cross-Certificates* to interconnect CAs. This can be done either at root level or at sub-CA level
- *Special Cross-Certificates* are intended to allow selective establishment of certification paths that may not conform to the restrictions imposed hierarchically

In this work we combine the hierarchical and cross-certification models, by using “special” cross-certificates to allow for locally managed trust path building for applications. In order to better understand how our model is particularly efficient and why it is supported out of the box by existing applications, the next subsection provides a brief description of the certification path building process in PKIX.

4.1 Certification Path Building

Assuming we have a set of trusted certificates and we need to build a trust path from certificate x up to one trust anchor, we have to identify the issuer of certificate x and check if it is trusted or not. In case it is not, we need to verify if its issuer is trusted. The process continues until a trusted certificate is identified in the chain or no suitable issuer is found among the available certificates or, finally, a self-signed certificate is reached. Therefore the identification of the issuer of a certificate is a crucial aspect of the path building process. Assuming we want to check if the subject of certificate y is the issuer of certificate x , the needed steps are:

- check the *Issuer* field of certificate x to be equal to the *Subject* field of certificate y
- If *authorityKeyIdentifier* extension exists in certificate x , then check it matches the data of certificate y

- Check that the *keyUsage* in certificate y supports certificate signing
- Check that the policy and name constraints requirements are fulfilled

To continue the chain building process, repeat the steps above till one trust anchor or a self-signed certificate is reached. Special attention should be made during step (b). In fact the presence and the contents of the *authorityKeyIdentifier* can vary depending on the certificate to be verified. Four different certificate profiles are hereby reported which summarize the possible contents of the *authorityKeyIdentifier* (*akid*) extension:

- α the extension is not present in the certificate. The chain is actually built by using the subject and issuer fields of the certificate
- β the extension is present in the certificate and it carries the *keyIdentifier* which, usually, contains the hash of the public key of the issuer
- γ the extension is present in the certificate and it carries the *authorityCertIssuer* and the *authorityCertSerialNumber*.
- δ the extension is present in the certificate and it carries both the *keyIdentifier* and the *authorityCertIssuer* and *authorityCertSerialNumber* couple.

The first two profiles do not require special care when dealing with cross-certification while, as we will discuss in detail in the next parts of this section, the other two do. In fact, within the last two profiles, the certificate’s issuer is identified both by the *Issuer* field contents and by its issuer and its serial number in the *akid* extension. For instance if we want to identify the certificate x we use the couple:

$$issuer(x) + serialNum(x) \quad (1)$$

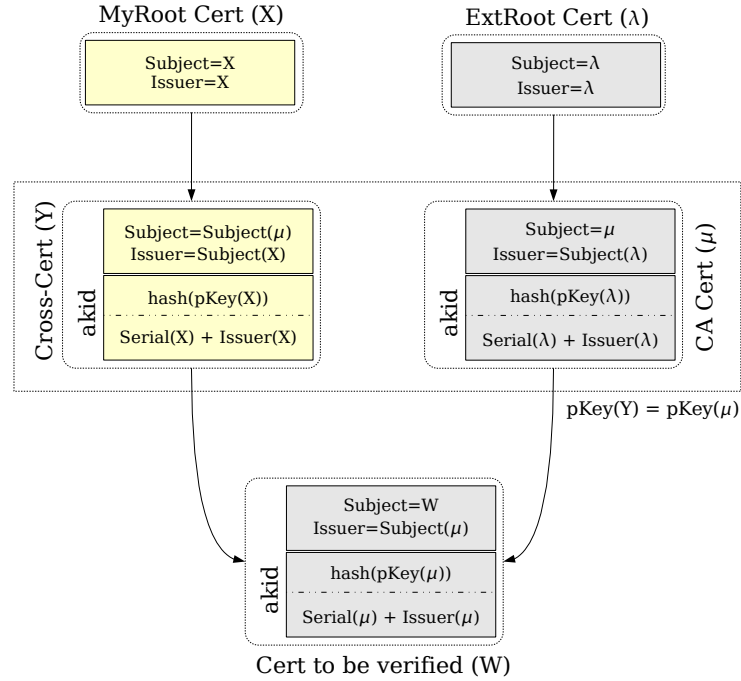


Figure 3: Cross-Certificate (Y) for an external Sub-CA (μ).

in the same way, if we want to identify the issuer of certificate x we use the couple:

$$issuer(issuer(x)) + serialNum(issuer(x)) \quad (2)$$

where the $issuer(x)$ is the *Subject* of the issuer of x . This explanation is needed to better understand differences in our trust model when trusting Root CAs or subCAs as explained in Sections 4.3 and 4.4.

4.2 The Model Basics

In our model we assume that at least one CA certificate is installed into the application certificate store and it is trusted, e.g. this may be the Apex TA in future as required by TAMP. To better introduce our solution, a practical example could be represented by a National Research Network (NREN)–named OrganizationA— which already established a PKI and wants to provide a way to automatically verify all the certificates issued by other n NRENs. The root CA from OrganizationA ($orgA_rootCA$) cross-certifies the other NRENs root-CAs (i.e. $extCA_1, extCA_2, \dots, extCA_n$) by issuing certificates carrying the same details as the original ones (e.g. Subject, Public Key and Extensions). The newly issued certificates will only differ from the original ones in that they are issued by $orgA_rootCA$ within the OrganizationA’s infrastructure. The cross-certificates will be referred as $extCrossCA_{1\dots n}$.

If a user from OrganizationA’s hierarchy wants to verify a certificate issued by one of the $extCA_{1\dots n}$ hierarchies (e.g. in order to verify a signed S/MIME message) then a path from that certificate ($extUserCert$) up to $orgA_rootCA$ has to be established.

By providing the $extCrossCA_{1\dots n}$ certificates to the client, it is possible to build a trusted chain of certificates up to the OrganizationA’s root-CA. Indeed by comparing the *authorityKeyIdentifier* in the $extUserCert$ with the locally stored $extCrossCA_{1\dots n}$ certificates’ contents, the path construction can proceed up to $orgA_rootCA$, thus establishing a trusted chain from $extUserCert$ up to the only trust anchor needed in the application.

Although the basic principles are very simple, further analysis is needed to correctly allow the path building process to take place. In the next subsections we provide the details of our hybrid trust model. In particular we will use the following terminology:

- $issuer(x)$ identifies the Issuer field in certificate x
- $subject(x)$ identifies the Subject field in certificate x
- $serialNum(x)$ identifies the serialNumber field in certificate x

4.3 Trusting Root CAs

Extending the trust to a whole external PKI is done by issuing a cross certificate for the external root-CA. Figure 2 reports the scenario where the certificate W of the external PKI is linked to the organizational PKI throughout Y . If the *akid* extension is not present (case α and β of Section 4.1), the cross-certificate Y will use the same public key and *Subject* of the original root-CA whilst having a different *Issuer* ($MyRoot$). If the *akid* extension is used (case γ and δ in Section 4.1) it is important to issue the cross-certificate in such a way it will be referred by the *akid* of certificate W . In this particular case of cross-certifying a root-CA, the *Issuer* and the *Subject* contents of the certificate are the

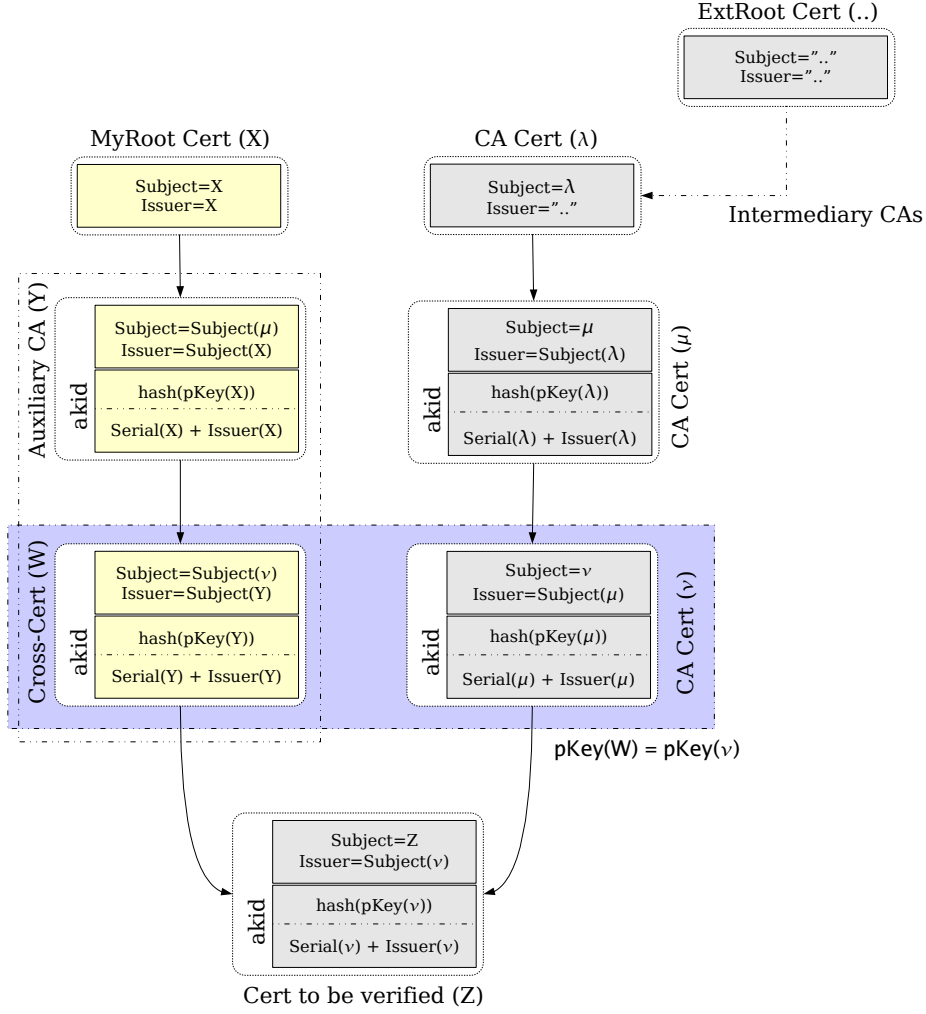


Figure 4: Our Hybrid Trust Model with Auxiliary CA (Y).

same. This means that, according to (2), the issuer of the root-CA is identified in the *akid* by:

$$issuer(issuer(rootCA)) + serialNum(issuer(rootCA)) \quad (3)$$

where:

$$issuer(rootCA) = subject(rootCA) \quad (4)$$

$$serialNum(issuer(rootCA)) = serialNum(rootCA) \quad (5)$$

therefore (3) becomes:

$$issuer(subject(rootCA)) + serialNum(rootCA) = subject(rootCA) + serialNum(rootCA)$$

hence if the certificate W (with the *akid* extension carrying the $serialNum(\lambda) + Issuer(\lambda)$ identifier) points correctly to Y if and only if:

$$subject(Y) = Issuer(\lambda) = subject(\lambda) \quad (6)$$

and

$$serialNum(Y) = serialNum(\lambda) \quad (7)$$

The condition (7) is the most hard to match. In fact typically rootCA certificates have the *serialNum* set to zero and, as the serial number must be unique within a CA, it is difficult that the serial number required for Y is available under X . As we will discuss in detail in the next sections, the *serialNum* constraint can be fulfilled by introducing into the model an auxiliary CA for each external CA/PKI we want to establish a trust with.

4.4 Extending the model: trusting subCAs

Figure 3 represents the scenario where we want to issue a cross certificate to include only a sub-CA, not a whole external PKI. In this case we want to be able to verify certificate W by building the chain:

$$W \longrightarrow Y \longrightarrow X \quad (8)$$

to do this, the cross certificate Y will have:

$$subject(Y) = subject(\mu)$$

if no *akid* extension is present in W or if it contains only the *keyIdentifier*, the path building process does not present

particular issues. On the contrary if we are in case γ or δ (Section 4.1), then the path building process will fail because it is impossible to issue Y obeying the *akid* constraints. In fact the *akid* of W identifies the *issuer*(W):

$$\begin{aligned} authorityCertIssuer(W) &= serialNum(\mu) + issuer(\mu) \\ &= serialNum(\mu) + subject(\lambda) \end{aligned}$$

and to fulfill its requirements (besides the *serialNum* problem) it should be:

$$issuer(Y) = issuer(\mu) = subject(\lambda) \quad (9)$$

unfortunately with the proposed infrastructure, this is not possible because:

$$issuer(Y) = subject(X) \neq subject(\lambda) \quad (10)$$

To overcome this problem, an addition to the model is needed, as detailed in the next section.

4.5 Introducing auxiliary CAs

To provide a generally applicable model we introduce an auxiliary CA (Figure 4) into our schema. The purpose of this CA, which is identified by the certificate Y , is to provide a more general solution that is capable of addressing the potential limitations imposed in the path building process by the *akid* extension.

Indeed, to have the *akid* to correctly point to the cross-certificate W , we have to set *subject* of the auxiliary CA Y to be equal to the *issuer* of the sub CA μ . By adding Y to the infrastructure, we have:

$$subject(W) = subject(\nu) \quad (11)$$

$$issuer(W) = subject(Y) = subject(\mu) = issuer(\nu) \quad (12)$$

$$serialNum(W) = serialNum(\nu) \quad (13)$$

therefore the *akid* of Z points correctly to W because the serial number of W is equal to *serialNum*(ν) as in (13), its issuer is equal to *Issuer*(ν) as in (12) and its public key is equal to *pKey*(ν) as W is the cross certificate for ν .

Moreover the *Subject/Issuer* content requirements are also satisfied because:

$$issuer(Z) = subject(\nu) = subject(W) \quad (14)$$

It is therefore possible to extend trust to external PKIs/CAs by introducing only one auxiliary CA that issues one cross-certificate for the CA to be trusted.

4.6 Revoking Trust Anchors

One interesting aspect of the proposed model is the possibility of revoking TAs by using standard PKIX mechanisms. In fact, because trust is built by issuing standard certificates, it is possible to revoke them by simply using revocation services that are already in place (e.g., CRLs or OCSP).

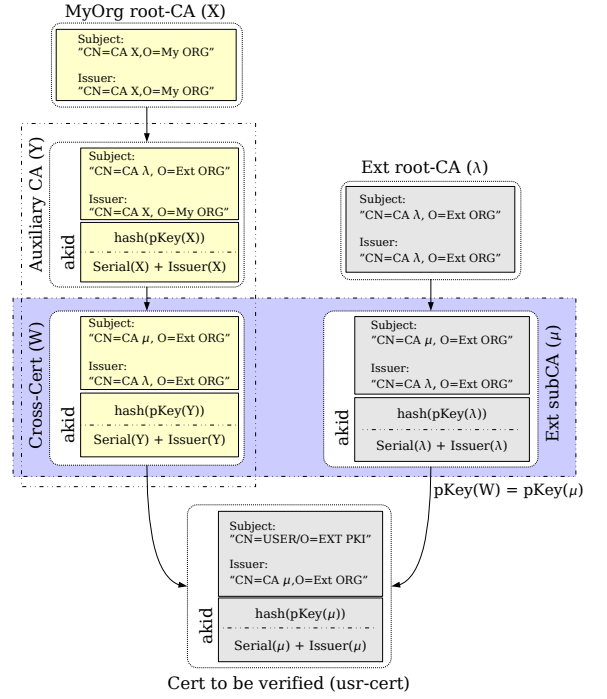


Figure 5: Test bed environment.

4.7 Application Support

One of the main advantages of this solution is that it is supported out-of-the-box by several widely diffused operating systems and applications. This section focuses its attention on performed tests and code analysis (whenever possible) to describe how the hybrid trust model is actually supported. Figure 5 depicts the used test environment. The certificate to be verified in the example is the “usr-cert” which is issued by subCA- Y from the external organization we want to establish a trust link with. In order to do that, we issue a certificate for the subCA- Y which acts as the auxiliary CA in our hybrid trust model. Then subCA- Y issues the cross-certificate subCA- W which, in our model, “replaces” the original certificate from the external organization in the path building process. The purpose of the performed tests was to establish if the proposed trust model was supported by current applications.

4.7.1 OpenSSL libraries

In the Unix world (e.g. Linux, BSD, Solaris, etc...), the OpenSSL suite provides the most used cryptographic libraries. By analyzing the OpenSSL code, it has been possible to discover that our model is actually supported by OpenSSL. Anyway further considerations are needed for special covered cases.

The OpenSSL cryptographic library provides the functions needed to build the chain of certificates and to verify them. To better understand how the library verifies certificates, it is useful to describe the main involved data structures. OpenSSL uses different objects during the verification process:

- the `X509_STORE` object is actually used to represent a collection of certificates and eventually certificates revocation lists (CRLs)
- the `X509_STORE_CTX` object holds the data used during an actual verification

After loading all the needed data in the `X509_STORE`, OpenSSL uses this datastructure to initialize the `X509_STORE_CTX` by calling the following function:

```
int X509_STORE_CTX *X509_STORE_CTX_init(...)
```

If the initialization function completes successfully a pointer to a `X509_STORE_CTX` data structure is returned (`ctx`). The following function is then used to perform the verification of the chain of certificates:

```
int X509_verify_cert(X509_STORE_CTX *ctx);
```

The verify function builds the chain up to the trust anchor by looking in the `ctx` for a suitable issuer of the current certificate. This process is then repeated until no issuer for the certificate is found in the `ctx` either because of an error or because a trust anchor has been reached. The checks performed to see if certificate *B* has been issued by certificate *A* are:

- Check the **Subject** field of *A* to be equal to the **Issuer** field of *B*
- If *authorityKeyIdentifier* exists in *B*, then check it matches details of certificate *A*
- If *keyUsage* extension is present in *A*, then check it supports certificate signing
- returns 0 on success, or a positive for the reason for mismatch

Thanks to the addition of the auxiliary CA, the OpenSSL verification function returns successfully if the calling application provides the chain up to the trust anchor. Tests have been carried out by using applications which use these libraries (i.e. KMail and Konqueror) and, as we expected, results were positive.

4.7.2 Mozilla Suite

The explained hybrid trust model provides a method for trust path building that follows RFC-5280 [6], therefore all applications should be able to support the model. From the performed tests, we found out that our model is fully supported when one of the following conditions is matched:

- the CA to be trusted is a rootCA
- the CA is a subCA and the client does not have the original the chain of certificates up to the external rootCA stored in the local repository.

Regrettably, Mozilla based applications (i.e. Firefox and Thunderbird) do not fully support our solution in the remaining cases because of the way certificates are stored in its local repository. In fact if the original chain of certificate is already present in the certificate store an error about the presence of two certificates issued by the same authority with the same serial number is displayed to the user when importing the cross certificate for the subCA. This is obviously an error in the certificate store as it reports wrong information to the user, i.e. the cross-certificate has the same

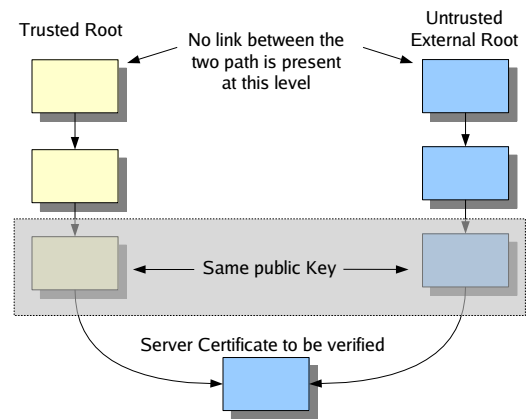


Figure 6: SSL/TLS connections and our hybrid trust model.

serial number but its issuer is actually different. Hopefully this behavior will be fixed in future versions of the software. As detailed in Section 4.7.4 issues are also present when SSL/TLS connections are considered.

4.7.3 Windows CryptoAPI

It has been possible to successfully test support for our model on Windows based systems, in particular it was possible to correctly verify certificates in:

- Windows 2000
- Windows XP
- Windows 2003
- Windows Vista

Unfortunately, it was not possible to verify how exactly the path building is done by Windows CryptoAPI because of the lack of the libraries source code. Anyway every application that relies on the Windows system libraries should automatically support this model. Performed tests show that support for the proposed model is available in Internet Explorer as well as in Outlook.

4.7.4 SSL and TLS connections

Besides applications based on Microsoft CryptoAPI that fully supports the proposed trust model, our tests show that some of the tested software presents issues when setting up SSL/TLS channels. In particular the problem we found is present only when extending trust to *non-root CAs* (Section 4.4). In OpenSSL, for example, the function used to build and verify the path up to a trusted anchor for secure channel setup is different than the one used for statically verify a chain of certificates. This is due to the fact that the RFC-2246 [8] standard allows the server to push the chain of the certificates to the client. Figure 6 depicts the scenario.

When setting up an SSL/TLS connection, the application makes use of the `SSL_CTX` family of functions. These functions use the chain of certificates that is pushed through the channel instead of the certificates in the local store up to the TA (which is verified against the local store). Therefore when the last certificate in the pushed chain is to be verified, there might be no way for the application to link the

trusted anchor (in the local store) to the external root (in the original trust path). A possible way to avoid this situation would be to check, at each step of the path building process, if either:

- a locally stored certificate is a suitable issuer of the certificate to be verified. If such a certificate exists, use it as the issuer instead of proceeding in the path building process by searching in the pushed certificate chain from the server
- a locally stored certificate has the same public key of the current certificate to be verified. If such a certificate exists, use its issuer from the local repository as the next step in the path building process instead of searching for the issuer in the pushed certificate chain from the server.

In other words, if path checking fails, a second attempt should be made by letting the locally stored certificates to take precedence over those pushed by the peer. By using these simple changes in the path building process of OpenSSL and Mozilla based applications our model is fully supported also for SSL/TLS channel setup.

5. INTEGRATING PRQP AND OUR HYBRID TRUST MODEL

The first interesting feature provided by this trust model is that *trust is locally managed by simply issuing or revoking “special” cross-certificates*. This approach saves users from having to perform security checks each time a new certificate is to be added to the application’s repository. By using the proposed model the verification of public-keys (and extCAs details) can be performed by CA managers (or experts in PKI policy verification and auditing) when issuing the cross-certificate in a reasonably and reliable secure way (e.g. all needed steps to verify details about the CA certificate to be cross-certificated will be checked).

The second attractive feature of this solution is that no extCA_{1...n} original certificate is needed on the application to verify the certificates chain. Thus it is possible to provide trust into applications by locally storing only certificates from one trusted organization which provides the needed TA in the application’s local store. This hybrid trust model could productively be used together with other Trust-Related projects.

An attempt to decouple the trust list from the applications is the TACAR (TERENA Academic CA Repository) project [21], started at the end of 2003. It aims to provide a trusted repository to hold the appropriate root CA certificates needed by applications. The collected certificates are those directly managed by the member National Research Networks (NRENs), or belonging to a national academic PKI, or to non-profit research projects. As in our solution, this project aims to solve the cross-domain usage nightmare of PKI. In particular, the TACAR project provides a certified process for gathering and verifying root-CA certificates, and publishes them in one easily downloadable and importable trusted file. Therefore, as detailed in the next subsection of this paper, by integrating our hybrid trust model with the TACAR repository as a trusted source of root-CA certificates, it could be possible to enable inter-domain verification of all TACAR’s provided certificates.

5.1 Distributing hybrid certificates

One issue that we have not yet addressed in this paper is how to provide the applications with the needed special cross-certificates. In fact, applications need to know that a cross-certificate for that particular rootCA or subCA exists in order to correctly build the verification path up to the trusted anchor. Since recently, there was no interoperable solution to dynamically provide applications with references (URLs) to certificate bundles. We considered several possibilities when trying to address this issue. For example some sort of automatic update system could be implemented to gather the special cross-certificates from a trusted source. Such mechanisms are already in place for updating many modern operating systems, therefore a similar solution could be adopted in some environments. Because we want to provide a generic and interoperable solution across operating systems and applications, we decided to adopt a different approach.

In particular, in our solution we used a PRQP server (i.e., an RQA) to distribute locators (URLs) to sets of certificates endorsed by CAs. In order to do that, we needed to extend the current specifications of the PRQP protocol. To identify the packet of endorsed CAs, we specified a new Object Identifier (OIDs) as follows:

```
id-ad-prqp-p7endorsedTA ::= { id-ad-prqp 100 }
```

which we used to identify the locator for a PKCS#7 signed object. This object contains the set of certificates that are endorsed by a particular CA and it should be signed by the CA directly.

In our infrastructure, when an application needs to import the set of TAs endorsed by a specific CA, it queries the RQA that carries the configuration for the specific CA. In the case of a single organization, the location of the RQA can be provided via DHCP or via DNS SRV records as specified in the current PRQP draft. Other configurations based on Peer-to-peer technologies [13] are also possible.

When the `id-ad-prqp-p7endorsedTA` OID is present in the PRQP request, the RQA will provide the client application with one (or more) pointer(s) to the available packages of endorsed CAs. The client application will then proceed by retrieving the PKCS#7 object from the URL received from the RQA. After verifying the signature on the PKCS#7 object, the application can safely import the list of TAs included in the retrieved object. As all the certificates present in the PKCS#7 object are issued by one single CA (the one that signed the data object), only a single trust decision is required from the user. In the case that the issuing CA is already trusted by the application—eg., the user’s organization CA certificate or the application’s vendor certificate—no interaction with the user is actually required. However, we do recommend that a simple dialog be presented to the user the first time the TA package signed by a CA is actually downloaded from a URL.

6. CONCLUSIONS AND FUTURE WORK

In this paper we described how Trust Anchor Management is a central point for PKI usability. We also explained some of the current standardization activities within IETF and in particular we discussed the possibilities offered by combining PRQP and TAMP. We then focused the central part of the paper on how it is possible to enhance the usability of PKIs from a different point of view. In particular, we advocate that by adopting a more dynamic approach to PKI services—in particular within browsers—both PKI and Application vendors can benefit from easier to use services and more flexible infrastructure. In particular we detailed different approaches to link certification structures, differing in their security properties, their scalability, their management requirements, their implications on path construction and validation, and their dependencies on directory services. Tests show that the hybrid model, which combines the flexibility of cross-certification with the ease of deployment typical of the hierarchical trust model, is supported by many available applications. The solution provided in this paper addresses also trust management issues by using the hybrid trust model to reduce the number of trust anchors needed by applications to just one. Still further investigation is required to better understand possibilities provided by the usage of this model. In particular our efforts are directed to integrate our trust management system with existing realities (e.g. TACAR) to provide practical support in real life environments. We are currently in the process of deploying RQA servers for all of the CAs participating in the TACAR project. We believe that operational experience will be important in validating the usability of this approach and its adoption in specific environments (e.g. research networks and grid communities) where simple trust management of certificates and linking of isolated PKIs is required. We also hope that our work will provide valuable feedback for the standardization of TAMP and provide a useful use-case to further promote the standardization of PRQP.

7. REFERENCES

- [1] EuroPKI Infrastructure. EuroPKI website. [Online] <http://www.europki.org>.
- [2] GSI working group of the Global Grid Forum. [Online] http://www.gridforum.org/2_SEC/GSI.htm.
- [3] The European Policy Management Authority for Grid Authentication in e-Science. [Online] <http://www.eugridpma.org/>.
- [4] The International Grid Federation. [Online] <http://www.gridpma.org/>.
- [5] Alma Whitten and J. D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *8th USENIX Security Symposium*, August 1999.
- [6] 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.
- [7] Denise Anthony, James Kitts, Chris Masone, and Sean W. Smith. Technology and Trust. In *Eastern Sociological Society Annual Meetings*, Feb 2008.
- [8] T. Dierks and C. Allen. The TLS Protocol. Internet Engineering Task Force: RFC 2246, January 1999.
- [9] ISO/TC68/SC2. Certificate management for financial services – Part 1: Public key certificates. ISO 15782-1:2003, August 2003.
- [10] Kelvin Yiu. 6th Annual PKI R&D Workshop, “Applications Driven PKI (It’s The Apps, Stupid!)”, April 2007.
- [11] Massimiliano Pala. PKI Resource Query Protocol (PRQP). Internet-Draft, Experimental Track, June 2008.
- [12] Massimiliano Pala and Sean W. Smith. AutoPKI: A PKI Resources Discovery System. In *Public Key Infrastructure, 4th European PKI Workshop: Theory and Practice, EuroPKI 2007*, volume 4582. LLNCS, Springer-Verlag, June 2007.
- [13] Massimiliano Pala and Sean W. Smith. PEACHES and Peers. In *5th European PKI Workshop: Theory and Practice*, volume 5057, pages 223–238. Lecture Notes in Computer Science, Springer Verlag, June EuroPKI 2008.
- [14] Massimiliano Pala, Marius Marian, Natalia Moltchanova, Antonio Liroy. PKI past, present and future. *International Journal on Information Security*, 5:18–29, January 2006.
- [15] M. Pala, A. Liroy, M. Marian, and N. Moltchanova. The EuroPKI Experience. In *Proceedings of the 1st European Workshop on PKI*, volume 3093, pages 14–27, Berlin, Germany, June 2004. Springer-Verlag.
- [16] R. Guida, R. Stahl, T. Bunt, G. Secrest, J. Moorcones. Deploying and Using Public Key Technology: Lessons Learned in Real Life. *IEEE Security and Privacy*, pages 67–71, September 2004.
- [17] R. Reddy, C. Wallace. Trust Anchor Management Requirements. Internet Draft: Informational, October 2008.
- [18] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. *Communications of the ACM*, 21 (2):120–126, 1978.
- [19] Sean W. Smith. A Funny Thing Happened on the Way to the Marketplace. *IEEE Security and Privacy*, 1 (6):74–78, November/December 2003.
- [20] Simson L. Garfinkel and Robert C. Miller. Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. In *Proceedings of the 2005 symposium on Usable privacy and security*, pages 13–24, 2005.
- [21] TACAR Project. TERENA Academic CA Repository. [Online] <http://www.tacar.org>.
- [22] W. Diffie, M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22 N.6:644–654, November 1976.

Massimiliano Pala <massimiliano.pala@dartmouth.edu>
Scott. A. Rea <Scott.A.Rea@dartmouth.edu>



Usable Trust Anchor Management

Trust, Today

- “Technical” Trust is achieved by building a path (chain) from the certificate to be verified up to a trust anchor
- Trust Anchors are mostly represented by root (self-signed) CA certificates
- Deployed Trust Models
 - **Hierarchies (Easiest for Implementers)**
 - **Cross-Certification**
 - Bridge-CAs
 - **Trust Lists (Applications)**

Trust Anchors, today

- The number of trust anchors built into applications and Operating System is increasing and applications were not thought for such a large number (Yiu, Kelvin; Microsoft; 2007)
 - **~130+ in Firefox Store**
 - **~150+ in OSX Store**
 - **~280+ in XP Store**
- Level of awareness is very low among users

Trust Anchors, today (cont.)

- Some communities need more flexible Trust Anchor Management (TAM)
 - **Computing Grids**
 - Distribution of Affiliated Cas
 - **Virtual Organizations**

Our Contribution

- While waiting for TAMP to be standardized (IETF)
- Usable TAM
 - **Reduce the number of trust anchors built into applications**
 - Ideally to just one
 - **Based on “semi”-cross certification and PRQP integration**
 - **Support for deployed applications**
 - **Centralized TA management**
 - Revocable set of Endorsed TAs

Background

- Path building process
 - **How to Identify a certificate**
 - **Simple path building example**
- More details on path building process
 - **AKID Extension & path building**
- The PKI Resource Query Protocol

How to identify a Certificate

- To identify certificate “ α ”

Issuer(α) + serialNumber(α)

- To identify the issuer certificate of “ α ”

**Issuer(Issuer(α)) + serialNumber(Issuer(α)) =
Issuer(β) + serialNumber(β)**

- In root CAs where Issuer and Subject are the same:

Issuer(Issuer(α)) = Issuer(α)

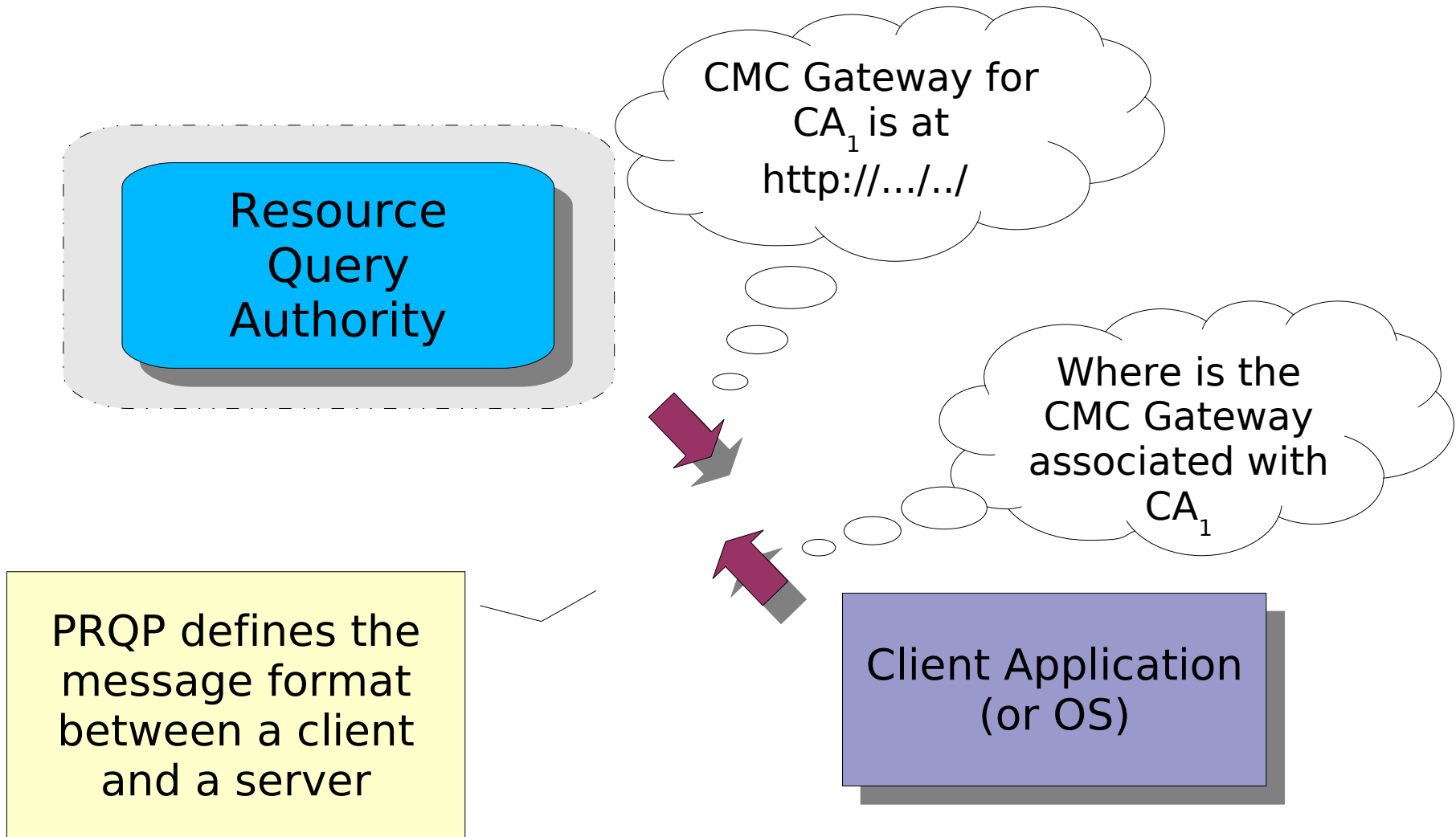
Path Building Process-1

- To verify certificate α starting from a set of trusted certificates we need to:
 - Identify *the issuer* of α (i.e., β)
 - Verify if β is trusted
 - If it is from the *set of trusted certificates*, ok
 - If it is *not trusted* repeat the process until a trusted or a root certificate is identified

Path building & AKID-2

- i. The AKID extension is absent - path building takes place by using Subject/Issuer coupling
- ii. The AKID extension is present and carries the `keyIdentifier` - path building takes place as (i.) and `keyIdentifier` contents are checked
- iii. The AKID extension is present and carries the `authorityCertIssuer` + `authorityCertSerialNumber` - path building takes place as (i.) plus the extension contents are checked
- iv. The AKID extension is present and carries both - path building takes place as (i.) plus extension contents are checked (both)

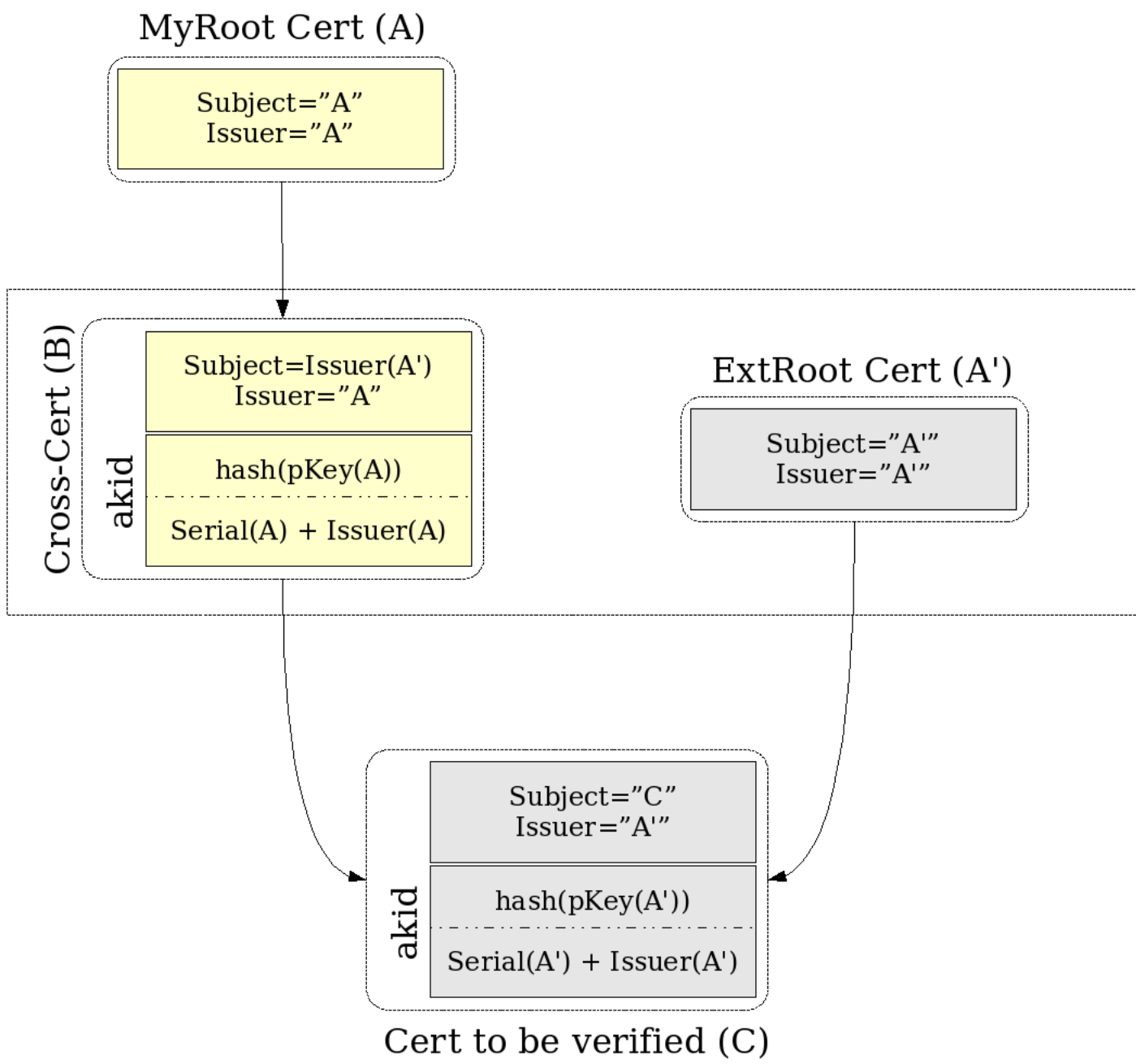
PRQP in a Nutshell



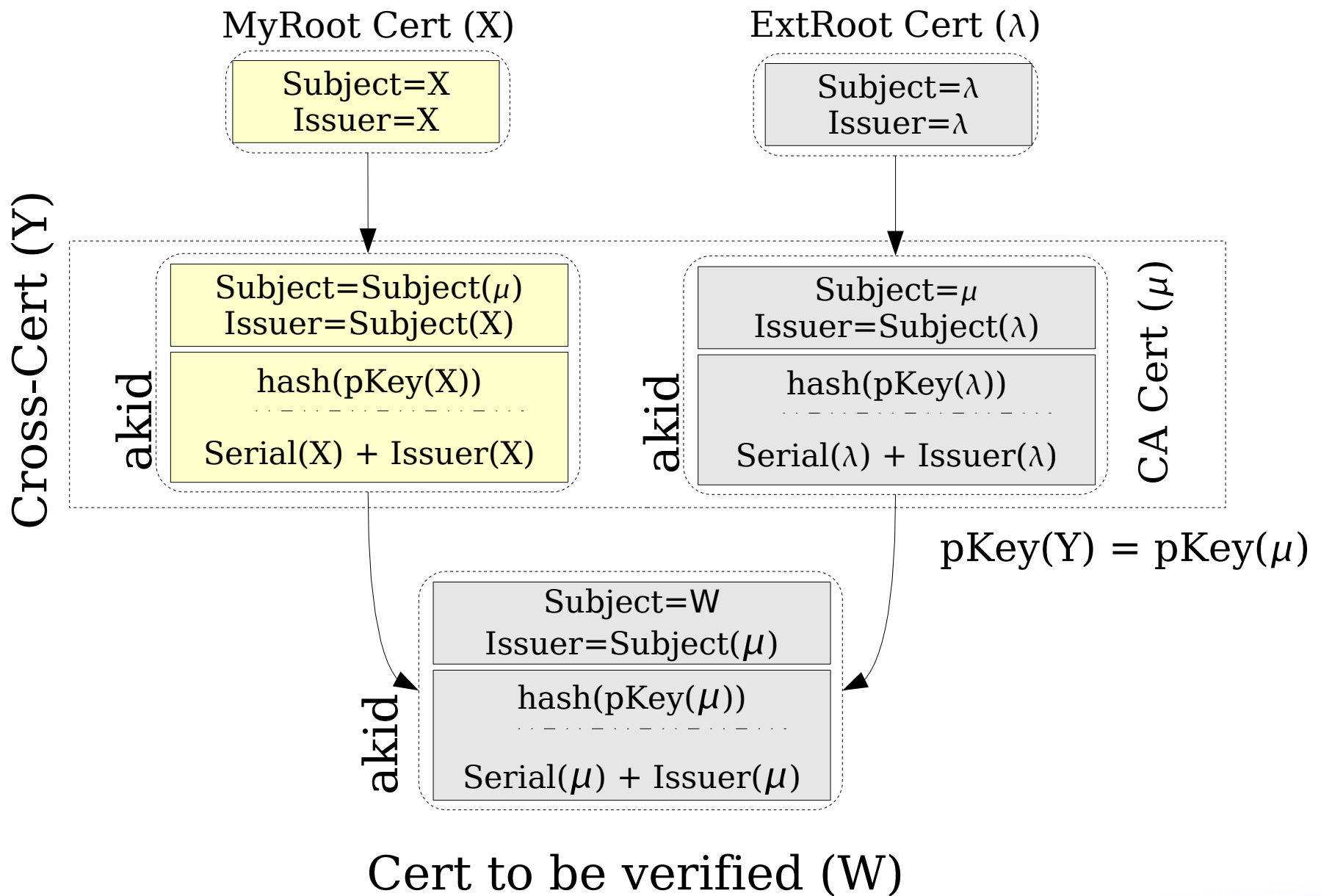
Our Model

- Trusting External PKIs
 - **Including a single PKI**
- Extending the Model
 - **Trusting Sub-CAs**
- The Final Model
 - **Introducing the Auxillary CAs**

Trusting a single PKI



Trusting Sub-CAs



Trusting Sub-CAs

- Certification Path

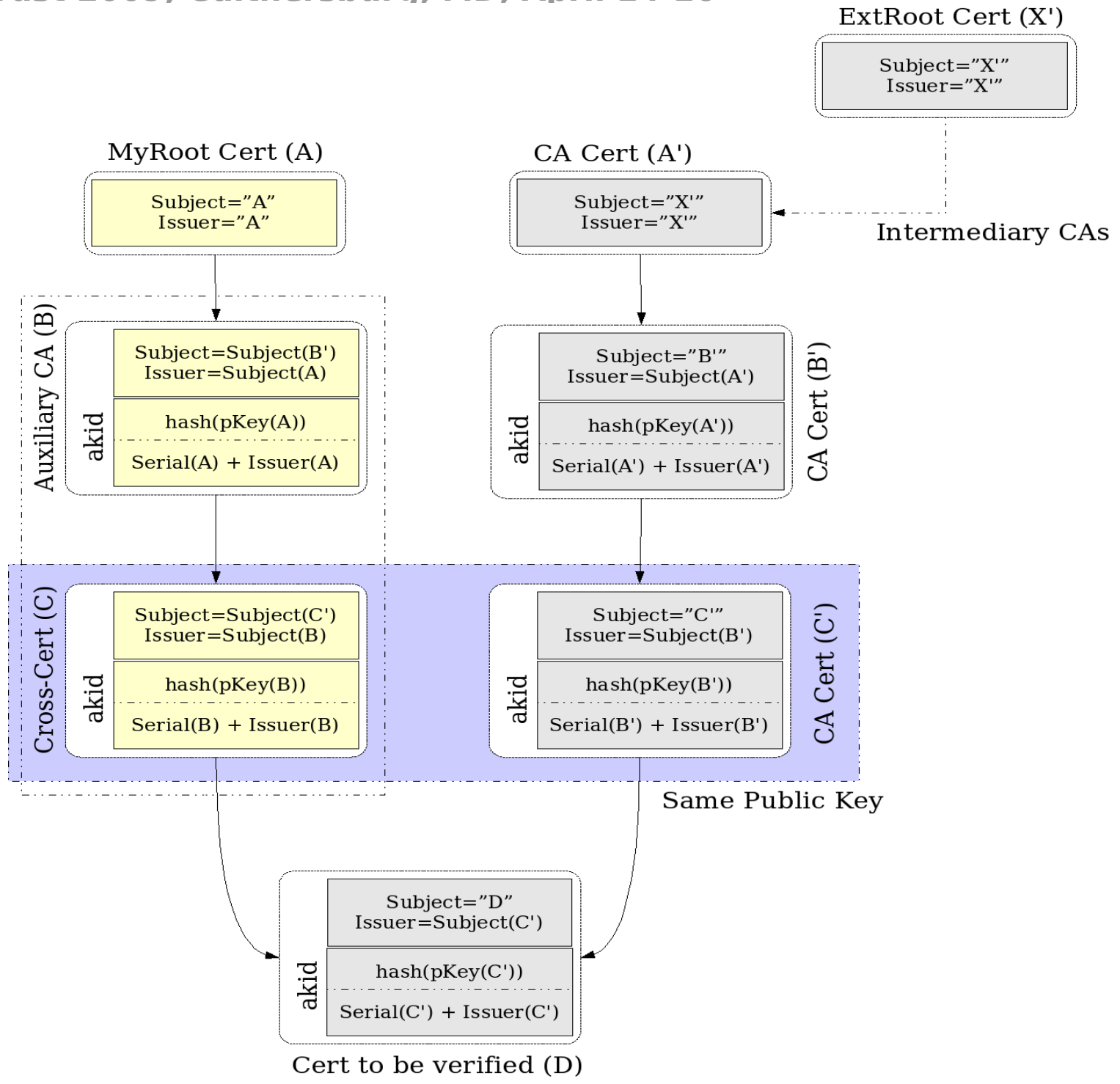
$W \rightarrow Y \rightarrow X$

- Cross certificate Y would be such as:

$\text{subject}(Y) = \text{subject}(\mu)$

- Path Validation may fail (AKID content!)
- Introducing the Auxillary CA

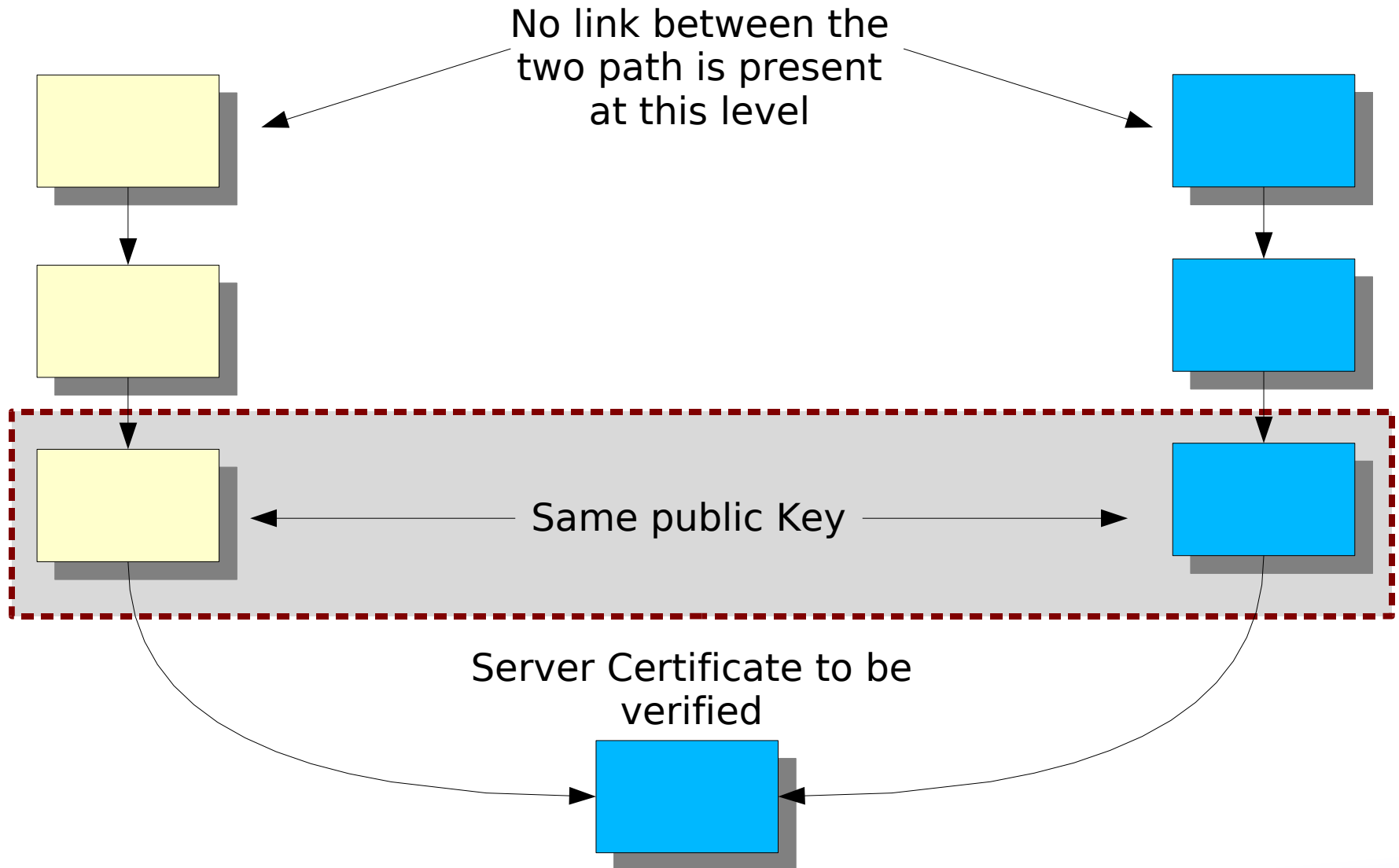
Our Model-2



Application Support

- Fits standard verification protocol
- Supported and works out of the box on different OSes / Crypto Libs:
 - **Windows Systems (XP/2003/Vista)**
 - IE, Outlook
 - **OpenSSL and OpenSSL based software**
 - Konqueror, Kmail
 - **Firefox / Thunderbird**
 - Some exceptions...

SSL/TLS Connections & Sub CAs



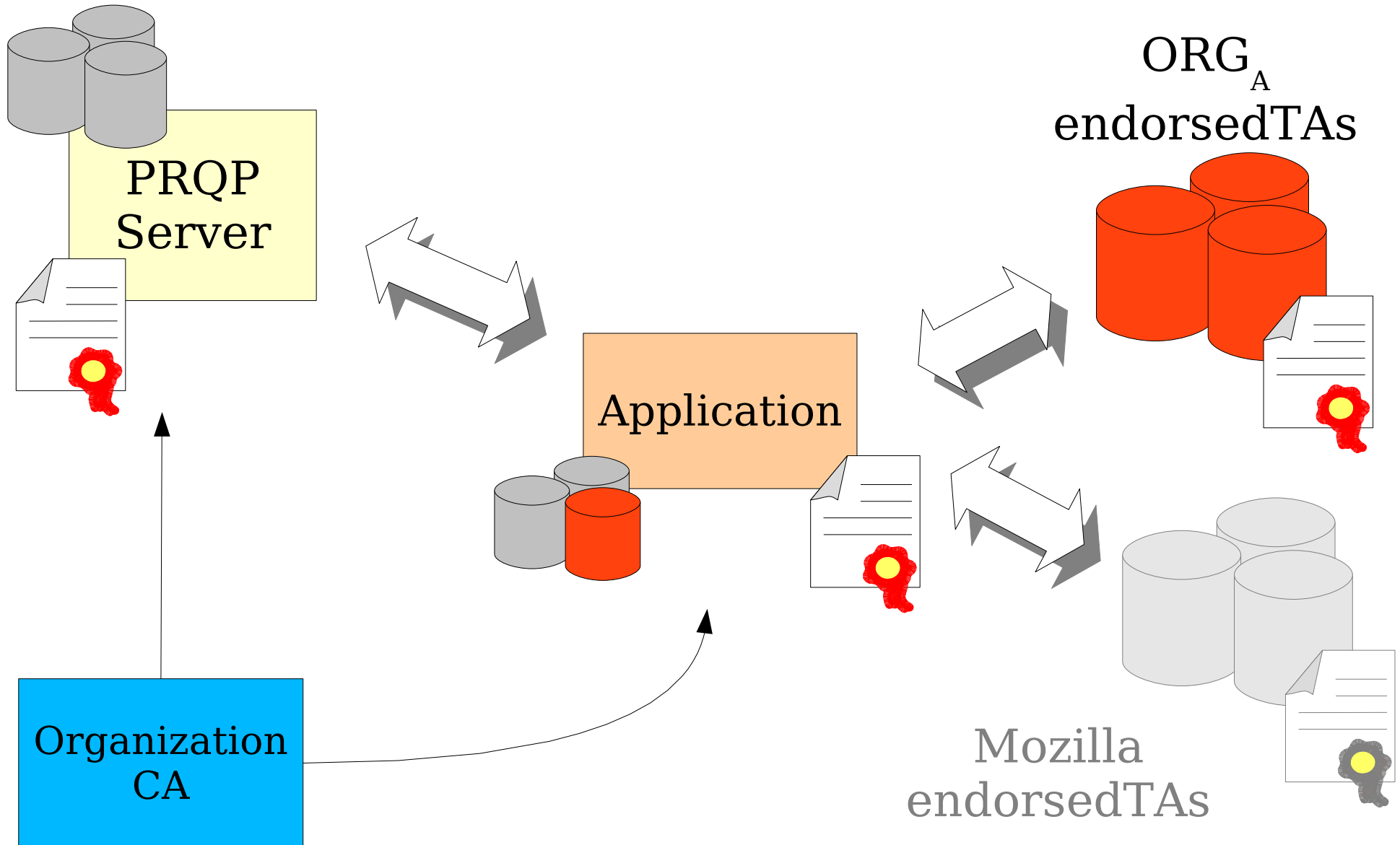
PRQP and Hybrid Trust Model-1

- PRQP provides a discovery system for endorsed TAs
 - Our model provides CA admins with simple certificate issuing / certificate revoking mechanisms for endorsedTAs
- We extended the PRQP with a new OID for endorsedTA pointer

`id-ad-prqp-endorsedTA`

- PRQP server provides pointer(s) to PKCS#7 signed object(s) carrying the endorsed TAs

PRQP and Hybrid Trust Model-2



Achieved Results

- No need of Trust Lists into Applications
 - **One TA (or ApexTA) only**
- Central Management of Trust
 - **Fingerprint verification of trusted certificates is actually (hopefully) performed**
- Trust Model Supported by existing applications
 - **works out of the box on different OSes / Crypto Libs**
- Dynamic support via PRQP integration
 - **endorsedTA**

Questions and Contacts

- Dartmouth College
pala@cs.dartmouth.edu
- OpenCA
madwolf@openca.org
- Website
<http://www.openca.org/projects/prqpd>
<http://www.openca.org/wiki/>



Advances in Browser Security

Anil Saldhana

Anil.Saldhana@redhat.com

The logo for IDtrust 2009. It features the text "IDtrust" in a white, sans-serif font with a black outline, followed by "2009" in a larger, bold, white font with a black outline. The text is set against a yellow background with a blue horizontal stripe running through the middle of the "2009" digits.

About the speaker

- Lead Security Architect, JBoss Division, Red Hat
- Co-editor of W3C Web Security Context Specification (<http://www.w3.org/TR/wsc-ui/>)
 - Targeted for Web User Agents (Browsers)

Overview

- Worldwide browser market
- Topics for Browser Security
- Report Card for the various popular browsers
- W3C WSC-UI Specification
- Tips for secure browsing

Worldwide Browser Market

- Microsoft IE – 67.55%
- Mozilla Firefox – 21.53%
- Apple Safari – 8.29%
- Google Chrome – 1.12%
- Opera – 0.7%

Net Applications Report, Jan 2009

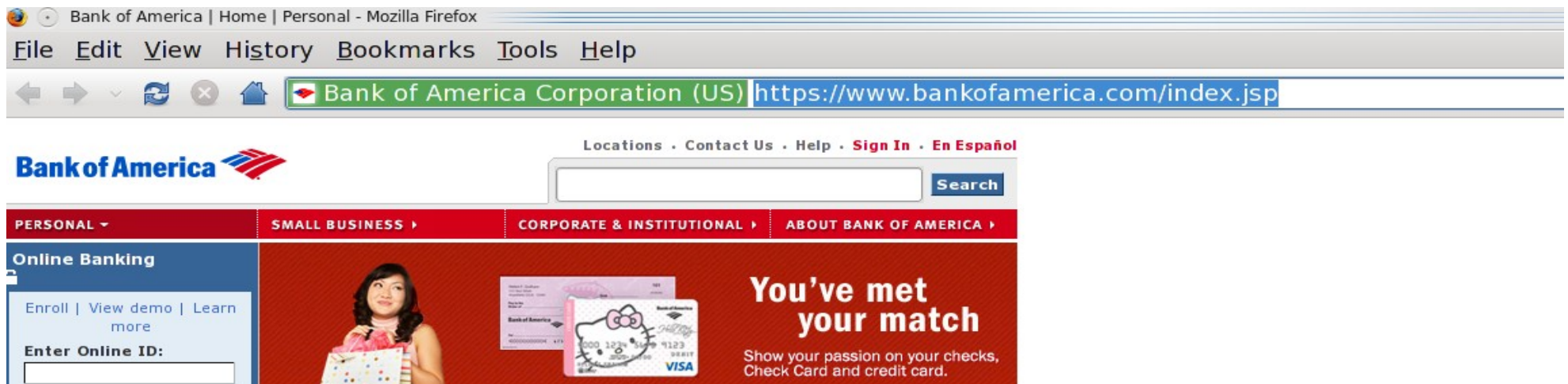
- <http://marketshare.hitslink.com/browser-market-share.aspx?qprid=1>

Topics for Browser Security

- Security Indicators
 - Green Bar (EVCerts)
 - Padlock
- Security Architecture
 - Google Chrome
- Private Browsing
- Plugins
- Phishing and Web Site Vulnerabilities

Security Indicators

- Extended Validation Certificates (EV Certs)
 - Special type of X509 Certificates
 - Certificate Policies extension field (Issuer has a oid)
 - CA does extensive background checks on requester
 - Guidelines issued by CA/Browser Forum



Bank of America | Home | Personal - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Bank of America Corporation (US) https://www.bankofamerica.com/index.jsp

Locations • Contact Us • Help • Sign In • En Español

Bank of America

PERSONAL ▾ SMALL BUSINESS ▸ CORPORATE & INSTITUTIONAL ▸ ABOUT BANK OF AMERICA ▸

Online Banking

Enroll | View demo | Learn more

Enter Online ID:

You've met your match

Show your passion on your checks, Check Card and credit card.

Security Indicators – EV Certs

- CA process for EV Certs
 - Verifying the *legal, physical and operational* existence of the entity
 - Verifying that the *identity* of the entity matches *official records*
 - Verifying that the entity has exclusive right to use the domain specified in the EV Certificate
 - Verifying that the entity has properly authorized the issuance of the EV Certificate

Security Indicators – EV Certs

The image shows a Firefox browser window displaying the British Airways website. The address bar shows the URL <https://www.britishairways.com> with a green lock icon and the text "British Airways, Plc (GB)". A security indicator overlay is present in the center of the page, displaying a green padlock icon and the following text:

You are connected to **britishairways.com** which is run by **British Airways, Plc**.
Hounslow, Middlesex, GB
Verified by: VeriSign, Inc.
Your connection to this web site is encrypted to prevent eavesdropping.

The background website content includes the British Airways logo, navigation links like "Home", "Flights and more", and "Executive Club", a flight search form with fields for "Country of departure" (United Kingdom), "From" (London), and "Class" (Economy), and promotional banners such as "WHAT'S THE WORD ON T5?" and "SALE Business class flights".

Security Indicators – Padlock

- Browser displays Padlock for a HTTPS site
 - Firefox 2 displays a YELLOW address bar.
 - FF3 dropped yellow bar – Tools -> PageInfo
 - Opera displays a yellow bar along with the padlock

Security Architecture

- Google Chrome
 - Two protection domains :
 - Browser Kernel with the OS and
 - Rendering Engine with limited privileges in a sandbox
 - HTML parsing, Javascript VM, DOM : rendering engine.
 - Complex + historical source of security vulnerabilities
 - Browser Kernel
 - Persistent Resources (Cookies/Password DB)
 - OS interaction, user input, network access

“The Security Architecture of the Chromium Browser”,

<http://crypto.stanford.edu/websec/chromium/chromium-security-architecture.pdf>

Security Architecture

- Google Chrome
 - Attacker cannot read/write user file system
 - No malware installation
 - Two protection domains – one for user, one for web
 - 70% of critical browser vulnerabilities avoided
 - 30% cannot be avoided via sandboxing

Private Browsing

- Temporary state where the browser stores no local data – cookies, history
- Use cases
 - Researching a medical condition
 - Surprise vacation/party
 - Internet cafes : shared computers on hourly basis
- Apparently an heavily user demanded feature
- IE8, FF3.1, Opera, Google Chrome and Safari

Plugins

- Typically plugins run outside of the browser process with the full rights of the user.
 - Plugin crash **should not** crash the browser
 - Adobe Flash plugin needs to write flash cookies

Phishing and Web Site Vulnerabilities

- Phishing
 - User taken to a rogue site imitating a legitimate site
 - User enters private information (passwords)
- Web Site Vulnerabilities
 - Cross-site scripting (XSS)
 - Cross-site Request Forging (CSRF)
 - *Confused Deputy Attack against the browser*
 - Header Injection
 - *HTTP headers generated dynamically based on user input*

Phishing and Web Site Vulnerabilities

- Browsers maintain a malware list
 - WARN users when a site is from the list
 - IE8 scheduled to incorporate
 - Google shares its list with Firefox and Chrome
- Tracking Cookies
 - Browsers provide you options to disable 3rd party cookies
 - Safari by default rejects 3rd party cooking

Report Card

	IE	FF	Safari	Chrome	Opera
EV Certs	Y	Y	Y	Y	Y
Padlock	Y	Y	Y	Y	Y
Malware Blacklist	Y	Y	Y	Y	Y
Private Browsing	IE8	FF3.1	Y	Y	Y
Parental Controls	Y	(via addons)	Y	N	(Mini)

W3C WSC Specification

- W3C WSC Working Group
 - W3C, IBM, Mozilla, Opera, Google, Verisign, Oracle, Wells Fargo etc
 - Mission: specify a baseline set of security context information accessible to Web users, and practices for secure and usable presentation of this information, to enable users to come to a better understanding of the context that they are operating in when making *trust* decisions on the Web.
- Targeted for Web User Agents
- <http://www.w3.org/TR/wsc-ui/>

W3C WSC Specification

- Presentation of identity (of website) information
- Error indicators in security protocol
- Augmented Assurance Certificates (EV Certs)
 - Mandatory: Organization (O) attribute of Subject
- Validated Certificates (Known Trust Anchor)
- Mixed Content
- Bookmarking API, Software Installation
- Spec includes Use Cases and Threat Trees

W3C WSC – Threat Trees

- Luring Attacks
 - User taken to a different site than what he believes
- Site Impersonation Attacks
- Cross Site Request Forgery
- Cross Site Scripting
- Network based eaves dropping
 - Session hijacking, credential stealing or private info

Tips for Secure Browsing

- Microsoft Internet Explorer Tips (Source:MS)
 - Set your browser security to **High**
 - Add safe websites to trusted sites
 - Block pop up windows
 - Avoids installation of malicious code

Tips for Secure Browsing

- Websites with plugins containing peer to peer technology may install software/viruses
 - Sites with plugins displaying International TV/sports
- Disable Javascript by default if possible.
 - NoScript firefox extension can enable it for trusted sites
- Lock down browser configuration based on policies
- Tracking Cookies
 - Browser setting to disable auto cookie setting->Block 3rd party cookies

Privacy-Preserving Management of Transactions' Receipts for Mobile Environments

Federica Paci
CS Department
Purdue University
West Lafayette, Indiana
paci@cs.purdue.edu

Kevin Steuer Jr
CS Department
Purdue University
West Lafayette, Indiana
ksteuer@cs.purdue.edu

Ning Shang
CS Department
Purdue University
West Lafayette, Indiana
nshang@cs.purdue.edu

Jungha Woo
CS Department
Purdue University
West Lafayette, Indiana
wooj@cs.purdue.edu

Sam Kerr
CS Department
Purdue University
West Lafayette, Indiana
skerr@cs.purdue.edu

Elisa Bertino
CS Department
Purdue University
West Lafayette, Indiana
bertino@cs.purdue.edu

ABSTRACT

Users increasingly use their mobile devices for electronic transactions to store related information, such as digital receipts. However, such information can be target of several attacks. There are some security issues related to M-commerce: the loss or theft of mobile devices results in a exposure of transaction information; transaction receipts that are send over WI-FI or 3G networks can be easily intercepted; transaction receipts can also be captured via Bluetooth connections without the user's consent; and mobile viruses, worms and Trojan horses can access the transaction information stored on mobile devices if this information is not protected by passwords or PIN numbers. Therefore, assuring privacy and security of transactions' information, as well as of any sensitive information stored on mobile devices is crucial. In this paper, we propose a privacy-preserving approach to manage electronic transaction receipts on mobile devices. The approach is based on the notion of *transaction receipts* issued by service providers upon a successful transaction and combines Pedersen commitment and Zero Knowledge Proof of Knowledge (ZKPK) techniques and Oblivious Commitment-Based Envelope (OCBE) protocols. We have developed a version of such protocol for Near Field Communication (NFC) enabled cellular phones.

Categories and Subject Descriptors

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

General Terms

Security

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDtrust '09, April 14-16, 2009, Gaithersburg, MD Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

Keywords

privacy, transaction record, registrar

1. INTRODUCTION

The combined use of the Internet and mobile technologies (e.g. mobile devices, mobile and wireless communication) is leading to major changes in how individuals communicate, conduct business transactions and access resources and services. People are able to communicate anytime, anywhere with anyone. Technological advances as well as the increased number of mobile applications have resulted in new additions in end-user equipment. Smart mobile devices are equipped with various communication technologies, such as GSM/GPRS, 802.11-WLAN, Bluetooth, NFC and RFID chips as well as GPS for location awareness. Mobile devices today offer a broad spectrum of functions, including web browsers, operating systems (e.g Symbian), environments (e.g., Java virtual machine) for running mobile applications, and e-mail clients.

In such context, establishing mutual trust between users and service providers is critical. A possible approach to establish trust is to view the transactions users have carried out in the past. The history of former transactions informs about users behavior, their ability and dispositions and thus helps to decide whom to trust. Yahoo! Auction, Amazon, eBay are examples of systems that rate both users and service providers based on their past interactions history. Maintaining the history of users' transactions and establishing trust based on these transactions and other factors is a complex task. An important component of any such solution is represented by systems managing *receipts* of transactions. By receipts we refer to information that characterizes a transaction, like the amount paid and the service provider with which the transaction was carried out.

Managing transaction receipts on mobile devices is very challenging. On one hand, the sharing of information about transactions should be facilitated among service providers. A customer should be able to disclose to a service provider a view of his/her past transactions with other service providers in order to get discounts or to prove good behavior over the past. On the other hand, transaction receipts need to be protected as they may convey sensitive information about

a user and can be the target of attacks. Moreover, users should be able to control which service provider has access to information about their past interactions. Assuring privacy and security of transactions' receipts, as well as of any sensitive information, in the context of mobile environments is further complicated by the fact that mobile devices are not secure. Recent statistics [4] show that millions of lost or stolen mobile devices which store users' sensitive data have been reported. In addition to loss or theft, there are an increasing number of viruses, worms and Trojan horses target mobile devices. Moreover, current attacks against Bluetooth and well-known WLAN and GPRS vulnerabilities show that it is very easy for attackers to compromise mobile devices [14]. Another issue is related to how service providers determine whether users are trusted based on their past transactions. Trust establishment should be a policy-driven process. Service providers should specify policies stating the conditions users' transaction receipts must satisfy for a user to be trusted and/or to get a service with favorable conditions. An example such a policy is that a user can receive a discount if he/she has spent \$50 or more. Thus an important requirement is the introduction of a policy language that allows service providers to express conditions against transaction receipts.

To address such issues, we propose a policy-based approach for the management of users transaction history on mobile devices that provides:

1. integrity, confidentiality and privacy of users transaction information;
2. selective and minimal disclosure of transaction information;
3. trust establishment based on transaction history.

Our approach allows a user to prove to a service provider that he/she has performed a transaction satisfying a set of conditions by such service provider without revealing any information about the transaction. The approach is based on the notion of *transaction receipts* issued by service providers upon a successful transaction. Our approach combines Pedersen commitment and Zero Knowledge Proof of Knowledge (ZKPK) techniques and Oblivious Commitment-Based Envelope (OCBE) protocols [6] to assure privacy of information recorded in the receipts. We have developed a version of such an approach for Near Field Communication (NFC) [9] enabled cellular phones. A NFC device embedded in the cellular phone is able to communicate not only with Internet via wireless connections but also with smart card readers. In addition, the cellular phone applications, referred to as MIDlets, can access the phone's tag for reading and writing data.

The rest of the paper is organized as follows. Section 2 introduces the basic notions on which our approach is based. Section 3 presents our privacy-preserving approach to manage transaction receipts; it introduces all key notions of our approach, including the notion of verification policy, and describes our protocols. Section 4 analyzes the properties of our approach. Section 5 introduces the system architecture whereas Section 6 discusses the implementation and reports experimental results. Section 7 overviews related work. Finally, Section 8 concludes the paper and outlines some future work.

2. BASIC NOTIONS

In this section, we introduce the basic cryptographic notions on which our transaction receipts management approach is based.

2.1 Pedersen commitment

The Pedersen Commitment scheme, first introduced in [10], is an unconditionally hiding and computationally binding commitment scheme that is based on the intractability of the discrete logarithm problem.¹ The scheme is originally described with a specific implementation that uses a subgroup of the multiplicative group of a finite field. We remark that this choice of implementation is not intrinsic to the Pedersen commitment scheme itself – it can be implemented with any suitable abelian groups, e.g., elliptic curves over finite fields. Therefore, we rewrite the Pedersen commitment scheme in a more general language as follows.

Pedersen Commitment

Setup

A trusted third party T chooses a finite cyclic group G of large prime order p so that the *computational Diffie-Hellman problem*² 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 finite field \mathbb{F}_p of p elements, which can be represented as the set of integers $\mathbb{F}_p = \{0, 1, \dots, p-1\}$. For a party U to commit a value $x \in \mathbb{F}_p$, it randomly chooses $r \in \mathbb{F}_p$, 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 $c = g^x h^r$.

2.2 Zero-knowledge proof of knowledge (ZKPK) protocol

It turns out that in the Pedersen commitment scheme described above, a party U referred to as the prover, can convince the verifier, V , that U can open a commitment $c = g^x h^r$, without showing the values x and r in clear. Indeed, by following the zero-knowledge proof of knowledge (ZKPK) protocol below, V will learn nothing about the actual values of x and r . This ZKPK protocol, which works for Pedersen commitments, is an adapted version of the zero-knowledge proof protocol proposed by Schnorr [12].

Zero-knowledge proof of knowledge (Schnorr protocol)

As in the case of Pedersen commitment scheme, a trusted party T generates public parameters G, p, g, h . A prover

¹Let G be a (multiplicatively written) cyclic group of order q and let g be a generator of G . The map $\varphi : \mathbb{Z} \rightarrow G, \varphi(n) = g^n$ is a group homomorphism with kernel \mathbb{Z}_m . The problem of computing the inverse map of φ is called the *discrete logarithm problem (DLP) to the base of g* .

²For a cyclic group G (written multiplicatively) of order q , with a generator $g \in G$, the *Computational Diffie-Hellman Problem* is the following problem: Given g^a and g^b for randomly-chosen secret $a, b \in \{0, \dots, q-1\}$, compute g^{ab} .

U who holds private knowledge of values x and r can convince a verifier V that U can open the Pedersen commitment $c = g^x h^r$ as follows.

1. U randomly chooses $y, s \in \mathbb{F}_p^*$, and sends V the element $d = g^y h^s \in G$.
2. V picks a random value $e \in \mathbb{F}_p^*$, and sends e as a challenge to U.
3. U sends $u = y + ex, v = s + er$, both in \mathbb{F}_p , to V.
4. V accepts the proof if and only if $g^u h^v = d \cdot c^e$ in G .

We use this protocol in Section 3.3.3 for proof of receipt ownership.

2.3 OCBE protocols

The Oblivious Commitment-Based Envelope (OCBE) protocols, proposed in [6], provide the capability of enforcing access control policies in an oblivious way. Three communications parties are involved in OCBE protocols: a receiver Re, a sender Se, and a trusted third party T. More precisely, the OCBE protocols ensure that the receiver Re can decrypt a message sent by Se if and only if its committed value satisfies a condition given by a predicate in Se's access control policy, while Se learns nothing about the committed value. The possible predicates are comparison predicates $=, \neq, >, \geq, <$ and \leq .

The OCBE protocols are built with several cryptographic components:

1. The Pedersen commitment scheme.
2. A semantically secure symmetric-key encryption algorithm \mathcal{E} , for example, AES, with key length k -bits. Let $\mathcal{E}_{\text{key}}[M]$ denote the encrypted message M under the encryption algorithm \mathcal{E} with symmetric encryption key Key.
3. A cryptographic hash function $H(\cdot) : \{0, 1\}^* \rightarrow \{0, 1\}^k$. When we write $H(\alpha)$ for an input α in a certain set, we adopt the convention that there is a canonical encoding which encodes α as a bit string, i.e., an element in $\{0, 1\}^*$, without explicitly specifying the encoding.

Given the notation as above, we summarize the EQ-OCBE and GE-OCBE protocols, i.e., the OCBE protocols for $=$ and \geq predicates, respectively, in what follows. The OCBE protocols for other predicates can be derived and described in a similar fashion. The protocols are stated in a slightly different way than in [6], to better suit the presentation in this paper.

EQ-OCBE Protocol Parameter generation

T runs a Pedersen commitment setup protocol to generate system parameters $\text{Param} = \langle G, g, h \rangle$. T also outputs the order of G , p , and $\mathcal{P} = \{\text{EQ}_{x_0} : x_0 \in \mathbb{F}_p\}$, where

$$\text{EQ}_{x_0} : \mathbb{F}_p \rightarrow \{\text{true}, \text{false}\}$$

is an equality predicate such that $\text{EQ}_{x_0}(x)$ is true if and only if $x = x_0$.

Commitment

T first chooses an element $x \in \mathbb{F}_p$ for Re to commit. T then randomly chooses $r \in \mathbb{F}_p$, and computes the Pedersen

commitment $c = g^x h^r$. T sends x, r, c to Re, and sends c to Se.³

Interaction

- Re makes a data service request to Se.
- Based on this request, Se sends an equality predicate $\text{EQ}_{x_0} \in \mathcal{P}$.
- Upon receiving this predicate, Re sends a Pedersen commitment $c = g^x h^r$ to Se.
- Se randomly picks $y \in \mathbb{F}_p^*$, computes $\sigma = (cg^{-x_0})^y$, and sends to Re 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 Se, Re computes $\sigma' = \eta^r$, and decrypts C using $H(\sigma')$.

GE-OCBE Protocol

Parameter generation

As in EQ-OCBE, T runs a Pedersen commitment setup protocol to generate system parameters $\text{Param} = \langle G, g, h \rangle$, and outputs the order of G , p . In addition, T chooses another parameter ℓ , which specifies an upper bound for the length of attribute values, such that $2^\ell < p/2$. T also outputs $\mathcal{V} = \{0, 1, \dots, 2^\ell - 1\} \subset \mathbb{F}_p$, and $\mathcal{P} = \{\text{GE}_{x_0} : x_0 \in \mathcal{V}\}$, where

$$\text{GE}_{x_0} : \mathcal{V} \rightarrow \{\text{true}, \text{false}\}$$

is a predicate such that $\text{GE}_{x_0}(x)$ is true if and only if $x \geq x_0$.

Commitment

This step is the same as EQ-OCBE. T chooses an integer $x \in \mathcal{V}$ for Re to commit. T then randomly chooses $r \in \mathbb{F}_p$, and computes the Pedersen commitment $c = g^x h^r$. T sends x, r, c to Re, and sends c to Se.⁴

Interaction

- Re makes a data service request to Se.
- Based on the request, Se sends to Re a predicate $\text{GE}_{x_0} \in \mathcal{P}$.
- Upon receiving this predicate, Re sends to Se a Pedersen commitment $c = g^x h^r$.
- Let $d = (x - x_0) \pmod{p}$. Re picks $r_1, \dots, r_{\ell-1} \in \mathbb{F}_p$, and sets $r_0 = r - \sum_{i=1}^{\ell-1} 2^i r_i$. If $\text{GE}_{x_0}(x)$ is true, let $d_{\ell-1} \dots d_1 d_0$ be d 's binary representation, with d_0 the lowest bit. Otherwise if GE_{x_0} is false, Re randomly chooses $d_{\ell-1}, \dots, d_1 \in \{0, 1\}$, and sets $d_0 = d - \sum_{i=1}^{\ell-1} 2^i d_i \pmod{p}$. Re computes ℓ commitments $c_i = g^{d_i} h^{r_i}$ for $0 \leq i \leq \ell - 1$, and sends all of them to Se.
- Se checks that $cg^{-x_0} = \prod_{i=0}^{\ell-1} (c_i)^{2^i}$. Se randomly chooses ℓ bit strings $k_0, \dots, k_{\ell-1}$, and sets $k = H(k_0 \parallel \dots \parallel k_{\ell-1})$. Se picks $y \in \mathbb{F}_p^*$, and computes $\eta = h^y, C = \mathcal{E}_k[M]$, where M is the message containing requested data. For each $0 \leq i \leq \ell - 1$ and $j = 0, 1$, Se computes $\sigma_i^j = (c_i g^{-j})^y, C_i^j = H(\sigma_i^j) \oplus k_i$. Se sends to Re the tuple

$$\langle \eta, C_0^0, C_0^1, \dots, C_{\ell-1}^0, C_{\ell-1}^1, C \rangle.$$

³In an offline alternative, T can digitally sign c and sends x, r, c and the signature of c to Re. Then the validity of the commitment c can be ensured by verifying T's signature. In this way, after Se obtains T's public key for signature verification, no communication is needed between T and Se.

⁴Similarly, an offline alternative also works here.

Open

After Re receives the tuple $(\eta, C_0^0, C_0^1, \dots, C_{\ell-1}^0, C_{\ell-1}^1, C)$ from Se as above, Re computes $\sigma'_i = \eta^{r_i}$, and $k'_i = H(\sigma'_i) \oplus C_i^{d_i}$, for $0 \leq i \leq \ell - 1$. Re then computes $k' = H(k'_0 \parallel \dots \parallel k'_{\ell-1})$, and decrypts C using key k' .

LE-OCBE, the OCBE protocol for the \leq predicates, can be constructed in a similar way as GE-OCBE. Other OCBE protocols (for $\neq, <, >$ predicates) can be built on EQ-OCBE, GE-OCBE and LE-OCBE.

All these OCBE protocols guarantee that the receiver Re can decrypt the message sent by Se if and only if the corresponding predicate is evaluated as true at Re's committed value, and that Se does not learn anything about this committed value.

We remark that for certain applications, we can let Se know whether Re's committed value satisfies the specified predicate, by extending the OCBE protocols with one more step: Re shows to Se the decrypted message. We discuss this in more details in Section 3.3.4.

2.4 Shamir's secret sharing scheme

Shamir's (k, n) threshold scheme [13] is a method that divides a secret into n shares and allows the secret to be reconstructed if and only if any k shares are present. Here k and n are both positive integers and $k \leq n$. It is also called Shamir's secret sharing scheme.

The scheme works as follows. A trusted party, T, chooses a finite field \mathbb{F}_p of p elements, with p large enough. Let the secret message S be encoded as an element $a_0 \in \mathbb{F}_p$. T randomly chooses $k - 1$ elements $a_1, \dots, a_{k-1} \in \mathbb{F}_p$, and constructs a degree $k - 1$ polynomial $f(x) = a_0 + a_1x + \dots + a_{k-1}x^{k-1} \in \mathbb{F}_p[x]$. T chooses n elements $\alpha_1, \alpha_2, \dots, \alpha_n \in \mathbb{F}_p$, and creates the secret shares S_i as pairs

$$S_i = (\alpha_i, f(\alpha_i)), 1 \leq i \leq n,$$

where $f(\alpha_i)$ is the polynomial evaluation of f at α_i . Given any subset of k such shares, the polynomial $f(x)$, of degree $k - 1$, can be efficiently reconstructed via interpolation (see, e.g., [5], Section 2.2). The secret S , encoded as the constant coefficient a_0 , is thus recovered.

Shamir's (k, n) threshold scheme has many good properties. Most prominently, it is information theoretically secure, in the sense that the knowledge of less than k shares gives no information about the secret S better than guessing; and it is minimal, in that the size of each share does not exceed the size of the secret. Interested readers can refer to [13] for more details.

3. PROTOCOLS FOR THE RECEIPTS MANAGEMENT

Our approach is based on the notion of *transaction receipts* that are issued by service providers to users upon a successful transaction. In the following sections, we first introduce the notion of transaction receipts, the policy language used by service providers to specify conditions against transaction receipts, and then the privacy-preserving protocol that allow a user to prove the possession of transaction receipts verifying the service provider policies.

3.1 Transaction Receipts

A service provider, upon the completion of a transaction, usually sends the user a receipt that specify a set of in-

TRAN-ID	ATTR	COM	SIG
1234	BUYER	John Smith	7645353 6366363 1124457 6590873 3647688
	SELLER	BookStore.com	1312425 54546
	CATEGORY	Books	2224223 525
	PRICE	30	1341515
	DATE	11-04-2008	1315657

Figure 1: A transaction receipt example

formation about the transaction such as the user identifier, the identifier of the service provider, the item(s) bought, the price paid for the item(s), the quantity, the date of the transaction, and shipment and billing information. We denote this type of information as *transaction attributes*. We consider only a subset of the possible attributes that can be associated with a transaction. The subset includes the user identifier, the service provider identifier, the category to which the item bought belongs to, the item price and the date of the transaction because they are the more relevant attributes to establish trust in the user.

We assume that service providers have a PKI infrastructure that allows them to issue users signed transaction receipts. In particular, we assume that each service provider is associated with a pair of keys (K_{Priv}, K_{Pub}) where K_{Priv} is the private key used to sign the transaction receipts and K_{Pub} is the public key used by other service providers to verify authenticity and integrity of receipts. In order to support a privacy-preserving proof of the possession of such receipts, the transaction receipts released under our protocol include the transactions' attributes in clear and their corresponding Pedersen commitment. The Pedersen commitments of a transaction attributes are used by a user to prove the possession of the receipt of this transaction to other service providers. To compute the Pedersen commitments of the transaction attributes, the service provider runs the Pedersen commitment setup protocol described in Section 2.2 to generate the parameters $\text{Param} = \langle G, g, h \rangle$. Then, the service provider publishes G, p, g and h and its public key K_{Pub} .

The structure of transaction receipts is defined as follows.

DEFINITION 3.1 (TRANSACTION RECEIPT). *Let SP be a service provider and B be a user with which SP has successfully carried out a transaction Tr . Let (G, p, g, h, K_{Pub}) be the public parameters of SP. The receipt for transaction Tr carried out by B and SP is a tuple $\langle \text{TRAN-ID}, \text{ATTR}, \text{COM}, \text{SIG} \rangle$, where TRAN-ID is the transaction identifier; ATTR is the set of transaction attributes {BUYER, SELLER, CATEGORY, PRICE, DATE} where 1) BUYER is the user identifier, 2) SELLER is the service provider's identifier, 3) CATEGORY is the selling category of the item being bought, 4) PRICE is the price of the item and DATE is the date of the transaction, respectively; COM is the set of the Pedersen commitments of the attributes in ATTR. Each element in COM is a tuple of the form $\langle A, \text{COMMIT} \rangle$ where A is the value of an attribute in ATTR and COMMIT is the Pedersen commitment $g^A h^r$ of A and r is a secret known only to B. SIG is the signature of service provider SP on COM⁵. ■*

⁵In what follows, we will use the dot notation to denote the different components of transaction receipt.

EXAMPLE 3.1. Suppose that John Smith has bought for \$ 30 a book from “BookStore.Com” on the 4th of November 2008. A receipt for this transaction, issued according to our protocol, is \langle “1234”, (“John Smith”, BookStore.Com”, “Books”, “\$ 30”, “11-04-2008”), ((BUYER, 45785687994674), (CATEGORY, 76553940894), (PRICE, 2223422262), (DATE, 58300242341)), 1375350748530-50356376037) (see Figure 1).

3.2 Verification Policy Language

Service providers usually evaluate users based on previous transaction interactions with service providers. Based on users’ historical transactions, service providers are able to determine whether a user can be trusted and whether he/she can be qualified to gain some benefits such as a discount or rebate. Service providers define policies, referred to as *verification policies*, to specify the conditions against attributes which are recorded in transaction receipts.

Verification policies are formally defined as follows.

DEFINITION 3.2 (TERM). A *Term* is an expression of the form $Name(attribute_list)$ where: *Name* is the name of a service or discount or an item, whereas *attribute_list* is a possible empty set of attribute names characterizing the service.

DEFINITION 3.3 (ATTRIBUTE CONDITION). An *attribute condition* *Cond* is an expression of the form: “ $name_A \text{ op } l$ ”, where $name_A$ is the name of a transaction attribute *A*, *op* is a comparison operator such as =, <, >, ≤, ≥, ≠, and *l* is a value that can be assumed by attribute *A*. ■

EXAMPLE 3.2. Examples of policies conditions are the following:

- SELLER = “BookStore.Com”
- DATE < “11-04-2008”
- PRICE > \$ 80

DEFINITION 3.4 (VERIFICATION POLICY). A *verification policy* *Pol* is an expression of the form “ $\mathcal{R} \leftarrow Cond_1, Cond_2, \dots, Cond_n$ ”, $n \geq 1$, where \mathcal{R} is a Term and $Cond_1, Cond_2, \dots, Cond_n$ are attribute conditions. ■

Given a transaction receipt \mathcal{Tr} and a verification policy $Pol : \mathcal{R} \leftarrow Cond_1, Cond_2, \dots, Cond_n$, $n \geq 1$, if for each $Cond_i \in Pol$, (ii) $\exists \bar{A} \in \mathcal{Tr}.ATTR$ such that $name_{\bar{A}} = Cond.name_A$ and $value_{\bar{A}}$ satisfies $Cond.(name_A \text{ op } l)$, we say that \mathcal{Tr} satisfies *Pol*, denoted as $\mathcal{Tr} \triangleright Pol$.

EXAMPLE 3.3. An example of verification policy is the following: $Pol : Discount(OnItem = “Glamour”, Amount = “$ 15”) \leftarrow SELLER = “BookStore.Com”, PRICE > “$ 80”, DATE < “11-04-2008”$. The policy states that a user is qualified for a \$ 15 discount on an yearly subscription to Glamour magazine, if the user has spent more than \$ 80 at “BookStore.Com” before “11-04-2008”.

3.3 Protocol to Manage Transaction Receipts

The privacy-preserving protocol proves the possession of a transaction receipt and is carried out between a user and a service provider. The protocol consists of four main phases (see Figure 2):⁶

⁶In what follows we use the term ‘user’; however in practice the steps are carried out by the client software transparently to the actual end user.

1. **Integrity verification of Receipts Attributes.** The user sends a transaction receipt to a service provider to satisfy the service provider verification policy. The service provider verifies the signature on the transaction receipt sent by the user to prove the satisfiability of service provider’s verification policy.
2. **Secret Sharing on the Mobile Phone.** The user reconstructs the secret r that has been used to compute the transaction attribute commitments. Remember that r has been split for better protection from unauthorized accesses.
3. **Proof of Receipt Ownership.** The user proves he/she is the owner of the transaction receipt by carrying out a zero-knowledge proof of knowledge protocol with the service provider.
4. **Verification of Conditions on Receipts.** The service provider verifies that the transaction receipt attributes satisfy its verification policy by carrying out an OCBE protocol with the user.

In the following sections, we describe the details of each phase of the protocol.

3.3.1 Integrity Verification of Receipts Attributes

This phase starts when a user makes a request to a service provider and the service provider sends the user the corresponding verification policy $\mathcal{R} \leftarrow Cond_1, Cond_2, \dots, Cond_n$, $n \geq 1$. First the user selects a transaction receipt \mathcal{Tr} that satisfies such policy. Then, the user sends the service provider $\mathcal{Tr}.COM$, $\mathcal{Tr}.SIG$, $\mathcal{Tr}.ATTR.SELLER$, and the identifier of the service provider which has issued \mathcal{Tr} . The service provider retrieves the public key K_{Pub} of the service provider that has issued \mathcal{Tr} to be able to verify the signature $\mathcal{Tr}.SIG$.

3.3.2 Secret Sharing on the Mobile Phones

In order for a user to be able to carry out ZKPK and OCBE protocols with the service provider, the user needs the random secret r , used to compute the Pedersen commitments of a transaction receipt’s attributes. The security of the protocols strongly depends on r so it is necessary to protect it from unauthorized access that can occur on mobile devices. Mobile device security can be compromised if the device is lost or stolen, or due to the vulnerabilities of the communication network and/or the device software. To prevent these security threats, we adopt Shamir’s secret sharing scheme that allows one to split a secret in n shares and then to reconstruct it if and only if k shares are present. The storage of the shares depends on the specific architecture of the mobile devices. Next we will focus on the Nokia NFC mobile phones that we have used in our implementation.

In our implementation the shares are stored on different mobile phone components and (possibly) on external devices such as a PC or an external storage unit. We split each random secret into four shares s_1, s_2, s_3 and s_4 . The first share s_1 is stored in the internal memory of the mobile phone. The second share s_2 is further split into two secrets. A user chosen PIN number P and a number P' are selected such that $P \oplus P' = s_2$. P' is stored in the phone external memory. The third share s_3 is stored in the smart card integrated in the phone. Finally the fourth secret share s_4 is stored in the user’s PC which has to be accessed remotely by the

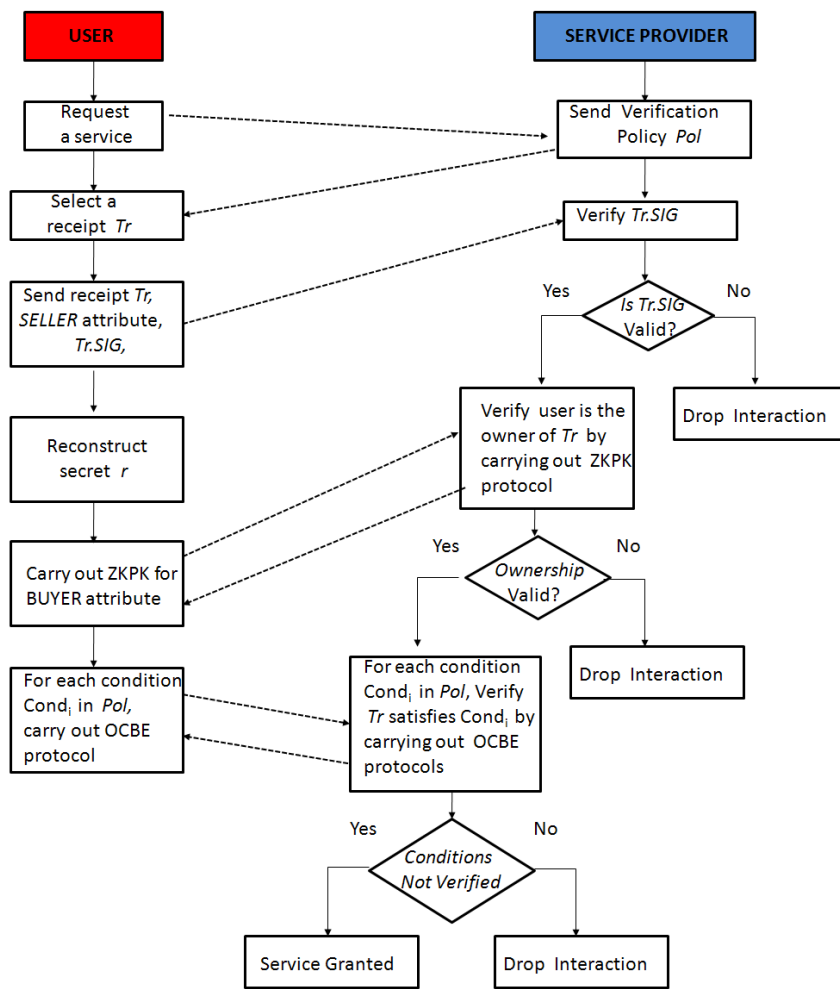


Figure 2: Approach schema

phone. We consider four levels of protection for the secret r that correspond to the number k of shares that are needed to reconstruct r . The possible levels of protection are *low*, *medium*, *medium-high* and *high*. The level of protection *low* requires no splitting of the secret r . In this case, r is stored in the phone smart card. The *medium* level corresponds to a value of k equal to 2. In this case the user has to retrieve two of the four shares s_1, s_2, s_3 and s_4 to obtain the secret r . If the *medium-high* level is chosen, three shares are needed while with level of protection *high*, all the four shares are needed to reconstruct the secret. The level of protection is set by the user⁷ once the issuer of a transaction receipt sends the user the random secret r along with the transaction receipt containing the Pedersen commitments computed using r . Once set, the level of protection cannot be changed by the user.

When the user has to prove the ownership of the transaction receipt sent to the service provider, the r needs to be reconstructed. In order to do that, a number of shares

according to the level of protection set up by the user needs to be retrieved and then combined to obtain r .

EXAMPLE 3.4. Suppose that John Smith has to prove the possession of receipt $\langle "1234", ("John Smith", BookStore.Com", "Books", "$ 30", "11-04-2008"), (\text{BUYER}, 45785687994674), \langle \text{CATEGORY}, 76553940894 \rangle, \langle \text{PRICE}, 2223422262 \rangle, \langle \text{DATE}, 58300242341 \rangle, 137535074853050356376037 \rangle$ to service provider "Borders". In order to accomplish that, John needs to reconstruct the secret r used to compute the Pedersen commitments contained in the receipt. John sets the security level for r to high and to retrieve each secret share he has to perform the following steps:

1. John retrieves s_1 from the phone internal memory.
2. To retrieve s_2 , John inputs the secret PIN number P using the phone keypad. P' is retrieved from the phone external memory and it is used to compute the second secret share $s_2 = P \oplus P'$.
3. John retrieves the secret s_3 from the phone smart card.
4. To retrieve the secret share s_4 stored at the user's PC, John connects to its PC by using the phone

⁷The specification of the security level and the entering of the PIN are the only steps that need to be carried by the actual end-user. The security level can however be set as a default and the end-user does not need to enter it each time it receives a new receipt.

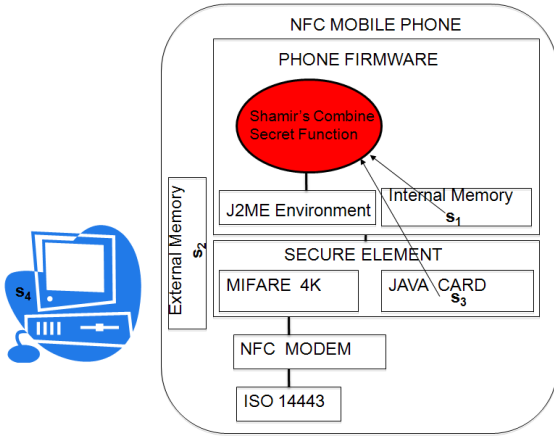


Figure 3: Random Secret Reconstruction

By contrast if John sets up a medium security level, he has to retrieve only two shares to obtain the secret r . For example, John can decide to get the shares s_1 and s_3 from the phone's internal memory and the phone smart card respectively without having to insert any PIN number (see Figure 3).

3.3.3 Proof of Receipt Ownership

Once the user has reconstructed the random secret r , the proof of the ownership of the transaction receipt can be achieved by engaging a ZPK protocol for the BUYER transaction attribute with the service provider. According to the ZPK protocol, the user randomly picks y, s in $\{1, \dots, p\}$, computes $d = g^y h^s$, where g and h are the public parameters of the service provider. The user then sends d to the service provider. Once received d , the service provider sends back a random challenge $e \in \{1, \dots, p-1\}$ to the client. Then the user computes $u = y + em$ and $v = s + er$ where m is the value of the BUYER transaction attribute and r is the random secret, and sends u and v to the service provider. The service provider accepts the aggregated zero knowledge proof if $g^u h^v = dc^e$. Otherwise, the interaction with the user is dropped.

3.3.4 Verification of Conditions on Receipts

We consider two scenarios that require the verification of conditions on transaction receipts. In the first scenario, a service provider provides a general service to all qualified users, and does not require to know the outcome of the transaction. For example, a book store may provide a transferable 10%-off coupon code to any user who presents a receipt showing a purchase of a product in the "Books" category. However, the book store does not care whether this coupon code is successfully received by the user; it only cares that a coupon code is valid when being used. The book store simply rejects a receipt if it is shown twice, to prevent a user from taking advantage of this offer for multiple times. In such a scenario, the OCBE protocols, (cfr. Section 2.3) can be used directly. Let the user be the receiver Re , and the service provider be the sender Se . Re sends a service request to Se , and Se responds with its verification policy. Based on the policy, Re selects a receipt Tr which satisfies Se 's policy, and sends $Tr.COM$, $Tr.SIG$, and the value of

SELLER attribute to Se . Se chooses the message M , as described in Section 2.3, to be the content of service (e.g., a coupon code). Then, it composes the envelope using the corresponding attribute value in the received receipt for M , and sends it to Re . Re can open the envelope if and only if the involved attribute value on the receipt satisfies the condition specified in the policy, but Se will not know if Re can open the envelope.

In the second scenario, the service provider needs to know the result of the condition verification, i.e., it should be informed if the attributes on the user's receipt satisfies the specified policy. There are many instances of such a scenario. For example, the service provider may require its policy be satisfied by a user's receipt in order to continue the transactions. In this case, for user privacy protection, the OCBE protocol for equality predicates, EQ-OCBE, should not be employed, because the service provider will be able to infer the attribute value if the verification is successful. However, other OCBE protocols which are for inequality predicates can still be used, with one more step appended to the protocol, described next.

In this additional step, the service provider acts as the sender Se , and the user acts as the receiver Re . The service provider chooses the message M to be a random bit string, which will be used as a secret of Se . The OCBE protocol for inequality predicates is executed between Se and Re , based on Se 's policy and the involved attribute value recorded in Re 's receipt, for this secret M . At the end of the protocol, after opening the envelope, Re shows Se the decrypted message M' . The attribute on the receipt passes Se 's verification if $M = M'$, or fails if otherwise. The service provider continues with the transactions in the former case, or aborts the transaction in the latter case. Such additional step has been added to the OCBE protocols, to allow the service provider to learn the result of the verification, at the user's will. Since the random bit string M contains no useful information about the service content itself, a qualified user must choose to show the correctly decrypted secret message M , in order to continue the transactions with the service provider. In this sense, the extended OCBE protocols (for inequality predicates) works as a zero-knowledge proof scheme for our application.

In both scenarios, if the user's receipt's attributes need to satisfy multiple conditions in the service provider's policy, a run of the OCBE protocol must be performed for each condition. A receipt's attributes satisfy the conditions in the policy if and only if the user can open all related envelopes.

4. PROTOCOL ANALYSIS

In this section we analyze the security properties of our transaction receipts management protocol.

Our protocol is built on provably secure cryptographic protocols: digital signature scheme, Shamir's secret sharing scheme, Pedersen commitment, Schnorr's zero-knowledge proof protocol, and OCBE protocols.

After a user sends a service request to a service provider and receives a policy, he/she selects a transaction receipt Tr , and sends back $Tr.COM$, $Tr.SIG$ and $Tr.ATTR.SELLER$, i.e., the receipt's parts containing Pedersen commitments, receipt issuer (seller)'s signature on these commitments and the identity of the issuer, respectively. On one hand, since the service provider verifies the issuer's signature on the Pedersen commitments, it is guaranteed that the Pedersen com-

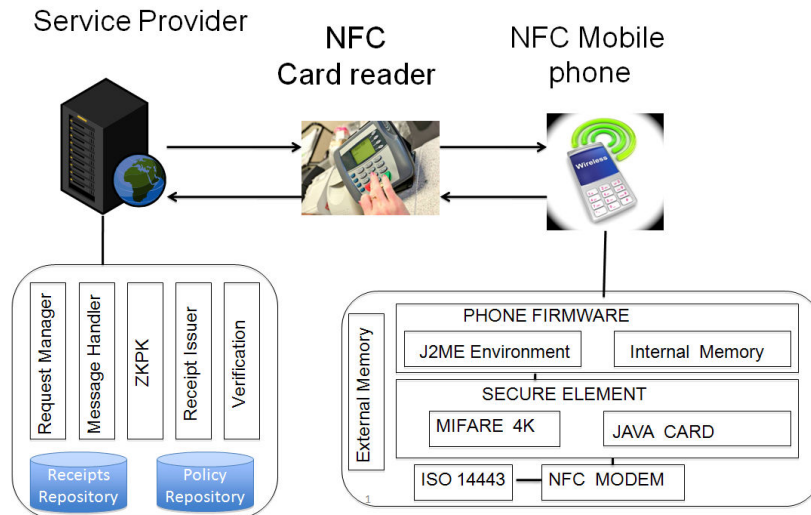


Figure 4: System architecture

mitments have not been modified. Thus, the integrity of the Pedersen commitments is assured. On the other hand, the service provider does not learn anything about the actual values of the transaction attributes. This is due to the unconditionally hiding property of the Pedersen commitment.

If the user passes the first step above, he/she starts to reconstruct the secret exponent r , which is used to prove the ownership of a receipt and the verification of conditions on receipts, from some of the shares s_1, s_2, s_3 and s_4 using Shamir's (k, n) threshold scheme. The number of shares needed for the reconstruction depends on the pre-defined level of protection. Since the shares are distributed at different locations, and protected by a PIN number, this makes it hard for a party other than the receipt owner to obtain all needed shares to recover r . Furthermore, since the Shamir's threshold scheme is information theoretically secure, unless enough shares are collected, any attempt to recover the secret r is not easier than guessing.

Once the secret r is reconstructed, the user carries out a zero-knowledge proof protocol for the BUYER attribute, in a manner like Schnorr's as described in Section 3.3.3, with the service provider. The user is able to convince the service provider that he/she knows how to open the commitment, only if he/she knows the values of both x and r such that the corresponding commitment is computed as $g^x h^r$. It prevents an entity who steals a valid receipt but does not know how to open the asked commitment in the receipt from authenticating with the service provider. Due to the zero-knowledge property of the protocol, the service provider does not learn the attribute value x for BUYER.

The last step of our protocol is the execution of the OCBE protocols for the verification of the conditions on the receipt attribute values. The OCBE protocols guarantee that a user can correctly retrieve a message, randomly chosen by the service provider, if and only if the user knows how to open the commitments whose committed values satisfy the conditions (equality or inequality) in the service provider's policy, while the service provider learns nothing about the actual values of the transaction attributes.

Based on the above considerations, our protocol guaran-

tees the integrity and the privacy of the information included in a transaction receipt and it also protects users against identity theft.

5. SYSTEM ARCHITECTURE

We have implemented our protocol on Nokia 6131 NFC [3] mobile phones. NFC enabled devices are gaining popularity because they provide easy-to-use mechanisms for ubiquitous accesses to systems and services. Based on a short-range wireless connectivity, the communication is activated by bringing two NFC compatible devices or tags within a few centimeters from one another.

The system architecture is shown in Figure 4. It consists of three main components: a service provider application, an external NFC reader and the Nokia 6131 NFC [3] mobile phone. The core architectural component is the NFC mobile phone. It consists of an **Antenna**, for detecting external targets such as tags, external readers, or other Nokia 6131 NFC mobile phones; an **NFC modem**, for providing the capability to send and receive commands between antenna, secure element and phone firmware including J2ME environment; a **Secure element**, for enabling third-party application development using tag/card emulation; **Phone firmware**, for providing mobile phone functions with NFC features; a **SIM card**, for GSM subscription identification and service management; **J2ME environment** included in phone firmware, for enabling third-party application development using Nokia 6131 NFC features; and an **External memory**.

The **Secure element** within Nokia 6131 NFC can store information securely, which can be used for payment and ticketing applications or for access control and electronic identifications. **Secure element** is divided into two sub-components, **Java Card** area (also referred to as smart card) and **Mifare 4K** area. **Mifare 4K** area can be considered as a memory with access control, and typically it is simpler to implement than a smart card application. **Mifare 4K** contains data, whereas smart card application contains an executable program. **Java Card** provides high security environment and executes code, which means it can be used for more complex applications. Therefore, we store in the

Java Card some of the shares in which the random secret r is split because of the high security provided by **Java Card**. **Secure element** is accessible through **NFC modem** internally from MIDLets and externally by external readers. MIDLets are Java applications running in the J2ME environment. In the next section we describe in details, how we have implemented our protocol to manage receipts by using MIDLets.

The NFC reader enables the communication between the service provider application and the mobile phone. It transmits and receives messages from the NFC cellular phone. The service provider application consists of five main modules: **Request Manager**, **Message Handler**, **ZKPK**, **Receipt Issuance** and **Verification**. The **Request Manager** module parses users requests and selects from a local repository the verification policy that applies to the request. The **Message Handler** module provides all functions supporting the communications between the service provider application and the external NFC reader. The **ZKPK** module supports the verification of receipts' integrity and the ZKPK protocol to verify the BUYER attribute. The **Receipt Issuance** module provides the functions for creating a transaction receipt, such as the generation of the Pedersen commitments and the signature of the commitments. Once created, the transaction receipts are stored in a local repository. The **Verification** module supports the steps for the verification of conditions on receipts described in Section 3.3.4.

6. IMPLEMENTATION AND EXPERIMENTAL EVALUATION

To evaluate the performance of our protocol, we have developed a prototype version of the system. We have implemented a MIDLet that supports the integrity verification of receipts attributes, the proof of receipt ownership and the verification of conditions against receipts. The implementation of the secret sharing phase is under development.

We store users' transaction receipts in the external phone memory, whereas the secret r used to compute the secure commitments included in the receipts is saved in the **Java Card** component. The execution of the MIDLet is triggered when the **Mifare 4K** captures the verification policy sent by the service provider's external NFC reader and the **Mifare 4K** transfers such policy to the phone main memory. The MIDLet retrieves from the external memory a transaction receipt that satisfies the service provider policy and sends the part of the receipt containing the transaction attributes commitments, the signature affixed on the commitments, and the value of SELLER attribute to **Mifare 4K** so that can be read by the service provider's external NFC reader. If the service provider application successfully verifies the signature on the receipts commitments, the MIDLet retrieves the secret r from the **Java Card**, and performs the other steps of the receipts management protocol.

The MIDLet runs on Java 2 Micro Edition (J2ME). Since J2ME is aimed at hardware with limited resources, it contains a minimum set of class libraries for specific types of hardware. In our implementation on conventional non-mobile platforms, we used the **BigInteger** and **SecureRandom** class, defined in J2SE `java.math` and `java.security` packages respectively, to implement secure commitments, but both packages are not supported in J2ME. Therefore, we have used the third-party cryptography provider **BouncyCastle** [2], a lightweight cryptography APIs for Java and C# that pro-

vide implementation of the **BigInteger** and **SecureRandom** classes. In addition, because of the limited memory size of mobile phones, we reduced the MIDLet's code size by using code obfuscation techniques provided by Sun's **NetBeans IDE**. Code obfuscation allows one to reduce a file size by replacing all Java packages and class names with meaningless characters. For example, a file of a size of 844KB can be reduced to a size of 17KB.

We have also implemented the service provider component as a web application using Java and the **Apache Tomcat Application Server**. The current implementation of the **Verification** module only supports the EQ-OCBE and GE-OCBE protocols for the verification of equality conditions and inequality conditions expressed by using the \geq comparison operator. We are extending the implementation with support for other comparison operators.

We have performed several experiments to evaluate the execution time of the MIDLet and the service provider (SP for short) application for the proof of the receipt ownership and the verification of conditions (equality and inequality) against receipts. We have collected data about the execution times for verifying the equality conditions on receipts and the time for verifying the inequality conditions by using respectively EQ-OCBE and GE-OCBE protocols by varying the value of parameter ℓ from 5 to 20. ℓ determines the number of commitments $c_i = g^{d_i} h^{r_i}$, $0 \leq i \leq \ell - 1$ that the user has to send to the service provider to prove he/she satisfies an inequality condition in service provider policy.

The experiment compares the envelope creation time at the service provider's side, and the envelope opening time at the MIDLet's side, which are the most computationally expensive part for both protocols. We also record the time required for generating the additional Pedersen commitments c_i in GE-OCBE at the MIDLet's side. No additional commitment needs to be generated by the user in EQ-OCBE. We do not include the communication time and the symmetric encryption time in the comparisons, which vary with different network settings and plaintext lengths, in order to focus on the main operations of the protocols. We also do not include the signature verification time in the comparison, for the same reason.

In the experiment, we have executed the verification protocol both at the service provider's side and at the MIDLet size, for 10 times, and we have computed the average of the obtained values.

	Verification of Receipt Ownership
MIDLet	0.042
SP's Application	0.0311

Table 1: Average time (in seconds) to verify the ownership of a receipt at MIDLet's side and at SP's side

Table 1 shows the execution times taken by the verification of receipt ownership phase at MIDLet side and at the SP application side.

	Commitments Creation	Opening Envelope	Total Execution Time
Equality Condition	0	1.126	1.126
Inequality Condition (\geq)	5.875	6.088	11.963

Table 2: Verification of conditions' execution time (in seconds) at MIDLet's side ($\ell = 5$)

	Envelope Creation
Equality Condition	0.0409
Inequality Condition (\geq)	0.165

Table 3: SP's application's average execution time (in seconds) for verifying one condition ($\ell = 5$)

Table 2 and Table 3 report the average verification of conditions' execution time taken, respectively, by the MIDLet and by the SP application for a value of parameter ℓ equal to 5. When multiple conditions are to be verified, the execution time increases accordingly, as the protocol is repeated for multiple rounds. As expected, the execution time to verify inequality conditions takes more time than the verification of equality conditions. In fact, the GE-OCBE used to verify inequality condition with comparison predicate \geq , requires the MIDLet and the SP application to perform more interactions steps. Figure 5 shows the SP application's execution time to create the envelope according to GE-OCBE protocol while Figure 6 shows the time taken by the MIDLet to open the envelope. In both cases, we have varied the value of ℓ parameter from 5 to 20. The graphs show how the value of parameter ℓ dramatically impacts on verification of conditions' execution time. With the increasing of the ℓ parameter values, the execution time linearly increases. The verification time increases because when ℓ parameter increases, the SP application has to compute a higher number of $\sigma_i^j = (c_i g^{-j})^y, C_i^j = H(\sigma_i^j) \oplus k_i$ to be sent to the MIDLet running on user's mobile phone and the MIDLet to decrypt the envelope has to compute a higher number of $\sigma_i' = \eta^{r_i}$, and $k_i' = H(\sigma_i') \oplus C_i^{d_i}$, for $0 \leq i \leq \ell - 1$.

Therefore, in the implementation of our protocol, the parameter ℓ must be kept as small as possible in order to reduce the computational cost.

We expect other OCBE protocols for inequality predicates to give performance results similar to those of GE-OCBE, because the design and operations are similar.

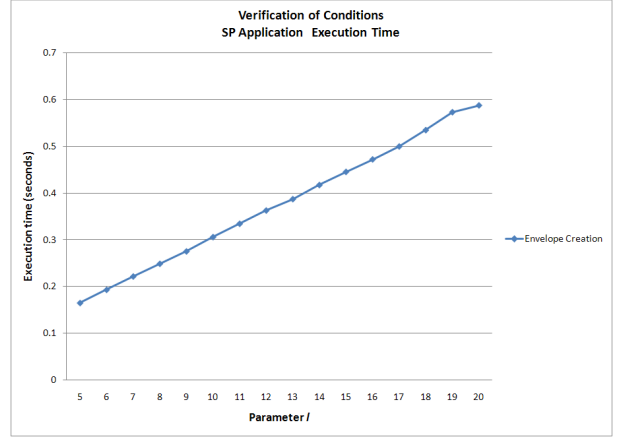


Figure 5: SP Application's Envelope Creation Time varying the value of parameter ℓ

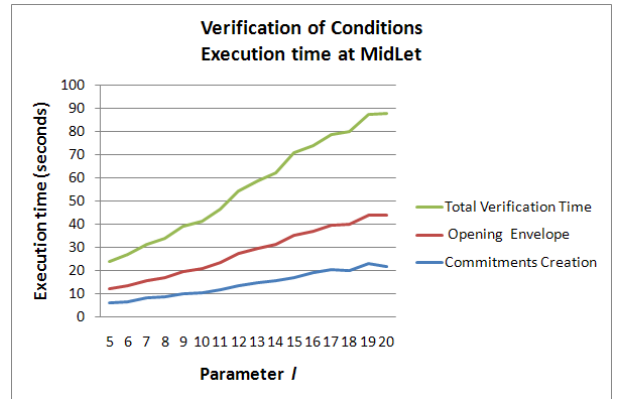


Figure 6: MIDLet's Envelope Opening Time varying the value of parameter ℓ

7. RELATED WORK

In this section, we compare our approach with other approaches for mobile transactions managers.

With the advent of high-speed data networks and feature-rich mobile, the concept of *mobile wallet* [8, 1] has gained importance. The ESPRIT project CAFE [1] has introduced the notion of electronic wallet, that is a small portable device which manages off-line digital payments to be used in commercial transactions. The electronic wallet transacts via a short range infrared channel either directly with compliant cash registers and wallets held by other individuals, or over the Internet, to merchants' tills or service points provided by banks and other organizations. The electronic wallet relies on a blind signature scheme to guarantee privacy and unlinkability for the electronic payment information while our approach preserves only the privacy of transactions information.

Mjolsnes et al. [8] have proposed a version of the electronic wallet for online payments. The authors exploits a credential server, denoted as *credential keeper* that securely stores the credentials issued to a user by different issuers. The credentials represents the wallet of the user. The user to access his/her credentials at the credential keeper and provide them to a service provider, has to present an access credential, e.g. a symmetric key, to the keeper server. To increase even more security, the access credential is encrypted and protected within a mobile device, and it can only be activated by using a PIN code or some other authentication method. In our approach we do not need a third component to guarantee a secure storage and management of the information included in a transaction receipt. The receipts can be securely stored on the phone external memory because the values of the transaction attributes are not stored in clear but they are substituted by their Pedersen commitments.

The Secure Electronic Transaction (SET) [11] protocol was developed to allow credit card holders to make transactions without revealing their credit card numbers to merchants and also to assure authenticity of the parties. SET deploys dual signature for merchant and payment gateway. Each party can only read a message designated for itself since each message is encrypted for a different target. To enable this feature, card holders and merchants must register with a Certificate Authority before they exchanging a SET message. SET assures both confidentiality and integrity of the messages among card holders, merchants and payment gateway whereas our protocol is designed to assure integrity and privacy of transactions information. SET authenticates the identity of the cardholder and the merchant to each other because both are registered with the same certificate authority. However, our protocols do not mandate this requirement. SET is considered to have failed because of its complexity. It requires cardholders and merchants to register in advance and get X.509 certificates to make transactions whereas the users need not to have such PKI certificate in our protocol. In our approach only service providers need to have a PKI certificate.

More recently, Veijalainen et al. [15], propose an approach to manage transaction on mobile devices. Their solution is based on the use of an application running on the phone denoted as *Mobile Commerce Transaction Manager* that provides the functionalities to start, terminate and resume a transaction with a service provider. With respect to security and privacy of transactions information, the *Mobile*

Commerce Transaction Manager only guarantees confidentiality by encrypting the messages exchanged between the service provider application and the application running on the phone. In our approach by using digital signatures, Pedersen commitment, ZKPK techniques and OCBE protocols, we are able to guarantee both privacy and integrity of transactions information.

Finally, MeT initiative [7] has the goal to develop secure and easy methods and platforms for conducting e-commerce transactions on mobile phones. The strategy for MeT is to base the framework on existing standards such as WAP, Wireless Transport Layer Security (WTLS), Wireless Identification Module (WIM), Public Key Infrastructure (PKI) and Bluetooth. Privacy and security are ensured with digital signatures and cryptography services for transaction verification, confidentiality, authentication, and non-repudiation.

8. CONCLUSIONS

We have proposed a privacy preserving approach to manage electronic transaction receipts on mobile devices. We have focused on such type of device because we believe that in the near future users will conduct business transactions and access resources and services mostly using their mobile phones and PDAs. However, we have also implemented a web-based version of our receipt management system.

Our approach is based on the notion of *transaction receipt*, that records the information characterizing a transaction, and combines Pedersen commitment, ZKPK techniques and OCBE protocols. We have implemented our approach on Nokia 6131 NFC mobile phones and have evaluated its performance of on these devices. The experimental results show that our protocol is quite efficient in verifying equality conditions on receipts; however we need to improve the performance of the inequality conditions' verification. We believe that the reasons for the high execution times when verifying inequality conditions are the limited computational capability of Nokia 6131 NFC mobile phones and the use of the BouncyCastle API that are not natively supported by these phones. We plan to test our protocol on the Nokia 6212 NFC mobile phones that support JSR-177 Security and Trust APIs. These APIs provide security services to J2ME enabled devices without the need of using BouncyCastle API's. Since these API's are natively supported by these kind of phones, we believe that our protocols should perform better on such phones. We are currently completing the implementation of our prototype system by developing a MIDlet supporting the secret sharing phase of our protocol. We plan to complete the implementation of service provider application's **Verification** module in order to support the verification of inequality conditions containing the comparison predicates $<$, $>$ and \neq .

9. ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under the ITR Grant No. 0428554 "The Design and Use of Digital Identities", by the AFOSR grant A9550-08-1-0260, and by the U.S. Department of Homeland Security under Grant Award Number 2006-CS-001-000001, under the auspices of the Institute for Information Infrastructure Protection (I3P) research program. The I3P is managed by Dartmouth College. 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 the U.S. Department of Homeland Security, the I3P, or Dartmouth College.

10. REFERENCES

- [1] J-P Boly, A. Bosselaers, R. Cramer, R. Michelsen, S. Fr. Mjolsnes, F. Muller, T.P. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallee, and M. Waidner. The ESPRIT project CAFE - high security digital payment systems. In *ESORICS*, pages 217–230, 1994.
- [2] Bouncy Castle Crypto APIs. <http://www.bouncycastle.org/>.
- [3] Nokia Forum. Nokia 6131 NFC Technical Description. <http://www.forum.nokia.com>.
- [4] Help for lost and stolen phones. <http://news.bbc.co.uk/1/hi/technology/4033461.stm>.
- [5] W. Gautschi. *Numerical Analysis: An Introduction*. Birkhauser Boston Inc., Cambridge, MA, USA, 1997.
- [6] J. Li and N. Li. OACerts: Oblivious attribute certificates. *IEEE Transactions on Dependable and Secure Computing*, 3(4):340–352, 2006.
- [7] Met initiative. <http://www.mobiletransaction.org>.
- [8] S.F. Mjolsnes and C. Rong. Localized credentials for server assisted mobile wallet. *ICCNMC'01: International Conference on Computer Networks and Mobile Computing*, 00:203, 2001.
- [9] Near Field Communication Forum. <http://www.nfc-forum.org>.
- [10] 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, 1991.
- [11] SET- Secure Electronic Transaction specification book 1: Business description, 1997. 1992. Springer-Verlag.
- [12] C-P Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO '89: Proceedings of the 9th Annual International Cryptology Conference on Advances in Cryptology*, pages 239–252, London, UK, 1990. Springer-Verlag.
- [13] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.
- [14] TechRepublic. Identify and reduce mobile device security risks. http://articles.techrepublic.com.com/5100-22_11-5274902.html.
- [15] J. Veijalainen, V. Y. Terziyan, and H. Tirri. Transaction management for m-commerce at a mobile terminal. *Electronic Commerce Research and Applications*, 5(3):229–245, 2006.

Privacy-Preserving Management of Transactions' Receipts for Mobile Environments

Federica Paci, Ning Shang, Sam Kerr,
Kevin Steuer Jr., Jungha Woo, Elisa Bertino

Purdue University

April 15, 2009

Offline Shopping



Online Shopping



How Can Receipts Help?



Receipts Can Help

- Establish transaction-history-based trust
- Facilitate services such as discounts/promotions



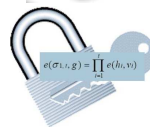
Challenge in e-Commerce: Receipt Management



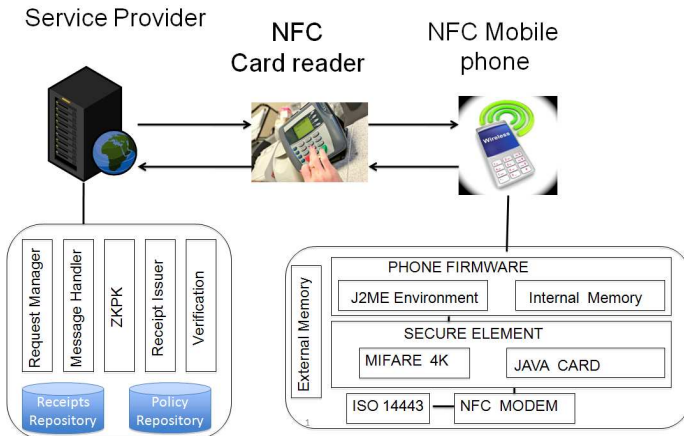
- Customer needs to get e-receipts from offline stores
- Customer needs to show online transactions to offline stores
- Privacy & security

Solution: M-Commerce and Cryptography

- Customer-SP communications
 - Cell phones (NFC)
- Privacy-preserving management & proofs
 - Digital signatures
 - Zero-knowledge proof of knowledge (ZKPK)
 - Oblivious commitment-based envelope (OCBE)
 - Shamir's secret sharing scheme



System Architecture



Near Field Communication (NFC) Technology



"Touch" to Become the New "Click"

Get information by touching smart posters!



Your NFC device is your ticket!



Your NFC device is your travel card!



TOUCH



Buy goods from vending machines with your phone!



Get information about your current job or task!



Your NFC device is your credit card!




ADVANCING NEAR FIELD COMMUNICATION TECHNOLOGY

© NFC Forum, Inc.

6

Courtesy <http://www.nfc-forum.org>

Near Field Communication (NFC) Technology



**NFC
FORUM**

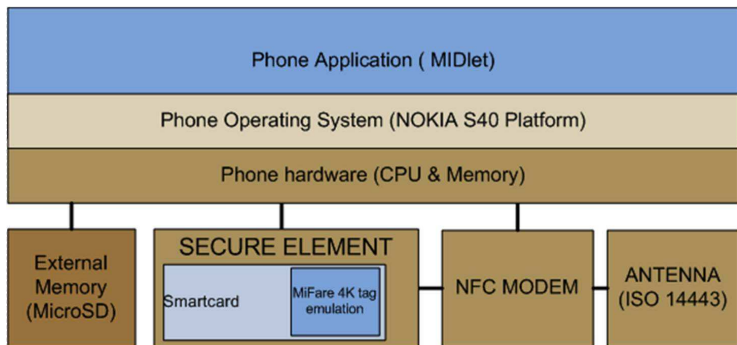
NFC Is Fast, Private and Easy

	NFC	RFID	IrDa	Bluetooth
Set-up time	<0.1ms	<0.1ms	~0.5s	~6 sec
Range	Up to 10cm	Up to 3m	Up to 5m	Up to 30m
Usability	Human centric Easy, intuitive, fast	Item centric Easy	Data centric Easy	Data centric Medium
Selectivity	High, given, security	Partly given	Line of sight	Who are you?
Use cases	Pay, get access, share, initiate service, easy set up	Item tracking	Control & exchange data	Network for data exchange, headset
Consumer experience	Touch, wave, simply connect	Get information	Easy	Configuration needed

© NFC Forum, Inc. 4

Courtesy <http://www.nfc-forum.org>

Nokia 6131 NFC Phone Architecture



Receipt Format

Public Param = $\langle G, p, g, h \rangle$

Pedersen commitment (COM): $c = g^{\text{attr-value}} h^r$

TRAN-ID	ATTR		COM		SIG
1234	BUYER	John Smith	BUYER	7645353 6366363	1124457 6590873 3647688
	SELLER	BookStore.com	SELLER	1312425 54546	
	CATEGORY	Books	CATEGORY	2224223 525	
	PRICE	30	PRICE	1341515	
	DATE	11-04-2008	DATE	1315657	

Integrity Verification: Digital Signatures



Service provider verifies digital signature according to “SELLER” attribute in receipt.



Options which allow signature aggregation

- Batch RSA signatures
 - Fast
 - Good if there is only one signer
- Boneh's aggregate signatures with bilinear maps (elliptic curve pairings)
 - Good for case of multiple signers
 - Slower than batch RSA

Ownership Proof: ZKPK



Service provider performs a ZKPK protocol with user (phone) on “BUYER” attribute in receipt.



ZKPK can

- convince SP that user knows the values name and r (authentication)
- prevent SP from learning the values name and r in clear text (privacy)

ZKPK (Schnorr's Scheme)

Public Param = $\langle G, p, g, h \rangle$



secret r
 $c = g^{\text{name}} h^r$



ZKPK (Schnorr's Scheme)

Public Param = $\langle G, p, g, h \rangle$

$(1).c, d = g^y h^s$

secret r
 $c = g^{\text{name}} h^r$



ZKPK (Schnorr's Scheme)

Public Param = $\langle G, p, g, h \rangle$

secret r
 $c = g^{\text{name}} h^r$



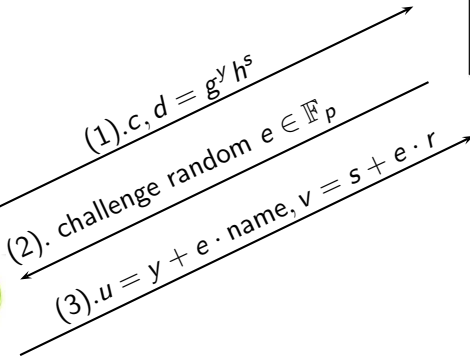
(1). $c, d = g^y h^s$
(2). challenge random $e \in \mathbb{F}_p$



ZKPK (Schnorr's Scheme)

Public Param = $\langle G, p, g, h \rangle$

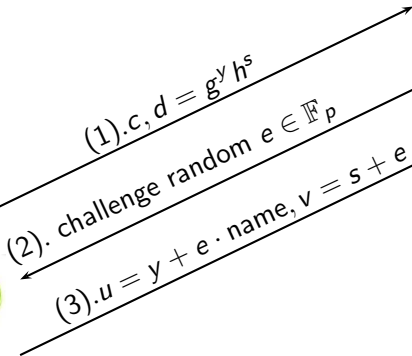
secret r
 $c = g^{\text{name}} h^r$



ZKPK (Schnorr's Scheme)

Public Param = $\langle G, p, g, h \rangle$

secret r
 $c = g^{\text{name}} h^r$

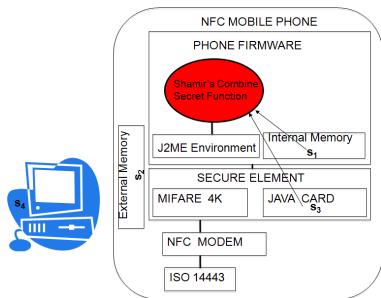


(4) accepts if $g^u h^v = dc^e$

Strong Protection of Secret Value: Shamir's Secret Sharing

(n, k) -threshold scheme

$n = 4, k = 2/3/4$ (security level L/M/H)



Secret value r can be re-constructed only if k shares are present

Verification of Conditions: OCBE Protocols



Service provider performs OCBE protocols with user (phone) to verify whether user satisfies conditions on attributes specified in policy



OCBE can

- convince SP that user's attribute values satisfy conditions given by comparison predicates (authentication)
- prevent SP from learning user's attribute values in clear text (privacy)

Verification of Conditions: Policy Language

Verification Policy Language: Example

```
Pol : Discount(OnItem = "Glamour", Amount = "$15")  
      ← SELLER = "bookstore.com", PRICE > "$80", DATE <  
      "11-04-2008."
```

The policy states that a user is qualified for a \$15 discount on an yearly subscription to Glamour magazine, if the user has spent more than \$80 at "bookstore.com" before the date "11-04-2008".

EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$



secret r
 $c = g^{\text{ctgry}} h^r$



EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$



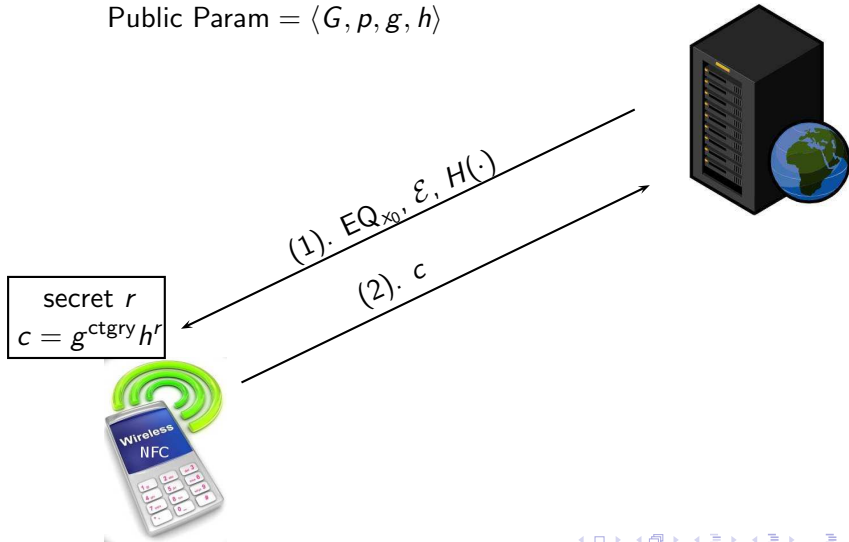
(1). $EQ_{x_0, \epsilon, H(\cdot)}$

secret r
 $c = g^{\text{ctgry}} h^r$



EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$



EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$

secret r
 $c = g^{\text{ctgry}} h^r$



(1). $EQ_{x_0}, \epsilon, H(\cdot)$

(2). c



(3). $y \xleftarrow{R} \mathbb{F}_q,$
 $\sigma = (cg^{-x_0})^y$

EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$

secret r
 $c = g^{\text{ctgry}} h^r$



(1). $EQ_{x_0}, \mathcal{E}, H(\cdot)$

(2). c

(4). $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{coupon}]$

(3). $y \xleftarrow{R} \mathbb{F}_q,$
 $\sigma = (cg^{-x_0})^y$

EQ-OCBE: Equality Predicates (Li & Li)

Public Param = $\langle G, p, g, h \rangle$

secret r
 $c = g^{\text{ctgry}} h^r$



(1). $EQ_{x_0}, \mathcal{E}, H(\cdot)$

(2). c

(4). $\eta = h^y, C = \mathcal{E}_{H(\sigma)}[\text{coupon}]$

(5). $\sigma' = \eta^r$, decrypts C with $H(\sigma)$ to get coupon



(3). $y \xleftarrow{R} \mathbb{F}_q,$
 $\sigma = (cg^{-x_0})^y$

GE-OCBE: “ \geq ” Predicates (Li & Li)

GE-OCBE

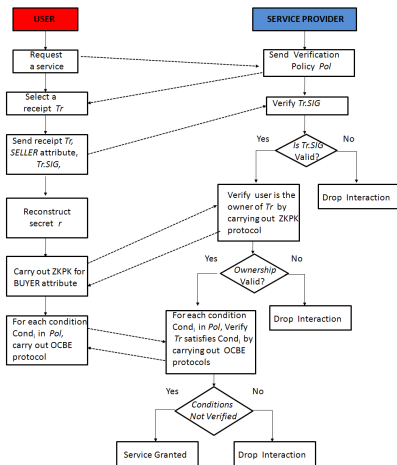


- Similar to EQ-OCBE
 - bit-by-bit fashion
- More computationally costly than EQ-OCBE
 - parameter ℓ controls capacity and efficiency



Other OCBE protocols are similar.

Protocol Overview



ZKPK Performance

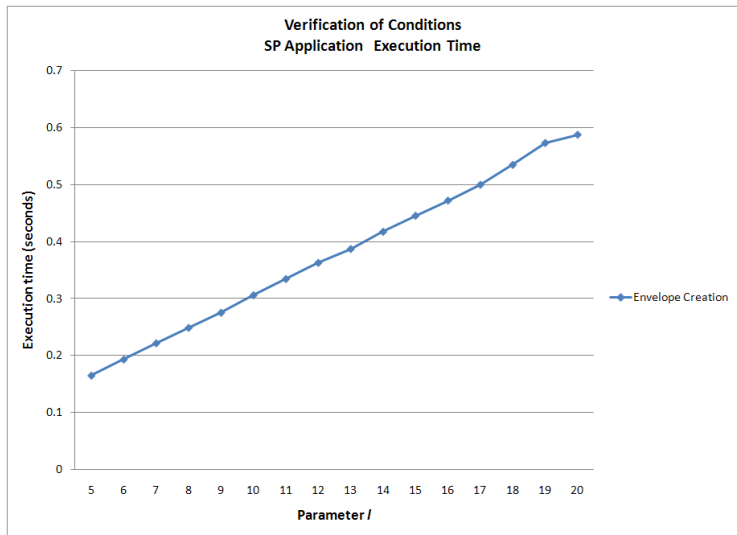


Time for receipt ownership verification via ZKPK

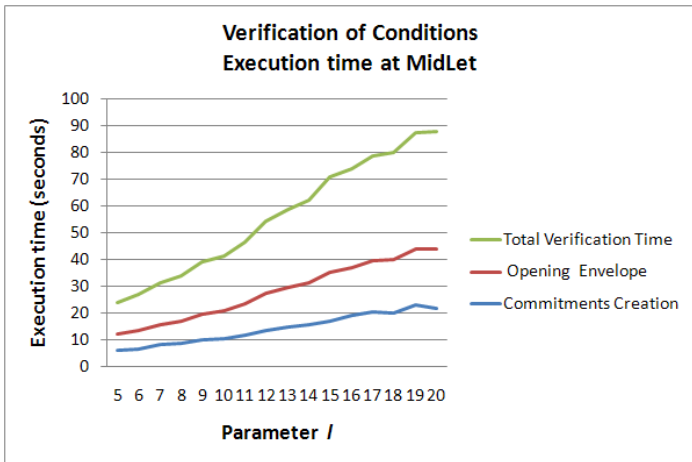
Customer MIDLet: 0.042 second

Service Provider Application: 0.0311 second

OCBE Performance on Service Provider



OCBE Performance on NFC Phone



Nokia 6131: ARM-9 228 MHz, JVM Interpreter (JBenchmark estimate)

VeryIDX Framework

The VeryIDX IdM Team at Purdue
<http://veryidx.cs.purdue.edu>

The screenshot shows the homepage of the VeryIDX Research Group at Purdue University. The page has a yellow header with the title "VeryIDX Research Group at Purdue University" and navigation links for "Edit this page" and "Old revisions". Below the header, it indicates the user is logged in as "Ning Shang" with links for "Update Profile", "Logout", and "Admin". There is a search bar and an "Index" section with a logo for VeryIDX. The main content area features sections for "VeryIDX Research Group", "What is VeryIDX?", "Goals", and "Application Scenarios".

VeryIDX Research Group at Purdue University

You are here: [public:start](#) - [public](#) Edit this page Old revisions

Logged in as: Ning Shang
[Update Profile](#) | [Logout](#) | [Admin](#)

search

Index

Homepage
Faculty
Participants
Internal

VeryIDX Research Group

Digital identity can be defined as the digital representation of the information known about a specific individual or organization. As such, it encompasses not only login names (often referred to as nyms), but many additional information, referred to as identity attributes or identifiers. The management of identity attributes raises a number of challenges, due to conflicting requirements. On the one hand, identity attributes need to be shared to speed up and facilitate authentication of users and access control. On the other hand, they need to be protected as they may convey sensitive information about an individual and can be a target of attacks like identity theft. Here, by identity theft we mean the act of impersonating others' identities by presenting stolen identifiers or proofs of identities. The problem of identity theft, that is, the act of impersonating others' identities by presenting stolen identifiers or proofs of identities, has been receiving increasing attention because of its high financial and social costs. In our project we address the problem of verification of such identifiers and proofs of identity by developing a solution for federated organizations.

What is VeryIDX?

VeryIDX is digital identity management framework based on the concept of privacy preserving multi-factor verification.

Goals

The VeryIDX project explores research issues concerning the privacy-preserving and secure management of digital identities, including multi-factor authentication of identity attributes, identity interoperability, and provenance. Other topics being explored in the project include context-aware authentication and access control, privacy-preserving content distribution techniques based on identity attributes, and identity management for health care applications. Results of our research are implemented in the VeryIDX system and in other prototypes.

Application Scenarios

- Electronic Healthcare

Acknowledgements

- The I3P Consortium
- The CERIAS of Purdue University

The End



Questions?

nshang@cs.purdue.edu

Quantum Resistant Public Key Cryptography: A Survey

Ray A. Perlner
ray.perlner@nist.gov

David A. Cooper
david.cooper@nist.gov

National Institute of Standards and Technology
100 Bureau Drive
Gaithersburg, Maryland 20899–8930

ABSTRACT

Public key cryptography is widely used to secure transactions over the Internet. However, advances in quantum computers threaten to undermine the security assumptions upon which currently used public key cryptographic algorithms are based. In this paper, we provide a survey of some of the public key cryptographic algorithms that have been developed that, while not currently in widespread use, are believed to be resistant to quantum computing based attacks and discuss some of the issues that protocol designers may need to consider if there is a need to deploy these algorithms at some point in the future.

Categories and Subject Descriptors

E.3 [Data]: Data Encryption—*Public key cryptosystems*

General Terms

Algorithms, Security

Keywords

Quantum computers, public key cryptography

1. INTRODUCTION

Since its invention, public key cryptography has evolved from a mathematical curiosity to an indispensable part of our IT infrastructure. It has been used to verify the authenticity of software and legal records, to protect financial transactions, and to protect the transactions of millions of Internet users on a daily basis.

Through most of its history, including present day, public key cryptography has been dominated by two major families of cryptographic primitives: primitives whose security is believed to be contingent on the difficulty of the integer factorization problem, such as RSA [46] and Rabin-Williams [44, 55], and primitives whose security is believed to be contingent on the difficulty of the discrete logarithm problem, such as the Diffie-Hellman key exchange [14], El Gamal signatures [19], and the Digital Signature Algorithm (DSA) [17]. Also included within the second family is elliptic curve cryptography (ECC) [32, 40], which includes all known, practi-

cal identity-based encryption schemes [5] as well as pairing-based short signatures [6].

While both the integer factorization problem and the general discrete logarithm problem are believed to be hard in classical computation models, it has been shown that neither problem is hard in the quantum computation model. It has been suggested by Feynman [16] and demonstrated by Deutsch and Jozsa [13] that certain computations can be physically realized by quantum mechanical systems with an exponentially lower time complexity than would be required in the classical model of computation. A scalable system capable of reliably performing the extra quantum operations necessary for these computations is known as a quantum computer.

The possibility of quantum computation became relevant to cryptography in 1994, when Shor demonstrated efficient quantum algorithms for factoring and the computation of discrete logarithms [51]. It has therefore become clear that a quantum computer would render all widely used public key cryptography insecure.

While Shor demonstrated that cryptographic algorithms whose security relies on the intractability of the integer factorization problem or the general discrete logarithm problem could be broken using quantum computers, more recent research has demonstrated the limitations of quantum computers [47]. While Grover developed a quantum search algorithm that provides a quadratic speedup relative to search algorithms designed for classical computers [24], Bennet, Bernstein, Brassard, and Vazirani demonstrated that quantum computers cannot provide an exponential speedup for search algorithms, suggesting that symmetric encryption algorithms, one-way functions, and cryptographic hash algorithms should be resistant to attacks based on quantum computing [4]. This research also demonstrates that it is unlikely that efficient quantum algorithms will be found for a class of problems, known as NP-hard problems, loosely related to both search problems and certain proposed cryptographic primitives discussed later in this paper.

The above research suggests that there is no reason, at the moment, to believe that current symmetric encryption and hash algorithms will need to be replaced in order to protect against quantum computing based attacks. Thus, any effort to ensure the future viability of cryptographic protocols in the presence of large scale quantum computers needs to concentrate on public key cryptography. Given how vital public key trust models are to the security architecture of today's Internet, it is imperative that we examine alternatives to the currently used public key cryptographic primitives.

This paper is authored by employees of the U.S. Government and is in the public domain.

IDTrust '09, April 14–16, 2009, Gaithersburg, MD
ACM 978-1-60558-474-4

In this paper, we provide an overview of some of the public key cryptographic algorithms that have been developed that are believed to be resistant to quantum computing based attacks. The purported quantum-resistance of these algorithms is based on the lack of any known attacks on the cryptographic primitives in question, or solutions to related problems, in the quantum computation model. This does not mean that an attack will never be found, but it does yield some confidence. The same type of argument is used to justify the security of all but a handful of cryptographic primitives in the classical computation model. One-time pads [50, 53] and universal hash functions [8] are unconditionally secure in any computation model, if used properly, but they are usually impractical to use in a way that doesn't invalidate the proof. Other cryptography often comes with a "security proof," but these proofs are generally based on at least one unproved security assumption—virtually any proof of security in the classical or quantum computation model not based on an unproved assumption would resolve one of the best known unsolved problems in all of mathematics [10].

Section 2 lists some of the issues that should be considered in comparing public key cryptographic algorithms. Section 3 describes a one-time signature scheme known as Lamport signatures, and Section 4 describes techniques that have been developed for creating long-term signature schemes from one-time signature schemes. Section 5 covers public key cryptographic algorithms based on lattices. Section 6 describes the McEliece signature and encryption schemes. Other potential areas of research are mentioned in Section 7 and Section 8 discusses issues that may need to be considered by protocol designers if one or more of the public key cryptographic algorithms described in this paper become widely used at some point in the future.

2. GENERAL CONCERNS

A number of factors can be considered when examining the practicality of a public key cryptographic algorithm. Among these are:

- Lengths of public keys, key exchange messages, and signatures: For public key cryptographic algorithms commonly in use today, these are all roughly the same size, ranging from a few hundred to a few thousand bits, depending on the algorithm. This is not always the case for candidate quantum-resistant algorithms. If public keys, key exchange messages, or signatures are much larger than a few thousand bits, problems can be created for devices that have limited memory or bandwidth.
- Private key lifetime: A transcript of signed messages often reveals information about the signer's private key. This effectively limits the number of messages that can safely be signed with the same key. The most extreme example of this is the Lamport signature scheme, discussed below, which requires a new key for each signed message. Methods have been developed for creating a long-term signature scheme from a short-term or even single-use signature scheme, but these often require extra memory for managing and storing temporary keys, and they tend to increase the effective length of signatures. Private keys used for decryption do not generally have limited lifetime, since

encryption does not use and therefore cannot leak information about the private key, and protocols can almost always be designed to prevent the decryptor from revealing information about his or her private key. This can be done by encrypting symmetric keys rather than the content itself, using integrity protection, and reporting decryption failures in a way that makes them indistinguishable from message authentication code (MAC) failures. This type of behavior is currently necessary for secure protocols using old RSA padding schemes, and is often considered good practice regardless of the key transfer mechanism.

- Computational cost: There are four basic public key operations: encryption, decryption, signing, and signature verification. On today's platforms, with currently used algorithms, these operations generally take a few milliseconds, except for RSA encryption and signature verification, which can be about 100 times faster due to the use of small public exponents. Key generation time may also be a concern if it is significantly more expensive than the basic cryptographic operations. Factoring based schemes such as RSA and Rabin-Williams tend to have this problem, as generation of the two high entropy prime factors requires several seconds of computation.

3. LAMPORT SIGNATURES

The basic idea behind Lamport signatures [33] is fairly simple. However, there is a wide variety of performance tradeoffs and optimizations associated with it. It derives its security strength from the irreversibility of an arbitrary one-way function, f . f may be a cryptographic hash function, although the scheme is secure even if f is not collision resistant. The Lamport scheme is a one-time signature scheme. In order for the scheme to be secure, a new public key must be distributed for each signed message.

In the simplest variant of Lamport signatures, the signer generates two high-entropy secrets, $S_{0,k}$ and $S_{1,k}$, for each bit location, k , in the message digest that will be used for signatures. These secrets ($2n$ secrets are required if the digest is n bits long) comprise the private key. The public key consists of the images of the secrets under f , i.e., $f(S_{0,k})$ and $f(S_{1,k})$, concatenated together in a prescribed order (lexicographically by subscript for example). In order to sign a message, the signer reveals half of the secrets, chosen as follows: if bit k is a zero, the secret $S_{0,k}$ is revealed, and if it is one, $S_{1,k}$ is revealed. The revealed secrets, concatenated together, comprise the signature. While the act of signing a message clearly leaks information about the private key, it does not leak enough information to allow an attacker to sign additional messages with different digests. Nonetheless, there is no way in general for the signer to use this type of public key to safely sign more than one message.

While conceptually the simplest, the above scheme is not the most efficient way to create a one-time signature scheme from a one-way function [20]. Firstly, the size of public keys and signatures can be reduced by nearly a factor of two, merely by using a more efficient method of choosing which secrets to reveal from a smaller pool. For each bit location, k , rather than creating two secrets, $S_{0,k}$ and $S_{1,k}$, the secret key may consist of only $S_{0,k}$, with the public key being $f(S_{0,k})$. In order to sign a message, the signer would reveal

Digest	Digest								Counter	
	6	3	F	1	E	9	0	B	3	D
Signature	$f^6(S_0)$	$f^3(S_1)$		$f(S_3)$	$f^{14}(S_4)$	$f^9(S_5)$	S_6	$f^{11}(S_7)$	$f^3(S_8)$	$f^{13}(S_9)$
Public Key	$f^{15}(S_0)$	$f^{15}(S_1)$	$f^{15}(S_2)$	$f^{15}(S_3)$	$f^{15}(S_4)$	$f^{15}(S_5)$	$f^{15}(S_6)$	$f^{15}(S_7)$	$f^{15}(S_8)$	$f^{15}(S_9)$

Figure 1: A Sample Lamport Signature with $b = 16$

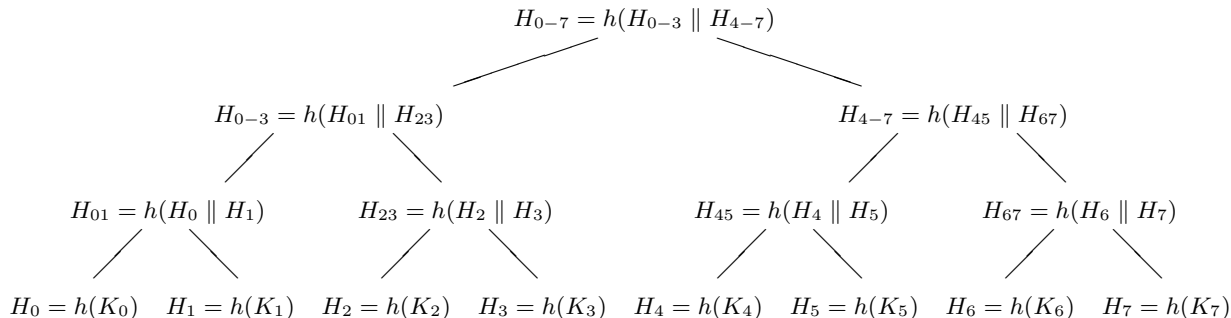


Figure 2: Merkle Hash Tree

$S_{0,k}$ for each bit position, k , in the message digest that has a value of zero. Thus, the signature would be the concatenation of $S_{0,k}$ for each bit location in the message digest that has a value of zero. The problem with this scheme is that an attacker could try to change the value of a signature by withholding some of the $S_{0,k}$ values, thus changing some of the zero bits to one. In order to protect against this, a binary encoding of the total number of zero bits in the message digest may be appended to the message digest. This counter would be signed along with the message digest as described above. Since an attacker could only try to change zero bits to one, the attacker could not reduce the value of the counter, which would be necessary to successfully change some of the zero bits to one in the message digest itself.

The sizes of signatures and public keys can also be traded off against computation by using hash chains. In such a scheme, the message digests would be encoded using digits with a base b that is greater than two (e.g., using hexadecimal digits, which would correspond to $b = 16$). To sign the k th digit of the digest, N_k , the private key would be S_k , the public key would be the result of applying a one-way function, f , to the secret $b - 1$ times, $f^{b-1}(S_k)$, and the signature value would be $f^{N_k}(S_k)$.¹ Thus if b were 4 and N_k were 1, then public key would be $f^3(S_k) = f(f(f(S_k)))$ and the signature value would be $f^1(S_k) = f(S_k)$. As with the binary scheme, there would be a need to append a “counter” to the message digest in order to prevent an attacker from increasing the values of any digits in the message digest. The value of the counter to be appended to the digest, for an n digit digest, would be $\sum_{k=0}^{n-1} (b - 1 - N_k)$. The reduction in signature size is logarithmic in the value of the base, while the cost of generating a one-time key pair is linear, so this process reaches diminishing returns fairly quickly, but using a base of 16 is often better than a base of 2. Figure 1 shows an example of a Lamport signature for a message digest that

consists of eight hexadecimal digits.

Analysis of the performance of Lamport’s one-time signatures is somewhat prone to confusion. As discussed above, the performance is dependent upon the choice of a one-way function and on the value of the base, b , used in generating the public key. Further, as the scheme is a one-time signature scheme the distinction between signing time and key generation time is not terribly useful, although it does provide a lot of opportunities for a signer to do precomputation. Nonetheless, with a fairly reasonable set of assumptions (e.g., $f = \text{SHA-256}$ with $b = 4$) one arrives at signature, verification, and key generation times that are similar to current schemes such as DSA.

4. LONG-TERM SIGNING KEYS FOR ONE-TIME SIGNATURE SCHEMES

If the signer can precompute a large number of single-use, public key - private key pairs, then at little additional cost, these keys can be used to generate signatures that can all be verified using the same public key [36]. Moreover, the long-term public key associated with this scheme need only be the size of a message digest. To do this, we use hash trees, a technique invented by Ralph Merkle in 1979 [35]. At the bottom of the tree, the one-time public keys are hashed once and then hashed together in pairs. Then those hash values are hashed together in pairs, and the resulting hash values are hashed together and so on, until all the public keys have been used to generate a single hash value, which will be used as the long-term public key. In this scheme, the signer can prove that a one-time public key was used in the computation that generated the long-term public key by providing just one hash value for each level of the tree—the overhead is therefore logarithmic in the number of leaves in the tree.

Figure 2 depicts a hash tree containing eight single-use public keys. The eight keys are each hashed to form the leaves of the tree, the eight leaf values are hashed in pairs to create the next level up in the tree. These four hash

¹As with the binary scheme above, the signer would not need to reveal the signature value for any digit k for which $N_k = b - 1$.

values are again hashed in pairs to create H_{0-3} and H_{4-7} , which are hashed together to create the long-term public key, H_{0-7} . In order for an entity to verify a message signed using K_0 , the signer would need to provide H_1 , H_{23} , and H_{4-7} in addition to K_0 and a certified copy of H_{0-7} . The verifier would compute $H'_0 = h(K_0)$, $H'_{01} = h(H'_0 \parallel H_1)$, $H'_{0-3} = h(H'_{01} \parallel H_{23})$, and $H'_{0-7} = h(H'_{0-3} \parallel H_{4-7})$. If H'_{0-7} is the same as the certified copy of H_{0-7} , then K_0 may be used to verify the message signature.

While the the number of additional hashes that need to be added to a public key grows logarithmically with the number of leaves in the tree, the cost of generating a hash tree is linear in the number of leaves. It may therefore be desirable to limit the size of hash trees. If the signer wishes to use a single public key to sign more messages than the number of single-use key pairs he or she is willing to generate in the process of generating a public key, then the signer may wish to use a certificate chain like construction where the longest term public key is used to sign a large number of shorter-term keys, which in turn are used to sign even shorter term keys and so on. The advantage of this is that short-term keys can be generated as needed, allowing the cost of generating new one-time keys to be distributed over the lifetime of the single long-term key. This technique can also be used for other signature schemes where the key has limited lifetime, not just those that are based on hash trees. One example is NTRUSIGN, which is discussed later in this paper.

One important point to note is that unlike current signature schemes, this scheme is not stateless. The signer needs to keep track of more than just a single long-term private key in order to sign messages. If the signer is using hash trees, the signer can save a lot of memory by using a pseudorandom number generator to generate one-time private keys from a seed and a counter rather than saving all of the one-time private keys in memory. The one-time private keys are large and are only used twice: once for the purpose of generating the hash tree, and again when the one-time private keys are needed to sign messages, so this makes fairly good sense. The hashes in the tree, however, are used more often, and they should therefore be saved in memory. If these management techniques are used, then the footprint of a signing module does not suffer terribly from the short lifetime of the underlying signature scheme, but the dynamic nature of the stored information does imply that read-only or write-once memory cannot be used to store it.

5. LATTICE BASED CRYPTOGRAPHY AND NTRU

Unlike Lamport signatures, most public key cryptographic schemes derive their security from the difficulty of specific mathematical problems. Historically, factorization and the discrete logarithm problem have been by far the most productive in this respect, but as previously noted, these problems will not be difficult if full scale quantum computers are ever built. Therefore, cryptographers have been led to investigate other mathematical problems to see if they can be equally productive. Among these are lattice problems.

An n -dimensional lattice is the set of vectors that can be expressed as the sum of integer multiples of a specific set of n vectors, collectively called the basis of the lattice—note that there are an infinite number of different bases that will all generate the same lattice. Two NP-hard problems related

to lattices are the shortest vector problem (SVP) [1] and the closest vector problem (CVP) [52]. Given an arbitrary basis for a lattice, SVP and CVP ask the solver to find the shortest vector in that lattice or to find the closest lattice vector to an arbitrary non-lattice vector. In both the quantum and classical computation models, these problems are believed to be hard for high dimensional lattices, containing a large number of vectors close in length to the shortest lattice vector.

Of the various lattice based cryptographic schemes that have been developed, the NTRU family of cryptographic algorithms [25, 26, 27] appears to be the most practical. It has seen some degree of commercial deployment and effort has been underway to produce a standards document in the IEEE P1363 working group. NTRU-based schemes use a specific class of lattices that have an extra symmetry. While in the most general case, lattice bases are represented by an $n \times n$ matrix, NTRU bases, due to their symmetry, can be represented by an $n/2$ dimensional polynomial whose coefficients are chosen from a field of order approximately n . This allows NTRU keys to be a few kilobits long rather than a few megabits. While providing a major performance advantage, the added symmetry does make the assumptions required for NTRU-based schemes to be secure somewhat less natural than they would otherwise be, and many in the theory community tend to prefer schemes whose security follows more directly from the assumption that lattice problems are hard. Such schemes include schemes by Ajtai and Dwork [2], Micciancio [39], and Regev [45].

In all NTRU-based schemes, the private key is a polynomial representing a lattice basis consisting of short vectors, while the public key is a polynomial representing a lattice basis consisting of longer vectors. A desirable feature of NTRU and other lattice based schemes is performance. At equivalent security strengths, schemes like NTRU tend to be 10 to 100 times faster than conventional public key cryptography, with cryptographic operations taking about 100 microseconds on contemporary computing platforms.

A number of minor attacks have been discovered against NTRUENCRYPT throughout its 10+ year history, but it has for the most part remained unchanged. Improvements in lattice reduction techniques have resulted in a need to increase key sizes somewhat, but they have remained fairly stable since 2001. NTRUENCRYPT has also been found to be vulnerable to chosen ciphertext attacks based on decryption failures [18, 21, 31, 38], but a padding scheme [30], which has provable security against these attacks, has been developed. In addition to security concerns, the recommended parameter sets for NTRUENCRYPT have been changed for performance reasons. In one case, this was done over-aggressively and this resulted in a security vulnerability that reduced the security of one of the parameter sets from 80 bits to around 60 [29].

A comparatively greater number of problems have been found in NTRU-based signature schemes. The first NTRU-based signature scheme, NSS [28], was broken in 2001 by Gentry, Jonsson, Stern, and Szydlo a year after its publication [22]. A new scheme called NTRUSIGN [25] was introduced in 2002, based on the Goldreich-Goldwasser-Halevi signature scheme [23]. In this scheme, the signer maps the message digest to a vector, and proves knowledge of the private key by finding the nearest lattice point to that vector. Since the set of vectors to which a given lattice point is the

nearest is non-spherical, it was known that a large number of messages signed with the same key would leak information about the private key. Because of this, the original signature scheme included an option, called perturbation, that would allow the signer to systematically choose a lattice point which was not necessarily the closest lattice point, but which was still closer than any point that could be found without knowledge of the private key. In 2006, it was shown by Nguyen that the unperturbed NTRUSIGN could be broken given only 400 signed messages [42]. The developers of NTRUSIGN estimate that with perturbation, it is safe to use the same NTRUSIGN key to sign at least one billion messages [54], but recommend rolling over to a new signing key after 10 million signatures [43].

6. MCELIECE

An additional hard problem that has been used to construct public key schemes is the syndrome decoding problem, which asks the solver to correct errors that have been introduced to an arbitrary, redundant linear transformation of a binary vector. There are, of course, easy instances of this problem, namely error correction codes, but in the general case, this problem is known to be NP-hard. One of the oldest of all public key cryptosystems, McEliece encryption [34], works by disguising an easy instance of the decoding problem as a hard instance. The security of McEliece therefore relies upon the presumed fact that it is difficult to distinguish between the disguised easy code and an arbitrary hard code.

The easy instance of the decoding problem used by McEliece is a family of error correction codes known as Goppa Codes. An (n, k) Goppa code takes a k -bit message to an n -bit code word in such a way that the original message can be reconstructed from any string that differs from the code word at fewer than $t = (n - k)/\log_2(n)$ bits. There are approximately n^t/t such codes. To disguise the code, it is written as an $n \times k$ matrix, then left-multiplied by an n -bit permutation matrix, and right multiplied by an arbitrary invertible binary matrix. The resulting $n \times k$ binary matrix is the public key, while the three matrices used to generate it remain private.

To encrypt a k -bit message, the encryptor treats the message as a binary vector, left-multiplies the public key, and randomly changes t of the resulting n bits. The private key holder can then decode the message stepwise. First the private key holder undoes the private permutation—this does not change the number of errors. The errors can now be corrected using the private Goppa code, allowing the private key holder to reconstruct the k -bit linear transformation of the original message. Since the private linear transformation used to construct the public key is invertible, the private key holder can now reconstruct the message.

McEliece has remained remarkably resistant to attack during its 30 year history, and it is very fast, requiring only a few microseconds for encryption and 100 microseconds for decryption on contemporary platforms. The primary drawback is that in order for the scheme to be secure, n and k need to be on the order of 1000, making the total size of the public key about a million bits.

It was recently demonstrated by Courtois, Finiasz, and Sendrier that there was a corresponding signature scheme [11], but this scheme is less desirable than the encryption scheme. To sign a message, the signer decrypts a string derived by

padding the message digest. However, since most strings will not decrypt, the signer will typically have to try thousands of different paddings before finding a string that will decrypt. As a result, signing times are on the order of 10 to 30 seconds. It is, however, possible to make the signatures reasonably short.

7. OTHER AREAS OF RESEARCH

In addition to hash based signatures and lattice based and code based cryptography, a number of additional approaches have been used as an alternative basis for public key cryptography [7]. While most of the resulting schemes are currently poorly understood or have been broken, it is still possible that breakthroughs in these areas could one day lead to practical, secure, and quantum-resistant public key schemes.

One of the first NP-complete problems used in public key cryptography was the knapsack problem. Merkle and Hellman first proposed a knapsack based cryptosystem in 1978 [37], but this was soon shown to be vulnerable to approximate lattice reduction attacks [49]. Many similar schemes were subsequently broken, with the last, Chor-Rivest [9], being broken in 1995 [48].

More complex algebraic problems have also been proposed as successors to the factoring and discrete logarithm problems. These include the conjugacy search problem and related problems in braid groups, and the problem of solving multivariate systems of polynomials in finite fields. Both have been active areas of research in recent years in the mathematical and cryptographic communities. The latter problem was the basis for the SFLASH signature scheme [12], which was selected as a standard by the New European Schemes for Signatures, Integrity and Encryption (NESSIE) consortium in 2003 but was subsequently broken in 2007 [15]. It remains unclear when these or other algebraic problems will be well enough understood to produce practical public key cryptographic primitives with reliable security estimates.

8. CONSIDERATIONS FOR PROTOCOL DESIGNERS

In order to enable a comparison of the costs associated with various algorithms, Table 1 presents information about key sizes, message sizes, and the amount of time required to perform certain operations for several public key cryptographic algorithms. The table includes the algorithms that are described in this paper that are believed to be quantum resistant (Lamport signatures, McEliece encryption and signatures, NTRUENCRYPT, and NTRUSIGN) as well as some of the public key cryptographic algorithms commonly in use today that are vulnerable to Shor's algorithm (RSA, DSA, Diffie-Hellman, and ECC). The numbers presented in the table are rough estimates, not benchmark results, but should be sufficiently accurate to enable comparison of the strengths and weaknesses of the different algorithms.

Compared to public key cryptographic algorithms commonly in use today, the algorithms presented in this paper differ in two ways that may be significant to protocol designers: key size and limited lifetime. Of the algorithms listed in Table 1, limited key lifetime is only an issue for Lamport signatures and NTRUSIGN. In the case of these two algorithms, the limited lifetimes should not pose significant

Table 1: A Comparison of Public Key Cryptographic Algorithms at the 80 Bit Security Level

	Estimated Time (PC)			Limited Lifetime?	Public Key Size (kbits)	Private Key Size (kbits)	Message Size (kbits)
	Setup (ms)	Public Key Operation (ms)	Private Key Operation (ms)				
Lamport Signature	1	1	1	1 signature	~10	~10	~10
Lamport w/Merkle	1	1	1	2^{40} signatures	0.08	~250	~50
McEliece Encryption	0.1	0.01	0.1	no	500	1000	1
McEliece Signature	0.1	0.01	20,000	no	4000	4000	0.16
NTRUENCRYPT	0.1	0.1	0.1	no	2	2	2
NTRUSIGN	0.1	0.1	0.1	2^{30} signatures	2	2	4
RSA	2000	0.1	5	no	1	1	1
DSA	2	2	2	no	2	0.16	0.32
Diffie-Hellman	2	2	2	no	2	0.16	1
ECC	2	2	2	no	0.32	0.16	0.32

problems, but more consideration will need to be used in deploying these algorithms in order to ensure that keys are not used too many times.

When Lamport signatures are used in conjunction with Merkle hash trees as described in Section 4, the number of signatures that may be created from a given long-term public key is strictly limited, but that limit may be set to any value that the creator of the key chooses. If public keys have expiration dates, as they do today, then the maximum can always be set to a value that will ensure that the long-term public key will expire before all of the one-time keys have been used. Even a high volume server creating a few thousand signatures a second would take several years to create 2^{40} signatures. For most key holders, the maximum number of signatures per long-term public key could be set at a much smaller value, which would allow for smaller private keys and signatures.

The situation with NTRUSIGN is less clear since there is no fixed limit on the number of times that a key may be used. While the developers of NTRUSIGN recommend rolling over keys after 10 million signatures in order to be conservative, they believe that a key may be safely used to sign at least a billion messages [43]. For most key holders, even a limit of 10 million signatures would not be an issue. For some high volume servers, however, obtaining a new key pair and certificate after every 10 million signatures would be unreasonable, whereas a new certificate could be obtained after every billion signatures if the process were automated and relatively fast. If NTRUSIGN is to be used in the future, and further research indicates a need to impose key lifetimes that are closer to 10 million signatures than to 1 billion signatures, then high volume servers may need to employ one of the techniques described in Section 4 in order to reduce the frequency with which new certificates need to be obtained.

Table 1 shows the estimated key sizes that would be required to achieve 80-bits of security (i.e., a security level comparable to that provided by an 80-bit symmetric key). While 80-bits of security may be considered adequate at the moment, it is recommended that within the next few years all such keys be replaced with keys that provide 112 to 128

bits of security [3]. For the McEliece algorithms, this would imply 1 megabit public encryption keys and 8 megabit public signature keys. With key sizes this large, the ways in which public keys are distributed must be carefully considered.

With many protocols in use today, it is common to include a copy of the sender’s certificate(s) in the message. For example, the server’s encryption certificate is usually sent to the client during the key establishment phase of the Transport Layer Security (TLS) protocol. Also, email clients typically include copies of the sender’s signature and encryption certificates in all digitally signed messages. Since most public key certificates that have been issued are less than 2 kilobytes, this is a reasonable practice at the moment, as the amount of bandwidth wasted by sending a copy of a certificate to a recipient that has previously received a copy is minimal. However, if the need to switch to quantum resistant algorithms were to lead to the use of public key cryptographic algorithms with key lengths comparable to those required by the McEliece signature and encryption schemes, this practice would need to be avoided and other means would need to be used to ensure that relying parties could obtain copies of the public keys that they need.

The most straightforward solution to this problem would be to avoid sending certificates in protocol messages, except in cases in which the recipient has requested a copy of the certificate. Instead, the protocol message could include a pointer to the certificate, which could be used by the recipient to obtain a copy of the certificate if it does not already have a copy in its local cache. For privacy reasons, many organizations prefer not to place end user certificates in publicly accessible directories. However, if the directories that hold certificates are not searchable and the URLs that point to the certificates are not easily guessable, this should provide an adequate amount of privacy protection.

An alternative solution would be to not include a copy of the public key in the certificate, but instead include a pointer to the public key along with a hash of the key. In this case, since the directory would only include the public key, there would be fewer privacy concerns with respect to the data in the directory. This would also allow the relying party to validate the certificate before downloading the

public key, in which case the relying party could avoid the cost of downloading a very large public key if the certificate could not be validated, and thus the public key could not be used.

With very large public signature keys, the organization of public key infrastructures (PKI) would also need to be carefully considered. Today, even a very simple PKI may consist of a hierarchy of certification authorities (CA), with a root CA that issues certificates to subordinate CAs that in turn issue end user certificates. While the relying party would have already obtained the public key of the root CA through some secure, out-of-band means, the public key of one of the subordinate CAs would need to be downloaded in order to verify the signature on an end user certificate. If responses from Online Certificate Status Protocol (OCSP) [41] responders were needed to verify that neither the intermediate nor the end user certificate had been revoked, this could require the relying party to download two more public keys in order to verify the responses from the two OCSP responders. So, validating an end user certificate in a simple two-level hierarchy could require the relying party to download three public keys in addition to the end user's public key. In some PKIs today, certification paths involving four or more intermediate certificates are not uncommon. While this is reasonable with the public key algorithms that are in use today, which use public keys that are smaller than one kilobyte, such PKI architectures will need to be reconsidered if there is a need in the future to move to public key algorithms that require the use of very large public keys.

9. CONCLUSION

While factoring and discrete logarithm based cryptography continue to dominate the market, there are viable alternatives for both public key encryption and signatures that are not vulnerable to Shor's Algorithm. While this is no guarantee that they will remain impervious to classical or quantum attack, it is at least a strong indication. When compared to current schemes, these schemes often have similar or better computational performance, but usually require more bandwidth or memory. While this should not be a major problem for PCs, it may pose problems for more constrained devices. Some protocols may also have problems with increased packet sizes.

It does not appear inevitable that quantum computing will end cryptographic security as we know it. Quantum computing is, however, a major threat that we probably will need to deal with in the next few decades, and it would be unwise to be caught off guard when that happens. Protocol designers should be aware that changes in the underlying cryptography may and almost certainly will be necessary in the future, either due to quantum computing or other unforeseen advances in cryptanalysis, and they should be at least passably familiar with the algorithms that are most likely to replace current ones. Cryptanalysts will also need to scrutinize these algorithms before they are urgently needed. While some work has been done already, more work is needed to convince the cryptographic community that these algorithms will be as safe, in the future, as factoring and discrete logarithm based cryptography are today.

10. REFERENCES

- [1] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions (extended

- abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pages 10–19, 1998.
- [2] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293, 1997.
- [3] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid. Recommendation for key management – part 1: General. NIST special publication 800-57, National Institute of Standards and Technology, Mar. 2007.
- [4] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computation. *Special Issue on Quantum Computation of the Siam Journal of Computing*, Oct. 1997.
- [5] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. *SIAM J. of Computing*, 32(3):586–615, 2003.
- [6] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In *Advances in Cryptology – ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 514–532, 2001.
- [7] J. Buchmann, C. Coronado, M. Döring, D. Engelbert, C. Ludwig, R. Overbeck, A. Schmidt, U. Vollmer, and R.-P. Weinmann. Post-quantum signatures. Cryptology ePrint Archive, Report 2004/297, 2004.
- [8] J. L. Carter and M. N. Wegman. Universal classes of hash functions (extended abstract). In *STOC '77: Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 106–112, 1977.
- [9] B. Chor and R. L. Rivest. A knapsack type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5):901–909, Sept. 1988.
- [10] S. Cook. The importance of the P versus NP question. *Journal of the ACM*, 50(1):27–29, 2003.
- [11] N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology – ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 157–174, 2001.
- [12] N. T. Courtois, L. Goubin, and J. Patarin. SFLASH^{v3}, a fast asymmetric signature scheme. Cryptology ePrint Archive, Report 2003/211, 2003.
- [13] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proc Roy Soc Lond A*, 439:553–558, Oct. 1992.
- [14] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, Nov. 1976.
- [15] V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical cryptanalysis of SFLASH. In *Advances in Cryptology – CRYPTO 2007, 27th Annual International Cryptology Conference*, pages 1–12, 2007.
- [16] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6&7):467–488, 1982.
- [17] FIPS 186-2. *Digital Signature Standard (DSS)*.

- National Institute of Standards and Technology, Jan. 2000.
- [18] N. Gama and P. Q. Nguyen. New chosen-ciphertext attacks on NTRU. In *Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography*, pages 89–106, 2007.
- [19] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *Advances in Cryptology, Proceedings of CRYPTO '84*, pages 10–18, 1984.
- [20] L. C. C. García. On the security and the efficiency of the Merkle signature scheme. Cryptology ePrint Archive, Report 2005/192, 2005.
- [21] C. Gentry. Key recovery and message attacks on NTRU-composite. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques*, pages 182–194, 2001.
- [22] C. Gentry, J. Jonsson, J. Stern, and M. Szydło. Cryptanalysis of the NTRU signature scheme (NSS) from Eurocrypt 2001. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security*, pages 1–20, 2001.
- [23] O. Goldreich, S. Goldwasser, and S. Halevi. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference*, pages 112–131, 1997.
- [24] L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [25] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte. NTRUSign: Digital signatures using the NTRU lattice. In *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003*, pages 122–140, 2003.
- [26] J. Hoffstein, N. Howgrave-Graham, J. Pipher, J. H. Silverman, and W. Whyte. NTRUEncrypt and NTRUSign: efficient public key algorithms for a post-quantum world. In *PQCrypto 2006: International Workshop on Post-Quantum Cryptography*, pages 141–158, May 2006.
- [27] J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In *Algorithmic Number Theory (ANTS-III): Proceedings of the Third International Symposium on Algorithmic Number Theory*, pages 267–288, June 1998.
- [28] J. Hoffstein, J. Pipher, and J. H. Silverman. NSS: An NTRU lattice-based signature scheme. In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques*, pages 211–228, 2001.
- [29] N. Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against NTRU. In *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference*, pages 150–169, 2007.
- [30] N. Howgrave-Graham, J. H. Silverman, A. Singer, and W. Whyte. NAEP: Provable security in the presence of decryption failures.
- [31] É. Jaulmes and A. Joux. A chosen-ciphertext attack against NTRU. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference*, pages 20–35, 2000.
- [32] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177):203–209, 1987.
- [33] L. Lamport. Constructing digital signatures from a one-way function. Technical Report CSL-98, SRI International, Oct. 1979.
- [34] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report 42–44, Jet Propulsion Laboratory, California Institute of Technology, pages 114–116, 1978.
- [35] R. C. Merkle. *Security, Authentication, and Public Key Systems*. PhD thesis, Stanford University, June 1979.
- [36] R. C. Merkle. A certified digital signature. In *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference*, pages 218–238, 1989.
- [37] R. C. Merkle and M. E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, 24(5):525–530, Sept. 1978.
- [38] T. Meskanen and A. Renvall. A wrap error attack against NTRUEncrypt. *Discrete Applied Mathematics*, 154(2):382–391, Feb. 2006.
- [39] D. Micciancio. Improving lattice based cryptosystems using the Hermite normal form. In *Cryptography and Lattices Conference - CaLC 2001*, pages 126–145, Mar. 2001.
- [40] V. S. Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology - CRYPTO '85*, pages 417–426, 1986.
- [41] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. RFC 2560 (Proposed Standard), June 1999.
- [42] P. Q. Nguyen. A note on the security of NTRUSign. Cryptology ePrint Archive, Report 2006/387, 2006.
- [43] NTRU Announces Signature Algorithm, NTRUSign, viewed November 12, 2008, (http://www.ntru.com/cryptolab/intro_ntrusign.htm).
- [44] M. O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, Jan. 1979.
- [45] O. Regev. New lattice-based cryptographic constructions. *Journal of the ACM*, 51(6):899–942, Nov. 2004.
- [46] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, Feb. 1978.
- [47] S. Robinson. Emerging insights on limitations of quantum computing shape quest for fast algorithms. *SIAM News*, 36(1), January/February 2003.
- [48] C.-P. Schnorr and H. H. Hörner. Attacking the Chor-Rivest cryptosystem by improved lattice

- reduction. In *Advances in Cryptology – EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques*, pages 1–12, 1995.
- [49] A. Shamir. A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem. In *Advances in Cryptology: Proceedings of CRYPTO '82*, pages 279–288, 1982.
- [50] C. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.
- [51] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th Symposium on Foundations of Computer Science*, pages 124–134, 1994.
- [52] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, University of Amsterdam, Department of Mathematics, Netherlands, 1981.
- [53] G. S. Vernam. US patent #1,310,719: Secret signaling system, July 1919.
- [54] W. Whyte. NTRUSign and P1363.1, Apr. 2006. <http://grouper.ieee.org/groups/1363/WorkingGroup/presentations/P1363.1-2006-04.ppt>.
- [55] H. C. Williams. A modification of the RSA public-key encryption procedure. *IEEE Transactions on Information Theory*, IT-26(6):726–729, Nov. 1980.

Quantum Resistant Public Key Cryptography: A Survey

Ray A. Perlner

(ray.perlner@nist.gov)

David A. Cooper

(david.cooper@nist.gov)

What is a quantum computer

- Short answer
 - A classical computer processes classical information.
 - A quantum computer processes quantum information.
- What is the difference?
 - Classical information is measured in bits (a unit of entropy in the classical limit of physics)
 - Quantum information consists of qbits (a unit of entropy in real physics)
 - Either way, available entropy scales with the size of a system.
 - So it should be possible to build a quantum computer.

What can a quantum computer do? (faster than a classical computer)

- Simulate a quantum computer
 - The best known classical algorithm is exponentially more costly in the worst case.
 - This does NOT mean that a quantum computer can always provide exponential speedup.
- Stuff that matters for cryptography
 - Quadratic speedup over classical brute force search. (Grover)
 - Polynomial time algorithms for factoring and discrete logs, including elliptic curves. (Shor)
 - This completely breaks every public key algorithm you've probably ever heard of.

Why haven't these monstrosities been built?

- Error correction/fault tolerance is much harder for quantum information.
 - Currently, we're better off using a classical computer to run simulations.
 - Threshold theorems say that if we can build good enough components, the cost is only polynomial.
- Components are not cheap like transistors
 - Options include ultra-cold ultra-small solid state devices and charged ions or neutral atoms controlled by lasers.
 - Pure optical systems may be an important component, but are unlikely to be the whole solution.

Quantum Resistance

- Quantum resistant algorithms are algorithms we don't know how to break with a quantum or classical computer.
 - This is the same criterion we use for security in the classical model (pending $P \neq NP$ proof)
 - As with classically secure algorithms, related “hard problems” add a measure of confidence.
 - (Classical) algorithms meeting the above criteria do exist at present.

The Algorithms

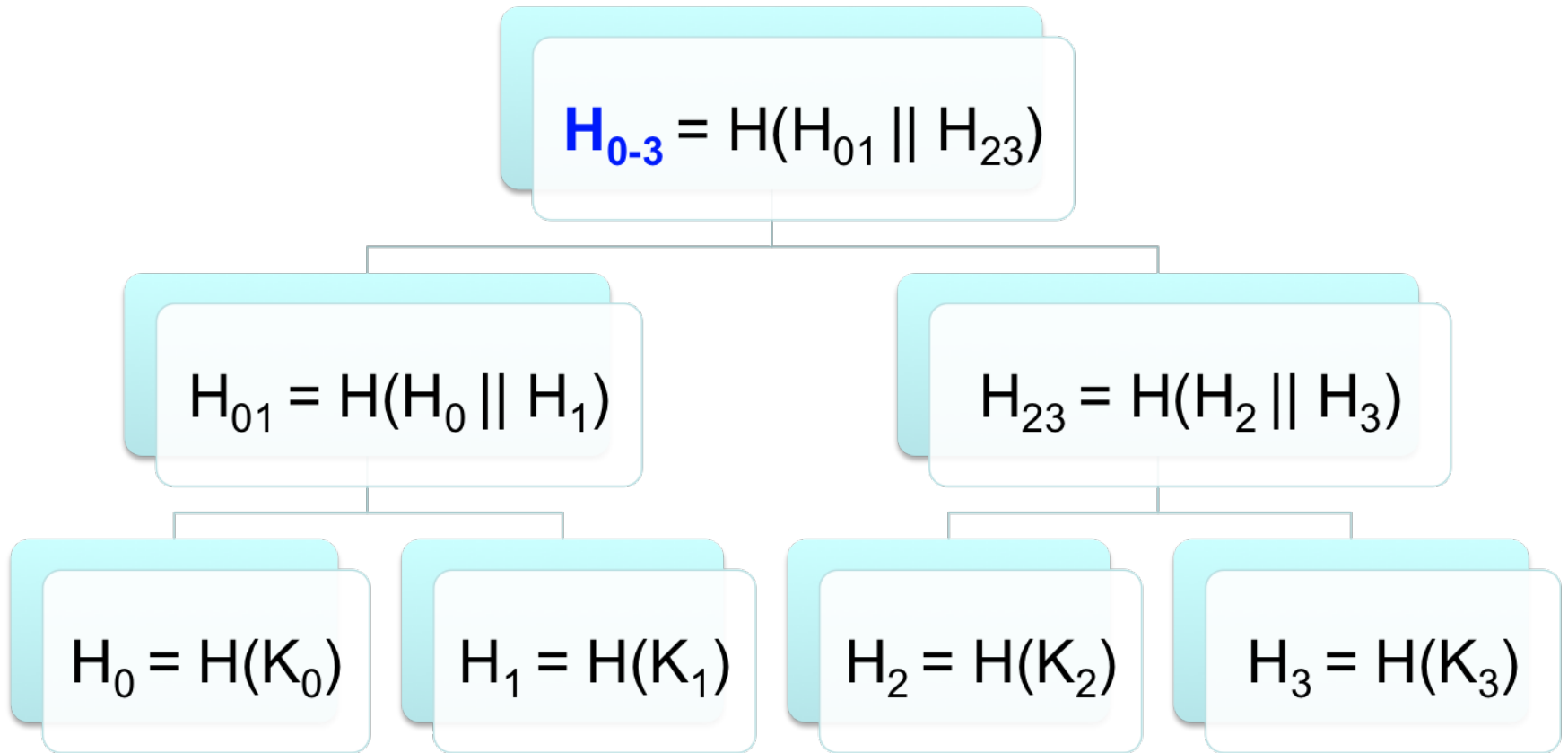
General Concerns

- Security Assumptions
- Public Key Length
- Signature Length/Ciphertext Expansion
 - E.g. RSA has ~1-2 kb (~10 - 20×)
- Public Key Lifetime
 - Mostly an issue for signatures
 - Can be dealt with using Merkle Trees and certificate chains
 - Memory (may need more than just the private key)
- Computational Cost

Lamport Signatures

- One time signatures
- Basic Scheme: Sign a single bit
 - Private key consists of two secrets S_0 and S_1
 - Public key is $H(S_0) \parallel H(S_1)$
 - Signature for 0 is S_0 , signature for 1 is S_1
- To sign an n -bit digest, just use n times as many secrets to sign the bits individually.
- Many optimizations are possible that trade increased computation for reduced key and/or signature size.

Merkle Trees



Lamport Signatures

- Security Assumption: preimage and second-preimage resistance of a one-way function
 - Only the message digest needs collision resistance.
- Public Key Length: $\sim n^2$ for an n -bit one-way function and a $2n$ -bit digest
 - ~ 10 kb for $n = 80$
 - ~ 20 kb for $n = 128$
- Signature Length: same
- Public Key Lifetime: 1 signature
- Computational Cost: ~ 1 ms (comparable to DSA)
 - Includes key generation

Lamport Signatures (with Merkle Trees and Chaining)

- Security Assumption: preimage and second-preimage resistance of a one-way function
 - Only the message digest needs collision resistance.
- Public Key Length: n for an n -bit one-way function and a $2n$ -bit digest
- Private Key Length: $\sim 250 - 500$ kb
- Signature Length: $\sim 50 - 100$ kb
- Public Key Lifetime: 10^{12} signatures
- Computational Cost: ~ 1 ms (comparable to DSA)
 - key generation: ~ 1 s

McEliece Encryption

- Start with an error correction code generator matrix, G
 - Rectangular matrix such that it's easy to reconstruct x from $Gx + e$.
 - x has dimension k
 - e has hamming weight t or less and dimension $n > k$
- Public key $K = PGS$
 - S is $k \times k$ and invertible
 - P is an $n \times n$ permutation
- To Encrypt m : compute $Km + e$

McEliece Encryption

- Security Assumption: indistinguishability of masked Goppa code and general linear code
 - Decoding problem for general linear codes is NP-complete
- Public Key Length: ~500kb
- Message Size: ~1kb
- Public Key Lifetime: potentially unlimited
- Computational Cost: ~100 μ s
 - Signatures exist, but very expensive for signer

NTRU

- Private key is a short basis for an N dimensional lattice
- Public key is a long basis for the same lattice.
- Save space by representing lattice basis as a polynomial rather than a matrix
 - This requires all lattice basis vectors to be cyclic permutations.
 - Many academic crypto schemes employ lattices but do not employ this technique, preferring security assumptions based on a less symmetric version of the lattice problems.
- Coefficients are generally reduced modulo $q \approx N \approx 256$

NTRU

- Security Assumption: unique closest vector problem
- Public Key Size: 2-4kb
- Ciphertext Size: 2-4kb
- Signature Size: 4-8kb
- Public Key Lifetime: ~1 billion signatures
 - Signature scheme has changed in response to a series of attacks.
- Computational Cost: ~100 μ s

Other

- Hidden Field Equations
- Braid Groups
- New schemes based on these crop up from time to time, but most have been broken.

Implications

- **Crypto Agility is a Minimum Requirement**
- **Long Signatures or Public Keys**
 - Transmitting certificates may become unwieldy (especially when revocation is considered)
 - Cache Certificates
 - Limit Cert Chain Depth
- **Limited Lifetime Signing Keys**
 - Mostly applicable to high load servers (e.g., OCSP responders)
 - Use a Merkle tree or subordinate public keys where applicable.

Conclusion

- All widely used public key crypto is threatened by quantum computing.
- We do have potentially viable options to consider.
- Protocol designers can think about how to deal with these algorithms now.

FileSpace

An Alternative to CardSpace that supports Multiple Token Authorisation and Portability Between Devices

David Chadwick
University of Kent
Computing Laboratory
Canterbury
+44 7796 44 7184

d.w.chadwick@kent.ac.uk

ABSTRACT

This paper describes a federated identity management system based on long lived encrypted credential files rather than virtual cards and short lived assertions. Users obtain their authorisation credential files from their identity providers and have them bound to their public key certificates, which can hold any pseudonym the user wishes. Users can then use these credentials multiple times without the identity providers being able to track their movements and without having to authenticate to the IdP each time. The credentials are worthless to an attacker if lost or stolen, therefore they do not need any special protection mechanisms. They can be copied freely between multiple devices, and users can use multiple credentials in a single transaction. Users only need to authenticate to their private key store in order for it to produce a signed token necessary for the service provider to authenticate the user and decrypt the authorisation credentials. The signed token is bound to the service provider and is short lived to prevent man in the middle attacks.

Categories and Subject Descriptors

C.2.4 Distributed Systems. K.6.5 Security and Protection

General Terms

Management, Design, Security, Human Factors

Keywords

Federated Identity Management, CardSpace, Authorisation, X.509 certificates, Information Cards

1. INTRODUCTION

Information Cards are the core component of Microsoft's CardSpace identity management and authorisation system. A good high level overview of CardSpace can be found in [1]. Information Cards are a representation of a person's online digital

identity. Information Cards have some excellent features in terms of both usability and security. From a usability perspective, the metaphor that Information Cards use for electronic credentials is the plastic card that everyone is familiar with. These are displayed on the user's desktop so that the user can select the card he wants to use in any transaction. Cards that are acceptable to the service provider (SP), and hence selectable, appear in full colour, whilst cards that are incompatible with the SP's requirements are greyed out and hence not selectable. Cards can be self generated or managed. Self generated cards contain information (attributes) asserted by the user himself, whereas managed cards contain attributes that are asserted by an Identity Provider (IdP) or Attribute Authority (AA) (i.e. a trusted third party, TTP). The fact that the attribute assertions (or claims) of the managed cards do not actually reside on the user's desktop, but are pulled from the IdP on demand, is largely hidden from the user. The only telling feature is that the user has to enter his login credentials with the IdP in order for the claim to be picked up and sent to the SP. This could be seen as a usability disadvantage or inconvenience to users, since the user is distracted from his/her primary task, which is accessing a service provider, into providing authentication credentials to an alternative party, the identity provider. But this is really not that much different to users entering their PINs today in order to activate their plastic cards.

From a security perspective, CardSpace also contains some excellent features. Firstly it is resistant to phishing attacks, since an SP cannot redirect users to a malicious entity masquerading as their identity provider, since the users store this information securely on their PCs in the meta-information of their identity cards. Phishing can only succeed if the attacker can manage to subvert the user's PC without his knowledge, in order to plant subversive cards in the user's identity selector. Secondly there is nothing of value on the user's desktop that can be stolen by an adversary, since the credentials or claims are only generated on demand by the IdP when required. The credentials are short lived, cryptographically protected, designed to be transferred as quickly as possible from the IdP to the SP via the user's desktop, and can be created to be read by the SP only. So there is little opportunity for an attacker to steal them.

However, CardSpace is missing some technical features that critically affect its ubiquity and utility. It may have user acceptance problems as well. These might explain its slow uptake to date. They are:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IdTrust '09, April 14-16, 2009, Gaithersburg, MD
Copyright 2009 ACM 978-1-60558-474-4...\$5.00

- i) the lack of mobility/portability. In initial versions of CardSpace, a user's cards were tied to his Microsoft PC and could not be moved between devices and operating systems. Whilst this has been fixed by defining a card transfer format [14, 15], the format is an XML encrypted file which may cause difficulties for constrained devices that currently do not process XML. Data structures in XML are typically an order of magnitude greater than equivalent binary structures in ASN.1, and XML cryptography is up to an order of magnitude slower than ASN.1 based cryptography [16]. Consequently it is still not clear how portable info cards will be in practice.
- ii) the inability to use multiple cards in a single transaction. Whilst many transactions only require a single card to ensure success, a large proportion require multiple cards e.g. showing a student card and visa card to buy a book with a student discount, or showing a General Practitioner certificate and employee certificate in order to access a patient's medical record;
- iii) Users have to learn a new paradigm for interacting with service providers – the whole CardSpace philosophy.

Consequently the CardSpace system needs to be enhanced to allow easy mobility and portability between devices, as well as to allow the use of multiple cards in a single transaction. Whilst adding these new features, we should not lose the existing good features of usability and security that CardSpace exhibits, but if we can improve upon them and upon the user experience of Identity Management (IdM) at the same time, for example, by not introducing new paradigms to users, then so much the better. As Landau and Mulligan state [10] "usability is the key to success".

2. AN EXISTING PARADIGM FOR CREDENTIALS – CONFIDENTIAL FILES

All computer users today are familiar with using computer files. They are fundamental to the use of any PC. One of the most ubiquitous sets of user abstractions that have been developed are those for using the file system. All users today are familiar with drag and drop, copy, delete, rename file etc. Thus they are already a more commonplace paradigm when using computers than are the virtual plastic cards of CardSpace. Users also understand that different files contain different contents, and that some files may contain confidential information whilst others may not. Thus users do not need to develop any new skills in order to manipulate their files. They do it today, all of the time.

One of the critical objectives we should have when developing an IdM system, is to make IdM and Authorisation very easy to use; as easy say, as manipulating user files is today. As Dhamua and Dusseault state [11] "To succeed in the marketplace, identity management system must.....most important, simplify the process of authentication, identification and assertion of credentials". This implies that authorisation tokens might be easier for users to handle if they are simply regarded as files on their desktop rather than card icons. Users are used to the fact that some of their files may contain public information and some may contain confidential information, so they do not need to be educated in handling different types of files differently. By turning a user's credentials into simple confidential files we make it easy for users

to drag and drop them between devices e.g. copy from a laptop to a mobile phone or memory stick etc. We also make it easy for users to pick and send multiple credentials to web service providers. Web browsers simply need to allow the user to navigate around their filestore and click on several credential files in order to automatically transfer (i.e. copy) them to the service provider. They have this functionality today. Other benefits of using the file paradigm for credentials, is that we can then use existing software for synchronisation of credential files between different devices, and existing protocols for transferring credential files between systems. If the credentials are self protected, in terms of integrity protection (via a digital signature) and confidentiality/privacy protection (via encryption), then no special protocols are needed for transferring them between systems, and we also protect the user from copying them by mistake to a malicious third party, or altering them either intentionally or by mistake.

Of course if we move to a file paradigm, there are still a significant number of security mechanisms that we will need to develop in order to protect the user from phishing attacks, from the theft of his credentials and from the loss of his privacy.

3. THE FILESPACE CONCEPTUAL MODEL

Credentials can be short lived or long lived. If they are long lived we need a revocation mechanism, if they are short lived we do not, the assumption being that during their short life time there is little chance of them being stolen and used to ill effect. Today we use long lived credentials for X.509 public key certificates (PKCs) and physical plastic cards, and short lived credentials for X.509 grid proxy certificates [2], SAML attribute assertions [3] (for example as used by Shibboleth [4]) and CardSpace credentials. Both short lived and long lived credentials have their advantages and disadvantages [5]. If we are to use the file paradigm for credentials, we need to make them long lived so that users can manipulate them, copy them, and use them multiple times, as they do today with their plastic cards. In this case we need to make it the responsibility of the SP to check if the presented credentials have been revoked or not. But this is their normal responsibility, since it is part of their risk management procedures. We can easily help the SP in this function, by including policy information in each issued credential which informs all relying parties (SPs) where they can find the revocation information. This is the standard X.509 model, and X.509 public key certificates use both the CRL Distribution Points extension [6] to point to revocation lists and the Authority Information Access extension [7] to point to OCSP responders [8]. So this is a well known and well used technology that we can also use for authorisation credentials.

Next we need to stop authorisation credentials from being stolen or lost by their owner, or from being sent to a malicious site by mistake by their owner. Clearly we cannot physically do any of this, so the next best thing, if they are stolen or lost or sent to a malicious site, is to ensure they are worthless to the thief. We can easily make authorisation credentials worthless to a thief by cleanly separating authentication from authorisation, by encrypting the authorisation credentials and then requiring their rightful owner to authenticate and prove possession by providing the decryption key. Then if anyone steals a user's authorisation

credentials they are worthless to the attacker unless the attacker can either i) authenticate as the user in order to use them (i.e. masquerade), or ii) trick the user into providing decryption of the authorisation credential's contents in order to invade the user's privacy. Modern day cryptography should protect against the second threat. We protect against the first threat by cleanly separating authentication credentials from authorisation credentials, and by requiring anyone who presents an authorisation credential to an SP, to also prove to the SP that they are the rightful owner of the authorisation credentials before the SP will or can use them. This is akin to protecting today's plastic credit cards with a PIN, and mandating that the PIN be presented before the credit card can be used. Then if an attacker steals a user's authorisation credentials they become worthless to him, unless he can prove to the SP that he is the user i.e. can authenticate as the user to the SP. This should be very difficult for an attacker to do if we use a strong authentication mechanism such as a digital signature (rather than the relatively weak 4 digit PIN mechanism of today's credit cards) and we keep the private signing key in a piece of hardware to make it difficult to steal. Unless an attacker can steal the strong authentication mechanism (i.e. my hardware) before or after he steals my authorisation credentials, that is, sometime during the validity period of my authorisation credentials, and *without me knowing that he has stolen my authentication mechanism*, then we do not care if he simply steals my authorisation credentials. They are useless to him. He cannot decrypt them and he cannot masquerade as me. The only thing a user needs to protect and look after herself is her authentication mechanism

3.1 Contents of Authorisation Credentials

All authorisation credentials (also known as attribute assertions, claims and attribute certificates) regardless of their syntax (ASN.1, XML or proprietary format) conceptually comprise the following fields:

- the unique identifier of the credential holder *
- the unique identifier of the credential issuer
- the serial number of the credential
- the authorisation attribute(s) e.g. organisational role, group membership, degree classification, status attribute, credit card number, etc. *
- the validity time of the credential
- policy information of the issuer to control how the credential should be used e.g. one time use, how to obtain revocation information for long lived credentials, which services it should be used for, limitations of liability etc.
- information about the cryptographic algorithm(s) used, to tell the receiver how to validate the credential
- the signature of the credential issuer.

In order to privacy protect these authorisation credentials we need to encrypt all the Personally Identifying Information (PII) of the holder. This comprises the unique identifier of the holder and the authorisation attribute(s) i.e. the information marked with an * above. Now if anyone steals one of these encrypted authorisation credentials it is useless to them since:

- i. they cannot read its contents because they don't have the decryption key, and
- ii. they cannot authenticate as the rightful holder because they don't have the authentication mechanism.

If we encrypt the fields of the authorisation credential using the public key of the holder, then the holder is the only person who can decrypt its contents and read it, by using their private key. If we introduce a level of indirection, by encrypting the credential contents with a randomly generated symmetric key, then encrypt the symmetric key with the public key of the holder and store the encrypted key in the authorisation credential, then this has the added advantage of speed (since symmetric encryption/decryption is much faster than asymmetric encryption) and we can subsequently use the symmetric key to give relying parties such as SPs read access to the authorisation credential (as described later).

3.2 Obtaining Authorisation Credentials

In order to issue such an authorisation credential, the issuer needs to know four things:

- a) who is the real person that is asking for this credential to be issued
- b) are they entitled to be issued with this credential
- c) what unique identifier (pseudonym) do they wish to be inserted into their authorisation credential
- d) which public key are they currently using.

Item a) is a registration issue and is typically solved at registration time, when a user first enrolls with an identity provider/attribute authority (IdP/AA¹). After registration the user will typically be given some issuer specific authentication credential with which to re-authenticate to their systems. Item b) is an internal issue and is solved by the issuer consulting its internal databases to see which privileges have been assigned to the user. The user may use their assigned authentication credential to prove to the issuer who they really are and the issuer will then consult its databases to see which privileges this user possesses. For example, when a student first registers at a university, they bring their passport, school qualifications, language certificates etc. with them to prove who they are and that they are qualified to enroll on a degree program. In exchange they might be given a unique login id by the university, which they can subsequently use in all their electronic interactions with the institution. It is this login id and its associated authentication credential (such as a password) that allows the user to authenticate to the university's computer systems and assure the university who is the real person that its computers are talking to. All the user's degree qualifications and transcripts will be linked to this login id. Note that this login id is of no value outside of the university context. Its raison d'être is to uniquely identify the user in the computer systems of the university. No two users will have the same login identifier (unless the system is broken!). The user may use this login id to authenticate to the university to prove who they really are and the university will then consult its databases to see which degree marks and awards this user possesses. In tune with other identity management systems such as CardSpace and Shibboleth, we don't dictate what registration and authentication mechanisms each credential issuer will use, but clearly some will use stronger mechanisms than others. NIST has issued guidelines for the registration and authentication procedures that can be used, and

¹ Note that we do not separate the functions of IdP and AA and assume the same entity performs both functions, since this is typically the case today.

this document defines four different levels of authentication assurance [12].

Items c) and d) can be obtained from a public key certificate of the credential holder, providing that the public key certificate contains a unique identifier in its subject field, and providing that the holder proves possession of the corresponding private key (see section 3.3 below).

The unique pseudonym and the requested attributes are then encrypted using a freshly generated symmetric key, and the symmetric key is encrypted using the public key, before they are all inserted into the authorisation credential. The authorisation credential is given a validity time that starts at or shortly after the time of issuing, and finishes when the accompanying public key certificate expires or earlier, at the option of the issuer. Finally the authorisation credential is signed by the issuer and returned to the user.

3.3 Obtaining public key certificates

In order to obtain an authorisation credential, after authenticating to the IdP/AA with their IdP specific mechanism², the user presents their public key certificate (PKC) and proof of possession of the private key, and asks for an authorisation credential to be issued containing a subset of their privilege attributes. The encrypted pseudonym that is inserted into the authorisation credential is not the unique identifier that the IdP/AA knows the user by, but is the unique pseudonym from the public key certificate presented by the user.

The only technical requirement we have from the PKC is that the pseudonym is unique within the scope of the authorisation system (IdP or federation) or systems that the user wishes to use it in, in order to prevent the user from masquerading as or being mistaken for another user of the same system. Whether the identifier is similar to the real name of the user, or is a completely fictitious pseudonym such as Father Christmas, or a random number, is not important from a technical perspective. The only technical requirement is that it is unique. Unique identifiers could be DNS names, OpenID identifiers, hashes of public keys etc. There are numerous options to choose from. It is trivial to create globally unique identifiers using user generated pseudonyms, for example, by simply pre-pending a base64 hash of the public key to the user's pseudonym, e.g. create an X.509 distinguished name of KID=12345678...9, CN=Father Christmas. This gives the user complete anonymity.

Additional considerations that the authorisation credential issuer might have for the pseudonym, are that the name is not offensive or illegal (e.g. a trade mark that does not belong to the real person whose credential this will be), and is not likely to confuse a relying party because it could be mistaken for a different client of the issuer. Each credential issuer will typically have its own policy for what comprises suitable unique names. Federation guidelines can be established for this. The user will need to ensure

that his pseudonym conforms to the policies of the various credential issuers or federations that he wishes to use. We propose two alternatives for public key certificate generation:

1. Self issued certificates in which the users create their own unique pseudonyms
2. Trusted Certification Authority issued certificates in which the CA has a policy for how names are assigned and validated before insertion into their certificates. CAs may issue certificates with genuine pseudonyms or may issue them with names that allow relying parties to identify their real world owners.

The important thing to note is that the pseudonym can be irrelevant from an authorisation and identification perspective. The only real requirement is that it can act as a primary key into the databases of both the authorisation credential issuer and the relying parties. This is why it must be unique within a federation. But it does not need to be used for either authorisation purposes or for identifying the real life person; it is the certified attributes in the authorisation credentials that are used for authorisation, and it is the user's personal information stored with the credential issuers that are used to identify the real life person. The identifier may only be used for identification purposes if it has been issued by a trusted CA whose policy states that it bears some relationship to a real life person or legal entity. Otherwise the SPs should only use the attributes for authorisation and the pseudonym for linking the user's transactions in order to build up their own usage profiles. In most developed countries SPs are prohibited by data privacy laws from sharing user profiles without the user's consent.

The IdP/AA must store the mapping between the pseudonym from the public key certificate and the login/authentication identifier that it keeps for the user. It may need this for legal reasons; for example, to identify the user should the user commit a fraud using the issued authorisation credential.

The user is free to generate as many pseudonyms and public key certificates as he wants to, with each public key certificate containing the same or different pseudonyms. If the same pseudonym is used in different public key certificates for example when the keys expire and are renewed, then the user will need to prove to the authorisation credential issuer that she is the owner of both private keys, otherwise the issuer will require unique pseudonyms with each public key (to stop masquerade). The IETF group has devised mechanisms for this as part of the certificate management procedures of CAs [7]. Revocation of the user's public key certificates is not an issue. If the user loses or has his private key stolen, he asks the authorisation credential issuers to revoke the authorisation credentials that are linked to the key pair. In this way, the thief may be able to masquerade as the pseudonymous user to an SP, but when he comes to assert the authorisation credentials, the SP will discover that they have been revoked, and the thief will not gain access to any resources. If the user has obtained his public key certificate from a CA, rather than using a self-issued one, then the user may ask the CA to revoke the public key certificate as well.

An IdP/AA may be willing to issue the same authorisation attributes to the same user in different authorisation credentials containing different pseudonyms, encrypted to different public keys. The willingness of the IdP/AA to keep a many to one

² Note that we are not concerned in this paper with how a user authenticates to his IdP/AA in order to be issued an authorisation credential. It could be with a username password, a single sign on system, a national ID card, etc. We regard this as a separate issue that is out of scope of this paper, as do the Shibboleth, SAML and CardSpace schemes.

mapping of public key identifiers to login identifiers is a policy issue of the IdP/AA.

The only other requirement we place on the generation of a public key certificate is that it should indicate the location of the corresponding private key. This is used to solve the discovery problem for SPs who subsequently want to obtain the decryption keys for the authorisation credentials they are about to accept (see section 3.3). If the user's private key is held in the SIM card of a mobile phone, then it would be the international telephone number of the SIM card. If the user's private key is held in a portable USB stick such as IBM's ZTIC [9], or a PKCS#12 file, or smartcard, then it would indicate that it is at the same location as the user making the service request.

Figure 1 below shows a user who has generated public key certificates with two different pseudonyms, which he has abbreviated to David Chadwick and Father Christmas, although the actual identifiers in the public key certificates will be longer than this and unique. The user has then authenticated to several IdP/AAs, in whatever way each IdP decrees, and has requested authorisation credentials from each. Under the pseudonym of Father Christmas the user has been given authorisation credentials from five different IdP/AAs, in which the identifiers and attributes have each been indirectly encrypted to the public key within the Father Christmas certificate (called Identity.card in Figure 1) as described above. Note that since these are simply computer files, the user can call the directories and files whatever nicknames he wants to. We would expect a unique three character file extension to be standardized for FileSpace files in due course.

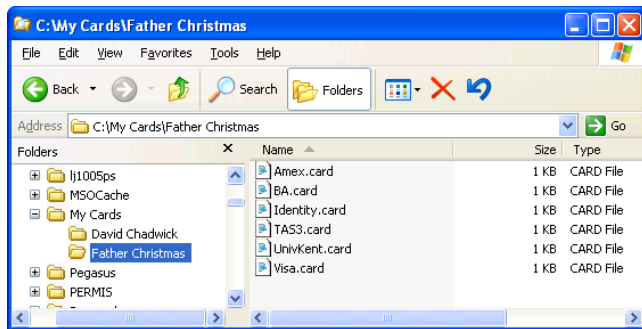


Figure 1. Filestore showing two pseudonyms and the authorisation files of Father Christmas

A critical requirement is that the user needs to keep the private keys of his various pseudonymous certificates safe and secure somewhere, preferably inside one or more hardware tokens such as a mobile phone SIM card or smart card or in a centrally managed key vault. National identity cards could be used, but then the user would not be able to use a pseudonym, and may have little or no privacy protection since the SPs will know the identity of the user from the public key certificate attached to the ID card. If someone can gain access to any of the user's private keys, they will be able to masquerade as the user in one or more of his pseudonyms, and utilise their privileges, until the user discovers this fact and revokes his authorisation credentials that are linked to the relevant key pair. But this is the case with all authentication credentials, and federated identity management systems including CardSpace, Shibboleth etc. Once an attacker can masquerade as a user, then as far as the system is concerned they are the user, so there is nothing new in this for the current

design. Consequently this paper does not address the issue of private key protection, since it has been well researched already and many different hardware tokens are commercially available. Needless to say this is a critical component of this design.

3.4 Using the Authorisation Credentials

Once a user has established a pseudonym, for example issued her own public key certificate, and obtained a set of authorisation credentials from her various Identity Providers, the system is very simple for end users to use. Copying credentials between devices is simply a matter of drag and drop. They don't need any special security mechanisms to protect them. As already explained, they are useless to an attacker without access to the corresponding private key, which the user must keep separately and securely, preferably in a hardware token so that the private key cannot be copied and cannot be lost or stolen without the user noticing it.

When contacting a service provider, the user simply selects the set of credentials he wants to present. This should always include the identity file (i.e. the public key certificate containing whichever pseudonym he wishes to use today) plus the set of authorisation files he needs that are linked to this pseudonym.

In order to accept these authorisation credentials, the SP needs to know three things:

- that the person presenting the credentials is the rightful owner
- that the credentials are still valid and have not been revoked by their issuers
- that it trusts the credential issuers to issue these types of authorisation credentials.

The SP can determine b) by reading the unencrypted contents of an authorisation credential and contacting its issuer as determined by the issuer's policy in the credential e.g. OCSP or CRL repository. The SP can only determine a) and c) by first getting the presenter to decrypt the contents of the credential in real time and then looking at its credential issuers policy to see which issuers it trusts to issue which credentials.

Decrypting the credentials in real time is achieved in the following way. The SP inspects the user's identity file (i.e. X.509 public key certificate) and determines the location of the user's private key from this. The SP then sends a signed request message to this location, including its public key certificate and the authorisation credentials it wants decrypting. The user's software validates the message's signature and SP's certificate, makes sure the message is not a replay, then asks the user if he wants to allow this SP to decrypt these confidential files. This is providing the user with the ability to confirm consent that she wishes this SP to use these credentials, and stops an active attacker such as a man in the middle from hijacking the credentials and masquerading as the user to a different SP. If the user answers yes, the software extracts the SP's public key, decrypts the symmetric keys in the authorisation credentials using the user's private key, and encrypts them to the public key of the SP. These encrypted keys, along with the serial numbers of their respective credentials, are then returned to the SP in a message signed by the user's private key. The signed message contains a nonce, timestamp and name of the SP, to prevent message replay, surreptitious forwarding, and to limit the time during which the message is valid. The SP can validate the signature, ensure the message is not a replay and

that it has not timed out, then decrypt the symmetric keys using its private key. It uses each symmetric key to decrypt the contents of the respective authorisation credential with the same serial number.

Formally the message exchange is as follows:

The message sent to the user's private key location contains:

$\{\{authzCred_i\}_{i=1 \text{ to } n}, nonce1, ts1, SP_{PKC}\}_{signed_{PSP}}$

The message returned from the user contains:

$\{\{sn_i, encKey_i\}_{i=1 \text{ to } n}, nonce1, nonce2, ts2, SP\}_{signed_{PUser}}$

Where: i is the number of authorisation credential sent from the SP to the user's private key location,

SP is the name of the service provider and SP_{PKC} is its public key certificate

nonce is a random number and ts is a short time in the future (say 2 seconds)

PSP is the private key of the SP and PUser is the private key of the user

sn_i is the serial number of an authorisation credential i

$encKey_i$ is the symmetric key used to encrypt the contents of authorisation credential i , encrypted to the public key of the recipient SP

The user can send several credentials to the SP, the SP can return several credentials to the user's private key location, and the private key location can return several decryption keys and serial numbers in one signed message. This message exchange allows the SP to know that the user with a particular pseudonym does possess these authorisation attributes and has on demand furnished the decryption keys to it. It allows the user to confirm that the only SP that will use these authorisation credentials is the one that it sent the decryption keys to. Using a private key device such as ZTIC [9] the user can be assured that no active MITM attack is taking place.

A malicious SP that wishes to act as a man in the middle in order to masquerade as the user with another SP (say the user's bank), will have difficulty in doing so. If it provided its own name to the private key location, then even though it has been given the symmetric key with which to decrypt the user's authorisation credentials, it does not have the user's private key and therefore cannot create a freshly minted reply from the user to the second SP. Thus it cannot forward the signed message to the second SP. If it provided the name of the second SP to the private key location, then the user would see that this is wrong and would not provide the decryption keys.

A group of malicious SPs that wish to correlate a user's activities between themselves, are technically able to do this with FileSpace but only if the user has used the same pseudonymous certificate with all members of the group. However such sharing of personal data is illegal under most data privacy laws e.g. [13] without the user's explicit consent.

The FileSpace system does require the user's private key location to have the software that is capable of performing the various decryption, encryption and signing operations that are described above. If the private key is accessible to a web browser e.g. as a PKCS#12 file or smart card etc. then the assumption is that this cryptography software will eventually be built into web browsers, in the same way that SSL/TLS and CardSpace cryptography

functions are built-in today. If the private key is held in a remote device such as a mobile phone then this software would need to be built into the phone's operating system.

4. User Experience with CardSpace and FileSpace

The following sections describe the procedures that users will experience as they use the CardSpace and FileSpace systems.

4.1 Obtaining a new identity/pseudonym

This user experience is only relevant to FileSpace. The user has a choice whether to use an existing CA for his new pseudonymous identity, or to generate his own PKC. The latter is simpler and gives the user complete pseudonymity.

If the user chooses an existing CA, then the current process of asking for an X.509 PKC may be used. For example, the user visits the CA's web site, requests a new certificate, completes the registration details including his new email address (say billg@gmail.com), and after clicking on the secret URL sent to his email address, the browser creates the X.509 PKC, sends it to the CA for signing, then stores the returned PKC in the user's certificate store.

If the user chooses to create his own self signed PKC, he will simply need to fill in his chosen pseudonym in the form that is provided by the key generation device. If the device is a web browser, a new option in the browser's menu could be added e.g. Tools>Options>Advanced>Create New Identity Certificate in Firefox 3 or Tools>Internet Options>Content>Certificates>Personal>Create New Identity Certificate in Internet Explorer 7. If it is a hardware device such as a mobile phone or IBM's ZTIC [9], then the device will need to display such a form. In all cases the user simply enters his pseudonym and the time period during which it should be valid, and the device then creates a new key pair and corresponding PKC. The user's new PKC file will need to be copied or exported from the device to the various filestores of the various computers and devices on which he wishes to use it. The format of the PKC could be a .cer file (as now) and the export/copy could be carried out by using a memory stick or by emailing the file as an attachment.

4.2 Obtaining New Cards and Files

The way that a new managed information card is obtained by a user is outside the scope of the CardSpace model and specification. IdPs are free to provide whatever web based interface they wish for this. One typical example will involve the user contacting his chosen IdP via his web browser, establishing an SSL/TLS session, logging in by sending his username and password over the encrypted link, and then being presented with a screen which asks the user which attributes he wishes to include in his new information card. The user will tick the set of attributes he wishes to be placed in his managed card, including the ability to include a new random permanent identifier, and a download screen will then appear allowing the user to navigate around his local filestore and choose the location where his new managed card is to be deposited. Once he has received the card (as a file with the prefix .crd on Windows systems), he will logout of his IdP, and enter his local identity selector program (CardSpace on Windows). This displays a new card icon. Clicking on this icon gives the user a choice between creating a new self issued card or importing a managed card. Selecting the latter allows the user to

navigate around his local filestore to choose the new card from the location where he has just downloaded it from his IdP.

The process in FileSpace will be very similar to that in CardSpace, once the user has his new identity PKC. The user contacts his chosen IdP via his web browser and opens an SSL/TLS session with it. The IdP asks the user to login by sending his username and password across the encrypted link (as above). If the user's private key is stored in his browser then mutual authentication will have been performed automatically by SSL/TLS, allowing the SP to link the authenticated user to the validated PKC. If the user has several pseudonyms (i.e. key pairs) then the browser will have asked the user (as now) which identity certificate he wished to use when establishing the SSL link. If the user's private key is not stored in his browser, the IdP will need to display an upload window, allowing the user to select and send his PKC file to the IdP. From this, the IdP will determine the location of the private key store and will send a challenge to it asking for Proof of Possession to be returned. This may entail the user entering his PIN into his private key device: mobile phone, smart card or ZTIC etc. Once the user and his pseudonym have been authenticated, the user is presented with a screen which asks him which attributes he wishes to include in his new authorisation file. The user will tick the set of attributes he wishes to be included and a download screen will then appear allowing the user to navigate around his local filestore and choose the location where his new authorisation file is to be deposited. Once he has received the authorisation file he will logout of his IdP, and login to the next one to obtain the next authorisation file. If the user has stored his IdP usernames and passwords in his browser's encrypted password file, and his key pair in the browser, then the entire authentication process will be automatic. The user will only need to select the attributes he wishes to be included in his new authorisation card.

4.3 Using existing Cards and Files

With CardSpace, the user contacts the SP requesting the service, whereupon his identity selector is activated and he is given the opportunity of choosing a single card to send to the SP. In CardSpace, the identity selector then asks the user for his username and password that go with this card, and these are relayed to the IdP. After a short delay, the user is returned to the SP's site and the service is provided.

With FileSpace, the user contacts the SP and requests a particular service, whereupon the user is presented with an upload screen, allowing him to select the authorisation files and PKC that he wishes to present to the SP. The SP processes these files, determines the location of the user's private key store, and then sends a message to this location. The device (web browser, mobile phone etc) pops up a message asking the user if he wants to allow this SP to read these authorisation credential files. The user checks the details and answers Yes, and in addition may be required to enter his PIN, whereupon the device returns a signed message to the SP and the SP provides the service to the user.

5. IMPLEMENTING THE MODEL

Protocols and encoding schemes are not the real issues that will affect user acceptance. Usability is. Protocols are relatively cheap to define (although not necessarily to implement and deploy!) New protocols are being defined all the time. (Take a look at the number of Internet RFCs and OASIS standards that are being

continually being produced.) The conceptual model, including its security properties, are more important factors from a design perspective. A conceptual model can be mapped onto dozens of different protocols.

Here is one suggestion using existing standard protocols. Other protocol bindings can equally well be chosen.

We suggest the use of X.509 public key certificates (PKCs) for identity files since these are ubiquitous. The user's PKC could be a self issued public key certificate, or a CA issued PKC. It does not really matter which. It does not matter either what distinguished name the public key certificate contains as long as it is unique. IdP/AAs should refuse to issue authorisation credentials to names that they judge to be non-unique, misleading, or likely to cause confusion to SPs. The PKC must contain a new X.509 certificate extension which points to the private key store. This extension will need to be defined, and software will need to be written to support it, both in adding it to new PKCs, and in reading it at relying parties. If open source software is provided, this will ensure faster take up.

We suggest the use of X.509 attribute certificates instead of signed SAML attribute assertions as the authorisation credentials. Whilst SAML attribute assertions have several advantages over X.509 attribute certificates as follows:

- they are an OASIS standard
- they have gained significant traction in the market place
- they are encoded in XML so that users can view (part of) their contents using simple text viewing tools, although this would not verify their signatures nor decrypt their encoded contents

However, the major disadvantage with SAML assertions is that there is no support for revocation in the OASIS standard. This is something of a showstopper until this feature is supported.

The advantages of using X.509 attribute certificates over SAML assertions are as follows:

- they are encoded using ASN.1 in the same way as PKCs so the same software can be used for processing both. Consequently no XML processing tools are required.
- they are compact and much smaller than SAML attribute assertions
- they can be used by mobile phones and other constrained devices such as IBM's ZTIC, since these already have the ability to use ASN.1 encoding and decoding, digital signing and signature validation
- they outperform the use of XML signatures
- they already have standard fields defined for holding revocation information
- it is an ISO/ITU-T standard
- they are used in biometric certificates

For either encoding, a special viewing tool is needed, so that when a user double clicks on a credential file, its contents are displayed, including the encrypted fields which have been decrypted. Also its signature should be validated, rather like public key certificates are validated and displayed in today's web browsers. This requires the user's private key to be present, which is not a problem if the private key is on a portable device like a smart card which can be attached to the viewing machine, but it is a problem if the private key is kept on a separate device such as a mobile phone SIM card and the user is trying to view his credentials elsewhere. For this reason the authorisation credentials

should have an optional clear text display field that the user or issuer can choose to include when they are issued, and in which the user or issuer can insert their own free form text.

In terms of usability, we expect that users will have a password manager in their web browsers in which they can store the multiple user names and passwords needed to authenticate to their various IdPs. Most users already have this today. This gives them an effective single sign on mechanism for getting their authorisation credentials issued. The user can either generate their own X.509 key pair and certificate, or get one issued by one of the numerous CAs that currently exist, and store this in their browser (or preferably keep their private key in a hardware device connected via a PKCS#11 interface). If the user's private key is accessible to the browser, then mutual SSL/TLS authentication can be performed when the user logs in to one of their Identity

Providers over SSL/TLS. Alternatively, their browser can transfer their public key certificate to the server and their private key device can be asked to provide proof of possession of the private key by signing a challenge from the server. After receiving the POP, the IdP can mint the long lived authorisation credential, and allow the user to download it and store it on their local hard drive under whatever filename they wish (see Figure 1). From now on, whenever the user contacts a service provider which requires a set of user credentials, the user simply select the credential files they wish to use through a typical file selection screen, and the browser will send these to the server. The server issues the challenge to the private key device, and this returns the signed response containing the decryption keys for the credentials. The user is assured that they are sending their consent to the correct service provider, since the contents of the challenge (returned

Table 1. Comparison of CardSpace and FileSpace

Feature	Information Cards/CardSpace	FileSpace
Modus Operandi	Short lived authorization tokens issued on demand to user for passing to SP when user authenticates to IdP/AA	Encrypted long lived authz credentials issued by IdP/AA to user to use as required and short lived authentication and decryption tokens issued on demand to SP by user
Authz tokens are portable between devices	Yes, but might be difficult to move identity cards to constrained devices	Yes, user simply copies files
Cards/Files open to attack?	Cards (meta data) are open to attack therefore they have to be strongly protected on the desktop and in the Identity Selector	Credential files are attack proof. Only the user's private authentication key(s) need to be protected
User authentication method at service provision time	Any that the IdP corresponding to the selected card chooses to use.	User proves possession of his private key typically by entering a PIN to his private key storage device
Same authentication credentials for each SP session	No, user must use credentials required by each card issuer	Yes, user uses the same PIN for a given pseudonym regardless of which SP and IdPs are used
User can use multiple authorization credentials from multiple IdPs per transaction	No	Yes
User's privacy is protected at the IdP?	Not always. In auditing mode, the IdP knows all the SPs that the user talks to and when he does this.	Yes. IdP is not aware which SPs user is talking to or when, unless it tracks OCSP requests (but SPs can use CRLs instead)
User's privacy is protected at the SP?	Yes. The IdP can send a one off or permanent pseudonym	Yes. The user determines his own pseudonyms to use when and where
Single Sign On	Yes but only for repeated use of same card. Not if user wishes to use different cards for different SPs	Yes, if private key store allows multiple accesses to private key after initial authentication e.g. input of PIN
User consent	Yes, the user has to select a card before it can be used	Yes, the user must select the authz files and specifically grant the SP the right to decrypt them
Credential renewal required	No, as short lived credentials are issued for each SP session, although IdPs may need to re-issue information cards periodically.	Yes, every time public key certificate expires new authz credentials are needed (typically annually)
Acquiring a new pseudonymous identity	Not needed	User generates his own key pair/PKC or asks a conventional CA to do this.
Acquiring a new authorisation credential	User logs into IdP and asks for a new managed card which is then imported into his Identity Selector	User logs into IdP and asks for a new authorisation file which is then copied to his local filestore. User may also need to enter his PIN into his private key store in order to prove possession of pseudonym.

credentials and SP's name) can be displayed by either the browser or private key device, before the user enters their PIN to unlock their private key.

6. COMPARISON WITH CARDSPACE

The table 1 provides a comparison of the FileSpace model with that of information cards and CardSpace. From this, it can be seen that FileSpace has some advantages over CardSpace, in terms of privacy protection, portability, usability during service provision, and single sign on, but has a couple of disadvantages in that the user has to create her own pseudonyms first and acquire new credentials periodically.

7. CONCLUSIONS

Whilst CardSpace has some notable and worthwhile security and usability properties, nevertheless it has some significant drawbacks as described earlier. In this paper we have looked at the problem of federated identity management from a different perspective, namely, how can we build a system using a paradigm that users are already comfortable with, namely computer files, and from this paradigm, how can we build in the security and usability properties that are necessary for a global identity management system. The resulting system, which we have cheekily called FileSpace, is one possible solution to this complex problem area. The usability advantages of FileSpace are that during service provision, the user provides the same authentication token (typically a PIN) to the same device (his private key store) regardless of the SP or IdPs that are being used, whereas in CardSpace the user has to use the credentials of the particular managed card issuer for each service request. Furthermore in CardSpace the user can only select one card, whereas with FileSpace the user can select as many authorisation files as are required. The procedures for obtaining a new authorization file or managed card are very similar in terms of usability and neither system has an advantage. The disadvantage of FileSpace is that the user has to create one or more pseudonymous identities, in terms of key pairs and PKCs, before he can use the system. This step is not necessary for CardSpace.

8. REFERENCES

- [1] David Chappell. "Introducing Windows CardSpace". MSDN. April 2006. Available from <http://msdn.microsoft.com/en-us/library/aa480189.aspx>
- [2] S. Tuecke, V. Welch, D. Engert, L. Pearlman, M. Thompson. "Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile". RFC3820, June 2004.
- [3] OASIS (2005) "Security Assertion Markup Language (SAML) V2.0", March, available at <http://saml.xml.org/saml-specifications> (accessed 24 October 2008).
- [4] Morgan, R. L., Cantor, S., Carmody, S., Hoehn, W., and Klingenstein, K. (2004), "Federated Security: The Shibboleth Approach", *Educause Quarterly*, Vol. 27, No. 4, available at <http://connect.educause.edu/Library/EDUCAUSE+Quarterly/FederatedSecurityTheShibb/39889> (accessed 24 October 2008).
- [5] David W Chadwick, Sean Anthony. "Using WebDAV for Improved Certificate Revocation and Publication". In LCNS 4582, "Public Key Infrastructure. Proc of 4th European PKI Workshop, June, 2007, Palma de Mallorca, Spain. pp 265-279.
- [6] ISO 9594-8/ITU-T Rec. X.509 (2005) The Directory: Public-key and attribute certificate frameworks
- [7] 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
- [8] Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C. "X.509 Internet Public Key Infrastructure: Online Certificate Status Protocol – OCSP", RFC 2560, June 1999.
- [9] Thomas Weigold, Thorsten Kramp, Reto Hermann, Frank Höring, Peter Buhler, Michael Baentsch. "The Zurich Trusted Information Channel – An Efficient Defence against Man-in-the-Middle and Malicious Software Attacks". In P. Lipp, A.-R. Sadeghi, and K.-M. Koch (Eds.): TRUST 2008, LNCS 4968, pp. 75–91, 2008.
- [10] Landau, S. and Mulligan, D.K. "I'm Pc01002/SpringPeep/ED2881.6; Who are You?", IEEE Security and Privacy, Vol 6, No 2, March/April 2008, pp13-15
- [11] Dhamua, R. and Dusseault, L. "The Seven Flaws of Identity Management", IEEE Security and Privacy, Vol 6, No 2, March/April 2008, pp24-29.
- [12] William E. Burr, Donna F. Dodson, Ray A. Perlner, W. Timothy Polk, Sarbari Gupta, Emad A. Nabbus. "Electronic Authentication Guideline", NIST Special Publication NIST Special Publication 800-63-1, Feb 2008
- [13] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. Available from http://ec.europa.eu/justice_home/fsj/privacy/law/index_en.htm
- [14] Arun Nanda. "Identity Selector Interoperability Profile V1.0" April, 2007. Microsoft Corporation.
- [15] Arun Nanda, Michael B. Jones. "Identity Selector Interoperability Profile V1.5" July 2008. Microsoft Corporation
- [16] D. Mundy and D.W. Chadwick. "An XML alternative for performance and security: ASN.1." IEEE IT Professional, 6(1):30-36, 2004.

FileSpace

An alternative to CardSpace that supports MultipleToken
Authorisation and Portability Between Devices

David Chadwick
University of Kent

Contents

- Problem statement and solution idea
- User experiences of using InfoCards and FileSpace
 - At service provision time
 - At Card/File issuing time
- Technical properties of FileSpace
- Detailed Comparison of FileSpace and InfoCards
- Conclusion

Problem Statement

- A user typically has multiple cards
 - today each plastic card issuer only puts one attribute on a card, Visa member, AAA frequent flyer, IEEE member etc. so why expect that in InfoCards all this information will be on one card. It wont.
- But she can only use **one** of these cards in any given InfoCard/CardSpace transaction
- Insufficient for many purposes
 - buy a book at a discount using Visa card and student card
 - access patient data using Doctor card and hospital employee card
- Users need to be able to select/present multiple cards
- Cards may not be easily transported to all devices
 - e.g. use on a mobile phone
 - initial version of CardSpace did not have an export capability

Solution Idea

- Instead of cards, use files
- Files are an existing well known concept to all computer users
- Users are already familiar them, know how to drag and drop, copy, delete them etc.
- So if every credential (plastic card) becomes a file, then user can copy them easily between devices, send multiple files to service providers etc.

Example user's directory with FileSpace files

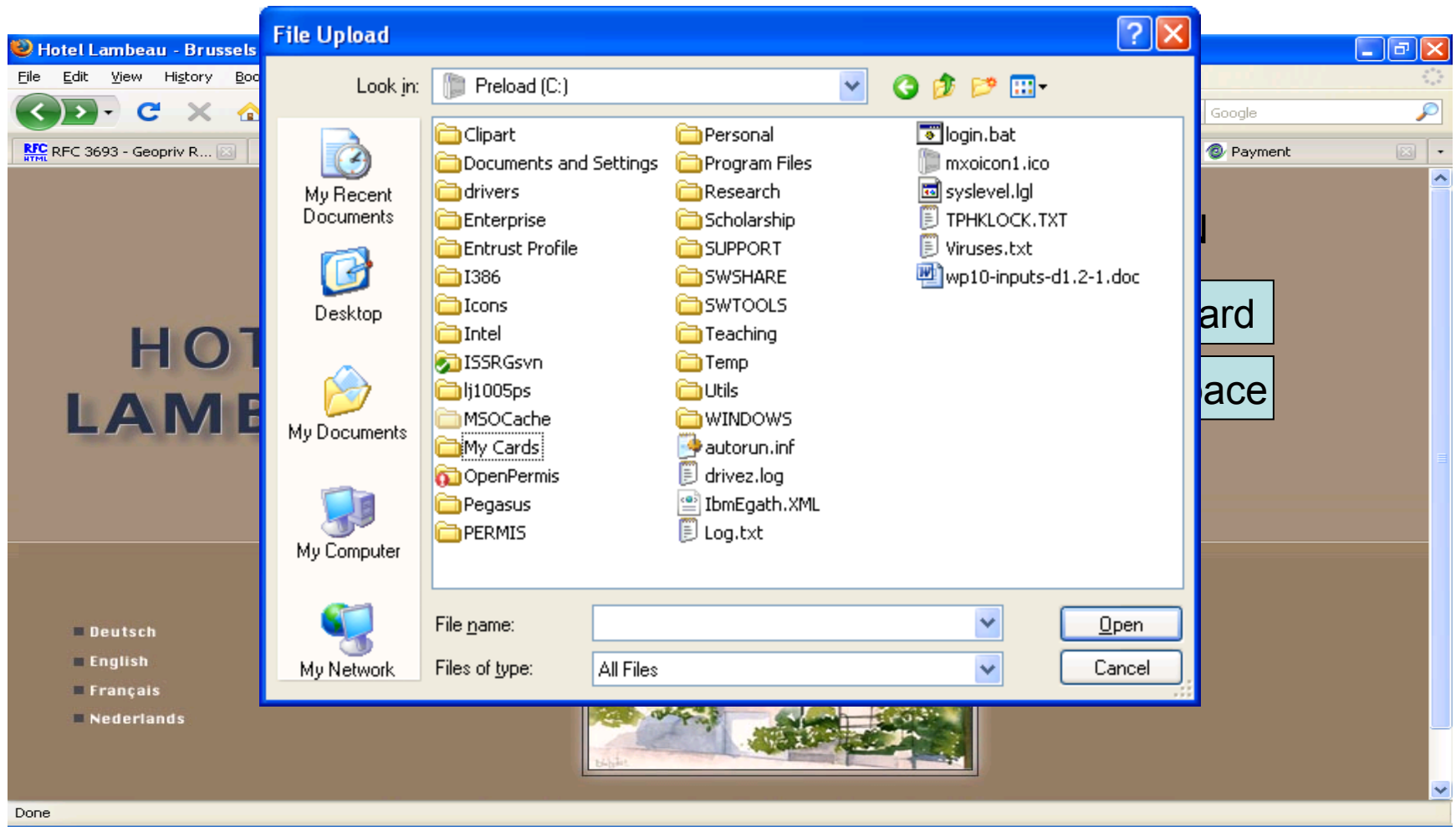
The screenshot shows a Windows Explorer window titled "C:\My Cards\Father Christmas". The address bar displays the path "C:\My Cards\Father Christmas". The left pane shows a tree view of folders, with "Father Christmas" selected. The right pane displays a list of files in a table format.

Name	Size	Type	Date Modified
Amex.card	1 KB	CARD File	02/06/2008 14:12
BA.card	1 KB	CARD File	02/06/2008 14:15
Identity.card	1 KB	CARD File	02/06/2008 14:32
TAS3.card	1 KB	CARD File	02/06/2008 14:16
UnivKent.card	1 KB	CARD File	02/06/2008 14:14
Visa.card	1 KB	CARD File	02/06/2008 14:12

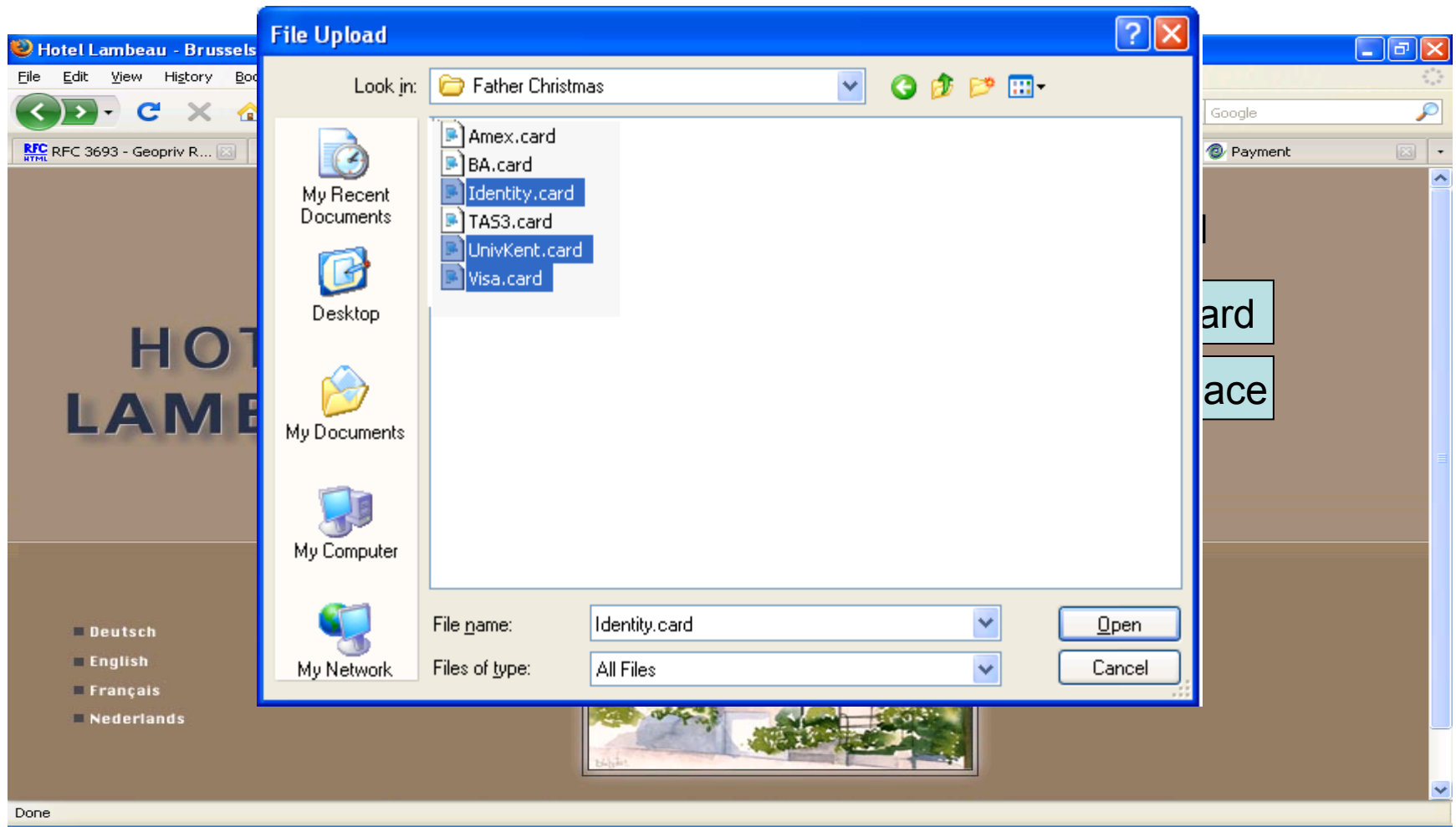
Comparison of User Experiences - at Service Provision Time

The screenshot shows a Mozilla Firefox browser window displaying the website for Hotel Lambeau in Brussels. The browser's address bar shows the URL <http://www.hotellambeau.com/>. The website layout includes the hotel's name 'HOTEL LAMBEAU' in large blue letters on the left, a central photograph of the hotel building, and a 'LOGIN' section on the right. The 'LOGIN' section contains two buttons: 'InfoCard' and 'FileSpace'. A red arrow points to the 'FileSpace' button with the text 'Click' written below it. In the bottom left corner of the browser window, the text 'Done' is visible. The browser's taskbar at the bottom shows several open tabs, including 'RFC 3693 - Geopriv R...', 'Itinerary 11962550921', 'Flight Search - Availa...', 'Hotel Lambeau - B...', 'easyJet.com - book c...', and 'Payment'.

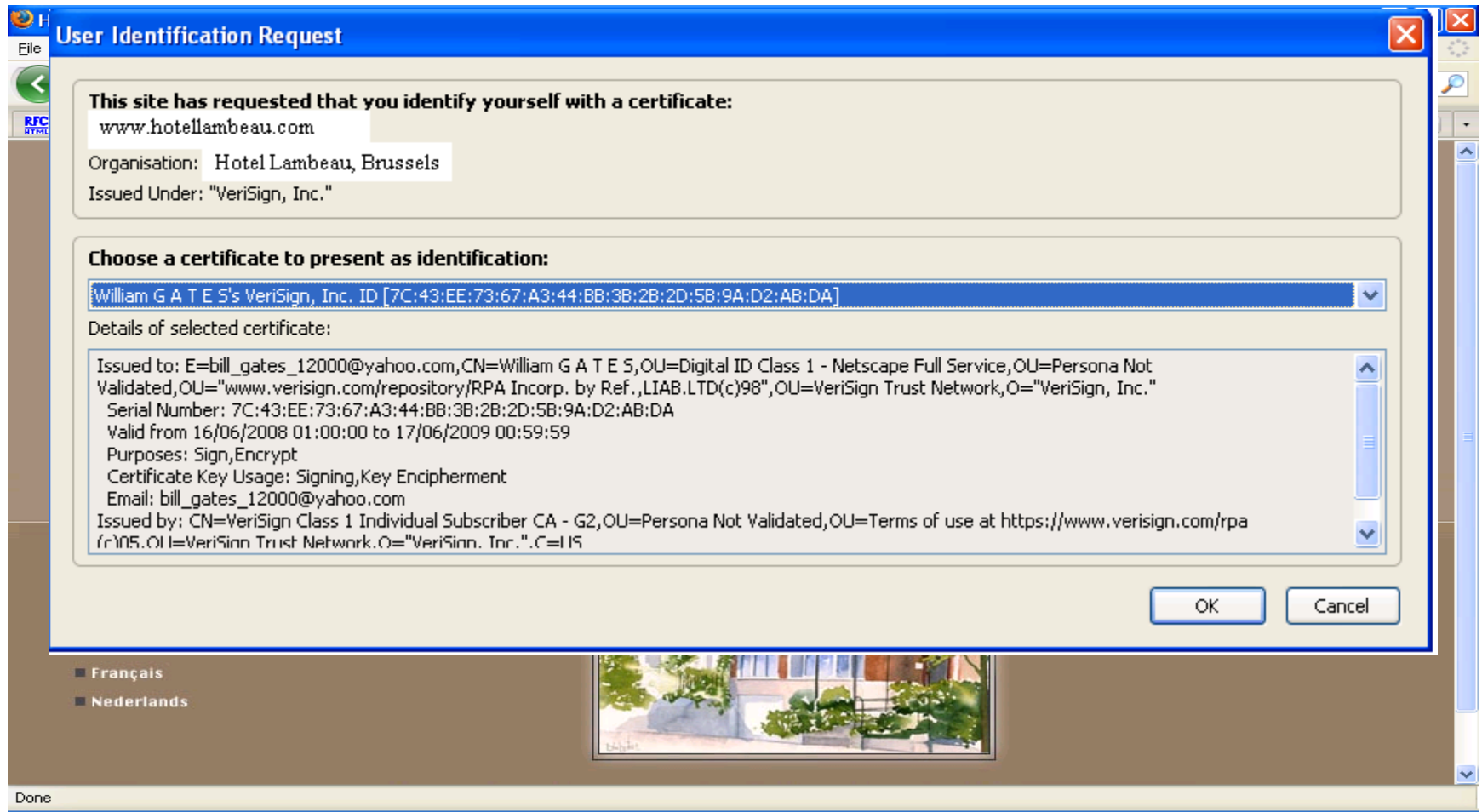
User Selects FileSpace



User Selects Files To Upload



User is asked to Confirm he is Father Christmas



User's Private Key could be in a hardware token



IBM's ZTIC USB device



Mobile Phone

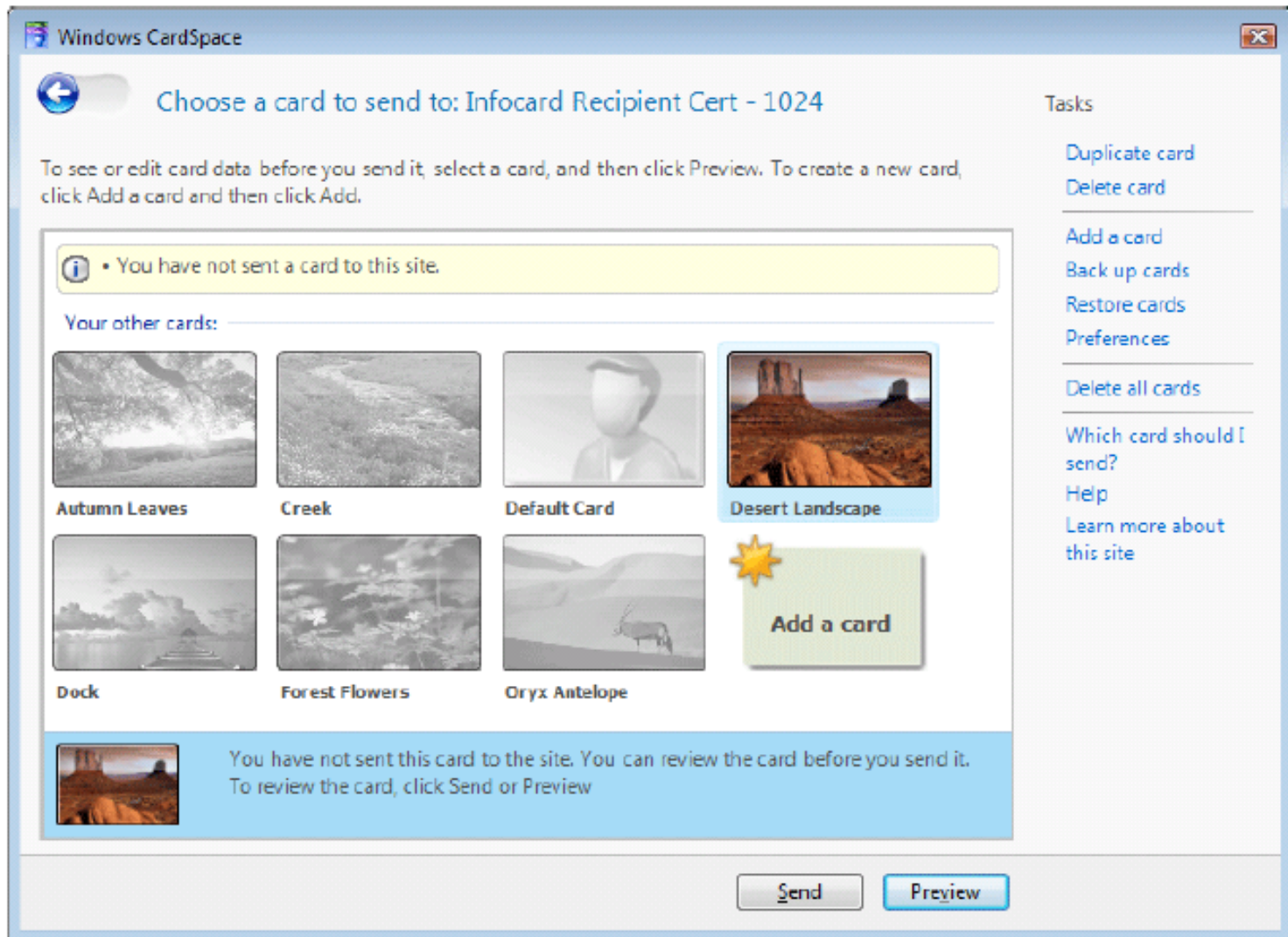
User is provided with service

The screenshot shows a Mozilla Firefox browser window displaying the Hotel Lambeau - Brussels reservation page. The browser's address bar shows the URL http://www.hotellambeau.com/reservation_en.php. The website header includes the hotel name "HOTEL LAMBEAU" and "Brussels" with a logo. A navigation menu contains "Presentation", "Location", "Room", "Rates", "Book a room", and "Links". The "Book a room" section is active, showing four room type options: "Single", "Twin", "Double", and "Mini-Suite". Each option is accompanied by a diagram of the room's bed configuration. Below the room type options, there are radio buttons for "Single", "Twin", "Double", and "Mini-Suite". The "Single" radio button is selected. Below the room type options, there are dropdown menus for "Arrival date" (15/04/2009) and "Nb of nights" (1), followed by a "search" button. The browser's status bar at the bottom shows "Done".

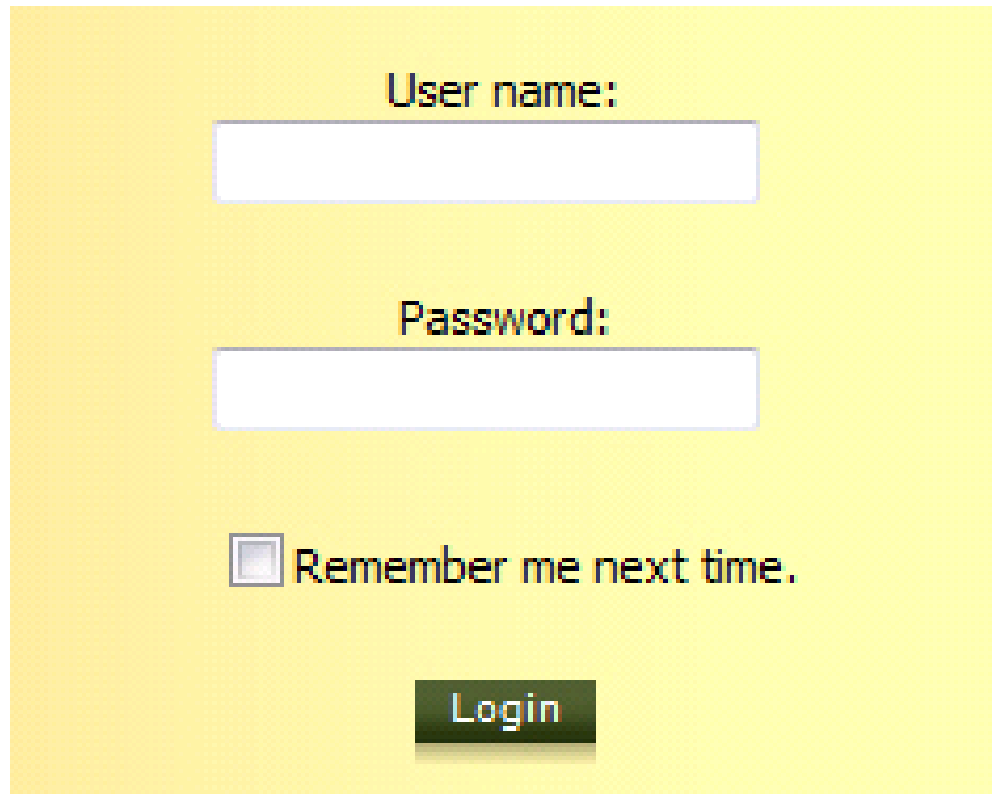
Comparison of User Experiences - at Service Provision Time

The screenshot shows a Mozilla Firefox browser window displaying the Hotel Lambeau website. The browser's title bar reads "Hotel Lambeau - Brussels - Mozilla Firefox". The address bar contains "http://www.hotellambeau.com/". The website layout includes the hotel name "HOTEL LAMBEAU" in large blue letters on the left, a central image of the hotel building, and a "LOGIN" section on the right. The "LOGIN" section contains two buttons: "InfoCard" and "FileSpace". An orange arrow points to the "InfoCard" button with the word "Click" next to it. The browser's status bar at the bottom left says "Done".

User is asked to choose a single InfoCard



User is asked to provide password for card provider



User name:

Password:

Remember me next time.

Login

User is provided with service

The screenshot shows a Mozilla Firefox browser window displaying the Hotel Lambeau - Brussels reservation page. The browser's address bar shows the URL http://www.hotellambeau.com/reservation_en.php. The website header includes the hotel name "HOTEL LAMBEAU" and "Brussels" with a logo. A navigation menu contains links for "Presentation", "Location", "Room", "Rates", "Book a room", and "Links". The "Book a room" section is active, showing four room type options: "Single", "Twin", "Double", and "Mini-Suite". Each option is accompanied by a diagram of the room's bed configuration. Below the room type options, there is a "Select a room type:" section with radio buttons for "Single", "Twin", "Double", and "Mini-Suite". The "Single" option is selected. Below this, there is an "Arrival date and number of nights:" section with dropdown menus for "Arrival date" (set to 15/04/2009) and "Nb of nights" (set to 1), followed by a "search" button.

Hotel Lambeau - Brussels - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.hotellambeau.com/reservation_en.php

RFC 3693 - Geopriv Requirements Itinerary 11962550921 Coram Premier Sliding Door Hotel Lambeau - Brussels

www.hotellambeau.com Send a em@il Home Contact us DE EN FR NL

HOTEL LAMBEAU

Brussels

Presentation - Location - Room - Rates - Book a room - Links

Book a room

Single Twin Double Mini-Suite

Select a room type:

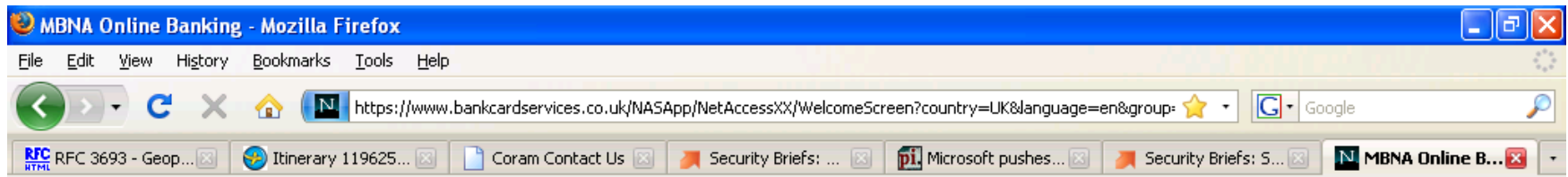
Single : Twin : Double : Mini-Suite :

Arrival date and number of nights :

Arrival date : 15 04 2009 Nb of nights : 1 >>>> search

Done

User Experience at Enrolment e.g. to get an electronic MasterCard



mbna

[About Online Banking](#) • [Security Statement](#) • [Email Security](#)

Online Banking

[Learn more](#) | [View demo](#)

User Name:

Password:

[Forgotten your logon details?](#)

First Time Visitor?
Enrol in online banking now.

Easy. Secure. Free.

Go Paperless
and view your statements online...

Card Protection
Protecting more than just your cards.
[Learn More >>](#)

Protect Your PC
40% discount on Norton Internet Security software
[Click Here >>](#)

Principles
on interest rate changes - our commitment to you.
[Click Here >>](#)

User must Login to his account (over SSL)

© 2009 MBNA Europe Bank Limited

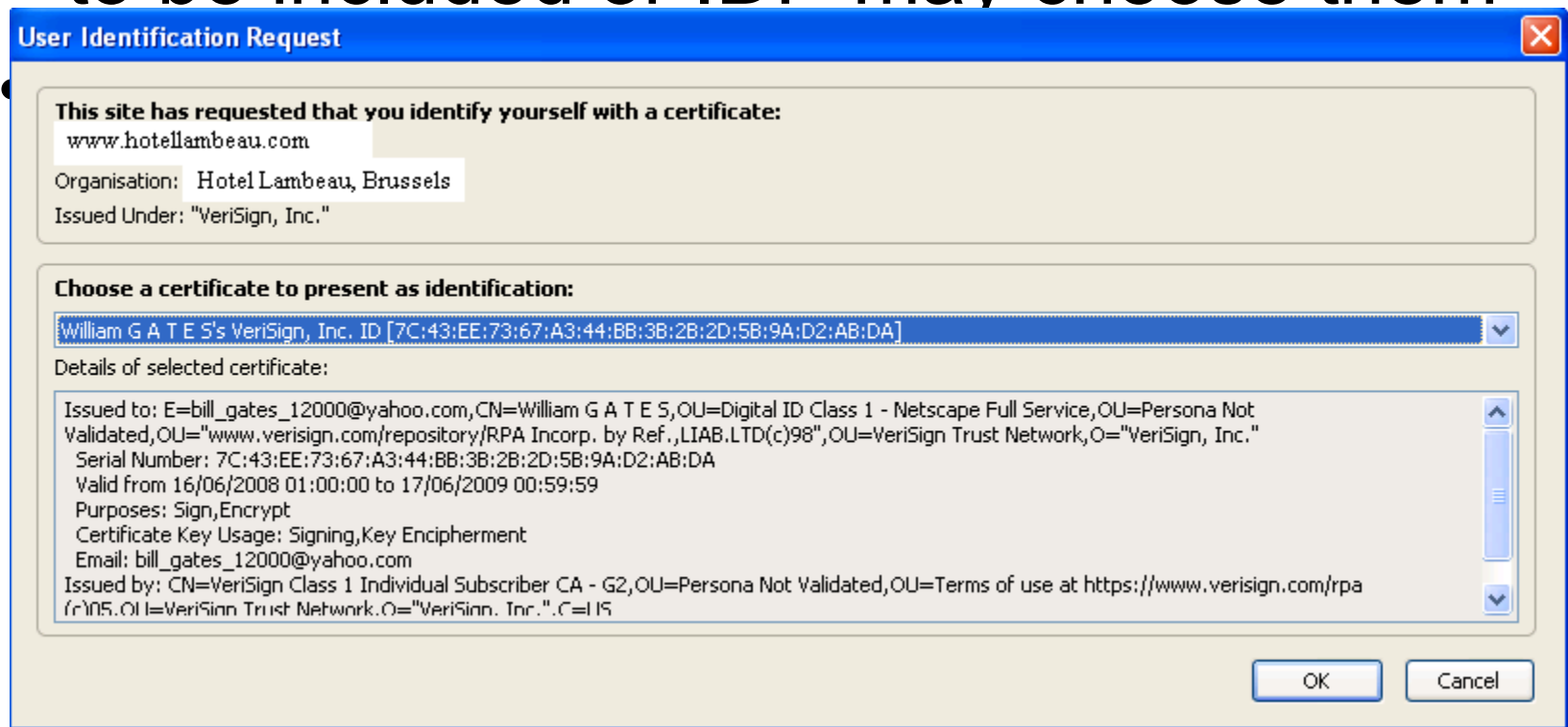
[Contact Us](#) • [Privacy](#) • [Terms of Use](#)

Done

www.bankcardservices.co.uk

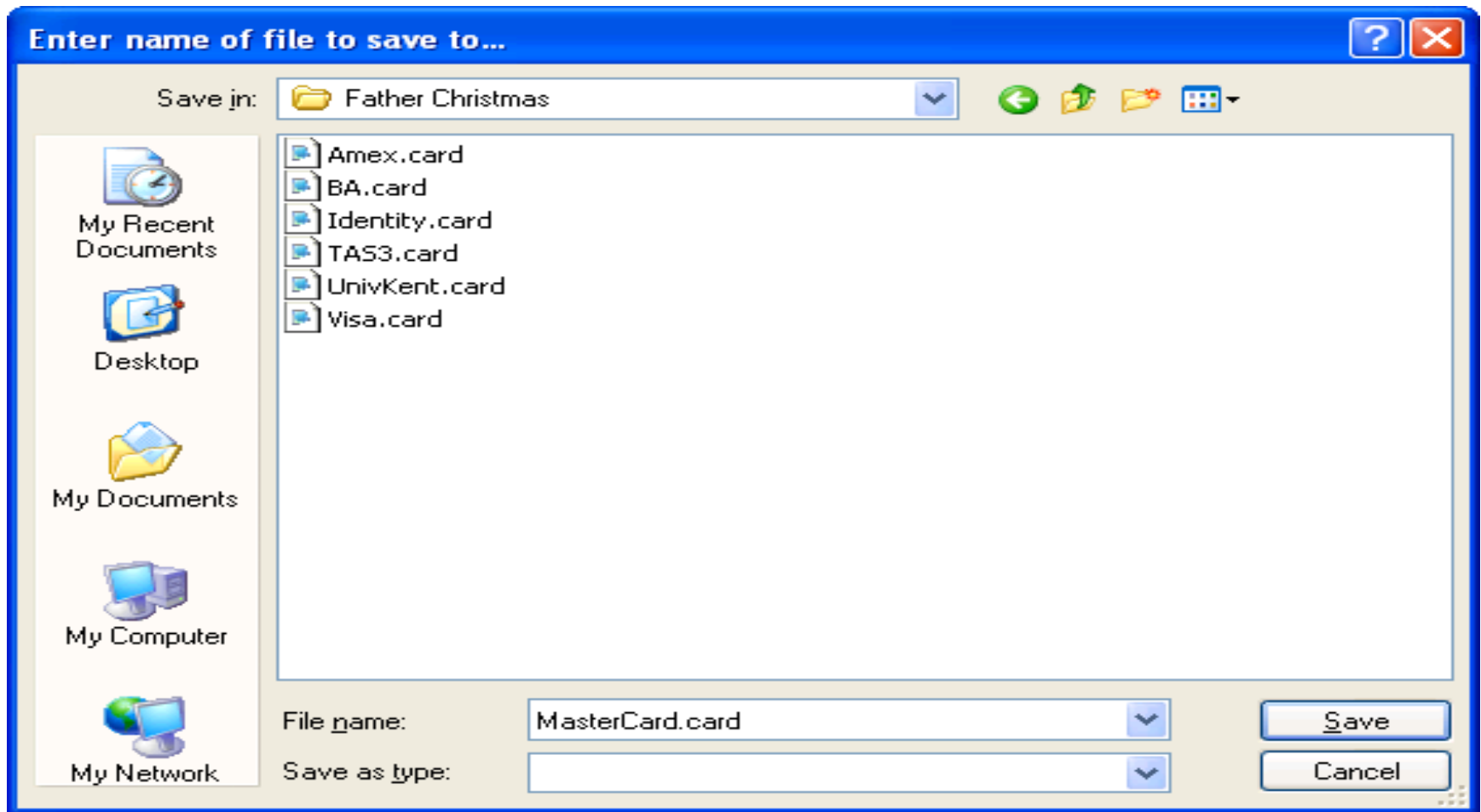
Then Ask For Electronic Card/File to be Created

- User may need to select which attribute(s) to be included or IDP may choose them

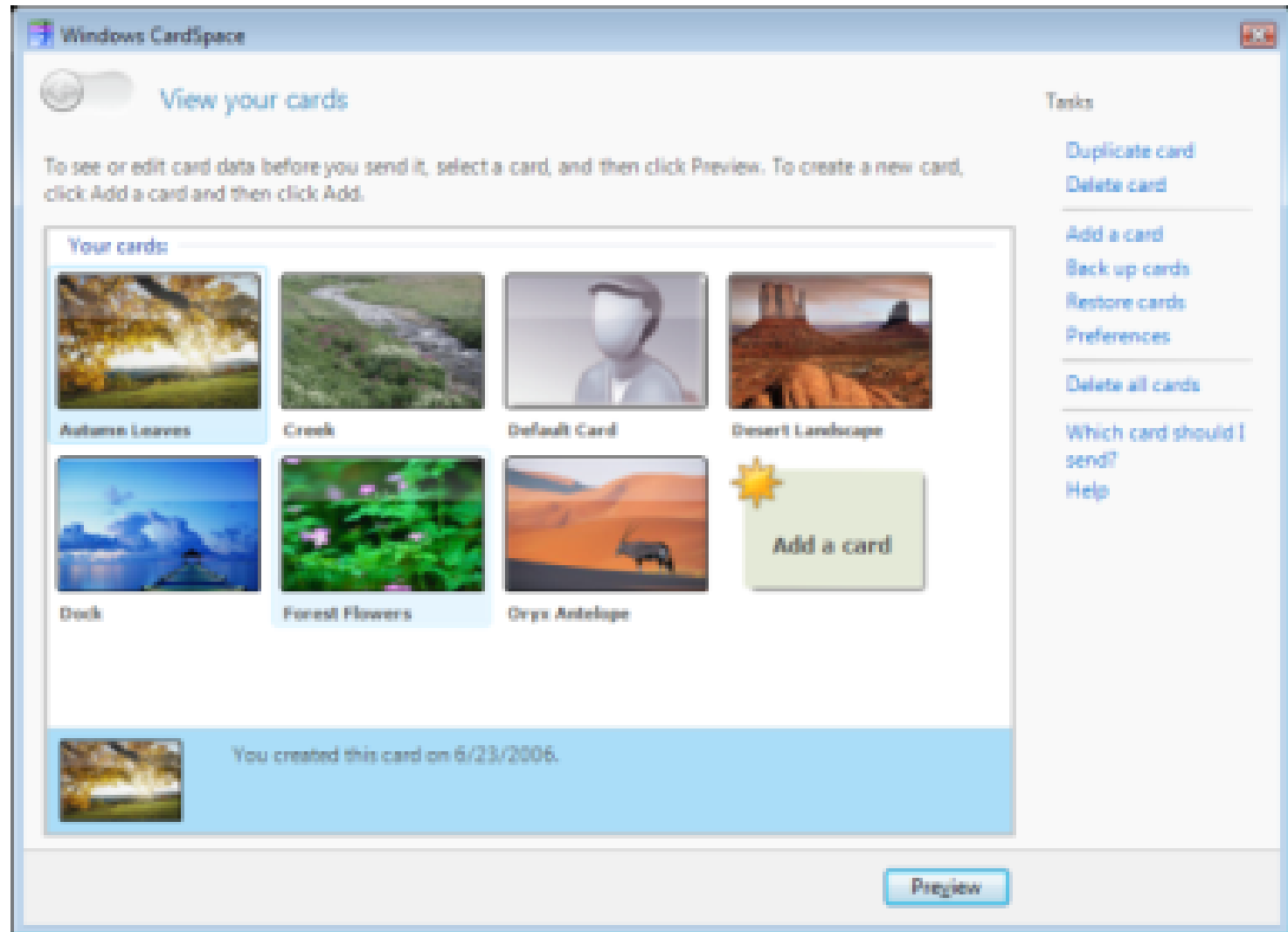


of

Then Select Where to Download File To



In CardSpace User then has to Import file into Card Selector



Creating a New Identity

- In CardSpace we have self issued cards, whereby a user enters his own attributes into his cards
- In FileSpace we have new identity cards, either self issued (e.g. Father Christmas) or issued by a CA such as Versign (e.g. my Bill Gates cert) (so not much difference there then !)

Basic Principles of FileSpace

- Clearly Separate Authentication from Authorisation by having separate tokens
- Have multiple Authz Tokens linked to one Authn Token
 - Each Authz Token provides an attribute assertion from a trusted authoritative source
- Have one Authn Token per Pseudonym
 - this is purely a handle on which to hang the Authz tokens. The pseudonym can be anything
 - user can have as many pseudonyms as he wishes

Token Lifetimes

- All FileSpace tokens are long lived and can be used repeatedly
 - User has to authenticate *to service provider* that he is the holder of all of them at time of use
- In CardSpace all tokens are short lived and issued on demand
 - this requires authn to *the card issuer* each time a service is required
- Each FileSpace Authz Token can be independently revoked by the issuer (AA)
 - so if user loses his private key he can ask AAs to revoke his authz credentials (no need to revoke authn credential)

Authn Tokens

- Public key certificate containing any subject DN the issuer chooses to put there, subject to it being globally unique (which is mandatory in X.509 anyway!!)
- Can be user generated (self issued)
 - unique DN can be generated by having public key id RDN + user provided CN RDN
- Can be CA generated
 - unique DN can be CA name plus user provided CN RDN
- It does not matter since it is not used for authorisation, only for authentication
- Contains a pointer to where relying party can find private key in order to validate that the current user is the holder of the private key – solves the Discovery problem
 - could be mobile phone number, or “user’s browser”

Authz Tokens

- Are standard claims/assertions/certificates that say this subject has this attribute, signed by issuer
- However, subject id and attribute are encrypted (indirectly) to key public key of subject
 - Can be lost or stolen but are worthless to finder/thief because he cant decrypt them
- Only valid once subject proves to RP that he has the decryption key
- In practise we encrypt to a symmetric key and encrypt symmetric key to public key of subject then subject can decrypt symmetric key and encrypt it to public key of RP allowing RP to read the contents

Service Provision

- User selects set of Authz Tokens and the matching Authn Token to send to service provider (through file upload function)
- SP reads location of private key from Authn token and sends a message containing tokens and asking for decryption keys
- User is asked to confirm SP can have these tokens, then enters PIN to private key and device creates decryption keys for the SP and returns them

Protocol Exchange

- SP-> private key location:
 $\{\{\text{authzCred}_i\} \ i=1 \text{ to } n, \text{nonce1}, \text{ts1}, \text{SPPKC}\}\text{signSP}$
- Private key location -> SP:
 $\{\{\text{sn}_i, \text{encKey}_i\} \ i=1 \text{ to } n, \text{nonce1}, \text{nonce2}, \text{ts2}, \text{SP}\}\text{signUser}$
- Where:
 - n is the number of authorisation credentials to be decrypted
 - SPPKC is public key certificate of Service Provider
 - nonce is a random number and ts is a short time in the future (say 2 seconds)
 - sn_i is serial number of ith authorisation credential
 - encKey_i is symmetric key used to encrypt the contents of authorisation credential i, encrypted to the public key of the recipient SP

Detailed Comparison (1)

Feature	Information Cards CardSpace	FileSpace
Modus Operandi	Short lived authorization tokens issued on demand by IdP to user for passing to SP when user authenticates to IdP/AA	Encrypted long lived authz credentials issued by IdP to user to use as required and short lived authentication and decryption tokens issued on demand to SP by user
Authz tokens are portable between devices	Yes, but might be difficult to move identity cards to constrained devices	Yes, user simply copies files
Cards/Files open to attack?	Cards (meta data) are open to attack therefore they have to be strongly protected on the desktop and in the Identity Selector	Credential files are attack proof. Only the user's private authentication key(s) need to be protected and these can be stored in hardware
User authentication method at service provision time	Any that the IdP corresponding to the selected card chooses to use.	User proves possession of his private key typically by entering a PIN to his private key storage device
Same authentication credentials for each SP session	No, user must use credentials required by each card issuer	Yes, user uses the same PIN for a given pseudonym regardless of which SP and IdPs are used

Detailed Comparison (2)

Feature	Information Cards CardSpace	FileSpace
User can use multiple authorization credentials from multiple IdPs per transaction	No	Yes
User's privacy is protected at the IdP?	Not always. In auditing mode, the IdP knows all the SPs that the user talks to and when he does this.	Yes. IdP is not aware which SPs user is talking to or when, unless it tracks OCSP requests (but SPs can use CRLs instead)
User's privacy is protected at the SP?	Yes. The IdP can send a one off or permanent pseudonym	Yes. The user determines his own pseudonyms to use when and where
Single Sign On	Yes but only for repeated use of same card. Not if user wishes to use different cards for different SPs	Yes, if private key store allows multiple accesses to private key after initial authentication e.g. input of PIN
User consent	Yes, the user has to select a card before it can be used	Yes, the user must select the authz files and specifically grant the SP the right to decrypt them

Detailed Comparison (3)

Feature	Information Cards CardSpace	FileSpace
Credential renewal required	No, as short lived credentials are issued for each SP session, although IdPs may need to re-issue information cards periodically.	Yes, every time public key certificate expires new authz credentials are needed (typically annually)
Acquiring a new pseudonymous identity	Not needed	User generates his own key pair/PKC or asks a conventional CA to do this.
Acquiring a new authorisation credential	User logs into IdP and asks for a new managed card which is then imported into his Identity Selector	User logs into IdP and asks for a new authorisation file which is then copied to his local filestore. User may also need to enter his PIN into his private key store in order to prove possession of pseudonym.

Merging InfoCards and FileSpace

Father Christmas

David Chadwick

The screenshot shows the Windows CardSpace application window titled "Windows CardSpace". The main heading is "Choose a card to send to: Infocard Recipient Cert - 1024". Below this, there is a message: "To see or edit card data before you send it, select a card, and then click Preview. To create a new card, click Add a card and then click Add." A yellow information box states: "You have not sent a card to this site." Under "Your other cards:", there are seven card thumbnails: "Autumn Leaves", "Creek", "Default Card", "Desert Landscape" (which is selected and highlighted in blue), "Dock", "Forest Flowers", and "Oryx Antelope". A green "Add a card" button with a star icon is also visible. At the bottom, there is a blue box with the text: "You have not sent this card to the site. You can review the card before you send it. To review the card, click Send or Preview". At the very bottom of the window are "Send" and "Preview" buttons. On the right side, there is a "Tasks" menu with options: "Duplicate card", "Delete card", "Add a card", "Back up cards", "Restore cards", "Preferences", "Delete all cards", "Which card should I send?", "Help", and "Learn more about this site".

Conclusion

- FileSpace overcomes a major disadvantage of CardSpace/InfoCards, that of not being able to send multiple cards
- FileSpace does not lose out on usability since users already know how to use files (and it could be integrated into Card Selectors)
- It has a number of security advantages as well
 - better privacy protection at the IdP
 - don't need to worry about securing the desktop (unless private keys are held there)

Any Questions?

Authorization Models

Radia Perlman
Radia.Pperlman@sun.com

Important problems

- Something that is understandable for someone to manage the policy
- Something that is efficient for a system to check policy
 - checking if A is allowed to do X when A asks to do X
 - checking everything A is allowed to do
 - checking who is allowed to do X
- Updating policy (including revocation) must be comprehensible, efficient, and timely

Stake in the ground

- Basically, most models map to groups and ACLs

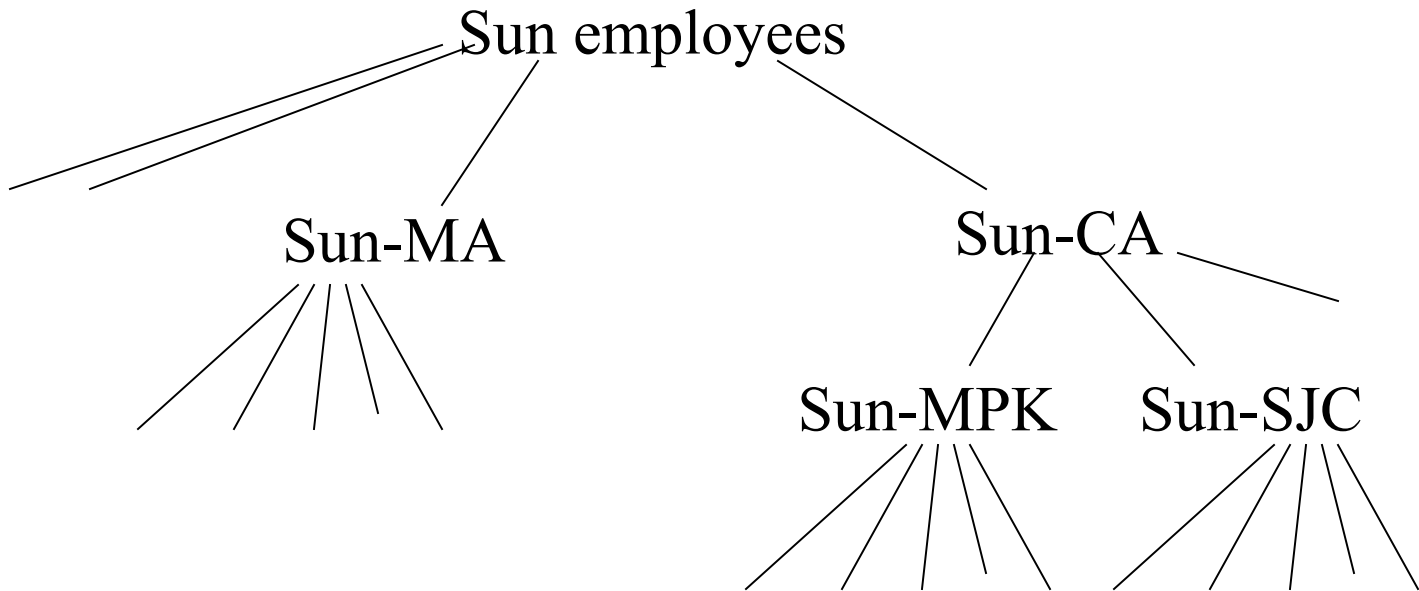
ACLs

- Associated with each resource is an ACL
 - set of (Who, what they can do)
 - Note: “resource” can be a set of resources, all with a common ACL
- Can be fancier
 - other things like time of day, IP addresses from which things must be accessed

What is who?

- Any Boolean combination of
 - Individuals
 - Groups
 - “Roles”
- Groups and roles are also any Boolean combination of individuals, groups, and roles
- Which means groups can be arbitrarily nested

Nested groups



Roles vs Groups

- Mostly in the literature used interchangeably
- Possible distinctions
 - Roles have to be explicitly invoked, and might be mutually exclusive, and might require authentication, vs groups: always a member of all groups you are a member of
 - Roles have names (like “administrator”) that are local to a resource

Attributes

- Can be treated like a group
- “over 21” can be “set of people over 21”
- “paid member of ACM” can be “set of people who have paid ACM membership”

Models around “what is A allowed to do”

- Really not “centrally controlled”
- Only within a “scope”
- Just like ACL on a file
 - Alice: read, write
 - Bob: read
 - Carol: read, write, delete

Proving membership

- Could have some things in your (name/key) cert
- Or could have a separate credential
 - Such as a cert vouching:
 - (public key, attribute/group name)
 - (name, attribute/group name)
 - Or knowledge of a group secret
 - Or coming from an IP address in the US
- Note: authorization doesn't necessarily imply you have to identify yourself

X.509 attribute cert model

- Attribute, like “clearance”, has an OID
- You need a separate PMI (privilege management infrastructure) starting with a SOA (start of authority) to vouch for the attribute
- You’d say “I trust US navy” for clearance

Name-based model

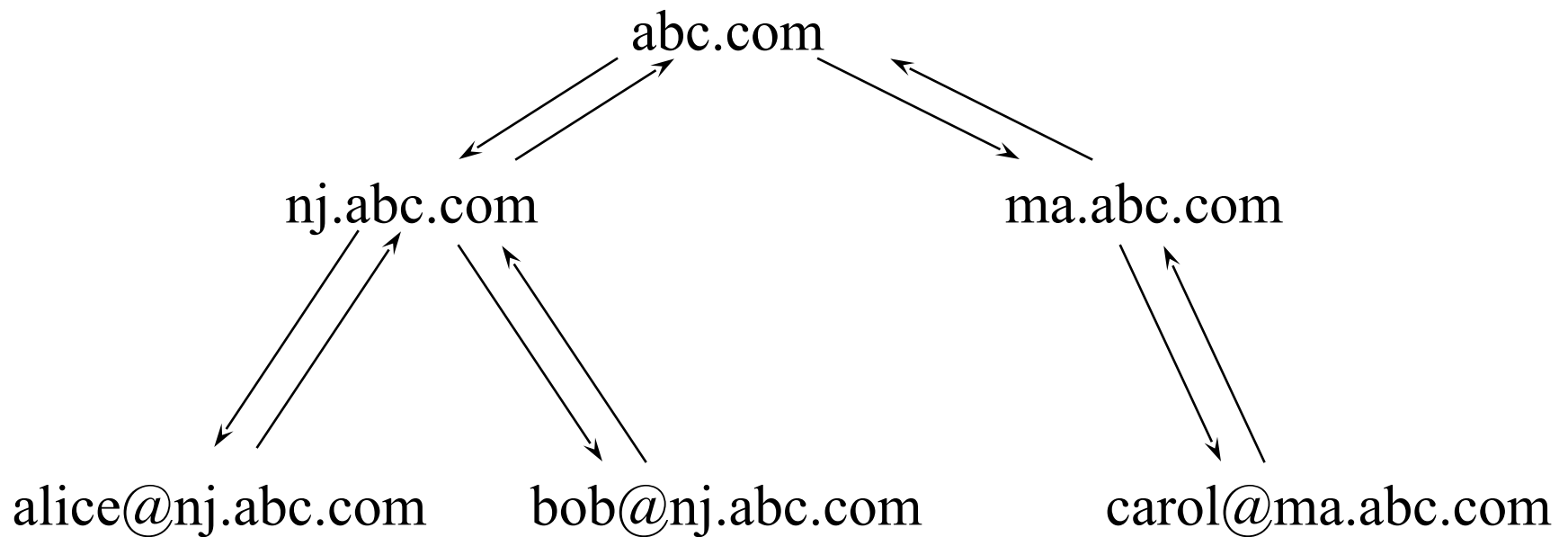
- Hierarchical name
- Name of attribute implies who is trusted to assert it
- gov.US.navy.clearance is a totally different attribute from gov.Russia.KGB.clearance

Name based trust chains, both for
identity and authorization

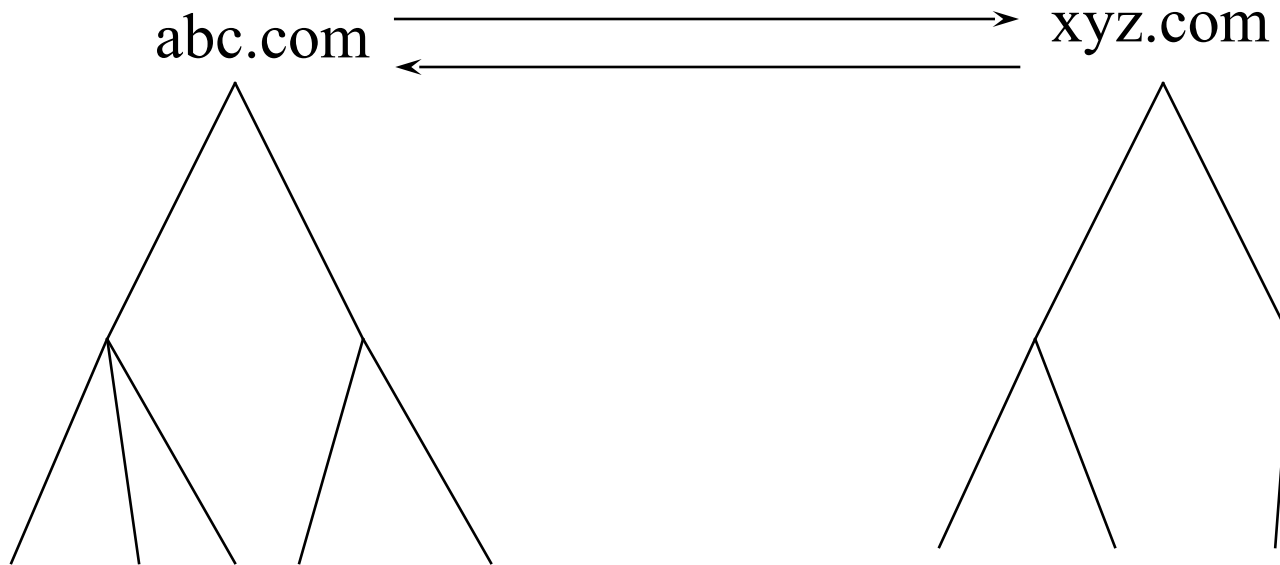
Bottom-Up Model

- Each arc in name tree has parent certificate (up) and child certificate (down)
- Name space has CA for each node
- “Name Subordination” means CA trusted only for a portion of the namespace
- Cross Links to connect Intranets, or to increase security
- Start with your public key, navigate up, cross, and down

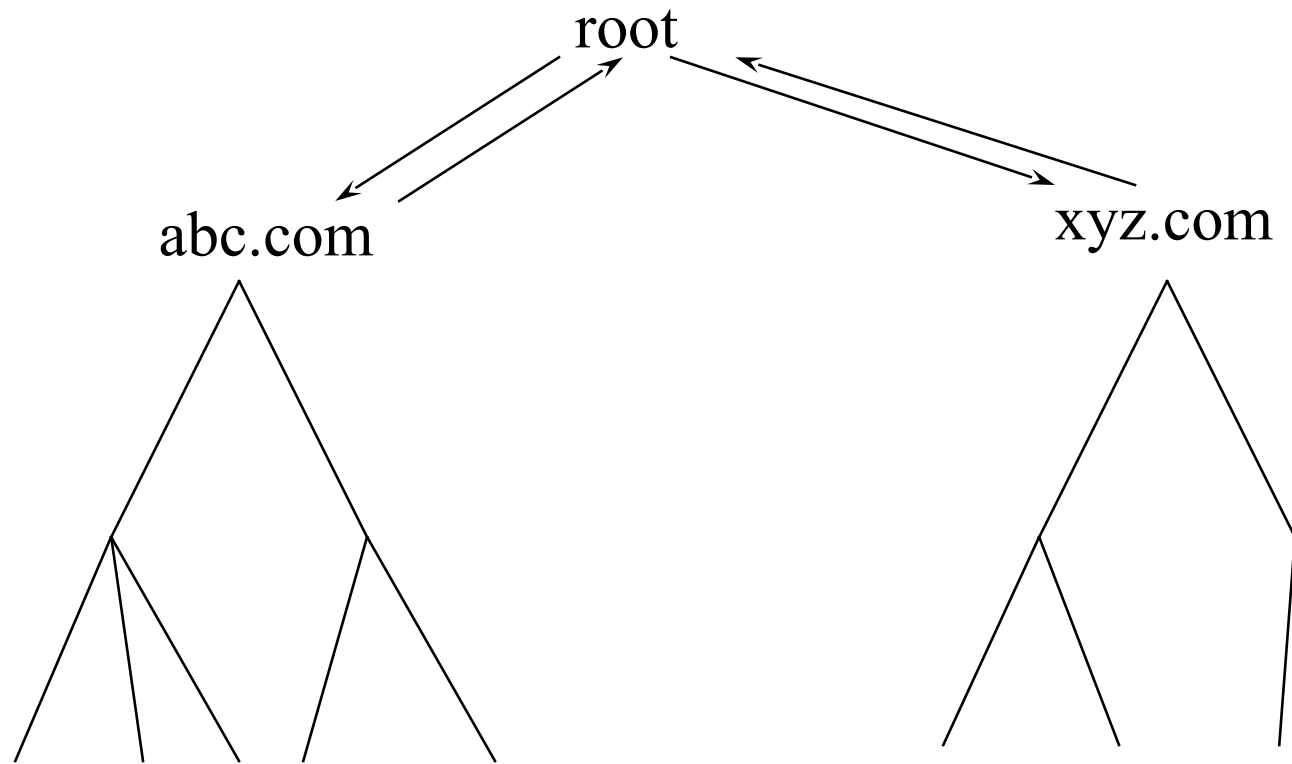
Intranet



Extranets: Crosslinks



Extranets: Adding Roots



Conclusion

- Groups, ACLs, Identities have been around for years
- Can do anything that the other models do

Aligning Access Control models with Attributes (RBAC, ResBAC, RiskBAC, TrustBAC & ABAC)

IDTrust 2009, 15th April

Rakesh Radhakrishnan
Principle Architect (Telco)
Technology Lead (OpenSSO)
Sun Microsystems, Inc.

Agenda

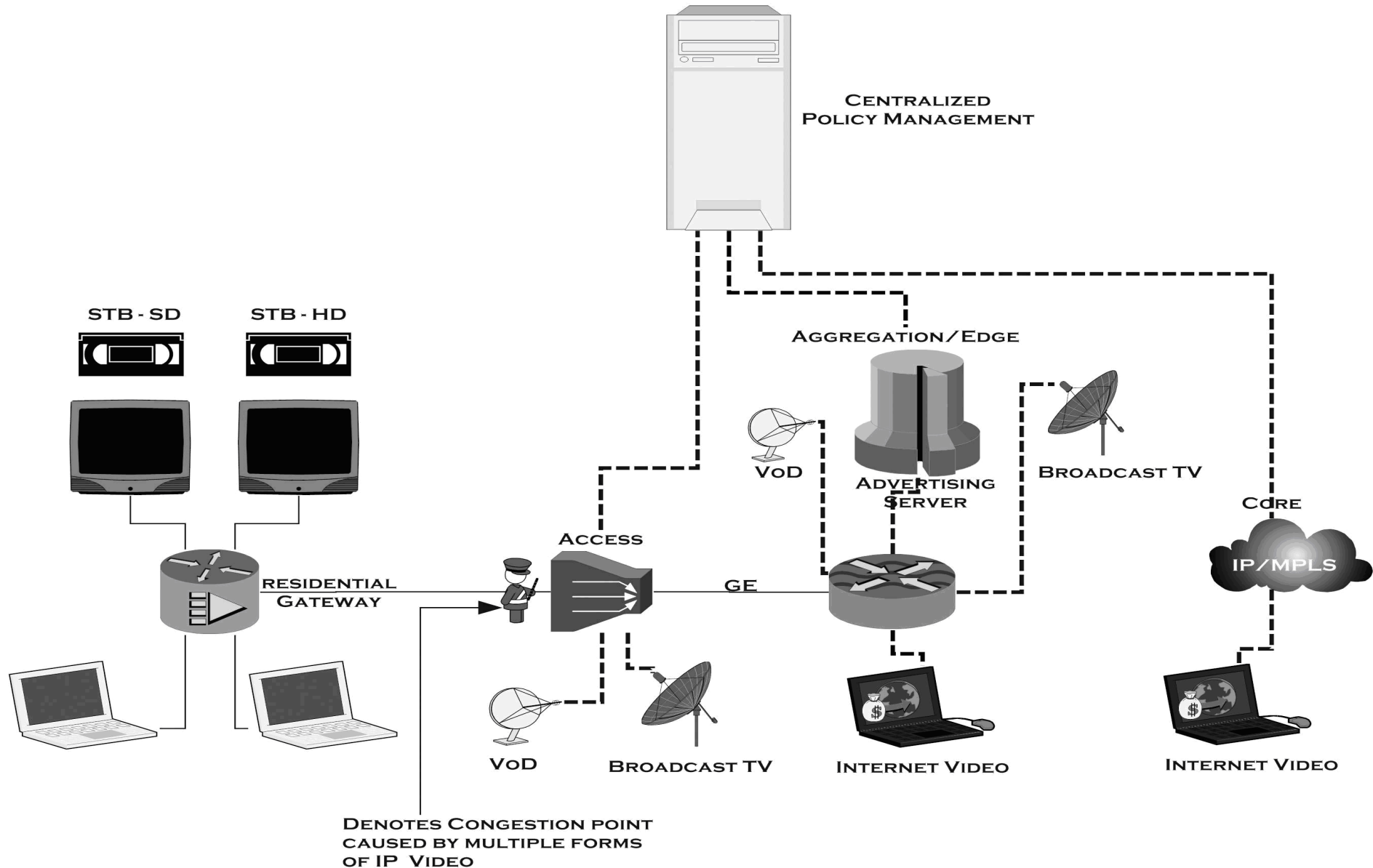
- **Context Aware Security (Adaptive AuthN+ AuthZ)**
- **The 5 AC models**
- **Aligning the AC models**
- **Aligning AuthN with AuthZ**
- **Alignment using Attributes**

Context Aware Security

- Adaptive AuthN (takes into account context and risk)
- Adaptive AuthZ (policy based adaptation of AC models)
- Alignment of Network, Resource and Service policies with Users
- Emphasis is on multiple Attribute Authorities Assertions
- Implementation via Abstraction and Master/Macro PDP

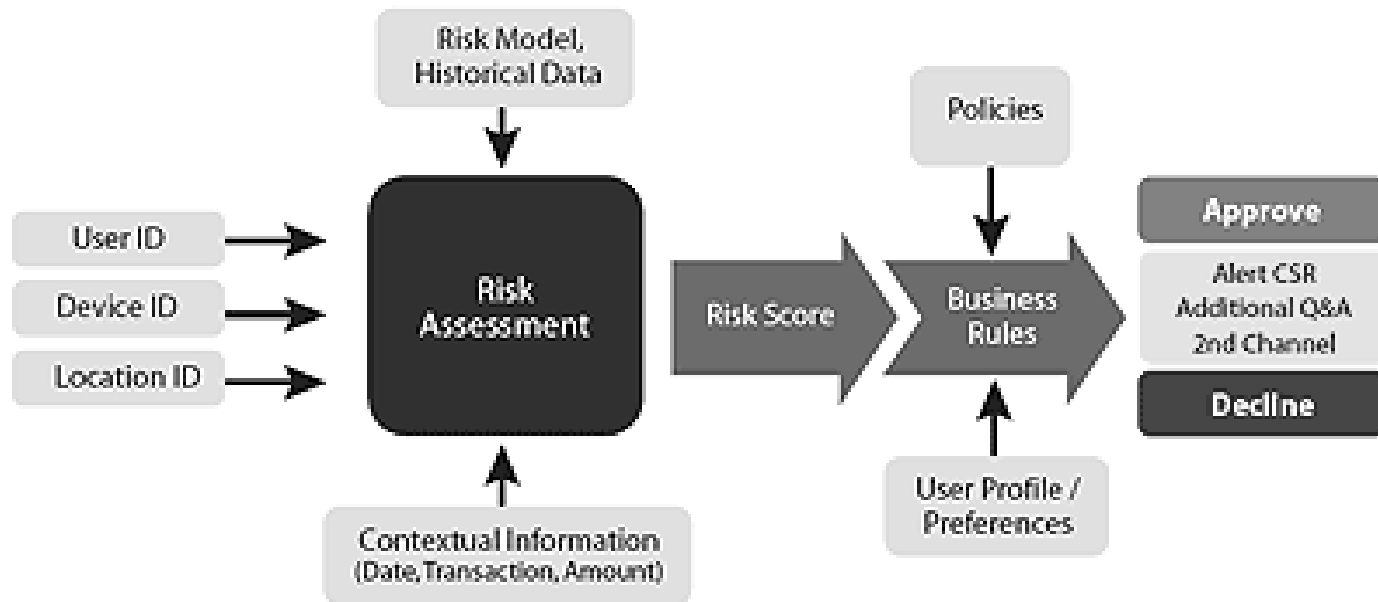


Context Aware Security



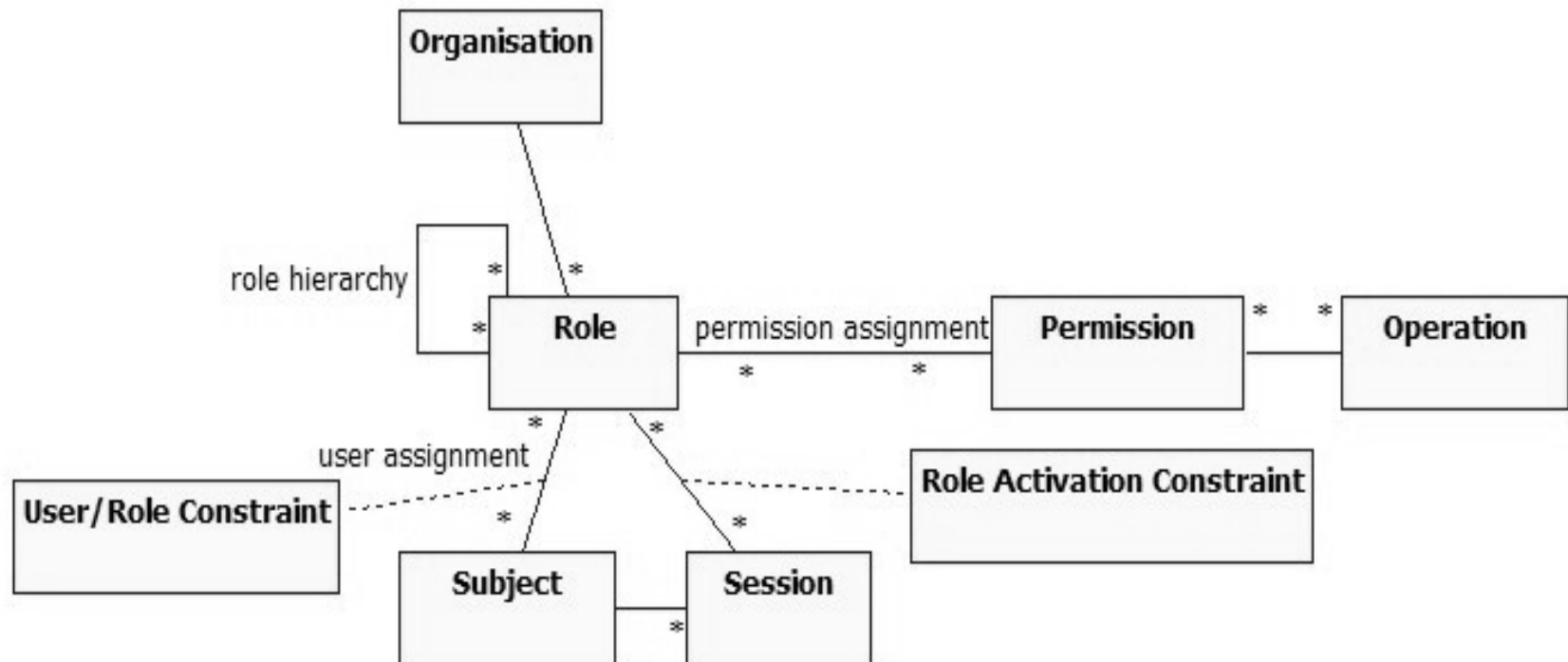
Context Aware Security

- Takes into account Risk (Risk Model, Risk Levels)
- Takes into account Mobility Context (location, device, etc.)
- Takes into account User Identity+ Profile/Preferences
- Takes into account Real-time Context (Date/Time, TranAmt, etc.)
- Takes into account Historical Context (Reputation)



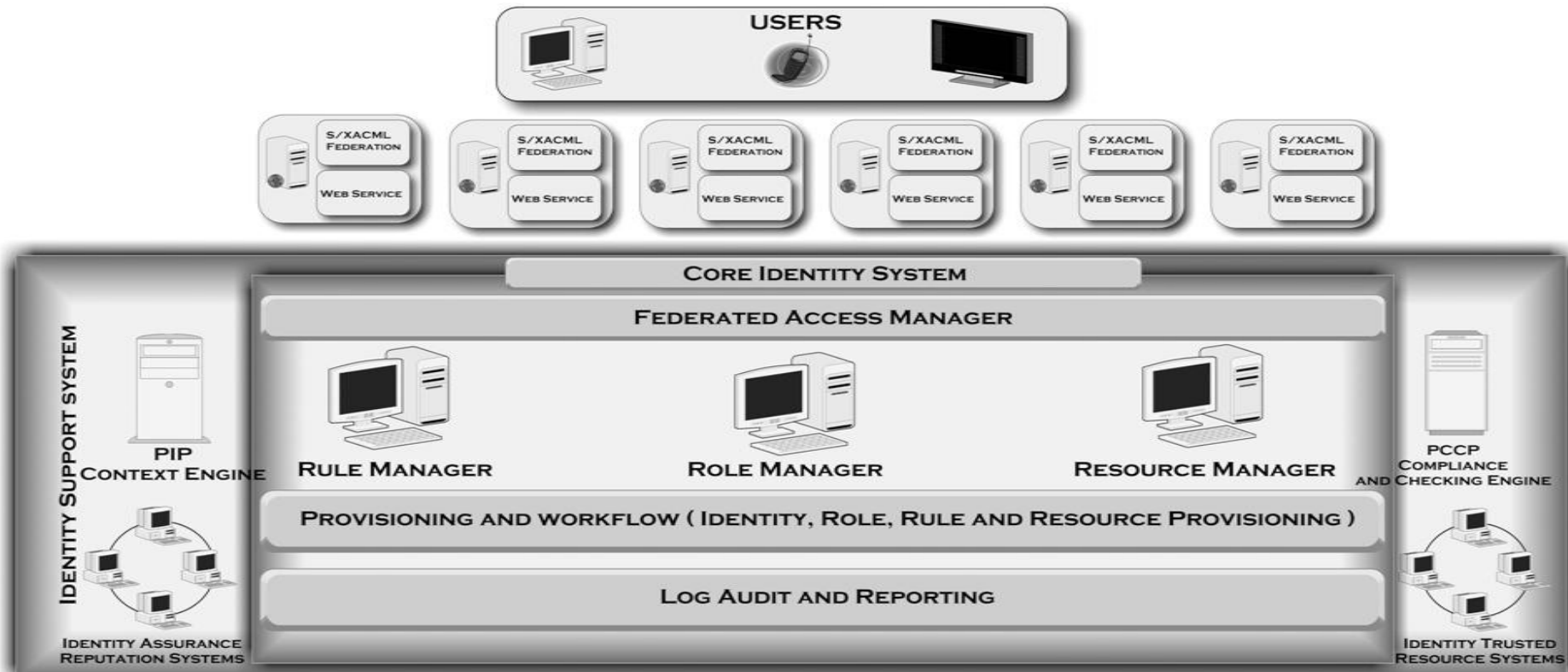
RBAC

- Role is an Identity Attribute (separation of duty)
- Role mining, discovery, mapping, hierarchy, etc. (full life cycle)
- Persona (social context), Role (business context)
- Role based Provisioning, Role based BP, Role based IT P
- Role to Rule to Resource



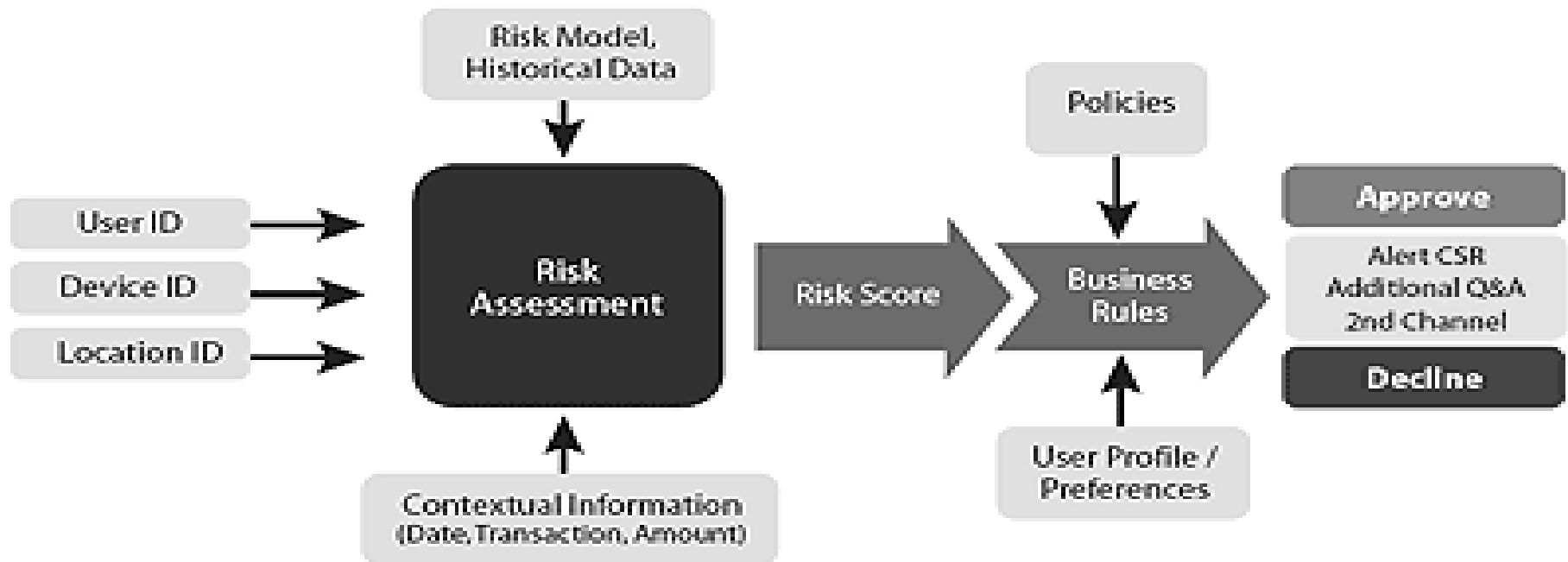
ResBAC

- Resource Classification, Categories and Compartmentalization
- Resource specific Rules (and life cycle management)
- Resources can be a Service (JEE/.NET, VM, OS, NE, Data, etc.)
- Labeling, Tagging, Resource specific Risks, Res based AuthN
- Resource Classification based Negotiations



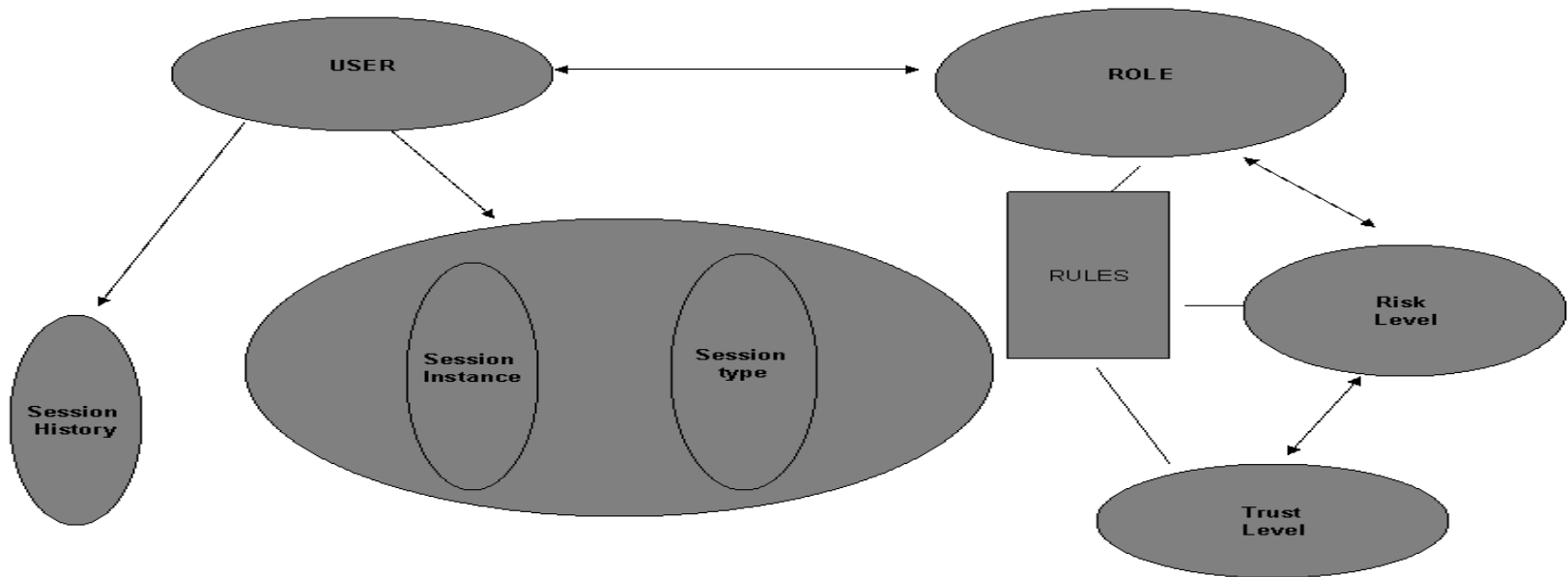
RiskBAC

- Risks at the Network Level (network threat levels)
- Risks at the User Level (Reputation)
- Device Risks (NAC, Client FW)
- Risks at the Transaction Level (value)
- Historical Risk Data



TrustBAC

- Trust based on TPM, TSS, TNC, etc. (Technical Trust)
- Trust based on Relationships (Business Trust)
- Trust Levels based on Resource Consumed & Risk Factor
- Highly Aligned to Resource and Risk Levels
- TrustBAC – Dr. James Joshi's work -leading edge

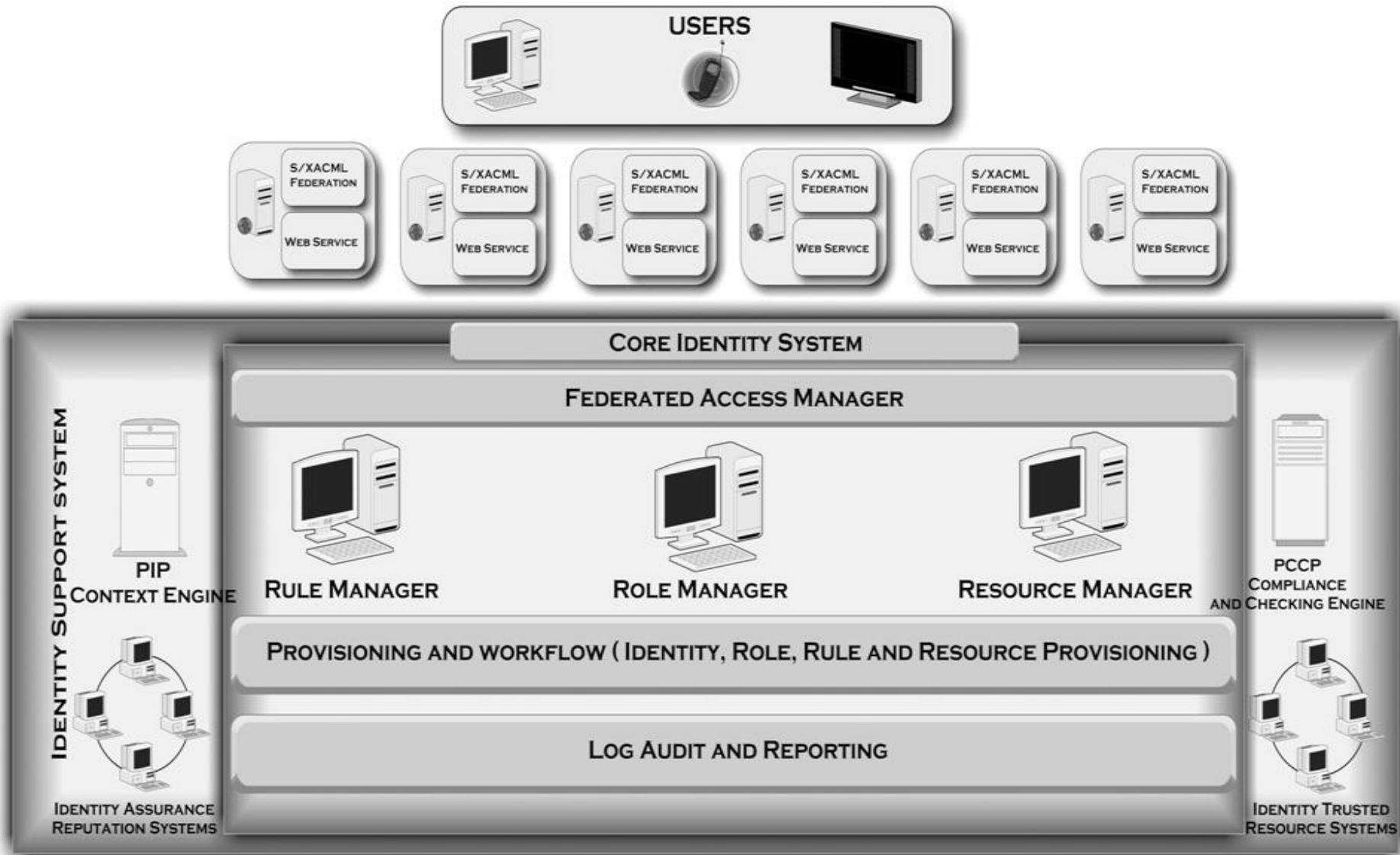


ABAC

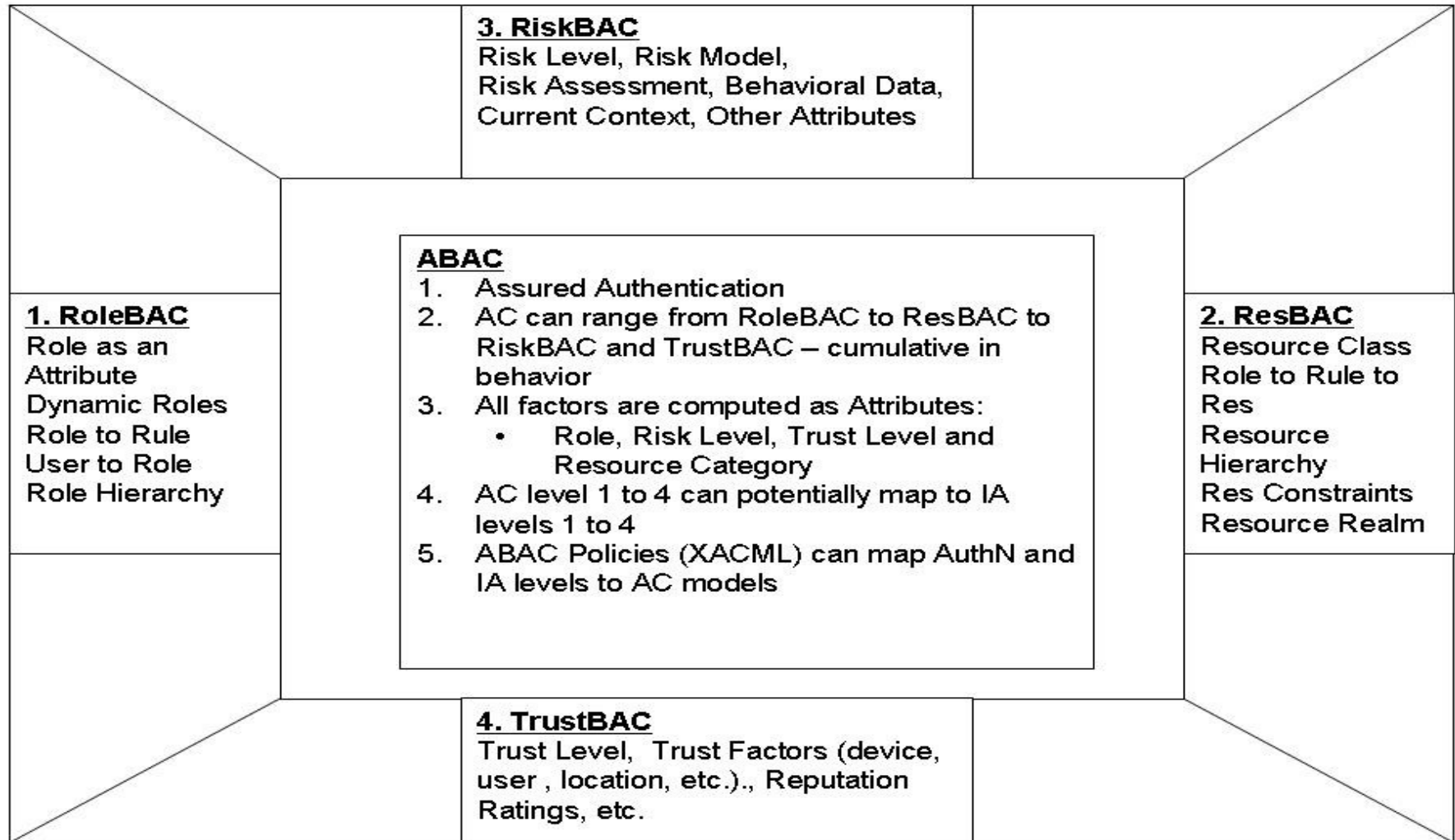
- Role and Persona as an Attribute
- Resource Classification and Category as an Attribute
- Assurance Level, Risk Level & Trust Level as an Attribute
- Device and Location data as Attributes
- Reputation Ratings as an Attribute



Alignment (Federated ID System)



Alignment (with Policies)



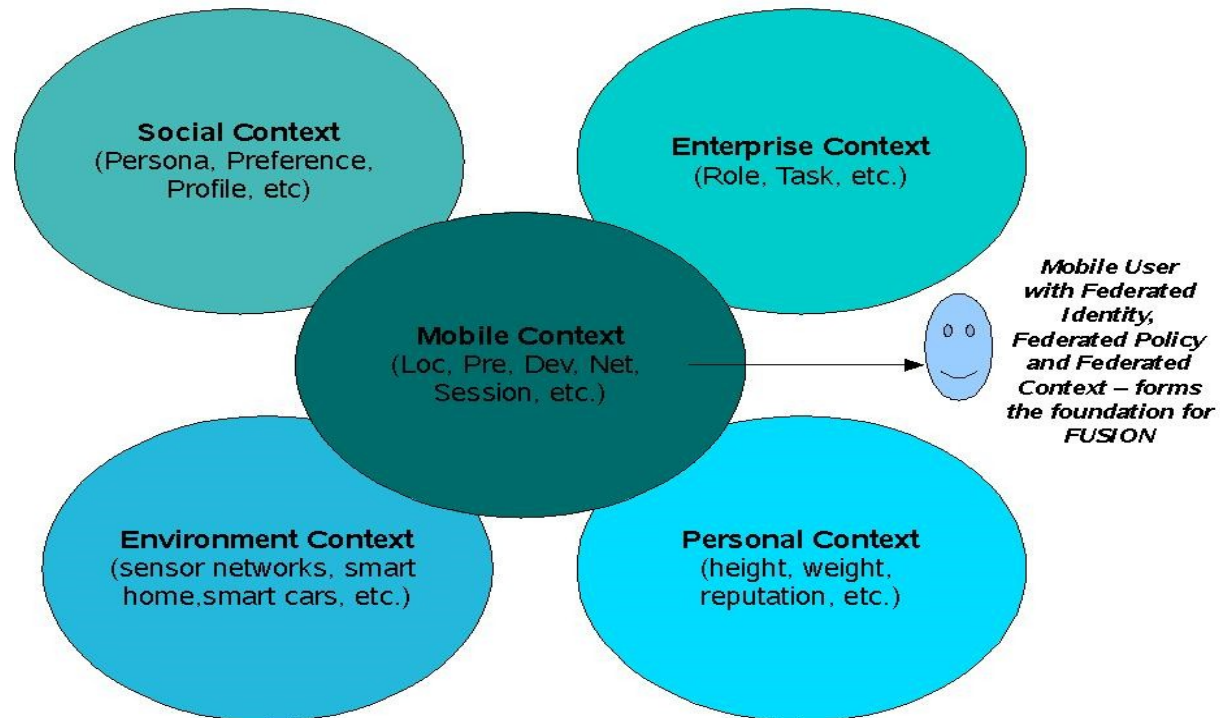
Alignment with Attributes/XACML

- Encapsulates RBAC, ACI, ACL, DAC, MAC, etc.
- Specialized PDP's (OS/VM, DLP, JEE/.NET, etc.)
- Attribute based Access Control implemented with XACML
- Policy Orchestration (Network facing to Service facing)
- Alignment of AC Model – Cumulative from RBAC/ResBAC + RiskBAC/TrustBAC



Relevance

- Relevant for Contextual COMPOSITION (SOA)
- Relevant for Convergence (Multi-media Broadband Networks)
- Relevant in Health care (Network of Networks)
- Relevant for eGOV and Emergency Services
- Federated Identity/Attributes as the Foundation for Federated Context



***In Defense of Role-based Access control
with Enhancements (RBAC PLUS)***

R. Chandramouli (Mouli)
mouli@nist.gov

ID Trust 2009

April 13,15 - 2009

NIST, Gaithersburg, MD, USA

Entitlements (or) Authorizations – Current Reality

1. Have to demonstrate that they meet certain compliance Requirements

2. Hence have to be policy driven
3. Inevitably end up being fine-grained & state-based

4. Fine-grained & state-based means that they are based on Attribute values that can change

- *Attributes of the User, Subject or Device*
- *Attributes of the Resource being accessed*

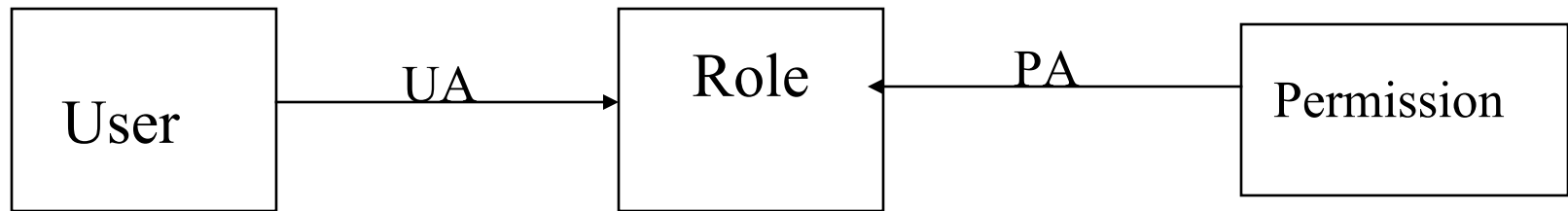
A New Perspective on Old Access Control Models

Access Control Model	Attribute & State
1. Access control Lists (ACLs)	<ul style="list-style-type: none">(a) Object-Centric.(b) Object Attribute - None.(c) User Attribute – Name and/or Group Membership (Static)
2. Protection Bits	<ul style="list-style-type: none">(a) Object-Centric(b) Object Attribute – None(c) User Attribute – Group Membership and/or Owner Status (Static)

A New Perspective on Old Access Control Models – Contd ..

Access Control Model	Attribute & State
3. Discretionary Access Control Model (DAC)	<ul style="list-style-type: none">(a) Object-Centric.(b) Object Attribute - None.(c) User Attribute – Owner, Grantee or Admin Status (Static)
4. Bell-LaPadula Model	<ul style="list-style-type: none">(a) Both Subject (User) & Object-Centric(b) Object Attribute – Sensitivity Level (Static)(c) User Attribute – Clearance Level (Static)

How Does RBAC Compare



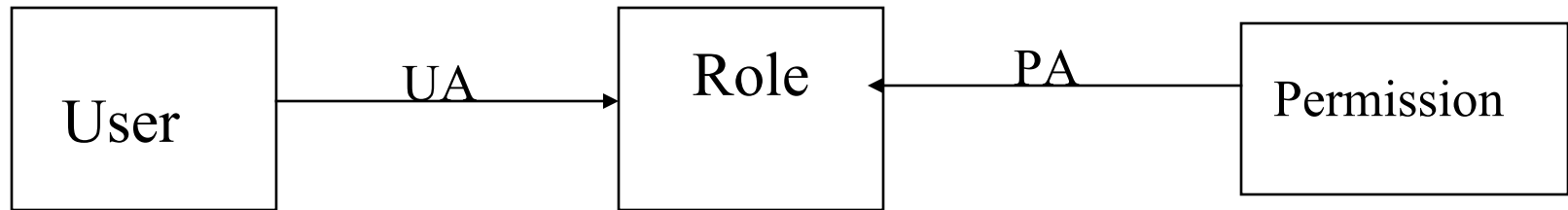
Entities – User, Role and Permission have no attached attributes
in ANSI Standard RBAC

Relations or Associations - UA and PA – Static Assignments

BUT

1. The Tight Coupling of User-Resource is removed by abstraction mechanisms such as *ROLES* – Grouping Permissions aligned to Business Process and *PERMISSION* – Object-Operation Pairs

How Does RBAC Compare – Contd ..



1. Parameterize the Entities - Attaching Attributes to User, Role and Permission
2. Make Associations UA and PA dynamic by consulting a rule-base (defined using XACML)

You have Fine-grained Dynamic Entitlements based on Attributes driven by Policies expressed using Rule Sets₆

How Does RBAC Compare – Contd ..

Entity & Attributes	Entitlements resulting from Policy Rule Instantiation
<u>Entity</u> – Role – Teller Attribute – Region = MD	Access Restricted to MD Customer Accounts
<u>Entity</u> – Objects underlying Permissions Attribute – Privacy Labels	Restricts Access only to Assigned Roles (Dynamic PA)
<u>Entity</u> – User Attribute – Location	Restrict or Deny Access to certain Resources/Objects
<u>Domain State Attributes</u> -Physician's Current Duty Station - Patients in Duty Station	Physician Role's entitlements restricted to EMRs of Patients in a particular Duty Station

Conclusion

- Enhancements to an Access Control Model that provides abstraction mechanisms such as RBAC can support fine-grained Policy Compliant Entitlements
- Still some Deployment Issues Remain – e.g., *Role Engineering* (in spite of tools for Role Mining etc)
- Enhancements such as Parameterized Roles do address issues such as *Role Proliferation or Role Explosion*.
- Support for Other Policies such as Least Privilege, Separation of Duty are well known.
- Landscape is not that *Gloomy* as there are reports of successful implementations of ROLES + RULES paradigm in some large Fortune 500 Corporations.

IDTrust 2009

Comparative Authorization Models

Tim Brown – VP Chief Architect Security



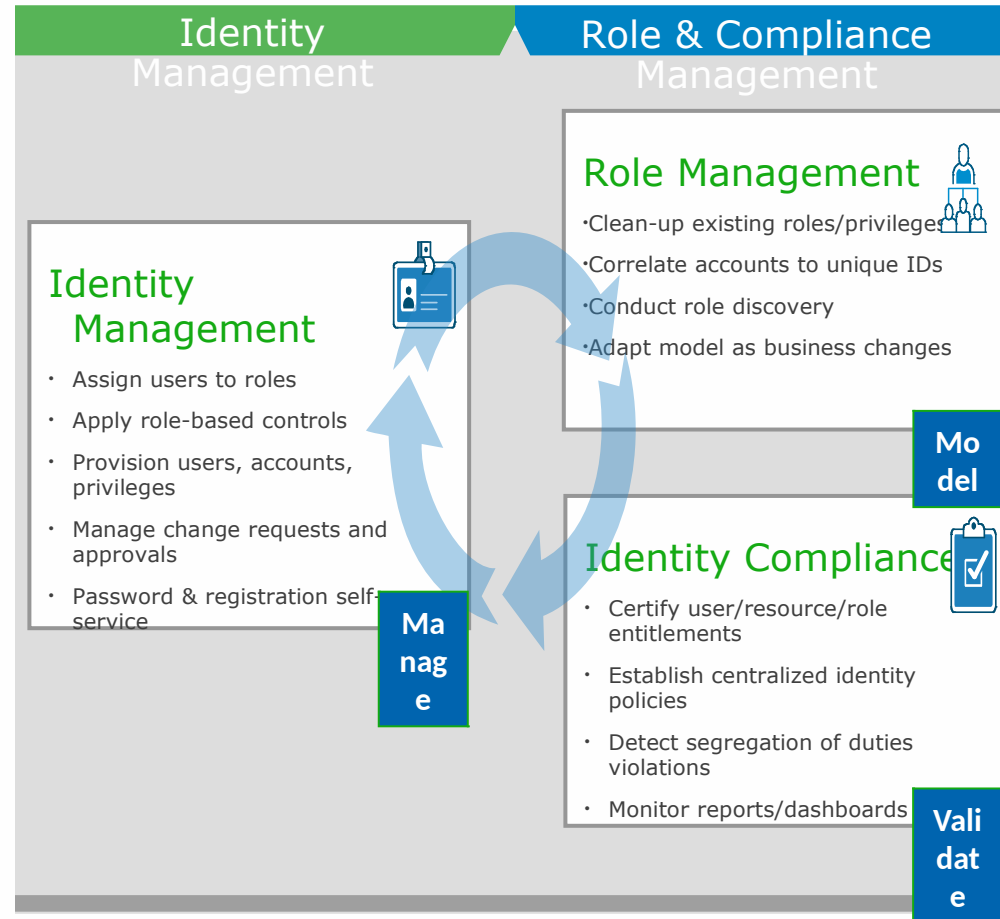
How do we get the Authorization issue under control

Implement an Identity Lifecycle Management process

Move towards business Roles

Implement sound authorization processes

Move towards new Authorization models



Move towards Business Roles and delegated administration model

- > **Move towards Business Roles**
 - Enriches identity-related business processes with real-time analytics
 - Simplifies user experience
 - Improves quality of access rights
 - Preventative controls
- > **Delegate Role creation and Authorization roles to appropriate party**
 - Suggesting roles
 - Identifying policy violations during provisioning
 - Highlighting out-of-pattern entitlements



Smarter with Suggest Roles Button

- > Attribute/Claims Based Identity services will help
 - More flexibility
 - More user Control
 - Additional weighting and risk management
- > But will come at a cost
 - New model with more flexibility
 - Authorization model needs to be redefined
 - Business models need to adapt to new models
- > Few Relying Parties accepting Claims
- > Few trusted Identity Providers none willing to accept liability

Defensive PKI

(What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

The Problem

- Dec 2008: Exploitation of MD5 collisions
- May 2008: Debian (OpenSSL) RNG error
- What's next?
 - Hash 2nd pre-image attack?
 - Dead key length?
 - Dead PK algorithm?
- The infrastructure is “too big to fail”
- What do we do about it?

Defensive PKI

(What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

Microsoft's Response Options

1. Remove affected root CA certificates (nuclear option #1)
2. Disable MD5 in certificate validation (nuclear option #2)
 - Also break CAs that use non-random serial numbers
3. Work with affected CAs to update their infrastructure before attack is practical to others
 - Trust Sotirov and team, and the security of their PS3 cluster

Constraints

- Cannot break millions of users (without sufficient justification)
 - Certificate error UX in latest browsers is very effective in stopping users
 - Previously issued certificates were not vulnerable
 - Did not affect all CAs that use MD5
 - Many long lived subordinate CA certificates use MD5

How Microsoft responded to the MD5 collision attack

- Microsoft immediately contacted affected CAs to assess situation
 - CAs were cooperative, but needed more time than usual because attack was announced over the holidays
 - Quick engineering fix, but long QA cycle
- Asked all CAs in our “Root CA Program” to provide information on crypto algorithms in use
 - 52 out of 80 CAs responded. The rest needed more time to gather information
 - Request includes the CA’s use of 1024 bit RSA and MD5 in their hierarchy
 - Most newer CAs have issued with SHA1 only
 - Most have already switched to SHA1

Some Observations...

- Revocation was not designed to handle pre-image attacks against hash algorithms
 - Old expired certificates are vulnerable
 - Rogue certs will not contain a CDP
 - CAs revoke by specifying a serial number, but serial number could be changed
 - Path validation code cannot require CDP since CDP is not present in many intermediate CA certs
- Subordinate CA may be signing new certificates with SHA-1, but its own certificate may have been issued a lot time ago and still uses MD5
 - Reissuing a subordinate CA certificate may trigger audit
 - Distribution of new root and subordinate CA certificates is very difficult and time consuming
 - Not scalable for MS to distribute sub-CA certs through Windows Update
- From the experience with the Debian bug, replacing certs would be a very slow process. <http://www.eweek.com/c/a/Security/CAs-Not-Getting-Big-Response-to-Debian-Encryption-Flaw/>

Defensive PKI

(What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

The Minor Issues

- Usability that degrades trust
 - Should I accept the NIST certificate?
 - Now, where did I write down the password protecting my private key?
- Fragility
 - Oops, my root CA certificate expired...
 - My applications were blocked because their server certificates expired
- Our business partners assumed that authentication == authorization
- How do I upgrade the hash algorithms (as opposed to how do I get a good one – see Majors)

Defensive PKI

(What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

Defense Begins at *Home*

- Relying Parties have ultimate responsibility to ensure a certificate is acceptable
- Acceptability decisions might be based on
 - Policy
 - Certificate status
 - Trust anchor
- But these tools are not enough!
 - Lifecycle issues, crypto issues not adequately addressed

Cryptographic Lifecycle

- Cryptographic Migration is part of the Lifecycle of a system, but is always an afterthought in the implementation
 - This is not unique to PKIs! Think about DES...
- Migration timelines may vary by application
 - E.g., NIST SP 800-78-1 requires use of
 - 2048 bit RSA for signatures after 12/31/2010
 - 2048 bit RSA for authentication after 12/31/2013

Overreliance on Policy Leaves Relying Party at Risk

- Relying parties depend on policies (implicitly or explicitly) to ensure that key sizes and hash algorithms provide acceptable levels of security
 - This is not agile, and may be inexact on details that matter to your application!
 - policy mapping can be more abstract, ignoring small discontinuities
 - To increase security and agility, relying parties need crypto based acceptance controls
 - Algorithm, key length, parameters

Defensive PKI

(What happens when PKI fails?)

Kelvin Yiu

Steve Whitlock

Tim Polk

Carl Ellison

It's a fault tolerance problem

- We know how to do fault tolerance.
 - Keep running in spite of failures!
- The failures we need to address are:
 - Bad specific key(s)
 - Bad key length
 - Bad algorithm
- Revocation
 - Flawed
 - Not fault tolerant

Straw-man Solutions

- Enroll not for 1 certificate but for a binding – and get multiple certificates, with different algorithms and keys, as they come available, during the lifetime of the binding.
- Get not one timestamp but a living sequence of timestamps, each with a newer, better algorithm or key (and sacrifice blindness).
- Fix revocation
 - CDP today in the attacked certificate
 - Revoke keys, algorithms, key lengths; not just certs
 - We need to choose authorities and channels for those

Q & A

Usable Secure Mailing Lists with Untrusted Servers

Rakesh Bobba, Joe Muggli, Meenal Pant, Jim Basney and Himanshu Khurana
University of Illinois, Urbana-Champaign
{rbobba, jmuggli, mpant, jbasney}@ncsa.uiuc.edu, hkhourana@iti.uiuc.edu

ABSTRACT

Mailing lists are a natural technology for supporting messaging in multi-party, cross-domain collaborative tasks. However, whenever sensitive information is exchanged on such lists, security becomes crucial. We have earlier developed a prototype secure mailing list solution called SELS (Secure Email List Services) based on proxy encryption techniques [20], which enables the transformation of cipher-text from one key to another without revealing the plain-text. Emails exchanged using SELS are ensured confidentiality, integrity, and authentication. This includes ensuring their confidentiality while in transit at the list server; a functionality that is uniquely supported by SELS through proxy re-encryption. In this work we describe our efforts in studying and enhancing the usability of the software system and our experiences in supporting a production environment that currently is used by more than 50 users in 11 organizations. As evidence of its deployability, SELS is compatible with common email clients including Outlook, Thunderbird, Mac Mail, Emacs, and Mutt. As evidence of its usability, the software is being used by several national and international incident response teams.

Categories and Subject Descriptors

H.4.3 [Information Systems Applications]: Communications Applications—*Electronic Mail*; H.5.2 [Information Interfaces and Presentation]: User Interfaces—*Evaluation/methodology*; E.3 [Data Encryption]

General Terms

Design, Security, Human Factors

Keywords

E-mail List Security, Proxy Re-encryption, Usability study

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust '09 Gaithersburg, MD

Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

1. INTRODUCTION

As more and more user communities are engaging in collaborative tasks, use of Email List Services (or simply *Mailing Lists* - MLs) is becoming common; i.e., emails exchanged with the help of a list server (examples of commonly used list server software include Mailman and Majordomo). Many tasks where MLs are used require exchange of private information, especially those that involve messaging in collaborations across multiple administrative domains. For example, a ML of security administrators that manage critical infrastructure would not want their emails disclosed to hackers. Specific instances include the multi-domain Open Science Grid (<http://www.opensciencegrid.org/>) and TeraGrid (<http://security.teragrid.org/>) systems where the Incident Handling and Response policies recommend the use of encrypted and signed mailing lists. In general, use of encrypted and signed lists is recommended for incident response by IETF [5] and CERT [31]. Additional examples include a list of (1) health care and pharmaceutical researchers who want to protect patient privacy, (2) corporate executives who want to protect proprietary information, and (3) researchers engaged in collaborative research involving multiple university, government and industry institutions who want to protect their intellectual property.

For such MLs cryptographic solutions are needed that provide adequate protection (i.e., confidentiality, integrity, and authentication) for the private content from threats at the client side, at the network paths where the emails are in transit, and at the server side where the emails are processed for distribution to the list. That is, there is a need to develop Secure Mailing Lists (SMLs) as illustrated in Figure 1. Threats to the server side are an important concern in practice and lack of good solutions today has forced users to develop their own clunky ones. For example, several critical infrastructure security protection groups today use out-of-band means to distribute passwords to members and require members to use password-based-encryption (supported by commonly available GnuPG plug-ins) so that the list server does not have access to email plain-text, minimizing the trust that must be placed in the mail server.

To address this challenge for mailing lists we have earlier developed a prototype SELS (Secure Email List Service) [20]. The SELS protocol and software prototype satisfy requirements in the categories of security properties (i.e., confidentiality, integrity, and authentication), infrastructure compatibility, key management and performance. For confidentiality SELS uses proxy encryption techniques that allow the list server to transform email cipher-text between list

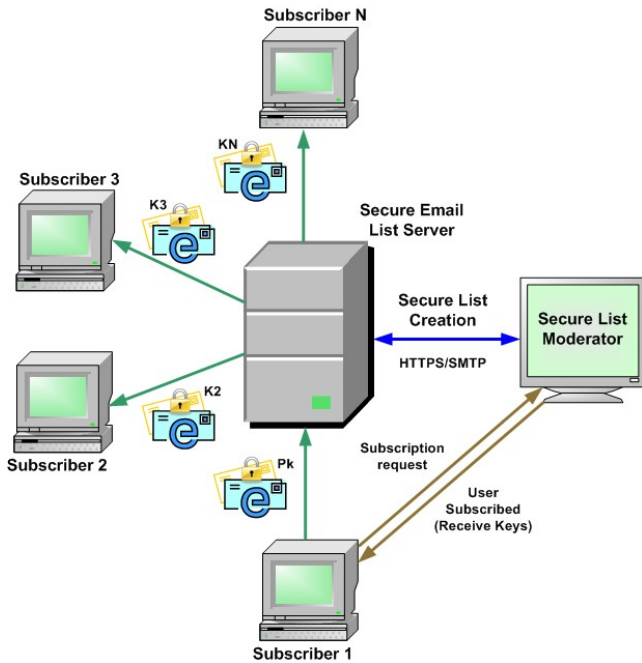


Figure 1: Secure Mailing Lists

members without gaining access to the plain-text. Proxy encryption techniques have been studied for over a decade [1, 2, 7, 15, 17, 18, 22, 35] and have been used to design several applications including simplification of key distribution [2], key escrow [17], file sharing [1], security in publish/subscribe systems [19] and multicast encryption [8]. SELS builds on COTS components to maximize ease of deployment. In particular, we were able to use the OpenPGP message format [6] and standard GnuPG plug-ins at the client side which facilitated deployment by eliminating the need for developing and distributing email-client specific plug-ins. We have implemented the protocol and the system using the Mailman list server, GnuPG and BouncyCastle cryptographic libraries, and standard GnuPG plug-ins and APIs. SELS is viable in enterprise settings, has compatibility with Microsoft Outlook, Emacs, Mac Mail, Mutt and Thunderbird, and has performance that scales to support enterprise mail servers that process hundreds of thousands of emails per day.

In this work we focus on usability and deployment experiences with SELS. We conducted a usability study whose high-level goals were to evaluate and enhance the usability (i.e., effectiveness, efficiency, and satisfaction) of the SELS key management system for list subscribers with the strong preference that the solution be compatible with commonly used email clients. Given this preference we further refined the goals as follows. First, to identify SELS key management tasks that pose usability challenges across several common email clients. Second, to determine the ability of users to complete key management tasks, how much time they take in doing so, and how satisfied they are with the software. Third, to assess how usability is influenced by familiarity with specific email clients, underlying security tools and learnability. Fourth, to determine if SELS introduces additional vulnerabilities for its users. To address these goals we combined several usability techniques in executing this

study. The study allowed us to evaluate and enhance the usability of SELS key management by (1) exploring two alternate key trust establishment techniques, (2) developing an effective password management solution that balances usability and security, and (3) identifying suitable interface cues that can be introduced to mitigate remaining vulnerabilities in environments where new software or plug-ins can be deployed.

A fully functional version of the software has been packaged and released for community use.¹ We are actively supporting the software via a public email list and making new releases to fix bugs and add features. Our first user community is incident responders and several national and international incident response teams have adopted SELS for email exchange. Currently, these include the TeraGrid Security Working Group and the International Grid Trust Federation (<http://www.igtf.net/>). We report on our experiences in supporting the TeraGrid Security WG for a period of ten months. We present SELS usage statistics, discuss usability and security issues observed in practice and present software engineering enhancements. Success of SELS is clearly indicated by the fact that there has been a nearly four-fold increase in the number of encrypted emails exchanged by the TeraGrid users since they adopted SELS, with anecdotal evidence suggesting that this increase is largely due to better usability provided by SELS.

The rest of this paper is organized as follows. In Section 2 we give an overview of the SELS protocol and software prototype. In Section 3 we evaluate the usability of SELS. In Section 4 we describe the SELS production environment and our experiences with supporting the TG Security WG. In Section 5 we discuss related work and conclude in Section 6.

2. SELS OVERVIEW

In this section we provide an overview of the SELS system architecture focusing more on the list subscriber interaction. We refer the reader to [20] for further details. SELS provides confidentiality, integrity and authentication for emails exchanged in a mailing list via public key encryption and signing by interactions among the following entities.

- *List Moderator (LM)*. *LM* is a user (or process) that creates a list to be maintained at the list server, authenticates users, and helps them subscribe to and unsubscribe from the list.
- *List Server (LS)*. *LS* creates lists, maintains membership information (email addresses and key material), adds and removes subscribers based on information received from *LM*, and forwards emails sent by a valid list subscriber to all current subscribers of that list.
- *Users/Subscribers*. Users subscribe to lists by sending join requests to *LM* and send emails to the list with the help of *LS*.

The first major goal of SELS is to minimize trust in *LS* such that *LS* is unable to access email contents while still providing the necessary list management and email forwarding capabilities. As mentioned in the Introduction, threats to *LS* are an important concern. First, compromise of a

¹Available at <http://sels.ncsa.uiuc.edu>

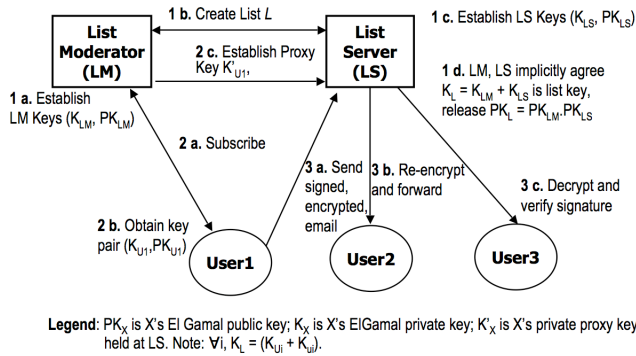


Figure 2: SELS Protocol Overview

trusted list server by an attacker could result in significant disclosure of sensitive information. Second, for lists with members from multiple organizations, it is difficult to trust a single organization to host the list server. Traditional solutions that provide server-side protection tend to have a high overhead for key management. For example, the common practice of out-of-band password distribution for password-based encryption requires users to remember a long list of passwords to decrypt previously sent emails. Instead, SELS achieves this via proxy encryption where LS re-encrypts messages between list subscribers by using proxy private keys but without requiring access to the plain-text.

A high-level view of the three-step SELS protocol is presented in Figure 2. In the first step LM and LS create a list L , which involves them establishing an ElGamal key pair each for the list and distributing the list public key computed to be the product of their public keys. Note that no single entity has access to the list private key, K_L . In the second step, users subscribe to list L by contacting LM. LM creates a key pair for each user and sends it to that user. In addition, LM sends keying material to LS, which allows LS to establish the private proxy key for that user. These keys are all computed from LM and LS's list keys by simple addition and subtraction of random numbers to ensure that the sum of each user's private and private proxy keys is the list private key, K_L , originally established by LM and LS. This invariant allows for the proxy re-encryption process to execute correctly. In the third step, users send email encrypted with the list public key, PK_L , and optionally signed by their own private keys. LM executes proxy re-encryption on these emails once for each subscriber and sends the output to that subscriber. Each subscriber can then decrypt the message with their private key.

The second major goal of SELS is to drive the design and development process with deployability and adoption in mind. To that end SELS (1) uses the OpenPGP message format [6] and standard GnuPG plug-ins (<http://gnupg.org>) on the client side allowing users to use their existing email clients and (2) uses open-source off-the-shelf components such as the Mailman list server. By using COTS GnuPG components as illustrated in Figure 3 on the client side SELS becomes compatible with any email client for which a GnuPG plug-in is available and that includes popular email clients such as Outlook, Thunderbird, Mac Mail, Emacs and Mutt. We believe this compatibility to be a crucial factor for successful deployment of SELS and this belief has been supported by experiences working with real user communi-

ties. The developed components for LM and LS use open-source GnuPG and Bouncycastle libraries as well as GnuPG key management functions.

In the world of secure email, OpenPGP and S/MIME are competing standards with native support of S/MIME in email clients being more prevalent. The proxy encryption technique used in SELS requires transformation of text between two public keys, which, in turn, requires that the keys share common parameters. The chosen cryptosystem, ElGamal, easily supports this as it is a group based cryptosystem. However, with the RSA cryptosystem this would imply that the modulus be shared between multiple key-pairs, which is inherently insecure. S/MIME supports RSA but not ElGamal (specifically for message encryption), limiting SELS compatibility to OpenPGP for now. Recently, Elliptic Curve Cryptography (ECC) has been added to S/MIME standards and is also being supported by some email clients. ECC is also a group based cryptosystem, therefore, it will enable the proxy encryption technique to be employed. In the future we plan to support ECC and, in turn, achieve compatibility with S/MIME for broader adoption of SELS.

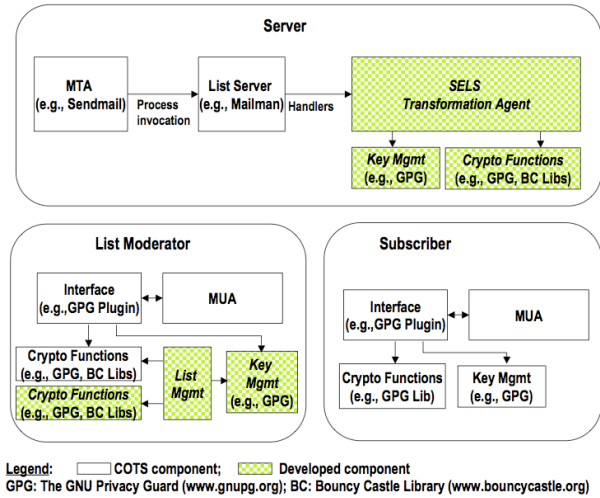


Figure 3: Component Architecture

We conducted a performance evaluation of SELS and focused on the most expensive operation – the proxy transformation step at the list server. Consequently, we measured throughput of the list server with varying list sizes and message sizes. Overall, we saw that even the worst throughput of 2.5 messages/sec for SELS with list size 10 and message size 100KB corresponds to a throughput of more than 200,000 messages per day. Since most mail servers in small and medium-sized organizations do not typically process more than 100,000 messages per day (of which only a subset are ML messages) we conclude that adding security to MLs will not impose an undue overhead on the mail servers.

3. USABILITY EVALUATION

3.1 Approach

The high-level goals of the usability study were to evaluate and enhance the usability (i.e., effectiveness, efficiency, and satisfaction) of the SELS key management system for list

subscribers with the constraint that the solution be compatible with existing deployed email clients. To address these goals we combined several usability techniques in executing this study. First, the groupware walkthrough [27] technique (which is based on cognitive walkthrough) was used to determine relevant key management tasks and candidate usable solutions. Application of the technique suggests two possible solutions for key management with a common password management approach. The two key management approaches utilize different GnuPG key trust establishment mechanisms, namely, key signing and key trust assignment.

We then undertook two rounds of focused user studies to design and evaluate the key installation and password management technique as well as the overall usability of SELS. The first round explored the GnuPG key signing approach of trusting keys while the second round explored the key trust assignment approach. In each round we assessed the users' ability to successfully complete specified key management tasks, we measured the time they took to do so, and then they filled out the SUS questionnaire [4] to convey their satisfaction. Among the specific tasks were a set of vulnerability assessment tasks whereby users were required to place trust in messages that may or may not be correct. Since we are evaluating security software it is important to ensure that the system does not introduce new vulnerabilities for the users.

These focused studies were undertaken with a relatively high expert-to-novice ratio [10, 23] with novices being defined as those with some basic security concepts and experts being defined as those familiar with GnuPG tools and advanced security concepts. Keeping the intended users in mind (e.g., our current TeraGrid users) as well as the known limitations of common email clients in their ability to provide usable security [13, 32], we decided that all subjects must have at least a basic grasp of security concepts. This allowed us to understand how familiarity with tools and concepts, that novices would learn over time, would affect SELS usability and security. To further help in obtaining this understanding we explicitly asked users to perform basic GnuPG two-party secure email tasks in addition to SELS tasks.

One deviation of our work from most studies of usable security is that we asked the subjects to complete tasks in the context of secure mailing lists but without a particular scenario. For usability studies of security systems these scenarios are often used to motivate the need for security. However, in our case since all users had some background in computer security we felt that a scenario to motivate the need for security may not be very helpful. Instead, we asked the subjects to focus exclusively on interface cues and their knowledge of security concepts for making security decisions. An advantage of this is that the results of the study depend only on the usability of the system and the users' knowledge of security concepts and not on the extent to which the scenario motivated security.

3.2 Groupware Walkthrough

In the early part of the SELS design and implementation process the authors undertook a groupware walkthrough with the goals of identifying key management tasks and problems as well as candidate solutions. The groupware walkthrough process involves specifying a description of tasks and teamwork and then using the technology to walk through

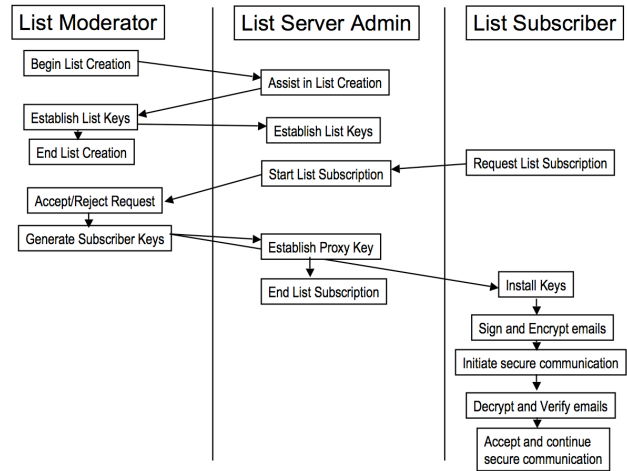


Figure 4: Tasks Identified by Groupware Walkthrough

the interfaces from the point of the team users in attempting to accomplish the tasks [27]. We defined these tasks and teamwork based on the SELS protocol described earlier and executed the walkthrough with the authors acting as multiple evaluators working simultaneously and recording results with detailed notes. Figure 4 identifies the tasks and required teamwork but details of subtasks have been omitted for simplicity. Based on interviews with several system administrators and security professionals we chose the following four email clients with available GnuPG plug-ins for the walkthrough: (1) Mac Mail, (2) Mutt, (3) Outlook, and (4) Thunderbird. The walkthrough identified the following three issues.

1. **Installation of multiple keys.** The List Moderator provides each subscriber with 1) a public key, common to all subscribers of the list, that is used to encrypt messages to the list and 2) a public-private key pair, unique to the subscriber, that is used to decrypt messages received on the list. The user must then add these keys to her GnuPG key ring and assign a password to the private key. She must also place appropriate trust in these keys to be able to use them for sending and receiving secure e-mails as per the GnuPG trust model. In addition the user also obtains a list of public keys belonging to list subscribers from the list moderator for the purpose of signature verification and she must similarly install and trust these keys. A major difference with standard GnuPG use is that SELS requires users to import and install private keys while in standard GnuPG users typically only deal with public keys. In addition to this new concept, we observed that users would have to install several keys so unless the key installation steps were simple the user might get frustrated.
2. **Managing and using multiple keys.** Whenever a message is received on any list the user must remember the password corresponding to that list and enter it in order to decrypt the message. All email clients studied provide passphrase caching capabilities but limit the caching to *only one* passphrase, which further compli-

cates management and use of the multiple private keys. We note that these issues of installing and managing multiple keys arises primarily because SELS uses an untrusted server. If the server were fully trusted, then the subscriber would be able to use the same key pair for all lists.

3. **Prior GnuPG experience.** While conducting the walkthrough it became clear that prior experience of secure email use with GnuPG would greatly benefit the users; however, it was not clear 1) how much prior experience would benefit the users and 2) whether lack of such experience would make the software unusable.

Analysis. The results from our walkthrough provided an opportunity to fix the usability problems early on and to design a focused user study to evaluate the usability and security of the system. For installing keys we designed a sequence of three emails sent from the list moderator with the following contents: (1) the list moderator’s public key that the user needs to import and trust, (2) the list public key as well as the user’s private key (for the list) that the user needs to import and trust, and (3) a set of subscriber keys that the user needs to import. GnuPG allows two ways of achieving trust: signing keys or using the GnuPG trust model. We decided to study both ways of achieving trust with the initial assumption and default implementation that the first approach will work because prior usability studies in secure email indicated that users find it very challenging to deal with the GnuPG trust model [12, 28, 32]. For managing and using multiple keys we proposed a simple approach, namely, recommend that users use the same passphrase to protect all private keys and use the passphrase caching tools to manage that passphrase. Clearly, this is a tradeoff between usability and security but we believe that as long as the users use a strong password their security is only minimally weakened by this approach. To evaluate these approaches we needed a user study that asked users to evaluate SELS with the suggested approaches for installing and managing multiple keys. To characterize how familiarity with the GnuPG interface (or lack thereof) affects the usability of SELS 1) we used both expert and novice users, i.e., those who were familiar with GnuPG and those who were not, and 2) we included tasks involving Two-Party Secure Email using GnuPG, hereafter referred to as TPSE, in our focused user study.

3.3 Focused User Study

To conduct the user study we identified a set of testing goals, recruited users, setup the study in a laboratory, conducted the study with one user at a time, and took extensive notes.²

3.3.1 Testing Goals

To measure the effectiveness of the interface, each user was asked to perform these tasks: (1) install and trust the key of another user in case of TPSE, (2) install and trust keys related to the list in case of SELS, (3) exchange signed and encrypted emails and (4) manage multiple keys in case of SELS.

²All of these steps were approved by the Institutional Review Board and signed consent forms were obtained from each user up front.

To evaluate the vulnerability of the interface, each user was asked to deal with malformed emails and use the security cues provided by the interface to decide whether or not to trust the email. Specifically, the user dealt with attacks where the emails were incorrectly encrypted, signed or both. Intuitively, (1) a correctly encrypted email should provide assurance that the message was intended for the recipient because only the recipient has the corresponding private key and (2) a correctly signed email should provide assurance that the sender actually generated the message because only she has the required signing key.

Measuring the success rate and time taken for these exercises allows us to evaluate the usability and security of TPSE and SELS to a reasonable objective extent. However, users often have subjective views and insights into usability of software. We use the common technique of usability questionnaires to study these subjective evaluations. Specifically, we used the System Usability Scale (SUS) [4], which has proven to be a good guide for gauging the usability aspects of effectiveness, efficiency and satisfaction. SUS is designed to give a quick assessment of overall usability and consists of ten questions rated on the Likert scale. SUS has proven effective in practical settings [29] and is also beginning to be used in usability studies of secure software (e.g., Polaris [9]).

We conducted two studies as discussed earlier. In **Study I** we evaluated the key signing approach and in **Study II** we used the GnuPG trust assignment approach. All the goals of Study I and Study II were identical to allow for an effective evaluation of proposed approaches.

3.3.2 Recruiting Users

Our process of recruiting users was guided by the desire to (1) get representative users for the system’s target audience and (2) be able to study the impact of familiarity with a similar interface on SELS usability. For studying the impact of familiarity we needed to categorize our users into experts and novices as follows:

- **Expert.** A user who uses GnuPG to send and receive secure emails at least occasionally *and* has security-related knowledge experience from one or more of the following: classroom, job, personal interest.
- **Novice.** A user who has used GnuPG at most rarely but has security-related knowledge experience from one or more of the following: classroom, job, personal interest. This knowledge includes basic understanding of concepts of confidentiality, integrity and authentication as well as how these concepts are enabled by public key cryptography.

To recruit users we sent out flyers outlining the study and asking for volunteers. Based on the responses we selected 20 users for the study. Among the users there were two system administrators, seven computer science graduate students, and eleven professional engineers.

3.3.3 Setup and Sequence

The two studies were conducted in a computer lab and administered by two people, a *studyadmin* who administered the study remotely via email and a *studymonitor* who observed the users during the study and took notes. Users were asked to choose one of the following four email clients for the

study: 1) Thunderbird, 2) Outlook, 3) Mac Mail or 4) Mutt. A GnuPG plug-in was installed on each of these clients and a key-pair was pre-generated for the users. A standard PC with Debian Linux was used for Mutt and Thunderbird and a standard PC with Windows XP was used for Outlook and Thunderbird. A Macbook Pro was used for Mac Mail.

The study was divided into five parts as described below. Part four differed in Study I and Study II. These differences are noted below.

Part I: Background Questions.

User provides background information. This information helps us classify him/her as an expert or novice.

Part II: TPSE Effectiveness.

Studyadmin sends email containing keying material and instructions to the user. The user is asked to 1) import a public-key and place trust in it by signing it, 2) send a signed and encrypted email using the imported key, and 3) decrypt and verify a received email. After completing these tasks the user was asked to fill a SUS questionnaire about the experience of exchanging two-party secure email using GnuPG (TPSE). The user has an option to provide additional feedback if he/she desires.

Part III: TPSE Vulnerability.

The user received six email messages and was asked to decide whether he/she trusted the message based on the security cues provided by the email client's GnuPG plug-in and their general knowledge of security. He/she was asked to forward the message to "studyadmin" with their trust decision (yes or no) and optionally an explanation for their decision. The five message types used are described in Table 1.

Part IV: SELS Effectiveness.

The user was subscribed to a mailing list hosted by the SELS server and received the email messages with the list keys as described previously. The user was asked to 1) import keying material for a list sent by the list moderator, 2) send signed and encrypted email to the list, and 3) decrypt and verify an email received on the list. Importing keying-material for a SELS list involved 1) importing the list moderator's GnuPG public-key and placing trust in it by signing it, 2) importing an encryption (decryption) GnuPG key-pair to be used for encrypting (decrypting) messages for (on) the list and placing trust in it by signing it, and 3) importing the GnuPG public-keys of all members of the list. Additionally we also recommended that users set the passphrase of the imported key-pair for the list to be same as that of their GnuPG key-pair for ease of use. After completing these tasks, the user was asked to fill a SUS questionnaire about the experience of using SELS for exchanging signed and encrypted email using a list. The user had an option to provide additional feedback if he/she desires.

In Study II users were asked to perform a variation in tasks for part four with the difference being in the way they trust keys. Instead of signing each key explicitly users were asked to place "ultimate" trust in the list moderator's public key when they received it. Thereafter, they did not have to place explicit trust in any keys that were already signed by the list moderator (i.e., the list encryption/decryption

key pair and other subscriber's public keys) as their email clients were able to leverage the transitive trust enabled by the GnuPG key trust assignment model.

Part V: SELS Vulnerability.

This part is similar to part III except that the email messages are sent over the SELS mailing list. The message types used are described in Table 1.

3.3.4 SELS Training and Documentation

Typically when new security software or new features in existing software are evaluated, users are given some training in the software/features. For SELS we decided that engaging in TPSE exercises served as sufficient training because if users are able to install keys and then use them to send and receive secure emails they should be capable of using SELS. Therefore, the initial TPSE usability evaluation served as both training for SELS as well as providing data for evaluating the usability issue of prior experience with GnuPG. To help users in completing these exercises we provided a minimal set of instructions in the emails that described the tasks and included key material. These instructions were independent of any email client in keeping with SELS objectives. For additional clarification, we referred the study user to the the SELS online documentation.

3.4 Observations and Analysis

For each user we took detailed notes of their ability to execute assigned tasks as well as the time taken for each task. The time taken between emails sent to the user that provided the instructions and emails sent back from the user with the results allowed for exact measurements of time taken to complete the tasks. For study parts two and four the *studymonitor* offered help to the user in the following ways (if requested): (1) when users were stuck they were asked to look at the instructions carefully, (2) when users found additional instructions on specific clients to be incorrect they were offered correct instructions, and (3) when they failed to proceed they were helped as much as needed so that they could proceed to the next part with this part being counted as a failed task.

A total of 12 users participated in Study I with 3 *expert* users and 9 *novice* users. Out of the 12 participants, 8 participants chose Thunderbird, 2 participants chose Mac Mail, 1 participant chose Mutt and 1 participant chose Outlook. All users except for one were able to complete enough tasks to allow for user study conclusions. The one that failed involved Microsoft Outlook that kept crashing so this client was excluded from further studies.³ A total of 8 users participated in Study II with 3 *expert* users and 5 *novice* users. All of them chose to use Thunderbird. While the number of *expert* users per study is small, previous research [26, 24] shows that a small number of users per class is sufficient to uncover most of the usability issues in a system. Specifically, 3 and 5 users can uncover about 67% and 85% of the usability issues respectively. However, our quantitative measures are preliminary as larger studies are needed to quantitatively measure usability [25]

³Interestingly, the GnuPG plug-in for Outlook seemed to crash only when used in conjunction with its key manager – a scenario that was not explored in our groupware walk-through.

Table 1: Message Types Sent to Users during GnuPG and SELS Focused Studies

Message Type and Description	Two Party Secure Email (TPSE) using GnuPG	Secure Email List Service (SELS)
Encrypted and signed correctly	This message is encrypted for the user and signed with a trusted key.	This message is signed and encrypted by a valid member of <i>studylist</i> , with a trusted signature key and the correct list encryption key.
Encrypted with wrong key	The email message is encrypted with a key that does not belong to the user. Hence the user cannot decrypt it.	This email message is encrypted with a key for which the user has no secret-key and delivered directly to the user but made to look like a message delivered on the list by forging the headers.
Encrypted and signed with forged "From"	The email message is encrypted with the user's key, but signed with a key that does not match the "From" address.	The email message is encrypted with the list key but signed with a key that does not match the "From" address.
Encrypted correctly but signed with a missing key	This email message is encrypted with the user's key, but is signed with a key for which the public-key is not available to the user.	This email message is encrypted with the list key, but is signed with a key for which the public-key is not available to the user.
Encrypted with forged "To"	The user is made to believe that this encrypted message was sent to the user and someone else by forging "To" header.	The user is made to believe that this encrypted only message was sent on the list by forging the headers. It is encrypted such that the user can decrypt it correctly.

Table 2: Usability observations from Study I

User Type	Key Install Success Rate		Key Install Time (Avg./Std. Dev. min.)		SUS Score		Changed Password
	TPSE	SELS	TPSE	SELS	TPSE	SELS	
Expert	2 of 3 (66.6%)	2 of 3 (66.6%)	6.5 / 2.12	11 / 1.41	85.83 / 5.20	76.67 / 11.55	3 of 3 (100%)
Novice	6 of 8 (75%)	2 of 8 (25%)	8.83 / 2.86	25.5 / 0.71	79.38 / 9.33	54.44 / 16.66	3 of 8 (37.5%)

Table 3: Usability observations from Study II

User Type	Key Install Success Rate		Key Install Time (Avg./Std. Dev. min.)		SUS Score		Changed Password
	TPSE	SELS	TPSE	SELS	TPSE	SELS	
Expert	3 of 3 (100%)	3 of 3 (100%)	4 / 0	12.66 / 2.01	74.17 / 20.21	74.16 / 23.23	2 of 3 (66.6%)
Novice	4 of 5 (80%)	5 of 5 (100%)	8.4 / 2.7	18.2 / 3.19	61.5 / 10.98	52 / 13.62	5 of 5 (100%)

Table 4: Security observations from Study I and Study II

User Type	Study I				Study II			
	% of Correctly Formed Msgs. Trusted (Avg. /Std. Dev.)		% of Incorrectly Formed Msgs. Trusted (Avg. /Std. Dev.)		% of Correctly Formed Msgs. Trusted (Avg. /Std. Dev.)		% of Incorrectly Formed Msgs. Trusted (Avg. /Std. Dev.)	
	TPSE	SELS	TPSE	SELS	TPSE	SELS	TPSE	SELS
Expert	100 / 0	100 / 0	16.67 / 14.43	8.33 / 14.43	100 / 0	100 / 0	8.33 / 14.43	16.67 / 28.87
Novice	93.75 / 17.68	100 / 0	18.75 / 17.68	15.63 / 12.94	100 / 0	100 / 0	30 / 20.92	35 / 13.69

Key Management: Installing and Managing SELS Keys.

The first of the two key management usability issues evaluated by this work is installation of per-list keying material by SELS users. The results of Study I are presented in Table 2. 66.6% of *expert* users (2 out of 3) and 75% of *novice* users (6 out of 8) were able to successfully complete the key installation tasks for the second part of the study (i.e., for TPSE) while 66.6% of *expert* users (2 out of 3) and only 25% (2 out of 8) of *novice* users were able to complete key installation tasks for the fourth part of the study (i.e., for SELS). Two-thirds of the users who failed to complete the key installation task for SELS and all the users who failed to complete the key installation task for TPSE did so because they either didn't know how to sign keys or what keys to sign in the case of SELS. In particular, users could not understand what a 'key-id' is and how to find the 'key-id' of a key.

Expert users took 6.5 minutes and 11 minutes on an average to complete key installation for TPSE and SELS respectively. On the other hand, *novice* users took 8.83 minutes on an average for TPSE key installation and 25.5 minutes on average for SELS key installation when they could successfully complete it. The increase in key-installation time from TPSE to SELS is due to the fact that SELS involves importing secret keys while TPSE only involves importing public keys and, furthermore, SELS involves importing multiple keys while our TPSE exercises involved importing only one key.

These results allowed us to quickly conclude that the SELS key installation process was too cumbersome for *novice* users. There are two ways to trust keys in GnuPG: using key signing and using explicit key trust assignment. We adopted the first approach for Study I because we assumed that understanding the GnuPG key trust assignment model would be

very complex for users. This assumption was based in part on the complexity of this model as it allows for transitive trust establishment and in part on previous usability studies with secure email [12, 28, 32]. However, it turns out that the key signing approach is not appropriate based on our usability observations. Instead, the key trust assignment model where the users place explicit trust in the list moderator's key and then use that trust to place transitive trust in all other SELS keys (as the list moderator's key signs all other keys) turns out to be more usable.

The evaluation results of this approach from Study II are presented in Table 3. All the *expert* and 80% (4 out of 5) of *novice* users were able to complete key installation tasks for the second part of the study (i.e., for TPSE) while all the *experts* and *novices* were able to complete key installation tasks for the fourth part of the study (i.e., for SELS). While *expert* users took 4 minutes and 12.6 minutes on average to complete key installation for TPSE and SELS, respectively, *novice* users took 8.4 minutes and 18.2 minutes on an average when they could successfully complete it. As can be seen from the results, there is a significant improvement in SELS key installation success rate in the case of novice users. We believe that the reason for this is that users find placing explicit trust in one key, namely, the list moderator's key, much easier than signing multiple keys. The second usability issue that we dealt with is managing and using multiple keys. The approach that we took to address this is to leverage the GnuPG key management capabilities whereby the plug-ins help locate the appropriate keys (for encryption, decryption, signing, and verification) but use a simplified password management solution. In using secret keys the user is prompted for his password for decryption and signing. Since all clients that we dealt with supported password caching for exactly one password, we recommended users (in SELS instructions) to set the passwords for all secret keys to be the same one. While all the *expert* users set the passphrase for the list key-pair to be the same as that of their GnuPG key-pair only 37.5% (3 out of 8) of *novice* users chose to change the passphrase in Study I. All the users who set the passphrase as recommended had little difficulty in sending (receiving) messages to (on) the list while users who did not set the passphrase as recommended had trouble figuring out which passphrase to use. 40% of users who didn't follow the recommendation initially did so later after realizing the ease of use it afforded. To address this issue we improved the instructions for Study II where we explained the consequences of password change in that it would be easier to manage decryption functions. Users were positively affected by the inclusion of this explanation in the instructions. Consequently, in Study II all *novice* users chose to change the password so that they have a single password to deal with. We note that once *novice* users chose to use a single password, they had no difficulty with sending and receiving signed and encrypted email as GnuPG plug-ins made it very straightforward. Only one *expert* user chose not to change the password but successfully completed the tasks.

To capture the effectiveness and satisfaction of interaction with SELS we asked users to fill out a SUS questionnaire. In study I, *Experts* gave TPSE and SELS SUS scores of 85.83 and 76.67 respectively on average while *Novices* gave SUS scores of 79.38 and 54.44 on average respectively. In study II, *Experts* gave TPSE and SELS SUS scores of 74.17 and 74.16 respectively on average while *Novices* gave 61.5 and

52. We believe that the key installation and management functions played a big role in the SUS scores that the software received though other factors such as prior familiarity with GnuPG have affected these scores. In order to account for such prejudices we look at the ratio of SELS SUS scores to TPSE SUS scores. In conducting the groupware walkthrough we realized that prior experience with GnuPG would help users in using SELS. Looking at the ratio of SUS scores between SELS and TPSE and comparing key installation success rates across SELS and TPSE will also help us better understand whether prior experience of GnuPG is necessary to use SELS. The ratio of SUS scores going from SELS to TPSE is 0.89 for *experts* on average, with a standard deviation of 0.1, and 0.68 for *novices* on average, with standard deviation of 0.18, in Study I. The average ratio of SELS SUS score to TPSE SUS score is 0.99⁴ with standard deviation of 0.09 for *experts* and 0.84 with standard deviation of 0.13 for *novices*. We see that ratio of SELS and TPSE SUS scores for *novices* improves significantly in Study II when compared to Study I and is comparable to that of *expert* users. Furthermore, the same number of *novices* completed SELS key installation as TPSE key installation in Study II while few novices could complete SELS key installation in Study I. This indicates 1) that the key management techniques adopted in Study II are significantly better than those in Study I and 2) that if the key management tasks are defined well with adequate instructions then even *novices* can quickly learn to use SELS effectively.

Vulnerability.

After designing usable security features/software it is important to evaluate these usable features for vulnerabilities. Such an evaluation helps identify limitations of usable features as well as early opportunities to address security problems. In keeping with our usability design we conducted our vulnerability evaluation on the interface mechanisms, namely, the cues provided by interfaces. We asked users to make a trust decision on both correctly and incorrectly formed emails where an incorrect email was either encrypted incorrectly, signed incorrectly, or both. The results for parts three and five for both Study I and Study II are shown in Table 4. In these studies users received two correctly formed and four malformed emails from the *studyadmin*. For Study I, the table shows all *experts* trusted all correctly formed TPSE and SELS messages. All *novices* trusted all correctly formed SELS messages but some did not trust some TPSE messages (6.25% with a standard deviation of 17.68). We see interesting results in the case of malformed messages for both TPSE and SELS. An average of 16.67% malformed TPSE messages and 8.33% malformed SELS messages were trusted by *experts*. In the case of *novices* however the average increases to 18.75% for TPSE and 15.63% for SELS. For both *experts* and *novices* the most commonly trusted malformed message is the **Encrypted and signed but with forged "From"** message. This case required the user to look carefully at the email headers to come to a right decision. The slight decrease in average of malformed messages trusted, going from TPSE to SELS, is due to the fact that a few users who failed to detect the mismatch between the

⁴ Surprisingly, in Study II an *expert* user gave SELS a better SUS score than TPSE. In the exit interview he remarked "given that the interface cannot be changed SELS is very well integrated with the email client".

email headers and security banner displayed by the email client's GnuPG plug-in, for the above mentioned malformed messages, did so for SELS.

For Study II, Table 4 shows that both *novices* and *experts* trusted all correctly formed messages. An average of 8.33% malformed TPSE messages and 16.67% malformed SELS messages were trusted by *experts*. In the case of *novices* however the average increases to 30% for TPSE and 35% for SELS. The increase in average going from TPSE to SELS in the case of *experts* is due to the fact that one *expert* user tended to trust unsigned messages that came or appeared to come on the list. However, the user wondered in his responses whether unsigned messages from the list should be trusted. The user even recommended in his comments that we mandate signed messages on the list.

In the case of novices the average of trusted malformed messages increased compared to Study I because apart from trusting **Encrypted and signed but with forged "From"** messages, some users tended to trust unsigned messages as long as the sender was known or a member of the list. Another observation made during Study II is that users that trusted encrypted but unsigned messages did not trust encrypted and signed messages if they could not verify the signature, i.e., they did not have and could not fetch the public-key of the sender. This is because the GnuPG plug-in for Thunderbird, the email client used by majority of users in our studies, alerted the users with a yellow banner that said "Unverified signature" whenever it could not verify the signature on a message. Whereas for encrypted-only messages it displayed a blue banner, as opposed to a green banner for an encrypted and signed message which was decrypted properly and whose signature was verified. The blue banner did not attract the user's attention to the fact that message was unsigned and hence could be untrustworthy. This leads us to believe that most users would not have trusted **Encrypted and signed but with forged "From"** if the message signer's key was unknown to the receiver. Thus making the above attack, which many *novices* and a few *experts* did not detect, viable only as an "insider attack".

Summary. Effectiveness of SELS is demonstrated by the fact that *all* novices in Study II were able to successfully install list keys, send and receive messages on the list, and trust correctly formed messages. An equally important measure of effectiveness is the vulnerabilities provided by SELS. While this number is greater than ideal (35% for novices), it is only slightly different from that for the TPSE evaluation indicating that SELS introduces minimal additional vulnerabilities, if any. Efficiency of SELS is demonstrated by the fact that novices were able to complete key installation in 18 minutes. The study also indicates that effectiveness and efficiency may improve with time as experts were able to successfully complete the key installation and send and receive messages in around 12 minutes and were more resistant to malformed messages (16.67%). In addition, many users were not familiar with the email client used in the study and we assume that gaining familiarity will help as well. Furthermore, user satisfaction for SELS, measured by the SUS score, was similar to that of the underlying email client and GnuPG plug-in combination (i.e., SELS SUS score to TPSE SUS score ratio is close to 1). This implies that users will find SELS almost as satisfying as secure email in general.

While we do not make changes to existing interfaces in

SELS, our study recommends three modifications for GnuPG plug-ins to further improve the usability and security of SELS. We recommend that GnuPG plug-ins for email clients, 1) support caching of multiple passwords, 2) flag encrypted only emails as untrusted and 3) alert users when signer and sender do not match. The first will allow users to establish different passwords for each private key in a usable manner thus making the system more secure. The second will strongly recommend users to trust only signed messages coming on SELS (which was the primary problem for the slightly increased vulnerability measurement for SELS). Interestingly, this feature is already provided for the Mac Mail GnuPG plug-in, and we recommend that all email clients do so. The third will help decrease vulnerabilities for both two-party secure email and SELS.

4. SELS DEPLOYMENT EXPERIENCE

After a successful usability study the SELS software was hardened and installed in a production environment to support users. Our first user community was the TeraGrid Security Working Group (or simply, TG-WG) that has been using encrypted group email for several years to exchange sensitive information about vulnerabilities, incidents and recovery procedures for TeraGrid high-performance computing systems distributed across 11 sites. Until SELS came along they were using password-based symmetric-key encryption in PGP with the passwords being distributed in telephone conference calls. This approach required secure distribution and maintenance of multiple passwords (a security issue) and mapping passwords to current and prior emails (a usability issue). The community adopted SELS in the hope for a more usable and secure solution. In this section we describe our experiences in supporting the TG-WG over a ten-month period from January through October 2008. We describe major issues in areas of usability, trust and security, and software engineering that arose while supporting TG-WG and how they were resolved. Overall, SELS has been very successful in supporting TG-WG as evidenced by the large number of encrypted emails exchanged by the group.

Table 5: SELS new features and enhancements since TG

SELS Release	New Features
0.5.5	New Trust Model, Decryption only user key pair.
0.5.8	Bounce Messages for wrong LK and HTML messages.
0.6	Two key lengths (1024 and 2048)
0.7	Generate and store Revocation Certificate for LK, Remove email address from User key pair.
1.0	Key Update, Delete a Subscriber, Mailman Patch, Improved error handling

4.1 SELS Usage Statistics

The List Moderator for TG-WG created two lists, we will refer to them in this paper as List-A and List-B, on the SELS List Server hosted at NCSA. List-A has 52 subscribers and an average of 32 encrypted emails were exchanged per month. List-B has 50 subscribers and an average of 2 encrypted emails were exchanged per month. This

Table 6: Bugs fixed based on feedback from TeraGrid users and code review)

Bug Number	Description	Fix
16	Minor coding error in LM (Code Review)	Code fixed
17	Add a check to see if Java is installed correctly (Code Review)	Check added
18	Minor coding error in LM (Code Review)	Code fixed
19	Minor coding error in LM (Code Review)	Code fixed
20	GnuPG prompts different on Windows and *nix platforms. Add functionality to support both.	This fix was added. LM Code is platform independent again.
21	Fix file name syntax for revocation certificates on Windows.	This fix was added. LM Code is platform independent again.
22	Support bounce for HTML messages from Outlook 2007 and Outlook 2003 (User Feedback)	Bounce support added to LS code.
23	Race condition for GPG interactive command “—edit-key” used in LM code.	This bug was hard to fix in absence of a multi-platform <i>expect</i> like module for Python. So used a different approach, using Java, where GnuPG interactive commands are no longer required.

data has been collected since the TeraGrid lists were created 10 months ago. Prior to using SELS, TG-WG used password based symmetric key encryption. In 2006 and 2007, using that approach, the average number of encrypted emails exchanged per month on List-A was 9 and on List-B was 3. Conversations with some of the list members indicated that the increase in number of messages on the lists was due to the ease of use with which members could exchange secure (encrypted) messages using SELS.

4.2 Usability and Security

About 8 percent of the TG-WG list subscribers, that is 4 out of 52,⁵ needed support while installing keys and sending messages using SELS. Given prior GPG experience we deemed all TG-WG users to be “experts”. Therefore, this observation matches our usability study results in that most expert users were able to install keys and send messages. Most of these subscribers who had trouble were using an email client without a GnuPG plugin, e.g. Microsoft Entourage, but were able to import these keys via the command line. However, when they wanted to send a message they encrypted it using the GPG command-line interface and sent an email with the encrypted message as an attachment. Since SELS supports encrypted attachments only when using PGP/MIME their messages were dropped at the server. Improved dissemination of compatibility information (e.g., via FAQs) provided the needed resolution; in particular the need to paste the encrypted message in the body of the email for Entourage. Some of the subscribers who had trouble did not realize that they needed to install a separate set of keys for each list and hence had trouble sending and receiving email on one of the lists. Again, improved instructions via email and on the SELS web site by the SELS team resolved the issue. After the users installed the second set of keys they were successful.

While most of the TG list users were comfortable with installing and trusting SELS keys, some of them had concerns about setting “Ultimate” trust in the moderator’s key. We chose this mechanism to place trust in SELS keys as it was simple, *i.e.*, required users to perform only one operation, and it enabled transitive trust in any key signed by the moderator without requiring any explicit action from users. In order to address this concern we adopted a recommenda-

tion made by one of the users to let the members set “full trust” in the moderator’s key which is a lower level of trust than “Ultimate” and had them locally sign the moderators key. Thus, we retained the transitive trust model which was shown to be more usable in our study while addressing users’ concerns. This is an example where users traded some usability for improved security. This is not surprising given that most TG users had prior experience with GPG/PGP and are much more security conscious than an average user.

Another concern that TG users had was in managing multiple private PGP keys for SELS. PGP keys have multiple key-pairs in them, namely a signing key-pair which is the main key and one or more decryption key-pairs called sub-keys. In older versions of PGP it was possible to have just one key-pair that is either used for signing or decryption or both depending the version. But in the recent version it is not possible to generate encryption-only keys. Therefore, our SELS keys, though intended only to be used for decryption, had a signing key-pair in them along with the decryption sub-key. So users had concerns that they might accidentally use the SELS keys for 1) signing messages or 2) encrypting messages outside of the mailing list. This is because many GPG plugins presented users with a drop down menu to select a key when signing messages. Those users that had multiple signing keys were worried that their chances of accidentally selecting the wrong key increased when they belonged to (multiple) SELS lists. To address the first concern we explicitly removed the secret key of the signing key-pair from SELS keys after key generation. To address the second concern we removed the email address from the name of the key. This distinguished SELS keys from most PGP key-pairs that can be used for signing/encrypting and reduced the chance of inadvertent misuse.

4.3 Features and Enhancements

SELS 0.5 was the first version that was used for TeraGrid lists. This version had the capability to bounce plaintext messages back to the sender if the list was configured to allow only encrypted or encrypted and signed messages. TeraGrid lists were initially configured to allow plaintext messages, but were changed to only allow encrypted messages a couple of months later, to avoid sending sensitive messages in plaintext by mistake. Over the last ten months SELS was enhanced with many features. Based on user feedback two new bounce messages were added for two cases in SELS re-

⁵Most members belonged to both lists.

lease 0.5.8. SELS software supports messages sent in a text format or messages encrypted as PGP/MIME when sent in some other format. So if a message is composed as HTML and not sent as PGP/MIME, it will be dropped at the List Server. Similarly a message encrypted with an incorrect key is dropped at the server. These messages were being dropped at the SELS List Server without any notification to the sender which left the sender wondering where his/her message went. To improve usability, SELS was modified to always generate a bounce message whenever a message is dropped.

While SELS was being used by the TG-WG, the SELS Team felt the need to support additional clients based on users' preferences.

1. Gmail with FireGPG: SELS can be used easily with Gmail and the FireGPG plug-in.
2. PGP Desktop: Some TG-WG users use PGP Desktop as a key manager with email clients such as Outlook 2007 and Apple Mail with SELS. A few changes to the SELS code were needed to make SELS compatible with both PGP and GnuPG. One major change was to revert the encryption algorithm between release 0.7 and 1.0. SELS is back to using CAST5 instead of AES256 since CAST5 is better supported by PGP.

Some TG-WG users were using email clients that do not have available GnuPG plug-ins, *e.g.*, Outlook 2007 and Entourage. However these users are able to use SELS by using the GPG command line for key management and encrypting or decrypting email messages and using Outlook 2007 or Entourage for sending and receiving messages to/from the List Server.

Other features added include support for automatically generating and storing a revocation certificate for the list key LK and functions that automate key updates for subscribers. In Table 5 we summarize new features developed in SELS since the TG-WG has been supported. We also undertook a software architecture and code review that helped us streamline our code and fix many bugs. In Table 6 we present major bug fixes undertaken during this time.

4.4 SELS Production Environment

The SELS production environment, shown in Figure 5, consists of primary and backup industrial-grade, rack mount host servers that support Linux virtual machines for each individual SELS instance. Both servers feature redundant power supplies (one of which is connected to an uninterruptible power supply) as well as hardware-based RAID mirrored disk drives. A monitoring script on a remote server watches both hosts for network connectivity, which indicates that the hosts are online and working properly.

A virtual machine (VM) is created for each instantiation of SELS, then replicated to the backup host. Each VM is configured with the hostname of the list, *e.g.*, `sels-example.ncsa.edu`, but has its primary network connection set up with a different name, *e.g.*, `sels-example1` or `sels-example2`. This allows each VM to be addressed separately, but the common hostname ensures that the web interface to the Mailman list server works correctly when creating new email lists. Then on either the primary or the backup VM, the SELS list name (`sels-example`) is set up as a virtual Ethernet interface, *i.e.*, `eth0:0`. This DNS configuration

allows the backup VM to function equally well as the primary VM. However, in order for a transition to the backup to function properly at the software level, all changes to the primary VM are synchronized to the backup by a script that securely copies SELS software changes, mailman list information, and new user data (user proxy-keys).

During the testing and pilot implementation phases of the project, several changes were made to the service infrastructure. Initially, all operating system software packages on the VMs were automatically updated. However, this caused a SELS failure (luckily prior to the pilot stage), so the automatic update configuration was modified to prevent any updates of specific packages, namely mailman and sendmail (which we use as the Mail Transfer Agent or MTA). Whenever updates to either of these two packages are available, a manual update is performed on the backup VM. If careful testing demonstrates that SELS functionality has been maintained, or after a fix to the package upgrade can be determined (usually a configuration file tweak), then the package update is rolled over to the primary VM. This periodic validation ensures that the backup VM can function properly equally well as the primary SELS server. Initially, we manually modified mailman python files in order to include SELS functionality. However, by slightly modifying the SELS code, we were able to amend the mailman instance after it was upgraded via a simple patch script.

In the duration of supporting TG-WG, there have been no hardware failures, so no fail overs have been needed. We decided against using automatic fail overs to prevent the instance where both the primary and backup VMs are both listening to the email list IP address. Manual fail overs are precipitated when the remote monitoring script indicates that the primary VM has experienced a hardware failure or that network has been disconnected from the primary VM. One such instance did occur: a power glitch restarted the network switches, although the UPS protected the primary VM, which merrily hummed along. The temporary network outage caused the mailman daemon to stop processing emails properly, so it had to be restarted. To take this problem into account, the monitoring script now checks whether mailman is working correctly and restarts the daemon if this is not the case.

5. RELATED WORK

Secure Mailing Lists. Secure mailing lists have been used for a while in the US Department of Defense as part of the Defense Message System (DMS). This system uses enhanced S/MIME techniques presented in [16]. DMS satisfies the identified security properties of strong confidentiality, authentication and integrity. It uses a hardware lockbox at the server to provide strong confidentiality combined with an externally supported key distribution system.⁶ DMS uses digital signatures to provide authentication and integrity. Additionally, it uses an outer layer signature to provide authentication at *LS* for encrypted messages. However, to achieve these security properties DMS uses specialized email clients that can process messages to provide these properties and a hardware lockbox at the server. In contrast, SELS is a purely software based solution and imposes no additional burden on client-side software and only a software plugin on

⁶Details on DMS confidentiality property were provided by Stephen Kent of BBN in personal communications.

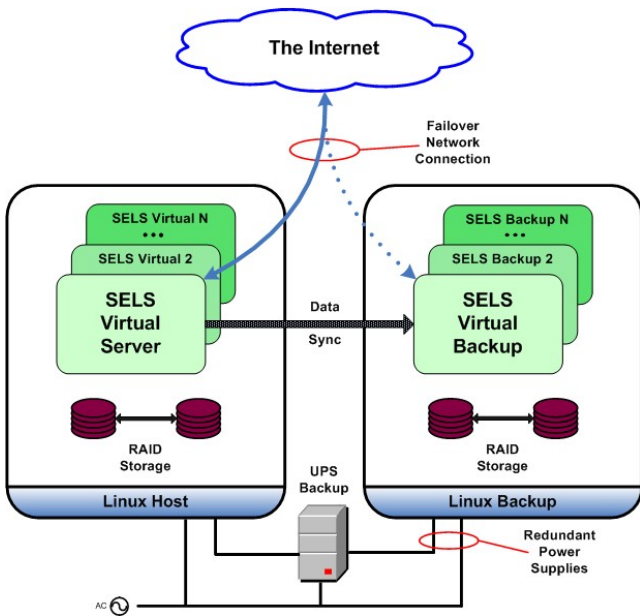


Figure 5: SELS Production Environment

the server. Furthermore, DMS leverages organizational CAs to build trust while SELS leverages personal trust among participants. At the same time we believe that there may be environments where DMS is a better fit than SELS.

In the open-source arena simple approaches that extend security solutions for two-party email to mailing lists have already been developed; e.g., SSLs⁷ and Sympa⁸. In these solutions, subscribers send emails to the list server encrypted with the list server’s public key. The list server decrypts the emails and then re-encrypts them for every subscriber using their registered public keys. Clearly, these solutions do not satisfy the confidentiality requirement as they allow the list server access to decrypted emails.

Usability of Secure Email has received considerable attention since the seminal work of Whitten and Tygar [32], which identified several shortcomings of PGP. Garfinkel *et al.* [11, 13] demonstrate the potential high success of digitally signed email in an e-commerce context with a user study. Gaw *et al.* [14] study the social issues that affect the adoption of secure email. Garfinkel and Miller [12] and Roth *et al.* [28] explore alternative key management techniques that make secure email easier to adopt and they demonstrate the effectiveness of their techniques with user studies. However, to date all usability studies focus on secure two-party email exchange. To the best of our knowledge ours is the first study to focus on secure mailing lists. Furthermore, our study is significantly different in that we utilize usability techniques to improve secure email software usability without imposing software enhancement requirements.

Usability Techniques employed in our study, namely, groupware walkthrough and focused user study with particular attention to skill level, are promising techniques that have not been fully applied to secure systems. Groupware walkthrough was proposed by Pinelle and Gutwin [27] to allow for the inclusion of context in groupware usability evalu-

ations and is based on the often used cognitive walkthrough technique. Contextual information such as dynamic nature of group work and variability of tasks for multiple concurrent users allows for the identification of usability problems that may not be possible with other techniques. Faulker and Wick [10] present an extensive analysis of the benefits of user studies that employ a mix of novice and expert users. In particular, they argue quantitative between-group comparisons offer exclusive insights into usability problems.

Proxy Encryption. Previous proxy encryption schemes enable unidirectional and bidirectional proxy transformations by first setting up a transformation agent that is given the proxy key and then sending messages to the agent for transformation [2, 17, 22]. Unidirectional schemes only allow transformations from some entity A to another entity B with a given proxy key while bidirectional schemes additionally allow transformations from B to A with the same proxy key. For SELS we need a proxy encryption scheme that allows for the transformation from one entity, LS , to many subscriber entities (i.e., to all list subscribers). The El Gamal based unidirectional proxy encryption scheme of Ivan and Dodis [17] is closest in nature to SELS with the additional relationship between the proxy keys (i.e., $\forall_i K_{U_i} + K_{U_i} = K_{LK}$) imposed to allow for a single list encryption key, PK_{LK} , to suffice. Extending the RSA based unidirectional scheme of [17] in a similar manner will not work because it would require the sharing of the modulus across all list subscribers. Jakobsson [18] and Zhou *et al.* [35] allow for proxy transformation without the need for distributing proxy keys but use costly threshold crypto-systems to ensure the necessary security. Ateniese *et al.* [1], Green *et al.* [15] and Canetti and Hohenberger [7] extend proxy encryption schemes with useful properties such as non-interactiveness, which for SELS might allow for generation of proxy keys without involving both LM ’s and LS ’s decryption keys. We believe that while deployment of applications based on these novel schemes faces challenges with infrastructure compatibility and lack of commonly available tools, it is an open problem to overcome these challenges. For example, our experiences suggest that users are unlikely to move to a different email client just to be able to use an advanced secure email solution. However, advanced future systems based on these schemes are likely to provide strong security guarantees and may prove to be very useful in practice.

Multi-recipient Email Encryption. The problem of sending confidential messages to multiple recipients has been addressed in the past via multi-recipient email encryption [30], multi-party certified email [34], secure group communication and broadcast encryption. A major difference between these approaches and ours is that by using a mailing list we remove the user’s burden of managing recipient addresses and public keys while still satisfying the confidentiality requirement. In these approaches the sender must manage the recipient list and address all of the intended recipients directly. In multi-recipient email encryption, Wei *et al.* [30] combine techniques from identity-based mediated RSA and re-encryption mixnets to enable a sender to encrypt messages to multiple recipients with only two encryptions (as opposed to one encryption for each recipient in the trivial case). To do so, they use a partially trusted demultiplexer that is akin to LS in terms of its security properties but also use an additional fully trusted CA. If their scheme were to be adapted for mailing lists it would require devel-

⁷<http://non-gnu.uvt.nl/mailman-ssls>

⁸<http://www.sympa.org>

opment of client-specific plugins. In SELS the sender needs to execute only one encryption allowing compatibility with existing messaging formats and tools thereby avoiding the need to develop client-specific plugins. In multi-party certified email [34], the sender must maintain each recipient's public key and encrypt the message individually to each recipient. This overhead is avoided in SELS via the use of mailing lists while still providing confidentiality.

In secure group communication either a trusted group controller (e.g., LKH [33]) distributes session keys to group members or the group members generate session keys in a distributed manner (e.g., TGDH [21]). In either case, list subscribers would have to maintain state on current session keys and update them on every membership change (whereas in SELS existing subscribers are not affected by the joins and leaves of other members). This makes the use of secure group communication techniques impractical for secure mailing lists as it goes against the nature of the largely offline email use. So-called "stateless" broadcast encryption schemes (e.g., [17], [3]) allow for encryption of messages to a dynamic set of group members without the members requiring to maintain state and executing key updates on membership changes. However, they vary the encryption key and cipher-text sizes depending on the group membership. This variation cannot be supported by today's email standards making such solutions difficult to implement. SELS, on the other hand, addresses the confidentiality and deployability requirements of secure mailing lists in a practical way.

6. CONCLUSIONS

In this work we have described the process of going from the existing SELS prototype [20] to a usable and deployed software solution. We conducted an usability study whose high-level goals were to evaluate and enhance the usability (i.e., effectiveness, efficiency, and satisfaction) of the SELS key management system for list subscribers with the strong preference that the solution be compatible with commonly used email clients. We have deployed SELS and report on our experiences in supporting the TeraGrid Security Working Group for a period of ten months. Success of SELS is clearly indicated by the fact that there has been a nearly four-fold increase in the number of encrypted emails exchanged by the TeraGrid users since they adopted SELS with anecdotal evidence suggesting that this increase is largely due to better usability provided by SELS. The SELS software is now available for community use. We look forward to continuing to support and improve the software based on input from the user community.

7. ACKNOWLEDGMENTS

The authors would like to thank James Marsteller and Tim Brooks for facilitating the adoption of SELS by the members of TeraGrid Security Working Group. We thank Pooja Agarwal for helping with unit testing and code review and all the users that participated in the usability evaluation. We also thank Von Welch for helpful discussions through the course of this effort. This work is funded by Office of Naval Research under Grant Nos. N00014-06-1-1108 and N00014-07-1-1173. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research.

8. REFERENCES

- [1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, 2006.
- [2] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *EUROCRYPT*, pages 127–144, 1998.
- [3] D. Boneh, C. Gentry, and B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of International Cryptology Conference (CRYPTO)*, pages 258–275, 2005.
- [4] J. Brooke. SUS: a quick and dirty usability scale. In P. W. Jordan, B. Thomas, B. A. Weerdmeester and A. L. McClelland (eds.). *Usability Evaluation in Industry*. London: Taylor and Francis., 1996.
- [5] N. Brownlee and E. Guttman. Expectations for Computer Security Incident Response. IETF Network Working Group, Requests for Comments, RFC 2350, June 1998.
- [6] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP Message Format. IETF Network Working Group, Request for Comments, RFC 2440, November 1998.
- [7] R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *CCS '07: Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194, New York, NY, USA, 2007. ACM.
- [8] Y.-P. Chiu, C.-L. Lei, and C.-Y. Huang. Secure multicast using proxy encryption. In *International Conference on Information and Communications Security (ICICS)*, pages 280–290, 2005.
- [9] A. J. DeWitt and J. Kuljis. Aligning usability and security: a usability study of polaris. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 1–7, New York, NY, USA, 2006. ACM Press.
- [10] L. Faulkner and D. Wick. Cross-user analysis: Benefits of skill level comparison in usability testing. *Interacting with Computers*, 17(6):773–786, 2005.
- [11] S. L. Garfinkel, D. Margrave, J. I. Schiller, E. Nordlander, and R. C. Miller. How to make secure email easier to use. In *CHI: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 701–710, 2005.
- [12] S. L. Garfinkel and R. C. Miller. Johnny 2: A User Test of Key Continuity Management with S/MIME and Outlook Express. In *Symposium on Usable Privacy and Security (SOUPS '05)*, 2005.
- [13] S. L. Garfinkel, J. I. Schiller, E. Nordlander, D. Margrave, and R. C. Miller. Views, Reactions and Impact of Digitally-Signed Mail in e-Commerce. In *Financial Cryptography*, pages 188–202, 2005.
- [14] S. Gaw, E. W. Felten, and P. Fernandez-Kelly. Secrecy, flagging, and paranoia: adoption criteria in encrypted email. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 591–600, New York, NY, USA, 2006. ACM Press.

- [15] M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security (ACNS)*, pages 288–306, 2007.
- [16] P. Hoffman. Enhanced Security Services for S/MIME. IETF Network Working Group Request for Comments (RFC) Document 2634, June 1999.
- [17] A.-A. Ivan and Y. Dodis. Proxy cryptography revisited. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium*, 2003.
- [18] M. Jakobsson. On quorum controlled asymmetric proxy re-encryption. In *PKC '99: Proceedings of the Second International Workshop on Practice and Theory in Public Key Cryptography*, pages 112–121, London, UK, 1999. Springer-Verlag.
- [19] A. Kapadia, P. Tsang, and S. W. Smith. Attribute-Based Publishing with Hidden Credentials and Hidden Policies. In *Proceedings of The 14th Annual Network and Distributed System Security Symposium (NDSS '07)*, February 2007.
- [20] H. Khurana, J. Heo, and M. Pant. From proxy encryption primitives to a deployable secure-mailing-list solution. In *International Conference on Information and Communications Security (ICICS)*, pages 260–281, 2006.
- [21] Y. Kim, A. Perrig, and G. Tsudik. Tree-based group key agreement. *ACM Transactions on Information and System Security*, 7(1):60–96, 2004.
- [22] M. Mambo and E. Okamoto. Proxy cryptosystem: Delegation of the power to decrypt ciphertexts. *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences*, E80(A(1)):54–63, 1997.
- [23] J. Nielsen. Novice vs. Expert Users. <http://www.useit.com/alertbox/20000206.html>, Feb 2000.
- [24] J. Nielsen. Why You Only Need to Test With 5 Users. <http://www.useit.com/alertbox/20000319.html>, March 2000.
- [25] J. Nielsen. Quantitative Studies: How Many Users to Test. http://www.useit.com/alertbox/quantitative_testing.html, June 2006.
- [26] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 206–213, New York, NY, USA, 1993. ACM.
- [27] D. Pinelle and C. Gutwin. Groupware walkthrough: adding context to groupware usability evaluation. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 455–462, New York, NY, 2002.
- [28] V. Roth, T. Straub, and K. Richter. Security and usability engineering with particular attention to electronic mail. *International Journal on Human Computer Studies*, 63(1-2):51–73, 2005.
- [29] T. S. Tullis and J. N. Stetson. A Comparison of Questionnaires for Assessing Website Usability. In *Usability Professional Association Conference*, 2004.
- [30] W. Wei, X. Ding, and K. Chen. Multiplex encryption: A practical approach to encrypting multi-recipient emails. In *International Conference on Information and Communications Security (ICICS)*, pages 269–279, 2005.
- [31] M. J. West-Brown, D. Stikvoort, K.-P. Kossakowski, G. Killcrece, R. Ruefle, and M. Zajicek. Handbook for Computer Security Incident Response Teams (CSIRTs). CERT Handbook, CMU/SEI-2003-HB-002, April 2003.
- [32] A. Whitten and J. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *8th USENIX Security Symposium*, 1999.
- [33] C. K. Wong, M. Gouda, and S. S. Lam. Secure group communications using key graphs. *IEEE/ACM Transactions on Networking*, 8(1):16–30, 2000.
- [34] J. Zhou. On the security of a multi-party certified email protocol. In *International Conference on Information and Communications Security (ICICS)*, pages 40–52, 2004.
- [35] L. Zhou, M. A. Marsh, F. B. Schneider, and A. Redz. Distributed blinding for distributed elgamal re-encryption. In *ICDCS '05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 824–824, Washington, DC, USA, 2005. IEEE Computer Society.

Usable Secure Mailing Lists with Untrusted Servers

Rakesh Bobba, Joe Muggli, Meenal Pant,
Jim Basney and Himanshu Khurana

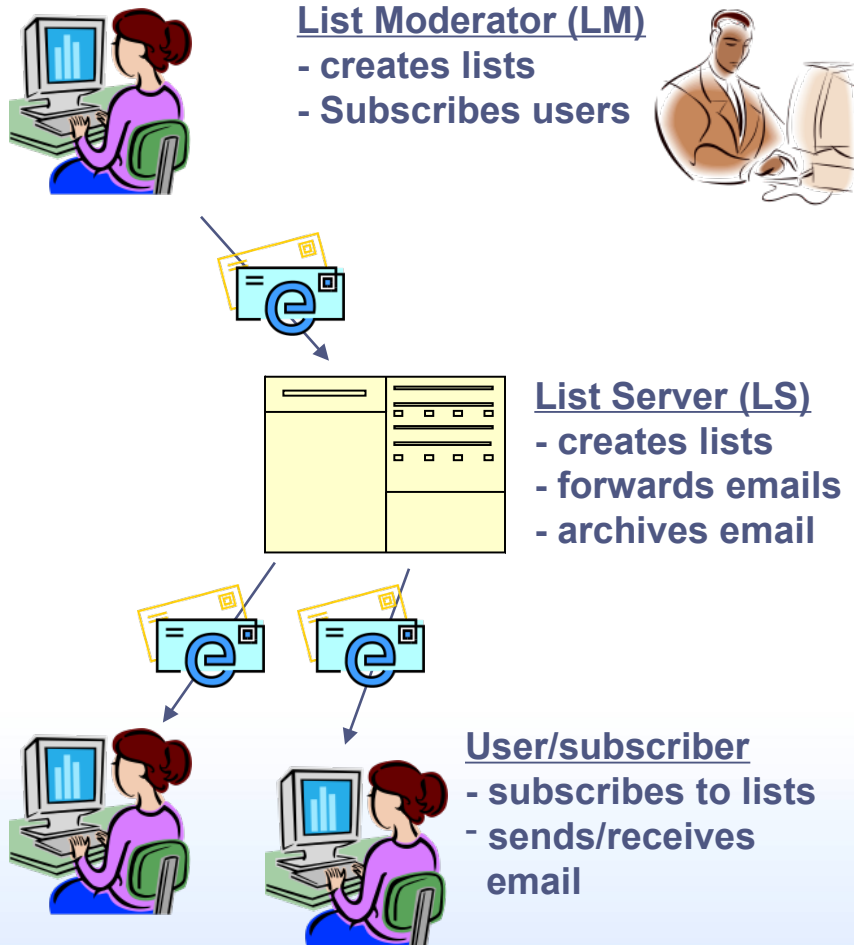
IDtrust, April 14 – 16, 2009.
Gaithersburg, MD



ILLINOIS

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Introduction to Mailing Lists



- **Mailing Lists (MLs) enable users to easily exchange emails**
 - LS bears all the overhead
- **Increasingly popular for exchange of both public and private content ⇒ security is an important concern**
- **Little or no work in providing security solutions for MLs**
 - We provide SELS: Secure Email List Services
 - solutions for confidentiality, integrity, and authentication

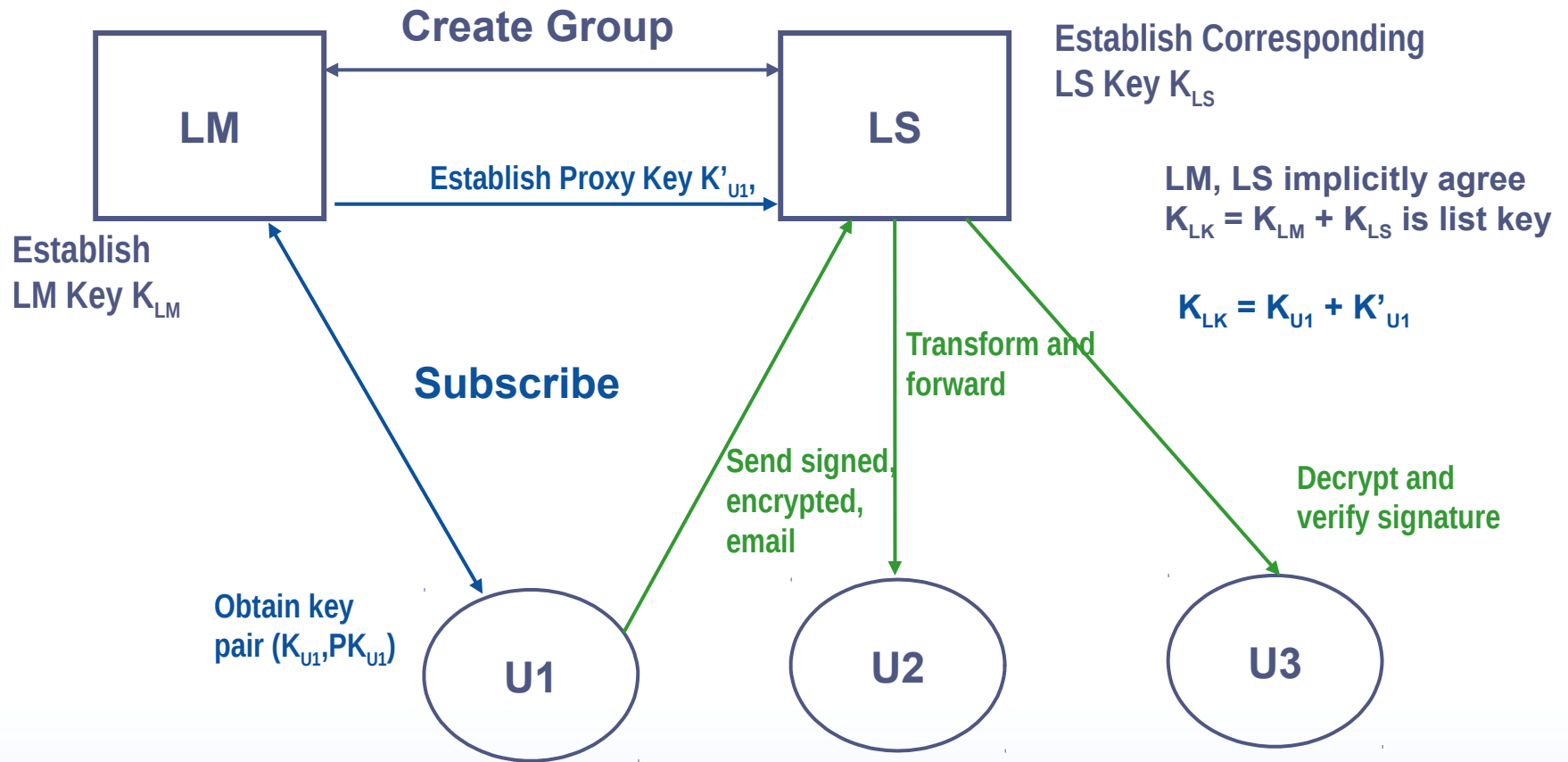
Untrusted Servers

- Existing Solutions
 - Password based encryption (end-to-end confidentiality)
 - Clunky to exchange and manage passwords out-of-band whenever a subscriber leaves
 - Encrypt to LS, which decrypted and re-encrypted with subscriber keys
 - LS takes care of key management
 - LS had access to plaintext messages.
- Desirable to Reduce Trust Liability
 - Trust LS to manage lists and forward messages correctly
 - But do not trust LS with content of messages – “untrusted server”

SELS History

- **Original SELS protocol.**
 - Himanshu Khurana, Adam Slagell, and Rafael Bonilla. SELS: A Secure E-mail List Service. In proceedings of the Security Track of the ACM Symposium on Applied Computing (SAC), March 2005.
- **Modified, practical version of SELS, with extensive experimentation and integration.**
 - Himanshu Khurana, Jin Heo, and Meenal Pant. From Proxy Encryption Primitives to a Deployable Secure-Mailing-List Solution. In the Eighth International Conference on Information and Communications Security (ICICS '06), Raleigh, North Carolina, December 2006.

Protocol Overview



Proxy re-encryption at LS ensures that plaintext is not exposed

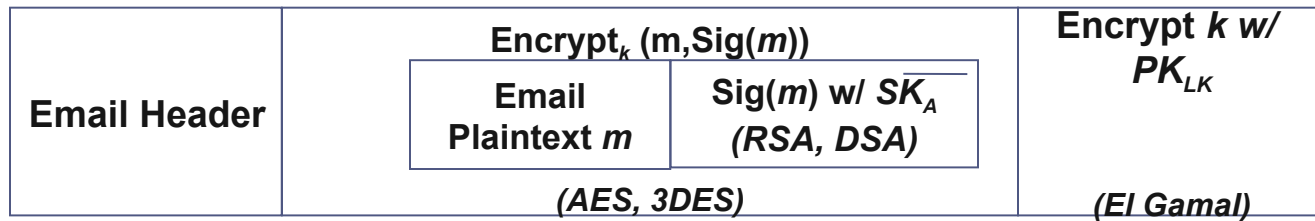
- **Assumption: LM is an independent entity not controlled by LS**

Sending Emails in SELS

Keyring: (\overline{SK}_A , PK_{LK})

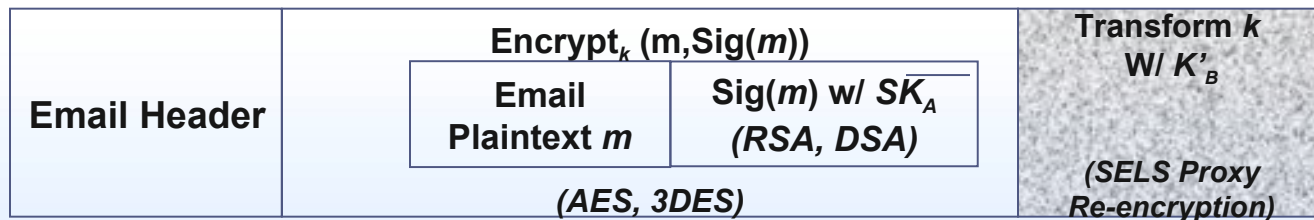
Keyring: Members' proxy keys K'_{ui}

Alice → LS



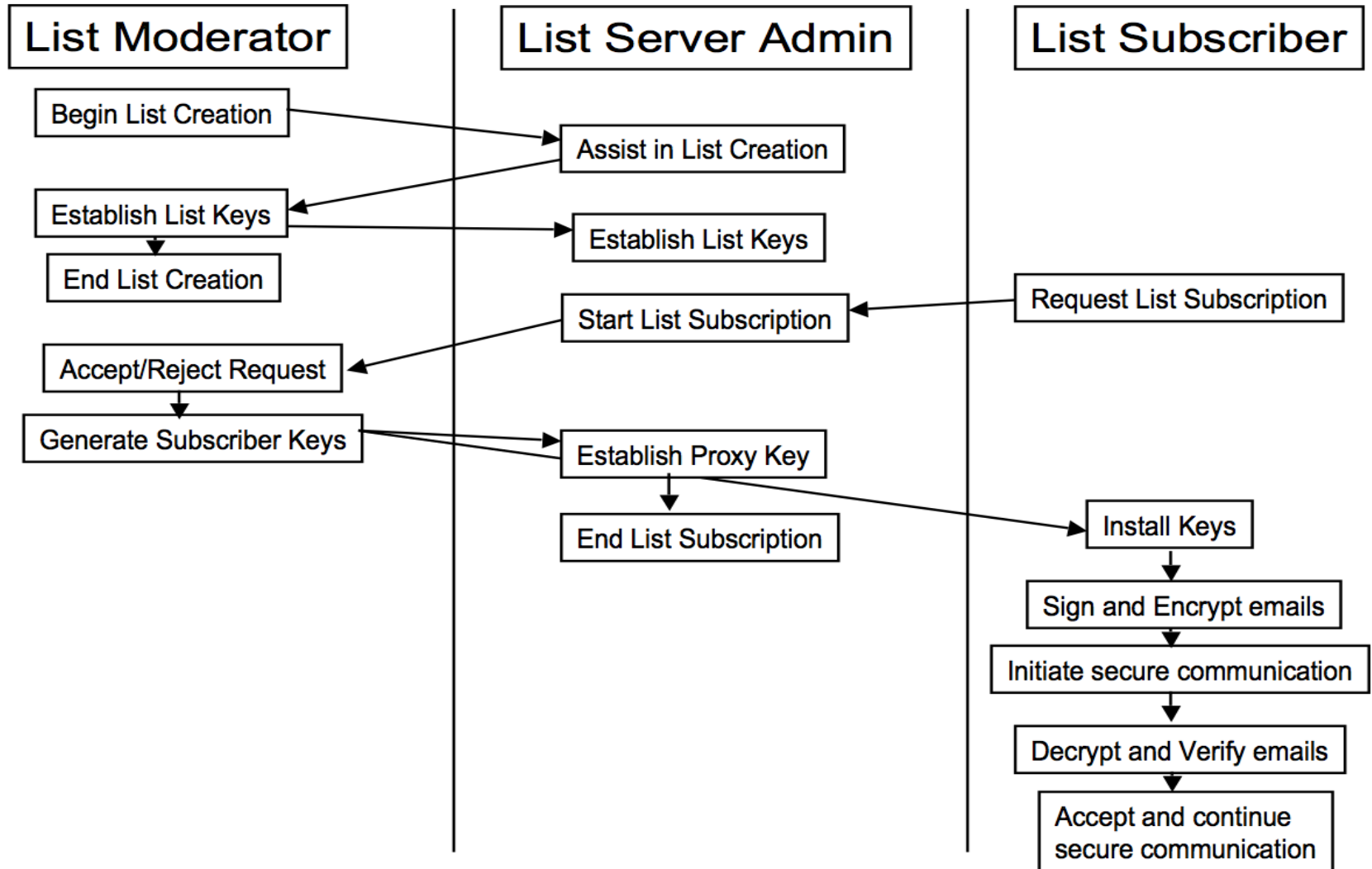
Keyring: (PK_A , \overline{SK}_B)

Bob ← LS



Suitable for environments where GPG is/can be used

Preliminary Usability Evaluation: Groupware Walkthrough



Potential Usability Issues

- **Installation of multiple keys**
 - List public-key and user decryption key pair (includes private key)
 - Installing a private key is not common operation
 - Place appropriate trust in the keys
 - Sign them or use PGP trust model
- **Managing and using multiple keys**
 - Users get a private key for every SELS list
 - Need to remember passwords for each key or set same password for all keys
 - Most GPG plug-ins cache only one password
- **Prior GPG experience**
 - Lack of GPG knowledge/experience might make it unusable

Focused User Study - Setup

- Two Studies
 - Study I – sign keys to place trust
 - Study II – use PGP trust model
- Two user groups in each study
 - Novice – no prior GPG experience (8 in study I and 5 in study II)
 - Experts – prior GPG experience (3 in study I and 3 in study II)
- 5 Parts to each study
 - Background questionnaire
 - Two Party Secure E-mail (TPSE) key installation and message exchange using GPG
 - SUS questionnaire
 - TPSE Vulnerability Evaluation
 - Tasks involving SELS key installation and message exchange
 - SUS questionnaire
 - SELS Vulnerability Evaluation

Focused User Study - Results

Observations from Study I

User Type	Key Install Success Rate		Key Install Time (Avg. / Std. Dev)		SUS Score		Changed Passwd.
	TPSE	SELS	TPSE	SELS	TPSE	SELS	
Expert	2 of 3	2 of 3	6.5 / 2.12	11 / 1.41	85.83 / 5.2	76.67 / 11.55	3 of 3
Novice	6 of 8	2 of 8	8.83 / 2.86	25.5 / 0.71	79.38 / 9.33	54.44 / 16.66	3 of 8

Observations from Study II

User Type	Key Install Success Rate		Key Install Time (Avg. / Std. Dev)		SUS Score		Changed Passwd.
	TPSE	SELS	TPSE	SELS	TPSE	SELS	
Expert	3 of 3	3 of 3	4 / 0	12.66 / 2.01	74.17 / 20.21.2	74.16 / 23.23	2 of 3
Novice	4 of 5	5 of 5	8.4 / 2.7	18.2 / 3.19	61.5 / 10.98	52 / 13.62	5 of 5

Focused User Study – Vulnerability

Message Type and Description	Two Party Secure Email (TPSE) using GPG	SELS Messages
Encrypted and signed correctly	This message is encrypted for the user and signed with a trusted key.	This message is signed and encrypted by a valid member of list, with a trusted signature key and the correct list encryption key.
Encrypted with wrong key	The email message is encrypted with a key that does not belong to the user. Hence the user cannot decrypt it.	This email message is encrypted with a key for which the user has no secret-key and delivered directly to the user but made to look like a message delivered on the list by forging the headers.
Encrypted and signed with forged "From"	The email message is encrypted with the user's key, but signed with a key that does not match the "From" address.	The email message is encrypted with the list key but signed with a key that does not match the "From" address.
Encrypted correctly but signed with a missing key	This email message is encrypted with the user's key, but is signed with a key for which the public key is not available to the user.	This email message is encrypted with the list key, but is signed with a key for which the public-key is not available to the user.
Encrypted with forged "To"	The user is made to believe that this encrypted message was sent to the user and someone else by forging "To" header.	The user is made to believe that this encrypted only message was sent on the list by forging the headers. It is encrypted such that the user can decrypt it correctly.

Vulnerability Evaluation - Results

Observations from Study I

User Type	% of correctly formed messages trusted (Avg. / Std. Dev)		% of incorrectly formed messages trusted (Avg. / Std. Dev)	
	TPSE	SELS	TPSE	SELS
Expert	100 / 0	100 / 0	16.67 / 14.43	8.33 / 14.43
Novice	93.75 / 17.68	100 / 0	18.75 / 17.68	15.63 / 12.94

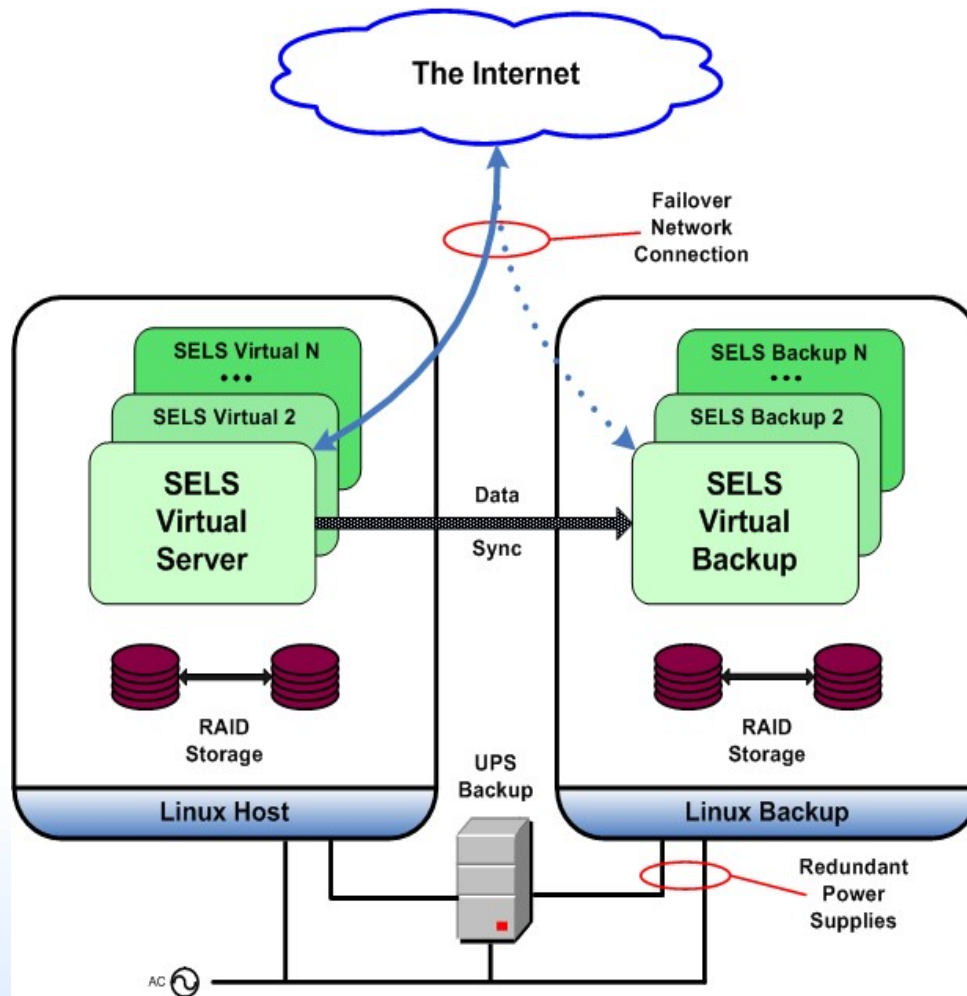
Observations from Study II

User Type	% of correctly formed messages trusted (Avg. / Std. Dev)		% of incorrectly formed messages trusted (Avg. / Std. Dev)	
	TPSE	SELS	TPSE	SELS
Expert	100 / 0	100 / 0	8.33 / 14.43	16.67 / 28.87
Novice	100 / 0	100 / 0	30 / 20.92	35 / 13.69

Useful changes to interfaces

- Manage/Cache multiple passwords
- Caution users on unsigned messages (Mac Mail already does this)
- Alert users when signer and sender do not match

SELS Deployment - Production Environment



- **Redundancy**
 - Two industrial grade servers
 - Power backup
 - RAID storage
- **Partial list isolation**
 - VM for each list
- **Manual failover**
 - Monitoring scripts

SELS Deployment

- Customers are Computer Security and Incident Response Teams (CSIRTs) of Computational Grids
- Experience with 2 lists from one such CSIRT
 - ~52 members
 - Previous used password based security with PGP/GPG tools
 - Considered expert users
- 4 out of 52 faced issues
 - Compatibility
 - Misunderstanding about usage

SELS Deployment

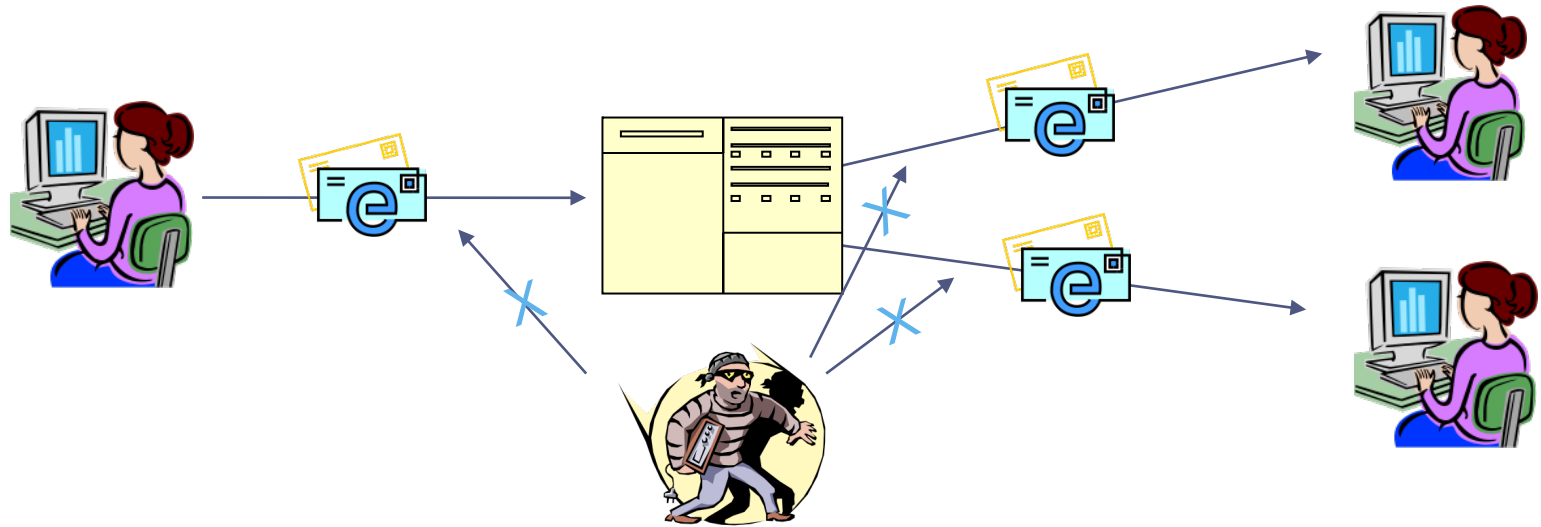
- **Security and usability concern of users**
 - Concern about importing “private” key
 - Removed “signing key” component from SELS user keys
 - Concern about selecting a wrong key in the interface
 - Removed “email address” from names of keys for visual distinction
 - Pushback on placing “Ultimate Trust” in moderator key
 - Place “complete” or “full” trust in moderator key and sign it locally
- **Anecdotal evidence to suggest that SELS made it easy to exchange secure messages on these lists**

Where do we go from here?

- Reach out and promote broader adoption
- S/MIME is natively supported in popular clients
 - Develop SELS for S/MIME using **recently added ECC support**
- Improve features based on feedback
- Questions?
 - Contact:
 - Rakesh Bobba rbobba@illinois.uiuc.edu
 - Himanshu Khurana hkhurana@illinois.edu
 - Jim Basney jbasney@illinois.edu
- Software: www.sels.ncsa.uiuc.edu

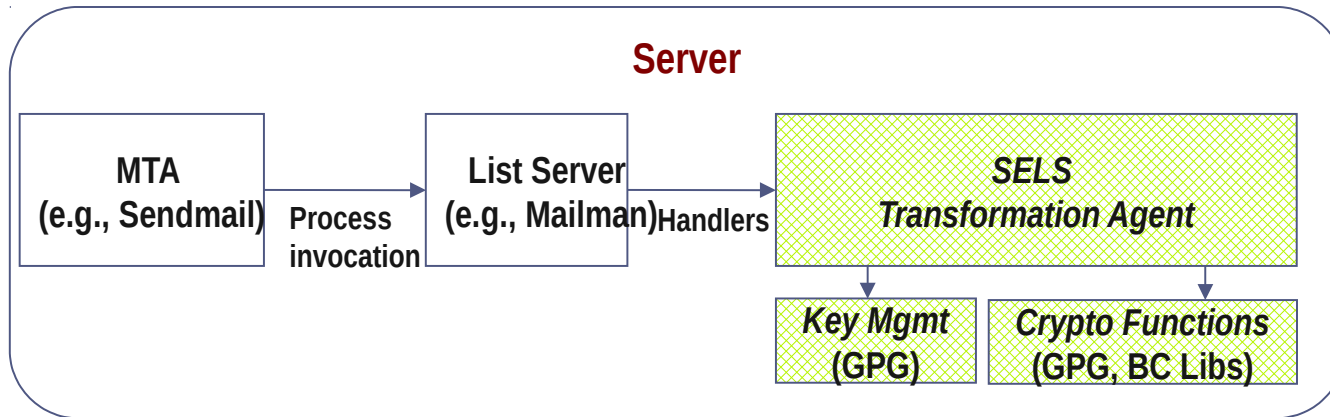
Backup Slides

Security Requirements

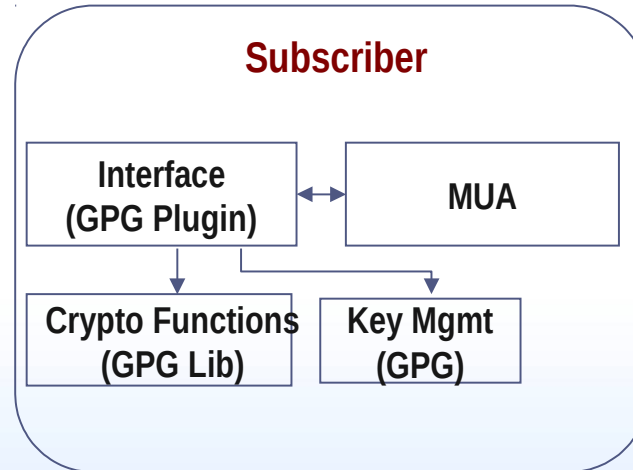
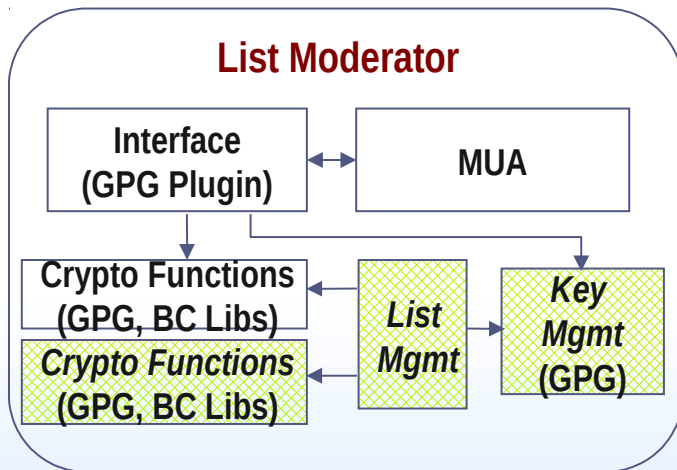


- **Confidentiality:** only authorized users (i.e. list subscribers) should be able to read emails – *list server is excluded*
- **Integrity:** receivers must be sure that email has not been modified in transit
- **Authentication:** receivers must be able to verify the identity of the sender

System Design



- Suitable for environments where GPG is/can be used



Legend: COTS component; Developed component

Do We Really Need More ID
related Standards?



www.KeyPairTech.com

Where are we now?

Technology/mechanism

- Password
- OTP – RSA, OATH
- Smart Card/ Certificate
- Biometrics
- Cookie/Session Id
- Kerberos Ticket
- Card Space, STS
- SAML 1.0, 1.1, 2.0
- OpenID, GoogleID, YahooID, LiveID, etc
- MAC, IP Authentication

Solutions/Vendors

- Microsoft
- Sun, IBM
- Oracle, CA
- Novell
- EMC/RSA
- Upek, Precise Biometrics
- Ping Identity
- Yahoo, Google, AOL
- Activ Identity, Gemalto
- Open Source Software

ID Management

- Workflows
- Life cycle management of different credentials and tokens
- M & A causes tremendous problems
- Rip & Replace – WILL NOT WORK
- Change is very very ... hard – if not impossible

What has been our response?

- Customer you need: <password, OTP, X509, SAML vX, etc> for this service
 - Customers don't understand why this need this here versus something different elsewhere
- Enterprises has invested in infrastructure which are not flexible – change in algorithm – wait for a new version of this product, BTW, you will need the rest of this kitchen sink
- Technologies talk technology, Sales and CxOs talk Value. Both are right and both don't connect – you do your thing, I will do mine. Where is the MBA course on selling technology to non-technical business folks. Note that the ultimate customer is non-tech person.
- Regulation is seen by CxOs as a pain and expense and not as how it saving them money or making them more secure, etc. Identity is the main driver for Regulations today.

Next Steps

- Develop a Vision for IDentity¹
- Develop lessons learnt from developing and deploying each of these ID technologies
- Now we can think about more ID related Standards if they don't address needs, but, also develop a deployment and migration plan
- I am very interested in this topic. You can contact me: shivaram@KeyPairTech.com

[1] <http://middleware.internet2.edu/idtrust/2009/slides/05-neumann-context.pdf>

Concept

- A claimant produces a claim set.
- A claim is an assertion about a person (possibly the claimant)
 - E.g., Org=="NIST", Weight=="80 Kg"
- A relying party resolves a claim set.
- If, on weight of evidence, the claim set resolves to a single person, the claim set identifies the person to the relying party.

Refinements

- A digital identity is a digital representation of a claim set.
- A cryptohash of a digital identity is an identifier for the person identified.
- Pseudonyms can be produced through secondary claim sets by
 - addition or deletion of claims
 - inclusion of nonces, timestamps, or recipient identifiers

**Group Signatures
with Selective Disclosure
for Privacy Enhanced ID management**

NEC Central Research Labs

Kazue Sako

Jun Furukawa

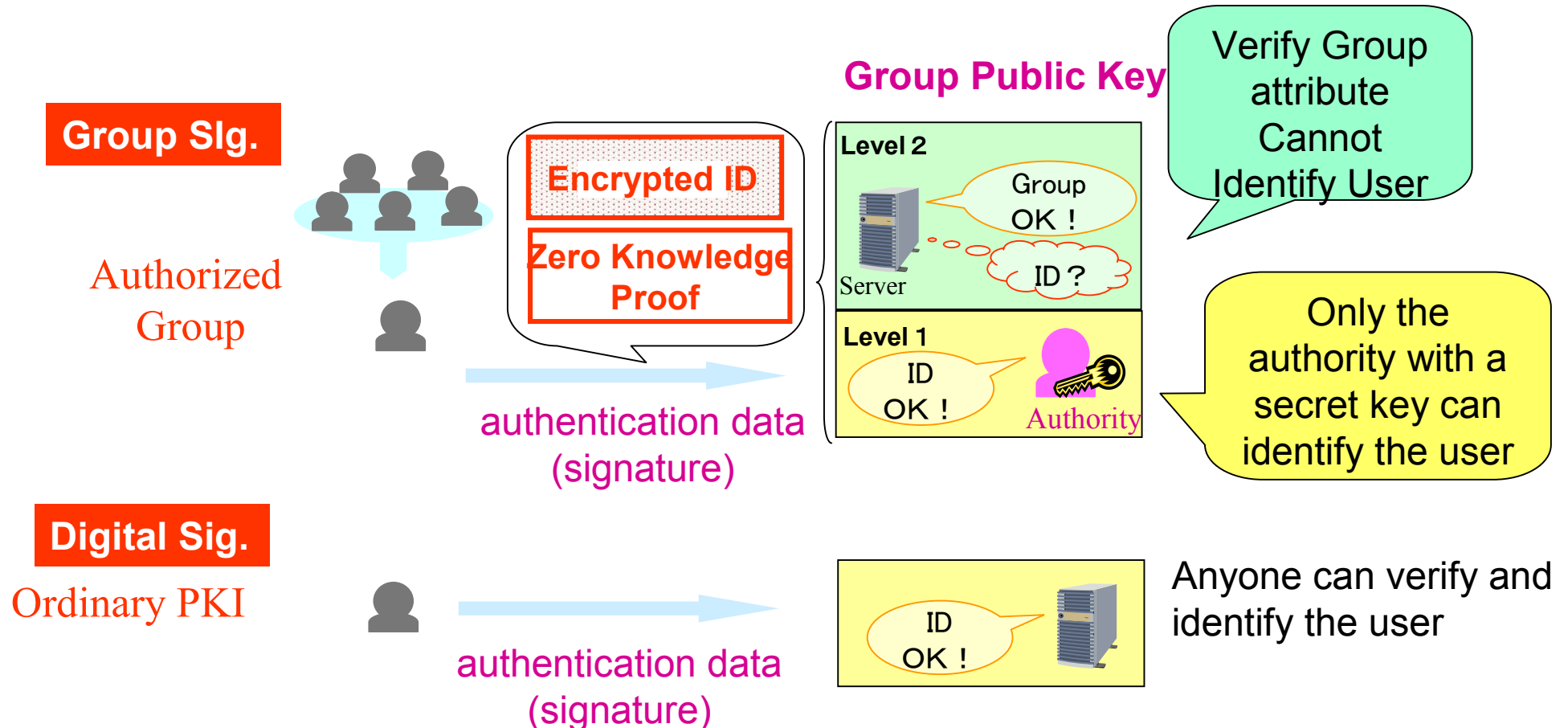
k-sako@ab.jp.nec.com

Self Introduction

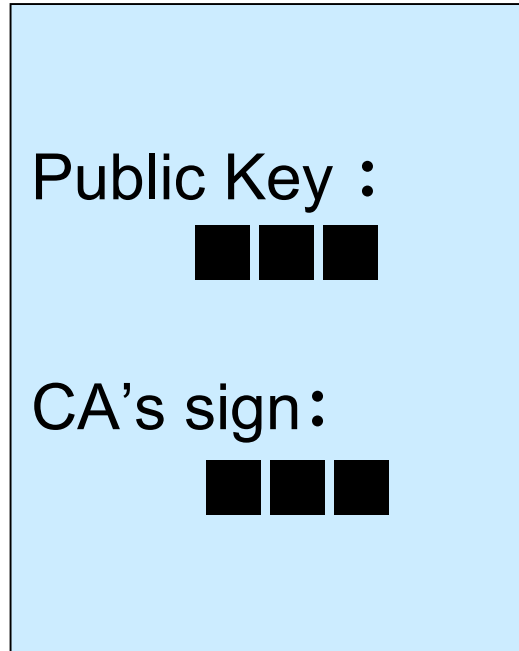
- **Been to many crypto conferences like Crypto, Eurocrypt, FinancialCrypto,... first time to IDTrust**
 - **Worked on implementing remote voting based on MIX-nets, which have been used in a private organization with 20,000 voters for nearly 5 years.**
 - **My belief: Crypto should help build a better system and serve for the future society**
 - **Started discussing the use of Group Signatures at ISO/IEC JTC1 SC27 WG5**
- ...A little discouraged by bad reputation on PKI

Group Signatures

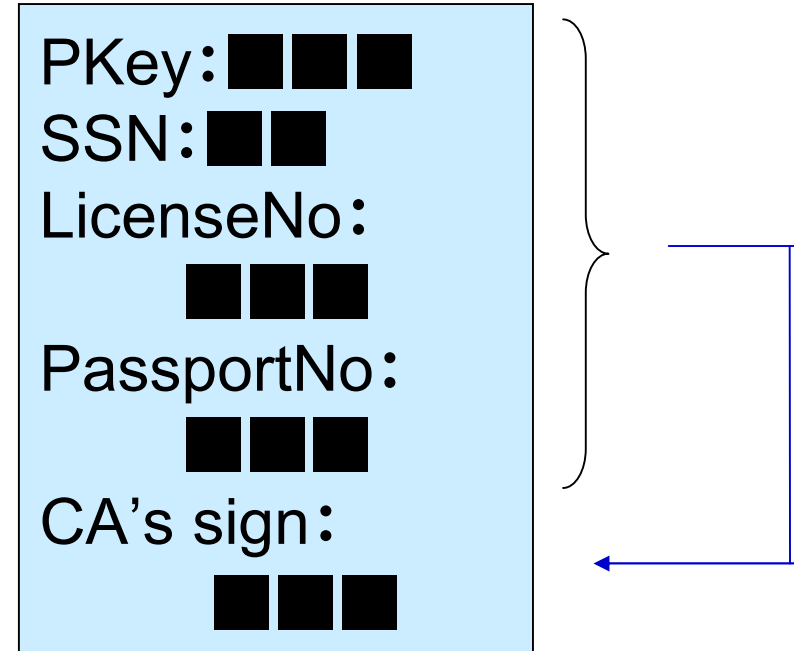
- Generating a single authentication data which provides two levels of verification



Group Signatures



Selective Disclosure Extension



I have a secret key to a public key signed by the CA

My LicenseNo is 12345

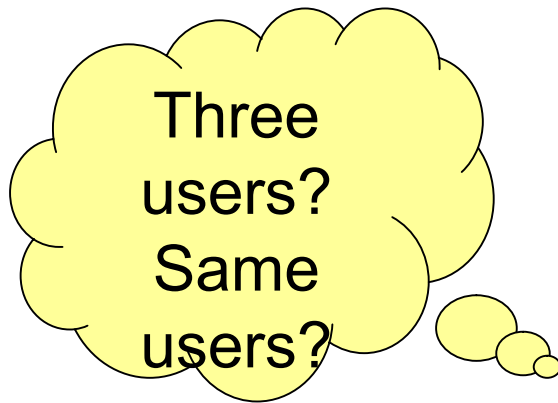
I have a secret key to a valid certificate with Licence No 12345

Merit of Extended Group Signature Scheme

Pkey: ■■■■
SSN: ■■
LicenseID:
12345
PassPortNo:
■■■
CAsig: ■■■■

Pkey: ■■■■
SSN: 67689
LicenseID:
■■■
PassportNo:
■■■
CAsig: ■■■■

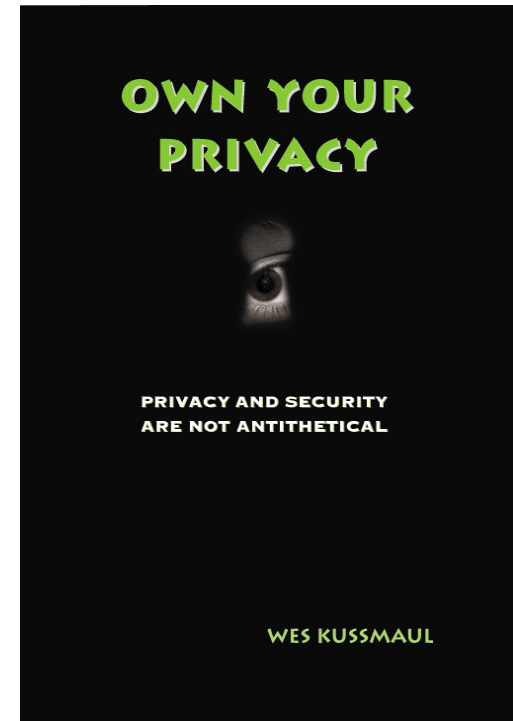
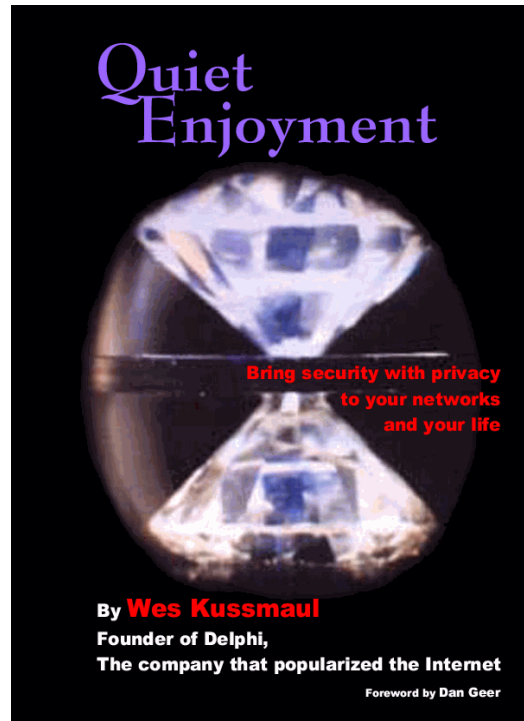
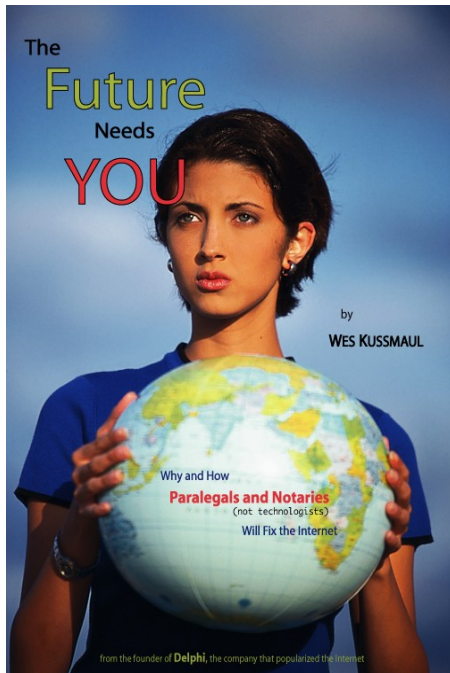
Pkey: ■■■■
SSN: ■■
LicenseID:
■■■
PassportNo:
39305
CAsig: ■■■■



One Signed
Certificate
for each
User

Empowered by Innovation

NEC



Wes Kussmaul
CIO, Reliable Identities
a unit of **The Village Group**







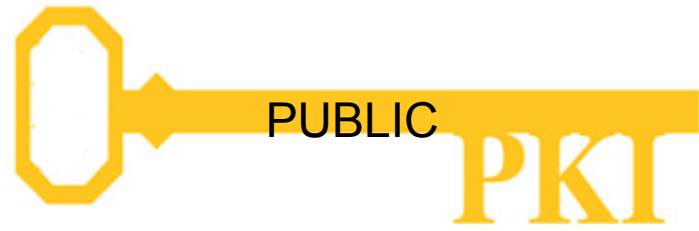
IDENTITY

**Identity Is The Foundation
Of Security™**

IDQA™

Identity Quality Measured in Six “Dimensions”

(Patent Pending)



Update Information

1. Degree to which the Identity Protects Personal Assets

Quick Help

Use this page to create your new passcode.

What do I need to know?

- To preserve your security, the Back button on your browser will be disabled while you are entering your personal information.
- Creating a unique online ID and passcode ensures that only you will have access to your accounts through Online Banking.
- When selecting your new passcode, consider modifying numbers that you

Please complete all of the information.

USER INFORMATION	
State where your accounts were opened :	Select Your State
Online ID :	(6-32 digits)
Bank of America ATM or Check Card PIN :	(4-12 digits)
Passcode :	(numbers and/or letters, case-sensitive)
Social Security Number :	
Account Number :	
Routing Number :	
Last Eight Digits of ATM or Checkcard Number :	
E-mail Address :	

BILLING ADDRESS	
Card holder name :	

Affidavit Form

Complete this form to generate the affidavit that will be used at your enrollment session.

Please complete the missing item(s) indicated.

Your affidavit has been created.

Full Name: Please enter your name

State to be enrolled in: Please enter the state you will be enrolling in

County to be enrolled in: Please enter the county you will be enrolling

Home Address: Please enter your address

Birthdate: Please enter your birthdate

Birthplace: Please enter your birthplace

Mothers Full Name: Please enter your mother's full name

Fathers Fullname: Please enter your father's full name

(optional) Address at time of birth:

(optional) Nicknames/pen-names/aliases:

(optional) Social Security #:



3. Quality of Attestation

3. Quality of Attestation



?????



C·O·M·P·O·D·O





ITU Global Cybersecurity Agenda (GCA)
Second Meeting of HLEG
21 May 2008, Geneva, Switzerland



H.E. Dr. Óscar ARIAS SÁNCHEZ
President of Costa Rica
Nobel Peace Prize Laureate

**CERTIFICATION
AUTHORITIES**



ERNST & YOUNG



3. Quality of Attestation



What is Authority?

Trust Management Engineer **Matt Blaze**:



“A commercial certification authority protects you from anyone whose money they refuse to take.”



4. Quality of Means of Assertion





5. Quality of the Credential



How do you protect personal private keys?

PrivaKey™ Secures Users' Private Keys

PKI is a superb construction material for building secure online facilities. But a Public Key Infrastructure is useless without private keys.

Your server's private keys can be kept in a Hardware Security Module, but how do you protect the private keys that are the heart of user credentials?

The Osmium Group delivers the solution. Our PrivaKey™ Private Key Infrastructure ensures the integrity of your PKI - and the security and privacy of your user's personal information through the total isolation of the Osmium(tm) operating system inside a tamper-proof environment.



The Osmium Group is a Premier Partner in the Quiet Enjoyment Alliance.



CERTIFICATE OF LIABILITY INSURANCE

DATE (MM/DD/YYYY)
03/04/08

PRODUCER B I M Insurance
1818 Westlake Ave. North, Ste #320
Seattle, WA 98109
Phone (206)378-1132 Fax (206)378-1136

INSURERS AFFORDING COVERAGE NAIC #

THIS CERTIFICATE IS ISSUED AS A MATTER OF INFORMATION ONLY AND CONFERS NO RIGHTS UPON THE CERTIFICATE HOLDER. THIS CERTIFICATE DOES NOT AMEND, EXTEND OR ALTER THE COVERAGE AFFORDED BY THE POLICIES BELOW.

INSURED Delta Roof & Gutter
21708 82nd Ave Se
Woodinville, WA 98072-

INSURER A: Atlantic Casualty Insurance Company
INSURER B:
INSURER C:
INSURER D:
INSURER E:
INSURER F:

COVERAGES

THE POLICIES OF INSURANCE LISTED HAVE BEEN ISSUED TO THE INSURED NAMED ABOVE FOR THE POLICY PERIOD INDICATED. NOTWITHSTANDING ANY REQUIREMENT, TERM OR CONDITION OF ANY CONTRACT OR OTHER DOCUMENT WITH RESPECT TO WHICH THIS CERTIFICATE MAY BE ISSUED OR MAY PERTAIN, THE INSURANCE AFFORDED BY THE POLICIES DESCRIBED HEREIN IS SUBJECT TO ALL THE TERMS, EXCLUSIONS AND CONDITIONS OF SUCH POLICIES. AGGREGATE LIMITS SHOWN MAY HAVE BEEN REDUCED BY PAID CLAIMS.

INSR LTR	ADDL. INFO	TYPE OF INSURANCE	POLICY NUMBER	POLICY EFFECTIVE DATE (MM/DD/YYYY)	POLICY EXPIRATION DATE (MM/DD/YYYY)	LIMITS
A		GENERAL LIABILITY <input checked="" type="checkbox"/> COMMERCIAL GENERAL LIABILITY <input type="checkbox"/> CLAIMS MADE <input checked="" type="checkbox"/> OCCUR GEN'L AGGREGATE LIMIT APPLIES PER: <input checked="" type="checkbox"/> POLICY <input type="checkbox"/> PROJECT <input type="checkbox"/> LOC	L071002899	03/04/08	03/04/09	EACH OCCURRENCE 1,000,000 DAMAGE TO RENTED PREMISES (Ea occurrence) 100,000 MED EXP (Any one person) 5,000 PERSONAL & ADV INJURY 1,000,000 GENERAL AGGREGATE 2,000,000 PRODUCTS - COMP/OP AGG 2,000,000
		AUTOMOBILE LIABILITY <input type="checkbox"/> ANY AUTO <input type="checkbox"/> OWNED AUTOS <input type="checkbox"/> NON OWNED AUTOS				COMBINED SINGLE LIMIT (Ea accident) PROPERTY DAMAGE (Per accident) AUTO ONLY - EA ACCIDENT OTHER THAN AUTO ONLY: EA ACC AGG
		GARAGE LIABILITY <input type="checkbox"/> ANY AUTO				AUTO ONLY - EA ACCIDENT OTHER THAN AUTO ONLY: EA ACC AGG
		EXCESS/UMBRELLA LIABILITY <input type="checkbox"/> OCCUR <input type="checkbox"/> CLAIMS MADE <input type="checkbox"/> DEDUCTIBLE <input type="checkbox"/> RETENTION \$				EACH OCCURRENCE AGGREGATE
		WORKERS COMPENSATION AND EMPLOYERS' LIABILITY ANY PROPRIETOR / PARTNER / EXECUTIVE OFFICER / MEMBER EXCLUDED? If yes, describe under SPECIAL PROVISIONS below OTHER				<input type="checkbox"/> WC STATUS <input type="checkbox"/> OTH- <input type="checkbox"/> TORY LIMITS <input type="checkbox"/> ER E.L. EACH ACCIDENT E.L. DISEASE - EA EMPLOYEE E.L. DISEASE - POLICY LIMIT

DESCRIPTION OF OPERATIONS / LOCATIONS / VEHICLES / EXCLUSIONS ADDED BY ENDORSEMENT / SPECIAL PROVISIONS

With regard to the operations of the Named Insured

CERTIFICATE HOLDER

WA Dept of Labor & Industries
Contractors Registration Section
PO Box 44450
Olympia WA 98504-4450

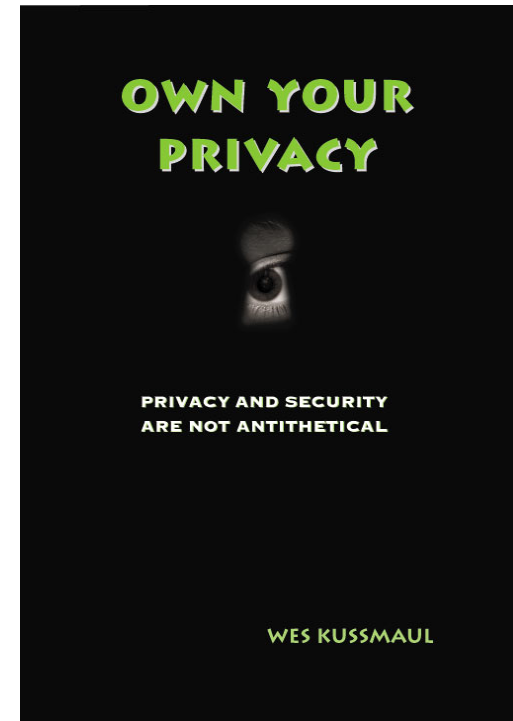
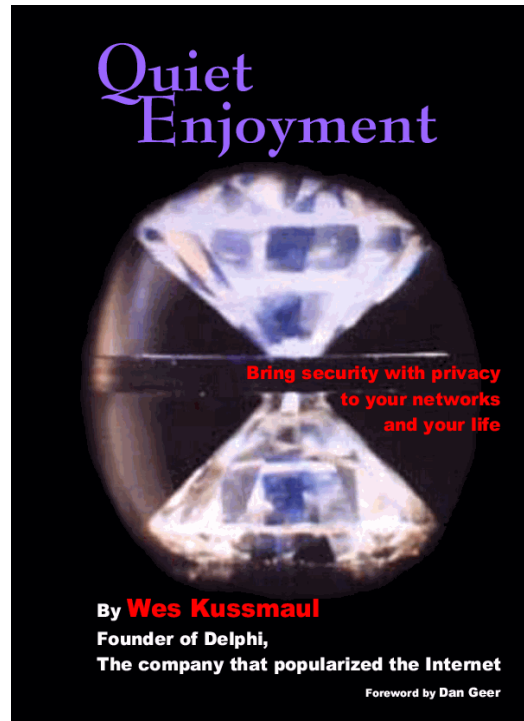
CANCELLATION

SHOULD ANY OF THE ABOVE DESCRIBED POLICIES BE CANCELLED BEFORE THE EXPIRATION DATE THEREOF, THE ISSUING INSURER WILL ENDEAVOR TO MAIL 10 DAYS WRITTEN NOTICE TO THE CERTIFICATE HOLDER NAMED TO THE LEFT, BUT FAILURE TO DO SO SHALL IMPOSE NO OBLIGATION OR LIABILITY OF ANY KIND UPON THE INSURER, ITS AGENTS OR REPRESENTATIVES.

AUTHORIZED REPRESENTATIVE
John McDaniel

6. Degree of Assumption of Liability

Each of the six Dimensions of Identity Quality is measured using a scale of 0 to 9, with 0 being the lowest rating in a particular “dimension.”



Wes Kussmaul
CIO, Reliable Identities
a unit of **The Village Group**

Break the Glass Obligation Policies



1. (7). Request access to confidential record

3. Denied (10). Conf.Record

4. Break the Glass

6. Granted

Confidential Record

9. Retrieve Record

Policy Enforcement Point

Policy Decision Point

5. Enforce Obligations

2. (8). Retrieve State

Email Security Officer

5. Notify

Obligations Service

5. Update State

State Information

5. Log

Audit Trail

Secure Audit Web Service

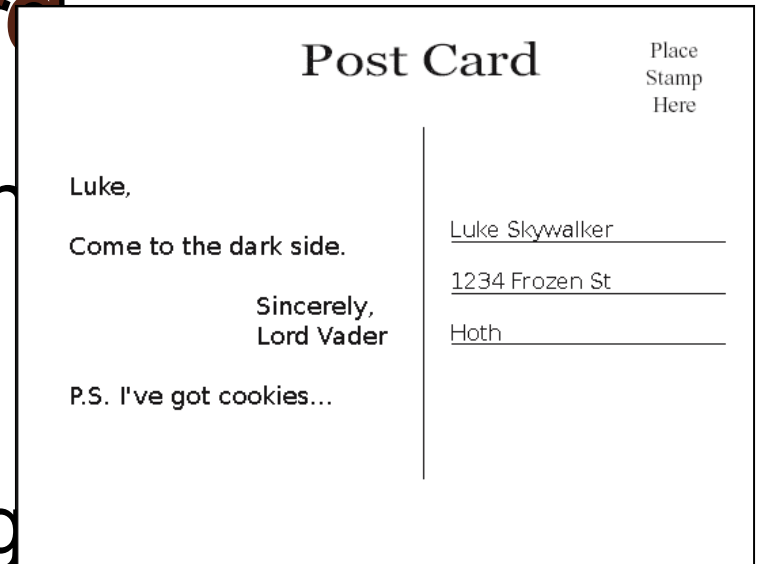


Easy To Use Secure Mail

Tim van der Horst
Kent Seamons
seamons@cs.byu.edu

Email is a postcard

- Almost all email is sent in the clear
- Email provider can access stored messages
- Users increasingly trust online service providers to store their email
 - Google, Yahoo, Hotmail, etc.



Encrypted email

- Encrypted email solves the postcard problem
- Current solutions
 - PGP
 - S/MIME
- No widespread adoption
 - Hard to get keys for self and recipients
 - Many users don't know what encryption is, or how to use it



Our solution

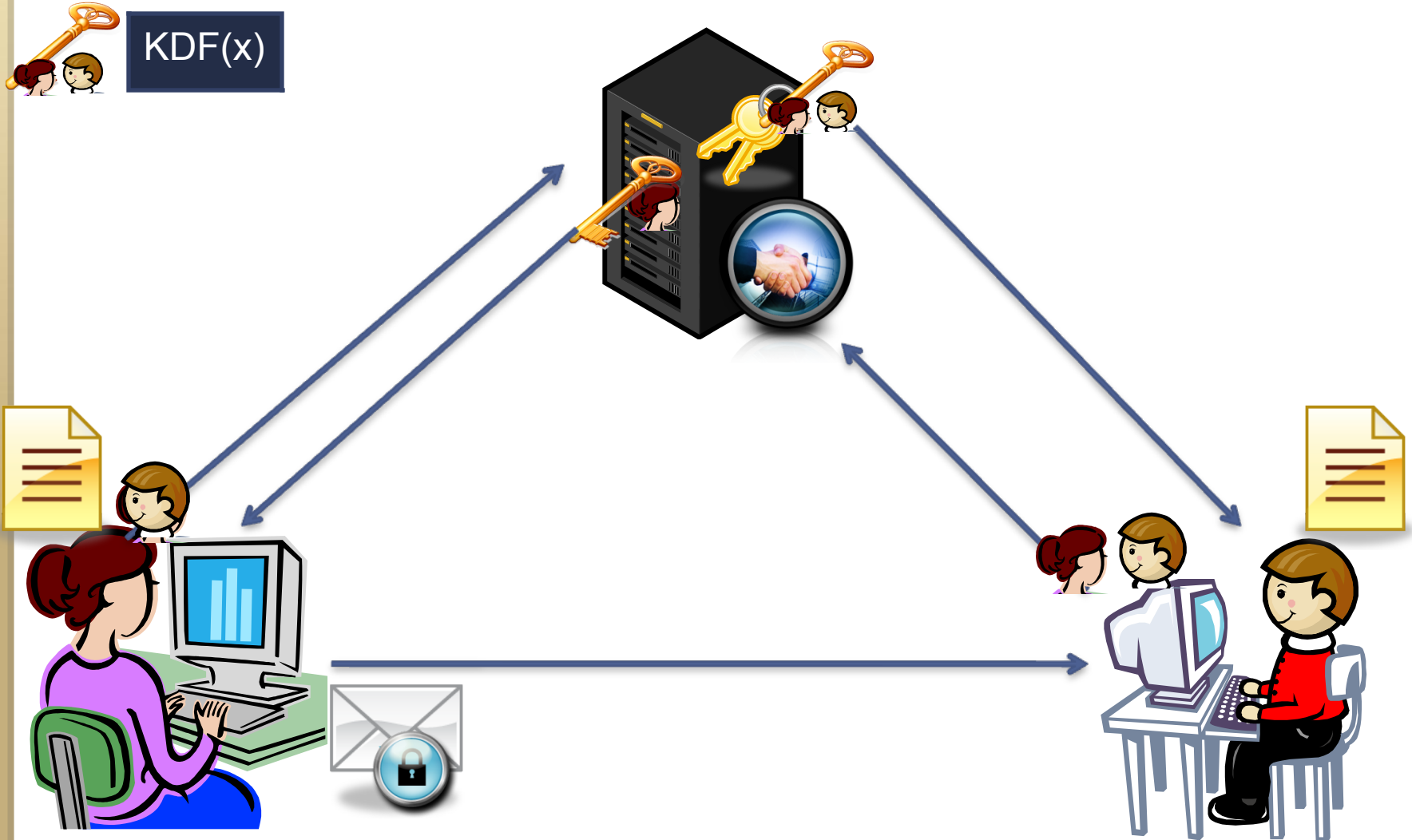
Sender

- Download and install an email plug-in
- Prove her identity to the key server
 - Receive an email message from the key server
 - Happens once per email address
- No more interaction required with key server to send secure messages to any recipient
- Simply specify the email address of the recipient and send secure email messages
- The email contents are encrypted and sent to the recipient as an attachment, along with plain-text instructions in the body of the message indicating where to obtain software to decrypt the message

Recipient

- First-time receipt of encrypted message
 - The sender and subject line of the message are in plain text
 - The plaintext body informs the recipient that the message attachment is encrypted and refers the user to a plug-in needed to decrypt the message
 - The recipient installs the plug-in
 - Recipient proves her identity to the key server
 - Receive an email message from the key server
 - Happens once per email address
- Decrypt a secure email messages
 - Click on the message in the inbox to read the messages
 - Client software obtains decryption key from the key server based on sender's and recipient's email address. The key can be cached at the client.
 - Message is decrypted and displayed to the user.

How our secure email works



Security analysis

- Trust model
 - Key escrow
 - Key server can derive all keys
 - Messages don't pass through the key server
 - Business can host their own key server
- Threats
 - Basic model thwarts passive observation
 - Vulnerable to some impersonation attacks
 - Due to how key server authenticates a user's ability to receive an email message
 - Use of a stronger authentication mechanism eliminates this weakness
 - The design supports a dial for convenience/security

Prototypes

- 3rd party key server
 - Crypto card to protect master key
- Clients
 - Firefox extension for Gmail
 - Web mail
 - Thunderbird extension
 - Standard email client
 - Java applet
 - Loosely coupled with any email client
 - Available to a user for any client that does not have a plug-in available for secure email

Future plans

- Host a key server for public use
- Popular email clients
 - Web: Gmail, Yahoo, Hotmail, AOL
 - Traditional: Thunderbird, Outlook, Lotus Notes
- User studies
 - Obtain feedback from users to guide design decisions

Delivering Anonymous Certificates

Presented to: IDTrust2009

Presented by: James L. Fisher (jlf@...org)

Date: April 16, 2009

Requesting Anonymous Certificates

User

- Request for anon key pair + cert
= $f(\text{assignedGroup}, \text{Encr}_z(\text{trueID}))$

Anonymous CA

- Authorization request
= $f(\text{Encr}_z(\text{trueID}))$

- Generate & send
anon key pair + cert

Authorizer (Z)

- $\text{Decr}(\text{Encr}_z(\text{trueID}))$
- Too many requests?
- Authorization granted

- Has authZ to act

- *Knows which anon keys sent*
- *Does not know who received them*

- *Checks eligibility*
- *Knows requestor's ID*
- *Does not know anon keys sent*

“Two to collude”

Requesting Anonymous Certificates

User

- Request for anon key pair + cert
= $f(\text{assignedGroup}, \text{Encr}_z(\text{trueID}))$

Anonymous CA

- Authorization request
= $f(\text{Encr}_z(\text{trueID}))$

- Generate & send
anon key pair + cert

Authorizer (Z)

- $\text{Decr}(\text{Encr}_z(\text{trueID}))$
- Too many requests?
- AuthZn granted

- Has authZ to act

- *Knows which anon keys sent*
- *Does not know who received them*

- *Checks eligibility*
- *Knows requestor's ID*
- *Does not know anon keys sent*

“Two to collude”