A Principled Approach Aiding the Development of a Compliant Internet PKI

OMAR HAIDER CHOWDHURY, UNIVERSITY OF IOWA

Cybersecurity Research Acceleration Workshop and Showcase, Indianapolis, IN
October 11, 2017

The University of Iowa

Towards a Compliant Internet Public-Key Infrastructure (PKI)

Omar Haider Chowdhury

Assistant Professor of Computer Science

Cybersecurity Research Acceleration Workshop and Showcase

October 11, 2017 | Indianapolis, IN

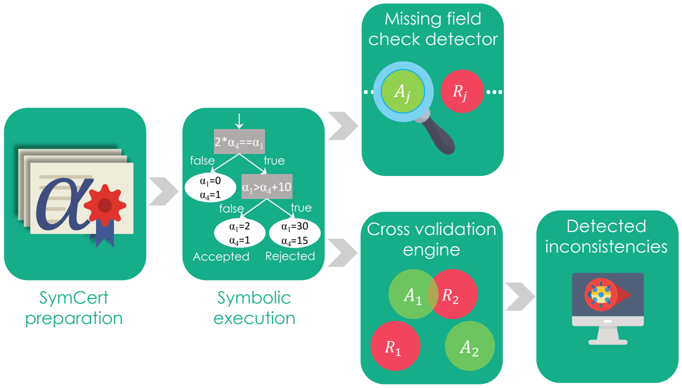
Quad Chart for: A Principled Approach Aiding the Development of a Compliant Internet PKI

Challenge:

- Complex structure of X.509 PKI certificate
- Cryptographic libraries
- Code/logic coverage:
- Standard specification

Solution:

- Employing Symbolic Execution (SE)
- Mitigating path explosion in SE
  - Using specially crafted input
  - Bypassing crypto. functions
- Extracting the certificate input universe
- Partitioning the universe to:
  - Accepting universe
  - Rejecting universe
- Launching differential testing



NSF CRII SaTC #1657124
The University of Iowa
PI: Omar Haider Chowdhury,
Assistant Professor of Computer Science

Value proposition:

- Fully leverage the open source nature of libraries
- Enabling users to find more in-depth bugs

What we need to TTP

- Automated instrumentation
- Analysis engines
- Your input

Contact

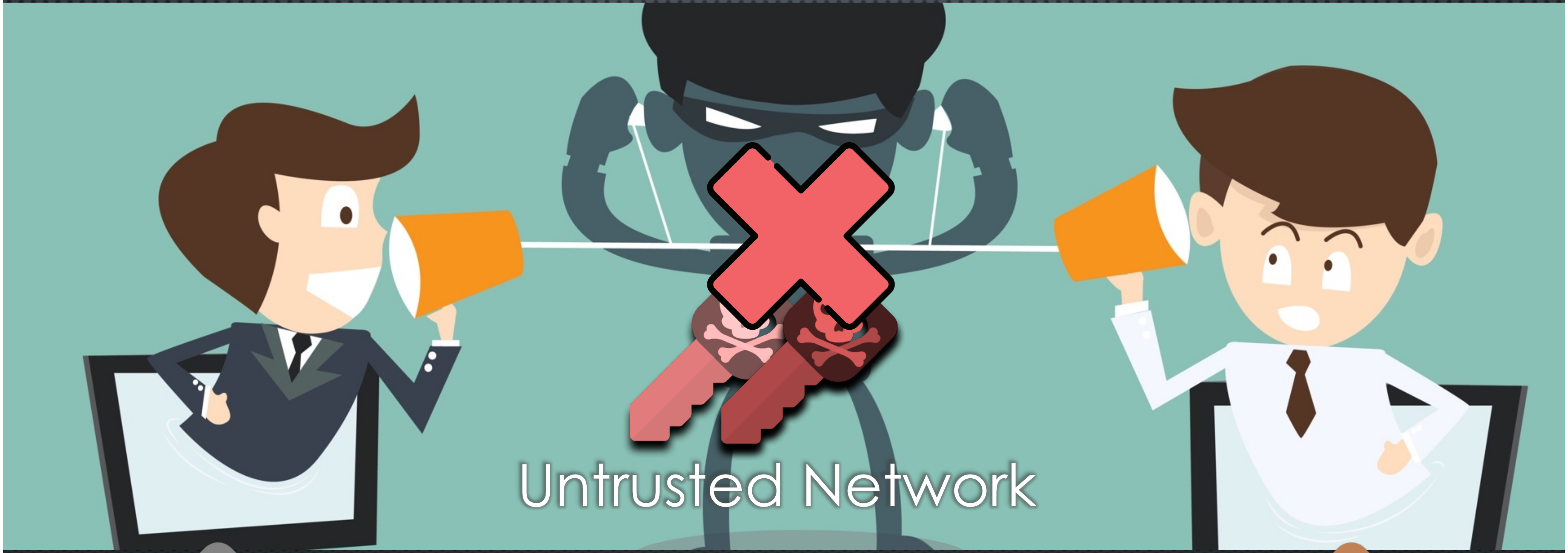
- omar-chowdhury@uiowa.edu

Untrusted Network

Problem?

Public key cryptography

Secure distribution of public key



Problem?

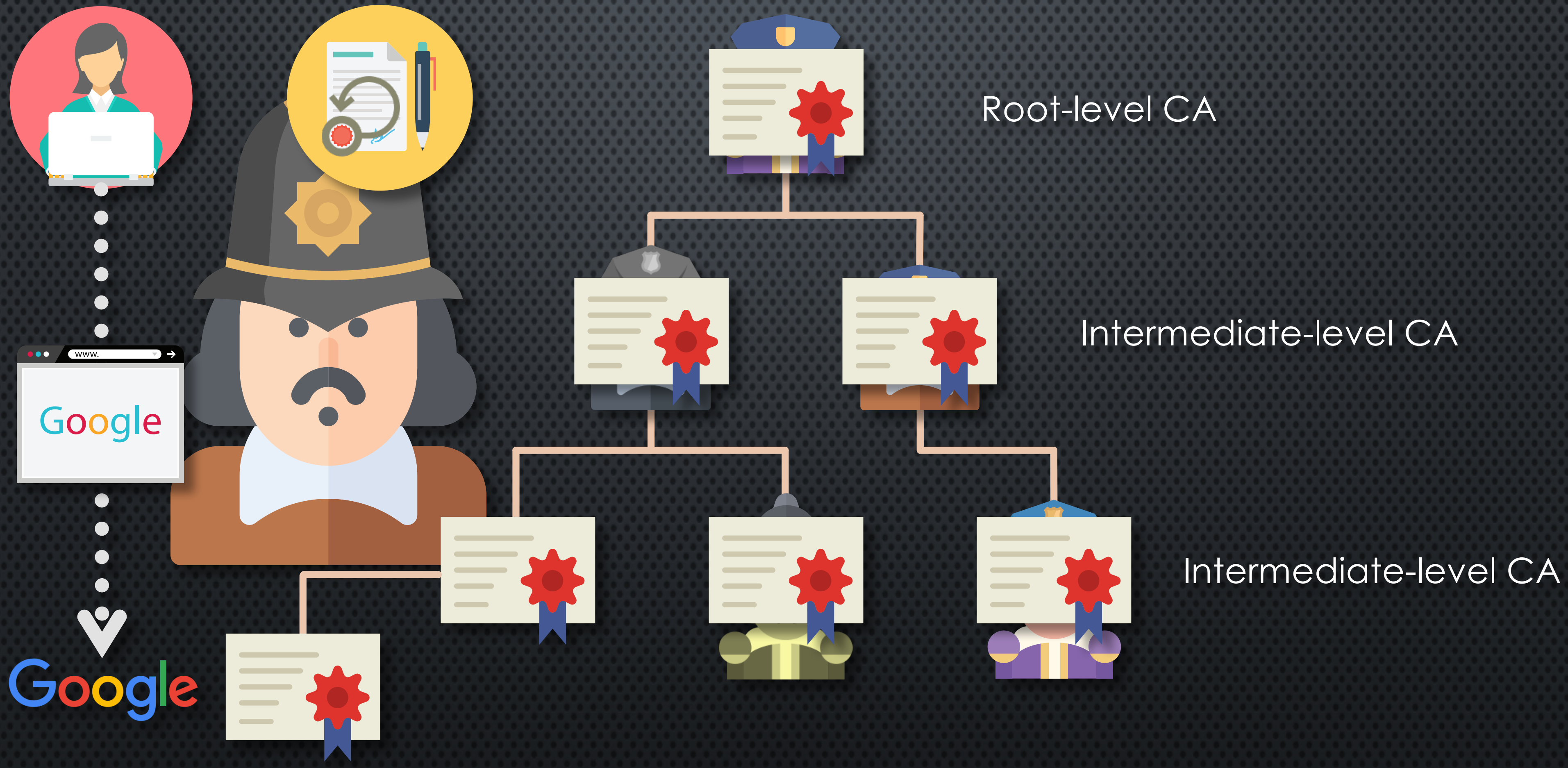
Public key cryptography

Secure distribution of public key

X.509 Public Key Infrastructure (PKI) Protocol

Now, how can we obtain the CA’s **public key**?

X.509 Public Key Infrastructure (PKI) Protocol



X.509 Usage

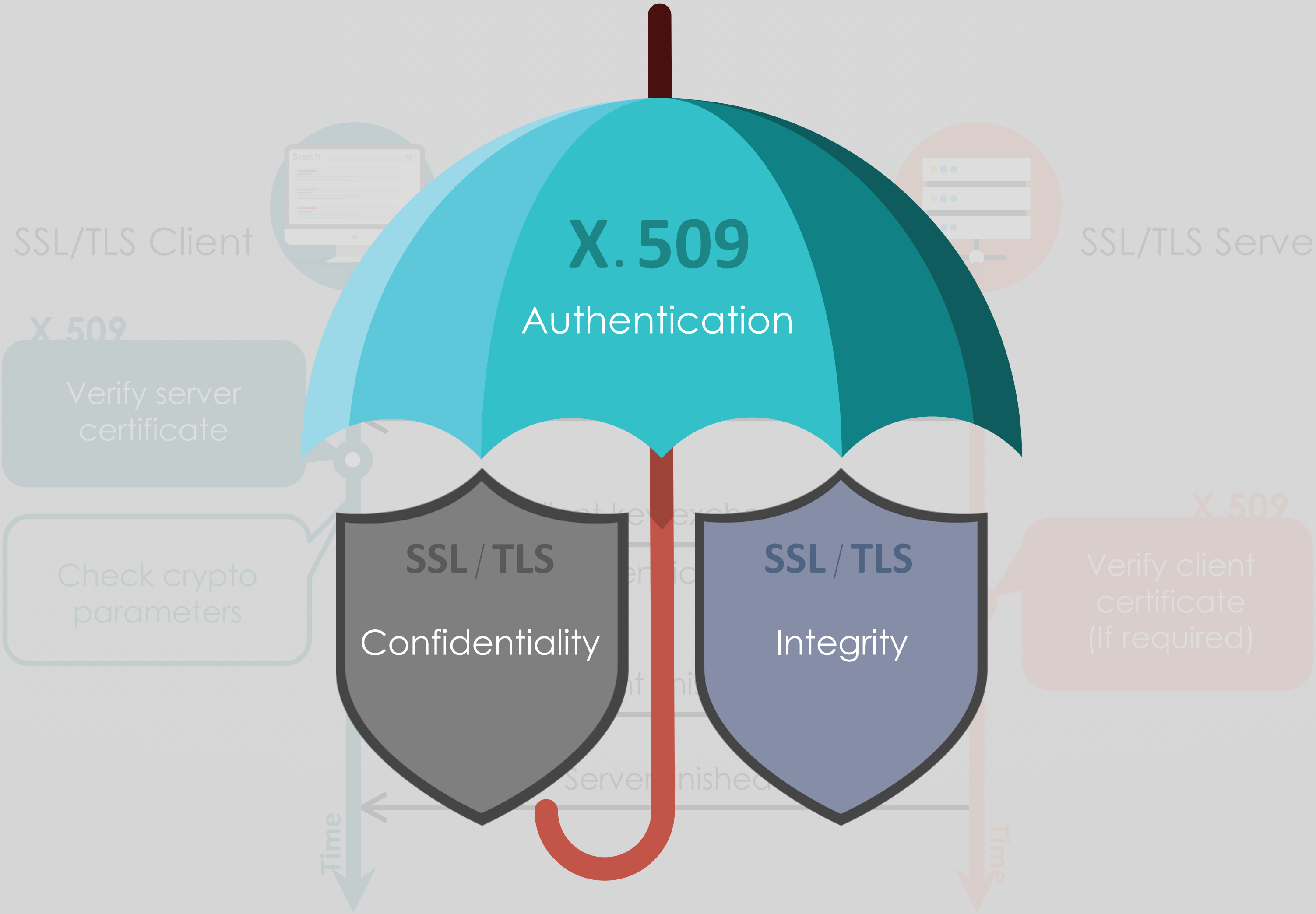
EAP-TLS
S/MIME
Code Signing
SSL/TLS
IPSec
...
NETFLIX
YouTube

37%

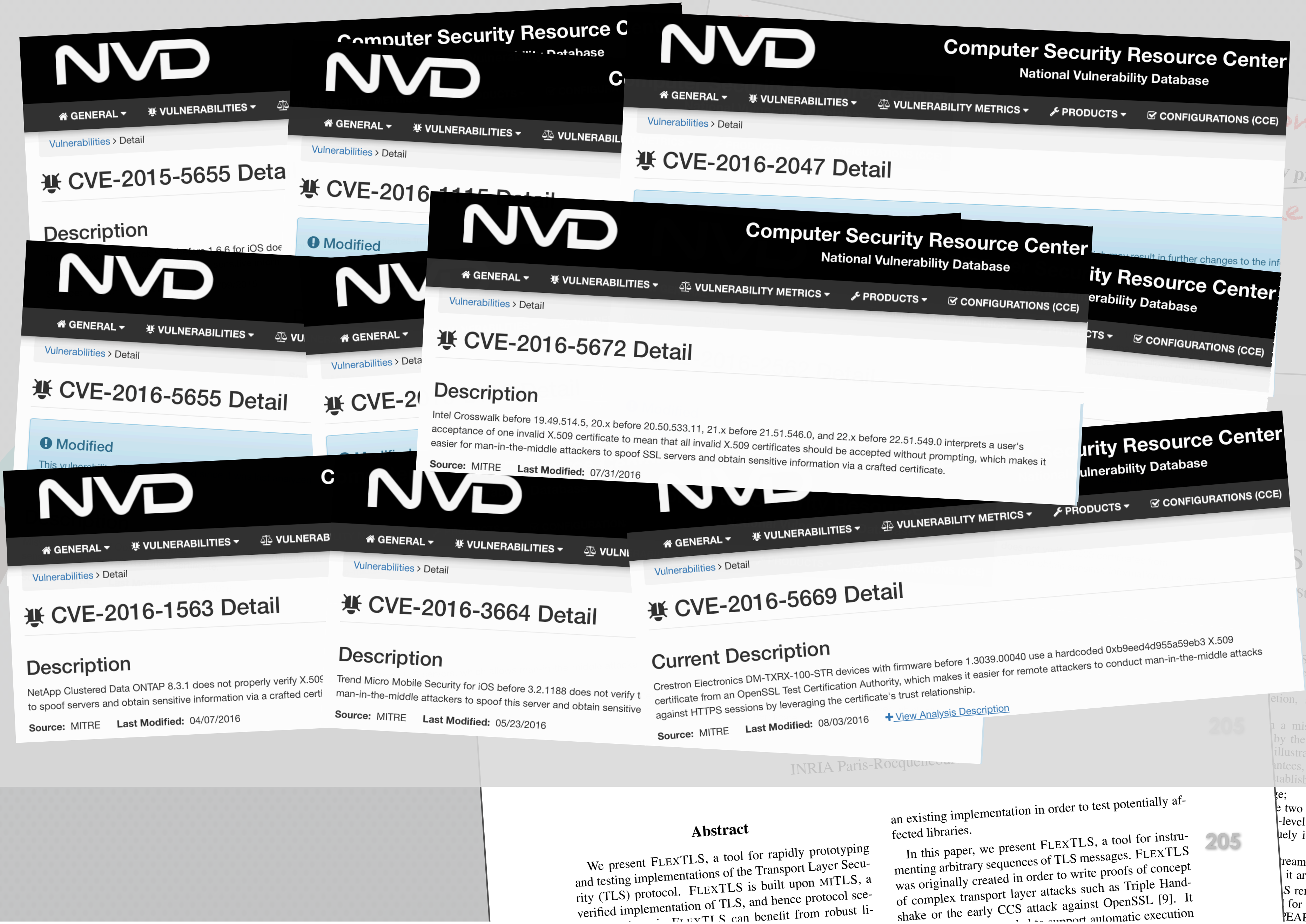
66%

18%

Role of X.509 in SSL/TLS



SSL/TLS Verification



CVE-2015-5655 Detail

CVE-2016-2047 Detail

CVE-2016-5655 Detail

CVE-2016-5672 Detail

Description

Intel Crosswalk before 19.49.514.5, 20.x before 20.50.533.11, 21.x before 21.51.546.0, and 22.x before 22.51.549.0 interprets a user's acceptance of one invalid X.509 certificate to mean that all invalid X.509 certificates should be accepted without prompting, which makes it easier for man-in-the-middle attackers to spoof SSL servers and obtain sensitive information via a crafted certificate.

Source: MITRE Last Modified: 07/31/2016

CVE-2016-1563 Detail

Description

NetApp Clustered Data ONTAP 8.3.1 does not properly verify X.509 to spoof servers and obtain sensitive information via a crafted certi

Source: MITRE Last Modified: 04/07/2016

CVE-2016-3664 Detail

Description

Trend Micro Mobile Security for iOS before 3.2.1188 does not verify t man-in-the-middle attackers to spoof this server and obtain sensitive

Source: MITRE Last Modified: 05/23/2016

CVE-2016-5669 Detail

Current Description

Crestron Electronics DM-TXRX-100-STR devices with firmware before 1.3039.00040 use a hardcoded 0xb9eed4d955a59eb3 X.509 certificate from an OpenSSL Test Certification Authority, which makes it easier for remote attackers to conduct man-in-the-middle attacks against HTTPS sessions by leveraging the certificate's trust relationship.

Source: MITRE Last Modified: 08/03/2016 + View Analysis Description

INRIA Paris-Rocquencou

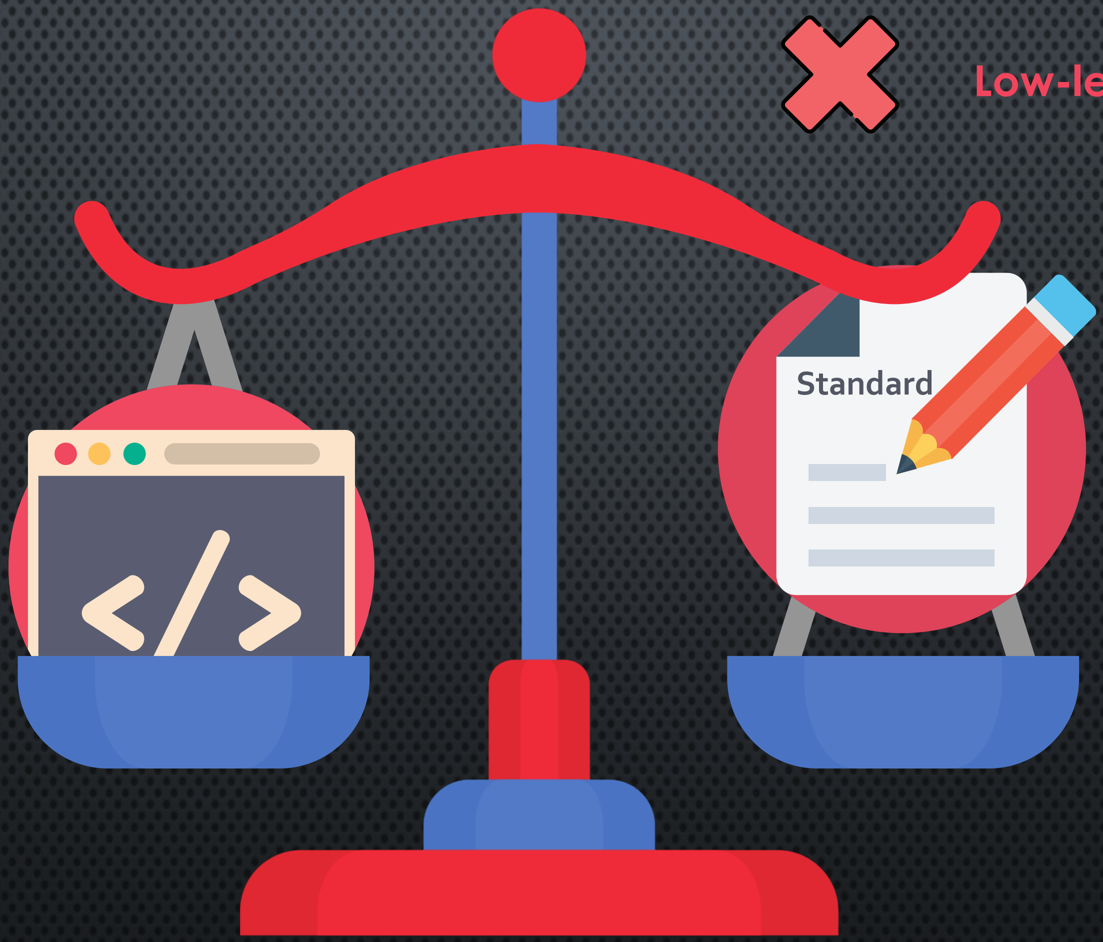
Abstract

We present FLEXTLS, a tool for rapidly prototyping and testing implementations of the Transport Layer Security (TLS) protocol. FLEXTLS is built upon MITLS, a verified implementation of TLS, and hence protocol scenarios ... FLEXTLS can benefit from robust li

an existing implementation in order to test potentially affected libraries.

In this paper, we present FLEXTLS, a tool for instrumenting arbitrary sequences of TLS messages. FLEXTLS was originally created in order to write proofs of concept of complex transport layer attacks such as Triple Handshake or the early CCS attack against OpenSSL [9]. It

X.509 Compliance Checking

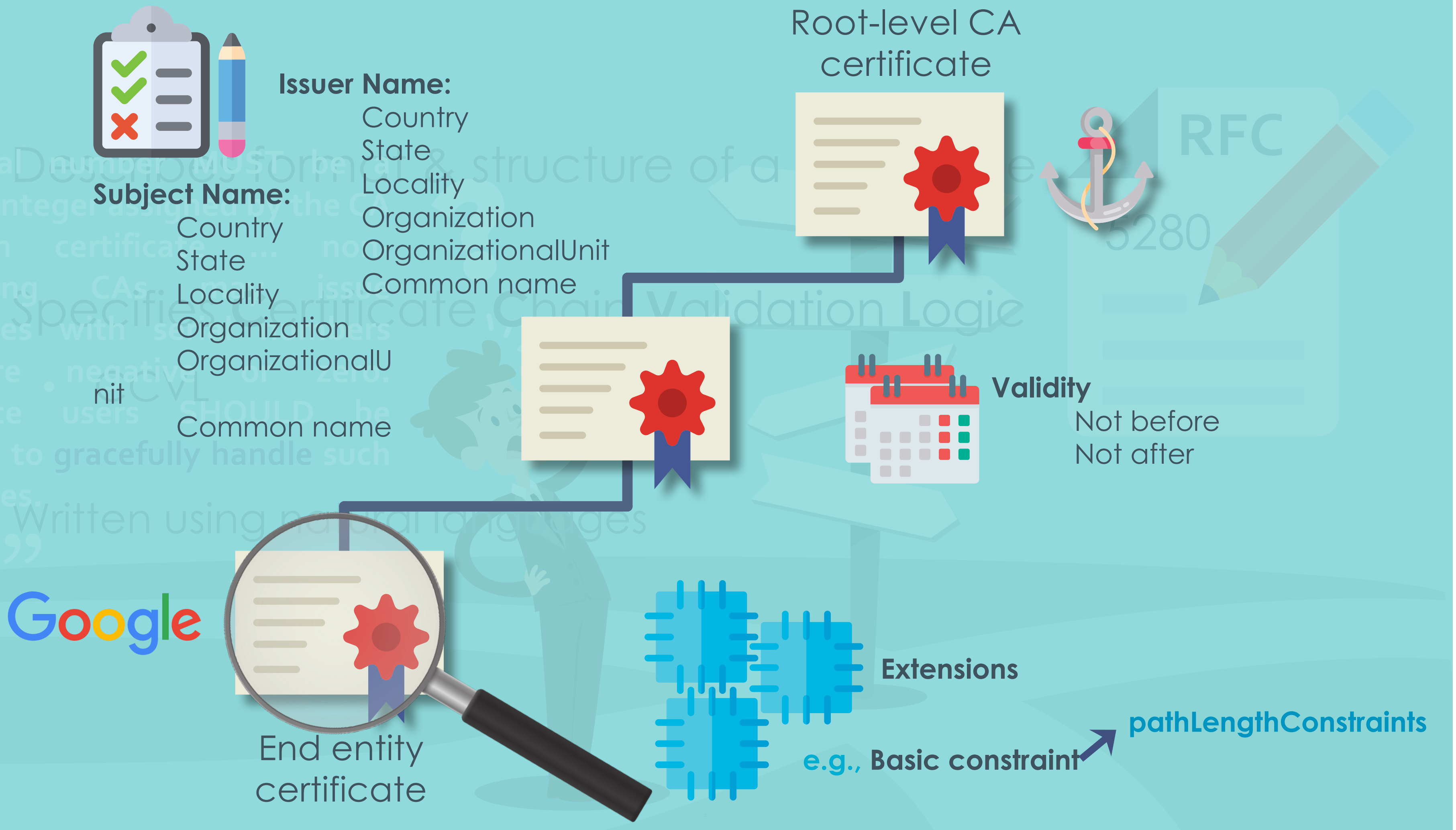


Low-level memory bugs

Implementation

Standard

X.509 RFC 5280



Issuer Name:
Country
State
Locality
Organization
OrganizationalUnit
Common name

Subject Name:
Country
State
Locality
Organization
OrganizationalU
nit
Common name

Root-level CA certificate

End entity certificate

Validity
Not before
Not after

Extensions

e.g., **Basic constraint** → **pathLengthConstraints**

Describes format & structure of a certificate

Specifies Certificate Chain Validation Logic

Written using natural languages

“the serial number MUST be a positive integer assigned by the CA to each certificate... non-conforming CAs may issue certificates with serial numbers that are negative or zero. Certificate users SHOULD be prepared to gracefully handle such certificates.”

Noncompliance in X.509

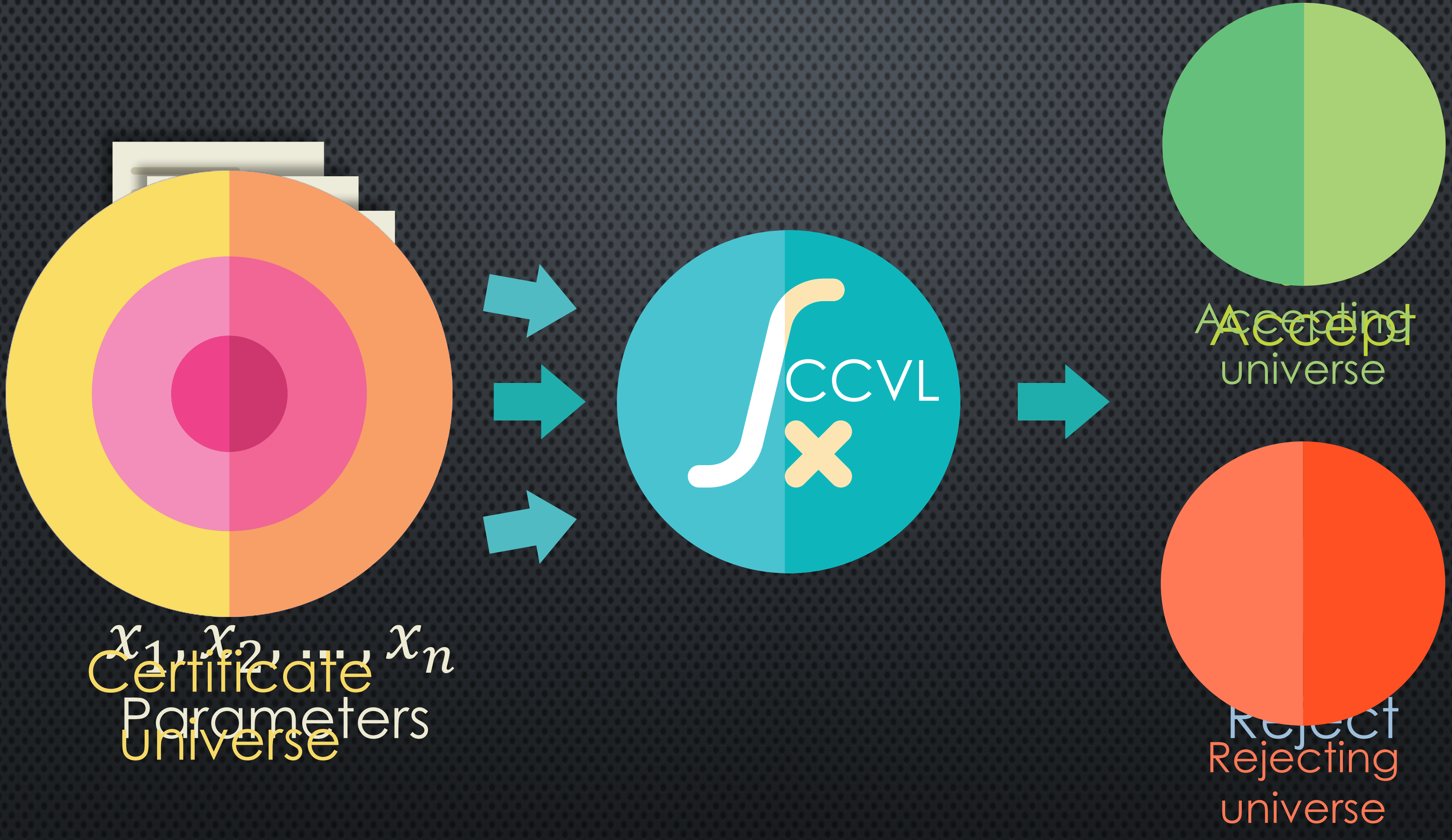
Overly permissive

Impersonation attack

Loss of service

Overly restrictive

Problem statement



$x_1, x_2, \ldots, x_n$

Certificate universe

Parameters

CCVL

Accepting universe

Accept

Reject

Rejecting universe

Problem statement

1

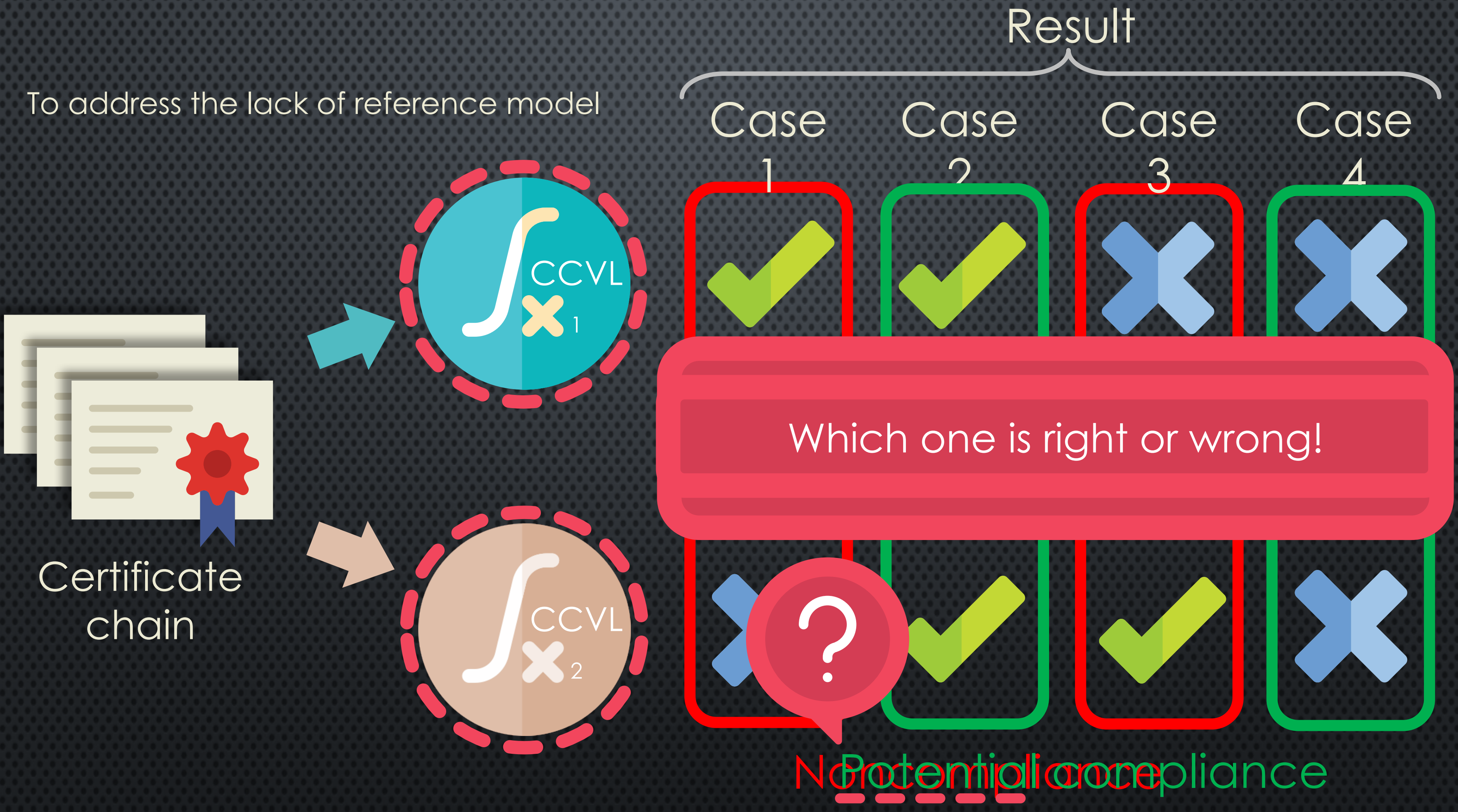
How can we check the noncompliance of an implementation in the lack of the reference model?

2

How can we obtain the accepting and rejecting universes?

Differential testing

To address the lack of reference model



	Case 1	Case 2	Case 3	Case 4
CCVL ×1	✓	✓	✗	✗
CCVL ×2	✗	✓	✓	✗

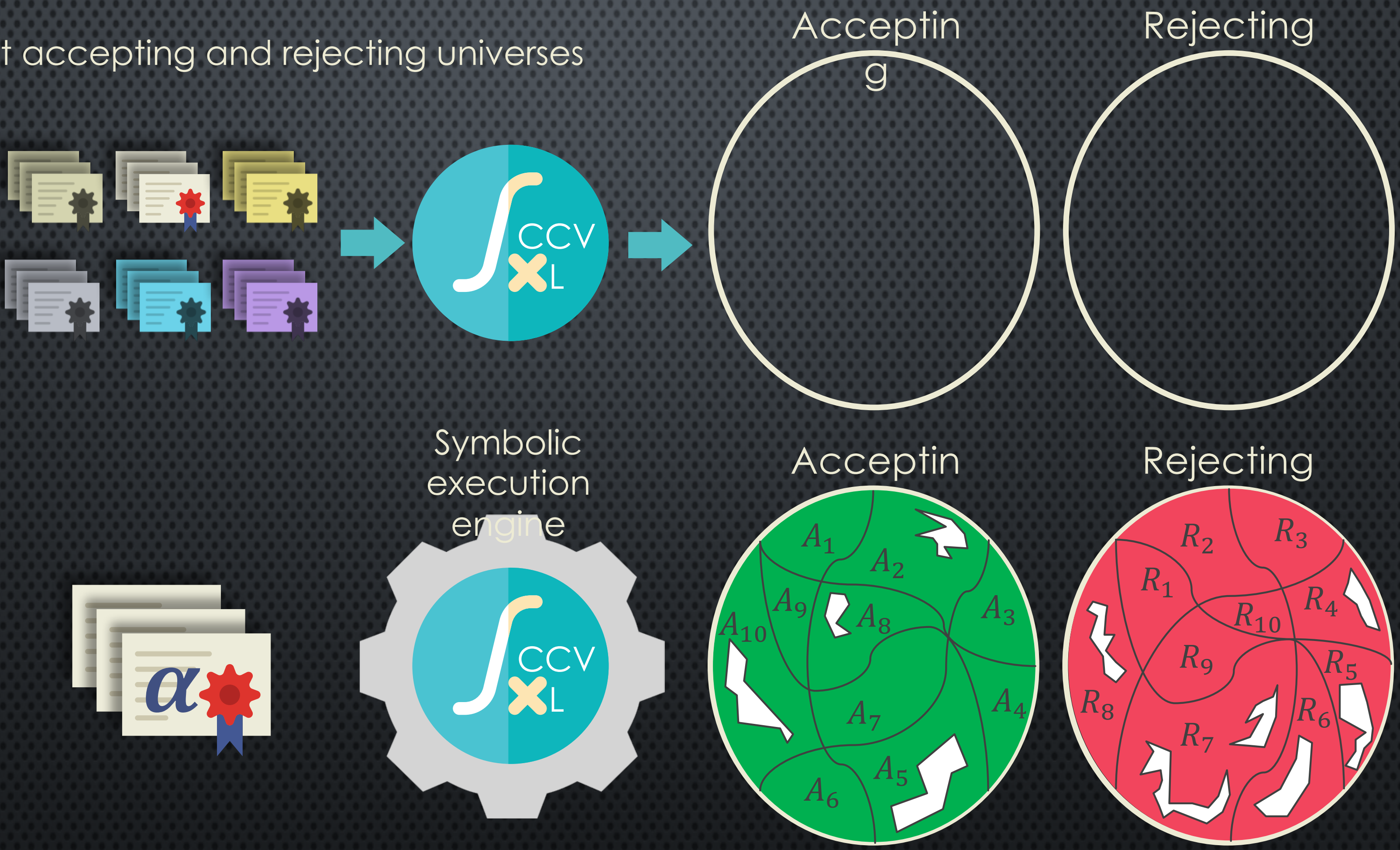
Which one is right or wrong!

Partitioning the universe

To construct accepting and rejecting universes

Fuzzing

Symbolic executio n



Acceptin g

Rejecting

Symbolic execution engine

Acceptin

Rejecting

SymCert

Chau et al., *IEEE Symposium on Security and Privacy*, 2017.

Employs symbolic execution technique

tations

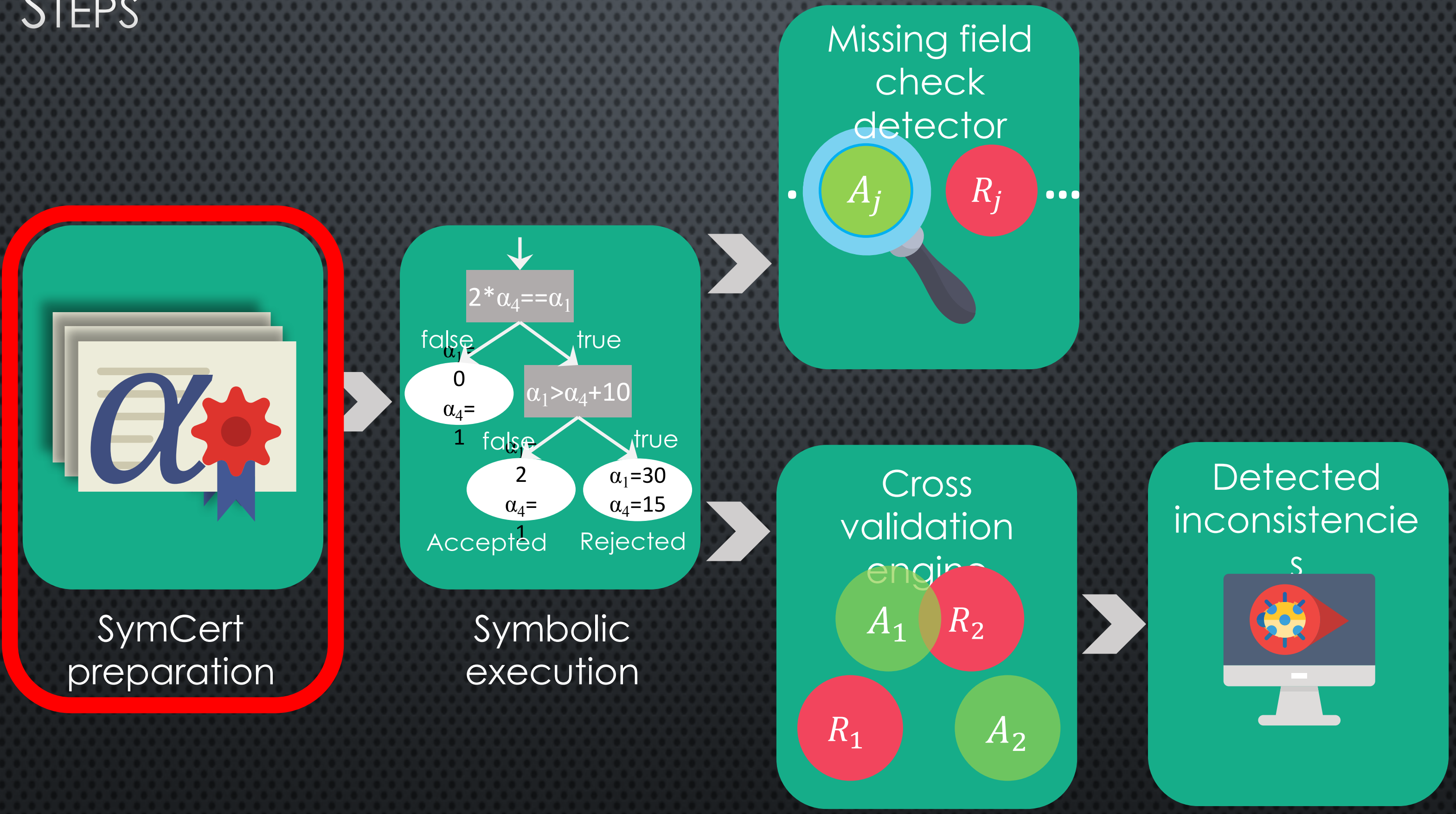
Testing implementations by providing a symbolic input, SymCert, and extracting regions in the universes instead of some samples

To compensate symbolic execution limitations

- Fully leveraging the open source nature of source code

SymCert

- STEPS



SymCert preparation

Symbolic execution

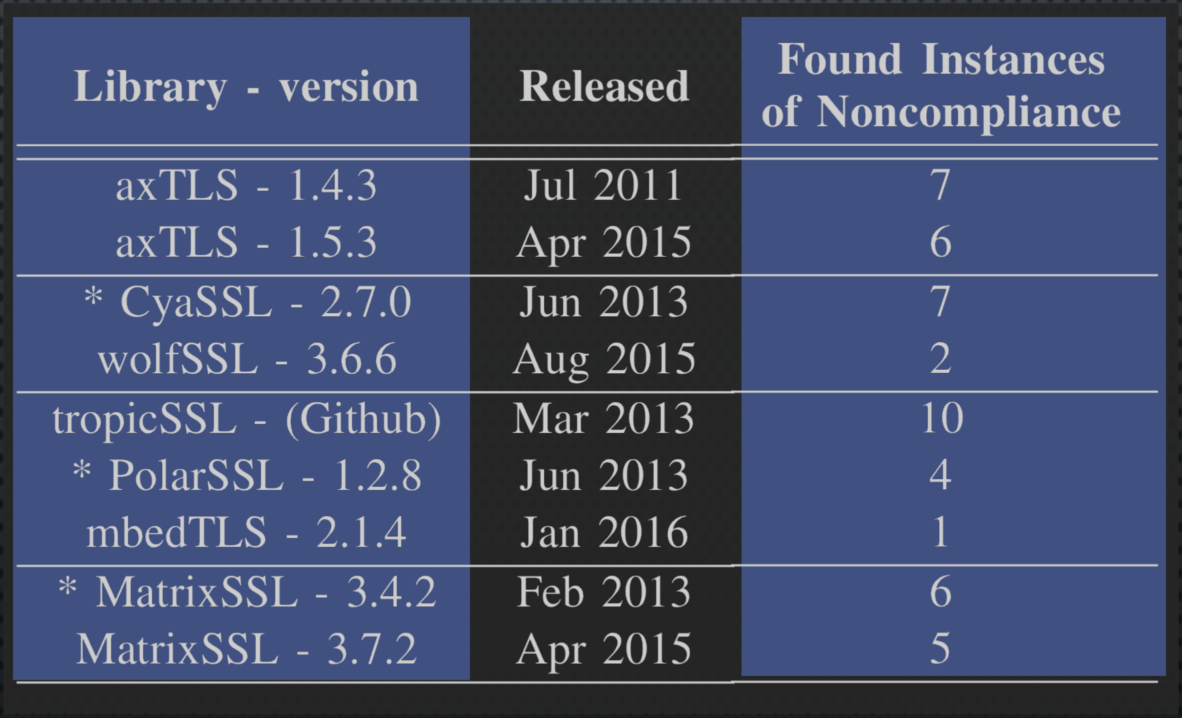
Missing field check detector

Cross validation engine

Detected inconsistencies

SymCert

- Results



Library - version	Released	Found Instances of Noncompliance
axTLS - 1.4.3	Jul 2011	7
axTLS - 1.5.3	Apr 2015	6
* CyaSSL - 2.7.0	Jun 2013	7
wolfSSL - 3.6.6	Aug 2015	2
tropicSSL - (Github)	Mar 2013	10
* PolarSSL - 1.2.8	Jun 2013	4
mbedTLS - 2.1.4	Jan 2016	1
* MatrixSSL - 3.4.2	Feb 2013	6
MatrixSSL - 3.7.2	Apr 2015	5

SymCert

- Exemplary finding (Extension Processing In CyaSSL)

```
switch (oid) {
...
case AUTH_INFO_OID:
DecodeAuthInfo(&input[idx], length, cert);
break;
case ALT_NAMES_OID:
DecodeAltNames(&input[idx], length, cert);
case AUTH_KEY_OID:
DecodeAuthKeyId(&input[idx], length, cert);
break;
... }
```

SymCert

- Exemplary finding
  - Correct UTCTime Year Range: **1950** to **2049**

MatrixSSL 3.7.2

```
y = 2000 + 10 * (c[0] - ’0’) + (c[1] - ’0’); c += 2;
/* Years from ’96 through ’99 are in the 1900’s */
if (y >= 2096) { y -= 100; }
```

1996 to 2095

tropicSSL

```
to->year += 100 * (to->year < 90);
to->year += 1900;
```

1990 to 2089

axTLS 1.4.3
axTLS 1.5.3

```
if (tm.tm_year <= 50) { /* 1951-2050 thing */
   tm.tm_year += 100;
}
```

1951 to 2050

SymCert

- Exemplary finding
  - Lax OID ExtKeyUsage Matching(MatrixSSL 3.7.2, wolfSSL 3.6.6)

ExtKeyUsage → **Purposes of using a key** → **Object Identifier**

a.b.c.d.e.f.g.h *e.g., 1.3.6.1.5.5.7.3.1* **Server Authentication**

1.3.6.1.5.5.7.3.1 *vs* *a+b+c+d+e+f+g+h=71*

Overly Permissive **Compatibility Issues**

SymCert

- Discussion

⇧ • Capable of finding more in-depth bugs

⇧ • Accepting and rejecting universes with high coverage

⇧ • Leverages the open source nature of the implementations

⇩ • Unable to handle traditional (Large-scale) libraries

Future work

Reference implementation

A substitution for existing implementation

Act as an oracle

Complete Formally verified SSL/TLS ecosystem