Revisiting the X.509 Certification Path Validation

RuhrSec 2018, Bochum

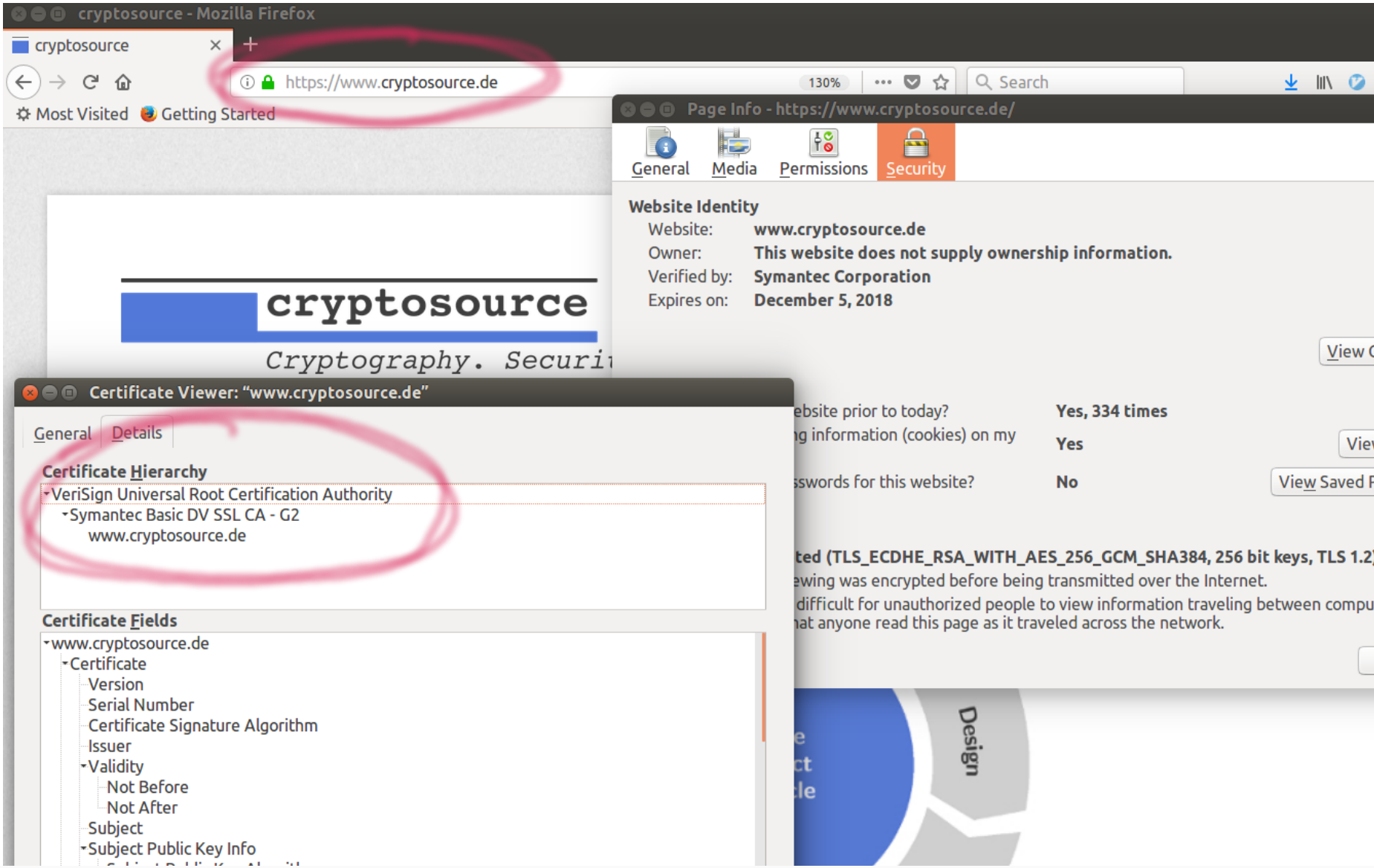
Dr. Falko Strenzke

cryptosource GmbH, Darmstadt
fstrenzke@cryptosource.de

cryptosource
Cryptography. Security.

June 6, 2018

X.509 Certification Path Validation



BSI Project on the Certification Path Validation

- X.509 certification path validation
- subject to many historical implementation errors
- creation of
  - a test tool
  - a test specification
- application to 10 test subjects

Joint Work

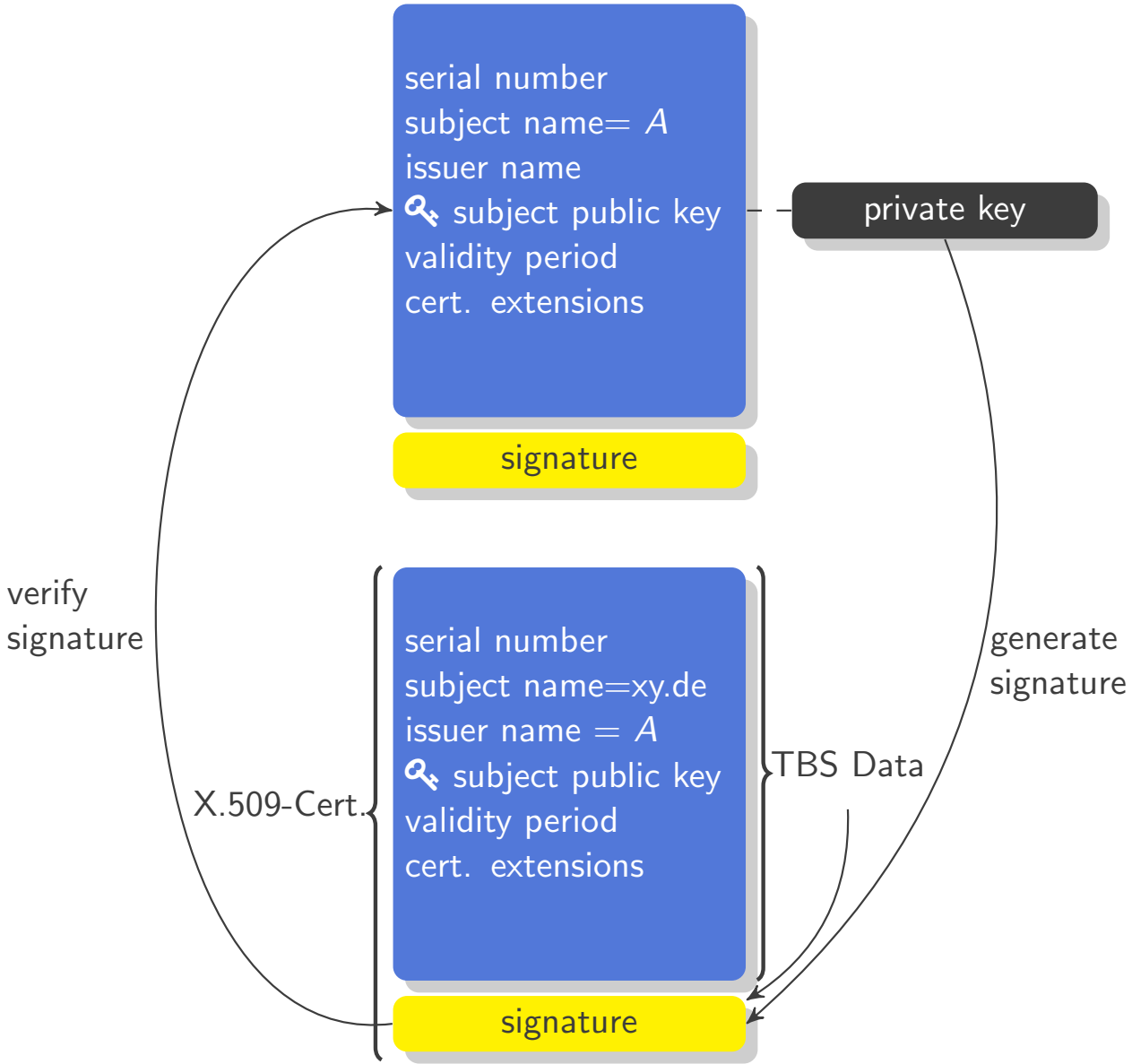
Armin Cordel
BSI

Heike Hagemeier
BSI

Evangelos Karatsiolis
MTG AG

Falko Strenzke
cryptosource GmbH

X.509 Certificates



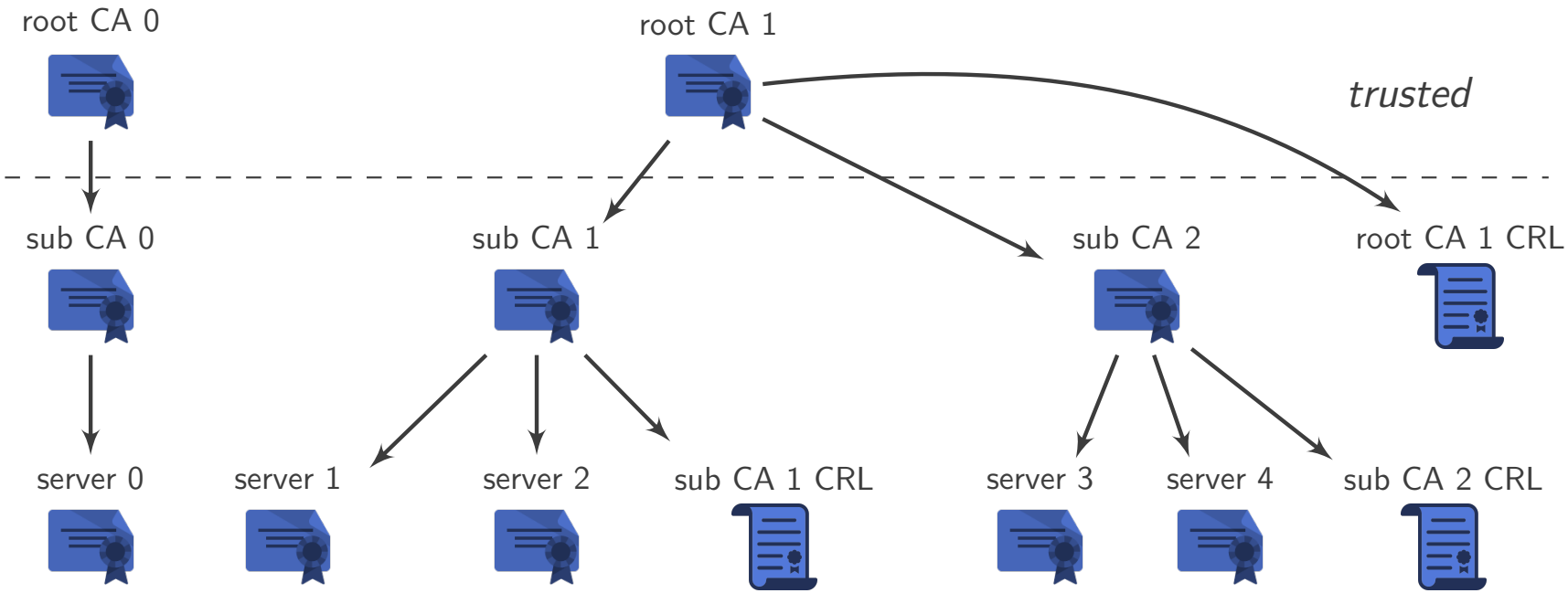
X.509 Certificate build-up

X.509 Certificate: ASN.1/DER encoding (TLV)

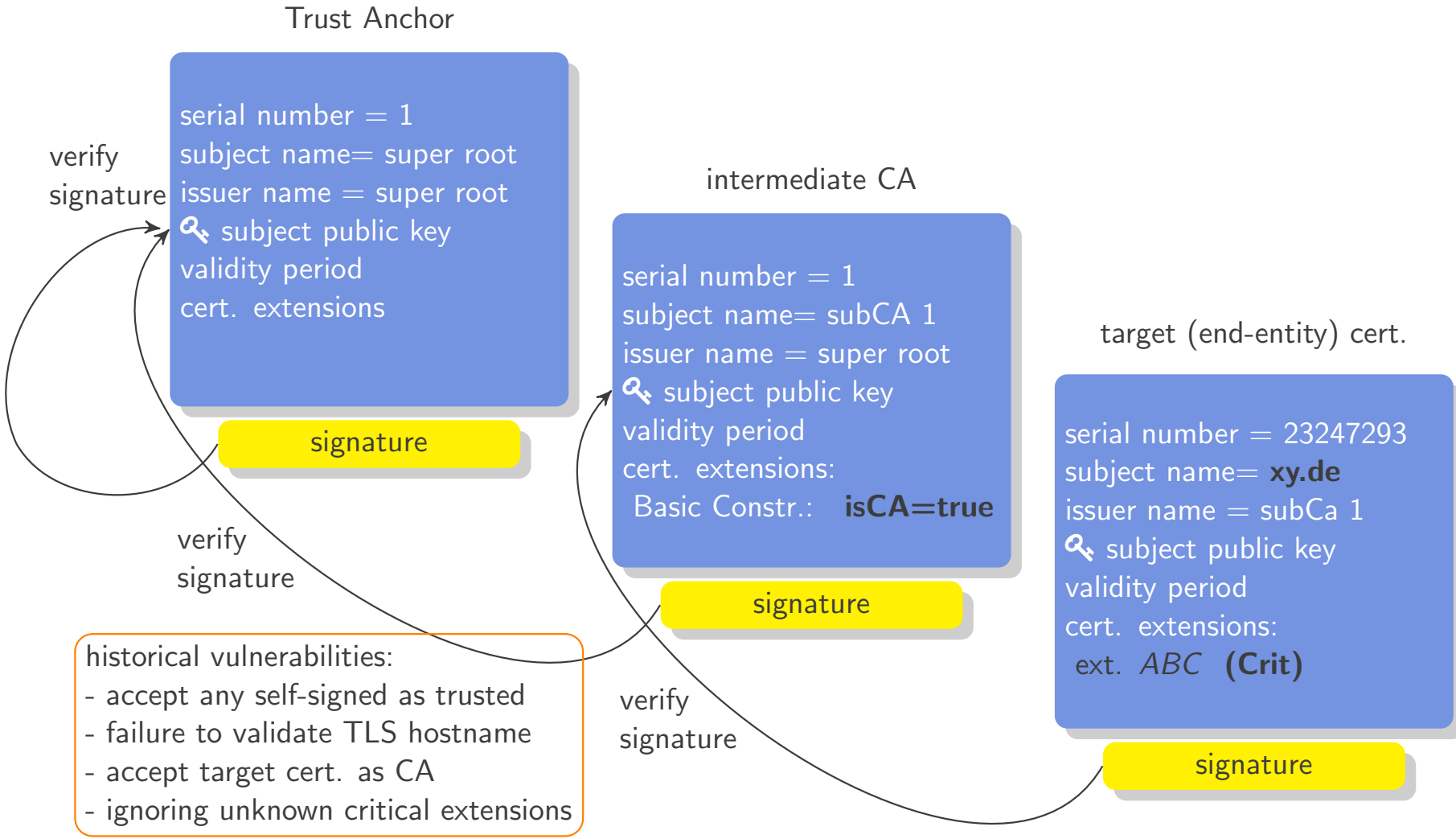
- TBS-Data
  - Version (v1,v2,v3)
  - Serial number
  - Signature algorithm
  - Issuer (Issuer DN)
  - Owner (Subject DN)
  - notBefore (creation date)
  - notAfter (expiration date)
  - Public key
  - Extensions (critical/non-critical(*)), e.g.
    - Basic Constraints (CA certificate yes/no)
    - Key Usage
    - Pointers to revocation information
- Signature

(*)Extension marked as critical: extension must be processed or cert. rejected

Internet PKI



Certificate Chains



Historical Vulnerabilities in X.509 Validation

Further historical Vulnerabilities in the X.509 Certificate Validation

- Null-Prefix Attack
  - certificate authority (CA) has to validate applicant's ownership of the domain
  - apply for certificate $\mathrm{xy.de\backslash 0abc.com}$
  - path validation
    - routines see \0 as byte with value 0
    - in C language this is the string terminator
    - and thus the certificate is considered valid for `xy.de`
- Cryptography related vulnerabilities
  - Bleichenbacher's low exponent attack: invalid parsing of "decrypted" RSA signatures
  - empty signatures accepted
  - etc.

Existing Test Tools: Frankencerts

- research project 2014 “Frankencerts”
- idea:
  - use the internet as a source for a diversity of X.509 certificates
  - use an algorithm to create mutants (combinations of parts) of this corpus
  - use **differential testing** to find deviating results for the same certificate chain
  - differential testing: input the same test data into multiple test subjects and observe if any behaves differently
- Pros:
  - no modelling of the test data or the validation algorithm necessary
  - identifies a large number of (subtle) errors
- Cons:
  - requires manual analysis when test results deviate
  - generation of test data satisfying application specific requirements is not straight forward

Existing Test Tools: NIST's PKITS Test Suite

- PKITS Test Suite (NIST)
  - Large number of static test cases
  - Users must organise data themselves
  - De-facto standard for libraries
  - Pros:
    - High test coverage especially for extensions
  - Cons: static test data
    - CommonName / SAN
    - Signature algorithms

    cannot be varied

New Test Specification and Tool

- Test Specification
  - Test suite with covering the most important aspects
  - “dynamic parametrization”
    - e.g. instantiate the same test with different signature algorithms
- Test Tool
  - Certification Path Validation Test Tool (CPT)
  - Open Source (EUPL, Apache 2.0, . . .)
  - generate the test data from test specification
  - execute the test against TLS, IPsec and S/MIME applications

Test Specification

Systematic derivation of the test specification:

- Rules from standards (RFC 5280 + Application specific)
- Historical errors:
  - CVE Vulnerability database (https://cve.mitre.org/)
  - Search terms (certificate validation, intermediate CA, ...)
  - Publications
  - Errors known to us (NULL character)

Test Specification

- 76 test cases
  - General
  - Extensions
  - Revocation
  - Cryptographic aspects
  - Email (S/MIME)
  - IPsec
  - TLS Server
  - TLS Client

Test Data Specification

```
<Certificate id="CERT_PATH_BASIC_01_ROOT_CA" type="TA">
  <VerifiedBy>CERT_PATH_BASIC_01_ROOT_CA</VerifiedBy>
  <Version> 2 </Version>
  <SerialNumber>1</SerialNumber>
  <Signature>1.2.840.113549.1.1.11</Signature>
  <IssuerDN encoding="UTF-8">CN=Root CA, C=DE</IssuerDN>
  <SubjectDN encoding="UTF-8">CN=Root CA, C=DE</SubjectDN>
  <NotBefore encoding="GEN">-3D</NotBefore>
  <NotAfter>+5Y</NotAfter>
  <PublicKey>RSA,2048</PublicKey>
  <Extension oid="2.5.29.15" critical="true" name="keyUsage"
    type="pretty">keyCertSign</Extension>
  <Extension oid="2.5.29.19" critical="true"
      name="basicConstraints" type="raw">MIIo...</Extension>
</Certificate>
```

Test Data Specification

```
<Certificate id="CERT_PATH_BASIC_01_ROOT_CA" type="TA">
  <VerifiedBy>CERT_PATH_BASIC_01_ROOT_CA</VerifiedBy>
  <Version> 2 </Version>
  <SerialNumber>1</SerialNumber>
  <Signature>1.2.840.113549.1.1.11</Signature>
  <IssuerDN encoding="UTF-8">CN=Root CA, C=DE</IssuerDN>
  <SubjectDN encoding="UTF-8">CN=Root CA, C=DE</SubjectDN>
  <NotBefore encoding="GEN">-3D</NotBefore>
  <NotAfter>+5Y</NotAfter>
  <PublicKey>RSA,2048</PublicKey>
  <Extension oid="2.5.29.15" critical="true" name="keyUsage"
    type="pretty">keyCertSign</Extension>
  <Extension oid="2.5.29.19" critical="true"
      name="basicConstraints" type="raw">MIIo...</Extension>
</Certificate>
```

Test Data Specification

```
<Certificate id="CERT_PATH_BASIC_01_ROOT_CA" type="TA">
  <VerifiedBy>CERT_PATH_BASIC_01_ROOT_CA</VerifiedBy>
  <Version> 2 </Version>
  <SerialNumber>1</SerialNumber>
  <Signature>1.2.840.113549.1.1.11</Signature>
  <IssuerDN encoding="UTF-8">CN=Root CA, C=DE</IssuerDN>
  <SubjectDN encoding="UTF-8">CN=Root CA, C=DE</SubjectDN>
  <NotBefore encoding="GEN">-3D</NotBefore>
  <NotAfter>+5Y</NotAfter>
  <PublicKey>RSA,2048</PublicKey>
  <Extension oid="2.5.29.15" critical="true" name="keyUsage"
    type="pretty">keyCertSign</Extension>
  <Extension oid="2.5.29.19" critical="true"
      name="basicConstraints" type="raw">MIIo...</Extension>
</Certificate>
```

Test Data Specification

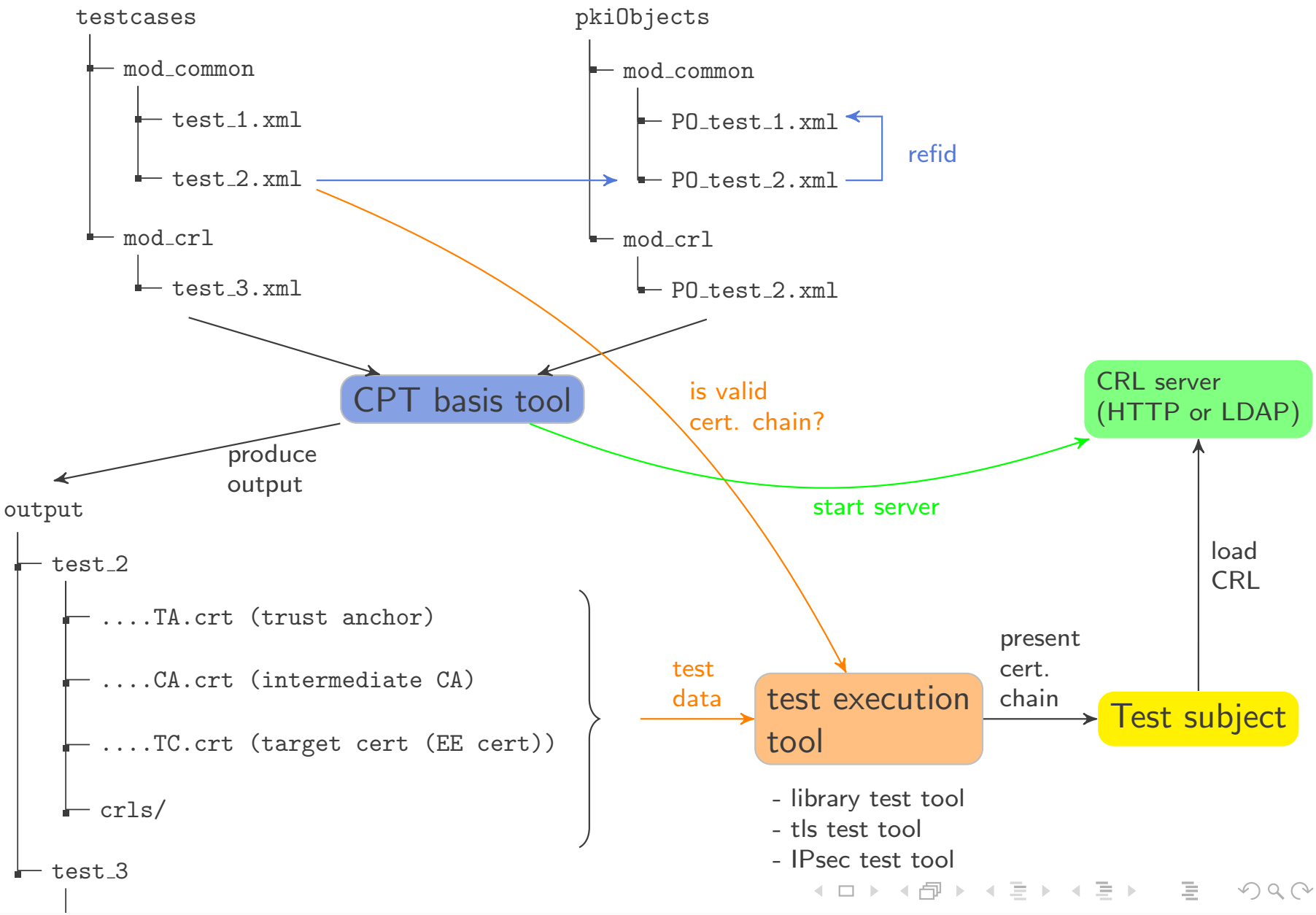
```
<Certificate id="CERT_PATH_BASIC_01_ROOT_CA" type="TA">
  <VerifiedBy>CERT_PATH_BASIC_01_ROOT_CA</VerifiedBy>
  <Version> 2 </Version>
  <SerialNumber>1</SerialNumber>
  <Signature>1.2.840.113549.1.1.11</Signature>
  <IssuerDN encoding="UTF-8">CN=Root CA, C=DE</IssuerDN>
  <SubjectDN encoding="UTF-8">CN=Root CA, C=DE</SubjectDN>
  <NotBefore encoding="GEN">-3D</NotBefore>
  <NotAfter>+5Y</NotAfter>
  <PublicKey>RSA,2048</PublicKey>
  <Extension oid="2.5.29.15" critical="true" name="keyUsage"
    type="pretty">keyCertSign</Extension>
  <Extension oid="2.5.29.19" critical="true"
      name="basicConstraints" type="raw">MIIo...</Extension>
</Certificate>
```

```
<CRL id="CERT_PATH_CRL_09_SUB_CA_CRL">
  <Location>http://cert_path_host/sub_ca_crl.crl</Location>
  <VerifiedBy>CERT_PATH_CRL_09_SUB_CA</VerifiedBy>
  <Version>1</Version>
  <Signature>1.2.840.113549.1.1.11</Signature>
  <IssuerDN encoding="UTF8">CN=Test Sub CA, C=DE</IssuerDN>
  <ThisUpdate>-8D</ThisUpdate>
  <NextUpdate>-1D</NextUpdate>
  <Extension oid="2.5.29.35" critical="false" name="AKI"
    type="pretty"></Extension>
  <Extension oid="2.5.29.20" critical="false" name="CRL Number"
    type="pretty">9</Extension>
</CRL>
```

Specification of a Certification Path

```
<PKIObjects>
<Certificate id="CERT_PATH_CRL_09_ROOT_CA" refid="ROOT_CA"
    overwrite="false" type="TA" />
  <Certificate id="CERT_PATH_CRL_09_SUB_CA_1" refid="SUB_CA"
    overwrite="true">
  ...
  </Certificate>
  <Certificate id="CERT_PATH_CRL_09_SUB_CA_2" refid="SUB_CA"
    overwrite="true">
  ...
  </Certificate>
  <Certificate id="CERT_PATH_CRL_09_EE" refid="CRL_02_EE"
    overwrite="true" type="TC">
  ...
  </Certificate>
  <CRL id="CERT_PATH_CRL_09_ROOT_CRL">
  ...
  </CRL>
  <CRL id="CERT_PATH_CRL_09_SUB_CA_CRL">
  ...
  </CRL>
</PKIObjects>
```

CPT Processing



Additional Test Tools

- library test tools
  - C/C++ command line tool
  - Java command line tool
- TLS test tool
  - TLS test client
  - TLS test server
  - based on the Botan library
  - additionally: Web frontend to test Browsers
- IPsec test tool
  - based on strongSwan IPsec implementation

Test Subjects

- Test subjects
  - Cryptographic Libraries
    - OpenSSL (C)
    - Botan (C++)
    - mbedTLS (C)
    - Bouncy Castle (Java)
    - OpenJDK (Java)
  - Applications
    - Apache (HTTP Server)
    - Firefox (Browser)
    - strongSwan (IPsec)
    - OpenVPN (VPN)
    - KMail (Email Client)

OpenJDK

- OpenJDK shows no single error
- implementation strongly oriented at the formal algorithms from RFC 5280
- `http://openjdk.java.net/`

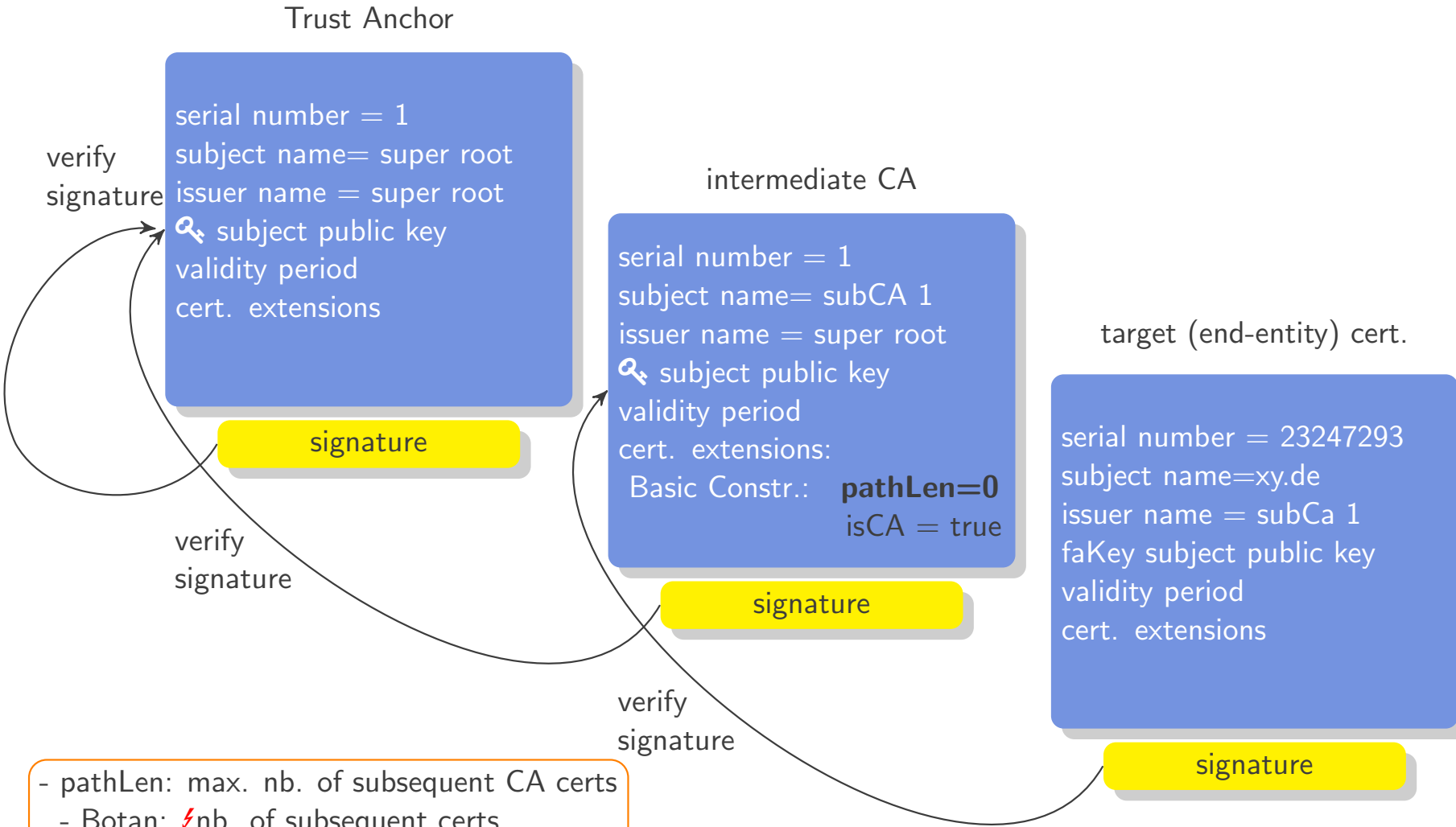
Compatibility Issues

Description	Botan	Bouncy Castle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
too restrictive handling of path length	E								
too restrictive handling of path length with self-issued certificates		E	E			E	E		E
performing non-exhaustive path search	E		E	E	–	–	–	–	–
Acceptance of MD5 as signature hash algorithm with default config		E		E	E		E	E	E

Compatibility

Description	Botan	Bouncy Castle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
too restrictive handling of path length	**E**								
too restrictive handling of path length with self-issued certificates		**E**	**E**			**E**	**E**		**E**
performing non-exhaustive path search	E		E	E	–	–	–	–	–
Acceptance of MD5 as signature hash algorithm with default config		E		E	E		E	E	E

Errors regarding the Path Length



- pathLen: max. nb. of subsequent CA certs
  - Botan: ↯nb. of subsequent certs
- self-issued certs not counted
  - self-issued if issuer=subject
  - 5 test subjects err
  - Mozilla: “deliberate choice”

Compatibility Issues

Description	Botan	Bouncy Castle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
too restrictive handling of path length	E								
too restrictive handling of path length with self-issued certificates		E	E			E	E		E
performing non-exhaustive path search	**E**		**E**	**E**	–	–	–	–	–
Acceptance of MD5 as signature hash algorithm with default config		E		E	E		E	E	E

Compatibility Issues

Description	Botan	Bouncy Castle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
too restrictive handling of path length	E								
too restrictive handling of path length with self-issued certificates		E	E			E	E		E
performing non-exhaustive path search	E		E	E	–	–	–	–	–
Acceptance of MD5 as signature hash algorithm with default config		**E**		**E**	**E**		**E**	**E**	**E**

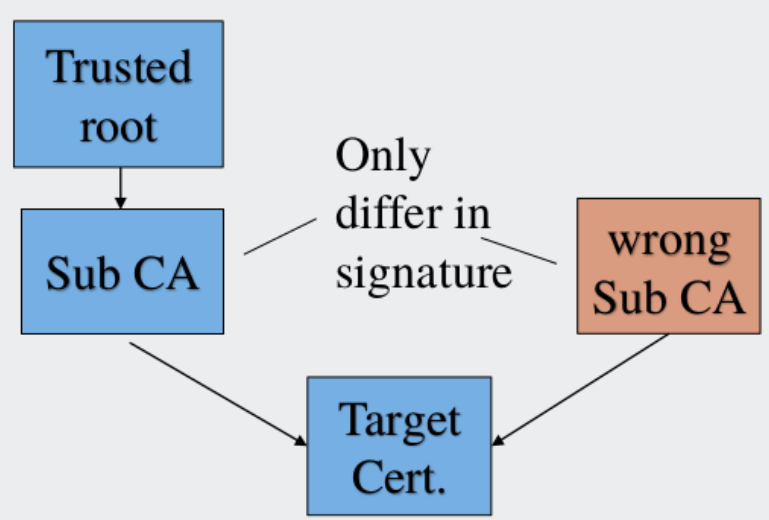
Certificate Path Validation

- Certificate Path Validation
  - **Path Construction** (not for TLS, etc.)
  - Validation of Certificate Chain (RFC 5280)
  - Application specific validations
    - Specific extensions in target certificate
    - E.g. Key Usage for TLS
  - Revocation Check (RFC 5280)

(Non-) Exhaustive Path Construction

Path Construction

- Input:
  - Target certificate
  - Set of trusted certificates
  - Pool of untrusted certificates
- Algorithm
  - Find issuer A to target certificate
  - Find issuer of A
  - ...
  - Until a trusted root is reached
- What if no issuer of sub CA can be found?
  - Exhaustive search:
    - discard that sub CA again
    - try next candidate
  - Non-exhaustive search: break, target certificate is invalid
    - Problem: DOS through wrong untrusted certificate in cache



Issues: Certificates

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of an invalid certificate version		E		E				E	
acceptance of intermediate certificate without basic constraints extension					E			E	
acceptance of intermediate certificate without KeyCertSign Key Usage									E
acceptance of target certificate with Key Usage extension only featuring keyAgreement key usage									E

Issues: Certificates

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of an invalid certificate version		E		E				E	
acceptance of intermediate certificate without basic constraints extension					E			E	
acceptance of intermediate certificate without KeyCertSign Key Usage									E
acceptance of target certificate with Key Usage extension only featuring keyAgreement key usage									E

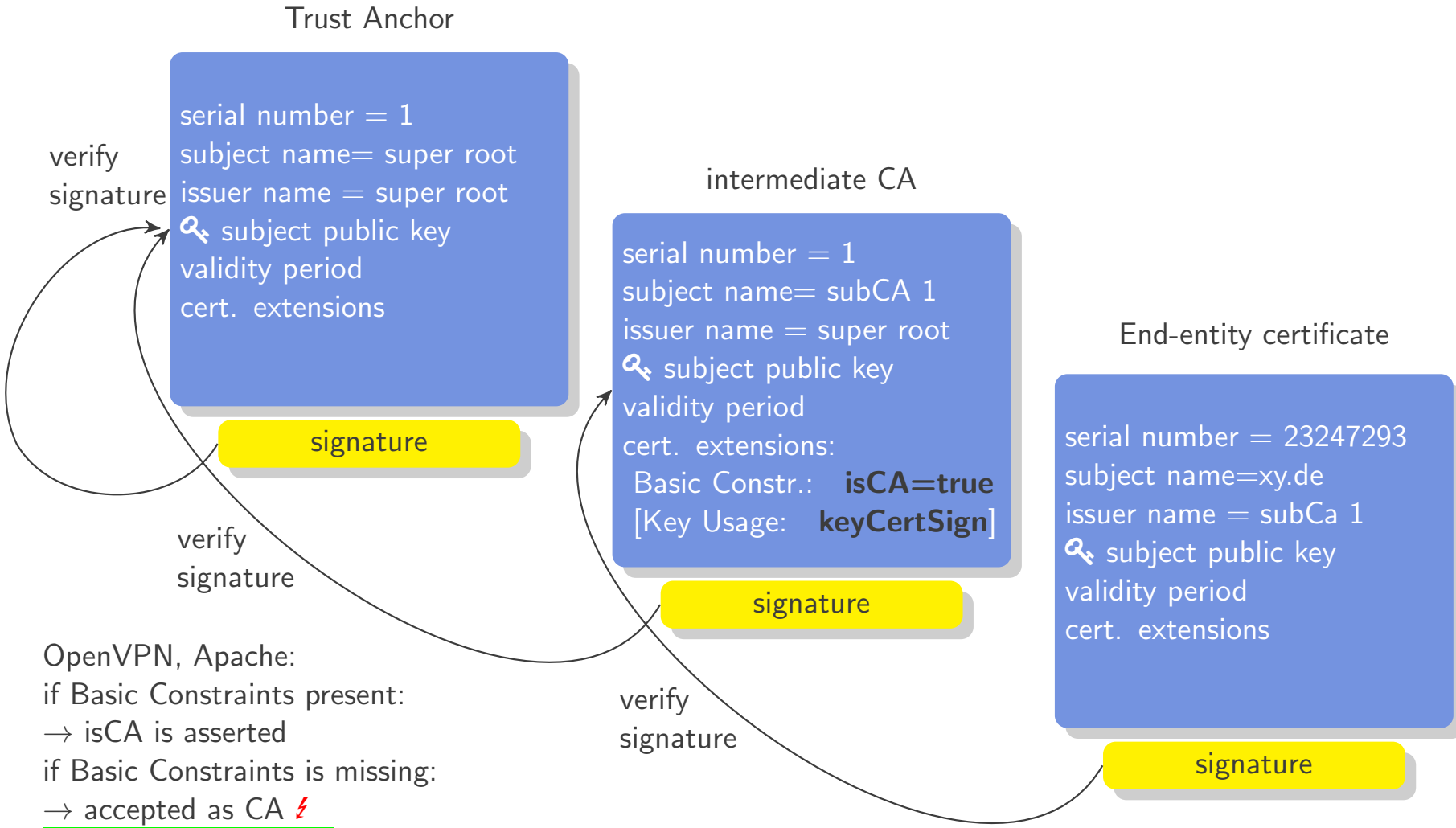
Invalid X.509 Certificate Version

- X.509 certificates carry a version number
- current (hightest) version is **v3**
- receiving certificate with **v4**
  - must be rejected: processing rules unknown
- system deployed now might have a vulnerability once version **4** is defined
  - compare with the transition from v1 or v2 to v3:
  - v3 introduced certificate extensions
  - assume an application processes a v3 certificate as v1 or v2 and ignores that is has **critical extensions** $\rightarrow$ vulnerability

Issues: Certificates

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of an invalid certificate version		E		E				E	
acceptance of intermediate certificate without basic constraints extension					E			E	
acceptance of intermediate certificate without KeyCertSign Key Usage									E
acceptance of target certificate with Key Usage extension only featuring keyAgreement key usage									E

Insufficient Criteria for CA Certificate



OpenVPN, Apache:
if Basic Constraints present:
→ isCA is asserted
if Basic Constraints is missing:
→ accepted as CA ↯

But: Key Usage with KeyCertSign is required.

(deliberate behaviour of older OpenSSL versions)

Issues: Certificates

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of an invalid certificate version		E		E				E	
acceptance of intermediate certificate without basic constraints extension					E			E	
acceptance of intermediate certificate without KeyCertSign Key Usage									E
acceptance of target certificate with Key Usage extension only featuring keyAgreement key usage									E

Key Usage for IPsec

- IPsec mandates the Key Usages *digitalSignature* or *nonRepudiation* in the target certificate
- strongSwan fails to verify this
- ↯ certificates not authorized for IPsec may be used

X.509 CRLs

X.509 CRL

- TBS-Data
  - Version (v2)
  - Signature algorithm
  - Issuer (Issuer DN)
  - thisUpdate (creation date)
  - nextUpdate (expiration date)
  - Revoked certificates
    - For each revoked certificate:
      - Serial number
      - Revocation date
      - CRL entry extensions (critical/non-critical), e.g. revocation reason
  - Extensions (critical/non-critical),
    - References to distribution locations
    - Revocation reasons covered
- Signature

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							**E**		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

Acceptance of Expired Certificates

- KMail doesn't report when a signature certificate or intermediate CA has expired
- the real problem is not using a certificate beyond its validity:
- an expired certificate may be removed from a CRL
- the risk is **accepting revoked certificates**

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			**E**						
ignoring lack of matching CRL for intermediate cert.									**E**
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

Ignoring Missing CRLs

- Opportunistic revocation check:
  - carried out if matching CRL for each certificate is input to the verification routine
  - if no CRL for a certificate is input, revocation check is skipped and certificate is accepted
  - mbedTLS: for all certificates in chain
  - strongSwan: only for intermediate CAs
- vulnerability
  - CRLs are downloaded over insecure connection
  - attacker renders CRL invalid (changes issuer name)
  - revoked certificate is valid!
- mbedTLS
  - deliberate choice (API doc) !
  - no reliable CRL check possible
  - not fixed for now, future version may contain a switch to enforce revocation checking
- strongSwan
  - is a bug, fixed in next release

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	**E**		**E**						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

Accepting Unknown Critical CRL Extensions

- a critical extension can alter processing rules
- **unknown** critical extension
  - New standardized extension
  - proprietary (application specific) extension

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

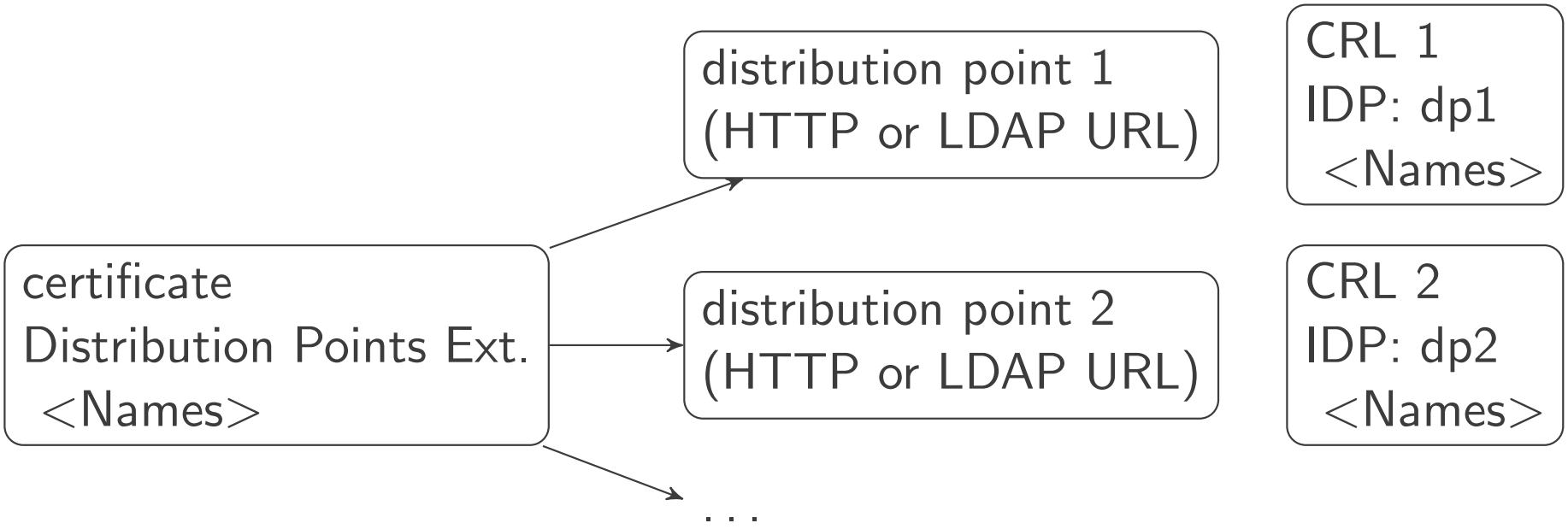
Not yet valid CRL

- `thisUpdate` date of CRL before *current time*
  - CRL is not yet valid
  - must have been issued by a system with deviating clock
- potential problem
  - revoked certificate removed from CRL when certificate expires
  - CRL issuer's system's clock is *ahead*
  - CRL issuer may have removed revoked certificates that the verifier still considers not expired
  - $\rightarrow$ revoked certificate accepted
- rather hypothetical problem
- (check of `thisUpdate` not mandated by RFC 5280)

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	**E**		**E**						**E**
usage of CRL from wrong Distribution Point							E		

Mismatching CRL-DP and IDP

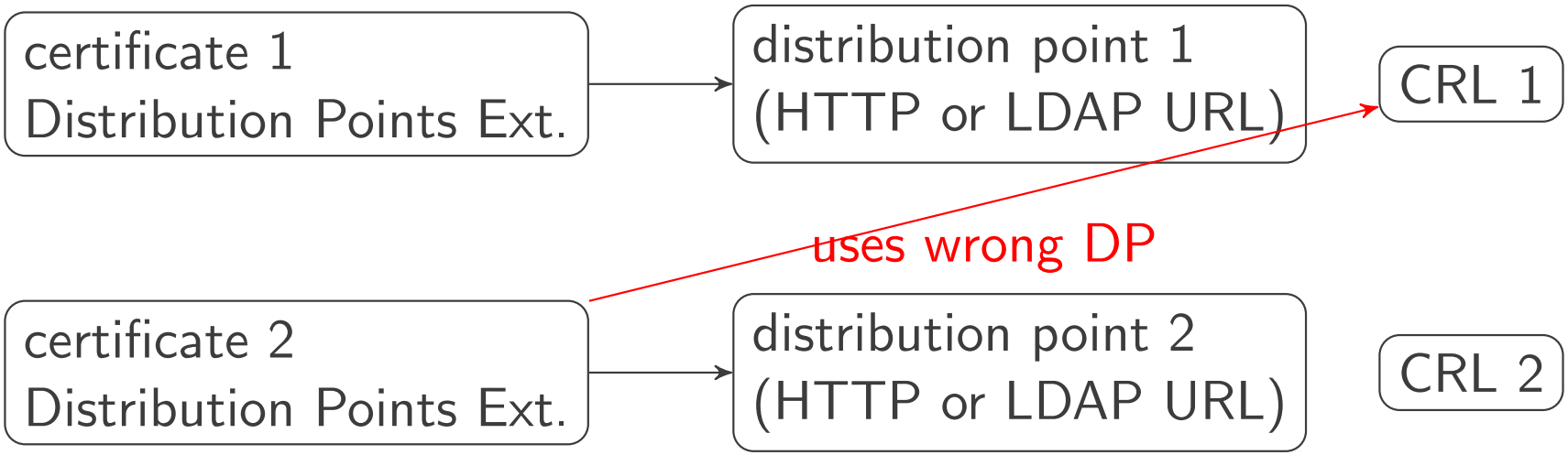


- certificate extension: CRL Distribution Points (CRL-DP)
- CRL extension: Issuing Distribution Point (IDP)
- each extension contains a set of names
- one name must match in both
- otherwise the CRL may not be used
- there may be multiple distribution points providing different CRLs from one issuer

Issues: CRLs

Description	Botan	BouncyCastle	mbedTLS	OpenSSL	Apache	Firefox	KMail	OpenVPN	strongSwan
acceptance of expired certificates							E		
ignoring lack of matching CRL for any cert.			E						
ignoring lack of matching CRL for intermediate cert.									E
acceptance of unknown critical CRL extensions	E		E						
acceptance of not yet valid CRLs									E
acceptance of mismatching certificate's CRL-DP and CRL's IDP	E		E						E
usage of CRL from wrong Distribution Point							E		

Using CRL from Wrong Distribution Point



- KMail caches CRLs after having downloaded them from a distribution point
- Caches only per CRL issuer, not by CRL DP
- Thus fails to check whether the verified certificate specifies another distribution point
- And thus uses potentially invalid CRL
- Standard (RFC 5280) is not completely exact here

Conclusion

- CPT as a new tool
  - dynamic test data generation
  - generic XML-based specification
  - tools for testing TLS, IPsec and Email (S/MIME)
  - fills a gap left by existing tools
- default test suite
  - derived from standards and previous errors
- applying tests to 10 well-known implementations
  - uncovering some relevant and interesting compatibility issues
  - some minor vulnerabilities in certificate validation
  - a number of more significant CRL-related issues
- insights in how implementers of widespread libraries / applications think
  - concern about compatibility sometimes higher than security
    - certificate version
  - some features are considered just irrelevant
    - self-issued certificates

Thank you for your attention!

- https://www.bsi.bund.de/EN/Topics/OtherTopics/CPT/cpt_node.html
- https://github.com/MTG-AG/cpt/
- https://github.com/cryptosource-GmbH/cpt-add-test-tools
- https://github.com/cryptosource-GmbH/cpt-native-lib-test