BTI 4202: X.509

Christian Grothoff

BFH

22.4.2022

Learning Objectives

Certificates

Standards

X.509 Certificates

X.509 Certificate Extensions

Creating a Certificate

Example: Protocol vulnerability

CV Certificates

Types of Certificates

Exercise: Reading a Certificate

Part I: Certificates

Digital Certificates

Digital certificates are data structures which bind a public key value to a subject (Identity). The binding is asserted by an issuer.

Digital Certificats

- Asymmetric cryptography uses 'private/public key' pairs → public keys have to be distributed.
- An attacker may distribute forged public keys with the purpose to mount a man-in-the-middle attack.
- Therefore a *public key* must be packed into a digital certificate together with some identifying information (e.g. the name of the holder) and has to be made tamper resistant[1].
- Only a digitally signed certificate can attest the integrity and the association of a public key to its owner in a trusted manner.

[1] Fälschungssicher

What is a certificate?

- A public-key certificate is a digitally signed statement that binds the identity of the entity to a public key
- If A trusts B, and knows B's public key, then A can learn C's public key if B issues a public key certification of C

Digital Certificats

- A digital certificate establishes a link between an identity and a cryptographic key.
- It contains mainly:
  - Identity and attributes of the subject to be protected.
  - Identity and attributes of the issuer of the certificate.
  - Public key of the subject, used for crypto operations.
  - Signature from an issuer, covering all certificate content.

  The signature authenticates the certificate content and makes it tamper resistant.
- The Issuer can be a person, governmental authority or a private or public provider. It must be trustworthy to verifiers (relying parties) in a specific context.

Part II: Standards

Certificate Standards

- X.509 Certificates [3]: The most common public key certificate standard for private, commercial and governmental use.
- Attribute Certificates [5]: A digital certificate that binds a set of descriptive data items (other than the public key) to a subject name.
- PGP Certificates [2]: Pretty Good Privacy defines its own certificate format. PGP certificates are used primarily to implement the PGP Web of Trust, a certification system far more decentralized than X.509.
- CV Certificates [ISO 7816/8]: Card Verifiable Certificates in a specially compact form used for chip and user authentication in smart-cards.

Unfortunately, X.509 certificates are not compatible to OpenPGP or CVC.

X.509 Certificates

The X.509 Authentication Framework designed by CCITT/ITU for the purpose of securing the X.500 Directory included a certificate specification where the format and data type remain important until today.
X.509 certificates are used in:

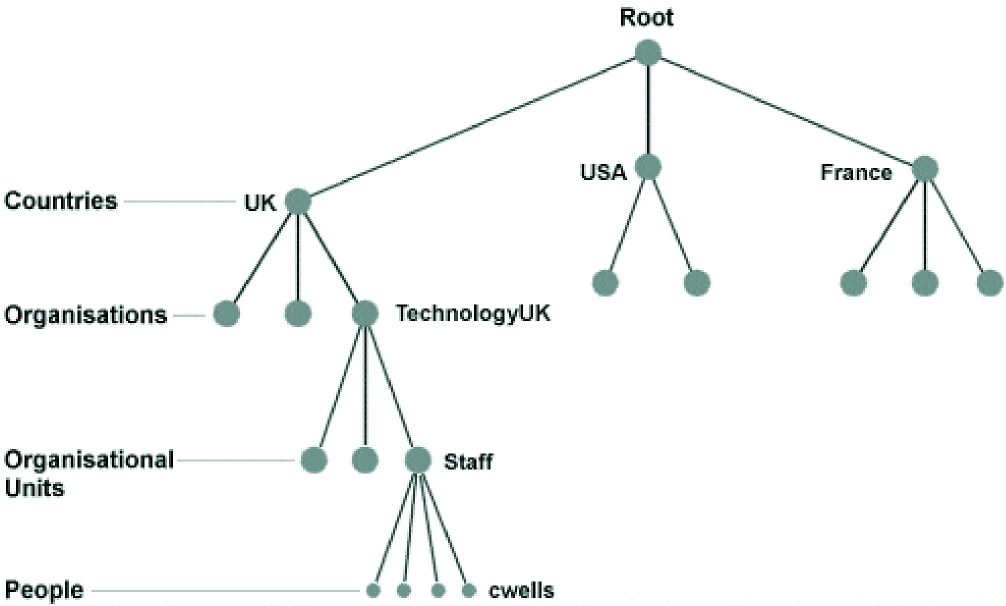
- The official IETF PKI standard: PKIX (Public Key Infrastructure for X.509) → `https://datatracker.ietf.org/wg/pkix`.
- The (older) industrial standard: PKCS (Public Key Cryptography Standards).

Public Key Cryptography Standards (PKCS)

- PKCS refers to a group of Public Key Cryptography Standards published by RSA Security.
- RSA Security, and its research division, RSA Labs, were interested in promoting and facilitating the use of public-key techniques.
- PKCS are not official open standards but industrial standards $\rightarrow$ they are still often used. Most of these standards where moved into IETF standard track or integrated in PKIX.

X.500 Naming

X.500 entries are based on a Distinguished Name (DN), which is part of a global unique namespace.



X.509 and TLS

X.509 is an ITU standard — but also [3].

- TLS servers (and sometimes clients) are identified by public key
- Public keys are *certified* by certificate authorities
- X.509 certificates are the format used for certificates
- Any certificate authority can certify keys for any domain

TLS is not the only major protocol using X.509!
There is also S/MIME for e-mail!

X.509 Standard Updates

- The "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" is actually standardized in:
  - RFC 5280: `https://tools.ietf.org/html/rfc5280` and has an update in:
  - RFC 6818: `https://tools.ietf.org/html/rfc6818`
- Algorithms and Identifiers for the "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile" are defined in:
  - RFC 3279: `https://tools.ietf.org/html/rfc3279`
  - With updates for additional algorithm and identifiers (e.g. ECC, ECDSA, etc.) in [7, 6, 8, 4].

ASN.1 and DER in X.509

Abstract Syntax Notation One (ASN.1):

- Defined by the ITU-T
- Specifies type and structure of data:
  - Type of value (integer, boolean, character string, etc.)
  - Structure (containment, order, options)
- Does not specify encoding (representation)

ASN.1 combined with a specific encoding rule facilitates the exchange of structured data.

The X.509 Standard uses Distinguished Encoding Rules (DER) for encoding. This provides the advantage to have a unique way of encoding each ASN.1 value.

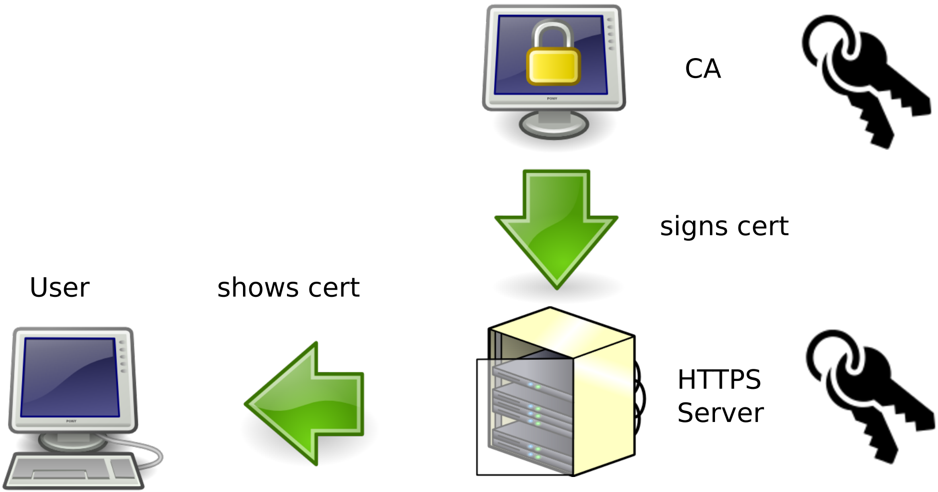
Object Identifiers (OIDs)

- An Object Identifier (OID) is an identifier used to *name an object* in a *hierarchically-assigned namespace*.
- In the security domain, OIDs serve to name almost every object type in X.509 certificates, such as components of Distinguished Names, Algorithms, etc.

Further information: `http://www.oid-info.com/`

Part III: X.509 Certificates

X.509 overview (reminder)



Contents of X.509 certificates

- X.509 version
- CA serial number
- A digital signature algorithm identifier
- The identity of the signer
- Validity period
- The identity of the subject (common name, org. unit, org., state, country)
- The public key of the subject
- Optional: URL to revocation center (OCSP!)
- Auxiliary information (identity address, alternative names)
- The digital signature

X.509 Standard Information (Overview)

Field	**Type**	**Example**
Version Number	Integer	3
Serial Number	Integer	4567
Signature Algorithm	Object Identifier	1.2.840.113549.1.1.5
Issuer	Distinguished Name	CN=VeriSign..., OU=VeriSign Trust Network, O="VeriSign, Inc.", C=US
Validity	Date/Time	Not Before: 02.03.2022 01:00:00, Not After: 02.03.2032 00:59:59
Subject Name	Distinguished Name	CN=e-finance.postfinance.ch, O=Die Schweizerische Post, L=Bern, ST=Bern, C=CH
Subject Public Key Info	Bit String	Subject Public Key Algorithm: PKCS #1 RSA Encryption Subject's Public Key: Modulus (2048 bits) 4b 44 0c b6 e7 3f 3a 73 47 6f 81 51... Exponent (24 bits): 65537

X.509 Certificate in ASN.1 Notation

```
Certificate ::= SEQUENCE {
  tbsCertificate TBSCertificate,
  signatureAlgorithm AlgorithmIdentifier,
  signatureValue BIT STRING }
```

tbsCertificate This part holds the information which will be signed

signatureAlgorithm The algorithm that is used for signing the TBS part

signatureValue The calculated signature

Version

```
Version ::= INTEGER {v1(0), v2(1), v3(2)}
```

Indicates the version of the certificate. Today, basically only Version 3 certificates are in use.

Serial Number

```
CertificateSerialNumber ::= INTEGER
```

The unique identifier for this certificate relative to the certificate issuer. This serial number is essential when revoking a certificate.

- Positive integer
- Must be unique for the same issuer. Two certificates from the same issuer are not allowed to have the same serial number.

Signature Algorithm

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL}
```

Specifies the algorithm that was used to sign the certificate (e.g. SHA-256 with RSA).

algorithm: an OID (e.g. 1.2.840.113549.1.1.11)

parameters: any needed parameters (like the elliptic curve to be used in ECDSA)

This is specified twice in "Standard Information" and in "Signature". Both MUST be identical.

Issuer

Holds the Distinguished Name (DN) in X.500 notation of the CA that issued the certificate:

```
CN = Thawte Personal Freemail Issuing CA,
O = Thawte Consulting (Pty) Ltd, C = ZA
```

Must always be present!

Validity

```
Validity ::= SEQUENCE {
  notBefore Time,
  notAfter Time  }
```

The window of time that this certificate should be considered valid unless otherwise revoked. This field is composed of *Not Valid Before* and *Not Valid After* dates/times that are represented in UTC Time.

Subject Name

Holds the DN in X.500 notation of the certificate owner that the public key will be associated with:

```
E  = hans.muster@gmx.com
CN = Hans Muster
G  = Hans
SN = Muster
```

The same DN is not allowed to be given to different entities.

Must always be present!

General Subject Public Key Info

Holds the algorithm (identifier) and the public key of the entity (subject name):

```
SubjectPublicKeyInfo ::= SEQUENCE {
  algorithm AlgorithmIdentifier,
  subjectPublicKey BIT STRING }
```

where

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL }
```

Must always be present!

Example: RSA Public Key

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm 1.2.840.113549.1.1.1 -- RSA
  parameters NULL
}
SubjectPublicKey ::= SEQUENCE {
  modulus INTEGER, -- n
  publicExponent INTEGER -- e
}
```

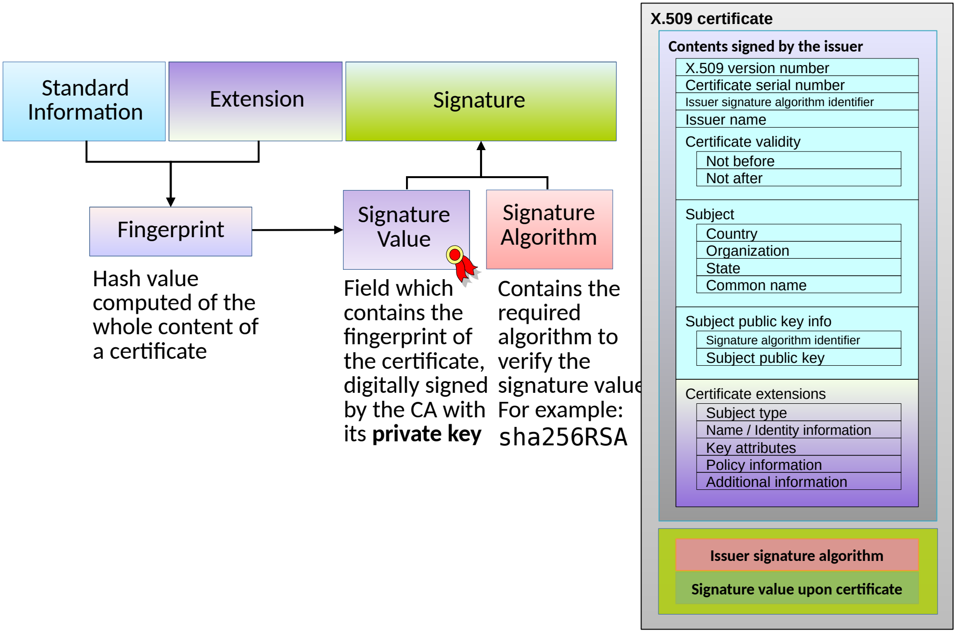
Example: DSA Public Key

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm 1.2.840.10040.4.1 -- DSA
  parameters DSS-Params
}
DSS-Params ::= SEQUENCE {
  p INTEGER, (prime modulus)
  q INTEGER, (prime divisor of p-1)
  g INTEGER  (generator)
}
SubjectPublicKey ::= BIT STRING {
  publicExponent INTEGER
}
```

Example: ECC Public Key

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm 1.2.840.10045.2.1 -- EC Public Key
  parameters 1.3.132.0.10 -- 256-bit Koblitz: secp256k1
}
SubjectPublicKey ::= BITSTRING {
  ECPoint OCTET STRING}
```

Tamperproofing X.509 certificates



Part IV: X.509 Certificate Extensions

X.509 Certificate Extensions

- Beside standard information, a X.509 certificate may include optional parameters (extensions).
- These extensions provide additional information to the *subject*, the *key material*, the *certificate usage* and the *issuer* of the certificate.

An extension can be marked as CRITICAL (usually extension “keyUsage” only). In this case an application must be able to process and handle this extension, if not it must reject the **whole** certificate!

X.509v3 subjectAltNames

Indicates alternative name forms associated with the owner of the certificate. If the subject DN (Standard Information) is null, one or more alternative name forms must be present, and this extension must be marked CRITICAL.
Examples:

- IP:192.168.2.0
- DNS:www.example.com
- email:user@example.com

Email addresses must be subjectAltNames and should not be used for the subject distinguished name (DN)!

basicConstraints

Indicates if the subject may act as a CA. If so, a certification path length constraint may be specified. If the path length is not present then there is no limit!
To identify a CA Root Certificate (for signing certificates): the basicConstraints field should always be present, and the CA attribute in the basicConstraints field must be set to a value of TRUE and must be marked CRITICAL.
Examples:

- CA:TRUE; critical
- CA:TRUE; pathLenConstraint = 0
- CA:FALSE

Typically, this field is absent in end-entity certificates. If it is present in an end-entity certificate, the value of the CA attribute in the basicConstraints field must be FALSE.

keyUsage

Bit string used to identify (or restrict) the functions or services that can be supported by using the public key in this certificate.

Leaf:

- digitalSignature: Bit 0
- nonRepudiation: Bit 1
- keyEncipherment: Bit 2
- dataEncipherment: Bit 3
- keyAgreement: Bit 4

CA:

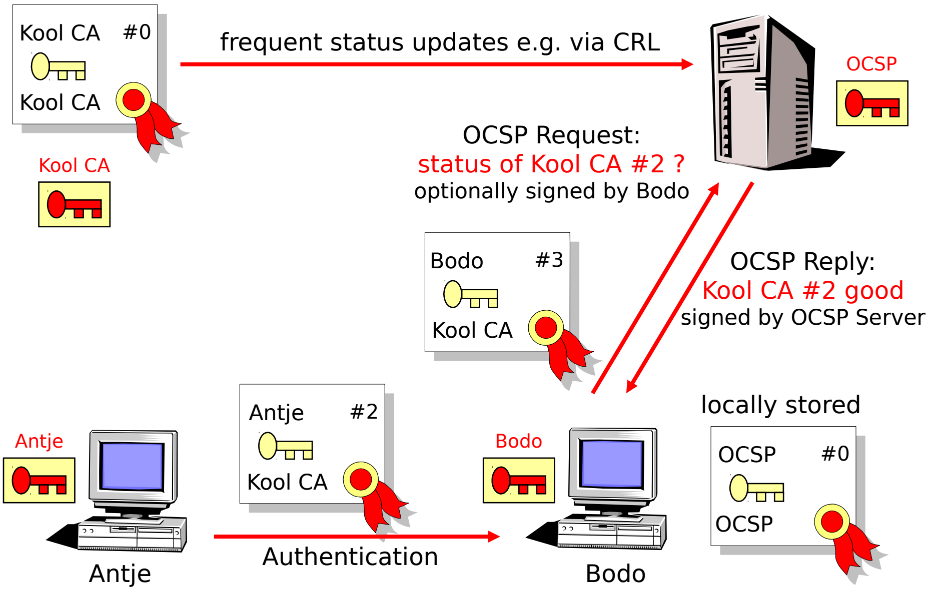
- certificateSign: Bit 5
- crlSign: Bit 6

Extended Key Usage (EKU)

Sequence of one or more Object Identifiers (OID) that identify specific usage of the certificate. OIDs associated with this extension can be:

- serverAuth – TLS Web Server Authentication
- clientAuth – TLS Web Client Authentication
- codeSigning – Code signing
- emailProtection – E-mail Protection
- timeStamping – Time stamping
- ocspSigning – Onlince Certificate Status Protocol (OCSP) signing

Online Certificate Status Protocol (OCSP)



Policy Information

Authority Key Identifier Identifies the public key that corresponds to the private key that has signed the certificate. This MUST be included in all certificates (non-critical), unless it is a self-signed certificate.

Subject Key ID Identifies certificates that contain a particular public key.[2] This MUST be included in all CA certificates (non-critical).

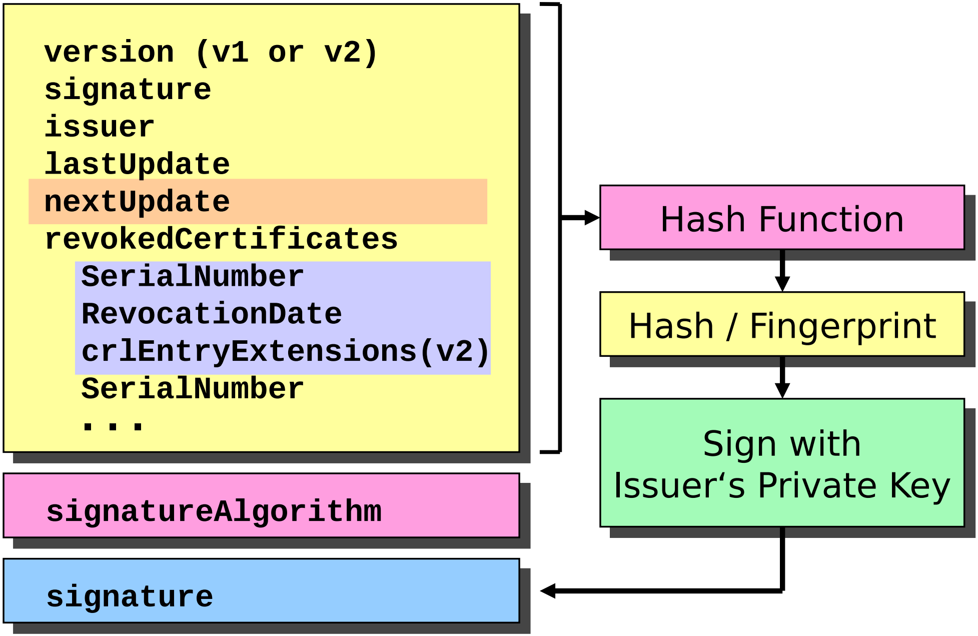
CRL Distribution Point Indicates the location of the CRL partition where revocation information associated with this certificate resides.

Certificate Policies The certificate policies extension contains a sequence of one or more policy information terms, each of which consists of an OID.

Private-key usage period Indicates the period of time to use the private key corresponding to the public key. For example, with digital signature keys, the usage period for the signing private key can be shorter than that for the signature verifying public key.

[2] Basically a hash of the subject's public key.

X.509v3 crlDistributionPoints



Additional Information

Authority Information Access:

A private extension present in end-entity and CA certificates, indicates how information or services offered by the issuer of the certificate can be obtained. Two access method OIDs have been defined:

- CA Issuers: This information can be used to help build certification paths or other information of services of the issuing CA (other policies, root certificates). Especially if applications like Secure Mail (S/MIME) or SSL/TLS do not receive enough information to build the whole certification path (missing certificate of an intermediate CA) this is a possibility to find the certificate of the issuing CA.
- OCSP Validation Service: This access method is for on-line validation services based on OCSP.

Additional Information

Subject Information Access:

A private extension present in end-entity and CA certificates, indicates how information and services offered by the subject in the certificate can be obtained.

One access method OID has been defined for CAs:

- CA Repository: This CA Repository access method identifies the location of the repository where the CA publishes certificate and CRL information

One access method OID has been defined for end entities:

- Time stamping: The Time-Stamping access method indicates that the subject identified in the certificate offers a time stamping service

Identity Certificates vs. Attribute Certificates

- Any change to the information contained within a given certificate - before it naturally expires - necessarily means that the existing certificate must be revoked and a new certificate must be issued.
- Therefore, attributes placed within the certificate should be fairly static in order to avoid wasteful certificate revocation and (re)issuance.
- Attributes associated with an ‘end entity’ that has a dynamic environment, should be conveyed through attribute certificates.
- Attribute certificate helps to separate authentication (identity) and authorization (permission).

Contents of Attribute Certificates

An attribute certificate (AC) is:

- a certified set of attributes like temporary network access privileges, electronic letters of credit access, signing and purchasing authority, etc.
- much the same syntax as a X.509v3 certificate (without a public key)
- bound to a serial number of an identity certificate
- typically have a much shorter lifetime than identity certificates
- grants temporary authorization to individuals

Individuals can hold several attribute certificates:

- each referencing the same identity certificate
- each issued by a different attribute authority (AA)
- and use them selectively according to the resource or service being accessed

Part V: Creating a Certificate

Certificate Authorities

- Entities that *claim* to be trustworthy to verify identities and issuing public key certificates (“let’s encrypt”)
- CAs can be organized into a directed graph
- X.509: Tree depth can be limited for a subtree
- X.509: Certificates of CAs signing intermediate-level CAs have the special “CA” bit set

Self-signed certificates

- Signer is self
- Allowed by TLS
- Used to sign CA tree roots

CA: creates a self-signed certificate

```
# create certificate:
$ openssl req -x509 -out cert.pem -outform PEM -days 3650
# private key will now be in privkey.pem
# convert to certificate request:
$ openssl x509 -x509toreq -in cert.pem -out req.pem \
    -signkey privkey.pem
# generate config
$ cp /usr/lib/ssl/openssl.cnf .
# self-sign using:
$ openssl x509 -req -in req.pem -extfile openssl.cnf \
  -extensions v3_ca  -signkey privkey.pem -out selfcert.pem
# view using:
$ openssl x509 -in cacert.pem -text -noout
```

PEM encoding is Base64 of DER bytestream with “begin certificate” and “end certificate” markers.

Client: creates a certificate request

```
# create private key using:
$ openssl genpkey -algorithm RSA -out key.pem \
  -aes-128-cbc -pkeyopt rsa_keygen_bits:2048
# create CSR using:
$ openssl req -new -key key.pem -keyform PEM \
  -out req.pem -outform PEM
```

CA: signs certificate request

```
# Prepare CA directory structure
$ wget https://grothoff.org/christian/teaching/ca.conf
$ mkdir dir certdir
$ touch dir/index.txt dir/index.txt.attr
$ echo 1 > dir/serial.txt
# sign CSR using:
$ openssl ca -in req.pem -out cert.pem -config ca.conf
```

Free certificates

- WoSign, 2 years validity
- Let's encrypt (EFF, Mozilla), 3 months validity, auto-renewal
- FreeSslCertificate, only for non-profits
- MeSign, e-mail certificates only
- Codegic, 2 months validity, incl. e-mail, code signing, document signing, client/server authentication and timestamping

Part VI: Example: Protocol vulnerability

Guiding questions “Making the Theoretical Possible”

- What is the root cause of the vulnerability exploited in the attack?
- What does the attack achieve?
- Summarize the attack (how does it work?, capture every step!)
- Comment on the different “levels” of breaking a hash function (i.e. what is achieved in the attack that goes beyond finding an arbitrary collision).

MD5: Making the Theoretical Possible

25c3, 2008

Part VII: Card Verifiable (CV) Certificates for smartcards

X.509 Certificates and Smartcards

- In many cases a smartcard must be able to securely identify its communication partner (e.g. the reader or another card) in order to grant access to stored data on card.
- Due to the reduced storage and memory resources of a smartcard, X.509 certificates are often too large and operation may be too complex.
- The ISO/IEC7816-8 Standard defines a reduced syntax for certificates, which are specially designed to be processed by smartcards.

These certificates are called Card Verifiable Certificates (CVC).

Properties of CVC

- CVC are not compatible with X.509 certificates.
- CVC are used for authentication only.
- CVC are encoded in Type-Length-Value (TLV) format.

The minimal set of attributes is:

- Issuer
- Subject
- Public key
- Access rights
- Validity period

Properties of CVC

To save space and time only the signature value will be stored and transmitted rather than the signature value together with the plaintext content.
This is realized according to the ISO 9796 signature scheme with message recovery:

- ISO 9796 uses RSA (amongst others) as signature algorithm, which has message recovery properties.
- The message itself is only readable/recoverable by verifying the signature.
- In other words: the message is stored/transmitted “encrypted” with the private key and can be decrypted using the public key.

Typical implementations of CVC

- eIDAS-Token
- nPA in Germany
- ICAO Passport
- Versichertenkarte/Gesundheitskarte (VK)
- Health Professional Card (HPC)

Part VIII: Types of Certificates

Types of certificates

Roughly, we legally distinguish:

- 4 levels of user certificates
- 3 levels of server certificates

User Certificates

- **Class 4 (class D, bronze, ...) certificates**: No identity check (at most per e-mail address only) purpose: digital signature, authentication, encipherment
- **Class 3 (class C, silver, rubin, ...) certificates**: low level identity check purpose: digital signature, authentication, encipherment
- **Class 2 (class B, gold, saphir, ...) certificates**: high level identity check of natural person and enterprises, called: advanced (regulated) certificates purpose: digital signature, authentication, [encipherment]
- **Class 1 (class A, platin, diamond, ...) certificates**: highest level identity check, for natural person only mandatory for qualified certificates purpose: non-repudiation only

What is a qualified certificate?

- The term has been defined by European Commission: EU Regulation 910/2014: eIDAS — regulation on electronic identification and trust-services for electronic transactions
- The IETF PKIX Working Group has specified in RFC 3739 the format of a "Qualified Certificate".
- Primary purpose is to identifying a person with highest level of assurance for non-repudiation services.
  - In Switzerland according to ZertES and TAV of BAKOM (see: Documents for Electronic Signatures)
  - See also: The e-Government Standard for certificate classifications (eCH-0048 Zertifikatsklassen)

Server Certificates

- **Domain Validation (DV):** Automated issuing, WHOIS check & simple confirmation (e.g. by e-mail, local agent, etc.) $\rightarrow$ `https://letsencrypt.org`
- **Organization Validation (OV):** Average examination of claimant. Provides higher level of trustworthiness, thanks to better validation of organization.
- **Extended Validation (EV):** Strong identity and authenticity verification based on CAB-Forum guidelines, operated by WebTrust for auditing CA's to be accredited for EV-Certificate issuance.

“Self-regulation”: The Certificate Authority Browser (CAB) Forum

The CAB forum is a voluntary group of CA’s and Browser suppliers with the goal to establish and enforce standards for certificate issuance and processing.

- Standardized Extended Validation Certificates (EV).
- Released „Baseline Requirements for Issuance and Management of Publicly-Trusted Certificates“[3] which has been incorporated by WebTrust audit program for CA’s.

Topics: Identity vetting, certificate content, audit requirements, cryptographic algorithms and key sizes, liability and delegation of authority.

[3] https://cabforum.org/baseline-requirements-documents

Server certificates: Extended Validation (EV)

- Extended Validation Certificate (EV) are X.509 certificates issued by a CA only after a regulated verification of the entity's identity.
- EV Certificates are structurally not different from other X.509 certificates, but have additional subject information and a policy identifier (OID 2.23.140.1.1).
- The criteria for issuing EV certificates are defined in the "Guidelines for the Issuance and Management of EV Certificates"[4] produced by the CA/Browser Forum.
- WebTrust for Extended Validation is used to assess CA's controls against the CA/B Forum guidelines. Only suitably certified CA's may issue EV. WebTrust for EV requires an annual audit.

[4] https://cabforum.org/extended-validation

Part IX: Exercise: Reading a Certificate

Reading a Certificate (1/3)

Use your favorite browser to have a deeper look at the `https://intranet.bfh.ch/` X.509 web-server certificate.

1. Which Certificate Authority (CA) has signed this server certificate?
2. How does the certification path of this X.509 certificate looks like?
3. Where in the server certificate may your browser find the location of the certificate of the signing CA?
4. For what purpose can we use the key in the server certificate?
5. What is the meaning of the flag CRITICAL?

Reading a Certificate (2/3)

1. According to [3, Section 4.2.1.12] the extended key usage extension indicates one or more purposes for which the certificate may be used, in addition to the purpose(s) defined in the key usage extension. Which extended key usage OID's are defined in this server certificate and what are the purpose of them?
2. For which other URL's (FQDN) your browser might accept this server certificate?
3. From which server may your browser request the status of this certificate online? What information does it need to preform such a query?
4. Where do we find information about CP/CPS published by the issuer?

Reading a Certificate (3/3)

Use your favorite browser to have a deeper look at the `https://www.google.com/` and `https://www.ubs.ch/` X.509 web-server certificate.

1. Which one is an EV-certificate? Which are the indicators defining the EV-level of the certificate?
2. What is the Object Identifier (1.3.6.1.4.1.11129.2.4.2) in one of these certificates for?
3. What is the length of the public key of the web-server certificate and the signature of the issuing CA?

References I

- L. Bassham, W. Polk, and R. Housley.
  Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
  RFC 3279 (Proposed Standard), April 2002.
  Updated by RFCs 4055, 4491, 5480, 5758.

- J. Callas, L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer.
  OpenPGP Message Format.
  RFC 4880 (Proposed Standard), November 2007.
  Updated by RFC 5581.

- D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk.
  Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
  RFC 5280 (Proposed Standard), May 2008.
  Updated by RFC 6818.

- Q. Dang, S. Santesson, K. Moriarty, D. Brown, and T. Polk.
  Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA.
  RFC 5758 (Proposed Standard), January 2010.

References II

- S. Farrell and R. Housley.
  An Internet Attribute Certificate Profile for Authorization.
  RFC 3281 (Proposed Standard), April 2002.
  Obsoleted by RFC 5755.
- S. Leontiev and D. Shefanovski.
  Using the GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms with the Internet X.509 Public Key Infrastructure Certificate and CRL Profile.
  RFC 4491 (Proposed Standard), May 2006.
- J. Schaad, B. Kaliski, and R. Housley.
  Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
  RFC 4055 (Proposed Standard), June 2005.
  Updated by RFC 5756.
- S. Turner, D. Brown, K. Yiu, R. Housley, and T. Polk.
  Elliptic Curve Cryptography Subject Public Key Information.
  RFC 5480 (Proposed Standard), March 2009.

References III

P. Yee.
Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.
RFC 6818 (Proposed Standard), January 2013.