The X.509 standard, PKI and electronic documents

Antonio Lioy
< lioy@polito.it >

Politecnico di Torino
Dipartimento di Automatica e Informatica

Certification Authority

Certification Authority

PC

repository (cert, CRL)

Registration Authority

Anna

(1) Kpub, Anna

(1) Kpri

(4) cert (Anna,Kpub)

(3) Anna OK

(2) Anna OK

(4) cert
X.509 certificates

- standard ITU-T X.509:
  - v1 (1988)
  - v2 (1993) = minor
  - v3 (1996) = v2 + extensions + attribute certificate v1
  - v3 (2001) = v3 + attribute certificates v2
- is part of the standard X.500 for directory services (white pages)
- is a solution to the problem of identifying the owner of a cryptographic key
- definition in ASN.1 (Abstract Syntax Notation 1)

X.509 version 3

- standard completed in June 1996
- groups together in a unique document the modifications required to extend the definition of certificate and CRL
- two types of extensions:
  - public, that is defined by the standard and consequently made public to anybody
  - private, unique for a certain user community
Critical extensions

- an extension can be defined as critical or non-critical:
  - in the verification process the certificates that contain an unrecognized critical extension MUST be rejected
  - a non-critical extension MAY be ignored if it is unrecognized
- the different (above) processing is entirely the responsibility of the party that performs the verification: the Relying Party (RP)

Public extensions

- X.509v3 defines four extension classes:
  - key and policy information
  - certificate subject and certificate issuer attributes
  - certificate path constraints
  - CRL distribution points
Key and policy information

- authority key identifier
- subject key identifier
- key usage
- private key usage period
- certificate policies
- policy mappings

Key usage

- identifies the application domain for which the public key can be used
- can be critical or not critical
- if it is critical then the certificate can be used only for the scopes for which the corresponding option is defined
Key and policy information

- **key usage** – the applications that can be defined are:
  - digitalSignature (CA, user)
  - nonRepudiation (user)
  - keyEncipherment (user)
  - dataEncipherment
  - keyAgreement (encipherOnly, decipherOnly)
  - keyCertSign (CA)
  - cRLSign (CA)

Certificate subject and certificate issuer attributes

- subject alternative name
- issuer alternative name
- subject directory attributes
Certificate subject and certificate issuer attributes

- subject alternative name
  - allows to use different formalisms to identify the owner of the certificate (e.g. e-mail address, IP address, URL)
  - always critical if the field subject-name is empty

X.509 alternative names

- various possibilities:
  - rfc822Name
  - dNSName
  - IPAddress
  - uniformResourceIdentifier
  - directoryName
  - X400Address
  - ediPartyName
  - registeredID
  - otherName
Certificate path constraints

- basic constraints
  - indicates if the subject of the certificate can act as a CA:
    - BC=true: the subject is a CA
    - BC=false: the subject is an EE (End Entity)
  - furthermore it is possible to define the maximum depth of the certification tree (only if BC=true)
  - critical or non critical
  - it is suggested to always mark this extension as critical
Certificate path constraints

- name constraints
  - only for CA
  - space of names that can be certified by a CA
  - critical or non critical

CRL distribution point

- CRL distribution point
  - identifies the distribution point of the CRL to be used in validating a certificate
  - can be:
    - directory entry
    - e-mail or URL
    - critical or non critical
Private extensions

- It is possible to define private extensions, that is, extensions common to a certain user community (i.e., a closed group).
- For example, IETF-PKIX defined three private extensions for the Internet user community:
  - Subject information access
  - Authority information access
  - CA information access

PKIX private extensions

- Authority information access
  - Indicates how to access information and services of the CA that issued the certificate:
    - CertStatus (e.g., URL for OCSP)
    - CertRetrieve
    - cAPolicy
    - caCerts
  - Critical or not critical
Extended key usage

- in addition or in substitution of keyUsage
- possible values:
  - (id-pkix.3.1) serverAuth [DS, KE, KA]
  - (id-pkix.3.2) clientAuth [DS, KA]
  - (id-pkix.3.3) codeSigning [DS]
  - (id-pkix.3.4) emailProtection [DS, NR, KE, KA]
  - (id-pkix.3.8) timeStamping [DS, NR]

CRL X.509

- Certificate Revocation List
- list of revoked certificates
- CRLs are issued periodically and maintained by the certificate issuers
- CRLs are digitally signed:
  - by the CA that issued the certificates
  - by a revocation authority delegated by the (indirect CRL, iCRL)
CRL X.509 version 2

CertificateList ::= SEQUENCE {
    tbsCertList TBSCertList,
    signatureAlgorithm AlgorithmIdentifier,
    signatureValue BIT STRING }

TBSCertList ::= SEQUENCE {
    version Version OPTIONAL,  -- if present, version must be v2
    signature AlgorithmIdentifier,
    issuer Name,
    thisUpdate Time,
    nextUpdate Time OPTIONAL,
    revokedCertificates SEQUENCE {
        userCertificate CertificateSerialNumber,
        revocationDate Time,
        crlEntryExtensions Extensions OPTIONAL
    } OPTIONAL,
    crlExtensions [0] Extensions OPTIONAL
}

Extensions of CRLv2

- crlEntryExtensions:
  - reason code
  - hold instruction code
  - invalidity date
  - certificate issuer

- crlExtensions:
  - authority key identifier
  - issuer alternative name
  - CRL number
  - delta CRL indicator
  - issuing distribution point
Certificate revocation timeline

OCSP

- RFC-2560: On-line Certificate Status Protocol
- IETF-PKIX standard to verify online if a certificate is valid:
  - good
  - revoked
    - revocationTime
    - revocationReason
  - unknown
- response signed by the server (not by the CA!)
- the OCSP server certificate cannot be verified with OCSP itself!
Architecture of OCSP

- possible pre-computed responses
  - decreases the computational load on the server … but makes possible replay attacks!
- possible to obtain information not from CRL

Models of OCSP responder

- Trusted Responder
  - the OCSP server signs the responses with a pair key:cert independent of the CA for which it is responding
  - company responder or TTP paid by the users
- Delegated Responder
  - the OCSP server signs the responses with a pair key:cert which is (can be) different based on the CA for which it is responding
  - TTP paid by the CA
Time-stamping

- proof of creation of data before a certain point in time
- TSA (Time-Stamping Authority)
- RFC-3161:
  - request protocol (TSP, Time-Stamp Protocol)
  - format of the proof (TST, Time-Stamp Token)

PSE (Personal Security Environment)

- each user should protect:
  - his own private key (secret!)
  - the certificates of the trusted root CAs (authentic!)
- software PSE:
  - (encrypted) file of the private key
- hardware PSE:
  - passive = protected keys (same as sw PSE)
  - active = protected keys + crypto operations
- mobility is possible in both cases (but with problems)
Cryptographic smart-card

- chip cards with memory and/or autonomous cryptographic capacity
- simple: DES
- complex: RSA
  - length of the key?
  - generation of the private key on board?
- few memory (EEPROM): 4 - 32 Kbyte

HSM (HW Security Module)

- cryptographic accelerator for servers
  - secure storage of private key
  - autonomous encryption capabilities (RSA, sometimes symmetric algorithms too)
- form factor: PCI board or external device (USB, IP, SCSI, …)
Security API (low level)

- PKCS-11 = (only) crypto engine
  - in software
  - in hardware
    - smart card
    - cryptographic card
  - part of the CDSA architecture
- MS-CAPI CSP (Crypto Service Provider)
  - same functions as PKCS-11 but proprietary API of MS

Secure data formats

- PKCS-7 = secure envelope
  - signed and/or encrypted
- PKCS-10 = certificate request
  - used in the communication among the client and CA / RA
- PKCS-12 = software PSE (Personal Security Environment)
  - transport of keys and certificates
- are not application formats:
  - S/MIME? IDUP-GSS-API? XML-DSIG?
  - legal electronic documents?
PKCS-7 and CMS formats

- Cryptographic Message Syntax
- PKCS-7 is the RSA standard for secure envelope (v1.5 is also RFC-2315)
- CMS is the evolution of PKCS-7 inside IETF
- allows signing and/or encryption of data, with symmetric or asymmetric algorithms
- supports multiple signatures (hierarchical or parallel) on the same object and can include the certs (and revocation info) to verify the signature
- is a recursive format
- syntax based on ASN.1-BER (DER solo per “signed attributes” e “authenticated attributes”)

Evolution of CMS

- RFC-2630 (jun’99)
  - compatible with PKCS-7 1.5
  - adds key-agreement and pre-shared keys
- RFC-3369 (aug’02)
  - adds pwd-based keys and an extension schema for generic key management
  - algorithms specified in a distinct RFC
- RFC-3852 (jul’04)
  - extension to support generic certificates
- RFC-5652 (sep’09)
  - clarifications about multiple signatures
Algorithms for CMS (I)

- RFC-3370 = base algorithms
  - digest MD5, SHA-1
  - signature RSA, DSA
  - key management
    - agreement = DH
    - transport = RSA
    - symmetric wrapping = 3DES, RC2
    - derivation = PBKDF2
  - content encryption = 3DES-CBC, RC2-CBC
  - MAC = HMAC-SHA1

Algorithms for CMS (II)

- encryption: (RFC-2984) CAST-128, (3058) IDEA, (3565) AES, (3657) Camellia, (4610) SEED
- RFC-4056 = RSASSA-PSS for digital signature
- RFC-4490 = GOST for encryption and digest
- RFC-5084 = AES-CCM and AES-GCM for auth.enc.
- RFC-5409 = Boneh-Franklin and Boneh-Boyen for Identity-Based Encryption
- RFC-5753 + RFC-6161 = ECC
- RFC-5754 = SHA-2
- key transport: (5990) RSA-KEM, (3560) RSAES-OAEP
PKCS-7: contentInfo

```
<table>
<thead>
<tr>
<th>contentType</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

PKCS-7: contentType

- **data**
  - encoding of a generic sequence of bytes
- **signedData**
  - data + parallel digital signatures (1..N)
- **envelopedData**
  - data encrypted symm. + key encrypted with RSA
- **signedAndEnvelopedData**
  - RSA encryption of (data + digital signatures)
- **digestData**
  - data + digest
- **encryptedData**
  - data encrypted with a symmetric algorithm
PKCS-7: signedData

```
signedData
  content
    version
    digestAlgorithm
    contentInfo
    certificates
    cRLs
    signerInfo

version
issuer + SN
encryptedDigest
```

PKCS-7: envelopedData

```
envelopedData
  content
    version
    encryptedContentInfo
    recipientInfo
    ... 
    recipientInfo

contentType
encryptionAlgorithm
encryptedContent
```

© A.Lioy - Politecnico di Torino (1997-2011)
PKCS-10

**data to `be certified**

- DN
- public key
- attributes

**private key of the entity to be certified**

**computation of signature**

**signature**

**PKCS#10**

---

**PKCS #10**

- RFC-2986 = PKCS #10 (v 1.7)
- RFC-5967 = application/pkcs10 media type
- format for a certificate request
- the request contains
  - DN + public key + (optional) attributes
- possible attributes:
  - a “challenge password” (for registration or revocation)
  - attributes to be inserted in the certificate (e.g. those described in PKCS #9)
  - other information about the requestor
PKCS-12 format (security bag)

- transport of (personal) cryptographic material among applications / different systems
- transports a private key and one or more certificates
- transports the digital identity of a user
- used by Netscape, Microsoft, Lotus, …
- criticized from the technical point of view (especially in the MS implementation) but widely used

Formats of signed documents

- signed data
- document
- signature
- enveloping signature (es. PKCS-7)

- document
- data
- signature
- enveloped signature (es. PDF)

- document
- signature
- detached signature (es. PKCS-7)
Multiple signatures (parallel / independent)

Multiple signatures (sequential / hierarchical)
EU Electronic Signature (ES)

- data in electronic form which are attached to or logically associated with other electronic data and which serve as a method of authentication

- BEWARE: a scanned signature is an Electronic Signature (!)

Advanced Electronic Signature (AES)

- an ES which meets the following requirements:
  - uniquely linked to the signatory
  - capable of identifying the signatory
  - created using means that the signatory can maintain under his sole control
  - linked to the data to which it relates in such a manner that any subsequent change of the data is detectable
Qualified Certificate (QC)

- a PKC certifying the identity of a person and containing:
  - an indication that it was issued as a QC
  - the name of the signatory or a pseudonym, which shall be identified as such
  - provision for a specific attribute of the signatory to be included if relevant, depending on the purpose for which the certificate is intended
  - limitations on the scope of the certificate, if any
  - limits on the value of transactions, if any
- RFC-3739 = IETF-PKIX profile for QC

Qualified Electronic Signature (QES)

- an AES (a) based on a QC, and (b) created by a secure-signature-creation device
- satisfies the legal requirements of a signature in relation to data in electronic form in the same manner as a handwritten signature satisfies those requirements in relation to paper-based data
- is admissible as evidence in legal proceedings
Legal effects

Member States shall ensure that an electronic signature is not denied legal effectiveness and admissibility as evidence in legal proceedings solely on the grounds that it is:

- in electronic form, or
- not based upon a qualified certificate, or
- not based upon a qualified certificate issued by an accredited certification-service-provider, or
- not created by a secure signature-creation device

ETSI standards for electronic signature

CMS Advanced Electronic Signatures (CAdES)
- ETSI TS 101 733 (version 1.4.0)
- ETSI TS 102 734 = profiles of CAdES

based upon other standards:
- RFC-2630 [CMS] Cryptographic Message Syntax
- RFC-2634 [ESS] Enhanced Security Services

“raw” signature format (i.e. binary over a blob)
- evolution to application formats (XML and PDF)

www.etsi.org/WebSite/Technologies/ElectronicSignature.aspx
ETSI: CAdES formats

and the extended formats ES-X …

Extended ES (ES-X)

- if the CA certificates may be compromised, then the formats ES-X are suggested
- ES-X-Timestamp (type 1):
  - ES-C with a TS over the whole ES-C
  - useful when OCSP is used
- ES-X-Timestamp (type 2):
  - ES-C with a TS over just the references to the certificates and the revocation informations
  - useful when CRL is used
TSL

- TSL = Trust service Status List
  - contains TSP (Trust-Service Provider)

- signed list
  - list of the TSP and their services (certification, revocation, time-stamping, …)
  - state of each TSP (supervised, suspended, revoked, …)
  - history of the state of each TSP
  - schema and schema operator

- “white list” for the accredited TSP
- “black list” for the not accredited TSP

Other ETSI ES formats

- XML Advanced Electronic Signatures (XAdES)
  - ETSI TS 101 903
  - ETSI TS 102 904 = profiles of XAdES
  - based upon XML-dsig

- PDF Advanced Electronic Signature Profiles (PAdES) for the ISO-32000 format (PDF)
  - ETSI TS 102 778-1 = overview
  - ETSI TS 102 778-2 = basic
  - ETSI TS 102 778-3 = enhanced (BES, EPES)
  - ETSI TS 102 778-4 = long-term validation (LTV)
  - ETSI TS 102 778-5 = XML content
The “macro” problem

- e-signing an e-document containing a macro is a bad idea

WYSIWYS

- What You See Is What You Sign
- highly desirable
- is a problem of the application developers
- in Austria, it is a fundamental requirement of the law about e-signatures and e-documents

- do we really need it? compare it to fine prints in paper documents