

E-documents

Antonio Lioy
< lioy @ polito.it >

Politecnico di Torino (Italy)
Dip. Automatica e Informatica

E-documents

- not just an electronic transposition of a document ...
- ... rather a data format:
 - secure
 - standard
 - open (i.e. can be read in 30 years because the specification is published)
- e.g. ASCII, Postscript, PDF, XML, ...
- ... encapsulated in a secure envelope

E-signature

- usually: e-document = doc.data + e-signature

The diagram shows two overlapping circles. The left circle is labeled 'digital signatures' and the right circle is labeled 'e-signatures'. The overlapping area in the center is labeled 'e-signatures based on digital signatures'.

Parallel / distributed / independent signatures

The diagram shows three separate document boxes. Each box contains a 'doc' and a signature 'es(doc, X)', 'es(doc, Y)', or 'es(doc, Z)'. Arrows point from each box to a corresponding user icon (X, Y, Z) sitting at a computer workstation.

Hierarchical signatures

The diagram shows a hierarchical signing process. User X signs a document 'doc' to create 'es(doc, X)'. User Y signs this to create 'es(-, Y)'. User Z signs this to create 'es(-, Z)'. Arrows point from each final document box to the user who signed it.

Signed document formats

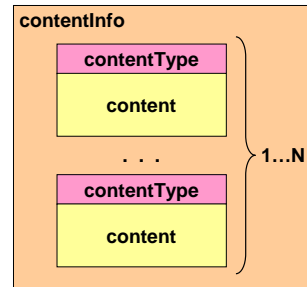
The diagram compares three signed document formats:

- enveloping signature (e.g. PKCS-7):** A box labeled 'signed data' contains a 'document' box and a 'signature' box.
- enveloped signature (e.g. PDF):** A box labeled 'document' contains a 'document data' box and a 'signature' box.
- detached signature (e.g. PKCS-7):** A 'document' box and a 'signature' box are separate.

PKCS-7

- PKCS-7 was the RSA standard for secure envelope (v1.5 = RFC-2315)
- format for data **signature and/or encryption**, with symmetric or asymmetric techniques:
 - can host multiple signatures (hierarchical or parallel)
 - can carry certificates and CRL
 - is a recursive format
- CMS (Cryptographic Message Syntax, RFC-2630) is the IETF evolution of PKCS-7

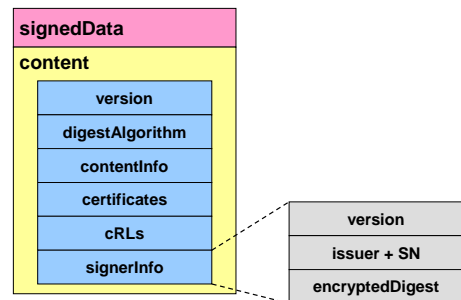
PKCS-7: general organization



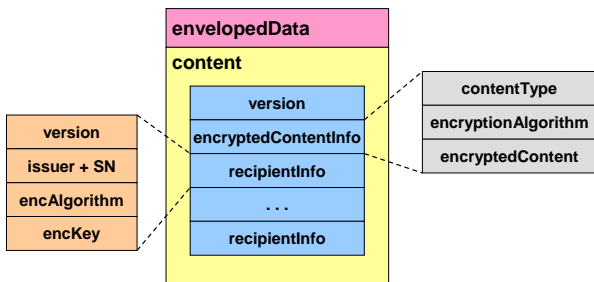
PKCS-7: contentType

- **data**
encoding of a generic byte sequence
- **signedData**
data + digital signatures (1..N, parallel)
- **envelopedData**
symm. encrypted data + RSA-encrypted key
- **signedAndEnvelopedData**
RSA encryption of (data + digital signatures)
- **digestData**
data + digest
- **encryptedData**
symmetrically encrypted data

PKCS-7: signedData



PKCS-7: envelopedData



CMS vs. PKCS-7

- five formats equal to PKCS-7
- signedAndEnvelopedData cancelled:
 - equivalent to signedData + envelopedData
- new type authenticatedData (uses a MAC for data authentication)
 - HMAC is supported
- envelopedData now supports key exchange via symm. techniques or key agreement
- signedData can use SubjectKeyIdentifier
- DH and DSA added to RSA

The S/MIME format

- originally defined to protect MIME e-mail, but now used as a general application format
- based on PKCS-7 (S/MIMEv2) or CMS (S/MIMEv3)

signed	signed and encrypted	encrypted
text Excel sheet Word document S/MIME digital signature	text Excel sheet Word document S/MIME digital signature S/MIME encrypted envelope	text Excel sheet Word document S/MIME encrypted envelope

S/MIME: an example

```

Content-Type: multipart/signed;
protocol="application/x-pkcs7-signature";
micalg=sha1;
boundary="-----aaaaa"


-----aaaaa
Content-Type: text/plain
Content-Transfer-Encoding: 7bit

Hello!
-----aaaaa
Content-Type: application/x-pkcs7-signature
Content-Transfer-Encoding: base64


MIIN2QasDDSDwe/625dBxgdhdsf76rHfrJe65a4F
fvVSW2Q1eD+SfDs543Sdwe6+25dBxfdeR0eDsrs5
-----aaaaa
    
```

User expectation

Is it possible to create an interoperable signed e-document?




Yes, if you use card X with reader Y via application W ... and you own a QC from provider Z!



User perception

- perceived difference between signature and document
- "electronic signature? wonderful, so I can e-sign a blank e-document ..."

Antonio Liroy



ETSI work

- ETSI TS 101 733 (version 1.4.0)
- builds on other standards:
 - RFC-2630 [CMS] Cryptographic Message Syntax
 - RFC-2634 [ESS] Enhanced Security Services
- great richness of options
- current work towards a simplification ...
- ... while retaining richness of expressivity

ETSI ES formats

Electronic Signature (ES)

signature policy ID

other signed attributes

digital signature

timestamp over digital signature

ES-T

ES-C

complete certificate and revocation references

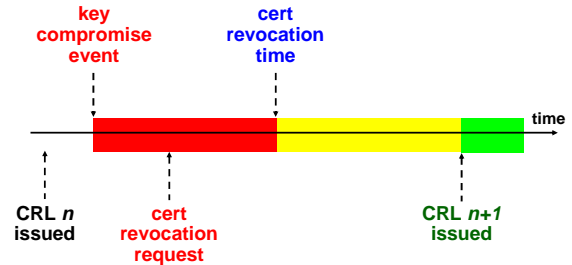
plus the ES-X formats ...

Timestamping

- attestation of signature time is important
 - to check that certificate is not revoked
 - to match against a deadline
- attestation can be:
 - inside the document itself (e.g. TST)
 - externally provided (e.g. by the receiving system)

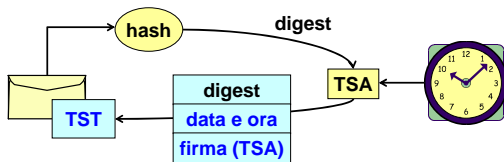


Certificate revocation timeline



Time-stamping

- proof of data existence at a certain time
- TSA (Time-Stamping Authority)
- RFC-3161:
 - C/S protocol (TSP, Time-Stamp Protocol)
 - proof format (TST, Time-Stamp Token)



WYSIWYS

- What You See Is What You Sign
- highly desirable
- it's a matter of the application developers
- do we really need it? let's compare it to fine prints in paper documents ...