Network security and management

Digital signature

Dieudonné KASONGO KABASELE, Nikola JOVICIC, Arnaud PALGEN, Yacine SAHLI

Electromagnetism and telecommunications department
University of Mons

UMONS POLYTECH MONS Faculté des Sciences

January 15, 2020

1 Introduction
- Our approach to the problem ...
- What is a Signature ?
- Digital Signature Function
- Difference between electronic and digital signature

2 Basic principles of digital signatures
- Diagram of the basic principle
- The hash function
- Public certificates

3 Existing solutions
- Adobe Sign
- GnuPg
- Microsoft Office: Word
- Online solutions
- LaTeX

4 Conclusions

Our approach to the problem ...

1. Understand what is fundamentally a signature and what it implies (conditions to respect)
2. Common points between electronic and digital signature
3. Explain the basic mechanisms behind digital signature
4. Find existing solutions and compare them to the basic principle
5. Compare solutions between them (pros and cons)

What is a Signature ?

There are two kinds :

- Electronic signature or e-signature
  refers to data in electronic form, which is logically associated with other data in electronic form and which is used by the signatory to sign or confirm their approval of a document or transaction.
- Digital signature
  These are a subset of electronic signatures because they are also in electronic form. They are a cryptographic mechanism often used to implement electronic signatures. Digital signatures go much further in terms of providing security and trust services.

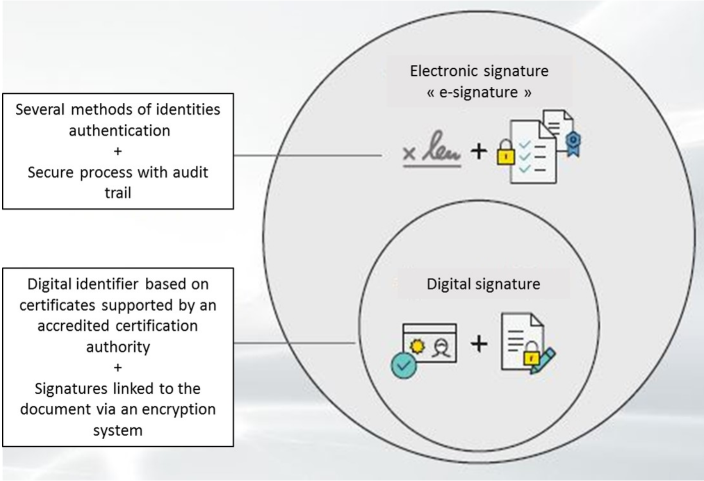
Source:https://www.4point.com/blog/2017/06/what_is_an_e-signatu.html

Digital Signature Function

- Properties
  -Identification
  -Integrity
- Conditions
  -Authentic: the identity of the signatory must be traceable with certainty;
  -Forgery-proof: the signature cannot be forged. Someone cannot pretend to be someone else;
  -Not reusable: It is part of the signed document and cannot be moved to another document;
  -Unalterable: Once it is signed, it cannot be modified;
  -Irrevocable: the person who signed cannot deny it.

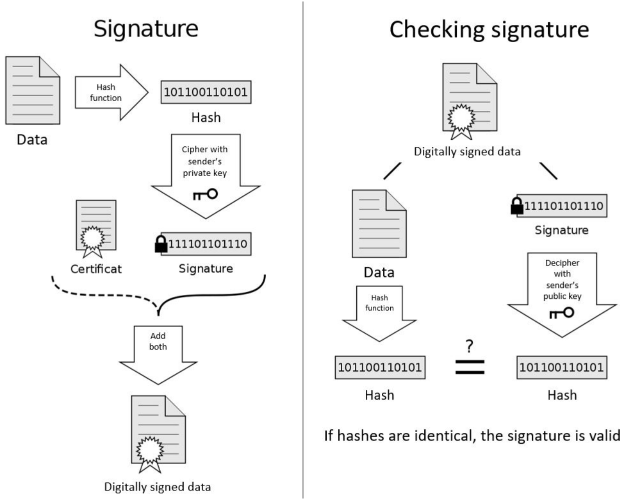
Source: `https://fr.wikipedia.org/wiki/Signature_num%C3%A9rique`

Difference between electronic and digital signature


Several methods of identities authentication
+
Secure process with audit trail
Digital identifier based on certificates supported by an accredited certification authority
+
Signatures linked to the document via an encryption system
Electronic signature
« e-signature »
Digital signature

Source: https://acrobat.adobe.com/content/dam/doc-cloud/fr/pdfs/adobe-sign-electronic-and-digital-signatures-wp-fr.pdf

Diagram of the basic principle



Source: https://fr.wikipedia.org/wiki/Signature_num%C3%A9rique

The hash function

- A hash function H takes a long string M as an input and outputs a different and shorter string *h*, called the hash
  - $h = \mathrm{H}(\mathrm{M})$
  - Example with SHA1 : "*Hello friend ! It is a beautiful day today, isn't it ?*" outputs "*73341cf4db5967d8ac42d5061d4f4b4c6ab666e1*"
  - The hash is usually expressed in an hexadecimal form
- Examples of a hash functions:
  - MD5, SHA1, SHA256, SHA384, ...
  - $h = H_{\oplus}(M) = M_1 \oplus M_2 \oplus M_3 \oplus \ldots \oplus M_N$
- **Key features:**
  - computing $h = \mathrm{H}(\mathrm{M})$ is easy but finding M from the hash should be very hard (in a reasonable time) !
  - a single change in the original in the original string shall output a completely different hash (e.g. : "Hello friend**s** ! ..." outputs "*aff42bd158ec3c095ca49f6d11c6f9961eb1de74*")

Public certificates

- Delivered by a **Certificate Authority**: trusted entity that issues and revokes public-key certificates and certifies the ownership of a public key by the "subject" of the certificate
- X.509 standard specified the format of these certificates:
  - Version: version of the certificate
  - Serial number: unique number linked to the certificate
  - Algorithm: hash function and public-key encryption algorithm
  - Issuer: the entity that issued the certificate
  - Validity period
  - Subject: owner of the certificate
  - Public key
  - Properties: Encrypted hash value of the certificate with the owner's private key
  - Extension: additional information (used from version 3 and onward)
- Comodo, Sectigo, Symantec, Geotrust, RapidSSL, ...

What's Adobe Sign ?

Adobe Sign is the Adobe's proprietary solution for signing documents.
Adobe Sign is compatible with

- Adobe PDF
- Microsoft Word, Excel, PowerPoint
- Text, Rich Text
- Images
- Web

Adobe Sign supports different types of authentication: Adobe Sign ID and Adobe ID, Google ID, Single Sign-On (SSO).
The prices are:

- 145,05€/year for a person
- minimum 508,05€/year for a small company

Security of Adobe Sign

1. Adobe is certified ISO 27001, SOC2, PCI DSS.
2. Adobe Sign uses a PKI to certify documents with numeric signature before distributing them to participants.
3. Adobe use private virtual cloud of Amazon Web Services and Microsoft Azure.
4. Adobe use encryption algorithms that comply with the PCI DSS standard. They encrypt the datas with AES 256 bits.
5. They use HTTPS TLS V1.2 to transmit datas.

GnuPG

features:

- Hybrid ciphers
- encryption
- signature
- key management

GnuPG encryption

Key generation an listing

```
gpg --gen-key
gpg --list-keys
```

GnuPG encryption

Key generation an listing

```
gpg --gen-key
gpg --list-keys
```

Export /Import public key

```
gpg --export --armor Bob >publickey.asc
gpg --import publickey.asc
```

GnuPG encryption

Key generation an listing

```
gpg --gen-key
gpg --list-keys
```

Export /Import public key

```
gpg --export --armor Bob >publickey.asc
gpg --import publickey.asc
```

Encrypt using public key

```
gpg --encrypt --recipient receiversname filename.txt
gpg --encrypt -r raman -r steve -r gopi a.txt
```

GnuPG encryption

Key generation an listing

```
gpg --gen-key
gpg --list-keys
```

Export /Import public key

```
gpg --export --armor Bob >publickey.asc
gpg --import publickey.asc
```

Encrypt using public key

```
gpg --encrypt --recipient receiversname filename.txt
gpg --encrypt -r raman -r steve -r gopi a.txt
```

Decrypt using private key

```
gpg filename.txt.gpg
```

GnuPG signature

Sign and verify signature

```
gpg --sign file.txt
gpg --verify file.txt.gpg
```

GnuPG signature

Sign and verify signature

```
gpg --sign file.txt
gpg --verify file.txt.gpg
```

Extract document

```
gpg --output doc.txt --decrypt file.txt.gpg
```

GnuPG signature

Sign and verify signature

```
gpg --sign file.txt
gpg --verify file.txt.gpg
```

Extract document

```
gpg --output doc.txt --decrypt file.txt.gpg
```

Clear sign

```
gpg --output file.sig --clearsign file.txt
```

GnuPG signature

Sign and verify signature

```
gpg --sign file.txt
gpg --verify file.txt.gpg
```

Extract document

```
gpg --output doc.txt --decrypt file.txt.gpg
```

Clear sign

```
gpg --output file.sig --clearsign file.txt
```

Sign and encrypt

```
gpg --sign --encrypt –recipient recipient file.txt
```

Microsoft Office: Word

- Provides trustworthy digital signature of documents, using digital certificates from entrusted CAs, such as GlobalSign or IdenTrust
- Easy to setup in the document (Insert → Signature Line)
- Requires to pay for certificates (10-30 euros per month - several hundreds of euros per year !!)
- Practical example: `https://piv.idmanagement.gov/userguides/signworddoc/`
- GlobalSign yearly subscriptions: `https://www.globalsign.com/en/microsoft-office-document-signing/`
- Mostly oriented for companies who can afford the fees of the digital certificates

Online solutions

There are many free or paid online solutions for signing documents. They are easy to use but we have no information on the security of these signatures. We can see an example ▸ here

LaTex

The package *eforms* provide commands to add signature into a pdf. We can use the command

sigField[1]{*2*}{*3*}{*4*} to add a signature.
Parameter Description:

1. optional, used to enter any modification of appearance/actions
2. the title of the signature field
3. the width of the bounding rectangle
4. the height of the bounding rectangle

The package uses Acrobat to sign the document.

Conclusion

- Adobe Sign
  - reliable, safe and easy to use ☺
  - expensive (not available in the free version) ☹
- GnuPg
  - free, secure and available for Windows Linux and MacOS ☺
  - can become complex ☹
- Microsoft Word
  - simple to set up and reliable ☺
  - certificates are expensive ☹
- Online solutions
  - free and easy to use ☺
  - not secure ☹
- LaTeX
  - free and relatively easy to set for the signer ☺
  - not as secure as the previous ones ☹

Thank you for your attention !