Digital Signature

Pr. Noureddine Chikouche

University of M'sila

https://sites.google.com/view/chikouchenoureddine

Course outline

- Definition
- Principles
- RSA signature
- DSA Signature

Is it possible for entity A to forge a given document and claim that it sent from entity B?

Definition

- There digital signature is a security mechanism that allows a message or document to be encrypted using the sender's (or author's) private key.
- Electronic signatures such as handwritten signature used to prove the identity of the signatory (of the issuer) and the integrity of the document.
- The electronic signature ensures integrity, authenticity and *non-repudiation of sender*.

Principles

Applications of digital signatures:

- Sign and verify the different formats of document: Word, Excel and PDF.
- Conduct secure online transactions.
- Identify participants in an online transaction.
- Verify digital certificates (e.g., X.509)

Principles

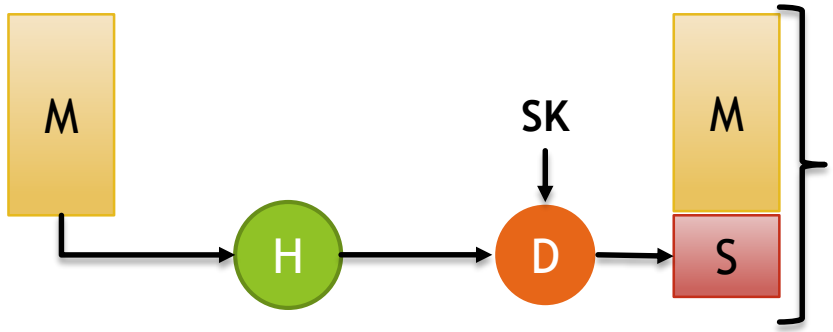
How does a digital signature work?

- To produce a signature, we use the hash functions and the public key encryption.
- A digital signature is produced by a digital signature generation algorithm.
- When the recipient receives the message and signature; it verifies the signature using an algorithm of digital signature verification.

Principle

Digital signature generation:

1) The sender calculates the hash of the message to be signed.
2) The fingerprint is encrypted with the issuer's private key using the decryption function. It's the digital signature.
3) The sender sends the message and the signature.

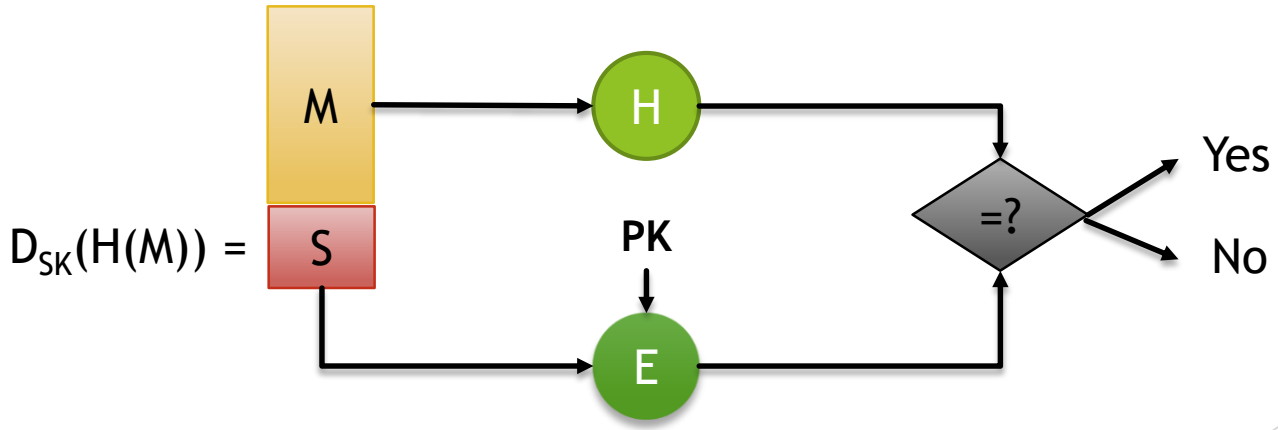


M:message **H:**hash function **S:**signature **D:**decryption algorithm

Principles

Digital signature verification:

1) The recipient recovers the digest of the signed message based on the signature. For this, it uses the encryption function with the issuer's public key.
2) He calculates the digest of the received message.



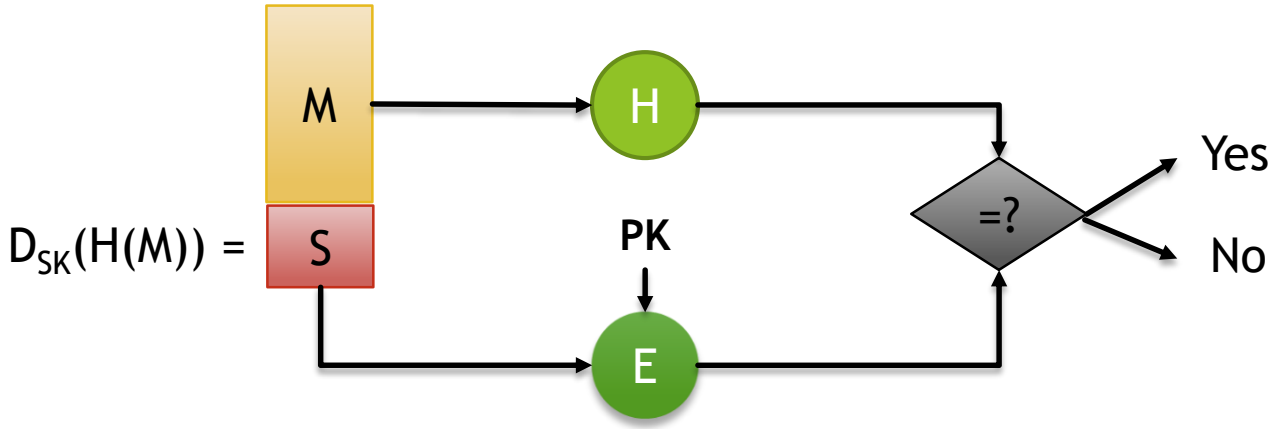
M:message **H:**hash function **S:**signature **E:**encryption algorithm

Principles

Digital signature verification (continued):

3) It compares two digests:

- if the two are equal, then the signature is authentic,
- Otherwise, either the message has been altered, or it was not written by the legitimate issuer.



M:message **H:**hash function **S:**signature **E:**encryption algorithm

Principles

Important:

- To improve the performance (save time and minimize communication costs) of digital signature generation and verification algorithms, the hash of a small message is signed instead of a large message (e.g., 10 MB).
- For example, if the SHA-256 function is used, the hash size to be signed is always 256 bits.

Principles

Signature properties:

- **The signature is tamper-proof:** only the signer knows the private signing key.
- **The signature is authentic:** the recipient is certain that the sender is the only one who can sign the message with their private key.
- **Public verification:** the recipient can verify the signature without needing the signer's assistance.
- **The signature is not reusable:** the signature belongs to a single document.

Signature Algorithms

- *RSA*
- *DSA (Digital Signature Algorithm)*
- *El Gamal*
- *Rabin*
- *ECDSA (Elliptic Curve Digital Signature Algorithm)*
- *NTRUSign*
- …

RSA signature

Key generation: (Issuer)

- Choose two prime numbers p and q of the same size.
- Calculate N =pq and $\Phi(N) = (p - 1)(q - 1)$.
- Choose e, $1 \leq e \leq \Phi(N)$.
- Calculate d, $1 \leq d \leq \Phi(N)$, $ed \equiv 1 \pmod{\Phi(N)}$.
- The private key: d used for generating signatures
- The public key: (n, e) used for verificationsignature.

RSA signature

Signature generation: (Issuer)

- The message m to be signed
- Calculate m' = h(m) where h is a hash function.
- Calculate $s = m'^{d} \bmod n$ and s is the signature of m.
- The issuer sends (m, s).

Signature verification: (Recipient)

- Obtain the issuer's public key (n, e),
- Calculate $m' = s^{e} \bmod n$,
- Calculate h(m),
- If h(m) = m' then the signature is authentic.

Signature RSA: Example

Key generation: (Issuer)

- Let p = 11 and q = 5
- n= 55 ,$\varphi(n)$=44
- Let e = 3, we find d =27
- The public key is 3 and the private key is (55,27).

Signature generation: (Issuer)

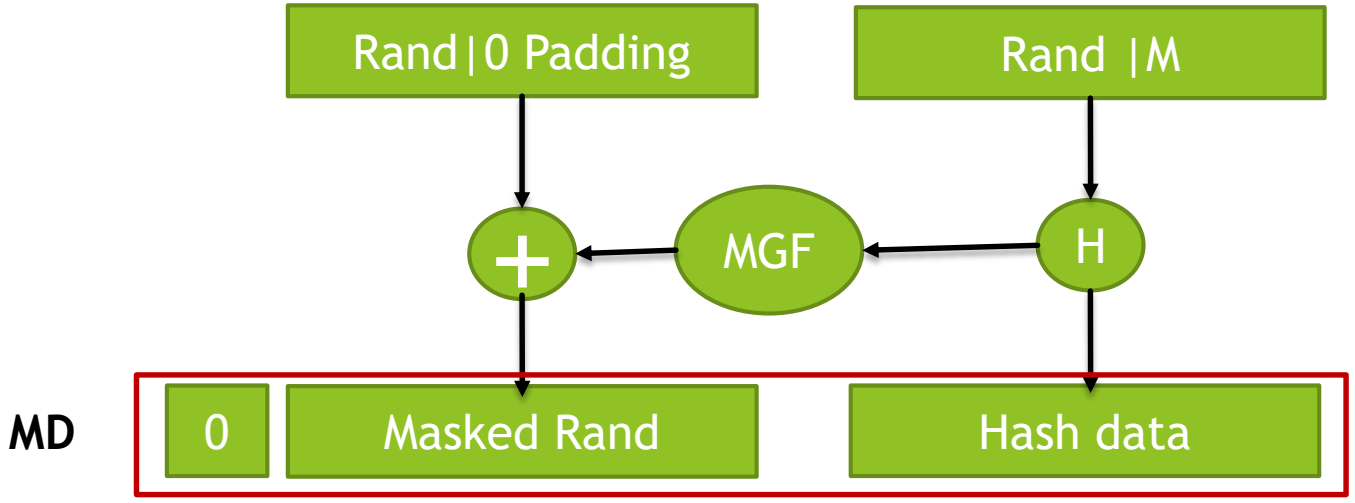
- Let m = 987, we assume m' = h(987) = 25.
- s = 11^{27} mod55 = 20.
- The sender sends (987, 20)

Signature verification: (Recipient)

- It calculates h(987) = 25.
- m'= 20^3 mod 55 = 25
- Therefore, h(m) = m', the signature is authentic.

Signature RSA PSS

- With the RSA signature, if we sign a message multiple times, we will always find the same signature.
- To avoid this problem, we use the signature RSA PSS (*Probabilistic Signature Scheme*) who used random numbers for randomize the signature.



$$s = (MD)^d \bmod n$$

MGF: mask generation function
Original Ref.PSS algorithm [BellareandRogaway, 1996]

Signature DSA

- *Digital Signature Algorithm*, known by the acronym DSA.
- It was proposed by NIST in 1991.
- It has become a standard FIPS 186, called *Digital Signature Standard* (DSS).
- It has some similarities with the El Gamal signature.
- The security of DSA relies on the difficulty of the problem of *discrete logarithm*.

Signature DSA

Key generation: (Issuer)

- Choose a prime number p of Lec512 $\leq L \leq$ 1024, and L is divisible by 64,
- Choose a prime number q of160bits, in such a way that $p-1=qz$, with z an integer,
- Choose h, with $1<h<p-1$ so that $g=h^z \bmod p>1$
- Randomly generate a x, with $0 <x<q$
- Calculate $y=g^x \bmod p$
- The public key is (p,q,g,y)
- The private key is x.

SignatureDSA

Signature generation: (Issuer)

- Choose a random number $k, 1<k<q$
- Calculate $s_1 = (g^k \bmod p) \bmod q$
- Calculate $s_2 = (H(m)+s_1 * x)k^{-1} \bmod q$,
- the signature is (s_1, s_2).
- The transmitter sends (m, s_1, s_2).

Signature verification: (Recipient)

- Reject the signature if $s_1<p$ or $s_2<q$ is not verified.
- Calculate $u_1 = H(m)\ s_2^{-1} \bmod q$
- Calculate $u_2 = s_1 s_2^{-1} \bmod q$
- Calculate $v = g^{u1} y^{u2} \bmod p$
- if v = s_1 , the signature is authentic.

SignatureDSA: Example

Key generation: (Issuer)

- We take q=101, p = 78q + 1 = 7879, z= 78 and h = 4
- $g=h^z \bmod p = 4^{78} \bmod 7879 = 105$
- $Let\ x = 62,\ y=g^x \bmod p = 105^{62} \bmod 7879 = 6907$
- The public key (7879, 101, 105, 6907)
- The private key: 62.

Generationof the signature: (Issuer)

- Let m=999, we assume h(m)=12.
- $Let\ k=17 \rightarrow k^{-1}=17^{-1} \bmod 101 = 6$
- $s_1=(g^k \bmod p) \bmod q = (105^{17} \bmod 7879) \bmod 101 = 84$
- $s_2= (H(m)+s1*x)k^{-1} \bmod q= (12+84*62)6 \bmod 101 = 10$
- So, the signature is (84.10).
- The sender sends (999, 84, 10)

SignatureDSA: Example

Signature verification: (Recipient)

- s_1=84<7879 and s_2=10<101
- $s_2^{-1}=(s_2)^{-1} \pmod q = 10^{-1} \bmod 101 = 91$
- $u_1=H(m)^* (s_2)^{-1} \pmod q = 12*91 \bmod 101 = 82$
- $u_2 = s_1^* (s_2)^{-1} \pmod q = 84*91 \bmod 101 = 69$
- $v = [g^{u1}*y^{u2} \bmod p] \bmod q$

  $v = (105^{82}*6907^{69} \bmod 7879) \bmod 101$

  v= 84
- So : $v=s_1$

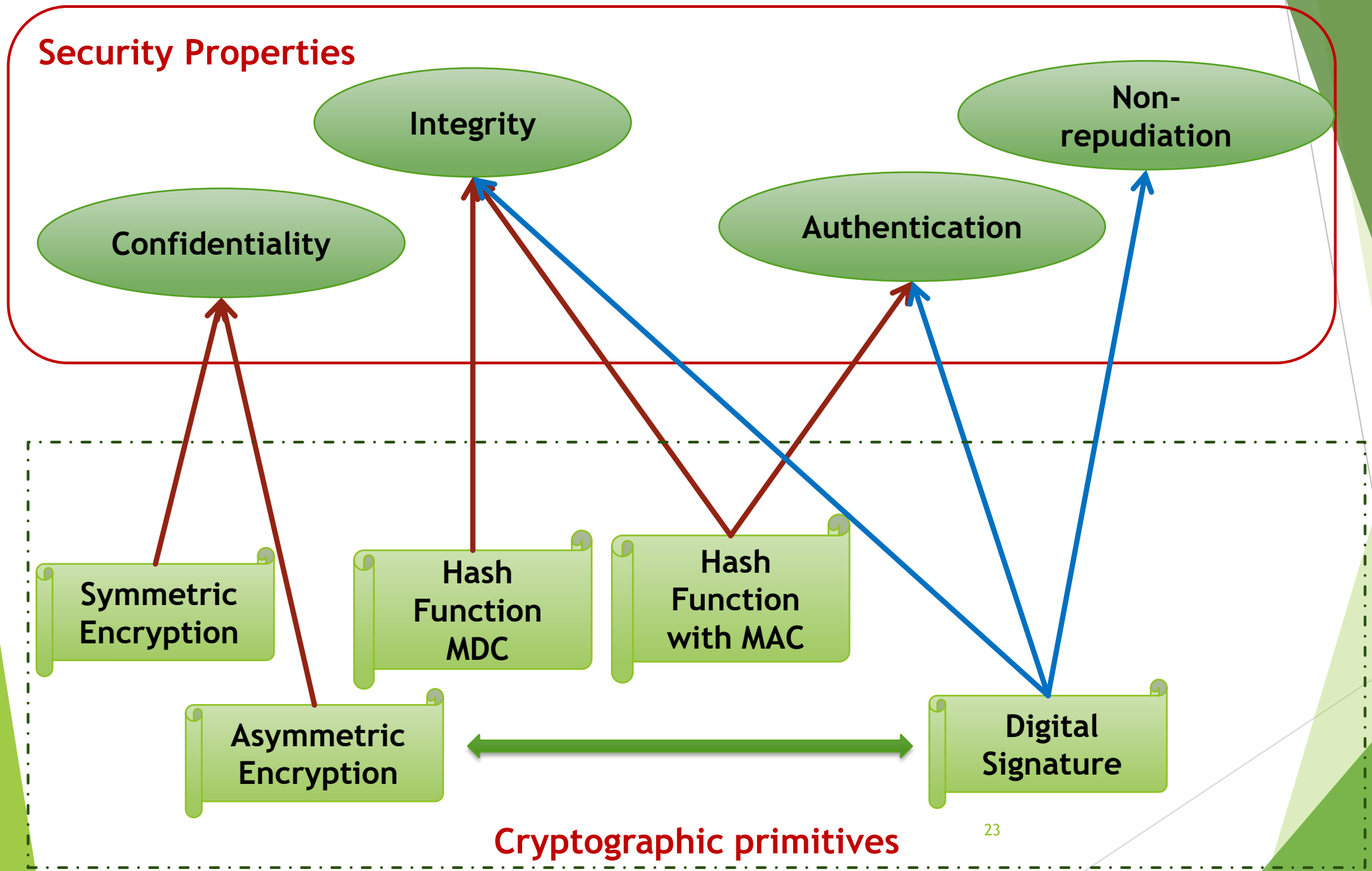
SignatureDSA

Advantage:

- The signature is randomized:

Limitations:

- *Calculation time*: in the verification algorithm, three exponentiation operations are calculated. It is expensive.
- *Signature size*: the signature is a couple of numbers of size *p*.


Security Properties
Integrity
Non-repudiation
Confidentiality
Authentication
Symmetric Encryption
Hash Function MDC
Hash Function with MAC
Asymmetric Encryption
Digital Signature
Cryptographic primitives