La signature éléctronique

Pr. Noureddine Chikouche

Université de M’sila

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

Plan du cours

- Définition
- Principes
- Signature RSA
- Signature DSA

Est-ce que possible une **entité *A*** forge un document donné et dire qu'il est venu de **l'entité *B***?

Définition

- La **signature électronique** (parfois appelée digitale/numérique) est un mécanisme de sécurité permettant de chiffrer un message ou un document en utilisant la clé privée de l'émetteur (ou l'auteur).
- La signature électronique comme signature manuscrite utilisée pour prouver l'identité du signataire (de l'émetteur) et l'intégrité du document.
- La signature électronique assure l'intégrité, l'authenticité et la *non-répudiation de l'origine*.

Principes

Applications des signatures numériques:

- Signer et vérifier les différents formats de document: Word, Excel et PDF.
- Effectuer des transactions en ligne sécurisées.
- Identifier les participants d'une transaction en ligne.
- Vérifier les certificats numériques (ex. X509)

Principes

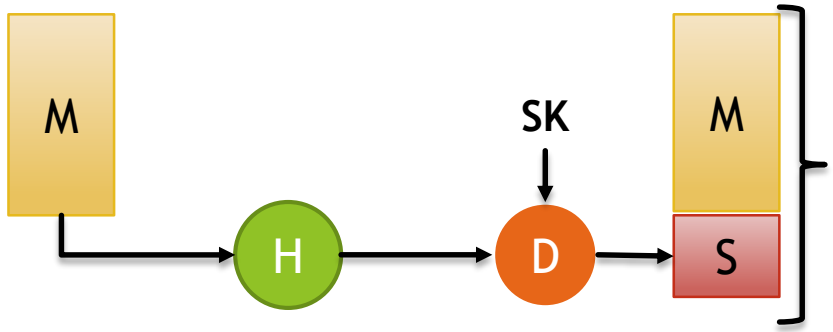
Comment fonctionne la signature numérique ?

- Pour produire une signature, on utilise les fonctions de hachage et le chiffrement à clé publique.
- Une signature numérique est produite par un algorithme de génération de signature numérique.
- Lorsque le destinataire reçoit le message et la signature, il vérifie la signature par un algorithme de vérification de signature numérique.

Principe

Algorithme de génération de signature numérique:

1) L'émetteur calcule l'empreinte du message à signer.
2) L'empreinte est chiffré avec la clé privée de l'émetteur en utilisant la fonction de déchiffrement. C'est la signature numérique.
3) L'émetteur envoie le message et la signature.



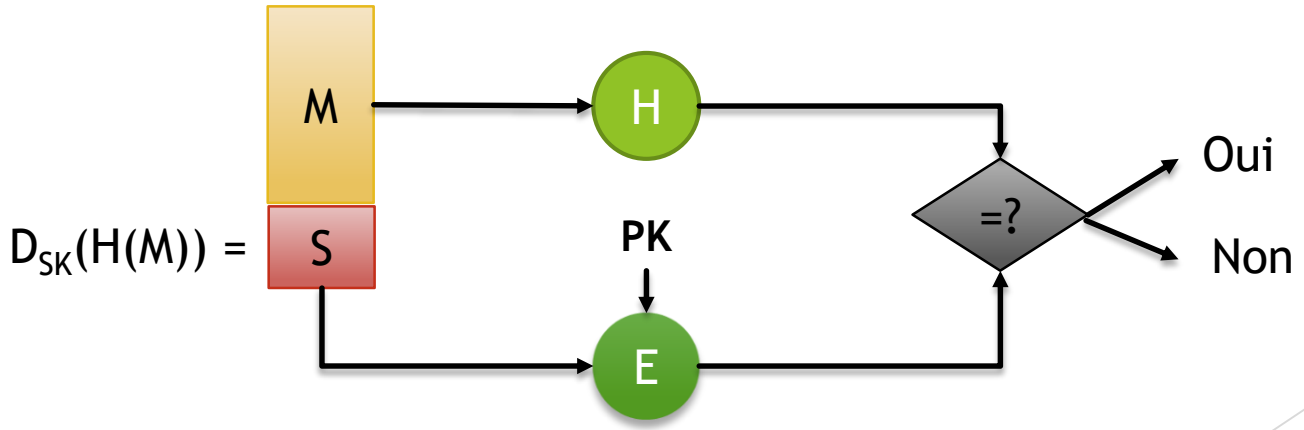
M: message **H:** fonction de hachage **S:** signature **D:** algorithme de déchiffrement

Principes

Algorithme de vérification de signature numérique:

1) Le destinataire récupère l'empreinte du message signé à partir de la signature. Pour cela, il utilise la fonction de chiffrement avec la clé publique de l'émetteur.

2) Il calcule l'empreinte du message reçu.



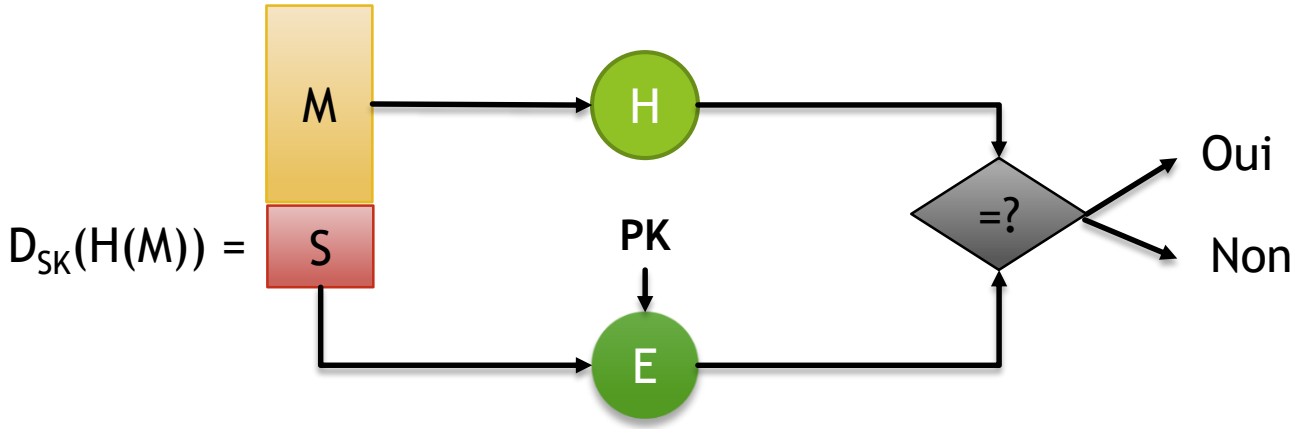
M: message **H:** fonction de hachage **S:** signature **E:** algorithme de chiffrement

Principes

Algorithme de vérification de signature numérique (suit):

3) Il compare les deux empreintes:

- si les deux sont égaux, alors la signature est authentique,
- Sinon, soit le message a été altéré, soit il n'a pas été rédigé par l'émetteur légitime.



M: message **H:** fonction de hachage **S:** signature **E:** algorithme de chiffrement

Principes

Important:

- Pour améliorer la performance (*gagner le temps* et *minimiser le coût de communication*) des algorithmes de génération et de vérification de signature numérique, on signe l'empreinte du message de petite taille au lieu de signer le message de grande taille (ex. 10MO).
- Par exemple, si on utilise la fonction SHA-256, alors la taille de l'empreinte à signer est toujours égal 256 bits.

Principes

Propriétés de signature:

- La signature est infalsifiable: le seul qui connait la clé privée de signature est le signataire.
- La signature est authentique: le destinataire est sûr que l'émetteur est le seul qui peut signer le message avec sa clé privée.
- Vérification publique: le destinataire peut vérifier la signature sans aucun besoin de l'aide de signataire.
- La signature n'est pas réutilisable: La signature appartient à un seul document.

Algorithmes de Signature

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

Signature RSA

Génération des clés: (Emetteur)

- Choisir deux nombres premiers p et q de même taille.
- Calculer N = p.q et $\Phi(N) = (p - 1)(q - 1)$.
- Choisir e N,1≤ e ≤ $\Phi(N)$.
- Calculer d N, 1 ≤ d ≤ $\Phi(N)$, e.d ≡1 (mod $\Phi(N)$).
- La clé privée: d utilisée pour la génération de signatures
- la clé publique: (n, e) utilisée pour la vérification de signature.

Signature RSA

Génération de la signature: (Emetteur)

- Soit le message m à signer,
- Calculer m' = h(m) où h est une fonction de hachage.
- Calculer s = m'^{d} mod n et s est la signature de m.
- L'émetteur envoie (m, s).

Vérification de la signature: (Destinataire)

- Obtenir la clé publique de l' émetteur (n, e),
- Calculer m' = s^{e} mod n,
- Calculer h(m),
- Si h(m) = m' alors la signature est authentique.

Signature RSA: Exemple

Génération des clés: (Emetteur)

- Soit p= 11 et q=5
- n= 55 , $\varphi(n)$=44
- Soit e= 3, on trouve d= 27
- La clé publique est 3 et la clé privée est (55, 27).

Génération de la signature: (Emetteur)

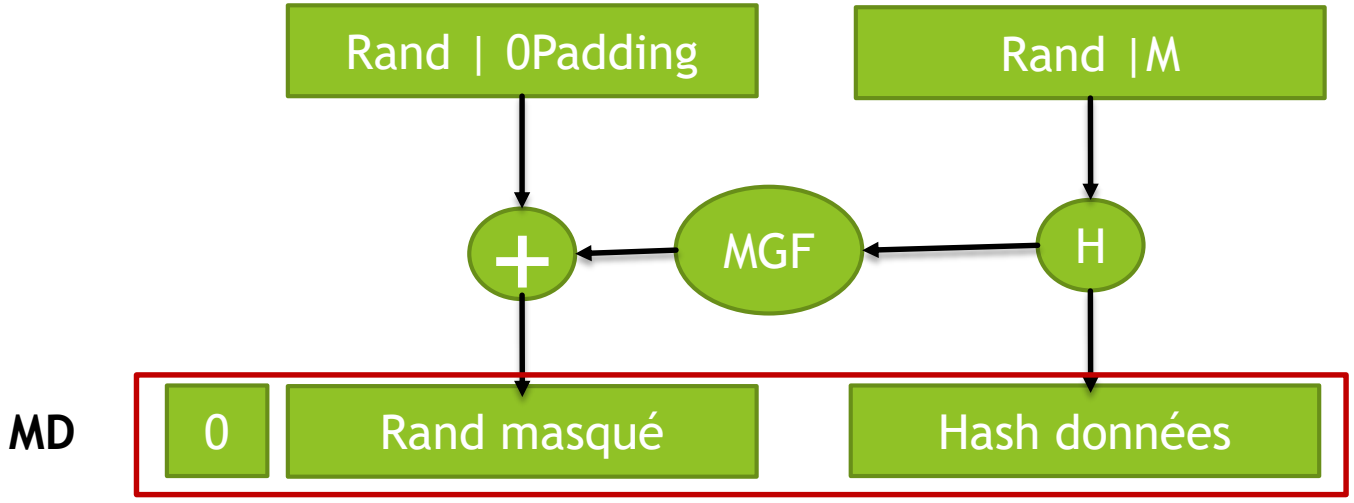
- Soit m= 987, on suppose m'=h(987)=25.
- s= 11^{27} mod 55 = 20.
- L'émetteur envoie (987, 20)

Vérification de la signature: (Destinataire)

- On calcule h(987)= 25.
- m'=20^3 mod 55 = 25
- Alors h(m)=m', la signature est authentique

Signature RSA PSS

- Avec la signature RSA, si on signe un message plusieurs fois, on va trouver toujours la même signature.
- Pour éviter ce problème, on utilise la signature RSA PSS (*Probabilistic Signature Scheme*) qui utilise des nombres aléatoires pour randomiser la signature.



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

MGF: mask generation function
Ref. Original PSS algorithm [Bellare and Rogaway, 1996]

Signature DSA

- *Digital Signature Algorithm*, plus connu sous l' acronyme DSA.
- Il est proposé par NIST en 1991.
- Il est devenu une norme FIPS 186, appelée *Digital Signature Standard* (DSS).
- Il a des points communs avec la signature El Gamal.
- La sécurité de DSA repose sur la difficulté du problème de *logarithme discret*.

Signature DSA

Génération des clés: (Emetteur)

- Choisir un nombre premier p de Lec $512 \leq L \leq 1024$, et L est divisible par 64,
- Choisir un nombre premier q de 160 bits, de telle façon que $p - 1 = qz$, avec z un entier,
- Choisir h, avec $1 < h < p - 1$ de manière à ce que $g = h^z \bmod p > 1$
- Générer aléatoirement un x, avec $0 < x < q$
- Calculer $y = g^x \bmod p$
- La clé publique est (p, q, g, y)
- La clé privée est x.

Signature DSA

Génération de la signature: (Emetteur)

- Choisir un nombre aléatoire k, $1 < k < q$
- Calculer $s_1 = (g^k \bmod p) \bmod q$
- Calculer $s_2 = (H(m) + s_1 * x) k^{-1} \bmod q$,
- la signature est (s_1, s_2).
- L'émetteur envoie (m, s_1, s_2).

Vérification de la signature: (Destinataire)

- Rejeter la signature si $s_1 < p$ ou $s_2 < q$ n'est pas vérifié.
- Calculer $u_1 = H(m)\, s_2^{-1} \bmod q$
- Calculer $u_2 = s_1 s_2^{-1} \bmod q$
- Calculer $v = g^{u1} y^{u2} \bmod p$
- si v = s_1, la signature est authentique.

Signature DSA: Exemple

Génération des clés: (Emetteur)

- On prend q=101, p = 78q + 1 = 7879, z= 78 et h = 4
- $g = h^z \bmod p = 4^{78} \bmod 7879 = 105$
- *Soit* $x = 62$, $y = g^x \bmod p = 105^{62} \bmod 7879 = 6907$
- La clé publique (7879, 101, 105, 6907)
- La clé privée: 62.

Génération de la signature: (Emetteur)

- Soit m=999, on suppose h(m)=12.
- *Soit* $k = 17 \rightarrow k^{-1} = 17^{-1} \bmod 101 = 6$
- $s_1 = (g^k \bmod p) \bmod q = (105^{17} \bmod 7879) \bmod 101 = 84$
- $s_2 = (H(m)+s1*x)k^{-1} \bmod q = (12+84*62)6 \bmod 101 = 10$
- Donc, la signature est (84,10).
- L'émetteur envoie (999, 84, 10)

Signature DSA: Exemple

Vérification de la signature: (Destinataire)

- s_1=84 < 7879 et 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
- Alors : $v = s_1$

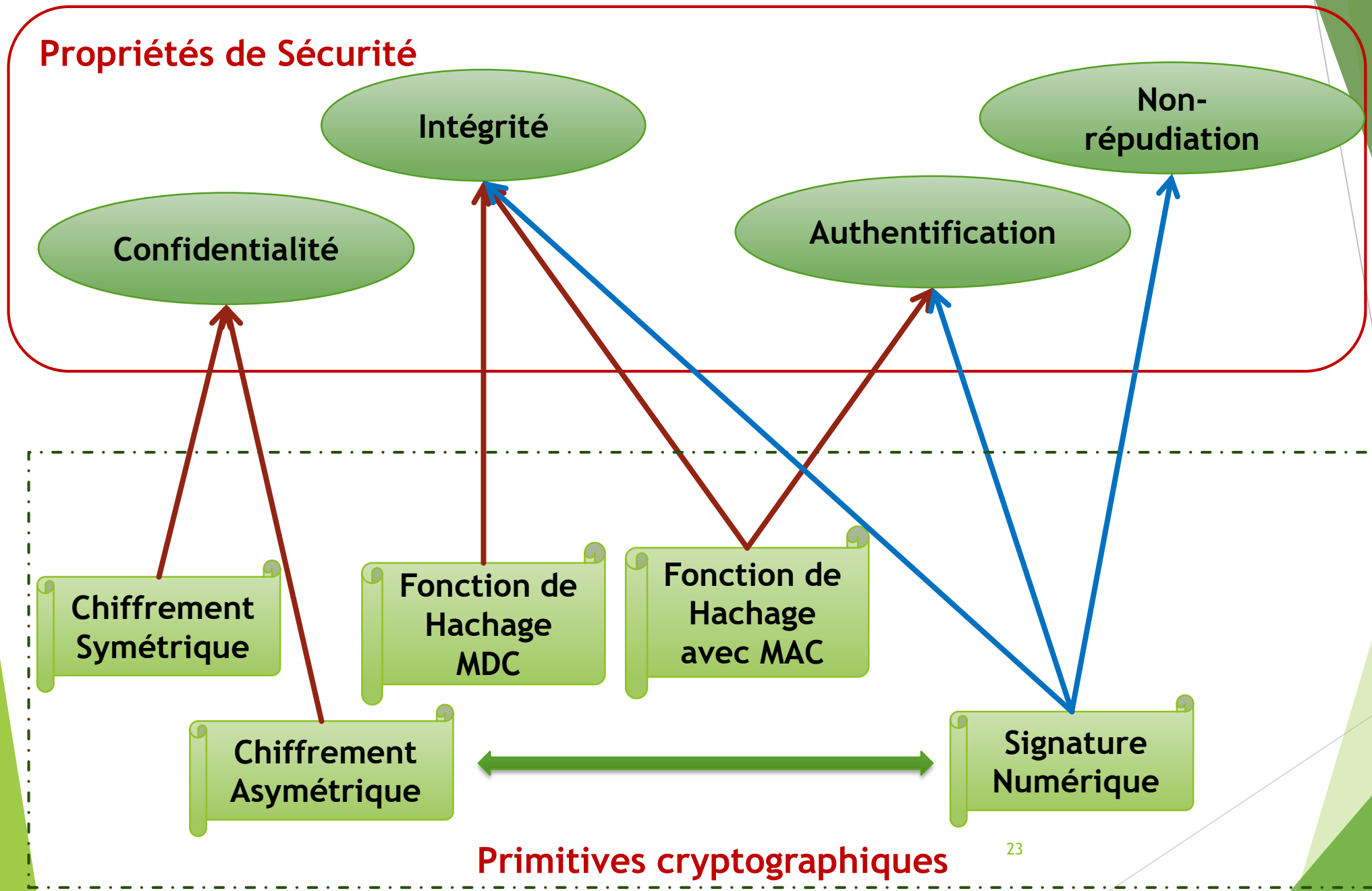
Signature DSA

Avantage:

- La signature est randomisée:

Limitations:

- *Temps de calcul*: dans l'algorithme de vérification, on calcule trois opérations d'exponentiation. Il est coûteux.
- *Taille de signature*: la signature est une couple des nombres de taille *p*.


Propriétés de Sécurité
Intégrité
Non-répudiation
Confidentialité
Authentification
Chiffrement Symétrique
Fonction de Hachage MDC
Fonction de Hachage avec MAC
Chiffrement Asymétrique
Signature Numérique
Primitives cryptographiques