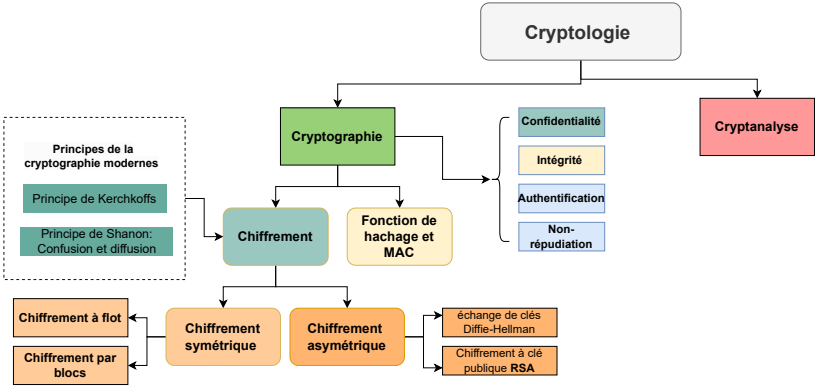
Initiation à la Cryptographie

Signature numérique et authentification

June 23, 2025

Awaleh HOUSSEIN

Recap de posts précédents [1]



Jusqu'ici nous avons vu la confidentialité (chiffrement) et l'intégrité (hachage). Aujourd'hui, découvrons un troisième pilier : comment prouver l'**authenticité** et empêcher le déni d'envoi ?

[1] Cinquième post: https://www.linkedin.com/feed/update/urn:li:activity:7330684313585278978/

Rappel et besoin de la signature numérique

Problématique

Comment prouver qu'un message vient bien de l'expéditeur prétendu, et l'empêcher de nier l'avoir envoyé ?

Limites des solutions précédentes

- Hachage : Garantit l'intégrité mais pas l'origine
- HMAC : Vérifie l'origine mais nécessite clé *partagée*
- Besoin : Mécanisme asymétrique avec preuve vérifiable par tous

Solution

La **signature numérique** : l'équivalent électronique d'une signature manuscrite, mais plus fiable !

Signature numérique : Définition et propriétés

Signature numérique — Définition et garanties

Mécanisme cryptographique basé sur le chiffrement à clé publique, permettant d'attacher à un document une preuve d'identité, tout en assurant :

- **Authenticité** : le message vient bien de l'expéditeur attendu;
- **Intégrité** : il n'a pas été modifié en route;
- **Non-répudiation** : l'expéditeur ne peut nier l'avoir envoyé [a].

[a] Définition : https://vitrinelinguistique.oqlf.gouv.qc.ca/fiche-gdt/fiche/8384641/signature-numerique

Fonctionnement : Comment ça marche ?

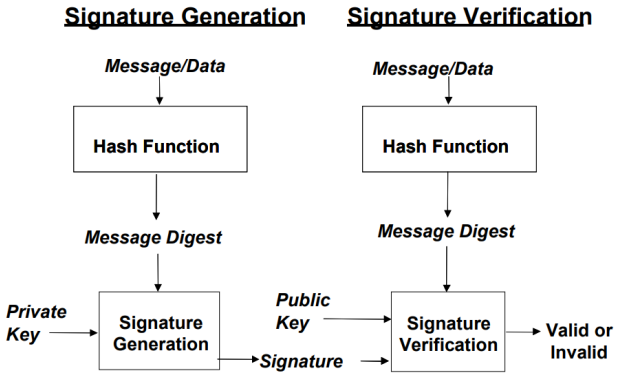


Figure: Processus de signature et vérification [2]

[2] Image source: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

Exemple concret : Signature RSA étape par étape [3]

Signature

1. Message M = "Paiement 100€"
2. Haché H: SHA-256$(M) = 437a5410d6 \cdots 4bbd$
3. Signature $S = H^d \mod n$ (avec clé privée (d)

Vérification

1. Bob reçoit (M, S)
2. Calcule $H' =$ SHA-256(M)
3. Déchiffre $H'' = S^e \mod n$ (clé publique (e, n))
4. Compare H' et H''
5. Si $H' == H''$ la signature est valide, sinon elle n'est pas valide.

Sécurité

Basée sur la difficulté de factoriser $n = p \times q$ (problème RSA).

[3] Problème de factorisation détaillé sur le post https://www.linkedin.com/feed/update/urn:li:activity:7319752681365950465/

Algorithmes courants : Comparatif [4]

Algorithme	Problème mathématique	Atouts	Limites
RSA	Factorisation d'entiers	Vérification rapide, chiffrement + signature, largement adopté	Clés volumineuses (3072 bits), génération lente, vulnérabilité quantique future
DSA	Logarithme discret	Standard NIST, génération rapide	Vérification lente, vulnérable à l'aléa faible.
ECDSA	Logarithme discret sur courbes elliptiques	Clés compactes, haute performance, idéal pour IoT/mobile	Sensible à l'aléa cryptographique
EdDSA	Logarithme discret sur courbes Edwards	Signatures déterministes, rapide, résistant aux attaques par canal auxiliaire	Support partiel sur anciens systèmes, adoption encore limitée.

[4] Pour plus de détails, consulez le document https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.186-4.pdf

Applications concrètes

Dans la vie quotidienne

- **Emails sécurisés** (PGP/GPG)
- **Contrats électroniques** (DocuSign)
- **Administration électronique** (FranceConnect)

Dans les systèmes critiques

- **Blockchain** : Signature des transactions (Bitcoin, Ethereum).
- **Certificats SSL/TLS** : Authentification des sites web.
- **Logiciels** : Vérification de l'intégrité des mise-à-jour.

Avantages des signatures numériques

Par rapport aux signatures manuscrites

- **Infalsifiable** : Impossible à reproduire sans clé privée
- **Vérification instantanée** : Automatisable
- **Lien indissociable** avec le document signé

Avantages pratiques

- Validité légale (eIDAS en Europe) [a]
- Gain de temps et réduction de papier
- Traçabilité des transactions

[a] Pour plus de détails sur le règlement eIDAS `https://cyber.gouv.fr/le-reglement-eidas-n9102014`

Limites et défis

Défis techniques

- **Gestion des clés** : Perte = impossibilité de signer
- **Performance** : Coût calcul pour les gros volumes
- **Menace quantique** : RSA/ECDSA vulnérables à long terme

Bonnes pratiques

- Utiliser des clés de 2048+ bits (RSA)
- Renouveler régulièrement les clés
- Stocker les clés privées sur HSM [a]

[a] Un HSM (Hardware Security Module) est un dispositif matériel sécurisé chargé de générer, stocker et protéger des clés cryptographiques, ainsi que d'exécuter des opérations cryptographiques sensibles.

À retenir

Essentiels

- La signature numérique = **triple garantie** (authenticité, intégrité, non-répudiation)
- Basée sur la cryptographie **asymétrique**
- Processus en deux temps : **signature** (clé privée) et **vérification** (clé publique)

En pratique:

- *Choisir l'algorithme adapté* : RSA pour généraliste, ECDSA/EdDSA pour performances[5]
- *Protéger sa clé privée* comme un bijou précieux
- *Vérifier les certificats* avant de signer

[5] Face au risque quantique, privilégier les signatures post-quantiques standardisées (NIST FIPS 204/205) : `https://csrc.nist.gov/pqc-standardization`