


Primitives Cryptographiques et Propriétés de Sécurité

Les fondamentaux de la cryptographie moderne, explorant les algorithmes symétriques et asymétriques

Les fonctions de hachage, et les applications pratiques via OpenSSL.

 par Stéphane LARCHER



Introduction et Rappel



La feuille de route

Algorithmes symétriques modernes et leurs modes de fonctionnement.

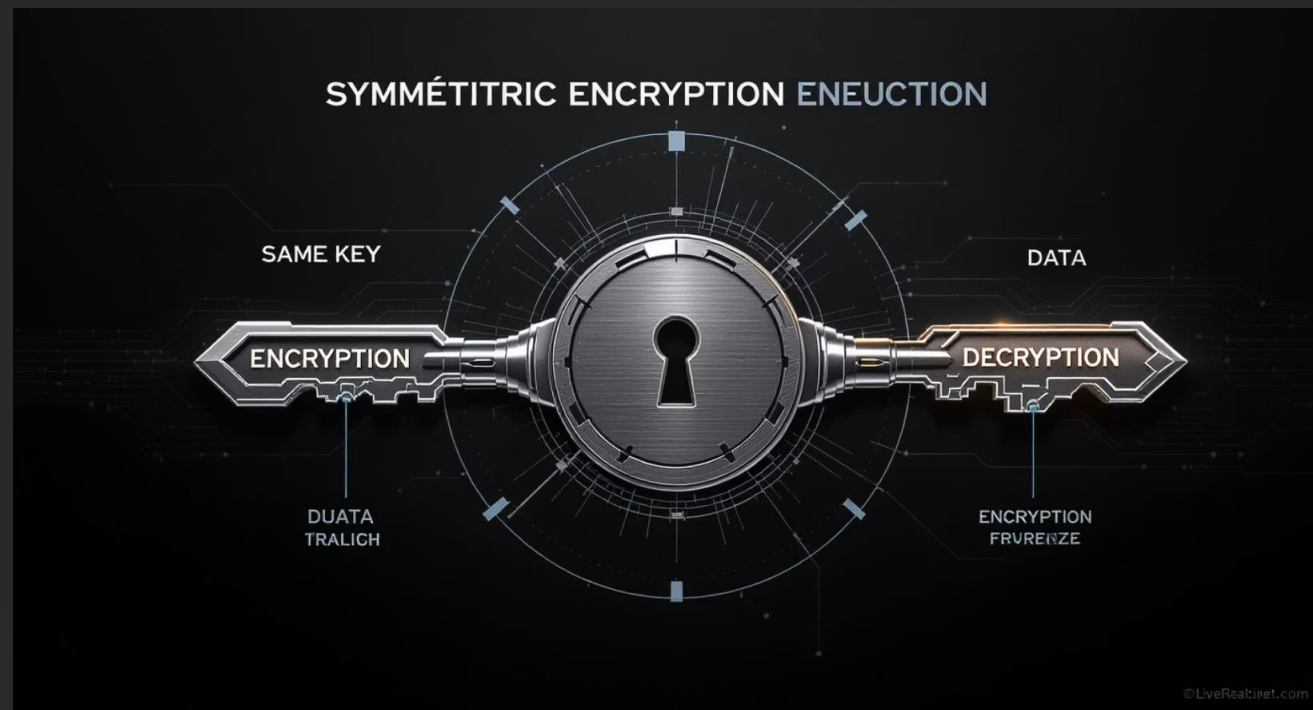
Algorithmes asymétriques et leurs applications (distribution de clés, signatures, certificats).

Fonctions de hachage et le MAC (Message Authentication Code).

Attaques et des faiblesses concrètes de ces primitives.



Algorithmes Symétriques – Principes et Modes



Définition : Un algorithme de chiffrement symétrique utilise la **même clé** pour chiffrer et déchiffrer.

Principes généraux

Exemples d'usages : Chiffrement de volumes (BitLocker, VeraCrypt), Wi-Fi (WPA2), TLS (phase de chiffrement des données après l'échange de clés).

Modes de chiffrement

- ECB (Electronic Codebook) - Chiffre indépendamment chaque bloc avec la même clé.
- CBC (Cipher Block Chaining) - Chaque bloc est combiné (XOR) avec le bloc chiffré précédent avant chiffrement.
- CFB (Cipher Feedback) - On chiffre le précédent bloc chiffré (ou l'IV) puis on XOR avec le clair pour obtenir le chiffré.
- OFB (Output Feedback) - Similaire à CFB, sauf qu'on réutilise la **sortie** de la fonction de chiffrement pour générer un keystream.
- CTR (Counter) - Convertit un bloc en mode flux en chiffrant un compteur qui s'incrémente à chaque bloc.

Cryptographic Keys



Algorithmes Symétriques – Implémentations Majeures

Structure de Feistel



Principe : le message est séparé en deux moitiés (Gauche, Droite).
À chaque tour, on applique une fonction F à l'une des moitiés, puis on la combine (XOR) avec l'autre moitié, avant de permuter.

Permet de construire des **chiffres à clé secrète** robustes à partir de fonctions F plus simples.

3DES (Triple DES)



Amélioration de DES en appliquant DES trois fois (chiffrer-déchiffrer-chiffrer).

Clé plus longue (jusqu'à 168 bits effectifs).

Plus sécurisé que DES mais moins performant que les algorithmes plus récents.



DES (Data Encryption Standard)



Historique : standardisé en 1977 par le NBS (ancêtre du NIST).

64 bits de bloc, clé de 56 bits (effectifs).

Faiblesses : Clé trop courte par rapport à la puissance de calcul moderne → attaques par force brute.

AES (Advanced Encryption Standard)

Successeur de DES, adopté en 2001.

Blocs de 128 bits, clés possibles de 128, 192 ou 256 bits.

Basé sur un réseau de substitution-permutation (SPN) plutôt qu'une structure Feistel pure.

Avantages : Rapide, sûr, standard mondial.

Algorithmes Asymétriques



Concepts et différences

Deux clés distinctes : **clé publique** (chiffrement, vérification de signature) et **clé privée** (déchiffrement, génération de signature).

Problématique résolue : Permet la distribution de clés sans canal sécurisé (contrairement au symétrique où la clé doit être partagée).



RSA

Basé sur la difficulté de factorisation de grands nombres entiers.

Génération de clés : choix de deux grands nombres premiers (p , q), calcul de $n = p \cdot q$, etc.

Chiffrement/déchiffrement : exponentiation mod n , rôles de e et d .

Faibles possibles : Clés trop courtes (< 2048 bits), mauvaise implémentation de padding (PKCS#1 v1.5).



Diffie-Hellman

Échange de clé : Permet à deux parties d'obtenir un secret partagé sans jamais le transmettre en clair.

Basé sur la difficulté du logarithme discret dans un groupe (ex. groupe multiplicatif modulo p).

Variante : DH sur courbes elliptiques (ECDH).



Elliptic Curves (ECC)

Utilise la difficulté du logarithme discret sur des **courbes elliptiques**.

Même sécurité avec des clés beaucoup plus courtes qu'avec RSA.

Très utilisé pour les environnements contraints (smartphones, IoT).



Signatures Électroniques et Certificats

1

Génération de clés

Nécessite un générateur de nombres aléatoires sûr (CSPRNG).

La qualité de l'aléa est cruciale pour la sécurité des clés générées.

2

Signatures électroniques

RSA, ECDSA (Elliptic Curve Digital Signature Algorithm).

Permettent l'authentification de l'expéditeur et la non-répudiation.

Utilisent la clé privée pour signer, la clé publique pour vérifier.

3

Certificats X.509

Format standard pour associer une clé publique à une identité.

Infrastructures de gestion de clés (PKI) : autorités de certification (CA), certificats racine, chaîne de confiance.

Utilisés dans HTTPS, S/MIME, et de nombreux protocoles sécurisés.

Fonctions de Hachage et MAC



Propriétés clés des fonctions de hachage

1. **Résistance aux collisions** : Il est difficile de trouver deux messages différents ayant le même haché.
2. **Résistance à la préimage** : Il est difficile de retrouver le message d'origine à partir du haché.
3. **Diffusion** : Une petite modification du message doit changer radicalement le haché.



Exemples de fonctions

MD5 (128 bits) : obsolète, collisions trouvées.

SHA-1 (160 bits) : vulnérable aux collisions (collision pratique démontrée en 2017).

SHA-2 (SHA-256, SHA-512) : encore considéré comme sûr aujourd'hui.

SHA-3 (Keccak) : sélectionné par le NIST en 2012, alternative à SHA-2.



MAC (Message Authentication Code)

Objectif : Assurer l'**authenticité** et l'**intégrité** d'un message en utilisant une clé secrète.

HMAC : MAC basé sur une fonction de hachage (ex. HMAC-SHA-256).

Très utilisé dans les protocoles de sécurité (ex. TLS, IPSec) et les API (authentification de requêtes).

Attaques et Faiblesses

Collisions MD5

Permet de fabriquer deux documents différents ayant la même empreinte.
Problème majeur pour la confiance dans la signature.

Force brute et implémentation

Attaque par force brute : possible si la clé est trop courte.
Failles d'implémentation : randomisation insuffisante, padding mal géré, etc.



Attaques par dictionnaire

Concernent le cassage de mots de passe stockés sous forme de hachés (faible entropie).

Contre-mesures : salage (salt), étirement de clé (PBKDF2, bcrypt, Argon2).

Side-channel attacks

Analyse de consommation d'énergie, fuites par canaux cachés.