

Introduction aux primitives cryptographiques et propriétés de sécurité

Délivrer la bonne donnée
à la bonne personne
au bon moment,
et le savoir

 par Stéphane LARCHER



Propriétés de sécurité

Confidentialité

Garantir que seules les entités autorisées ont accès à l'information.

Exemples : Chiffrement des données stockées sur un disque ou lors de la transmission sur un réseau (HTTPS).

Mécanismes : Chiffrement symétrique (AES), chiffrement asymétrique (RSA), etc.

Intégrité

Assurer que l'information n'a pas été altérée, volontairement ou par accident.

Exemples : Fonction de hachage (SHA-256), code d'authentification de message (MAC).

Mécanismes : Contrôle d'intégrité lors des téléchargements de fichiers, validation de l'intégrité dans les transactions bancaires.

Disponibilité

S'assurer que le service ou la ressource reste accessible et fonctionnel pour les utilisateurs légitimes.

Exemples : Stratégies de reprise après sinistre (Disaster Recovery), redondance de serveurs, protection contre les attaques par déni de service (DDoS).

Authentication et Non-répudiation

Authentication

Vérifier l'identité d'une entité (utilisateur, machine, service...).

Exemples : Mots de passe, certificats X.509, protocoles de défi-réponse (Challenge-Response).

Mécanismes : Authentication forte à deux facteurs (2FA), protocoles comme Kerberos.

Non-répudiation

Empêcher qu'une entité ayant effectué une action puisse nier l'avoir faite.

Exemples : Signature numérique (avec clés asymétriques), horodatage électronique.

Mécanismes : Acte d'engagement contractuel, journaux d'audit (logs).



Contrôle d'accès



Définition

Ensemble de méthodes et de mécanismes permettant de réguler l'accès à des ressources (fichiers, bases de données, services...).



DAC (Discretionary Access Control)

Chaque propriétaire de ressource définit qui peut y accéder.



MAC (Mandatory Access Control)

Politique de sécurité globale imposée par le système (ex. niveaux de classification).



RBAC (Role-Based Access Control)

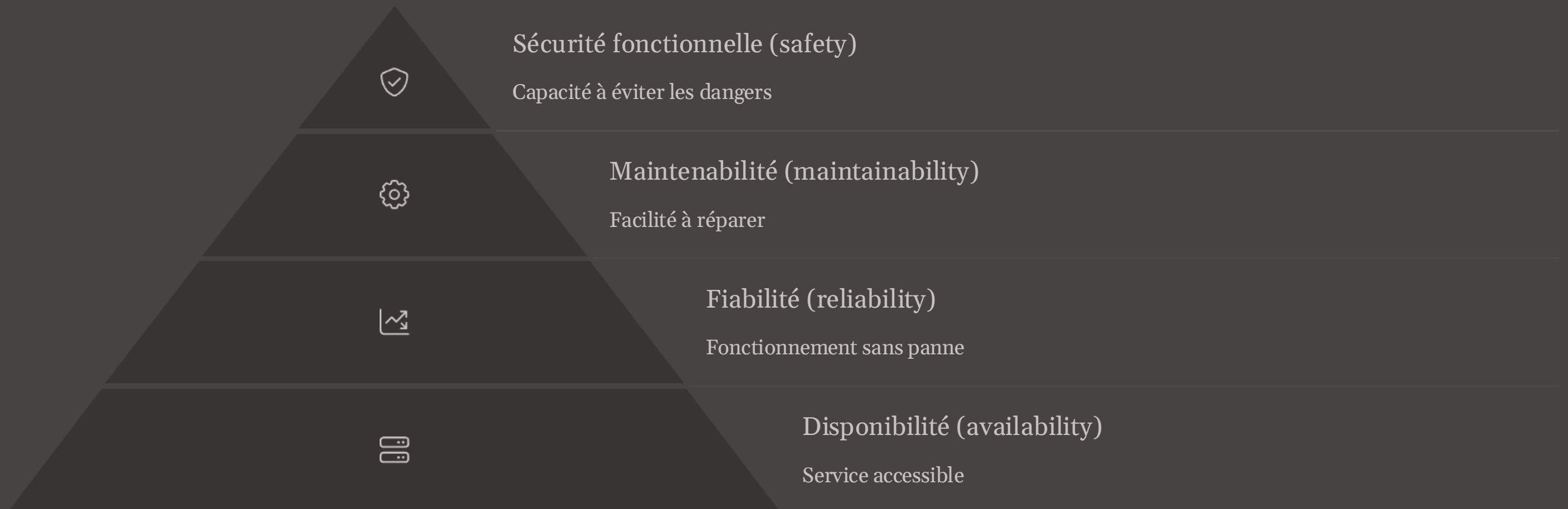
Accès basé sur des rôles, couramment utilisé en entreprise.

Exemples pratiques :

Gestion des droits UNIX (lecture/écriture/exécution).

ACL (Access Control Lists) dans les systèmes Windows.

Sûreté de fonctionnement (dependability)



Objectif : Assurer que le système poursuit son fonctionnement même en cas de pannes ou d'attaques.

Exemples :

Systèmes redondants : duplication de serveurs en cluster, RAID pour les disques durs.

Politiques de sauvegarde et de restauration (backup/restore).

Stéganographie



Définition

Art de cacher l'existence même d'un message dans un autre flux de données (image, audio, vidéo, texte).

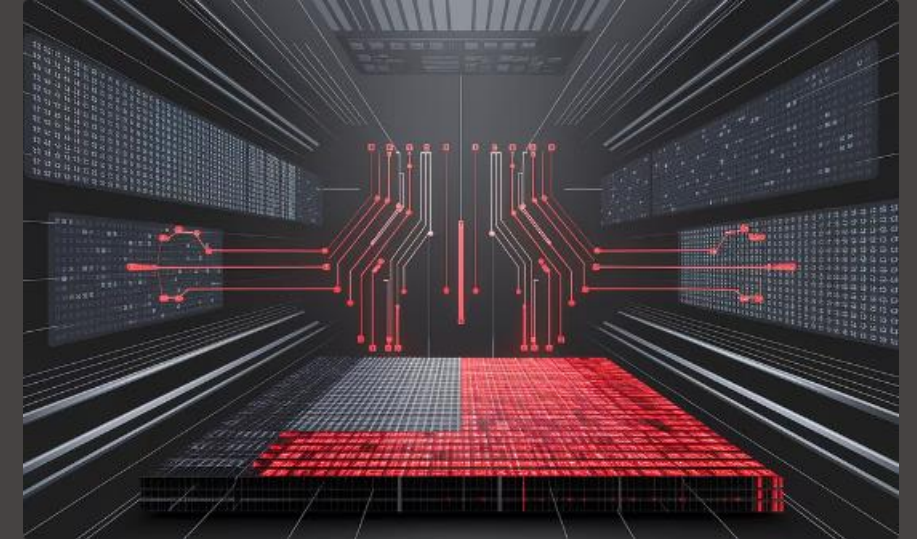
Objectif : La confidentialité passe ici par la discrétion de l'existence du message (contrairement au chiffrement, où le message est transformé mais clairement visible).



Exemples

LSB (Least Significant Bit) : insertion de bits dans les bits de poids faible d'une image numérique.

Insertion de données dans des paquets réseau « inoffensifs ».



Limites

La détection de stéganographie est possible via l'analyse statistique du support (image, audio...).

Capacité d'information souvent limitée, moindre robustesse si le fichier est recomprimé ou modifié.

Principe de Kerckhoffs

Formulation

Le principe de Kerckhoffs (du nom du cryptographe Auguste Kerckhoffs) énonce qu'un système cryptographique doit rester sûr même si tout y est connu, à l'exception de la clé secrète. Autrement dit :

« La sécurité ne doit pas reposer sur le secret de l'algorithme, mais uniquement sur le secret de la clé.
»

Conséquences et intérêts

Transparence de l'algorithme : Puisque la robustesse du système n'est pas censée dépendre de la discrétion de l'algorithme, celui-ci doit pouvoir être analysé, testé et audité par la communauté scientifique (ou les autorités compétentes).

Sécurité par la clé : L'idée est que le cryptanalyste (la personne qui tente de casser le chiffrement) doit être confronté à un problème « décourageant », en l'occurrence trouver une clé suffisamment complexe, plutôt que de deviner un fonctionnement opaque.

Standards ouverts : Aujourd'hui, la plupart des algorithmes utilisés (tels que AES, RSA, ECC, etc.) sont publiquement accessibles, et leur sécurité repose sur des principes mathématiques éprouvés, non sur le fait que personne ne connaît la méthode.

Avantages pratiques

Vérification par les pairs : Grâce à la publication des spécifications, l'algorithme subit un examen minutieux, ce qui permet de détecter des failles éventuelles et d'éprouver sa solidité.

Confiance accrue : Les utilisateurs du système peuvent avoir confiance dans l'algorithme, sachant qu'il a été scruté et validé par un large écosystème d'experts, plutôt que de se baser sur un « secret industriel ».

Chiffrement classique

Historique

Les premiers systèmes de chiffrement (dits « classiques ») reposaient souvent sur des transformations simples du texte en clair (le message à protéger). On trouve notamment :

Le chiffre de Vernam ou « One-Time Pad » (1917)

Principe : Utiliser une clé de la même longueur que le message à chiffrer, et faire un XOR (ou une addition modulo, selon la formulation) entre chaque caractère du message et la clé.

Utilisation unique : La clé doit être utilisée une seule fois, puis détruite.

Sécurité parfaite : Claude Shannon a démontré mathématiquement que si la clé est véritablement aléatoire, qu'elle a la même longueur que le message et qu'elle n'est jamais réutilisée, alors ce système est inattaquable (on parle de "perfect secrecy").

Limite pratique : Il est très contraignant de gérer autant de clés aléatoires que de messages, surtout si ces clés doivent rester secrètes et ne jamais être réutilisées. Dans la pratique, le « One-Time Pad » n'est donc guère employé qu'à des fins très particulières (ex. télégrammes diplomatiques ultra-sensibles, communications militaires pointues, etc.).



Le chiffre de César

Principe : Décaler chaque lettre de l'alphabet d'un certain nombre de positions (par exemple 3 : $A \rightarrow D$, $B \rightarrow E$, $C \rightarrow F$, etc.).

Faible sécurité : Une analyse de fréquences (observer la fréquence des lettres chiffrées) permet de retrouver assez facilement le décalage. En effet, dans une langue donnée, certaines lettres apparaissent plus souvent que d'autres (en français, E est très fréquent, en anglais E ou T, etc.).

Théorie de l'information de Shannon



Claude Shannon est souvent surnommé le « père de la théorie de l'information ». Ses travaux (années 1940) ont révolutionné la vision qu'on avait de la communication et de la cryptographie.

Notion d'entropie : L'entropie mesure l'imprévisibilité ou l'incertitude d'une source d'information. Plus un message est « aléatoire » ou imprévisible, plus son entropie est élevée. Si un texte contient beaucoup de redondances (par exemple, en français, des structures de phrases courantes), il est plus facile pour un attaquant de deviner une partie du contenu ou d'exploiter des régularités.

Sécurité parfaite (perfect secrecy) : Selon Shannon, un système de chiffrement est dit parfaitement sûr si la connaissance du texte chiffré n'apporte **aucune** information sur le texte en clair. Le « masque jetable » remplit cette condition, car pour un message chiffré donné, **tous** les messages de même longueur sont possibles, rendant toute interprétation aussi probable qu'une autre.

Longueur de clé et sécurité : Pour atteindre cette sécurité parfaite en théorie, il faut que la clé soit au moins aussi longue que le message. Si la clé est réutilisée pour un autre message, on perd la propriété d'imprévisibilité parfaite.

Évolution vers la cryptographie moderne



Des chiffres classiques aux algorithmes modernes

Les chiffrements classiques (César, Vigenère, etc.) sont aujourd’hui trop faibles pour la moindre utilisation sérieuse. Les besoins (banque en ligne, e-commerce, télécommunications sécurisées, etc.) demandent des algorithmes bien plus robustes.



Algorithmes symétriques et asymétriques

Symétriques : Comme le DES (aujourd’hui obsolète), Triple DES, AES, ChaCha20, etc. Ici, la même clé sert à chiffrer et à déchiffrer.

Asymétriques : Comme RSA, Diffie-Hellman, ECC (cryptographie à courbes elliptiques). On sépare la clé en deux moitiés : une clé publique (pour chiffrer) et une clé privée (pour déchiffrer).



Sécurité aujourd’hui

Robustesse mathématique : Ces algorithmes reposent sur des problèmes mathématiques difficiles (logarithme discret, factorisation de grands nombres, etc.).

Standardisation : L’important est d’utiliser des protocoles et des standards approuvés (TLS/SSL, IPSec, PGP, etc.) et des longueurs de clé suffisantes.



Quantique et post-quantique

Cryptographie quantique : Les futurs ordinateurs quantiques pourraient casser certains algorithmes asymétriques actuels (RSA, ECC).

Cryptographie post-quantique : Des nouvelles primitives sont en cours de standardisation (algorithmes basés sur les réseaux euclidiens, par exemple).

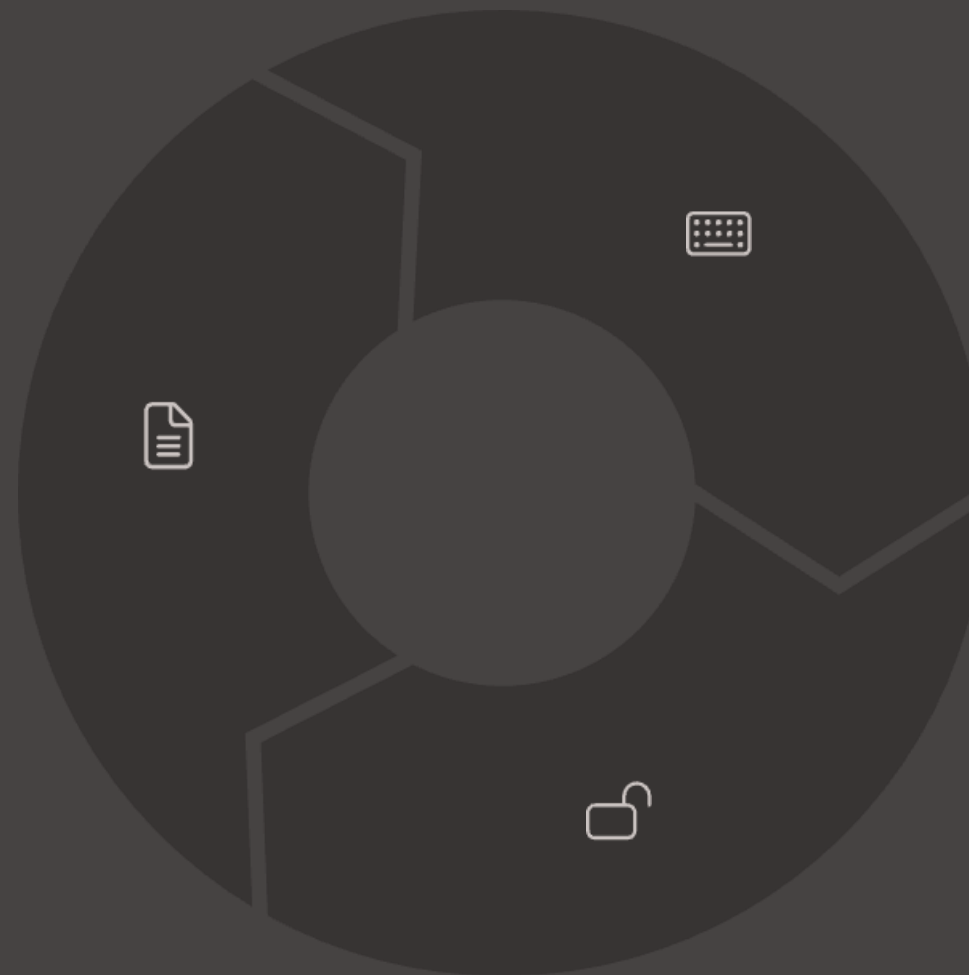
Les principes fondamentaux de la cryptographie (tels que le principe de Kerckhoffs et les travaux de Shannon) restent d’actualité : Un bon système doit rester **transparent** (pas de "sécurité par l’obscurité"). La **solidité** doit reposer principalement sur la **qualité de la clé** et les fondements mathématiques. Les chiffrements classiques, bien qu’historiquement importants, ne suffisent plus pour protéger les communications actuelles. La **théorie de l’information** de Shannon et l’idée d’**entropie** permettent de comprendre pourquoi certains systèmes (One-Time Pad) offrent une "sécurité parfaite", mais aussi pourquoi ils sont peu pratiques. Les algorithmes modernes (AES, RSA, ECC, etc.) sont conçus et analysés selon ces principes, et la cryptographie post-quantique se prépare à la prochaine évolution.

En somme, la cryptographie est un domaine qui tire sa fiabilité à la fois d’une réflexion théorique rigoureuse et d’une mise en œuvre pratique robuste et ouverte au regard de la communauté scientifique.

Introduction à la cryptanalyse

KPA (Known Plaintext Attack)

L'attaquant connaît un couple (clair, chiffré). Exemple : Un espion découvre qu'un message chiffré correspond à « RENDEZ-VOUS DEMAIN 10H ». Il peut utiliser cette information pour déduire la clé ou des informations partielles sur l'algorithme.



CPA (Chosen Plaintext Attack)

L'attaquant peut choisir un texte en clair et obtenir son chiffré. Exemple : Une situation où on a accès à un encrypteur (oracle de chiffrement). On envoie « AAAA... » pour essayer de repérer des patterns dans le chiffré.

CCA (Chosen Ciphertext Attack)

L'attaquant peut choisir un texte chiffré et obtenir le texte en clair (ou une partie d'information sur le clair). Exemple : Oracle de déchiffrement. Si l'attaquant peut intercepter un chiffré, le modifier et le soumettre au système, il peut exploiter les réponses pour en déduire la clé ou le clair.

Exemples simples d'attaques



Analyse de fréquences

Principe : La plupart des langues naturelles ont des fréquences statistiques de lettres caractéristiques (en français : E, S, A, R, T... ; en anglais : E, T, A, O, I...).

Application : Pour un chiffre de César ou toute substitution monoalphabétique, en comparant les fréquences obtenues sur un texte chiffré avec les fréquences usuelles d'une langue, on peut en déduire la clé de substitution.

Limites : Peu efficace si le chiffrement mélange ou masque suffisamment les fréquences (ex. chiffre polyalphabétique, chiffre moderne).



Coincidence de Friedman (Indice de coïncidence)

Principe : Permet d'estimer la longueur de la clé pour un chiffrement de type polyalphabétique (ex. Vigenère).

Explication simplifiée : On décale le texte chiffré de 1, 2, 3... positions et on compte le nombre de « coïncidences » (lettres identiques). Plus la clé est courte, plus on retrouve ces corrélations.

Usage : Une fois la longueur de la clé estimée, on peut découper le texte en sous-ensembles correspondant à chaque lettre de la clé, puis appliquer une analyse de fréquences classique sur chaque sous-message.