Table of contents

Notions de base en cryptographie

- Introduction aux **concepts fondamentaux** de la cryptographie.
- Présente les **principes de sécurité**, les **types de systèmes**, et les **modèles d'attaque**.
- Inclut des **systèmes historiques** et des techniques complémentaires.

Principe de Kerckhoffs

- Principe fondamental : la **sécurité repose uniquement sur la clé**, pas sur le secret de l'algorithme.
- Le système doit rester **sûr même si l'algorithme est public**.
- La clé doit être **facilement modifiable** et le système **simple à utiliser**.

- Rejet explicite de la **sécurité par l'obscurité**.

Ultra-synthèse

- Sécurité basée sur la clé
- Algorithme public
- Pas de sécurité par l'obscurité

Version originale

Principe de Kerckhoffs

Auguste Kerckhoffs publie en **1883** deux articles définissant **six principes** pour les chiffrements militaires :

1. Le système doit être **pratiquement, voire mathématiquement indéchiffrable**.
2. Il ne doit **pas nécessiter de confidentialité** et rester sûr même s'il tombe aux mains de l'ennemi.
3. La **clé** doit pouvoir être **mémorisée, transmise et modifiée facilement**, sans notes écrites.
4. Le système doit être **compatible avec les communications télégraphiques**.
5. Il doit être **portable** et utilisable par **une seule personne**.
6. Il doit être **simple à utiliser**, sans procédures complexes ni contraintes excessives.

Kerckhoffs affirme dès le **XIXe siècle** que la sécurité doit être **mathématiquement démontrable** et qu'il **n'existe pas de sécurité par l'obscurité**.

Classification des systèmes de cryptage

Sécurité inconditionnelle

(*unconditional security / perfect secrecy*)

- Sécurité **indépendante de la puissance de calcul**.
- **Ciphertext** n'apporte aucune info sur le **plaintext**.
- Conditions : **clé message**, **jamais réutilisée**.
- Usage surtout **théorique**.
- **Exemple** : *one-time pad*.

As hard as / équivalent / provable security

- Cryptanalyse **aussi difficile** qu'un **problème mathématique difficile**.
- **RSA** et **Rabin** prouvés équivalent à la factorisation.
  - Démontrée par **réduction** (*reduction proof*).
- Concept central mais **controversé**.

Sécurité calculatoire

(*computational security / practical security*)

- Sécurité basée sur le **coût irréaliste des attaques**.
- Catégorie la plus utilisée en pratique.
- **Exemples** : **AES**, **DES**, **IDEA**, **RC4**.

Ultra-synthèse

- **Inconditionnelle** : parfaite, théorique (*one-time pad*).
- **Provable security** : équivalence à problème mathématique difficile.
- **Calculatoire** : sûre en pratique.

Version originale

- **Sécurité inconditionnelle** (*unconditional security aussi appelée* ***perfect secrecy***) :
  - La sécurité du système de cryptage **n'est pas compromise par la puissance de calcul** destinée à la cryptanalyse.
  - Cette catégorie s'appuie sur la **théorie de l'information** publiée par **Shannon en 1949**.
  - Plus précisément, un système de cryptage est **inconditionnellement sûr** si la probabilité de rencontrer un **plaintext x** après l'observation du **ciphertext correspondant y** est identique à la probabilité à priori de rencontrer le plaintext x.
  - En d'autres termes, le fait de disposer de couples **plaintext/ciphertext (x,y)** ne constitue **aucune aide pour la cryptanalyse**.
  - Une condition nécessaire pour qu'un système soit inconditionnellement sûr est que la **clé soit au moins de la même taille que le message** et, surtout, qu'elle **ne soit pas réutilisée** pour encrypter des messages différents.
  - Cette condition rend ces systèmes **peu adaptés aux besoins cryptographiques habituels** et réduit leur domaine d'intérêt à un **cadre**

théorique.
  - L'exemple classique est le **one-time pad** inventé en **1917** par **J. Mauborgne** and **G. Vernam**.
  - Fondements théoriques des systèmes inconditionnellement sûrs + d'autres exemples dans [**Sti06**].

- **As hard as / équivalent / provable security**
  - Lorsqu'on peut prouver que la cryptanalyse de l'algorithme est **aussi difficile que de résoudre un problème mathématique réputé difficile**.
  - Par exemple la **factorisation de grands nombres**, le calcul de **racines carrées modulo un "composite"**, le calcul de **logarithmes discrets dans un groupe fini**, etc.
  - L'algorithme de **Rabin** et **RSA** (cas générique^1) sont "prouvés" **équivalents à la factorisation**.
  - Une telle preuve s'appelle de "réduction" (**reduction proof**).
  - La notion de **provable security** est à l'origine d'une **importante controverse** dans le monde cryptographique.

- **Sécurité calculatoire** (*computational security aussi appelé* ***practical security***)
  - Un système de cryptage est dans cette catégorie si l'**effort calculatoire nécessaire à le "casser"** en utilisant les meilleures techniques possibles est **au delà** (avec une marge raisonnable) des ressources de calcul d'un adversaire hypothétique.
  - La grande majorité de systèmes de cryptage symétriques (**AES, DES, IDEA, RC4**, etc.) sont dans cette catégorie.

Entropie

- L'**entropie** (Shannon, 1948) mesure la **quantité d'information effective** contenue dans un message.
- L'**entropie conditionnelle** mesure l'incertitude qui reste sur le **plaintext** après observation du **ciphertext**.

Propriétés

- $0 \leq H(X) \leq \log n$
- $H(X) = 0 \rightarrow$ aucune incertitude
- $H(X) = \log n \rightarrow$ tous les résultats équiprobables

Interprétation

- Approxime le **nombre de bits nécessaires** pour encoder X.
- La **redondance** = différence entre codage effectif et entropie.

Entropie conditionnelle

- $H(X \mid Y = y) = -\sum_x P(X = x \mid Y = y) \log P(X = x \mid Y = y)$
- $H(X \mid Y) = \sum_y P(Y = y) H(X \mid Y = y)$
- Mesure l'incertitude restante sur le **plaintext** après observation du **ciphertext**.

Ultra-synthèse

- **Entropie** : quantité d'information d'un message.
- **Entropie conditionnelle** : incertitude sur le plaintext après le ciphertext.
- **Redondance** : différence entre codage effectif et entropie.

Version originale

- Une définition essentielle en cryptographie est la quantité d'information **effective** contenue dans un message.
- Par exemple, les jours de la semaine (*lundi*, ..., *dimanche*) peuvent intuitivement être encodés comme des chaînes de caractères de longueur ($\leq$ len("mercredi")), soit ($8 \times 8 = 64$) bits. Cependant, la quantité d'information effective de la variable *jour de la semaine* peut être encodée de manière optimale sur **3 bits** (car ($2^3 = 8$) est suffisant pour représenter les 7 variations possibles).
- L'**entropie** (Shannon, 1948) est la formalisation mathématique de cette définition.

Définition formelle

Soit X une variable aléatoire avec un ensemble fini de valeurs possibles $x_1, x_2, \ldots, x_n$, telles que $P(X = x_i) = p_i$, avec $0 \leq p_i \leq 1$ et $\sum p_i = 1$. L'entropie de X, notée $H(X)$, est définie par
$H(X) = -\sum_{i=1}^{n} p_i \log p_i = \sum_{i=1}^{n} p_i \log \left(\frac{1}{p_i}\right)$
Par convention : $p_i \log p_i = 0$ si $p_i = 0$. Tous les logarithmes sont en **base 2**.

Interprétation

- Approximation du nombre de bits nécessaires pour encoder les éléments de X.
- La **redondance** est la différence entre le codage effectif et l'entropie.

Propriétés

1. $0 \leq H(X) \leq \log n$
2. $H(X) = 0 \iff \exists i : p_i = 1, p_j = 0 \ \forall j \neq i$
3. $H(X) = \log n \iff p_i = 1/n \ \forall i$

Entropie conditionnelle

- $H(X \mid Y = y) = -\sum_x P(X = x \mid Y = y) \log P(X = x \mid Y = y)$,
- $H(X \mid Y) = \sum_y P(Y = y) H(X \mid Y = y)$

Mesure l'incertitude sur X (plaintext) après avoir observé Y (ciphertext).

Attaques sur les systèmes de cryptage

- **Ciphertext-only** : Adversaire a seulement le ciphertext.
- **Known-plaintext** : Adversaire a des couples plaintext/ciphertext.
- **Chosen-plaintext** : Adversaire peut choisir le plaintext et voir le ciphertext (et essaye de trouver le plaintext pour d'autres messages).
- **Adaptive chosen-plaintext** : dépend des ciphertexts reçus.
- **Chosen-ciphertext** : Adversaire choisit le ciphertext et obtient le plaintext (vise à trouver la clé).
- **Adaptive Chosen-ciphertext** : **Chosen-ciphertext** dépend des plaintexts reçus

Oracles et Modèles de Sécurité

Oracles Aléatoires et Modèles de Sécurité

- **Oracle Aléatoire (Random Oracle)** : Une fonction théorique "parfaite" qui renvoie une valeur uniforme et aléatoire pour chaque nouvelle entrée, mais reste déterministe pour une entrée déjà vue.
- **ROM (Random Oracle Model - Modèle de l'Oracle Aléatoire)** : Cadre de preuve mathématique utilisant cet oracle idéal comme substitut aux fonctions de hachage.
- **Modèle Standard** : Cadre où la sécurité repose uniquement sur la puissance de calcul de l'adversaire face à des algorithmes réels.
- **Limite** : Une preuve de sécurité en ROM ne garantit pas la sécurité absolue dans le monde réel (avec SHA-256, etc.).

Version originale

Un **oracle aléatoire** est une entité abstraite accessible aux parties légitimes et aux

adversaires.

- **Comportement** : Il répond aux requêtes d'entrée x par des réponses parfaitement aléatoires $Orc(x)$.
- **Déterminisme** : La seule exception réside dans les entrées précédemment traitées $(x_1, x_2, ..., x_n)$. Si $x'_1 = x_1$, alors $Orc(x'_1) = Orc(x_1)$.
- **Modélisation** : On le modélise par une fonction $Orc : X \to Y$ où $\forall x \in X, \Pr(Orc(x) = y) = \frac{1}{|Y|}$.
- **Utilité** : Il se comporte comme une **fonction de hachage cryptographique « idéale »**, outil précieux pour prouver la sécurité dans le **Modèle d'Oracle Aléatoire**.
- **Comparaison** : Le **modèle standard** limite les adversaires par des facteurs computationnels. Un protocole sûr dans le modèle d'oracle aléatoire peut devenir vulnérable s'il est utilisé avec une fonction de hachage « réelle » (SHA-1, SHA-256).

Oracles de Chiffrement, Déchiffrement et Signature

- **Fonction** : Entités qui exécutent des opérations (chiffrer/signer) pour l'adversaire en utilisant des clés secrètes sans jamais les révéler.
- **Cryptographie symétrique** : L'oracle fournit $E_k(x)$ ou $D_k(y)$.
- **Cryptographie asymétrique** : L'oracle est crucial pour les opérations privées (déchiffrement/signature), car les opérations publiques sont déjà libres d'accès.

Version originale : Oracles Opérationnels

Un **oracle de chiffrement/déchiffrement/signature** est une entité abstraite offrant un service « à la demande ».

- **Accès aux clés** : Il utilise les **mêmes clés que les propriétaires légitimes** (systèmes symétriques et asymétriques) sans les divulguer.
- **Primitives symétriques** : Pour une primitive E et une clé k, il renvoie $y = E_k(x)$ ou le clair x correspondant.
- **Systèmes à clé publique** : L'oracle n'est nécessaire que pour les opérations à **clé privée** ($priv_k$).
  - **Déchiffrement** : renvoie x tel que $E'_{pubk}(x) = y$.
  - **Signature** : Pour un système S, il renvoie $y = S_{privk}(x)$.
- **Attaques** : Les modèles d'attaques par **texte clair choisi** (CPA) et par **texte chiffré choisi** (CCA) reposent sur la mise à disposition de ces oracles pour l'adversaire.

Indiscernabilité et Sécurité Sémantique (IND-CPA)

- **Propriété** : Un adversaire ne doit pas pouvoir distinguer les chiffrés de deux messages clairs différents.
- **IND-CPA (Indistinguishability under Chosen Plaintext Attack - Indiscernabilité sous attaque à texte clair choisi)** : Si l'adversaire ne devine le bon message qu'avec une probabilité de $1/2 + \epsilon$, le système est considéré comme sûr.
- **Sécurité Sémantique** : Équivalente à l'IND-CPA, elle assure qu'aucune information utile ne fuite du chiffré.

Version originale : Sécurité Sémantique

L'indiscernabilité des textes chiffrés garantit l'incapacité de distinguer les chiffrés de messages clairs donnés.

- **Expérience (Jeu de sécurité IND-CPA)** :
  1. L'adversaire choisit deux messages M_0 et M_1.
  2. L'oracle choisit un indice aléatoire $i \in \{0, 1\}$ et renvoie $c_i = E_k(M_i)$.
  3. L'adversaire peut effectuer d'autres calculs ou appels oracles.
- **Définition IND-CPA** : Le système est sûr si l'avantage de l'adversaire est **négligeable** ($Prob = 1/2 + \epsilon$ avec ϵ petit).
- **Note** : En clé publique, l'oracle de chiffrement est inutile car l'adversaire possède déjà la clé publique. L'IND-CPA offre la **sécurité sémantique**.

Le Chiffrement Probabiliste et l'OAEP

- **Problème** : Le chiffrement déterministe permet les **attaques par dictionnaire** (comparaison de chiffrés connus).
- **Solution** : Ajouter de l'aléa au message avant chiffrement pour que $E(M)$ soit différent à chaque exécution.
- **OAEP (Optimal Asymmetric Encryption Padding - Remplissage asymétrique optimal)** : Standard utilisé avec RSA. Il combine le message P avec un nombre aléatoire R via des fonctions de hachage h et des XOR ($\oplus$).

Version originale : Déterminisme vs Probabilisme

Le comportement **déterministe** (mêmes entrées = mêmes sorties) crée des failles.

- **Exemple** : Si Alice envoie “Oui” ou “Non”, l’adversaire peut calculer $C_{yes} = E_{pub}(\text{"}Oui\text{"})$ et comparer. Il peut créer un **livre de codes** (dictionnaire) pour identifier les messages sans casser la clé.
- **Chiffrement probabiliste** : Ajoute un caractère aléatoire. L’objectif est la sécurité sémantique pour la clé publique.
- **OAEP** : Utilisé dans **RSA-PKCS1**. Le texte P est combiné avec un aléa R :
  - $M_1 := P \oplus h(R)$
  - $M_2 := R \oplus h(M_1)$
  - Le chiffrement porte sur M_1 et M_2. Au déchiffrement, on retrouve $R = M_2 \oplus h(M_1)$, puis $P = h(R) \oplus M_1$.

Ultra-synthèse

- **Oracle Aléatoire** : Fonction de hachage “idéale” (modèle théorique).
- **Oracles CPA/CCA** : Simulent un accès à la clé secrète pour tester la résistance.
- **IND-CPA** : Impossibilité de distinguer deux chiffrés (Sécurité Sémantique).
- **Chiffrement Probabiliste** : Indispensable pour contrer les livres de codes (attaques par dictionnaire).
- **OAEP** : Méthode de padding (remplissage) ajoutant l’aléa nécessaire au RSA.

Histoire de la Cryptographie et Sécurité Inconditionnelle

Systèmes de Cryptage Historiques

La cryptographie a longtemps été limitée à la seule recherche de la **confidentialité**. Les systèmes historiques reposent sur deux principes fondamentaux : la **substitution** et la **transposition**.

- **Chiffre de César** (substitution mono-alphabétique) : Décalage fixe des lettres. Très vulnérable à l’**analyse de fréquences**.
- **Chiffre de Vigenère** (substitution polyalphabétique) : Utilisation d’une clé pour varier le décalage. Plus complexe, mais cassable en identifiant la longueur de la clé.
- **Chiffre de Transposition** : Réorganisation des caractères du texte original selon une permutation définie par une clé.

Version originale : Cryptographie Historique

Pendant des siècles la **confidentialité** a été la seule application de la cryptographie…

- **I av. JC, Caesar Cipher** : **Cryptage à substitution mono-alphabétique** $e_k(x) = (x + k) \pmod{26}$, $d_k(y) = (y - k) \pmod{26}$ où $x, y, k \in \mathbb{Z}_{26}$.

  - Exemple: E_1('bonjour') = 'cpokpws'.
  - **Cryptanalyse** : facile, basée sur la **fréquence des caractères**.
- **XVI siècle, Vigenère** : **Cryptage à substitution polyalphabétique** $e_k(x_1, \dots, x_n) = (x_1 + k_1, \dots, x_m + k_m, x_{m+1} + k_1, \dots)$ (mod 26).
  - **Cryptanalyse** : trouver la **taille** m **de la clé** en identifiant les portions de ciphertext répétées et analyser les blocs séparés comme dans le Caesar Cipher.
- **Transposition Ciphers** (Porta, 1563) : La clé définit une **permutation** sur le plaintext.
- Ces techniques sont toujours à la base des systèmes de cryptage actuels (ex: **Enigma**, qualifiée par W. Churchill d'arme secrète ayant gagné la guerre).

Le One-Time Pad (Masque Jetable)

Le **One-Time Pad** (OTP), ou **chiffre de Vernam**, est le seul système prouvé **inconditionnellement sûr** (sécurité parfaite).

- **Principe** : Le message est combiné à une clé de même longueur via l'opération XOR ($\oplus$).
- **Sécurité Inconditionnelle** : L'observation du message chiffré n'apporte aucune information sur le message clair. Même un adversaire avec une puissance de calcul infinie ne peut pas le briser.
- **Contraintes de Shannon** : La clé doit être **aussi longue que le message**, purement **aléatoire**, et **utilisée une seule fois**.
- **Réutilisation de la clé** : Si une clé est réutilisée pour deux messages, un attaquant peut éliminer la clé par XOR ($y_a \oplus y_b = x_a \oplus x_b$) et retrouver les messages clairs.

Version originale : Le One-Time Pad

Soit $n \geq 1$ et les espaces P, C, K tels que $P, C, K = (\mathbb{Z}_2)^n$. Les opérations d'encryption et decryption d'un **one-time pad** (Vernam Cipher) sont : $E_k(x_i) = x_i \oplus k_i$ et $D_k(y_i) = y_i \oplus k_i$ pour $1 \leq i \leq n$.

- **Sécurité inconditionnelle** : Si k_i sont aléatoires et indépendants, l'observation des ciphertexts n'aide pas la cryptanalyse. L'**entropie** de X ne diminue pas : $H(X|C) = H(X)$.
- **Théorème de Shannon** : Condition nécessaire : $H(K) \geq H(X)$. La longueur de la **clé aléatoire** doit être au moins aussi grande que celle du plaintext.
- **Réutilisation de clé** : $y_a \oplus y_b = x_a \oplus x_b$. Avec des messages de faible entropie, on retrouve les clairs et la clé ($k = y_a \oplus x_a$).
- Vulnérable à l'attaque **Known Plaintext** (si la clé est réutilisée).

- Problème majeur : La **distribution et gestion des clés** de grande taille. Relancé par la **cryptographie quantique** proposant des canaux confidentiels de distribution de clés de longueur illimitées.

Stéganographie

À l'inverse de la cryptographie qui rend le message illisible, la **stéganographie** dissimule l'existence même du message.

- **Méthode** : Utiliser un "canal subliminal" (un support innocent comme une image ou un texte banal).
- **Technique moderne** : Insertion de données dans les **bits les moins significatifs** (LSB - *Least Significant Bits*) de fichiers multimédias, permettant de cacher de gros volumes de données sans altération visible.

Version originale : Stéganographie

La **stéganographie** cache un message à l'intérieur d'un autre. Éléments constituants :

1. Un **canal physique ou logique** différent (canal subliminal).
2. Un **mécanisme secret** pour identifier ce canal.

- **Exemples classiques** : Premières lettres des mots d'un texte, encre invisible.
- **Exemple moderne** : Utiliser les **least significant bits** (bits les moins significatifs) des frames d'un CD Photo.
- Pour une image 2048x3072 (RGB 24 bits), cacher un message sur 1 bit permet de stocker **2.3 Mb** sans détériorer la qualité.

Ultra-synthèse

- **Historique** : Substitution (César/Vigenère) et Transposition (permutation).
- **One-Time Pad** : Sécurité absolue si la clé est aléatoire, unique et aussi longue que le message ($H(K) \geq H(X)$).
- **Stéganographie** : Cacher l'existence du message (ex: technique des LSB dans les images).