Sécurité avec Python : Introduction à la Cryptographie

Présenté par: Arij AZZABI

June 15, 2025

Qu'est-ce que la Cryptographie ?

Définition

La cryptographie est l'art et la science de sécuriser les communications en les rendant inintelligibles sans une clé spécifique.

Objectifs principaux :

- **Confidentialité :** Protéger les données contre les regards indiscrets.
- **Intégrité :** Garantir que les données n'ont pas été modifiées.
- **Authentification :** Vérifier l'identité des parties impliquées.
- **Non-répudiation :** Empêcher qu'une partie nie avoir envoyé un message.

Les trois principaux types

Cryptographie Symétrique

Utilisation : Une seule clé partagée pour chiffrer et déchiffrer.
Exemple : AES (Advanced Encryption Standard).

Cryptographie Asymétrique

Utilisation : Une paire de clés : clé publique et clé privée.
Exemple : RSA (Rivest-Shamir-Adleman).

Hachage

Utilisation : Génère une empreinte numérique fixe à partir d'un message.
Exemple : SHA-256 (Secure Hash Algorithm).

Principe de la Cryptographie Symétrique

Concept

Une même clé est utilisée pour le chiffrement et le déchiffrement.

Exemple concret :

- Texte clair : “HELLO”
- Clé : “SECRET”
- Texte chiffré : “KHOOR”

Avantages : Rapide, simple à mettre en œuvre.
Inconvénients : Distribution sécurisée des clés.

Principe de la Cryptographie Asymétrique

Concept

Utilise une clé publique pour chiffrer et une clé privée pour déchiffrer.

Exemple concret :

- Clé publique : “PUB123”
- Texte clair : “HELLO”
- Texte chiffré : “@4F!Z” (par la clé publique)
- Déchiffré par la clé privée : “HELLO”

Avantages : Pas besoin de partager la clé privée.
Inconvénients : Plus lent que la cryptographie symétrique.

Principe du Hachage

Concept

Transforme un message en une empreinte numérique fixe.

Exemple concret :

- Message : “HELLO”
- Hachage SHA-256 : “2CF24DBA5FB0A30E26E83B2AC5B9E29E1B161E5C”

Avantages : Permet de vérifier l’intégrité des données.
Inconvénients : Non réversible.

Applications de la Cryptographie

Exemples d'Applications

- **Sécurisation des communications :** HTTPS, messageries chiffrées (Signal, WhatsApp).
- **Protection des données :** Chiffrement des fichiers.
- **Authentification :** Signatures numériques, mots de passe hashés.
- **Blockchain :** Utilisation de hachage pour valider les transactions.

Les systèmes numériques

Les systèmes numériques sont utilisés pour représenter des informations à l'aide de symboles ou de chiffres dans un format compréhensible par les ordinateurs. Les trois systèmes de numération les plus courants sont :

- **Binaire** (base 2)
- **Décimal** (base 10)
- **Hexadécimal** (base 16)

L'importance de ces systèmes réside dans leur utilisation dans la programmation, le stockage de données et les opérations cryptographiques.

Système Binaire (Base 2)

Le système binaire utilise deux symboles : 0 et 1. Chaque bit (binary digit) représente une puissance de 2. **Exemple :**

- Le nombre binaire 1011 représente :

  $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11$ (en décimal).

- Le nombre binaire 1101 représente :

  $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$ (en décimal).

Le binaire est la base de la représentation des données dans les ordinateurs.

Système Hexadécimal (Base 16)

Le système hexadécimal utilise 16 symboles : 0-9 et A-F (où A = 10, B = 11, ..., F = 15). **Exemple :**

- Le nombre hexadécimal `A3` en base 16 représente :

$$A \times 16^1 + 3 \times 16^0 = 10 \times 16 + 3 = 163 \text{ (en décimal).}$$

- Le nombre hexadécimal `7F` représente :

$$7 \times 16^1 + F \times 16^0 = 7 \times 16 + 15 = 127 \text{ (en décimal).}$$

Conversion de binaire en hexadécimal :

- $1101_2 = \mathtt{D}_{16} = 13$ (décimal).$1111_2 = \mathtt{F}_{16} = 15$ (décimal).

Conversion entre systèmes

Conversion Binaire $\leftrightarrow$ Décimal :

- Pour convertir un nombre binaire en décimal, additionnez les puissances de 2 pour chaque bit égal à 1.
- Exemple : $1011_2 \rightarrow 8 + 2 + 1 = 11$ (décimal).

Conversion Hexadécimal $\leftrightarrow$ Décimal :

- Multipliez chaque chiffre hexadécimal par la puissance de 16 correspondant à sa position.
- Exemple : $A3_16 = 10 \times 16 + 3 = 163$ (décimal).

Exemple de Conversion :

- $1101_2 = D_{16} = 13$ (décimal).$15(dcimal) = 1111_2 = F_{16}$.

Définition du XOR

Le XOR (OU Exclusif) est une opération logique binaire. Les règles sont simples :

- $0 \oplus 0 = 0$
- $1 \oplus 0 = 1$
- $0 \oplus 1 = 1$
- $1 \oplus 1 = 0$

En d'autres termes, XOR renvoie `true` si les deux bits sont différents, et `false` s'ils sont identiques.

Propriétés du XOR

Le XOR possède plusieurs propriétés intéressantes :

- **Commutativité :**
$$A \oplus B = B \oplus A$$
- **Associativité :**
$$(A \oplus B) \oplus C = A \oplus (B \oplus C)$$
- **Identité :**
$$A \oplus 0 = A$$
- **Inversité :**
$$A \oplus A = 0$$

Ces propriétés rendent le XOR utile pour des applications telles que la cryptographie (notamment le chiffrement par XOR).

Exemple de XOR avec des valeurs binaires

Prenons l'exemple de deux nombres binaires :

- $1101_2 \oplus 1011_2 = 0110_2$

Explication : Chaque bit est comparé en appliquant la règle du XOR :

- $1 \oplus 1 = 0$
- $1 \oplus 0 = 1$
- $0 \oplus 1 = 1$
- $1 \oplus 1 = 0$

Le résultat est donc 0110_2.

Exemple pratique avec des valeurs hexadécimales

Si nous appliquons XOR à deux nombres hexadécimaux :

- 11011011_2 XOR $10101010_2 = 01011010_2$

Chaque bit est comparé :

$$1 \oplus 1 = 0,\ 1 \oplus 0 = 1,\ 0 \oplus 1 = 1,\ 1 \oplus 1 = 0$$

XOR avec des nombres hexadécimaux : Exemple détaillé

Prenons l'exemple de XOR entre les valeurs hexadécimales `A3` et `7F`.

- $\texttt{A3}_{16} = 10100011_2$
- $\texttt{7F}_{16} = 01111111_2$

Effectuons maintenant l'opération XOR bit par bit :

$$\begin{array}{r} 10100011 \text{ (A3 en binaire)} \\ \oplus \\ 01111111 \text{ (7F en binaire)} \\ \hline 11011100 \text{ (résultat en binaire)} \end{array}$$

Le résultat en binaire est `11011100`, ce qui est `DC` en hexadécimal.
Donc, `A3` XOR `7F` = `DC`.

Qu'est-ce que l'Encodage ?

Définition

L'encodage est le processus de transformation des données dans un format lisible ou transmissible, sans les sécuriser.

Objectifs principaux :

- Assurer la compatibilité des systèmes (transmission, stockage).
- Rendre les données lisibles.

Principaux Types d'Encodage

Base64

Principe : Transforme des données binaires en caractères lisibles (A-Z, a-z, 0-9, +, /).

Exemple détaillé :

- Données d'origine : Texte clair : `HELLO`
- Conversion en binaire : `01001000 01000101 01001100 01001100 01001111`
- Division en groupes de 6 bits : `010010 000100 010101 001100 010011 001100 001111`
- Correspondance Base64 : `SEVMTE8=`

Principaux Types d'Encodage

Hexadécimal

Principe : Représente chaque octet en deux chiffres hexadécimaux.
Exemple détaillé :

- Données d'origine : Texte clair : `HELLO`
- Conversion en ASCII : `H=72, E=69, L=76, O=79`
- Conversion en hexadécimal : `48 65 6c 6c 6f`
- Encodé : `48656c6c6f`

Principaux Types d'Encodage

URL Encoding

Principe : Encode des caractères spéciaux pour les rendre compatibles avec les URLs.

Exemple détaillé :

- Données d'origine : Texte clair : `Hello World!`
- Substitution des espaces et caractères spéciaux :
  - `H`, `e`, `l`, `o`, `W`, `r`, `d` restent inchangés.
  - L'espace devient %20.
  - Le point d'exclamation `!` devient %21.
- Encodé : `Hello%20World%21`

Principe du Chiffrement de César

Définition

Le chiffrement de César est un chiffrement par substitution où chaque lettre d'un texte est décalée d'un nombre fixe de positions dans l'alphabet.

Formule mathématique :

$$C(x) = (x + k) \mod 26$$

- x : position de la lettre dans l'alphabet (A=0, B=1, ..., Z=25).
- k : clé de décalage.

Utilisation historique : Utilisé par Jules César pour transmettre des messages secrets.

Exemple du Chiffrement de César

Texte clair : HELLO
Clé (k) : 3
Étapes :

- H $\rightarrow$ K
- E $\rightarrow$ H
- L $\rightarrow$ O
- L $\rightarrow$ O
- O $\rightarrow$ R

Texte chiffré : KHOOR

Principe du Chiffrement ROT13

Définition

Le chiffrement ROT13 est un cas particulier du chiffrement de César où le décalage est fixé à 13. Appliquer ROT13 deux fois revient au texte original.

Formule mathématique :

$$C(x) = (x + 13) \mod 26$$

Utilisation : Principalement utilisé pour cacher des spoilers ou des messages triviaux.

Exemple du Chiffrement ROT13

Texte clair : HELLO
Étapes :

- H $\rightarrow$ U
- E $\rightarrow$ R
- L $\rightarrow$ Y
- L $\rightarrow$ Y
- O $\rightarrow$ B

Texte chiffré : URYYB

Principe de la Substitution Simple

Définition

La substitution simple remplace chaque lettre d'un texte par une autre selon une correspondance fixe prédéfinie. L'ordre des lettres peut être complètement réorganisé.

Exemple de correspondance :

- A $\rightarrow$ Z
- B $\rightarrow$ Y
- C $\rightarrow$ X
- ...
- Z $\rightarrow$ A

Exemple de Substitution Simple

Texte clair : HELLO
Correspondance : A $\rightarrow$ Z, B $\rightarrow$ Y, C $\rightarrow$ X, ...
Étapes :

- H $\rightarrow$ S
- E $\rightarrow$ V
- L $\rightarrow$ O
- L $\rightarrow$ O
- O $\rightarrow$ L

Texte chiffré : SVOOL

Principe du Chiffrement de Vigenère

Définition

Le chiffrement de Vigenère utilise une clé répétitive pour effectuer plusieurs décalages de César en fonction des lettres de la clé.

Formule mathématique :

$$C(x) = (x + k_i) \mod 26$$

- k_i : décalage déterminé par la position de la lettre dans la clé.

Exemple du Chiffrement de Vigenère

Texte clair : HELLO
Clé : KEY
Étapes :

- H (clé K = 10) $\rightarrow$ R
- E (clé E = 4) $\rightarrow$ I
- L (clé Y = 24) $\rightarrow$ J
- L (clé K = 10) $\rightarrow$ V
- O (clé E = 4) $\rightarrow$ S

Texte chiffré : RIJVS

Principe de l'opération XOR

Définition

XOR (**OU exclusif**) est une opération logique qui renvoie vrai si une seule des deux entrées est vraie.

Formule :

$$C(x) = P(x) \oplus K$$

où $\oplus$ représente l'opération XOR, $P(x)$ est le texte clair, et K est la clé.

Propriété intéressante :

- Appliquer XOR deux fois avec la même clé retourne le texte original :

$$C(x) \oplus K = P(x)$$

Exemple de chiffrement avec XOR

Texte clair : HELLO
Clé : 0x4F
Étapes :

- Convertissez chaque caractère en binaire.
- Appliquez l'opération XOR avec la clé K.
- Reconvertissez le résultat en texte.