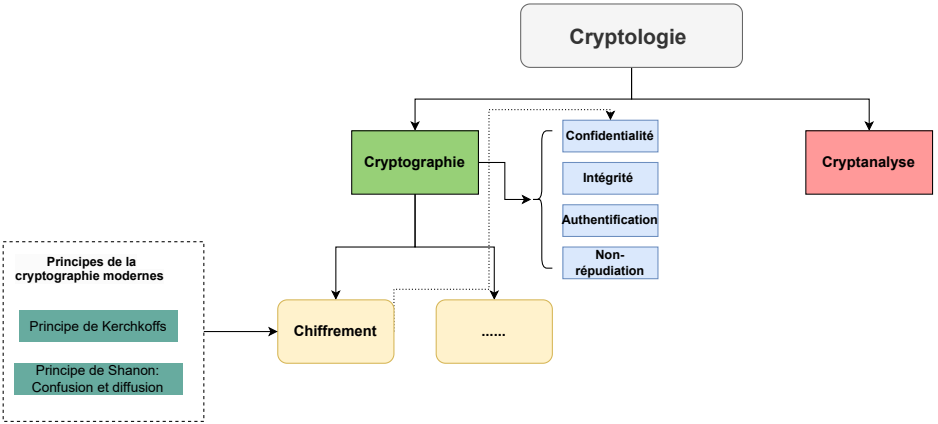
Initiation à la Cryptographie

Standard du chiffrement symétrique AES

February 3, 2025

Awaleh HOUSSEIN

Recap du premier post [1]



[1] https://www.linkedin.com/feed/update/urn:li:activity:7284685244455157761/

Chiffrement symétrique

Le **chiffrement symétrique**, ou à *clé secrète*, utilise une clé k pour transformer un message en clair m en texte chiffré c, rendant le message illisible sans la clé. Le déchiffrement, effectué avec la même clé, récupère le texte en clair, et la sécurité dépend de la confidentialité de la clé.

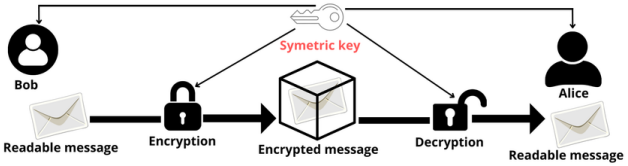


Figure: Chiffrement symétrique (figure source[2])

[2] Mongetro Goint, Cyrille Bertelle, and Claude Duvallet (2023). "Secure Access Control to Data in Off-Chain Storage in Blockchain-Based Consent Systems". In.

Familles de chiffrement symétrique

Deux grandes familles de chiffrement symétrique:

Chiffrement à flot

L'algorithme chiffre chaque bit du message en clair en l'XOR-ant avec un flux de clé de même longueur (ex: comme OTP).

Chiffrement par blocs

L'algorithme chiffre des blocs de texte de taille fixe, chaque bloc étant traité séparément avec une fonction déterministe. Un mode de chiffrement relie les blocs entre eux.

Nous nous concentrons ici sur le chiffrement symétrique par blocs.

Historique de chiffrement par blocs

- **1967** : Feistel chez IBM
  - Horst Feistel (IBM) développe le concept du réseau de Feistel.
  - **LUCIFER** : 1er algorithme (bloc 128 bits, clé 128 bits).
- **1972** : Le NBS (ancêtre du NIST) demande un standard de chiffrement
- 1975 : IBM propose **DES** (version simplifiée de LUCIFER) :
  - Bloc 64 bits / Clé 56 bits.
  - Modifications suggérées par la NSA.
- **1977** : Le NBS adopte le **DES** comme standard de chiffrement (FIPS 46-1, 46-2).
- **1997** : La sécurité du **DES** n'était plus garantie face à une recherche exhaustive de la clé ; **triple-DES** était jugé trop lent.
- **2001**: Après plusieurs tours d'évaluation, le NIST standardise Rijndael (AES) comme remplacement du DES

Source : NIST, FIPS 46-3 (1999), FIPS 197 (2001)

Advanced Encryption Standard (AES)

Algorithme symétrique par blocs

- Taille de clé : 128, 192 ou 256 bits $\Rightarrow$ 10, 12 ou 14 tours.
- Bloc de 128 bits = 16 octets organisés en une matrice 4x4 (**État/State**).

Construction de l'État :

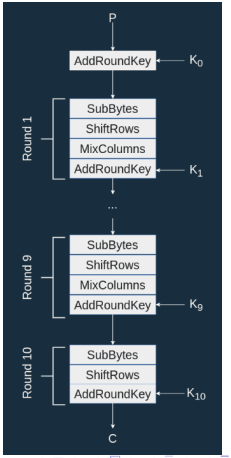
- Chaque bloc de 128 bits est découpé en 16 octets $b_0, b_1, ..., b_{15}$.
- Remplissage **colonne par colonne** :

$$\text{État} = \begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

Étapes du chiffrement AES

Étapes du chiffrement :

- **Key Expansion** : Génère $N + 1$ sous-clés (ex: 11 clés pour AES-128).
- **Tours (Rounds)** :
  - *SubBytes* : Substitution non linéaire via S-Box (ex: `1A` $\rightarrow$ `A2`).
  - *ShiftRows* : Décalage des lignes (0, +1, +2, +3 positions).
  - *MixColumns* : Combinaison linéaire des colonnes (sur le corpd de Galois [a]).
  - *AddRoundKey* : XOR avec la sous-clé du tour [b].



[a] L'AES réalise ses calculs dans le corps de Galois GF(2^8) avec :$P(X) = X^8 + X^4 + X^3 + X + 1$ comme polynôme irréductible. Les lois de compositions internes sont alors : $\oplus$ et $\cdot$

[b] Image source et détails : `https://braincoke.fr/blog/2020/08/the-aes-encryption-algorithm-explained/`

SubBytes (Étape de substitution)

Principe :

- Chaque octet de la matrice d'État est remplacé via une **S-Box** unique.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	...	...	...	...	...	...	...	...	...	...	...	...	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
F	8C	A1	89	...	...	...	...	...	...	...	...	...	...	...	...	...

Résultat :

$$8F(\textit{hex}) \xrightarrow{\text{S-Box}} \mathbf{73}(\mathbf{hex}) \quad \text{(valeur à l'intersection ligne 8 / colonne F)}$$

Note : La S-Box applique une inversion (GF(2^8)) puis une fonction affine pour briser les motifs statistiques (**confusion**).
Exemple : Les octets du tableau sont en **Hexadécimal (hex)** : représente les 8 bits d'un octet en 2 chiffres

Shift rows (décalage de lignes)

- Ligne 1 : Pas de décalage.
- Ligne 2 : Décalage de 1 octet vers la gauche.
- Ligne 3 : Décalage de 2 octets.
- Ligne 4 : Décalage de 3 octets.

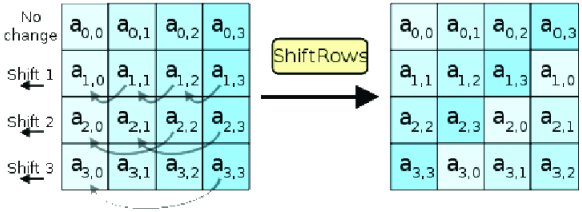


Figure: Fonctionnement de la fonction Shift Rows (figure source[3])

[3] El-Sayed Abdoul-Moaty ElBadawy et al. (n.d.). "A new chaos advanced encryption standard (AES) algorithm for data security". In.

MixColumns : Combinaison linéaire des colonnes

Principe : Chaque colonne de la matrice d'État est multipliée par une matrice fixe dans GF(2^8).

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a' \\ b' \\ c' \\ d' \end{bmatrix}$$

Exemple de calcul (pour un élément) : $a' = (02 \cdot a) \oplus (03 \cdot b) \oplus (01 \cdot c) \oplus (01 \cdot d)$

- $\cdot$ = multiplication dans GF(2^8).
- $\oplus$ = XOR (addition binaire).

Objectif :

- *Diffusion* : Mélanger les octets entre colonnes.
- *Résistance* : Complexifie l'analyse statistique du texte chiffré.

GF(2^8) : Corps fini pour les opérations algébriques (octets = polynômes de degré 7).

AddRoundKey: XOR entre l'État et la sous-clé

Principe : Chaque octet de la matrice d'État est combiné avec la sous-clé via $\oplus$.

État (Text) :

$$\begin{bmatrix} 01 & 23 & 45 & 67 \\ 89 & AB & CD & EF \\ FE & DC & BA & 98 \\ 76 & 54 & 32 & 10 \end{bmatrix}$$

Sous-clé (Key) :

$$\begin{bmatrix} 0F & 15 & 71 & C9 \\ 47 & D9 & E8 & 59 \\ 0C & B7 & AD & D6 \\ AF & 7F & 67 & 98 \end{bmatrix}$$

Calcul pour un octet :

$$01\,(\text{hex}) \oplus 0F\,(\text{hex}) = 0E\,(\text{hex}) \quad (00000001 \oplus 00001111 = 00001110)$$

Résultat final :

$$\begin{bmatrix} 0E & 36 & 34 & AE \\ CE & 72 & 25 & B6 \\ F2 & 6B & 17 & 4E \\ D9 & 2B & 55 & 88 \end{bmatrix}$$

Key Expansion : génération des sous-clés pour chaque tour

Objectif : Générer 10/12/14 sous-clés uniques à partir de la clé principale (selon AES-128/192/256).

Étapes pour AES-128 :

- **Input** : Clé principale (16 octets).
- **Key Schedule** : Découpe en mots de 4 octets (ex: w_0, w_1, w_2, w_3).
- **Pour chaque nouveau mot w_i :**
  - Si i multiple de 4 : - RotWord (rotation de 1 octet). - SubWord (S-Box). - XOR avec RCon[i/4] (constante anti-répétition).
  - Sinon : $w_i = w_{i-4} \oplus w_{i-1}$.

Exemple : Clé principale : `2b7e1516 28aed2a6 abf71588 09cf4f3c`

$$w_4 = w_0 \oplus \text{RotWord}(w_3) \oplus \text{RCon}[1]$$
$$= \texttt{2b7e1516} \oplus \texttt{cf4f3c09} \oplus \texttt{01000000}$$
$$= \texttt{e531291f}$$

RCon : Constantes prédéfinies (ex: `01, 02, 04, 08...` en hexa) pour éviter les motifs répétitifs.

Source : Pour plus de détails sur la génération des sous clés de l'AES: `https://braincoke.fr/blog/2020/08/the-aes-key-schedule-explained`

Déchiffrement de l'AES

Processus inverse du chiffrement :

- Utilisation des sous-clés dans l'ordre inverse
- Transformations inverses :
  - InvSubBytes (S-Box inverse)
  - InvShiftRows (Décalages droits)
  - InvMixColumns (Matrice inverse)
- Même structure en tours (10/12/14)
- AddRoundKey reste identique (XOR)

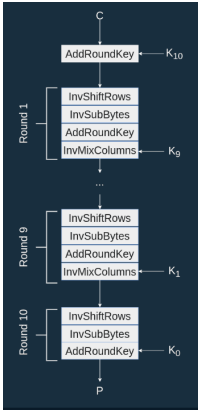


Schéma du déchiffrement AES *Source :* `https://braincoke.fr/blog/2020/08/the-aes-decryption-algorithm-explained/`

Applications concrètes de l'AES

Où utilise-t-on AES ?

- **Sécurité web** : HTTPS (SSL/TLS) pour les protocoles de communications.
- **VPN** : chiffrement des données en transit (OpenVPN, IPsec)
- **Stockage** : chiffrement des fichiers (BitLocker, FileVault, VeraCrypt)
- **Messagerie** : signal, WhatsApp (protocoles E2EE [4])
- **Cloud** : Chiffrement des données chez AWS, Google Cloud.

Pourquoi AES ? Rapidité (matrices 4x4), Sécurité prouvée (NIST), Compatibilité universelle. [5]

[4] End-to-End Encryption (E2EE) : méthode de chiffrement de bout en bout où seuls l'expéditeur et le destinataire peuvent lire les données, sans accès possible par des intermédiaires. Pour en savoir plus : `https://fr.wikipedia.org/wiki/Chiffrement_de_bout_en_bout`

[5] Source : NIST FIPS 197 (2001), `https://csrc.nist.gov/publications/detail/fips/197/final`

À retenir

Chiffrement symétrique : deux catégories principales :

- chiffrement à flot (traitement bit par bit).
- chiffrement par blocs (traitement par blocs fixes).

AES (Advanced Encryption Standard) :

- standard de chiffrement symétrique par blocs (128 bits/bloc).
- clés : 128/192/256 bits $\longrightarrow$ 10/12/14 tours de traitement.

Structure d'un tour AES :

- 4 transformations par tour (sauf dernier tour, sans MixColumns)
  - `AddRoundKey` : XOR avec la sous-clé $\longrightarrow$ ajoute de la **confusion**.
  - `SubBytes` : Substitution non linéaire (S-Box) $\longrightarrow$ ajoute une non linéarité et **confusion**.
  - `ShiftRows` : Décalage des lignes $\longrightarrow$ ajoute de la **diffusion**.
  - `MixColumns` : Combinaison algébrique des colonnes $\longrightarrow$ ajoute de la **diffusion**.

Déchiffrement :

- Étapes inverses (`InvSubBytes`, `InvShiftRows`, etc.).
- Sous-clés appliquées dans l'ordre inverse.