



ALGORITHME DE CHIFFREMENT SYMETRIQUE-AES

Réalisé par:
Jasser Lafi
Aya Skhiri

Plan

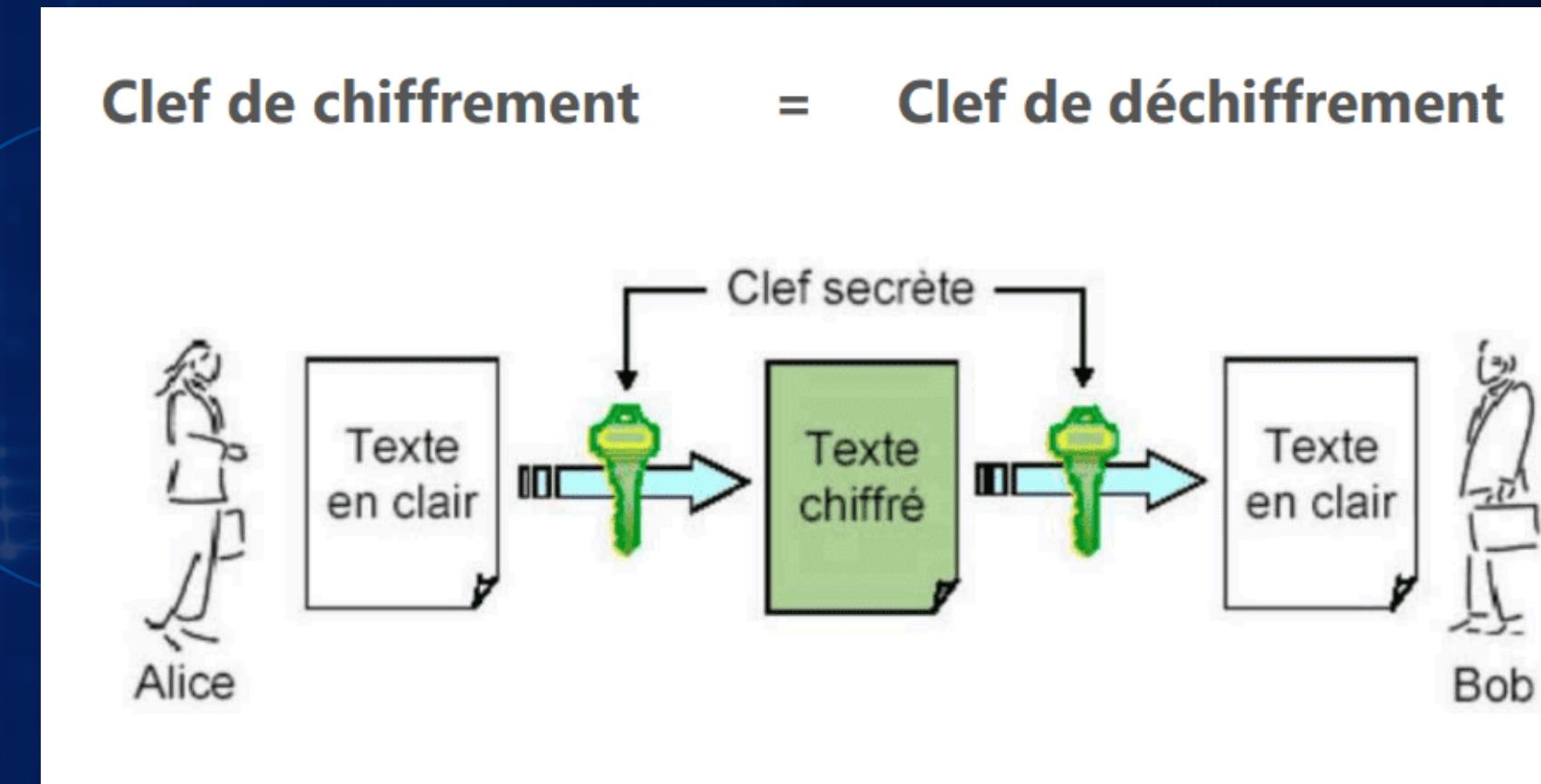
- Introduction
- Vulnérabilités de l'AES
- Caractéristiques et Fonctionnement
- Applications
- Exemple simplifié
- Conclusion



Introduction

AES (ADVANCED ENCRYPTION STANDARD)

- Algorithme de chiffrement symétrique : utilise une seule clé pour chiffrer et déchiffrer les données.
- Aussi connu sous le nom " Rijndael ".
- Inventé en 1997 aux Etats-Unis.
- A été standardisé en 2001 par le NIST (National Institute of Standards and Technology) pour remplacer l'ancien algorithme DES (Data Encryption Standard) qui est devenu vulnérable aux attaques modernes.



Caractéristiques principales de l'AES



Taille de bloc fixe

- AES opère sur des blocs de données (matrice 4x4) de 128 bits (16 octets). Si les données sont plus longues, elles sont divisées en plusieurs blocs de 128 bits.

Taille de clé variable

- AES supporte trois tailles de clé :
 - 128 bits (10 tours de chiffrement)
 - 192 bits (12 tours de chiffrement)
 - 256 bits (14 tours de chiffrement)

Structure en réseau de substitution-permutation (SPN)

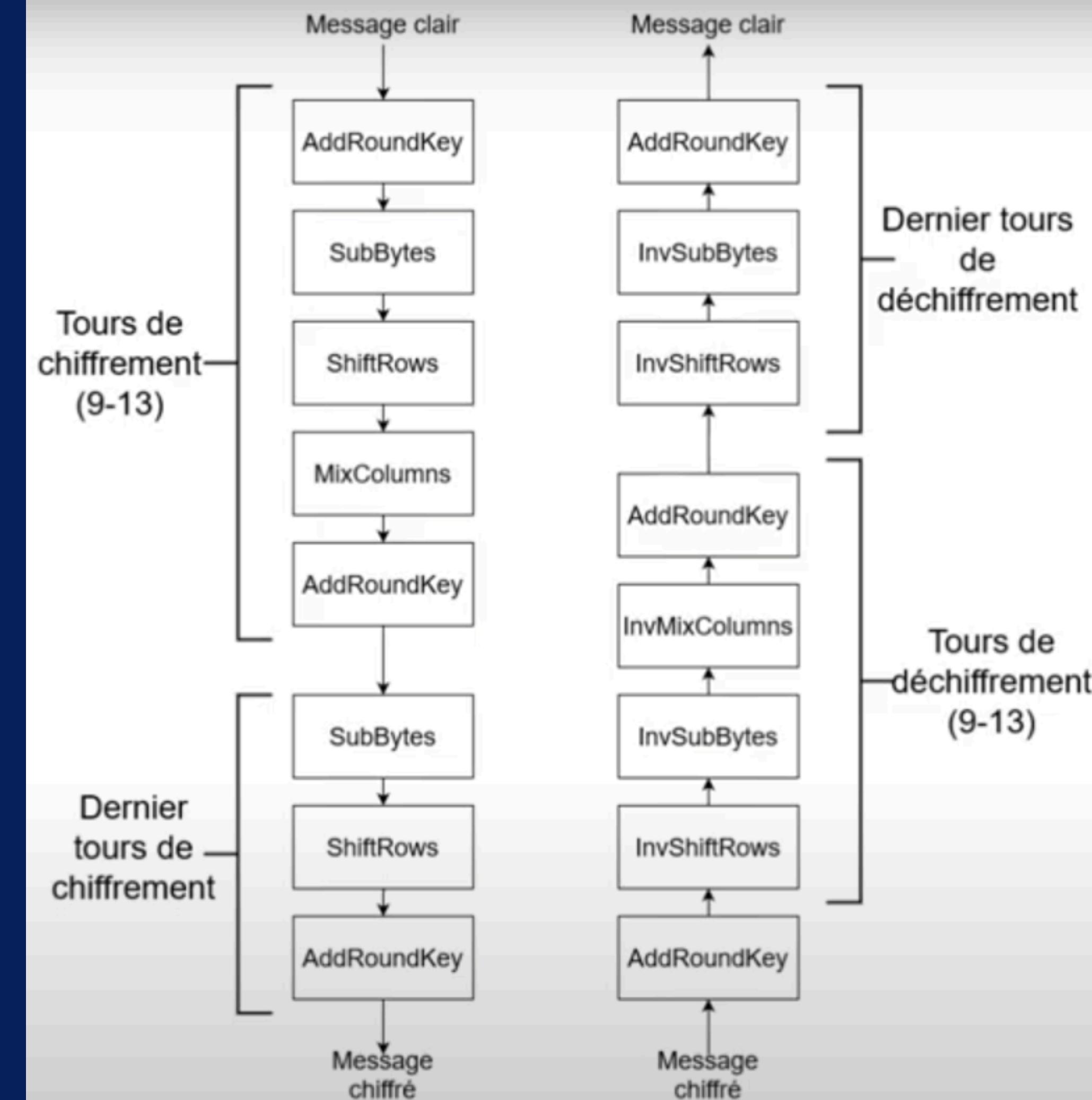
- AES utilise une série d'opérations répétées (appelées rounds) pour transformer le texte clair en texte chiffré. Ces opérations incluent des substitutions, des permutations et des mélanges de données.

Fonctionnement

Le chiffrement AES se déroule en plusieurs étapes, appliquées de manière itérative sur le bloc de données. Voici les étapes principales :

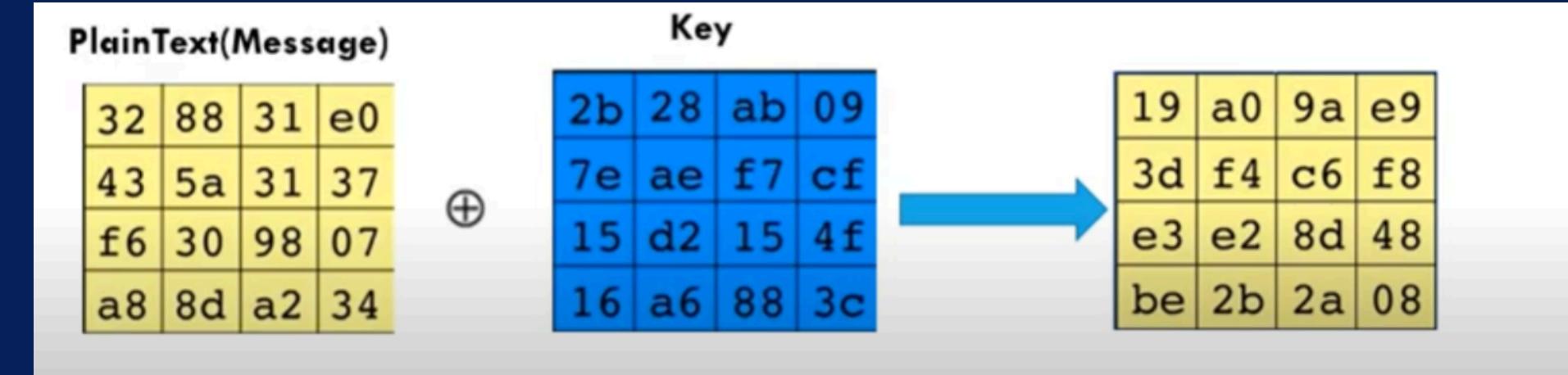
Key Expansion (Expansion de la clé) :

- La clé initiale est étendue pour générer une série de sous-clés (une pour chaque round).



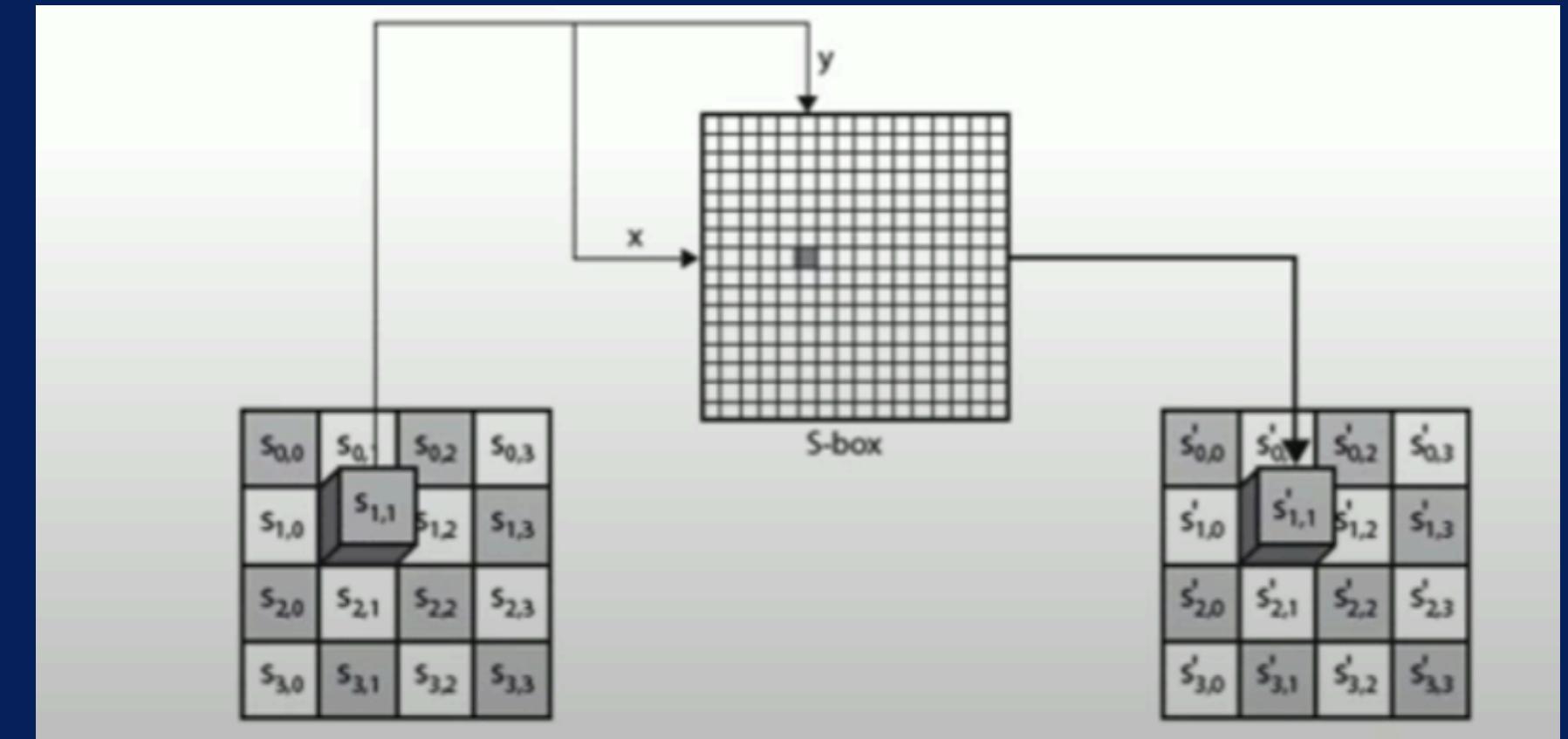
AddRoundKey (Ajout de la sous-clé) :

- Une opération XOR est effectuée entre le bloc de données et la sous-clé du round actuel.



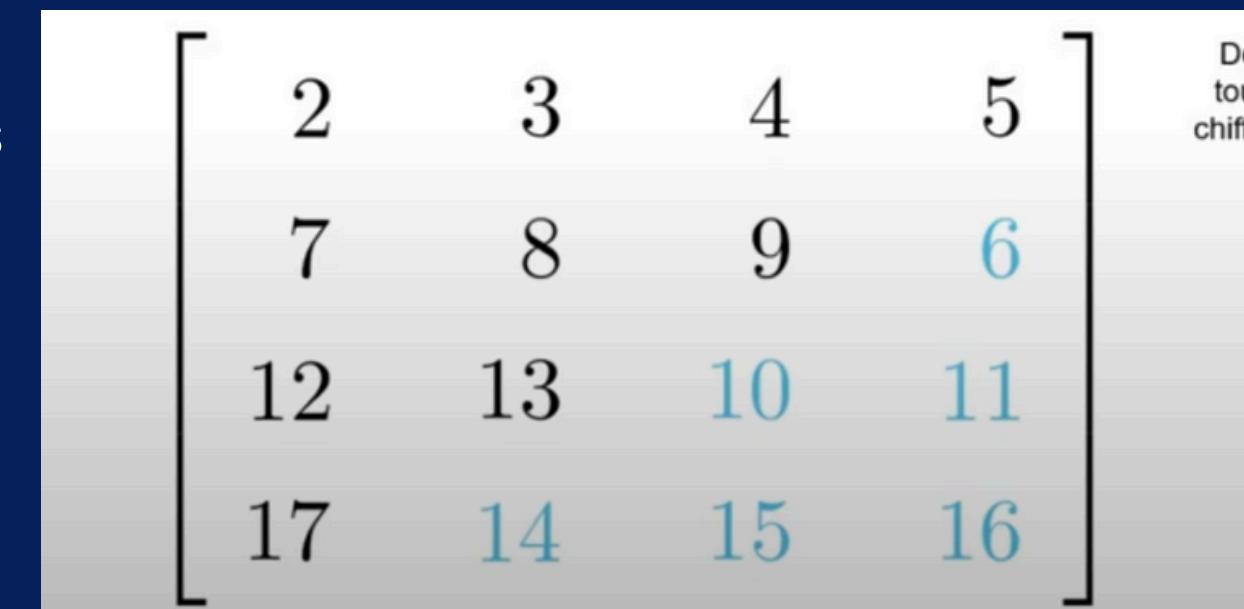
SubBytes (Substitution d'octets) :

- Chaque octet du bloc est remplacé par un autre octet en utilisant une table de substitution appelée S-box.



ShiftRows (Décalage des lignes) :

- Les octets de chaque ligne du bloc sont décalés cycliquement pour introduire de la diffusion.



MixColumns (Mélange des colonnes) :

- Une transformation linéaire est appliquée à chaque colonne du bloc pour mélanger davantage les données.

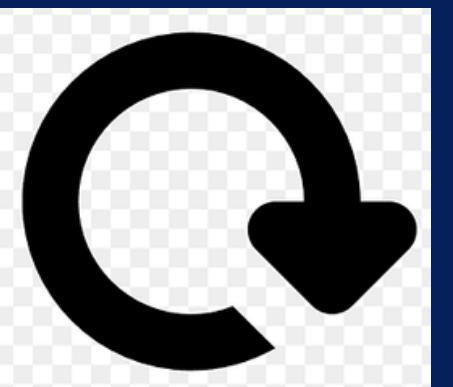
$$\begin{matrix} \text{d4} & \text{e0} & \text{b8} & \text{1e} \\ \text{bf} & \text{b4} & \text{41} & \text{27} \\ \text{5d} & \text{52} & \text{11} & \text{98} \\ \text{30} & \text{ae} & \text{f1} & \text{e5} \end{matrix} \times \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} = \begin{matrix} \text{04} & \text{e0} & \text{48} & \text{28} \\ \text{66} & \text{cb} & \text{f8} & \text{06} \\ \text{81} & \text{19} & \text{d3} & \text{26} \\ \text{e5} & \text{9a} & \text{7a} & \text{4c} \end{matrix}$$

AddRoundKey (Ajout de la sous-clé) :

- Une autre opération XOR est effectuée avec la sous-clé du round actuel.

Répétition des rounds :

- Les étapes SubBytes, ShiftRows, MixColumns et AddRoundKey sont répétées pour un nombre de rounds dépendant de la taille de la clé (10, 12 ou 14 rounds).



Round final :

- Le dernier round omet l'étape MixColumns pour des raisons de symétrie

EXEMPLE SIMPLIFIÉ DE CHIFFREMENT ET DÉCHIFFREMENT



Données initiales :

Bloc de données en clair (plaintext) : 1 2 3 4
(imaginons que c'est un bloc de 4 octets pour simplifier).

Clé AES : 5 6 7 8 (supposée de 4 octets).

Étape 1 : Chiffrement

1. AddRoundKey [Ajouter la clé]

Combiner le bloc de données avec la clé avec l'opération XOR .

$$1 \oplus 5 = 4$$

$$2 \oplus 6 = 4$$

$$3 \oplus 7 = 4$$

$$4 \oplus 8 = 12$$

Résultat après AddRoundKey : 4 4 4 12.

2. SubBytes [Substitution des octets]

On remplace chaque valeur par une autre en utilisant une table de substitution (S-box).

Pour simplifier, imaginons que : 4 devient 9 / 12 devient 3.

Résultat après SubBytes : 9 9 9 3.

3. ShiftRows [Décalage des lignes]

On décale les lignes du bloc.

Pour simplifier, imaginons que nous décalons la deuxième valeur d'une position vers la gauche :

$$9 \ 9 \ 9 \ 3 \rightarrow 9 \ 9 \ 3 \ 9$$

4. MixColumns [Mélange des colonnes]

On applique une transformation mathématique aux colonnes.

Pour simplifier, imaginons que nous ajoutons 1 à chaque valeur :

$$9 \ 9 \ 3 \ 9 \rightarrow 10 \ 10 \ 4 \ 10$$

5. AddRoundKey [Ajouter la clé du tour suivant]

Supposons que la clé du tour suivant est 2 2 2 2.

Bloc de données : 10 10 4 10.

On applique à nouveau opération XOR :

$$10 \oplus 2 = 8$$

$$10 \oplus 2 = 8$$

$$4 \oplus 2 = 6$$

$$10 \oplus 2 = 8$$

Résultat final (texte chiffré) : 8 8 6 8.

Étape 2 : Déchiffrement

1. AddRoundKey [avec la clé du tour]

On applique XOR avec la même clé du tour [2 2 2 2] :

Texte chiffré : 8 8 6 8.

Clé du tour : 2 2 2 2.

XOR : $8 \oplus 2 = 10$ et $6 \oplus 2 = 4$

Résultat : 10 10 4 10.

4. InvSubBytes [Inverse de SubBytes]

On utilise la table de substitution inverse pour retrouver les valeurs d'origine :
9 devient 4 / 3 devient 12
Résultat : 4 4 4 12.

2. InvMixColumns [Inverse de MixColumns]

On soustrait 1 à chaque valeur : 10 10 4 10 ---> 9 9 3 9

5. AddRoundKey [avec la clé initiale]

On applique XOR avec la clé initiale [5 6 7 8] :
Bloc de données : 4 4 4 12.

$$4 \oplus 5 = 1$$

$$4 \oplus 6 = 2$$

$$4 \oplus 7 = 3$$

$$12 \oplus 8 = 4$$

Résultat final (texte en clair) : 1 2 3 4.

Vulnérabilités de l'algorithme AES

- Attaques par canaux auxiliaires : Exploitation d'informations physiques (temps d'exécution, consommation d'énergie) pour déduire la clé.
- Attaques par faute : Introduction de fautes matérielles pour perturber le chiffrement et récupérer des informations sur la clé.
- Implémentations faibles : Une mauvaise implémentation peut exposer AES à des attaques comme les key recovery attacks.
- Attaques cryptanalytiques : Attaques théoriques (ex: attaque biclique) qui réduisent légèrement la complexité de la recherche de clé, mais restent inefficaces en pratique.
- Les attaques par force brute ne sont pas réalistes pour des clés de 128 bits ou plus en raison du nombre astronomique de combinaisons.

Applications de l'AES

- Chiffrement de fichiers : Protéger des fichiers sensibles sur un disque dur ou un cloud.
- Sécurisation des communications : Utilisé dans des protocoles comme TLS/SSL pour sécuriser les échanges sur Internet.
- Protection des données dans les bases de données : Chiffrement des informations sensibles stockées.
- Systèmes embarqués : Utilisé dans des appareils IoT et des cartes à puce.

Conclusion

AES est un algorithme de chiffrement robuste, flexible et largement utilisé pour protéger les données dans de nombreux contextes. Sa sécurité et son efficacité en font un choix privilégié pour les applications modernes.





Merci pour votre attention !

