

Chiffrement symétrique : AES

# Sommaire

<b>Présentation du chiffrement AES :</b>	<b>2</b>
Le chiffrement informatique	2
Le chiffrement dit “symétrique”	2
Le chiffrement AES	2
<b>Fonctionnement du chiffrement AES :</b>	<b>4</b>
La SBox	5
Illustration du fonctionnement d'AES	6
Les modes de chiffrement	11
Le chiffrement par bloc et le chiffrement par flots	12
Réseau de Feistel	13
Utilisation Concrète de l'AES : Le Protocole SSH	14

# Présentation du chiffrement AES :

## Le chiffrement informatique

L'objectif du chiffrement dans le milieu de l'informatique est de sécuriser les données en les rendant illisibles pour toutes les personnes qui n'ont pas le droit de les déchiffrer. C'est un atout vraiment important dans notre ère où beaucoup de données personnelles cruciales sont stockées sur des dispositifs informatiques.

À l'inverse du hachage, qui est une autre technique de sécurisation des données, le chiffrement est bidirectionnel. Alors que l'objectif du hachage est de créer une empreinte numérique à partir d'opérations bien définies, le chiffrement quant à lui chiffre les données de sorte à ce qu'elles soient récupérées par la suite grâce à une clé.

Comme toutes les innovations en informatique et en électronique : le chiffrement doit son utilisation de la guerre.

## Le chiffrement dit "symétrique"

Le chiffrement symétrique (c'est le cas d'AES) est un type de chiffrement dans lequel il n'y a qu'une clé : la même est utilisée lors du chiffrement et du déchiffrement. À l'opposé du chiffrement asymétrique où on utilise une clé différente pour le chiffrement et déchiffrement : une clé privée et publique. La plus grosse préoccupation dans un chiffrement symétrique est la distribution de la clé de manière sécurisée.

## Le chiffrement AES

Le chiffrement AES (Advanced Encryption Standard) est l'algorithme de chiffrement symétrique. Ouvert au public, la NSA l'utilise pour chiffrer ses documents qui portent le sceau "secret défense."

L'histoire de l'AES a débuté en 1997 lorsque le NIST (National Institute of Standards and Technology) décide de trouver un successeur à un algorithme plus ancien, le DES (Data Encryption Standard) en lançant un appel d'offres (une compétition nationale). Le 15 juin 1998, date de la fin des candidatures, 21 projets ont été déposés. Certains sont l'œuvre d'entreprises comme IBM, d'autres regroupent des universitaires (CNRS,...), les derniers sont écrits par à peine quelques personnes.

Pendant deux ans, les algorithmes ont été évalués par des experts, avec forum de discussion sur Internet, et organisation de conférences. Le 2 octobre 2000, le NIST a donc choisi ce nouvel algorithme qui se nomme Rijndael, devenu AES, en l'honneur de ses créateurs, les chercheurs Belges Daemen et Rijmen. Il a été jugé le meilleur en termes de sécurité, performance, efficacité, simplicité et flexibilité. Sa résistance aux diverses attaques crypto analytiques connues à l'époque a été un facteur clé dans cette décision. L'AES est beaucoup plus sûr et flexible que son prédécesseur le DES (Data Encryption Standard) devenu obsolète.

La décision de rendre le processus de sélection public était une idée ingénieuse. Alors que jusqu'à présent la robustesse des algorithmes reposait sur le fait qu'ils soit en totalité confidentiel : ici le fait que tout le monde puisse le connaître était un très bon pari : la transparence du processus permettait à n'importe qui dans le monde de tester l'algorithme pour vérifier qu'il n'existait aucun moyen de contourner la sécurité, cela augmentait donc la couverture de test. Cela augmentait également la confiance de son utilisation, en exposant son fonctionnement à tout le monde.

Cet algorithme est officiellement devenu la norme de chiffrement AES après sa victoire sur ses concurrents lors d'une compétition internationale organisée en 2001. Il a été reconnu pour sa sécurité, sa rapidité de traitement, ses faibles besoins en ressources et mémoire, ainsi que sa flexibilité d'implémentation tant en logiciel qu'en matériel. Il est particulièrement adapté pour des implémentations embarquées soumises à des contraintes strictes en termes de ressources, de puissance de calcul, de taille mémoire, etc.

Le chiffrement AES est maintenant largement utilisé dans de nombreuses applications et industries, et pas seulement par la NSA, de part sa robustesse et de sa fiabilité. Depuis son adoption, l'AES est devenu un standard mondial pour le chiffrement de données, utilisé dans des protocoles de sécurité comme SSL/TLS pour la sécurisation des communications Internet.

```
1  """
2  Exemple de chiffrement avec la librairie Cryptography
3  Source : https://pypi.org/project/cryptography/
4  """
5
6  from cryptography.fernet import Fernet
7
8  key = Fernet.generate_key() # génération d'une clé
9  # (on voit que c'est un chiffrement symétrique puisque on ne génère qu'une clé.)
10 f = Fernet(key)
11
12 token = f.encrypt(b"Ceci est le message chiffré et déchiffré") # message à crypter (b pour convertir en bit)
13 token
14
15 f.decrypt(token) # message déchiffré
16
17 b'Ceci est le message chiffré et déchiffré'
```

# Fonctionnement du chiffrement AES :

Le chiffrement AES, basé sur une structure de réseau de substitution-permutation, intègre l'utilisation de polynômes pour mener à bien certaines de ses fonctions clés. Parmi celles-ci, l'opération de mélange de colonnes, connue sous le nom de MixColumns, ainsi que parfois dans le processus d'expansion de clé, mettent en œuvre ces polynômes. Ces derniers sont définis dans un corps fini, fréquemment désigné par  $GF(2^8)$ , où "GF" signifie "Galois Field". Ce champ de Galois est une structure mathématique composée de termes dont les coefficients sont limités à l'ensemble fini de  $2^8$  éléments.

Cette spécificité du polynôme dans AES est fondamentale pour la sécurité cryptographique. En effet, elle joue un rôle essentiel dans la diffusion et la confusion, deux principes de base qui renforcent la résilience du chiffrement. La diffusion se manifeste par une dispersion efficace des caractéristiques du texte clair à travers l'ensemble du texte chiffré, tandis que la confusion vise à brouiller la relation entre la clé de chiffrement et le texte chiffré.

L'efficacité de l'AES repose également sur sa capacité à transformer systématiquement les données d'entrée à travers une série d'étapes bien définies. Ces étapes comprennent la substitution des octets, où chaque octet est remplacé par un autre selon une table de substitution prédéfinie, et le décalage des rangées, qui permute les octets dans les rangées de la grille de l'AES. Le mélange des colonnes (MixColumns) intervient ensuite, mélangeant de manière complexe les données au sein de chaque colonne, suivi de l'ajout de la clé de tour (AddRoundKey), où les données sont combinées avec une sous-clé générée à partir de la clé de chiffrement principale.

L'intégration de ces polynômes dans le cadre du champ de Galois  $GF(2^8)$  est cruciale pour le processus de chiffrement AES. Elle assure non seulement la robustesse de l'algorithme face aux tentatives de déchiffrement non autorisées, mais contribue également à sa flexibilité et son efficacité, le rendant adapté à une large gamme d'applications, de la protection de données sensibles à la sécurisation des communications électroniques.

## La SBox

L'Advanced Encryption Standard (AES), reconnu pour sa robustesse et sa sécurité, intègre un composant essentiel appelé la S-box ou boîte de substitution, qui joue un rôle crucial dans le processus de chiffrement. L'AES fonctionne en transformant les données initiales, organisées en une grille de 4x4 octets, soit un bloc de 128 bits, à travers plusieurs étapes de chiffrement.

Au cœur de ces étapes se trouve la S-box, qui remplace chaque octet du bloc par un autre, suivant une table de substitution préétablie. Cette table n'est pas choisie au hasard, mais est le résultat d'une conception méticuleuse visant à assurer une complexité maximale et une non-linéarité dans le processus de chiffrement. L'objectif est d'éviter toute forme de substitution prévisible, éliminant ainsi les points fixes où un octet pourrait être substitué par lui-même, et rendant impossible de déduire directement l'octet de sortie à partir de l'octet d'entrée.

La substitution par la S-box s'intègre dans ce qu'on appelle le réseau SP (Substitution-Permutation) d'AES. Après la substitution des octets, d'autres transformations, telles que le décalage des rangées et le mélange des colonnes, sont appliquées. Ces étapes additionnelles, couplées à la substitution opérée par la S-box, dispersent les modifications apportées à un octet à travers tout le bloc, amplifiant ainsi l'effet de la transformation initiale et renforçant la sécurité du chiffrement. Ce processus se répète sur plusieurs tours - 10, 12 ou 14 tours, selon la taille de la clé choisie (128, 192 ou 256 bits). À chaque tour, une clé différente, dérivée de la clé principale à travers un calendrier de clés, est intégrée au bloc de données à l'aide d'une opération XOR, suivi des étapes de substitution, de permutation et de mélange.

Les opérations effectuées par la S-box, ainsi que les autres transformations au sein de l'AES, reposent sur la théorie des champs de Galois, un cadre mathématique qui garantit que toutes les opérations se déroulent dans un ensemble fini d'éléments. En l'occurrence, pour l'AES, cet ensemble correspond à des octets, chaque octet étant un élément du champ de Galois de  $2^8$  éléments. Cette approche assure que les transformations restent cohérentes et ne débordent pas hors des limites fixées, préservant ainsi la structure et la taille du bloc de données tout au long du processus de chiffrement.

En somme, la S-box d'AES, grâce à sa conception complexe et son intégration dans le réseau SP, joue un rôle déterminant dans l'efficacité de l'algorithme. Elle permet de transformer les données de manière non linéaire et hautement sécurisée, contribuant ainsi à la robustesse d'AES face aux diverses attaques cryptographiques, y compris les attaques par force brute et les attaques cryptanalytiques plus

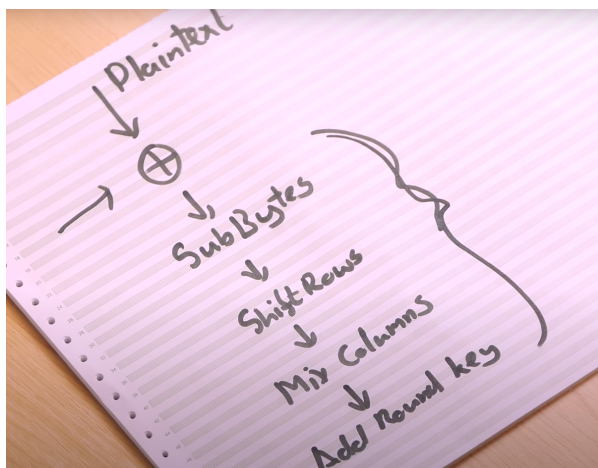
sophistiquées. C'est cette combinaison d'efficacité, de sécurité et de flexibilité qui a fait d'AES la norme de choix dans le domaine du chiffrement symétrique.

Voici une représentation de la S-box :

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

## Illustration du fonctionnement d'AES

Ici nous avons notre texte qui va subir plusieurs modifications, en passant par plusieurs étapes (SubBytes, ShiftRows, ...) :



Prenons ici une table de 4x4 de 128 bytes afin de faire la première étape

le SubBytes qui consiste à faire des substitutions grâce à la S-Box :

Round 1

19	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	e9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

**S-BOX** byte substitution table

Nous allons ensuite prendre la valeur correspondante dans la S-Box afin de crypter ce message :

Round 1

19
----

	a0	9a	e9
3d	f4	c6	f8
e3	e2	8d	48
be	2b	2a	08

hex	y															
	0	1	2	3	4	5	6	7		b	c	d	e	f		
0	63	7c	77	7b	f2	6b	6f	c5		2b	fe	d7	ab	76		
1	ca	82	e9	7d	fa	59	47	f0		af	9c	a4	72	c0		
2	b7	fd	93	26	36	3f	f7	cc		f1	71	d8	31	15		
3	04	c7	23	c3	18	96	05	9a		e2	eb	27	b2	75		
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3		
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c		
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c		
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff		
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d		
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e		
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95		
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a		
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd		
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1		
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55		
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54		

**S-BOX** byte substitution table



Puis nous allons faire pareil pour toutes les autres valeurs :

Round 1

d4	e0	b8	1e
27	bf	b4	41
11	98	5d	52
ae	f1	e5	30

hex		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

**S-BOX** byte substitution table

Une fois cette étape terminée nous devons passer à l'étape ShiftRows :



Cette étape consiste à déplacer la valeur de 1 rang pour la 2ème ligne, 2 rangs pour la 3ème et 4 rangs pour la dernière.

On se retrouve donc avec le résultat suivant :

d4	e0	b8	1e
bf	b4	41	27
5d	52	11	98
30	ae	f1	e5

..... rotate over 3 bytes

Désormais il nous reste 2 étapes à faire passons à l'étape MixColumns :

Round 1

e0	b8	1e
b4	41	27
52	11	98
ae	f1	e5

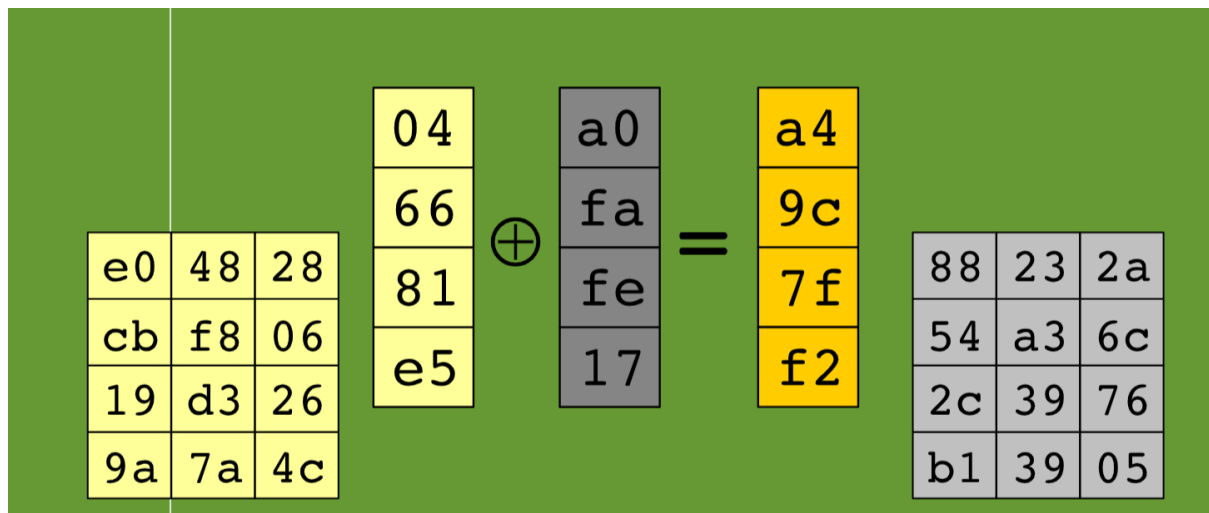
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \cdot \begin{bmatrix} d4 \\ bf \\ 5d \\ 30 \end{bmatrix} = \begin{bmatrix} 04 \\ 66 \\ 81 \\ e5 \end{bmatrix}$$

The four numbers of one column are modulo multiplied in Rijndael's Galois Field by a given matrix.

Une fois appliqué pour toutes les lignes on obtiendra le résultat suivant :

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

Il nous reste une dernière étape, Add round Key :



Une fois cette opération effectuée pour chaque lignes on obtient :

a4	68	6b	02
9c	9f	5b	6a
7f	35	ea	50
f2	2b	43	49

On applique cette transformation 9 fois encore car on a un message de 128 bytes et le dernier tour on n'effectue pas l'étape du mixColumns.

Ce qui nous donne le message crypté suivant :

39	02	dc	19
25	dc	11	6a
84	09	85	0b
1d	fb	97	32

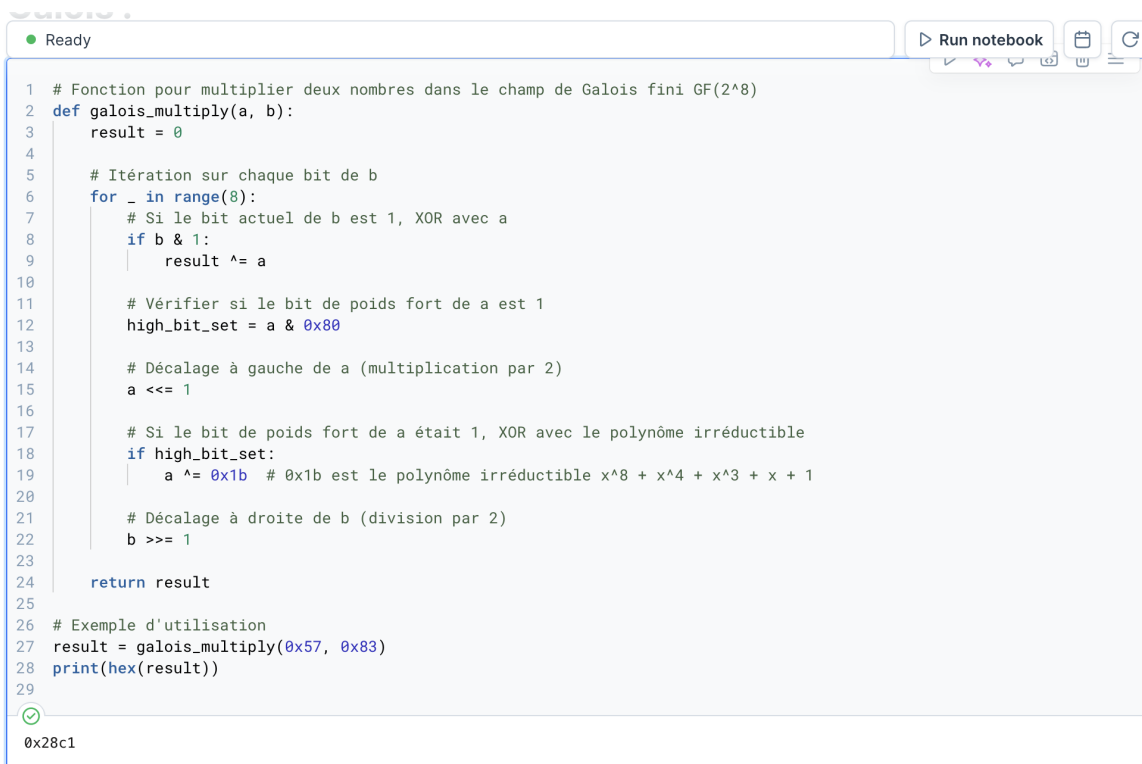
**Ciphertext**

## Les modes de chiffrement

Un mode de chiffrement est une méthode de chiffrement spécifique dans le cas du chiffrement symétrique. Cela fonctionne uniquement pour un chiffrement par bloc (voir 'Chiffrement par flots et par bloc'). Il en existe des multitudes, et ils ont tous leur utilisation spécifique.

- ECB (Electronic Codebook): Dans ce mode de chiffrement, chaque bloc est chiffré indépendamment.
- CBC (Cipher Block Chaining): Dans celui-ci, chaque bloc contient des traces de chiffrement du bloc précédent. Ce principe est par exemple utilisé dans le cas de la blockchain.
- Cipher Feedback (CFB), Output Feedback (OFB), Counter (CTR)....

Le plus utilisé pour AES est GCM (Galois/Counter Mode). Il est très utilisé dans le contexte d'une utilisation où l'on a besoin d'un chiffrement et d'une authentification. Il combine un chiffrement de compteur avec une authentification de message basée sur Galois.



```
1 # Fonction pour multiplier deux nombres dans le champ de Galois fini GF(2^8)
2 def galois_multiply(a, b):
3     result = 0
4
5     # Itération sur chaque bit de b
6     for _ in range(8):
7         # Si le bit actuel de b est 1, XOR avec a
8         if b & 1:
9             result ^= a
10
11         # Vérifier si le bit de poids fort de a est 1
12         high_bit_set = a & 0x80
13
14         # Décalage à gauche de a (multiplication par 2)
15         a <<= 1
16
17         # Si le bit de poids fort de a était 1, XOR avec le polynôme irréductible
18         if high_bit_set:
19             a ^= 0x1b # 0x1b est le polynôme irréductible x^8 + x^4 + x^3 + x + 1
20
21         # Décalage à droite de b (division par 2)
22         b >>= 1
23
24     return result
25
26 # Exemple d'utilisation
27 result = galois_multiply(0x57, 0x83)
28 print(hex(result))
29
0x28c1
```

Le Théorème de Lagrange et le Théorème Fondamental de l'Arithmétique sont une des notions mathématiques utilisées pour le fonctionnement de cet algo.

## Le chiffrement par bloc et le chiffrement par flots

Il existe deux méthodes de chiffrement : celle par blocs et celle par flots.

Dans le chiffrement par bloc, le processus consiste à découper le message en blocs de même taille (128 bits). Chaque bloc est ensuite traité indépendamment par l'algorithme. La manipulation sur les blocs est décrite par les modes de chiffrement vus plus haut.

Le chiffrement par flots quant à lui, contrairement à celui par bloc, traite le message en continu, bit par bit. Nous ne nous attarderons pas sur ce type de chiffrement. Voici en l'occurrence un exemple de code de chiffrement par flots :

```
"""
Exemple d'un chiffrement par flots
"""
def chiffrementFlot(message, cle):
    messageChiffre = ""
    for i in range(len(message)):
        messageChiffre += chr(ord(message[i]) ^ ord(cle[i % len(cle)]))
    return messageChiffre

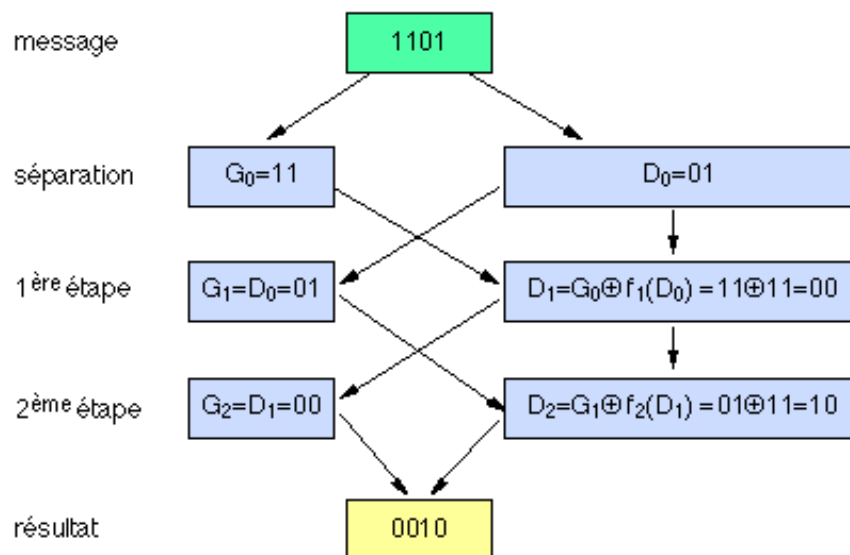
message_original = "Bonjour"
cle_secrete = "clesecrete"

messageChiffreFlot = chiffrementFlot(message_original, cle_secrete)
print("Message chiffré par flot:", messageChiffreFlot)
```

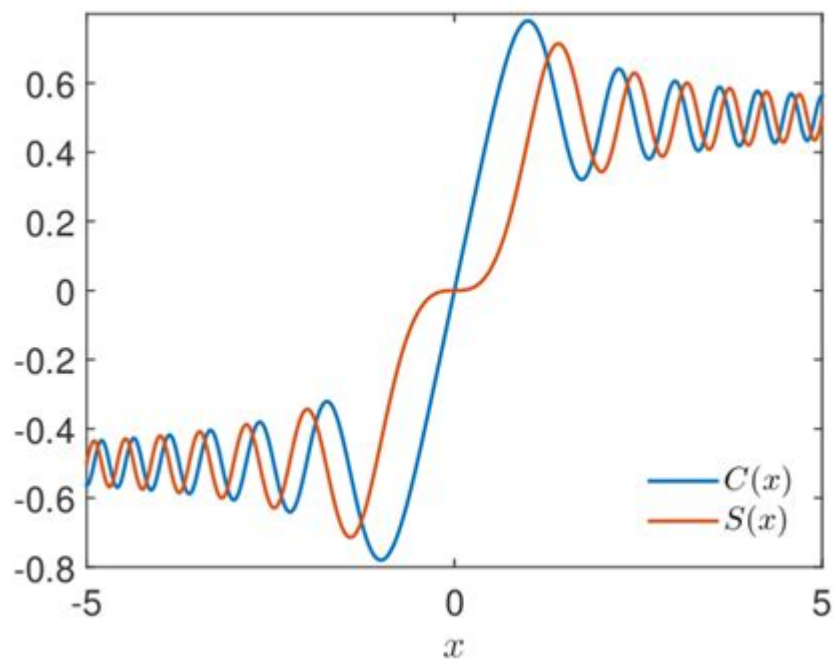
Avec AES, celui le plus utilisé est par blocs, mais il ne faut pas négliger celui par flots car il arrive qu'il soit utilisé dans certains cas nécessitant ce type de chiffrement.

## Réseau de Feistel

Le réseau de Feistel est une construction, une méthode de fonctionnement utilisée dans certains algos de chiffrement symétrique. Il était notamment utilisé avec DES, le prédécesseur d'AES.



Son fonctionnement est le suivant : il divise le bloc à chiffrer en deux moitiées, et effectue des opérations arithmétiques dessus, de manière itérative. Ces opérations suivent la fonction non-linéaire de Feistel.

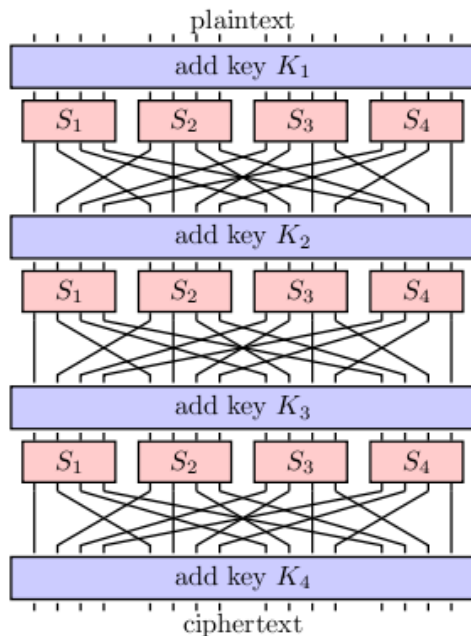


Ce sont des opérations de substitution et de permutation. Les deux moitiées sont alors échangées, et le processus est répété.

Mais AES n'utilise plus le réseau de Feistel : la clé utilisée était trop courte (56 bits) donc présentait des failles de sécurité. À la place, AES utilise une structure appelée

Substitution-Permutation Network.

Il est plus efficace, plus sécurisé, et permet l'utilisation de clés de 128, 192 ou 256 bits.



## Utilisation Concrète de l'AES : Le Protocole SSH

Dans le cadre du protocole Secure Shell (SSH), l'AES est utilisé pour crypter les données échangées, ce qui assure un certain niveau de confidentialité et d'intégrité des informations. Le processus d'établissement d'une connexion SSH commence par une négociation entre le client et le serveur pour déterminer les paramètres de sécurité, parmi lesquels le choix de l'algorithme de cryptage est crucial. AES est fréquemment choisi pour cette tâche du fait de sa robustesse. Avant le cryptage, une étape cruciale est l'établissement d'une clé secrète partagée, généralement obtenue via un échange de clés Diffie-Hellman. Cette méthode renforce la sécurité en évitant l'échange explicite de la clé sur le réseau.

SSH implémente divers modes de fonctionnement pour AES, tels que le Cipher Block Chaining (CBC) ou le Galois/Counter Mode (GCM) dont on a parlé précédemment. À la fin d'une session SSH, il est courant de détruire la clé secrète, renforçant ainsi la protection contre les tentatives d'interception et de déchiffrement des communications.

L'intégration de l'AES dans le protocole SSH démontre comment un algorithme de chiffrement peut être intégré de manière transparente et efficace dans un protocole de communication, ce qui assure la confidentialité, l'intégrité et la sécurité des données.



Bibliographie :

<https://csrc.nist.gov/files/pubs/fips/197/final/docs/fips-197.pdf>

[https://fr.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://fr.wikipedia.org/wiki/Advanced_Encryption_Standard)

<https://www.securiteinfo.com/cryptographie/aes.shtml>

<https://www.hds.utc.fr/~wschon/sr06/txPHP/aes/AesAlgo/AesAlgo.php>

[https://www.youtube.com/watch?v=O4xNJsitN6E&ab\\_channel=Computerphile](https://www.youtube.com/watch?v=O4xNJsitN6E&ab_channel=Computerphile)

<https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-informati-on-th9/cryptographie-authentification-protocoles-de-securite-vpn-42314210/le-protocole-ssh-h5235/>

Source schéma Feistel :

<https://www.apprendre-en-ligne.net/crypto/blocs/feistel.html>

Source fonction de Feistel :

[https://www.researchgate.net/figure/Fonctions-de-Fresnel-Cx-et-S-x\\_fig4\\_345788620](https://www.researchgate.net/figure/Fonctions-de-Fresnel-Cx-et-S-x_fig4_345788620)

Source substi-permut :

[https://www.barrywatson.se/crypto/crypto\\_sp\\_network.html](https://www.barrywatson.se/crypto/crypto_sp_network.html)

Lien vers notre repo GitHub :

<https://github.com/Munozmu/chiffrement-aes-mmh>