La cryptographie

2.1 Introduction

La sécurité informatique est devenue une préoccupation majeure pour tous ceux qui sont intéressés par l'informatique et la plupart des développeurs se concentrent sur les techniques de cryptage pour fournir de bons résultats. En effet, la cryptographie, ou l'art de chiffrer est devenue aujourd'hui une science à part entière. Au croisement des mathématiques, de l'informatique, et parfois de la physique, on l'utilise lorsqu'il y a un échange sensible de données.

Dans ce chapitre nous allons décrire le concept de la cryptographie et ses deux types à savoir la cryptographie symétrique et asymétrique ainsi que les algorithmes de chiffrement les plus utilisés.

2.2 Définition de la cryptographie

Le terme cryptographie vient en effet de deux mots grecs : Kruptus qu'on peut traduire comme secret et Graphein pour écriture. La cryptographie est l'art de cacher l'information pour qu'elle soit incompréhensible, elle désigne l'ensemble des techniques qui permettent de chiffrer les messages, son objectif principale est de permettre à deux personnes **Alice** et **Bob** de communiquer à travers un canal peu sécurisé de telle sorte qu'un opposant **Eve** ne puisse pas comprendre ce qui est échangé, on utilise une clé appelée clé de chiffrement pour le processus de chiffrement. Pour rendre l'information à nouveau compréhensible on utilise une clé appelée clé de déchiffrement pour le processus de déchiffrement. [21]

2.3 Terminologie [21] [22]

Les principaux termes utilisés dans la cryptographie sont :

Cryptologie : C'est une science mathématique regroupant la cryptographie et la cryptanalyse.

Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.

Cryptanalyse : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.

Cryptosystème : un Cryptosystème est constitué d'un algorithme cryptographique ainsi que toutes les clés possibles et tous les protocoles qui le font fonctionner.

Cryptogramme : **Texte chiffré** : **Ciphertext** : est le résultat de l'application d'un chiffrement d'un texte clair.

Texte clair : Plaintext : le message à chiffrer.

Chiffrement: la fonction permettant de transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a crée le message et ainsi que le destinataire.

Déchiffrement : la fonction permettant de retrouver le texte clair à partir du texte chiffré.

Clé : une clé est un paramètre utilisé en entrée d'une opération cryptographique (chiffrement, déchiffrement).On distingue généralement deux types de clefs :

Clés symétriques : il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou à clé secrète.

Clés asymétriques : il s'agit de clés utilisées dans le cas du chiffrement asymétrique ou à clé publique. Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

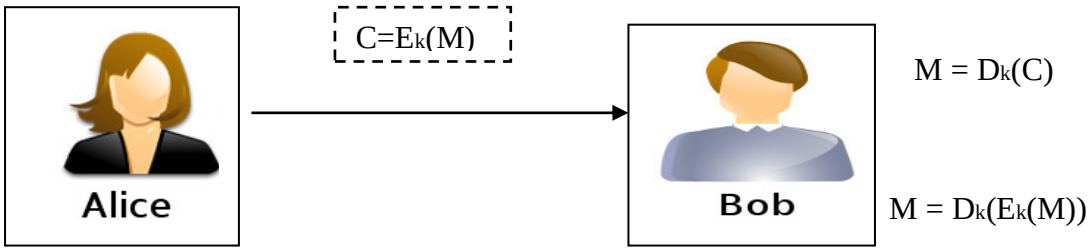

Figure 2.1 : Une représentation d'un Cryptosystème.

- M représente le texte clair.
- C est le texte chiffré tel que $C=E_k(M)$.
- K est la clé de chiffrement et déchiffrement (dans le chiffrement symétrique).
- E(x) est la fonction de chiffrement.
- D(x) est la fonction de déchiffrement.

2.4 Objectif de la cryptographie

Les principaux objectifs garantis par l'application de la cryptographie sont [23]:

- **La confidentialité**:

  Le message chiffré ne doit pas être compréhensible que par les destinataires légitimes. Il ne peut pas être déchiffré par un intrus.

- **L'intégrité :**

  Le destinataire peut vérifier le message reçu qui n'a pas été modifié en chemin par l'utilisation de mécanisme de la signature électronique.

- **L'authentification**:

  Le destinataire d'un message doit pouvoir s'assurer de son origine. Un intrus ne doit pas être capable de se faire passer pour quelqu'un d'autre.

- **La non répudiation**:

  Un expéditeur ne peut pas nier d'avoir envoyé un message et le destinataire ne peut pas nier de l'avoir reçu.

2.5 Classification de cryptographie

Dans la cryptographie moderne toute la sécurité est basée sur la clé et non dans les détails des algorithmes utilisés. On trouve principalement deux grandes familles de cryptographie moderne : la cryptographie symétrique ou à clé secrète et la cryptographie asymétrique ou à clé publique.

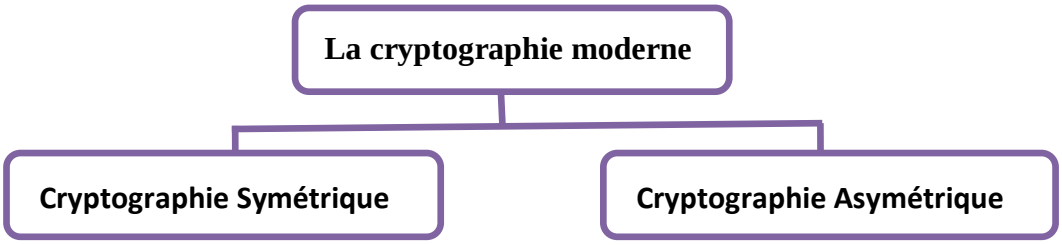

Figure 2.2: Les méthodes de la cryptographie moderne.

2.5.1 Cryptographie symétrique

Cryptographie symétrique utilise une même clé secrète pour chiffrer et déchiffrer des données dont elle assure la confidentialité. Les algorithmes symétriques sont très rapides en termes de calcul, cependant ils posent le problème de distribution de clés entre un émetteur et un récepteur. Le partage d'une clé avec chaque entité communicante dans un groupe de n entités est difficile et conduit à un grand nombre de clés à gérer. [24]

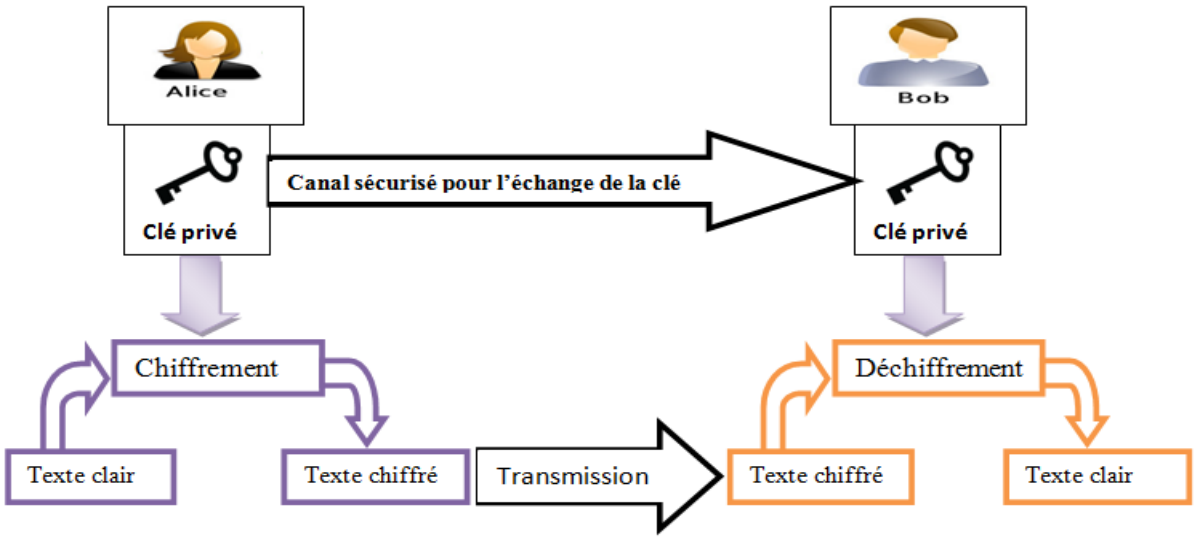

Figure 2.3 : Principe d'un chiffrement à clé secrète (symétrique).

2.5.1.1 Les catégories de la cryptographie symétrique

Les schémas de chiffrement symétrique peuvent être classés en deux catégories, le chiffrement par flux et le chiffrement par bloc.

2.5.1.1.1 Chiffrement par flux : [22] [23]

Le chiffrement par flux est aussi appelé chiffrement en continu. L'opération de chiffrement par flux s'opère sur chaque élément du texte clair (caractère, bits) au fur et à mesure de leurs arrivées. Sa structure repose sur un générateur de nombres pseudo-aléatoires dont la sortie est couplée via un XOR avec l'information à chiffrer. Ces nombres pseudo-aléatoires produits à partir d'une clé secrète.

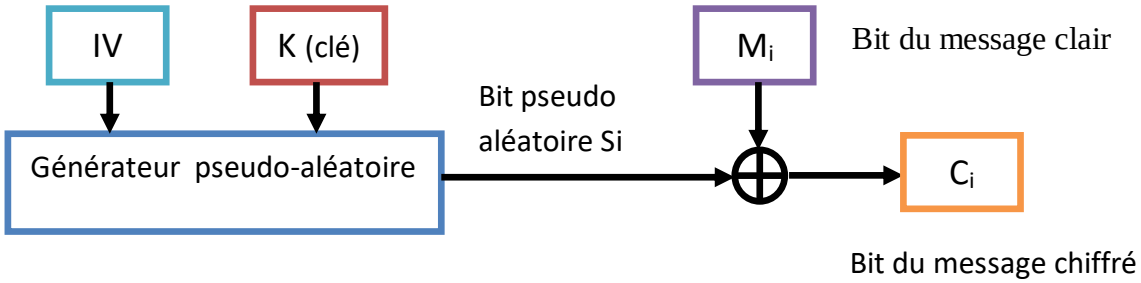

Figure 2.4 : Schéma de chiffrement par flux

Exemples d'algorithmes de chiffrement par flux :

- ✓ **A5 :** est l'algorithme à flux utilisé pour assurer la confidentialité des communications hertziennes dans la norme GSM.
- ✓ **RC4 :** C'est un algorithme de chiffrement à flot conçu en 1987 par Ron Rivest. Il est dédié aux applications logicielles, et largement déployé dans le protocole SSL/TLS et la norme WEP pour les réseaux sans fil.
- ✓ **E0 :** le chiffrement Bluetooth c'est l'algorithme de chiffrement à flot utilisé pour protéger la confidentialité des communications dans le protocole de transmission sans fil de courte portée Bluetooth.

2.5.1.1.2 Chiffrement par bloc

Dans le chiffrement par bloc Chaque message est d'abord partitionné en blocs de tailles fixes, les fonctions s'appliquent alors sur chaque bloc.

- ➢ **Le mode ECB (Electronic Code Book): [25]**

  Le mode le plus simple est le mode ECB, dans lequel le texte en clair traite chaque bloc indépendamment des autres blocs et chacun d'eux est chiffré à l'aide de la même clé. Pour une clé donnée, il existe un texte chiffré unique pour chaque bloc de texte en clair $\mathbf{C_i=E_k(M_i)}$.

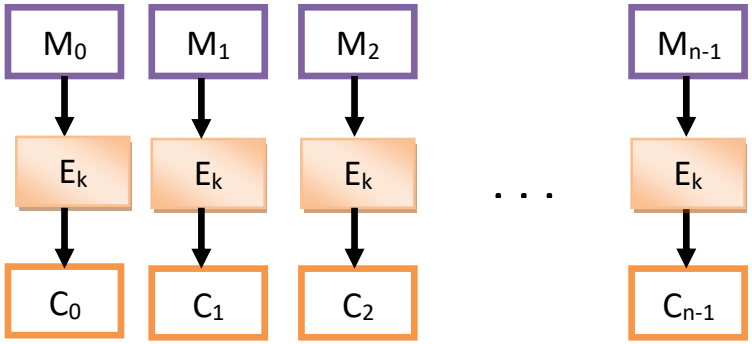

Figure 2.5 : chiffrement en mode ECB.

Le déchiffrement : Chaque bloc chiffré est déchiffré indépendamment pour donner un bloc en clair correspondant $\mathbf{M_i=D_k(C_i)}$.

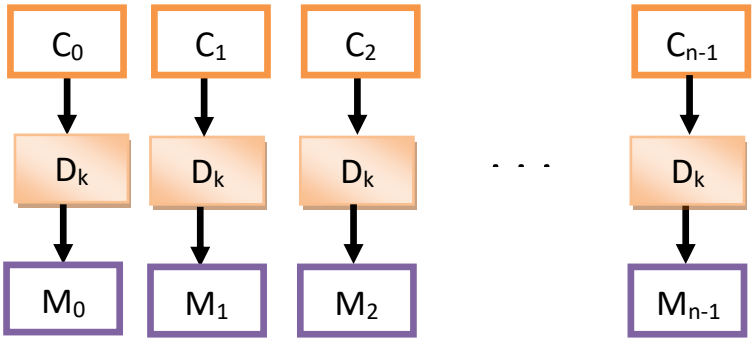

Figure 2.6 : Déchiffrement en mode ECB.

La caractéristique la plus importante de l'ECB est que si le même bloc de texte brut apparaît plusieurs fois dans le message, il produit toujours le même texte chiffré. Pour les messages longs, le mode ECB peut ne pas être sécurisé. Si le message est hautement structuré, il peut être possible pour un cryptanalyste d'exploiter ses régularités.

- **Le mode CBC (Cipher-Block Chaining) :**

  Pour surmonter les lacunes de sécurité de l'EBC, nous utilisons le mode CBC, même si le bloc du texte clair est répété, il produit différents blocs de textes chiffrés.

  Dans ce schéma, l'entrée de l'algorithme de chiffrement est le XOR du bloc de texte en clair actuel et du bloc de texte chiffré précédent; la même clé est utilisée pour chaque bloc.

  Pour produire le premier bloc de texte chiffré, un vecteur d'initialisation (IV) est XOR avec le premier bloc de texte en clair on a:

  $$C_0 = E_k([M_0 \oplus IV])$$

  Pour produire les prochains blocs de texte chiffré on a:

  $$C_i = E_k([Mi \oplus C_{i-1}])$$

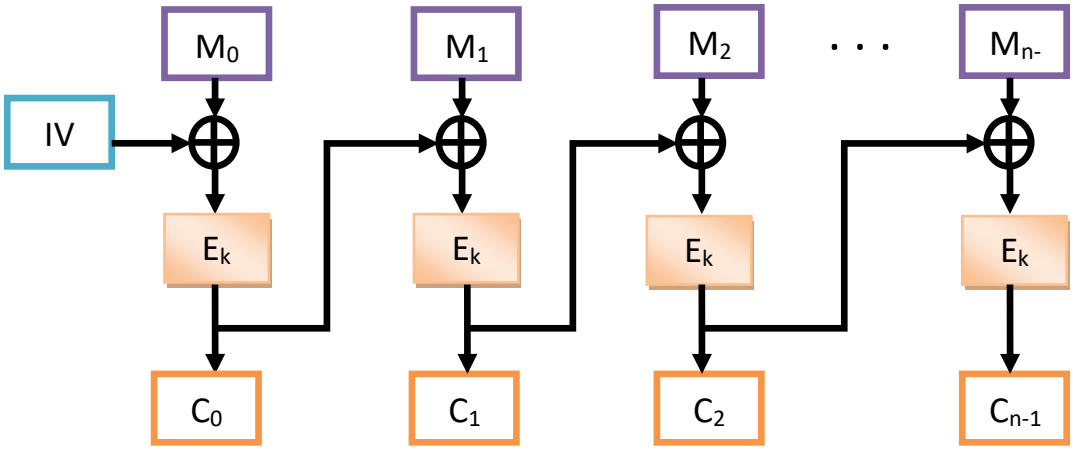

Figure 2.7 : chiffrement en mode CBC.

Au déchiffrement, l'IV est XOR avec la sortie de l'algorithme de déchiffrement D_k pour récupérer le premier bloc de texte en clair :

$$M_0 = D_k(C_0) \oplus IV$$

Pour produire les prochains blocs de texte clair M_i :

$$M_i = D_k(C_i) \oplus C_{i-1}$$

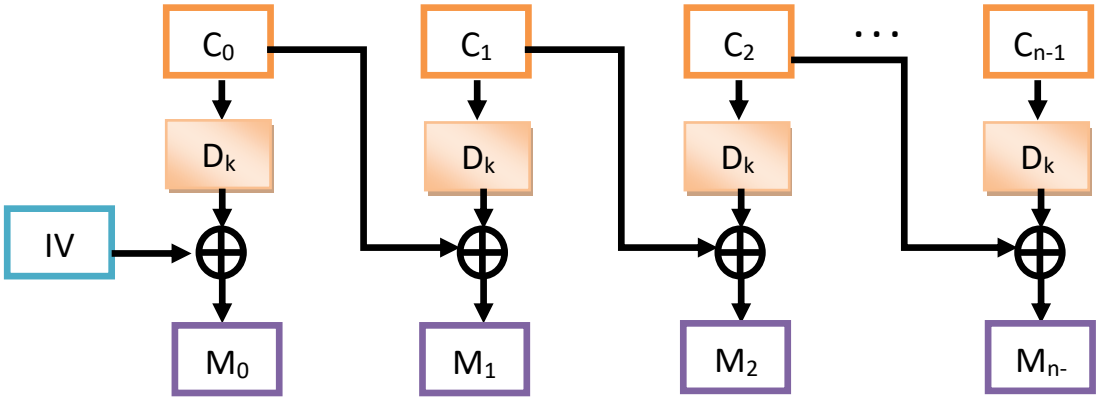

Figure 2.8 : Déchiffrement en mode CBC.

- **Le mode CFB (Cipher Feedback mode):**

L'entrée de la fonction de cryptage est un vecteur d'initialisation (IV). La sortie de la fonction de cryptage est XOR avec le premier segment de texte en clair M_0 pour produire le premier bloc de texte chiffré C_0. Ce processus se poursuit jusqu'à ce que toutes les unités de texte en clair aient été chiffrées.

Pour produire le premier bloc de texte chiffré C_0, un vecteur d'initialisation (IV) est chiffré par la fonction de chiffrement E_k, le résultat est XOR avec le premier bloc de texte en clair M_0 on a :

$$C_0 = E_k(IV) \oplus M_0$$

Pour produire les prochains blocs de texte chiffré on a:

$$C_i = E_k(C_{i-1}) \oplus Mi$$

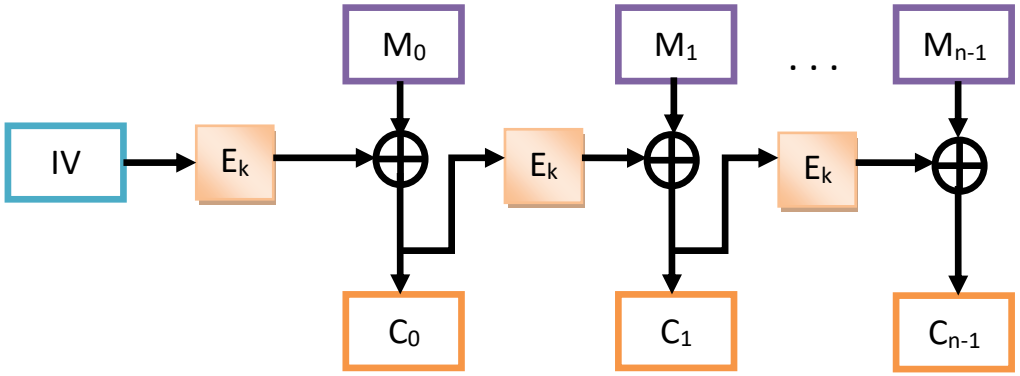

Figure 2.9 : Chiffrement en mode CFB.

Pour le déchiffrement, le même schéma est utilisé, sauf l'unité de texte chiffré reçue est XOR avec la sortie de la fonction de chiffrement de chiffré de bloc précédent pour produire l'unité de texte en clair. Notez que c'est la fonction de cryptage E_K qui est utilisée, pas la fonction de décryptage on a:

$$M_0 = E_k(IV) \oplus C_0$$

$$M_i = E_k(C_{i-1}) \oplus C_i$$

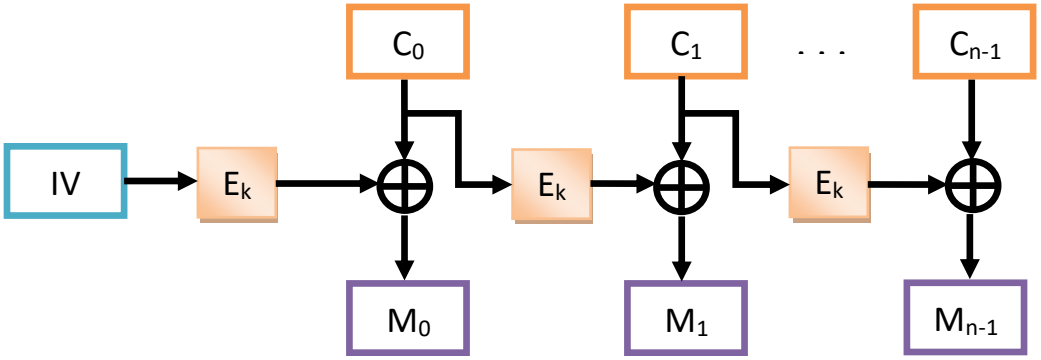

Figure 2.10 : Déchiffrement en mode CFB.

- **Le mode OFB (Output Feedback mode) :**

  Comme avec CBC et CFB, le mode OFB nécessite un vecteur d'initialisation. Dans le cas de l'OFB, l'IV doit être unique à chaque exécution de l'opération de cryptage, car la séquence de blocs de sortie de cryptage dépend uniquement de la clé et de l'IV et ne dépend pas du texte en clair. Par conséquent, pour une clé et un IV donnés, le bloc de sortie utilisé pour XOR avec le bloc en clair pour avoir le bloc de texte chiffré on a :

$$C_i = M_i \oplus E_k(R_{i-1}) \text{ tel que } R_0 = IV$$

$$M_i = C_i \oplus E_k(R_{i-1}) \text{ tel que } R_0 = IV$$

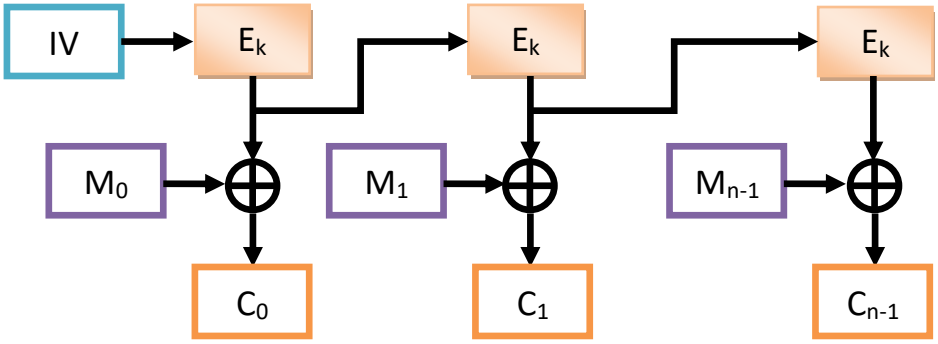

Figure 2.11 : Mode OFB.

2.5.1.2 Les algorithmes les plus connus dans la cryptographie symétrique

2.5.1.2.1 DES

Jusqu'à l'introduction de la norme de chiffrement avancé (AES) en 2001, la norme de chiffrement des données (DES) était le schéma de chiffrement le plus utilisé. Il a été conçu dans les années 1970 par la firme américaine IBM pour être un système de chiffrement par blocs à clé secrète, et fut adopté comme standard de chiffrement en 1977.

Comme illustre la figure 2.12, DES commence par une permutation IP de bits initiale à l'aide de la matrice de permutation illustrée à la figure 2-13 (b), effectue le chiffrement de Feistel à l'aide de sous-clés générées par un programme de génération de clés, et enfin exécute la permutation initiale inverse IP^{-1} à l'aide d'une matrice de permutation.

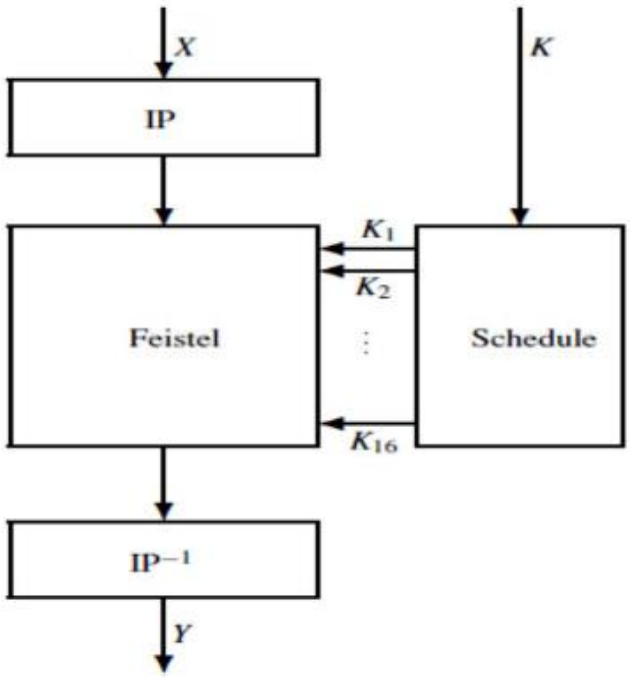

Figure 2.12 : Schéma de cryptage DES.

8	7	6	5	4	3	2	1
16	15	14	13	12	11	10	9
24	23	22	21	20	19	18	17
32	31	30	29	28	27	26	25
40	39	38	37	36	35	34	33
48	47	46	45	44	43	42	41
56	55	54	53	52	51	50	49
64	63	62	61	60	59	58	57

(a) La matrice originale

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) la matrice IP

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	27	25

(c) la matrice IP^{-1}

Figure 2.13 : matrices de permutation initiale et de permutation initiale inverse.

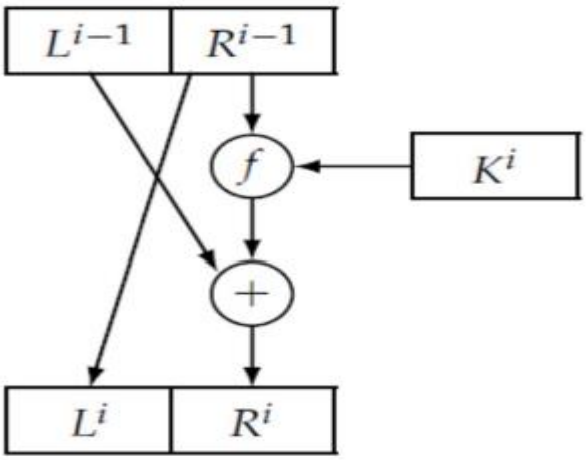

Figure 2.14 : Un tour de chiffrement DES.

DES traite des messages de 64 bits avec des clés de 56 bits, et il fournit un cryptogramme de 64 bits en sortie. Il consiste alors à réitérer la structure Feistel 16 tours. [25]

2.5.1.2.2 AES

AES est le chiffre le plus utilisé dans l'univers, ce chiffre symétrique est normalisé par le NIST en 2000 en remplacement du DES, de nos jours, les processeurs presque modernes intègrent des instructions AES telles que les processeurs Intel, les processeurs AMD, plutôt que de nombreux autres produits logiciels tels que OpenSSL.[26]

contrairement à la majorité des algorithmes de chiffrement asymétriques dont la sécurité repose sur des problèmes mathématiques difficiles tels que le logarithme discret dans le cas d'ECC, ou la factorisation entière dans le cas de RSA, AES tire sa force de la combinaison entre permutation et substitution, plus communément connu sous le nom de réseaux de substitution-permutation (SPN), nous pouvons dire que l'AES est lui-même un problème difficile, car de nombreuses personnes qualifiées ont tenté de briser le cryptage AES et ont échoué. [27]

AES est un chiffrement par bloc symétrique prend une taille de bloc de texte en clair de 128 bits ou 16 octets. La longueur de clé peut être de 128, 192 ou 256 bits. [25]

La description détaillé de schéma AES sera présenté dans le chapitre suivant (figure 3.1).

2.5.1.3 Avantages et inconvénients de la cryptographie symétrique :

Avantages	Inconvénients
Chiffrement / Déchiffrement rapide	Problème d'échange de la clé secrète
Volumes importants de données à chiffrer	n (n-1)/2 Clés pour n partenaires
Clés relativement courtes	Pas de signature électronique
Utilise peu de ressources systèmes	

Table 2.1 : Avantages et Inconvénients de chiffrement symétrique.

2.5.2 Cryptographie Asymétrique

La cryptographie asymétrique, ou cryptographie à clé publique repose sur l'utilisation d'une clé publique (qui est diffusée) et d'une clé privée (gardée secrète), l'une permettant de chiffrer le message et l'autre de le déchiffrer. Ainsi, l'expéditeur peut utiliser la clé publique du destinataire pour chiffrer un message que seul le destinataire (en possession de la clé privée) peut le déchiffrer, garantissant la confidentialité du contenu.

Inversement, l'expéditeur peut utiliser sa propre clé privée pour chiffrer un message, le destinataire peut déchiffrer avec la clé publique ; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message. Dans la cryptographie asymétrique impossible de trouver la clé privée à partir de la clé publique. [21]

2.5.2.1 Principe de fonctionnement

Alice souhaite envoyer des données chiffrées à Bob, ils procéderont ainsi :

1. Bob crée une paire de clés asymétriques : clé privée qu'il conserve précieusement, et une clé publique qu'il diffuse notamment à Alice.
2. Alice chiffre son message avec la clé publique de Bob.
3. Alice envoie le message chiffré à Bob.
4. Bob reçoit le message chiffré d'Alice.
5. Enfin Bob déchiffre le message avec sa propre clé privée.

2.5.2.2 Caractéristiques de la cryptographie asymétrique

- $\mathbf{D_{SK}\ (E_{PK}(M)) = M}$
- PK la clé publique.
- SK la clé privée secrète.

- La connaissance de PK ne permet pas de déduire SK.
- La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire.
- Les algorithmes de chiffrement Asymétrique se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le problème des logarithmes discrets (El Gamal, courbe elliptique), ou encore le problème du sac à dos (Merkle-Hellman). [2]

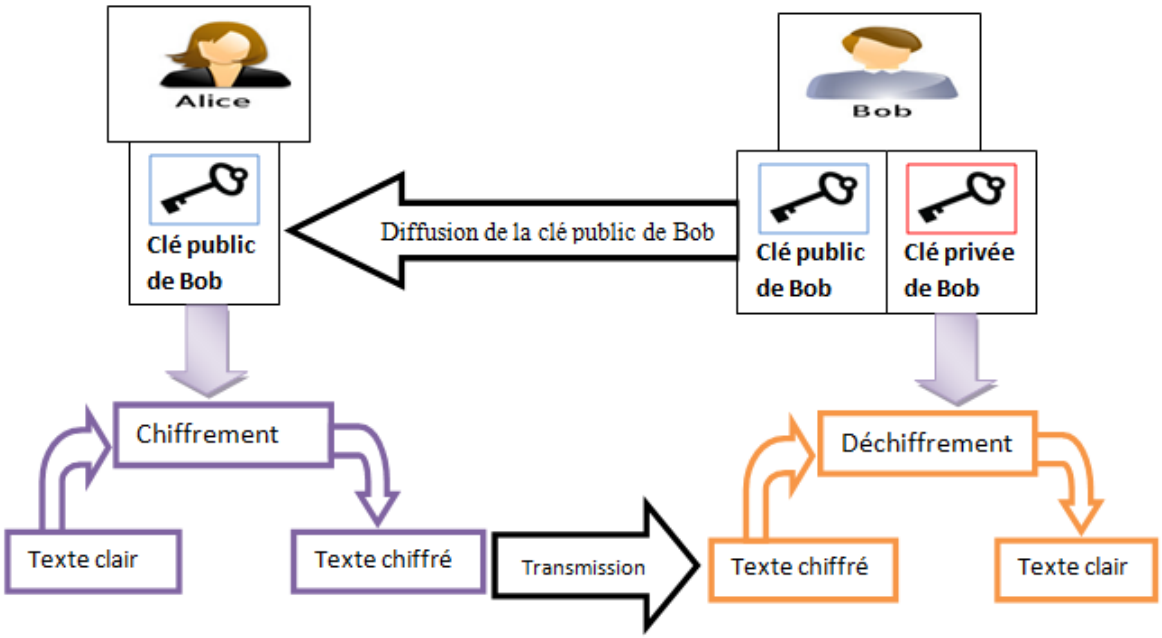

Figure 2.15 : Principe d'un chiffrement à clé publique (Asymétrique).

2.5.2.3 Les algorithmes les plus connus dans la cryptographie asymétrique

2.5.2.3.1 RSA [21]

L'algorithme RSA a été inventé en 1977 par Ron Rivest, Adi Shamir et Len Adleman. Le principe de base de RSA et de considérer un message comme un grand nombre entier et de faire des calcules dessus pour le chiffrer. Il repose sur la factorisation en nombres premiers d'un entier.

1.4.4.3.1 Principe de fonctionnement

- Générer aléatoirement deux nombres premiers **p** et **q**, puis les multiplier pour générer le nombre **n**.
- Déterminer ϕ (n) **: φ (n) = (p-1)*(q-1).**
- Choisir un entier naturel **e** premier tel que **1 < e <φ (n)** et **PGCD(e,φ (n))=1.**
- Calculer l'entier naturel **d** tel que **e*d modφ(n)=1.**

- Le couple **(e, n)** est la **clé publique** du chiffrement, alors que le couple **(n, d)** est sa **clé privée**.
- Pour chiffrer un texte, nous calculons **c** avec : **$c=m^e \bmod n$**.
- Pour déchiffrer un texte chiffré, nous calculons **m** avec **: $m=c^d \bmod n$**.

Avant d'être chiffré, le message original doit être décomposé en une série d'entiers M de valeurs comprises entre 0 et n-1. Pour chaque entier m, il faut calculer $c = m^e \bmod n$.

L'algorithme 1 décrit l'étape de génération des clés de RSA :

Algorithme 1 : Génération de clés avec RSA

1 Sortie : Clé publique (n, e) et clé privée d générées
2 **Début**
3 Sélection au hasard de deux nombres premiers p et q ;
4 Calculer n=p * q et φ (n) = (p-1)*(q-1) ;
5 Choisir un entier e tel que 1<e< φ (n) et le PGCD (e, φ (n))=1 ;
6 calculer d tel que 1<d<φ (n) et e*d _1 (modφ (n)) ;
7 **Fin**

L'algorithme 2 décrit l'étape de chiffrement de message avec RSA :

Algorithme 2 : Chiffrement avec RSA

1 Entée :(n, e) et m // la clé publique et le texte clair m∈ [0,n-1] ;
2 Sortie : c //le texte chiffré ;
3 **Début**
4 Calculer $c=m^e \bmod n$;
5 **Fin**

L'algorithme 3 décrit l'étape de déchiffrement de message avec RSA :

Algorithme 2 :Déchiffrement avec RSA

1 Entée :(n, d) et c // la clé privé et le texte chiffré
2 Sortie : m // le texte clair ;
3 **Début**
4 Calculer $m=c^d \bmod n$;
5 **Fin**

2.5.2.3.2 ECC

Pour former un système cryptographique en utilisant des courbes elliptiques, nous devons trouver un «problème difficile» correspondant à la factorisation du produit de deux nombres premiers comme dans la méthode « RSA » ou à la prise du logarithme discret.

Considérez l'équation Q= kP où Q,P ϵ E_P(a,b)et k<p. Il est relativement facile de calculer Q étant donné K et P, mais il est relativement difficile de déterminer K étant donné Q et P. C'est ce qu'on appelle le problème du logarithme discret pour les courbes elliptiques.

Nous donnons un exemple, considérons le groupe E_{23} (9,17), c'est le groupe défini par l'équation $y^2 \bmod 23 = (x^3 + 9x + 17) \bmod 23$. Quel est le logarithme discret k de Q = (4, 5) à la base P=(16,5) ? La méthode par force brute consiste à calculer des multiples de P jusqu'à ce que Q soit trouvé. Donc

P = (16, 5); 2*P* = (20, 20); 3*P* = (14, 14); 4*P* = (19, 20); 5*P* = (13, 10); 6*P* = (7, 3); 7p=(8,7) ; 8*P* = (12, 17); 9*P* = (4, 5)

Car 9*P* = (4, 5) = *Q le* logarithme discret *Q* = (4, 5) à la base P=(16,5) est K=9 .Dans une application réelle, K serait si grand qu'il serait impossible de rendre l'approche par force brute.

Dans le prochain chapitre, nous allons bien expliquer la cryptographie à base des courbes elliptique.

2.5.2.4 Les fonctions de hachage

Une bonne cryptographie doit pouvoir offrir une garantie de l'intégrité des informations. En effet, il ne doit pas être possible de pouvoir modifier les informations cryptées de façon totalement transparente, un processus de vérification de l'intégrité du message doit être mise en place, Ce processus est réalisé par une fonction de hachage.[28]

Une fonction de hachage H est une fonction qui accepte en entrée un bloc de données M de longueur variable du B bits et produit une valeur de hachage de taille fixe de N bits h=H (M). Où h est appelé une préimage de M. [25]

Figure 2.16 : principe des fonctions de hachage.

2.5.2.4.1 Degrés de résistance des fonctions de hachage

- ✓ Résistance à la première préimage (propriété unidirectionnelle ou one-way property): pour une valeur de hachage donnée h, il est impossible de calculer le message y à partir de sa valeur de hachage h. en d'autres termes, il est impossible par calcul de trouver y tel que H (y) = h.
- ✓ Résistance à la second préimage (faible résistance aux collisions ou weak collision resistance): pour un bloc x donné, il est impossible de trouver deux messages différents (x, y) avec la même valeur de hachage h: Tels que $y \neq x$ avec H (y) = H (X).
- ✓ Résistance à la collision (forte résistance aux collisions ou strong collision resistance): il est impossible de trouver une paire de messages (x, y) telle que H (x) = H (y).

Une fonction de hachage qui satisfait les deux premières conditions est une fonction de hachage faible, si la troisième condition est également satisfaite, elle est alors appelée fonction de hachage forte. [25]

De nombreux algorithmes peuvent être considérés comme des fonctions de hachage telles que MD4, MD5, SHA1, SHA2, SHA3.

2.5.2.4.2 L'utilisation des fonctions de hachage cryptographique

Les fonctions de hachage sont utilisées dans divers applications telles que : [25]

- ➢ Authentification des messages: est obtenue à l'aide du code d'authentification des messages.
- ➢ Signatures numériques: la valeur de hachage du message est chiffrée avec la clé privée de l'utilisateur, afin que le destinataire connaisse la clé publique de l'utilisateur, le destinataire puisse vérifier l'intégrité du message.
- ➢ Fichiers de mots de passe unidirectionnels: dans la plupart des cas, la valeur de hachage des mots de passe est stockée au lieu du mot de passe lui-même, cela protège les mots de passe même si quelqu'un accède aux fichiers qui contiennent des mots de passe.
- ➢ Détection de virus: une valeur de hachage de chaque fichier est stockée en toute sécurité, plus tard, il est possible de déterminer si un fichier a été modifié ou non en recalculant simplement la valeur de hachage du fichier et en comparant la nouvelle valeur avec celle stockée en toute sécurité.

2.5.2.5 Avantages et inconvénients de la cryptographie asymétrique :

Avantages	Inconvénients
Authentification des messages par signature électronique	lent à l'exécution en raison de la charge de calcul
Pas besoin de partager des clés secrètes via une voie de transmission sécurisée	Attaques par substitution de clé possibles
n paires de clés pour n partenaires	la longueur de clé largement augmentée

Table 2.2 : Avantages et Inconvénients de chiffrement asymétrique.

2.5.3 La cryptographie hybride

La cryptographie hybride est un système de cryptographie faisant appel aux deux grandes familles des systèmes cryptographiques : la cryptographie symétrique et la cryptographie asymétrique.

Un cryptosystème hybride consiste à utiliser les avantages des chiffrements symétrique et asymétrique tels que la rapidité d'un cryptosystème symétrique et la possibilité de transmettre la clé secrète par un cryptosystème asymétrique.

Le principe est simple, l'échange des clés pour un chiffrement symétrique est effectue grâce à la cryptographie à clé publique, et les données à échanger sont chiffrées en utilisant un algorithme de chiffrement symétrique, cela rend la communication assez rapide. [21]
Les cryptosystèmes hybrides les plus connus :

2.5.3.1 GnuPG (GNU PrivacyGuard)

GnuPG est un logiciel de cryptage hybride car il utilise une combinaison de cryptographie à clé symétrique pour la vitesse de chiffrement et de cryptographie à clé publique pour faciliter l'échange de clés sécurisées, généralement en utilisant la clé publique du destinataire pour crypter une clé de session qui est utilisée uniquement une fois. Ce mode de fonctionnement fait partie du standard OpenPGP.

GnuPG prend en charge les algorithmes suivants :

Algorithmes asymétrique : RSA, ElGamal, DSA.

Algorithmes symétrique : 3DES, IDEA, CAST5, Blowfish, AES, Camellia.[41]

2.5.3.2 PGP (Petty Good Privacy)

PGP est un logiciel de chiffrement cryptographique, développé et diffusé aux Etats-Unis par Philip Zimmerman en 1991.

PGP se propose de garantir la confidentialité et l'authentification pour la communication des données. Il est souvent utilisé pour la signature de données, le chiffrement et le déchiffrement des textes, des courriels et des fichiers. Utilisant la cryptographie asymétrique mais également la cryptographie symétrique, il fait partie des logiciels de cryptographie Hybride.

Pour le fonctionnement de PGP, une clé secrète est générer (nommée clé de session, valable pour un seul fichier ou un seul message). Le message ou le fichier est chiffré au moyen de cette clé de session avec un algorithme de cryptographie symétrique. Puis cette clé secrète est chiffrée au moyen de la clé publique RSA ou ElGamal du destinataire et ajoutée au début du message ou du fichier. Le destinataire du message déchiffre l'en-tête du message avec sa clé privée RSA ou ElGamal et en extrait la clé secrète qui lui permet de déchiffrer le message. [40]

PGP propose une sélection de plusieurs algorithmes de clés secrètes afin de crypter le message. Par algorithme de clé secrète, nous désignons un chiffrement symétrique utilisant la même clé pour le cryptage et le décryptage. Les trois chiffrements par bloc symétriques offerts par PGP sont CAST, triple DES et IDEA, qu'ils sachent que les trois chiffrements fonctionnent sur des blocs de texte en clair et chiffré de 64 bits. La taille des clés CAST et IDEA atteint 128 bits et celle de DES triple, 168 bits. [41]

2.5.4 Cryptographie quantique

La cryptographie quantique, plus correctement nommée distribution quantique de clés, désigne un ensemble des protocoles permettant de distribuer une clé de chiffrement secrète entre deux interlocuteurs distants, tout en assurant la sécurité de la transmission grâce aux lois de la physique quantique et de la théorie de l'information. Cette clé secrète peut ensuite être utilisée dans un algorithme de chiffrement symétrique, afin de chiffrer et déchiffrer des données

2.6 Travaux attachés

Le travail situé dans **[10]** propose un protocole d'authentification dédie aux réseaux mobiles, basé sur l'échange des clés, tout en tenant compte des contraintes liées à ces réseaux (mobilité, usage économique de l'énergie,…). L'objectif est d'authentifier les nœuds du réseau et de distinguer les nœuds intrus en utilisant comme mécanisme la cryptographie elliptique.

Le travail cité dans **[43]** propose un cryptosystème hybride associant trois algorithmes qui sont le DES, le RSA et SHA-1. Faisant subir à ce cryptosystème hybride une série de tests de

sécurité à travers plusieurs métriques (analyses d'histogrammes, tests de corrélations, contrôle d'intégrité ...etc.). La qualité de ces résultats va mettre en évidence la robustesse du cryptosystème qu'on va ensuite appliquer à la transmission d'images.

Dans un autre travail, situé dans **[20]**, est une application du chaos dans le cryptage des images numériques: un nouvel algorithme de cryptage des images numériques basé sur une combinaison d'une fonction génératrice modifiée de suit des nombres de Lucas balancing et la fonction logistique.

Le travail cité dans **[43]** propose un modèle de cryptographie basé sur les courbes elliptiques qui permet aux applications qui sollicitent un niveau de sécurité de choisir la(les) méthode(s) de chiffrement approprie(s) selon les ressources disponibles afin d'assurer la protection des données sensibles échangées au sein des réseaux mobiles en tenant compte de la capacité des ressources limitées de ses terminaux.

2.7 Conclusion

Étant donné que la sécurité des réseaux mobiles est une nécessité, nous lui avons consacré tout un chapitre dans lequel nous avons mis le point sur les algorithmes de cryptographie ainsi que leur fonctionnement. Les deux systèmes cryptographiques de base à clé secrète et à clé publique souffrent de problèmes et chacun a ses spécificités. La force des algorithmes à clés publiques réside dans la distribution des clés alors que les algorithmes à clés secrètes sont très performants en vitesse de chiffrement. Ainsi, l'intérêt pour augmenter la rapidité et la sécurité des systèmes de cryptage passe certainement par l'utilisation combinée de ces deux techniques, ce que l'on nomme la cryptographie hybride.

Dans le chapitre suivant nous allons expliquer en détaille l'algorithme le plus fort dans la cryptographie symétrique qui est AES (Advanced Encryption Standard ainsi que la cryptographie à base des courbes elliptiques.

Chapitre 03

Chapitre 03: Les algorithmes de base de notre application AES et ECC

3.1 Introduction

La cryptographie moderne se compose de deux grandes familles de cryptographie, chacune d'elle a des points positifs et des point négatifs, la cryptographie symétrique ou la cryptographie à clé secrète est caractérisée par la rapidité d'un cryptosystème symétrique qui utilise peu de ressources systèmes, la cryptographie asymétrique ou la cryptographie à clé publique permet la transmission de la clé secrète.

Dans ce chapitre, nous allons parler de l'algorithme le plus fort dans la famille de cryptographie symétrique qui est AES (Advanced Encryption Standard) qui fournit un niveau de sécurité satisfaisant et un chiffrement rapide en terme de temps d'exécution, et de la cryptographie à base des courbes elliptiques qui se base sur le problème du logarithme discret et qu'elle représente le cryptosystème le plus fort dans la famille de la cryptographie asymétrique.

3.2 AES (Advanced Encryption Standard)

Dans cette section nous allons présenter et détailler le principe de fonctionnement de l'AES.

3.2.1 Présentation de l'AES

Depuis le 26 novembre 2001, l'algorithme de chiffrement par bloc "Rijndael", dans sa version 128 bits, est devenu le successeur du DES sous le nom d'Advanced Encryption Standard (AES). [29]

Issu d'un concours lancé par le National Institute of Standards and Technology(NIST) en 1997, Rijndael a franchi toutes les étapes de sélection et maintenant un standard fédéral américain enregistré sous le numéro FIPS 197. Inscrit dans la National Security Agency (NSA). [29]

Promu par le gouvernement américain, à devenir un standard pour l'échange sécurisé des informations, aux États-Unis et entre les États-Unis et leurs partenaires. Le champ d'application de l'AES a évolué et devient, à compter du premier octobre 2015, l'algorithme de chiffrement des informations jusqu'au niveau TOP SECRET aux États-Unis. De même, il est, aujourd'hui, l'algorithme symétrique de chiffrement par bloc le plus couramment utilisé

en occident. L'AES est un algorithme de chiffrement symétrique par bloc. Il chiffre et déchiffre des blocs de données à partir d'une seule clef.

Contrairement au DES, basé sur un réseau de Feistel, l'AES s'appuie sur un réseau de substitutions et de permutations (SP-network). Ce dernier est constitué de fonctions de substitutions non linéaires contenues dans une S-Box et de fonctions de permutation linéaire. Chaque boîte prend un bloc de texte et la clé en entrée puis fournit un bloc de texte chiffré en sortie.

Les entrées et sorties de l'AES sont des blocs de 128 bits et la longueur de la clé peut être de 128, 192 ou 256 bits.

Pour son fonctionnement interne, chaque bloc de données est organisé en tableau de quatre colonnes et quatre lignes, chaque case contenant un octet, soit $4 \times 4 \times 8 = 128$ bits par tableau. Les opérations de chiffrement et de déchiffrement sont effectuées sur ce tableau, puis, le résultat est copié dans un tableau de sortie. [29]

3.2.2 Principe de fonctionnement [25]

AES est un chiffrement par bloc symétrique prend une taille de bloc de texte en clair de 128 bits ou 16 octets. La longueur de clé peut être de 128, 192 ou 256 bits.

L'entrée des algorithmes de chiffrement et de déchiffrement est un seul bloc de 128 bits. Ce bloc est représenté comme une matrice carrée 4 * 4 d'octets. De même, la clé est représentée sous la forme d'une matrice carrée d'octets, puis la fonction d'extension de clé (key expansion function) génère N+1 clés de tour. Chaque clé de tour sert comme l'une des entrées de la transformation AddRoundKey dans chaque tour.

- Les premiers N-1 tours se composent de quatre fonctions de transformations distinctes: SubBytes, ShiftRows, MixColumns et AddRoundKey, qui sont décrites par la suite. Le tour final ne contient que trois transformations, et il y a une seule transformation initiale (AddRoundKey) avant le premier tour, qui peut être considéré comme le tour 0. La sortie du tour final étant le texte chiffré de 128 bits.
- Le chiffrement se compose de N tours, où le nombre de tours dépend de la longueur de la clé: 10 tours pour une clé de 128 bits, 12 tours pour une clé de 192 bits et 14 tours pour une clé de 256 bits.

Le pseudo code de la fonction de chiffrement de l'AES peut s'écrire de la façon suivante. Dans ce cas, Nb correspond au nombre de mots de 32 bits et par conséquent au nombre de colonnes du tableau d'états et Nr correspond au nombre de tours utilisés dans l'algorithme.