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.

```
1: function Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
2:     byte state[4,Nb]
3:     state ← in
4:     AddRounkey(state, w[0, Nb-1])
5:     for round=1 step 1 to Nr-1 do
6:     SubBytes(state)
7:     ShiftRows(state)
8:     MixColumns(state)
9:     AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
10:    end for
11:    SubBytes(state)
12:    ShiftRows(state)
13:    AddRounkey(state, w[Nr*Nb, (Nr+1)*Nb-1])
14:    return state
15:endfunction
```

3.2.3 Fonctions de transformation AES

Nous passons maintenant à la discussion de chacune des quatre transformations utilisées dans AES. Les opérations de chiffrement s'appuient sur quatre fonctions: AddRoundKey, SubBytes, ShiftRows et MixColumns. Chacune de ces fonctions est exécutée sur le tableau d'états. Le cycle de chiffrement comprend une transformation initiale, des tours intermédiaires et un tour final. Le déchiffrement est réalisé en effectuant les opérations inverses des quatre fonctions de chiffrement, dans l'ordre inverse : InvShiftRows, InvSubBytes et InvMixColumns. La fonction AddRoundKey reste inchangée.

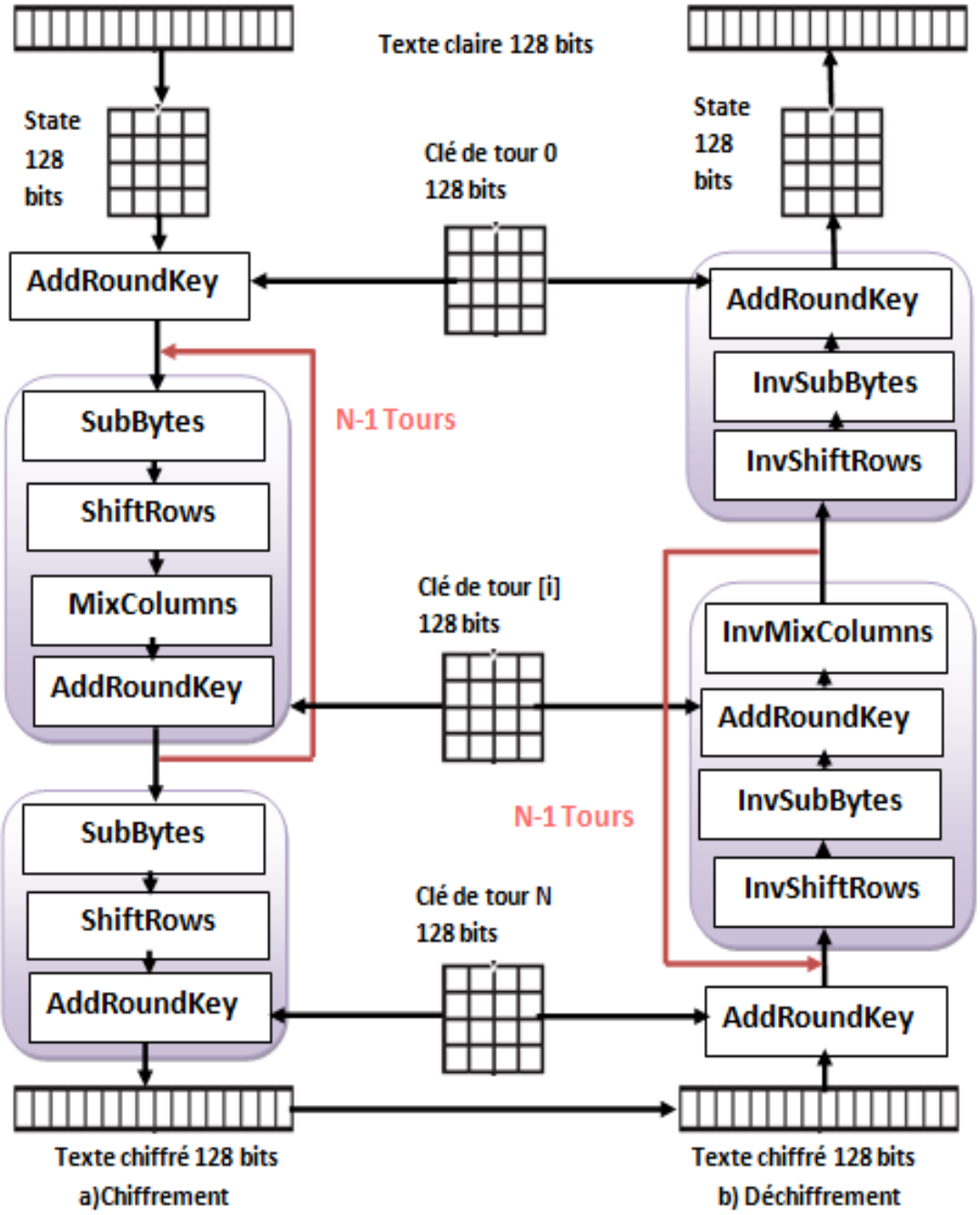

Figure 3.1 : Chiffrement et Déchiffrement AES.

3.2.3.1 SubBytes/InvSubBytes

La transformation d'octet de substitution avant, appelée SubByte, est une simple recherche de table. AES définit une matrice 16 * 16 de valeurs d'octets, appelée S-box, qui contient une permutation de toutes les 256 valeurs de 8 bits possibles. Chaque octet d'état individuel est mappé dans un nouvel octet de la manière suivante: les 4 bits les plus à gauche de l'octet sont utilisés comme valeur de ligne et les 4 bits les plus à droite sont utilisés comme valeur de colonne. Ces valeurs de ligne et de colonne servent d'index dans le S-box pour sélectionner une valeur de sortie 8 bits unique. Par exemple, la valeur hexadécimale [95] fait

référence à la ligne 9, colonne 5. Pour le processus de déchiffrement InvSubByte, c'est les mêmes étapes de SubByte, sauf que la matrice utilisée est Inverse S-box pour faire les permutations.

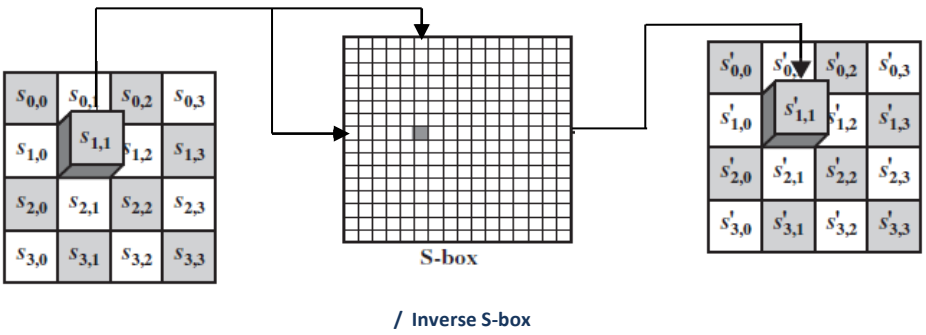

Figure 3.2: SubBytes/InvSubBytes transformation.

	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	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

Figure 3.3: Table de transformation S-Box.

3.2.3.2 ShiftRows/InvShiftRows

La transformation de la ligne de décalage avant, appelée ShiftRows, est illustrée dans la figure 3.4. La première ligne d'état n'est pas modifiée. Pour la deuxième ligne, un décalage circulaire gauche de 1 octet est effectué. Pour la troisième ligne, un décalage circulaire gauche de 2 octets est effectué. Pour la quatrième ligne, un décalage circulaire gauche de 3 octets est effectué.

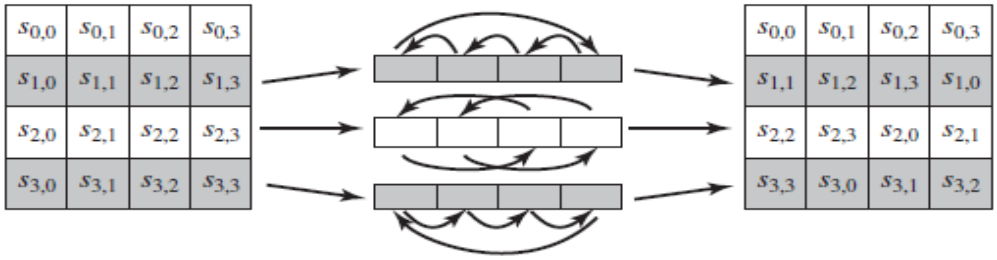

Figure 3.4 : ShiftRows transformation.

La transformation de ligne de décalage inverse, appelée InvShiftRow, effectue les décalages circulaires dans la direction opposée pour chacune des trois dernières lignes, avec un décalage circulaire de 1 octet vers la droite pour la deuxième ligne, et ainsi de suite.

3.2.3.3 MixColumns/InvMixColumns

La transformation de colonne de mixage direct, appelée MixColumns, opère sur chaque colonne individuellement. Chaque octet d'une colonne est mappé dans une nouvelle valeur qui est fonction des quatre octets de cette colonne.

La transformation peut être définie par la multiplication d'une matrice spécifique sur la matrice d'état. Chaque élément de la matrice des produits est la somme des produits des éléments d'une ligne et d'une colonne.

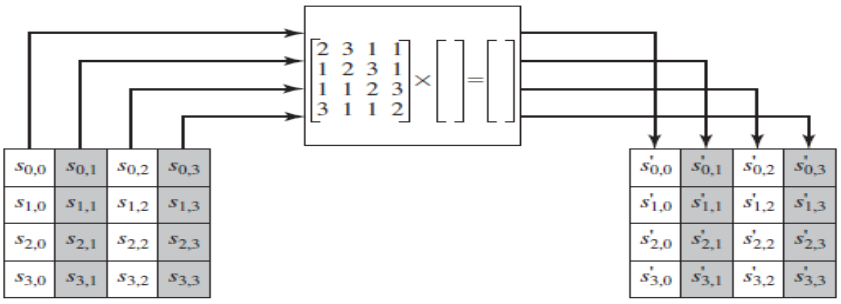

Figure 3.5 : MixColumns Transformation.

La transformation de colonne de mixage inverse, appelée InvMixColumn, est définie par la multiplication de matrice suivante:

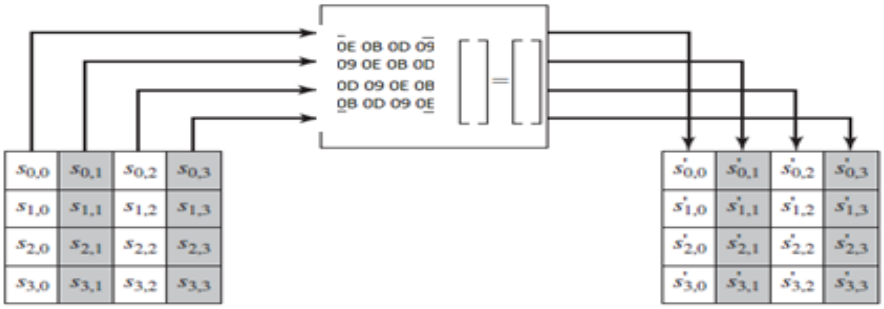

Figure 3.6 : InvMixColumns Transformation.

3.2.3.4 AddRoundKey

Pour avoir la transformation AddRoundKey on ajoute une clé de tour, les 128 bits d'état sont XOR au niveau du bit avec les 128 bits de la clé de tour. L'opération est considérée comme une opération en colonne entre les 4 octets d'une colonne d'état et un mot de la clé de tour.

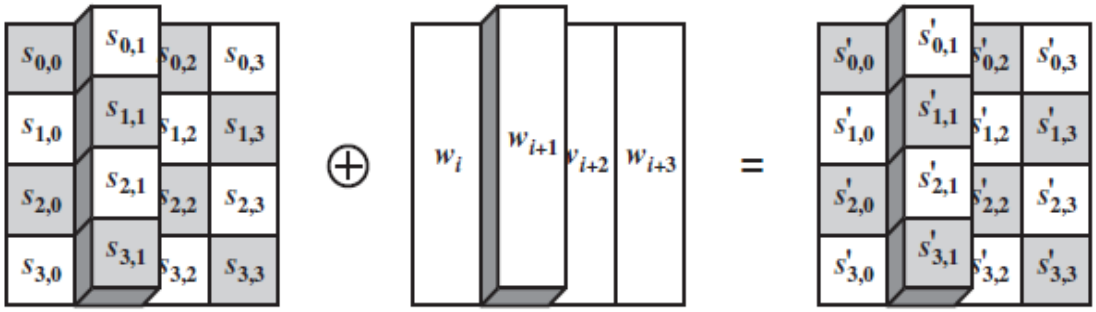

Figure 3.7 : AddRoundKey transformation.

3.2.3.5 Expansion de la clé

Pour rappel, dans l'algorithme de l'AES, Nb = 4 mots et Nr = (10, 12, 14) mots avec 1 mot = 4 octets = 32 bits.

L'algorithme de l'AES, à partir d'une clef de chiffrement K de 128, 192 ou 256 bits, exécute une procédure d'expansion de la clef pour générer les clés de tours. La procédure d'expansion de la clef génère un total de Nb(Nr+1) mots. En effet, l'algorithme a besoin d'un ensemble initial de Nb mots de clef et chacun des Nr tours nécessite également Nb mots de clef pour chiffrer les données. L'algorithme fait intervenir deux fonctions SubWord et RotWord ainsi qu'une constante de tour Rcon.

Le résultat de la procédure d'expansion de la clef est un tableau contenant des mots de 4 octets appelés [wi] avec $0 \leq i <$Nb(Nr + 1).[29]

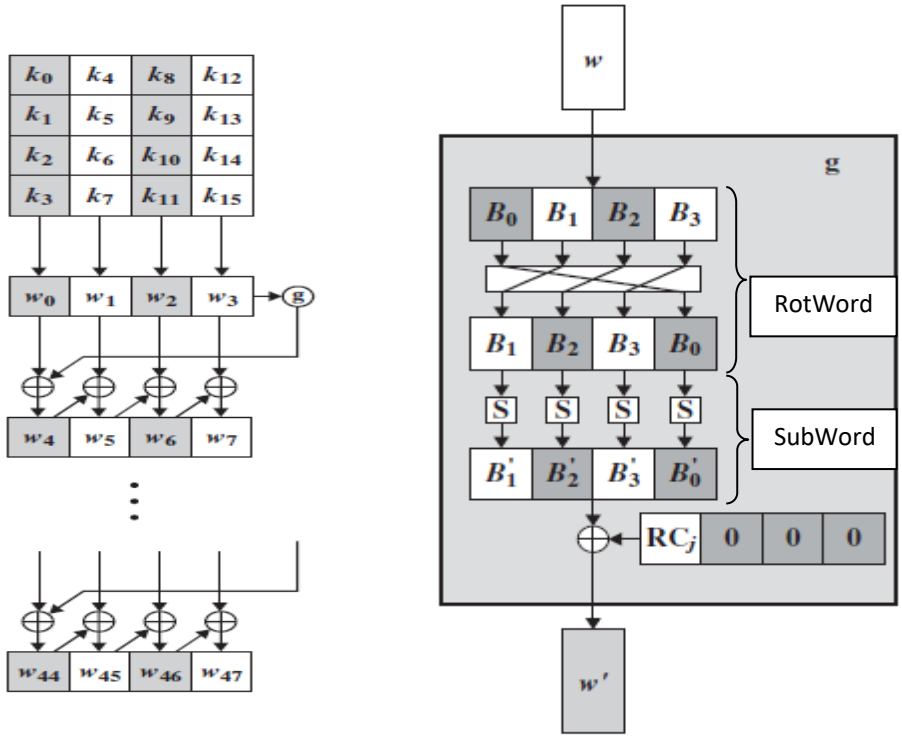

Figure 3.8 : L'algorithme globale de l'expansion de la clé.

3.2.4 Les avantages et les inconvénients de l'AES

Dans cette section nous allons présenter les avantages et les inconvénients de l'AES, ainsi que des solutions pour éliminer ces inconvénients :

- **Les avantages**
  - ✓ Un chiffrement rapide en termes de temps d'exécution
  - ✓ Fournit un très bon niveau de sécurité.
  - ✓ Ses besoins en ressources mémoires sont également très faibles.
  - ✓ Utilise peu de ressources systèmes.
- **Les inconvénients**
  - Problème d'échange de la clé secrète.
  - Problème de distribution des clés n (n-1)/2 Clés pour n partenaires.
  - Tables statiques : SBOX INVSBOX RCON prédéfinie.

3.3 la cryptographie à base des courbes elliptique (ECC)

La plupart des produits et des normes qui utilisent la cryptographie à clé publique pour le chiffrement et les signatures numériques utilisent RSA. L'utilisation sécurisée de RSA exige une longueur considérable de clé, ce qui a alourdi la charge de traitement des applications utilisant RSA, ce qui a des ramifications, en particulier pour les sites de commerce électronique qui effectuent des transactions sécurisées considérables.

Un système concurrent met au défi RSA: la cryptographie à courbe elliptique (ECC). ECC apparaît dans les efforts de normalisation, y compris la norme IEEE P1363 pour la cryptographie à clé publique. L'attraction principale d'ECC, par rapport à RSA, est qu'il semble offrir une sécurité égale pour une taille de clé beaucoup plus petite, réduisant ainsi les frais généraux de traitement. [25]

3.3.1 Généralités

Cette section donne un aperçu des courbes elliptiques et de l'ECC. Nous commençons par une brève revue du concept de groupe abélien et groupe cyclique ensuite nous examinons le concept des corps finis premier et binaire, suivi par un regard sur les courbes elliptiques définies sur des champs finis. Enfin, nous pouvons étudier des chiffrements à courbe elliptique.

3.3.1.1 Groupe

En mathématique, un groupe est un couple (E, ·) où E est un ensemble et · est une loi de composition interne qui combine deux éléments a et b de E pour obtenir un troisième élément a · b. Il faut que la loi satisfasse les quatre axiomes ci-dessous. [30]

- **Fermeture** :∀ (a, b) ϵE :a · b ϵE
- **Associativité** : ∀(a, b) ϵ E :(a · b) · c = a · (b · c)
- **Élément neutre** : ∃e ϵE :a · e = e · a = a
- **Symétrique** : ∀a ϵE, ∃b ϵE :a · b = b · a = e

- **Groupe abélien**

Un groupe abélien, est un groupe dont la loi de composition interne est commutative. Un ensemble E est un groupe commutatif lorsque. [30]

$$\forall (a, b) \epsilon E : a \cdot b = b \cdot a$$

- **Groupe cyclique**

Un groupe fini G est cyclique si tout élément du groupe peut s'exprime sous forme d'une puissance ou d'un multiple d'un élément particulier g, appelé le générateur du groupe, c'est-à-dire G =<g>= $\{g^n | n \epsilon Z^*\}$. Par exemple si G = $\{g^0, g^1, g^2, g^3, g^4, g^5\}$ et $g^6 = g^0$, alors G est un

groupe cyclique. Tout groupe cyclique est abélien car $g^n g^m = g^{n+m} = g^{m+n} = g^m g^n$. L'ordre d'un élément e d'un groupe cyclique est le nombre entier n positif le plus petit tel que ne = 0 (en notation additive) ou $e^n = 1$ (en notation multiplicative). Reprenons le même groupe G du paragraphe précédent, par exemple l'ordre de l'élément g2 est 3 car l'élément neutre du groupe est g0 = 1 et (g2)3 = g6 = 1. [30]

3.3.1.2 Corps

Un corps est un ensemble E muni de deux lois de composition, notée respectivement +et ·. Il faut que les deux lois satisfassent les conditions suivantes :

- Le couple (E, +) forme un groupe abélien, il existe un élément neutre, noté 0, tel que

$$\forall a \in E : a + 0 = 0 + a = a.$$

- Le couple (E\{0}, ·) forme aussi un groupe abélien dont l'élément neutre est 1 :

$$\forall a \in E : a \cdot 1 = 1 \cdot a = a.$$

- La multiplication · est distributif pour l'addition, c'est-à-dire :

$$\forall (a, b, c) \in E |\ a \cdot (b + c) = a \cdot b + a \cdot c \text{ et } (b + c) \cdot a = b \cdot a + c \cdot a.$$

Autrement dit, un corps est un anneau dont les éléments non nuls forment un groupe abélien pour la multiplication. [30]

- **Corps finis**

Un corps fini F est un corps dont le nombre d'éléments est fini. Le nombre d'éléments est l'ordre du corps, noté q, qui peut être représenté par la puissance d'un nombre premier $q=p^n$, où p est un nombre premier, appelé la caractéristique du corps, et $n \in Z^+$. Pour étudier la cryptographie sur les courbes elliptiques, il faut que nous comprenions les deux types de corps ci-dessous. [30]

- **Corps premier**

Un corps est un corps premier, noté F_p lorsque l'ordre du corps $q = p^n$ et p est un nombre premier. Le corps est constitué des nombres entiers {0, 1, 2, ..., p − 1}, et $\forall a \in Z$, a mod p donne le reste unique r qui est compris entre [0, p − 1]. [30]

- **Corps binaire**

Un corps fini de l'ordre 2n est un corps binaire, noté $F_2{}^n$, qui peut être construit en utilisant une représentation polynomiale. Les éléments du corps sont des polynômes binaires dont les coefficients $a_i \in \{0, 1\}$ et les degrés sont inférieurs à n. C'est-à-dire :

$$F_2{}^n = \{a_{n-1z}{}^{n-1} + a_{n-2z}{}^{n-2} + ... + a_{1z} + a_0 : a_i \in \{0, 1\}\}.$$

3.3.2 Présentation des courbes elliptiques

Après la présentation des notions mathématiques nécessaires, nous allons passer, dans cette section, à la définition des courbes elliptiques avec l'ensemble d'opérations que nous pouvons effectuer sur elles.

3.3.2.1 Equation de Weierstrass

Une courbe elliptique sur K, définie comme l'ensemble des solutions de l'équation de Weierstrass suivante.

$$E : F(x ; y ; z) = y^2z + a_1xyz + a_3yz^2 – x^3 – a_2x^2z – a_4xz^2 - a_6z^3$$

Où les coefficients a1, a2, a3, a4 et a6 sont dans K [31]

Pour alléger les notions, nous allons écrire l'équation de Weierstrass avec coordonnées non homogènes : X = x/z et Y = y/z.

$$E : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6.$$

3.3.2.2 Définition des courbes elliptiques

Soit K un corps fini, on appelle courbe elliptique sur K une courbe dans le plan projectif, cubique et sans points singuliers, et munie d'un point distingué qui jouera un rôle particulier : élément neutre.

Elle est donc définie par un polynôme irréductible homogène en trois variables à coefficient dans K . Par un changement de variables homographique, on peut toujours se ramener à une équation dite de Weierstrass : [32]

$$Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$$

Où les coefficients a1, a2, a3, a4 et a6 sont dans K et $\Delta \neq 0$avec : [34]

$$\left.\begin{aligned} \Delta &= -d_2^2d_8 – 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \\ d_2 &= a_1^2 + 4a_2 \\ d_4 &= 2a_4 + a_1 + a_3 \\ d_6 &= a_3^2 + 4a_6 \\ d_8 &= a_1^2 + 4a_2a_6 – a1a_3a_4 + a_2a_3^2 – a_4^2 \end{aligned}\right\}$$

La condition $\Delta \neq 0$ garantit que la courbe est «lisse», c'est-à-dire chaque point a une seule ligne de tangente. [32]

Cependant, l'équation de Weierstrass n'est pas utilisée dans la pratique. Au lieu de cela, selon la caractéristique de K, cette équation peut être grandement simplifiée.

Alors le bon changement de variables transforme l'équation de Weierstrass en courbe:

$$E : Y^2 = X^3 + AX + B$$

Où les constantes A et B doivent satisfaire :

$$4A^3 + 27B^2 \neq 0$$

La courbe elliptique E est l'ensemble des points (x,y) ϵk2 satisfaisant cette équation.[30]

La forme de la courbe peut varier en fonction des paramètres choisis, dans la figure 3.9 nous avons deux exemples des courbes elliptiques.

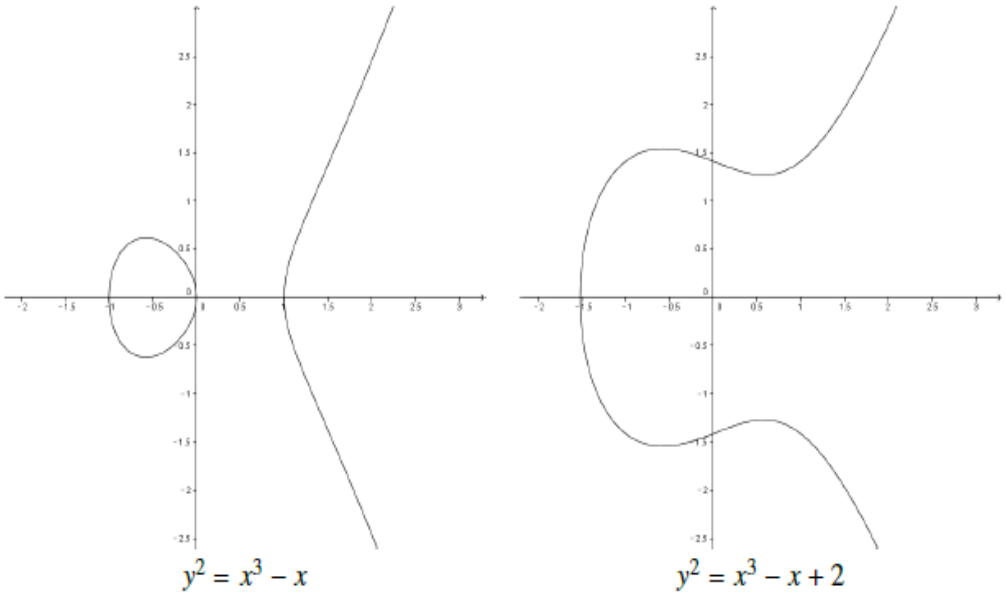

Figure 3.9 : Exemples des courbes elliptiques.

3.3.2.3 Problème du logarithme discret sur les courbes elliptique

Le problème du logarithme discret sur le groupe de points d'une courbe elliptique est fondamental pour nous, puisqu'il s'agit du problème « difficile » sur lequel repose la cryptographie des courbes elliptiques. Ce problème s'exprime comme suit :

Soit G un groupe (note additivement) cyclique fini d'ordre n engendré par un élément P.
Soit H =< P > le sous-groupe engendré par P, alors :
∀Q ϵ H ; ∃n ϵ N : Q = nP

Cet entier n est appelé le logarithme discret de Q en base P et nous noterons $log_p(Q)$.
Le problème du logarithme discret dans un groupe consiste donc à retrouver l'entier n à partir des données publiques (H, P, Q). La sécurité des protocoles basés sur les courbes elliptique repose sur la résolution de ce problème.
Usuellement ce problème est plutôt présenté pour un groupe noté multiplicativement ce qui donne :

Soit G un groupe (noté multiplicativement) cyclique fini d'ordre N engendré par un élément g. Soit h un élément de G. Comme G est un groupe cyclique engendré par g, il existe un unique entier n compris entre 1 et N tel que h = gn. Cet entier n est appelé le logarithme discret de h en base g et nous le noterons log_g(h).

Maintenant nous faisons la correspondance entre les courbes elliptiques et le logarithme discret tel que : Soit E une courbe elliptique définie sur Fp.
Soit P et Q deux points de E(Fp) tels que Q = nP. En résumé, le problème du logarithme discret revient donc à déterminer un entier n tel que Q = nP. [33]

3.3.3 Arithmétiques sur les courbes elliptiques

Dans cette section nous présentons les opérations arithmétiques sur les points dans les courbes elliptiques : multiplications de points, addition de points et doublement de points.

3.3.3.1 Multiplication des points

La multiplication scalaire des points est une opération très importante dans les modèles cryptographiques basés sur les courbes elliptiques. Etant donné un nombre entier k et un point $P \epsilon E(F_p)$, la multiplication scalaire est considérée comme une suite d'additions de P à lui-même k fois. [35]

$$\mathbf{Q = kP =} \underbrace{\mathbf{P + P + \ldots + P}}_{\text{k fois}}$$

Le calcul de l'addition de points est compliqué, notamment quand la courbe est définie dans un corps premier, Il est donc inefficace de répéter successivement l'addition de points. La méthode de base pour accélérer la multiplication scalaire est d'utiliser l'algorithme doublement-et-addition qui est représenté ci-dessous.

Supposons que P est un point sur une courbe elliptique qui est définie sur un corps premier, notée E(Fp), pour calculer kP où k est un nombre entier positif de longueur l bits, nous représentons k en binaire k =$\sum_{i=0}^{l-1} k_i \, 2^i$, et ensuite nous parcourons k du bit de poids faible au bit de poids fort.[30]

Algorithme 1 : Algorithme doublement-et-addition pour calculer Q = kP

Données : k =$\sum_{i=0}^{l-1} k_i\, 2^i$ et PєE(Fp)

Résultat : Q = kP

1 Q ← ∞;

2 **pour** i de 0 à l − 1 **faire**

3 **si** $k_i = 1$ **alors**

4 Q ← Q + P

5 **Finsi**

6 P ← 2P

7 **FinPour**

8 **retourner** Q

3.3.3.2 Addition des points

Soit E(K) :$y^2 = x^3 + ax + b$ une courbe elliptique. Prenons deux points P et Q sur cette courbe. En général, la courbe passant par P et Q recoupe la courbe en un troisième point de coordonnées (x, y).Son symétrique (x, -y) est lui aussi sur la courbe et on le désigne par P+Q pour signifier qu'il est construit à l'aide de P et Q. La chose surprenante est que cette opération "+" possède toutes les propriétés de l'addition des nombres.

Le calcul de l'addition de point est montré dans les formules suivantes :

❖ Soient P = (x1; y1) et Q = (x2; y2) deux points sur E :

Le point (x1;- y1) est l'opposé du point Q et il est noté -Q.

➢ Si P ≠ ±Q, alors R = P + Q = (x3, y3)est défini par :

$$\begin{cases} x3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \\ y3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \end{cases}$$

Une représentation géométrique est donnée dans la figure 3.10. Pour additionner les points P et Q, nous traçons une droite qui passe par ces 2 points, le résultat de l'addition est le point symétrique par rapport à l'axe abscisse du 3e point d'intersection avec la courbe.

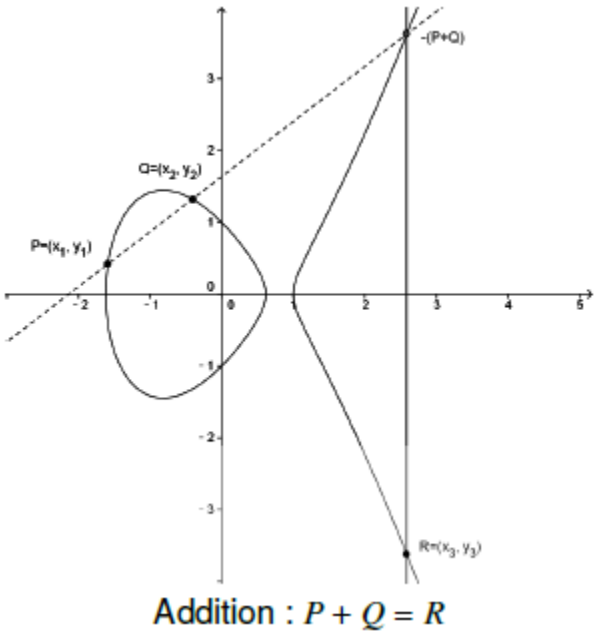

Figure 3.10 : Addition de point sur les courbes elliptiques.

- Si x1 = x2 mais y1 ≠ y2; alors R =V, avec V point à l'infini.

Si P = -Q la droite passant par ces deux points dessinée se croise à un point à l'infini V. D'où P+ (-P) = V. Un négatif d'un point est le symétrique de ce point en ce qui concerne l'axe des abscisses, La figure 3.11 est une l'illustration de ce cas.

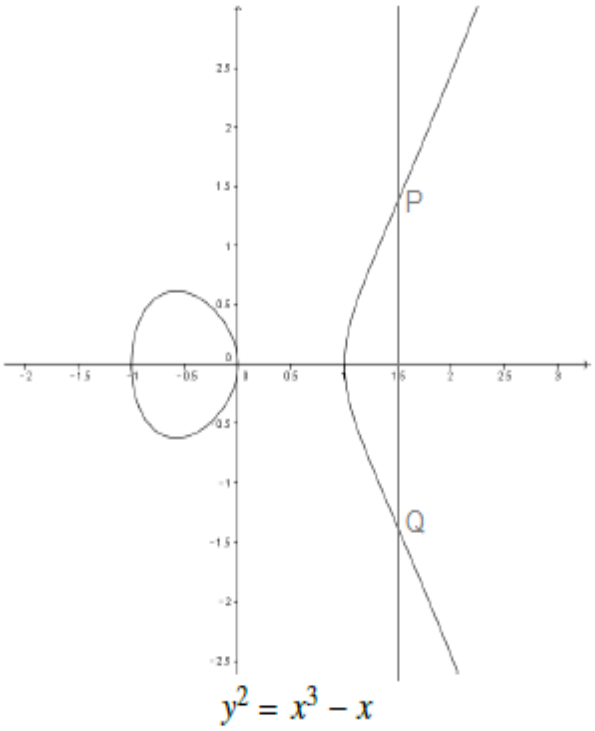

Figure 3.11 : Addition de point lorsque P=-Q.

- Si P = Q et $y_1 = 0$; alors R =V, La figure 3.12 est une l'illustration de ce cas.

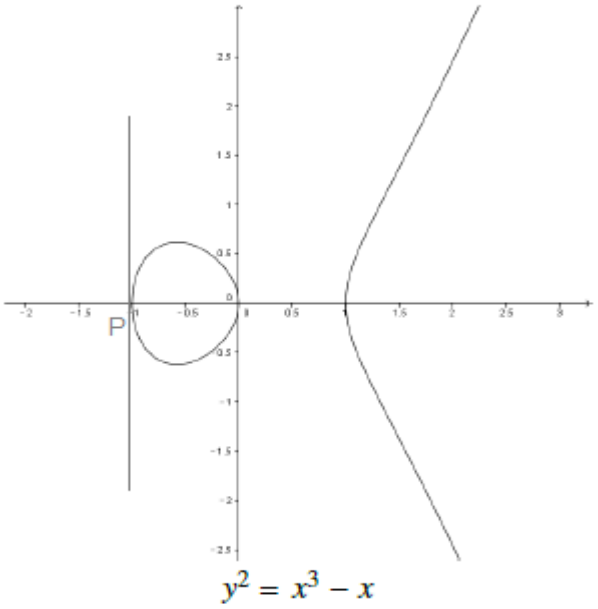

Figure 3.12 : Addition de point P et Q Si P = Q et y1 = 0.

3.3.3.3 Doublement de point

Si la coordonnée du point P n'est pas le zéro alors la ligne de tangente à P croisera la courbe elliptique a exactement encore un point -Q. Le symétrique du point Q. En ce qui concerne l'axe des abscisses donne le point Q, qui est le résultat du doublement du point Q. Ainsi Q = 2P. Comme le montre la figure suivante :

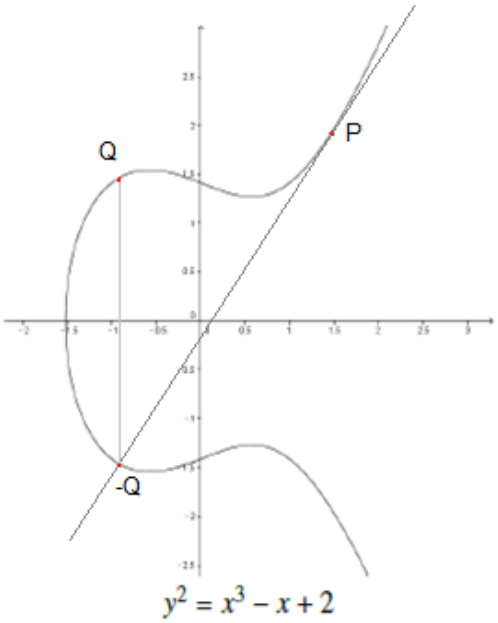

Figure 3.13 : Doublement de point.

Le calcul de doublement de point est montré dans les formules suivantes :

- P = Q, alors le point 2P = (x_3; y_3) est définit par :

$$\begin{cases} x3 = \left(\frac{3x_1^2+a}{2y_1}\right)^2 - 2x_1 \\ y3 = \left(\frac{3x_1^2+a}{2y_1}\right)(x_1 - x_3) - y_1 \end{cases}$$

3.3.3.4 Nombre de points sur une courbe elliptique

Il existe une méthode naïve pour observer que le nombre de points d'une courbe elliptique a une borne maximale. Pour un x donné il existe au maximum 2 points avec des cordonnées opposées solutions de l'équation $_{(p)}$. Nous pouvons donc affirmer qu'il y aura $2p$ + 1 points au maximum sur la courbe.

On connait depuis 1933 des bornes plus précises pour le nombre de points sur une courbe elliptique grâce au théorème de Hasse. On notera $\#E_{a4,6,p}$le nombre de points d'une courbe elliptique, aussi nommé ordre de la courbe elliptique. Le théorème de Hasse assure que :

$$p + 1 - 2\sqrt{p} \leq \#E_{a4,a6,p} \leq p + 1 + 2\sqrt{p}$$

$\#E_{a4,6,p}$= nombre de points

L'ordre de la courbe $E_{-12,20,37}$ est donc compris entre 26 et 50 points. L'ordre exact de cette courbe est de 47.Il faut connaitre l'ordre d'une courbe elliptique pour deux raisons. Tout d'abord ce nombre intervient dans les protocoles cryptographiques et dans la procédure de génération de clé secrète. La complexité du problème du logarithme discret peut être réduite (exemple :# Ea4; a6; p = p) ; ces courbes doivent donc être évitées pour sélectionner uniquement des courbes plus sûres. [35]

3.3.4 Cryptosystème basé sur les courbes elliptiques

Dans cette section nous présenterons quelques cryptosystèmes basés sur les courbes elliptiques tel que : protocole d'échange de clé Deffie-Hellman ECDH et protocole de signature digitale.

3.3.4.1 Protocole d'échange de clé Deffie-Hellman ECDH

L'objectif principal des protocoles d'échange de clés est de mettre en contact deux ou plusieurs entités communiquant via un canal ouvert et probablement non sécurisé, partageant une clé secrète qui assurera la confidentialité et l'intégrité des données pour toute information échangée. Le terme ECDH, nous référons à des schémas d'échange clés basés sur le mécanisme Diffie-Hellman appliqué aux courbes elliptiques.[32]

Le protocole ECDH permet à deux entités A et B de s'accorder sur une clef secrète en communiquant sur un canal non sûr étant donné une courbe elliptique *E(Fp)* et un point *P* de cette courbe générateur du groupe des points. Le protocole se déroule comme suit :

- A tire aléatoirement un entier *a* et calcule $P_a=aP$
- B tire aléatoirement un entier *b* et calcule $P_b=bP$
- A envoie P_a à B et B envoie P_b à A
- A et B calculent alors le secret partagé $P_{ab}=bP_a=aP_b$

Un espion surveillant le canal de communication peut capturer les valeurs des points P_a et P_b lors de l'échange, mais, s'il veut en déduire P_{ab}, il se trouve confronté au problème du logarithme discret pour extraire *a* de P_a ou *b* de P_b.[36]

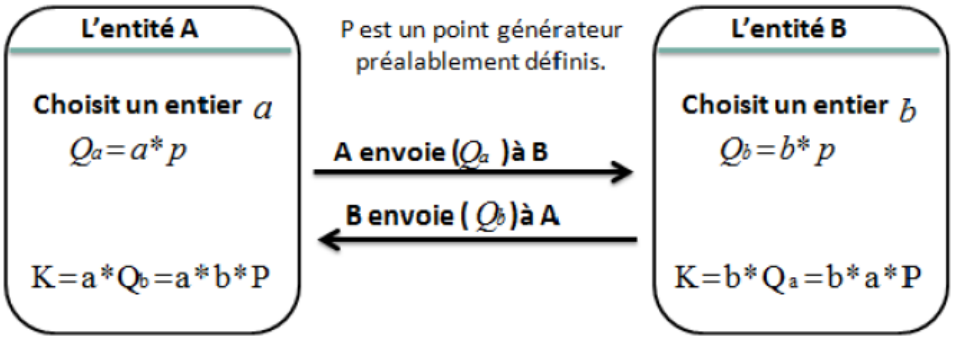

Figure 3.14 : Application de la méthode de Diffie-Hellman aux courbes elliptiques.

3.3.4.2 Cryptosystème elliptique d'ElGamal

Il est facile de créer un analogue direct du cryptosystème ElGamal. Alice veut envoyer un message secret à Bob. Tout d'abord, Bob fabrique une clé publique de la manière suivante. Il choisit une courbe elliptique E définie sur un corps fini Fq de telle manière que le problème du logarithme discret soit plus difficile à résoudre. Sur E(Fq) q sur Fq. Il choisit aussi un point P sur E tel que l'ordre de P soit un grand nombre premier. Il choisit un nombre entier secret s et calcule B = sP. La courbe E, le corps fini Fq et les points P et B sont la clé publique de Bob. La clé secrète de Bob est s. Pour envoyer le message, Alice fait comme suit

1. Elle télécharge la clé publique de Bob.
2. Elle transforme son message en un point M ϵ E(F_q).
3. Elle choisit un nombre entier secret k et calcule M_1 = kP:
4. Elle calcule M_2 = M + kB:
5. Elle envoieM_1 etM_2 à Bob. Bob déchiffre le message en calculant M = M_2 -sM_1:

Nous avons cette égalité parce que :

M_2 -sM1 = (M + kB) -s(kP) = M + k(sP) -skP = M Pour récupérer le texte en clair. [36]

3.3.5 Utilisation de la cryptographie a base des courbes elliptiques

3.3.5.1 Efficacité de la courbe elliptique

Les courbes elliptiques fournissent de bons candidats de groupes [37] :

- Dispose d'une arithmétique efficace.
- Pas de meilleure attaque connue que les attaques génériques.
- Clés plus faciles à générer et plus petites qu'avec RSA.

3.3.5.2 Equivalence entre RSA et ECC

Des travaux existants montrent l'équivalence des tailles des clés offrant le même niveau de sécurité. Le tableau ci-dessous compare la taille des clés utilisées en cryptographie dans RSA et ECC.

Taille de clé	
RSA	**ECC**
1024	160
2048	224
3072	256

Table 3.1 : Comparaison entre ECC et RSA

Nous remarquons que la cryptographie basée sur les courbes elliptiques permet d'utiliser des clés de taille moyenne comparativement à celles du RSA tout en fournissant les mêmes performances. [30]

3.3.5.3 Des applications qui utilisent ECC

De nos jours, plusieurs protocoles et plateformes utilisent la cryptographie à base des courbes elliptiques à cause de sa rapidité et sa sécurité. Parmi celles-ci :

- TLS Le protocole de sécurisation d'échange sur internet utilise la ECC dans la phase "Handshake" pour l'agrément sur la clé à utiliser pour chaque session.
- Smart Cards A coté de RSA, les nouvelles smart-card support la ECC.
- Passeport biométrique La ECC est utilisée pour construire la clé de session.
- Bitcoin : ECC est utilisée pour la signature Transport Layer Security électronique pendant les transactions des paiements avec Bitcoin, la courbe utilisée par Bitcoin c'est Secp256k1 avec l'équation :$y_2 = x_3 + 7$ [38].

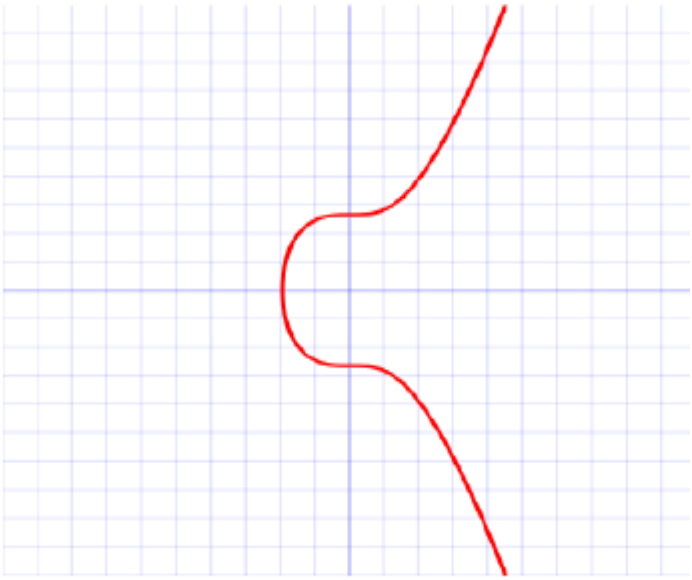

Figure 3.15 : La courbe Secp256k1 utilisé par Bitcoin.

3.4 Conclusion

Dans ce chapitre nous avons présenté l'algorithme le plus fort dans la famille de cryptographie symétrique qui est AES (Advanced Encryption Standard) avec détaille de leur fonctionnement qui fournit un très bon niveau de sécurité et un chiffrement rapide en terme de temps d'exécution. Nous avons présenté de même la cryptographie sur les courbes elliptiques, ainsi que l'ensemble de termes mathématiques indispensables pour comprendre son fonctionnement.