République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Centre Universitaire Belhadj Bouchaib d'Aïn-Témouchent

Institut des Sciences
Département des Mathématiques et de l'Informatique

Mémoire

Pour l'obtention du Diplôme de Master en Informatique

Option : Réseaux et Ingénierie des Données (RID)

Présenté par :
Melle. Hind Ikni

Partage de Secrets Cryptographiques en Utilisant les Courbes Elliptiques

Encadrant :
M. Hichem Bouchakour Errahmani
Maitre Assistant "A" à C.U.B.B.A.T.

Soutenu en 2018

Devant le jury composé de :

Président :	M. Mohamed Réda Saidi (M.C.B)	C.U.B.B.A.T.
Examinateur :	M. Djalal Merad Boudia (M.A.A)	C.U.B.B.A.T.
Encadrant :	M. Hichem Bouchakour Errahmani (M.A.A)	C.U.B.B.A.T.

Partage de Secrets Cryptographiques en utilisant les Courbes Elliptiques

IKNI Hind

juin 2018

Dédicaces

Á mon soutien moral, mon exemple et ma source de tendresse, à ceux qui se sont toujours sacrifiés pour mon bonheur et ma réussite. Mes chers parents, je vous dédie ce modeste travail qui représente les fruits de vos sacrifices, en souhaitant qu'il vous apportera du bonheur, car aucune parole ne saurait exprimer l'estime que j'ai pour vous, et en priant Dieu miséricordieux qu'il vous garde à mes côtés.

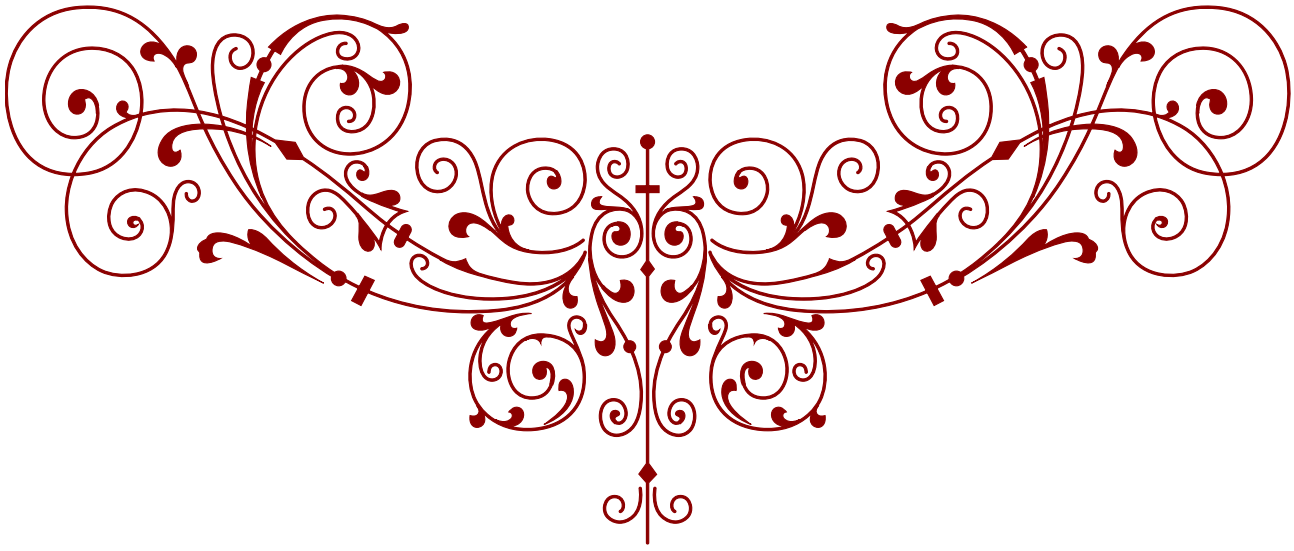
Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner ma reconnaissance.

Je voudrais tout d'abord adresser toute ma gratitude à mon encadrant, monsieur *Hichem BOUCHAKOUR ERRAHMANI*, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion. Merci pour votre inestimable support.

Je voudrais exprimer ma reconnaissance envers mes enseignants du département de mathématique et d'informatique qui m'ont apporté leur support moral et intellectuel tout au long de mon cursus. Un grand merci aux membre de jury qui m'ont honoré en acceptant de juger ce modeste travail.

Je désire également remercier les collègues qui m'ont apporté leur support moral, leur confiance et leur collaboration tout au long de ces années.



Résumé

Dans les systèmes distribués, comme pour n'importe quelle architecture réseau, la cryptographie joue un rôle primordial dans la sécurité des communications, où le partage de secrets en utilisant les courbes elliptiques représente un grand saut pour une multitude d'applications qui requièrent un niveau de sécurité plus ou moins élevé, tel que le vote en ligne, le lancement de missiles, l'accès à un compte bancaire, etc.

Dans ce travail, nous étudions les pré-requis d'un tel cryptosystème avec les différentes bases mathématiques et cryptographiques, comme nous analysons quelques approches qui adoptent le partage de secret basées sur le problème du logarithme discret elliptique, où nous apprenons l'existence des différents protocoles conçus.

En exploitant les notions étudiées, nous nous trouvons aptes de répondre à une multitude de questions, en proposant une méthode de partage d'un secret composé, représenté par une image sous format matricielle qui pourra par la suite être reconstitué sans perte par une structure d'accès qualifiée sur un réseau Client-Serveur, tout en permettant à chacun des clients de vérifier le partage qui lui est attribué.

Mots clés : Partage de secret vérifiable, courbe elliptique, problème du logarithme discret elliptique

Abstract

In distributed systems as in any network architecture, cryptography plays a critical role in communications security, where sharing secrets using elliptic curves is a big leap for a multitude of applications that require a higher level of security, such as online voting, missile launching, bank account access, etc.

In this work, we study the prerequisites of such a cryptosystem with different mathematical and cryptographic bases, we also analyze some approaches that adopt secret sharing based on the elliptic curve discrete logarithm problem, where we learn the existence of different protocols designed.

By exploiting the studied notions, we were able to answer several questions, by proposing a method of multi secret sharing of a given image taken in input as a matrix that could subsequently be reconstructed without loss by a qualified access structure on a Client-Server network, while allowing each client to check the share assigned to it.

Keywords : Verifiable secret sharing, elliptic curves, elliptic curve discrete logarithme problem

Table des matières

Table des figures

Liste des tableaux

Liste des Exemples

Liste des Algorithmes

CHAPITRE I

Introduction générale

Sommaire

L'usage des techniques de sécurité classique pour les données jouaient un rôle dans les domaines diplomatiques et militaires. Cependant, avec le développement des réseaux, des services du web, l'apparition du cloud,etc. Le partage et l'échange d'informations devient de plus en plus des fonctions presque innées qui réclament des droits personnels de confidentialité et nécessitent plus de sécurité pour les données transitantes. C'est dans ce but qu'apparait toujours l'intérêt de la cryptologie comme étant la science du secret, qui prend en charge l'évolution des différentes méthodes de protection telles que le chiffrement, déchiffrement, hachage, etc. Dépendantes toutes d'un paramètre important appelé clé. Suivant les principes de cette science posés par *Kerckhoff*, cette clé doit représenter l'unique secret pour n'importe quel cryptosystème, de ce fait, une information aussi importante nécessite elle même une protection, donc le problème qui reste à se poser a une relation avec la protection d'une telle clé, d'où la notion de partage de secret.

I.1 Motivation

Imaginons que vous êtes en possession d'un coffre fort muni d'une serrure sécurisée par une clé, et que vous vouliez transmettre cette dernière à une autorité qui pourra par la suite se servir de son contenu, échanger la clé avec une seule personne mettra votre clé en danger de perte, de falsification, d'altération, etc. Par contre, si vous ajoutiez deux serrures par exemple, de telle manière où chacune des trois possède deux clés et que vous les dispersiez sur six personnes différentes, la probabilité de perte de clé pour l'accès au contenu du coffre diminue, car on aura plus de combinaisons possibles pour son ouverture ($2^3 = 8$) et le secret sera plus confidentiel qu'avant, car un intrus qui intercepte l'une des clés ne sera jamais en mesure d'accéder au contenu de votre coffre.

I.2 Problématique

Il apparaît ainsi qu'avec les techniques modernes, le partage de secret devient un sujet d'actualité où les chercheurs du domaine de sécurité font de grands efforts pour développer de nouvelles méthodes apportant des solutions à une variété de problèmes. Par conséquent plusieurs travaux ont vu le jour, de sorte où l'on peut se poser les questions suivantes :

- Comment sécuriser une clé (un secret) par partage ?
- Quelles sont les techniques exploitées pour effectuer un partage fiable ?
- Pourquoi l'utilisation des courbes elliptiques ? qu'apportent-elles au domaine de la cryptographie moderne ?
- Comment mettre en place un système de partage de secret en utilisant les courbes elliptiques ?

C'est en se fondant sur l'abondante bibliographie consacrée à la matière, et tout particulièrement sur l'article "How to share a secret " d'Adi Shamir, qu'il fut possible de déterminer comment partager un secret. L'exploitation de ces sources devait permettre de répondre à une série d'interrogations inhérentes au sujet : le partage de secret garantit-il une confidentialité de la clé ? L'utilisation des courbes elliptiques, apporte-elle plus de confidentialité et de complexité au système de partage ? Est ce que les principes de la cryptographie sont maintenus par le partage de secret ?

I.3 Organisation du mémoire

Après un chapitre consacré aux courbes elliptiques, où nous apprenons les différentes notions mathématiques nécessaires sur les opérations de bases dans les courbes elliptiques, l'étude du

partage de secrets nous permet de connaitre quelques techniques de partage. Afin de suivre l'évolution des travaux dans le domaine du sujet, nous allons consacrer un chapitre d'état de l'art où nous étudions quelques approches. Enfin, nous verrons dans un dernier chapitre sur notre contribution dans le partage de secret en utilisant les courbes elliptique, en respectant les points étudiés.

CHAPITRE II

Les Courbes Elliptiques

Sommaire

II.1 Introduction

La contrainte majeur rencontrée dans la conception des cryptosystèmes, est le temps d'exécution face à la complexité des opérations utilisées. Le problème de la taille des clés représente également un problème pour les concepteurs, car ils optaient pour des clés de taille considérable afin d'augmenter le niveau de sécurité de leurs systèmes, ce qui rends les opérations beaucoup plus lourdes, par conséquent ils prennent plus de temps de calcul et exigent un matériel plus sophistiqué.

L'une des techniques qui ont révolutionné la sécurité informatique en apportant des solutions à ces contraintes sont les courbes elliptiques, qui commencent à être de plus en plus utilisées dans les cryptosystèmes modernes, car elles offrent des avantages de performances, que ce soit du point de vue taille de clés, temps de calcul ou même complexité des opérations.

Dans ce chapitre, nous allons d'abord définir qu'est ce qu'une courbe elliptique en présentant ses différentes caractéristiques. Nous entamons par la suite les principales opérations nécessaires à la conception d'un cryptosystème sécurisé, ainsi que le problème du logarithme discret elliptique qui représente la base sur laquelle repose ces systèmes. Enfin, nous aborderons la notion d'échange de clé par les courbes elliptiques, qui nous sera de grande utilité pour une bonne compréhension des différentes approches étudiées.

II.2 Notions de base sur les courbes elliptiques

Bien que les courbes elliptiques apparaissent comme un système purement géométrique, ce système admet différentes opérations algébriques. Cependant il peut être vue comme un ensemble élémentaire, ce qui nous donne l'opportunité d'entamer la notion du groupe et par la suite celle du corps sur une courbe elliptique.

Définissons d'abord ce qu'est une courbe elliptique !

II.2.1 Définition d'une courbe elliptique

Une courbe elliptique est l'ensemble des solutions à une équation de la forme

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

tel que les constantes A et B doivent vérifier l'équation $4A^3 + 27B^2 \neq 0$. Cette condition (appelé discriminant Δ) force le polynôme cubique à avoir des racine distincts (voir contre exemple Figure [II.2]).

Comme présenté sur la Figure [II.1], On peut remarquer que ces courbes ne ressemblent nullement à des ellipses, donc d'où est ce qu'elles ont tiré leurs noms d'elliptique ? En fait, il s'agit de la première fois où est apparue une courbe elliptique, qui était lors d'un calcul de périmètre d'une ellipse qui a mené à ce qu'on appelle une intégrale elliptique [8].

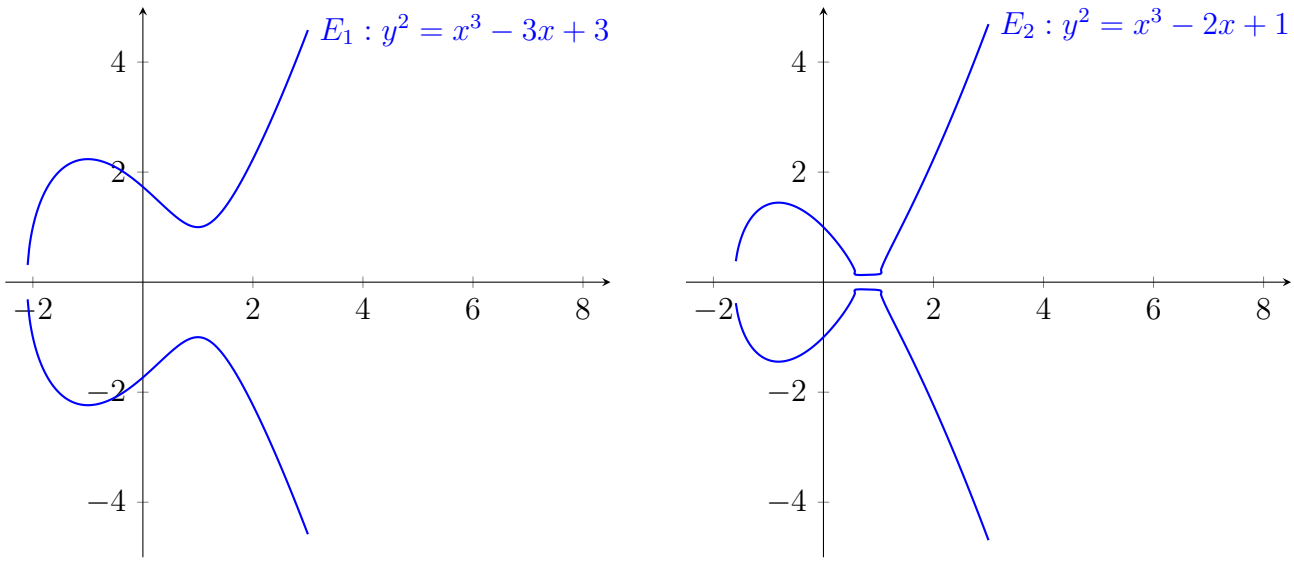


Figure II.1 – Exemples de courbes elliptiques

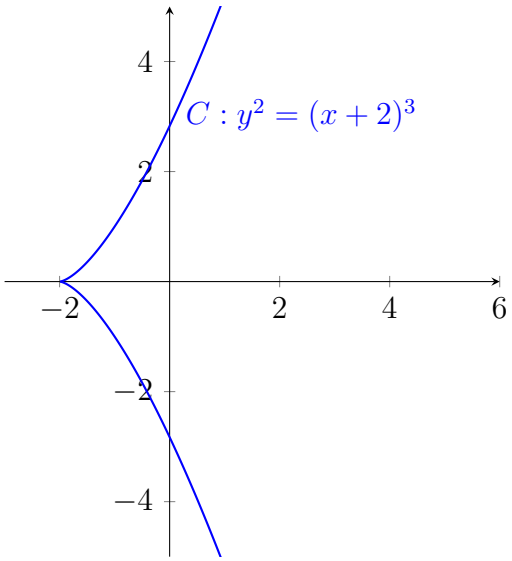


Figure II.2 – Exemple d'une courbe non elliptique ($\Delta = 0$)

Une courbe elliptique est un ensemble qui admet quelques opérations basiques comme n'importe quel ensemble algébrique, que nous détaillons des les sections suivantes :

II.2.2 Opérations sur les courbes elliptiques

Les opérations sur les courbes elliptiques ne sont que les opérations basiques rencontrés dans n'importe quel système (addition, soustraction, multiplication,$\cdots$), sauf qu'elles ont une certaine spécificité de calcul.

II.2.2.1 L'inverse d'un point

Étant donné que les solutions de l'équation d'une courbe elliptiques sont des racines carrées, leurs tracés ont une forme symétrique par rapport à l'axe des abscisses, tel que pour chaque valeur de x on trouve deux solutions $y = \sqrt{x^3 + ax + b}$ et $y' = -\sqrt{x^3 + ax + b}$. Par conséquent, l'inverse d'un point sur la courbe, n'est autre que son image symétrique par rapport à l'axe des

abscisses. En considèrant un point $P(x_p, y_p)$, son inverse $-P$ est définit par les coordonnées $-P(x_p, -y_p)$ [10].

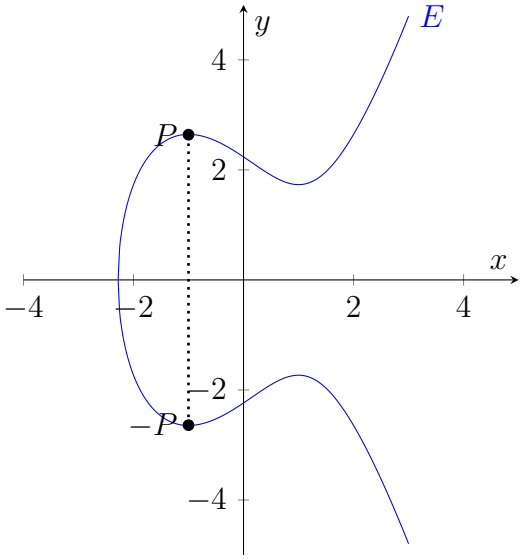


Figure II.3 – L'inverse d'un point

II.2.2.2 Addition de deux points

Ce qui caractérise les courbes elliptiques des autres formes géométriques, est une propriété d'addition particulièrement étonnante, tel que la droite L qui passe par les deux points P et Q de la courbe, croisera obligatoirement E en un troisième point R. L'inverse de R qui est R' représente la somme de P et Q [10].

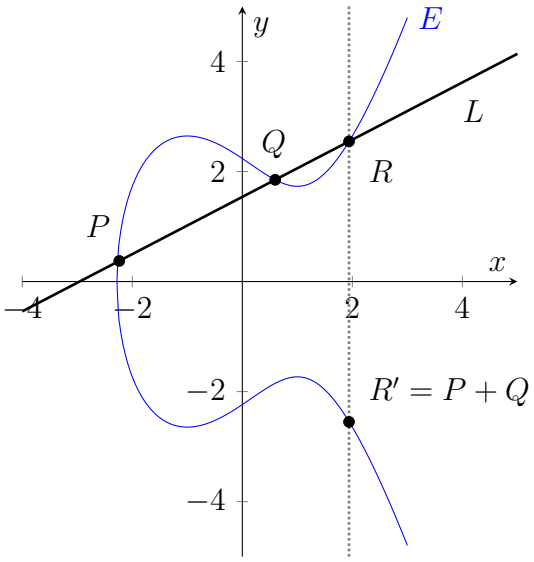


Figure II.4 – Addition de deux points

II.2.2.3 Le double d'un point

Il s'agit d'un cas particulier de l'opération d'addition qui s'offre à nous, où les deux points P et Q soient égaux, donc nous doublons le point. Afin de résoudre ce problème, imaginons le point Q qui se glisse en se rapprochant du point P jusqu'à ce qu'ils se superposent, la ligne L devient la tangente de E en P. Donc, afin de doubler un point, il suffit de tracer la tangente en

ce point, et définir R comme point de croisement de E avec cette tangente, puis le réfléchir par rapport à l'axe des abscisses, ce qui donne le point résultat du double de P [10].

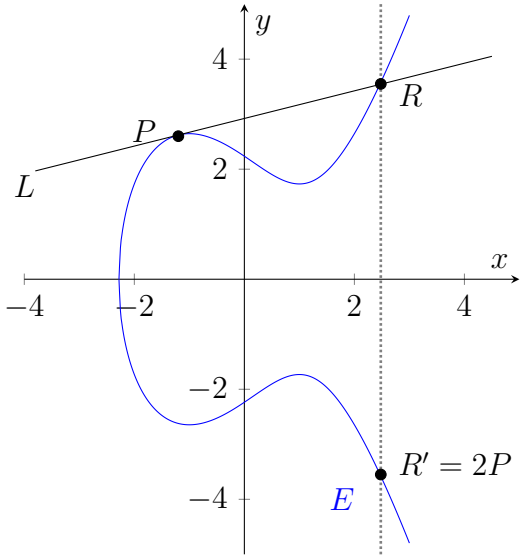


Figure II.5 – Le double d'un point

II.2.2.4 Multiple d'un point

La multiplication d'un point par un scalaire n, est définie comme une suite d'opération d'addition sur ce point

$$n \cdot P = \underbrace{P + P + P + \cdots + P}_{n \ fois}$$

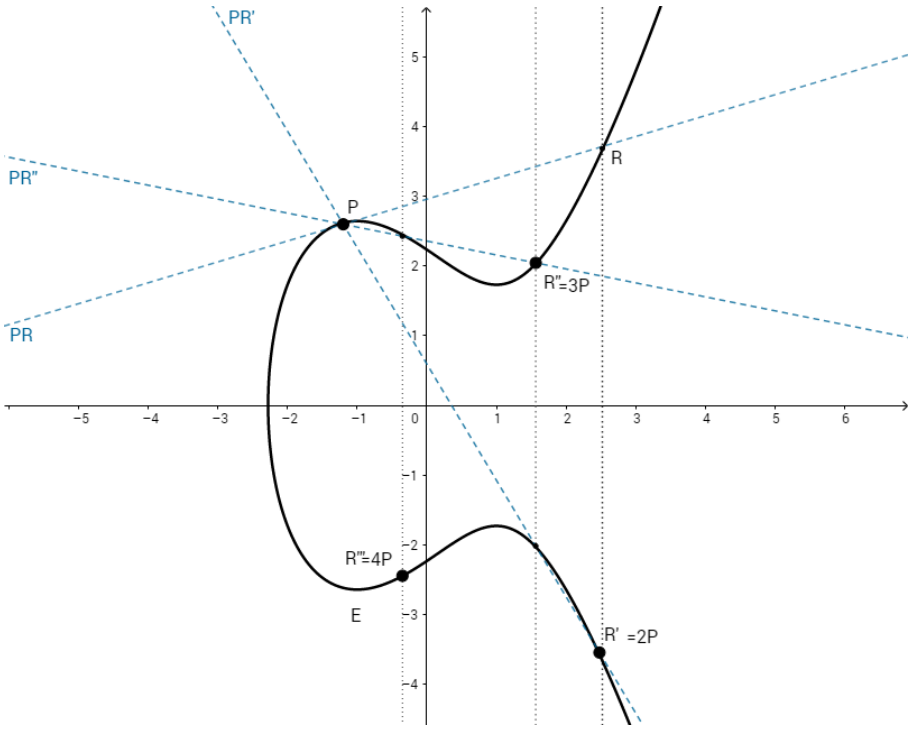


Figure II.6 – Multiple d'un point

II.2.2.5 L'élément neutre

En outre le cas du double d'un point pour l'opération d'addition, nous pourrions rencontrer un deuxième cas particulier, il s'agit de l'addition d'un point $P(x_p, y_p)$ avec son inverse $P'(x_p, -y_p)$. La ligne L traversant P et P' devient une ligne verticale d'équation $x = x_p$, cette droite ne croise pas E dans un troisième point R. afin de déterminer l'addition dans ce cas, un point $\mathcal{O}$ définit à l'infini est présenté, tel qu'on prétend son existence sur chaque ligne verticale du plan, ce qui donne $P + P' = \mathcal{O}$ [10].

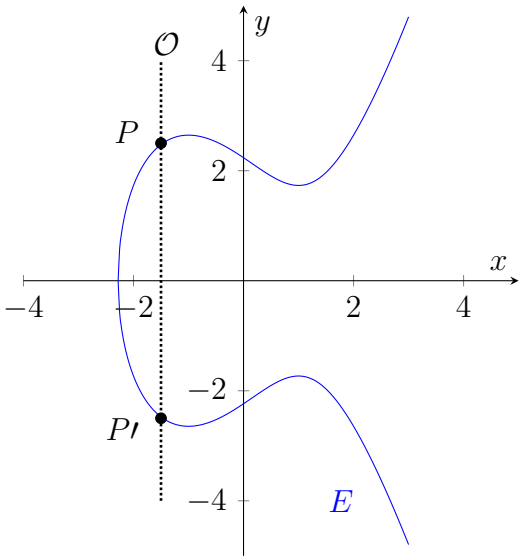


Figure II.7 – L'élément neutre

II.2.3 Algorithme d'addition et double d'un point

Après avoir détaillé les différentes opérations pour une courbe elliptique comme notions mathématiques, on se trouve dans l'obligation de trouver un algorithme efficace qui nous permette de les implémenter pour nos cryptosystèmes. L'algorithme synthétisant ces opérations -de l'anglais *Double and Added*- est présenté comme suit :

Soit E une courbe elliptique définie par $E : Y^2 = X^3 + AX + B$, $4A^3 + 27B^2 \neq 0$. Et soient $P(x_p, y_p), Q(x_q, y_q)$ deux points de E.

Afin de définir les coordonnées de $R(x_r, y_r)$, tel que $R = -(P + Q)$, l'équation suivante est donnée :

$$x_r = \lambda^2 - x_p - x_q \qquad \text{et} \qquad y_r = \lambda(x_p - x_r) - y_p$$

tel que :

$$\lambda = \begin{cases} \dfrac{y_q - y_p}{x_q - x_p}, & \text{si } P \neq Q \\ \dfrac{3x_p^2 + A}{2y_p}, & \text{sinon} \end{cases}$$

Il suffit ensuite d'inverser l'ordonnée y_r en $-y_r$ pour définir $R' = -R = P + Q$.

Algorithme 1 Addition et Double d'un point [16]

Entrées : $P(x_P, y_P), Q(x_Q, y_Q)$
Sorties : $R(x_R, y_R)$
Algorithme :
  Si $P = -Q$ **alors Retourner** $\mathcal{O}$
  Sinon Si $P = \mathcal{O}$ **alors Retourner** Q
  Sinon
    Si $P \neq Q$ **alors**
      $\lambda \leftarrow (y_q - y_p)/(x_q - x_p)$
    Sinon
      $\lambda \leftarrow (3 \times x_p^2 + A)/(2 \times y_p)$
    Fin Si
    $x_r \leftarrow \lambda^2 - x_p - x_q$
    $y_r \leftarrow \lambda \times (x_p - x_r) - y_p$
    Retourner R
  Fin Si

II.2.4 Notion de groupe

Comme détaillé précédemment, pour n'importe quels deux points d'une courbe elliptique, l'opération d'addition donne toujours un point de la même courbe, ce qui fait de cette opération une *loi de composition interne* sur l'ensemble des points E qui résolvent son équation.

A partir des précédentes définitions sur les opérations dans une courbes elliptique, nous pouvons synthétiser les propriétés suivantes :

a) **Loi interne :** Si $P, Q \in E$ et $P + Q = R$ donc $R \in E$.

b) **Associativité :** Pour tout $P, Q, R \in E, (P + Q) + R = P + (Q + R)$ (voir Figure [II.8]).

c) **Élément neutre :** L'élément à l'infini $\mathcal{O}$ est un élément neutre tel que

$$\forall\ P(x, y) \in E,\ P + \mathcal{O} = P.$$

d) **L'inverse d'un élément :** Chaque élément dans E admet un inverse tel que

$$\forall\ P(x, y) \in E,\ \exists\ -P(x, -y) \in E.$$

e) **Commutativité :** L'addition de deux points de la courbe est une opération commutative, car la droite joignant P et Q croisant E dans R est la même droite joignant Q et P (l'ordre des points n'a pas d'importance) $\forall\ P, Q \in E : P + Q = Q + P$.

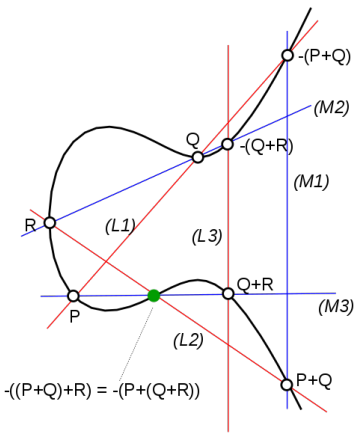


Figure II.8 – Associativité de l'addition sur une courbe elliptique [18]

D'après les propriétés suscitées, nous déduisons que $(E, +)$ forme un groupe abélien (commutatif).

II.3 Les courbes elliptiques sur les corps finis

Dans les exemples de courbes présentés précédemment, les valeurs des coordonnées des points ont été prises dans l'ensemble des réels $\mathbb{R}$. En revanche, ces valeurs peuvent être définies sur un corps fini, et c'est même exigé en cryptographie, où on implémente des courbes elliptiques définies sur un certain corps $\mathbb{F}_p$, tel que p est premier [16]. Donc, la définition d'une courbe elliptique cryptographique est donnée comme suit :

Une courbe elliptique sur un corps fini $\mathbb{F}_p$ avec p premier, est l'ensemble des coordonnées $X, Y \in \mathbb{F}_p$ qui résolvent l'équation

$$Y^2 \equiv X^3 + AX + B \pmod{p}$$

union $\mathcal{O}$, tel que $A, B \in \mathbb{F}_p$ et $4A^3 + 27B^2 \neq 0 \pmod{p}$.

Afin de définir les coordonnées du résultat $P + Q$, on complète les équations de calcul précédentes comme suit :

$$x_r = \lambda^2 - x_p - x_q \pmod{p} \qquad \text{et} \qquad y_r = \lambda(x_p - x_r) - y_p \pmod{p}$$

tel que :

$$\lambda = \begin{cases} (y_q - y_p)(x_q - x_p)^{-1} \pmod{p}, & \text{si } P \neq Q \\ (3x_p^2 + A)(2y_p)^{-1} \pmod{p}, & \text{sinon} \end{cases}$$

et pour définir l'ordonnée de l'inverse de R, $-y_r \pmod{p}$.

Exemple 1: Courbe elliptique sur un corps fini

Soit la courbe $E : y^2 = x^3 + 3x + 8$, le graphe de E sur l'ensemble des réels $\mathbb{R}$ est représenté par la Figure [II.9].

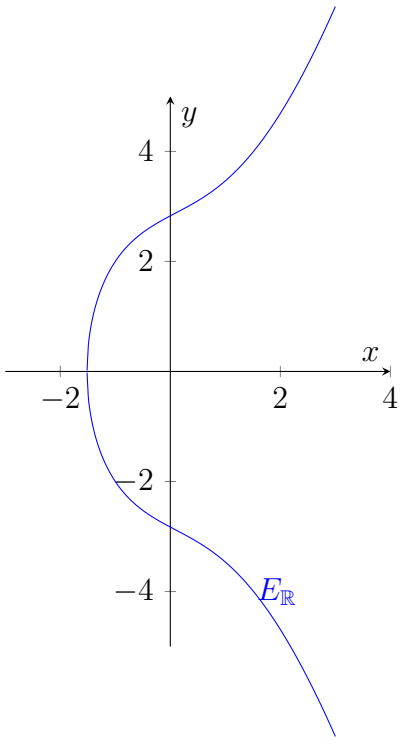


Figure II.9 – Graphe de la courbe $E : y^2 = x^3 + 3x + 8$ sur $\mathbb{R}$

Si nous voulons tracer le graphe de la même courbe sur le corps $\mathbb{F}_{13}$, il apparait sous forme de points représentant l'ensemble des couples $(x, y) \in \mathbb{F}_{13}^2$ qui résolvent l'équation $E : y^2 = x^3 + 3x + 8 \pmod{13}$ (Figure [II.10]).

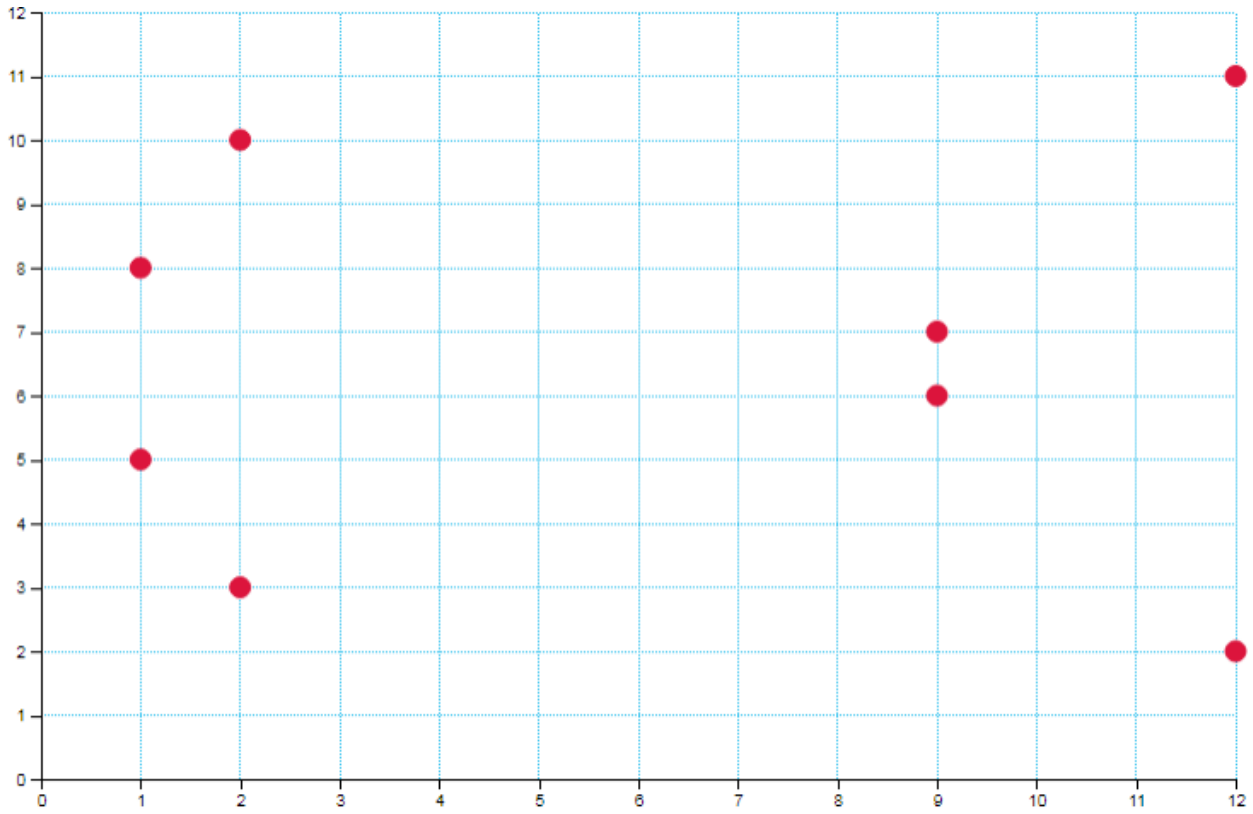


Figure II.10 – Les points représentants la courbe $E : y^2 = x^3 + 3x + 8$ sur $\mathbb{F}_{13}$

L'ensemble des points résolvant l'équation :

$$E(\mathbb{F}_{13}) = \{\mathcal{O}, (1,5), (1,8), (2,3), (2,10), (9,6), (9,7), (12,2), (12,11)\}$$

La définition de toutes les combinaisons d'additions de points possibles sur ce corps, peut être présentée sous forme d'une table [II.1] :

+	$\mathcal{O}$	(1,5)	(1,8)	(2,3)	(2,10)	(9,6)	(9,7)	(12,2)	(12,11)
$\mathcal{O}$	$\mathcal{O}$	(1,5)	(1,8)	(2,3)	(2,10)	(9,6)	(9,7)	(12,2)	(12,11)
(1,5)	(1,5)	(2,10)	$\mathcal{O}$	(1,8)	(9,7)	(2,3)	(12,2)	(12,11)	(9,6)
(1,8)	(1,8)	$\mathcal{O}$	(2,3)	(9,6)	(1,5)	(12,11)	(2,10)	(9,7)	(12,2)
(2,3)	(2,3)	(1,8)	(9,6)	(12,11)	$\mathcal{O}$	(12,2)	(1,5)	(2,10)	(9,7)
(2,10)	(2,10)	(9,7)	(1,5)	$\mathcal{O}$	(12,2)	(1,8)	(12,11)	(9,6)	(2,3)
(9,6)	(9,6)	(2,3)	(12,11)	(12,2)	(1,8)	(9,7)	$\mathcal{O}$	(1,5)	(2,10)
(9,7)	(9,7)	(12,2)	(2,10)	(1,5)	(12,11)	$\mathcal{O}$	(9,6)	(2,3)	(1,8)
(12,2)	(12,2)	(12,11)	(9,7)	(2,10)	(9,6)	(1,5)	(2,3)	(1,8)	$\mathcal{O}$
(12,11)	(12,11)	(9,6)	(12,2)	(9,7)	(2,3)	(2,10)	(1,8)	$\mathcal{O}$	(1,5)

Table II.1 – Table d'addition des points pour $E : y^2 = x^3 + 3x + 8$ sur $\mathbb{F}_{13}$ [10]

Puisque une courbe elliptique définie sur un corps fini est représentée sous forme de points, est il possible de calculer le nombre de ces points qui résolvent l'équation ?

La réponse est oui, on l'appelle l'*ordre* d'une courbe elliptique. Cependant il est considéré comme un problème difficile de l'ordre $O(p)$ si p est grand. Par contre, différents algorithmes ont vu le jour, parmi les plus utilisés celui de *René Schoof* qui se base sur un théorème limitant le cardinal de E dans un intervalle. Proposé par *Helmut Hasse* (1936), ce théorème donne un majorant et un minorant de l'ordre du groupe de points de la courbe elliptique sur un corps fini :

Théorème (***Hasse***)**.** *Étant donné une courbe elliptique E sur $\mathbb{F}_p$, le nombre de points sur la courbe $\#E$ est délimité comme suit [16]*

$$\#E(\mathbb{F}_p) = p + 1 - t_p \text{ avec } |t_p| \leq 2\sqrt{p}$$

ce qui donne

$$p + 1 - 2\sqrt{p} \leq \#E(\mathbb{F}_p) \leq p + 1 + 2\sqrt{p}$$

donc

$$t_p = p + 1 - \#E(\mathbb{F}_p)$$

Exemple 2: Théorème de *Hasse*
Prenons la courbe de l'exemple précédent $E(\mathbb{F}_{13}) : y^2 = x^3 + 3x + 8 \pmod{13}$ et appliquons le théorème de *Hasse* :

$$13 + 1 - 2\sqrt{13} \leq \#E(\mathbb{F}_{13}) \leq 13 + 1 + 2\sqrt{13}$$

$2\sqrt{13} = 7,21$ donc :

$$14 - 7,21 \leq \#E(\mathbb{F}_{13}) \leq 14 + 7,21$$
$$6,79 \leq \#E(\mathbb{F}_{13}) \leq 21,21$$

Effectivement, suivant la liste de l'ensemble des points déjà donné pour cette courbe, en incluant $\mathcal{O}$, on compte 9 points et $6,79 \leq 9 \leq 21,21$.

Le théorème de *Hasse* sert largement à identifier le corps approprié pour un nombre de points précis dans une courbe elliptique, et ce pour des fins de sécurité purement cryptographiques. Par exemple, si nous avons besoin d'une courbe avec 2^{160} éléments, nous devons utiliser un nombre premier de longueur d'environ 160 bits [16].

II.4 Problème du logarithme discret elliptique

La notion du logarithme discret est une théorie largement utilisée dans la conception des cryptosystèmes, elle consiste à résoudre le problème suivant sur un groupe cyclique $\mathbb{G}_p$.

Ayant deux nombre h et g tel que $h \equiv g^x \pmod{p}$, où g est un générateur de $\mathbb{G}_p$, La résolution du problème de logarithme discret revient à trouver la valeur de x.

Il en est de même si nous transformons ce problème dans l'ensemble E des points d'une courbe elliptique sur $\mathbb{F}_p$. Mais, avant d'entamer la notion de logarithme discret, il nous faut définir un groupe cyclique [1] $\mathbb{G}_p$ dans l'ensemble d'éléments de E.

1. Un groupe $(G, *)$ est un groupe cyclique s'il existe un élément $a \in G$ tel que : $\forall\ x \in G,\ \exists k \in \mathbb{Z}$ tel que $x = a^k$

II.4.1 Groupe cyclique dans E

L'ensemble des points qui résolvent l'opération de multiple d'un point de E sur un corps fini $\mathbb{F}_p$, représente un groupe cyclique pour une courbe elliptique, tel que :

Étant donné deux points P et Q choisis de E, et partagés publiquement, la résolution du problème de logarithme discret revient à déterminer la valeur du scalaire n satisfaisant l'équation :

$$Q = \underbrace{P + P + P + \cdots + P}_{n \ additions} = n \cdot P$$

tel que : $1 \leq n \leq \#E$

Exemple 3: Problème du logarithme discret dans E

Soit la courbe elliptique $E : y^2 \equiv x^3 + 2x + 2 \pmod{17}$. L'ordre de cette courbe $\#E = 19$, et les points résolvants $E(\mathbb{F}_{17})$ forment un groupe cyclique.

Soit le point $P(5, 1) \in E$, calculons tous les multiples de P :

$2P = (5,1) + (5,1) \pmod{17} = (6,3)$	$11P = (13,10)$
$3P = 2P + P = (10,6)$	$12P = (0,11)$
$4P = (3,1)$	$13P = (16,4)$
$5P = (9,16)$	$14P = (9,1)$
$6P = (16,13)$	$15P = (3,16)$
$7P = (0,6)$	$16P = (10,11)$
$8P = (13,7)$	$17P = (6,14)$
$9P = (7,6)$	$18P = (5,16)$
$10P = (7,11)$	$19P = \mathcal{O}$

On a $19P = \mathcal{O}$, donc l'ordre de P est $19 = \#E$, ce qui fait de P un générateur de notre ensemble de points résolvants l'équation E car tous les éléments de E on été engendrés par P.

L'algorithme permettant de calculer nP, n'est qu'une version adaptée à celui d'exponentiation rapide, il suffit de remplacer les opérations exponentiation et multiplication dans $\mathbb{Z}$ par le double d'un point et l'addition de deux point dans $E(\mathbb{F}_p)$ respectivement.

L'algorithme prend en entrée le point P qu'on voudrait multiplier ainsi que l'entier n qu'on convertit en binaire tel que :

$$n = n_0 \cdot 2^0 + n_1 \cdot 2^1 + n_2 \cdot 2^2 + \cdots + n_{t-1} \cdot 2^{t-1} \text{ avec } n_0, n_1, n_2, \cdots, n_{t-1} \in \{0, 1\}$$

L'algorithme utilise deux registres (qu'on notera D et A) pour accumuler les résultats du dédoublement et d'addition respectivement, ensuite boucle en parcourant le scalaire n depuis le poids le plus fort au plus faible et teste chacune de ses valeurs (0 ou 1). Le registre D est initialisé par la valeur du point P, tandis que A par le point à l'infini car il représente l'élément neutre de l'opération d'addition dans E.

Algorithme 2 Multiplication d'un scalaire par un point [16]

Entrées : P :point, n :entier
Initialisation :
$D \leftarrow P \qquad A \leftarrow \mathcal{O}$
Sorties : A
Algorithme :
$d \leftarrow \sum_{i=0}^{t-1} n_i \times 2^i$
Pour $i \leftarrow t-1$ à 0 **faire**
$D \leftarrow D + D \pmod{n}$
Si $d_i = 1$ **alors**
$A \leftarrow A + D \pmod{n}$
Fin Si
Fin Pour
Retourner A

II.4.2 Complexité du problème du logarithme discret

La totalité des protocoles cryptographiques basés sur les courbes elliptiques, reposent sur la difficulté de résoudre le problème du logarithme discret, car les opérations d'addition et de multiplication que nous venons d'entamer sur l'ensemble $E(\mathbb{Z}_p)$, sont plus complexes que leur équivalentes dans $\mathbb{R}$.

La sécurité des cryptosystèmes basés sur les courbes elliptiques, repose sur le manque d'un algorithme de calcul qui pourrait résoudre le problème du logarithme discret elliptique en un temps polynomial. Autrement dit, dans moins de $O(\sqrt{p})$ étapes. Malgré la nature très structurée du groupe $E(\mathbb{F}_p)$, les algorithmes les plus rapides connus pour résoudre le problème du logarithme discret elliptique ne sont pas mieux que l'algorithme générique qui fonctionne aussi bien pour résoudre le problème du logarithme discret dans n'importe quel groupe. Il existe des courbes et des nombres premiers pour lesquels le problème du logarithme discret dans $E(\mathbb{F}_p)$ esr comparativement facile. Ces cas particuliers sont à éviter dans la conception des cryptosystèmes sécurisés [10].

II.5 Protocole d'échange de clé pour les courbes elliptiques

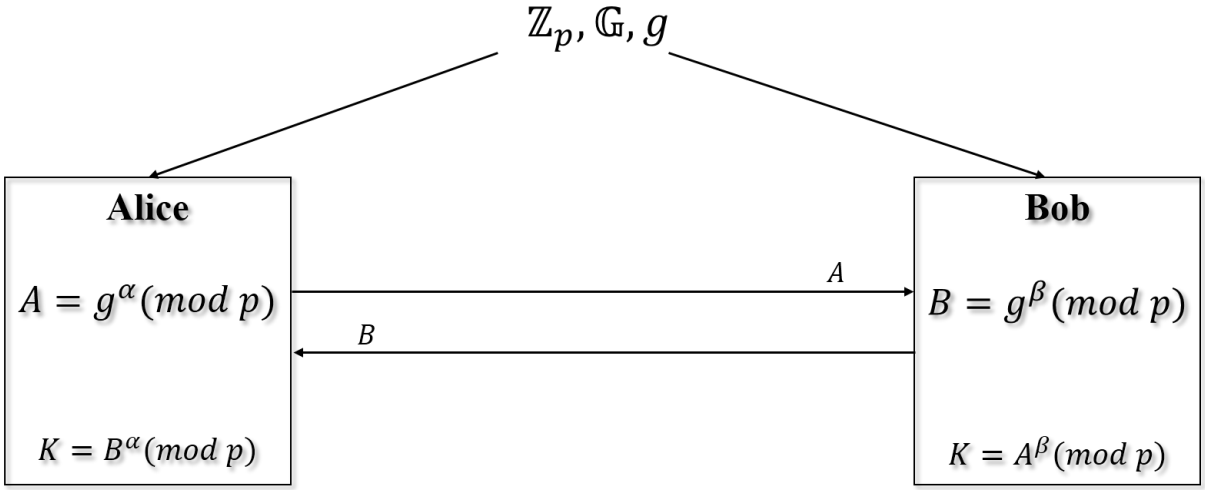
En cryptographie, on appelle échange de clé toute opération permettant à deux parties communicantes sur un canal non sécurisé, d'établir une clé secrète commune. Ce type de protocole exige des fonctions mathématiques dites à sens uniques. Le premier protocole ayant vu le jour a été publié par *Whitfield Diffie* et *Martin Hellman* en 1976, d'où son nom *Diffie-Hellman*. Il fut également le premier schéma cryptographique asymétrique.

Basé sur le problème du logarithme discret et se servant de la propriété de commutativité de l'exponentiation, le schéma de Diffie-Hellman apporta une solution a une variété de protocoles de communication réseau, tel que SSH (Secure Shell), IPSec (Internet Protocol Security) et TLS (Transport Layer Security) [16].

$$k = (\alpha^x)^y \equiv (\alpha^y)^x \pmod{p}$$

L'échange entre les deux communicants *Alice* et *Bob* nécessite pour chacun d'eux une clé secrète et une autre publique, cette dernière sera échangée afin de pouvoir calculer une clé commune pour le chiffrement et de déchiffrement des messages chacun de son côté.

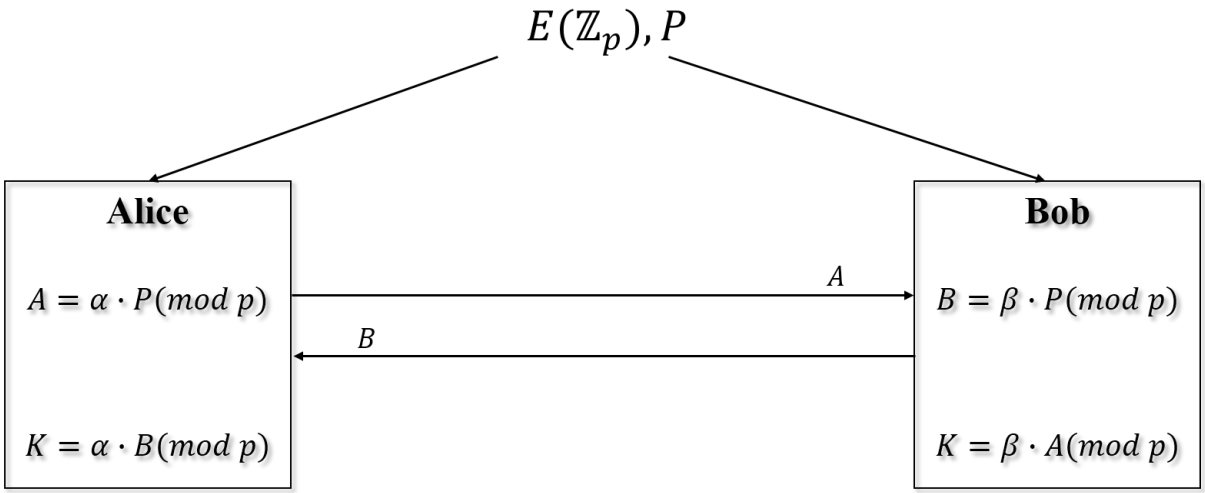
Alice et *Bob* doivent se mettre d'accord sur un corps $\mathbb{Z}_p$ commun, tel que p est premier, ils choisissent un générateur g d'un groupe cyclique $\mathbb{G}$. Chacun d'eux choisira ensuite une clé secrète $\alpha, \beta \in \mathbb{Z}_p$ par lesquelles ils calculent leur clés publiques A et B respectivement, qui seront échangées entre eux par la suite comme démontré dans la Figure [II.11].



$$K = (g^{\beta})^{\alpha} = g^{\alpha\beta} = g^{\beta\alpha} = (g^{\alpha})^{\beta} = K$$

Figure II.11 – Échange de clé de Diffie-Hellman

Le schéma de Diffie-Hellman pourrait être appliqué sur le corps fini représenté par une courbe elliptique $E(\mathbb{Z}_p)$, pour donner naissance à ce qu'on appelle le protocole d'échange de clé elliptique de Diffie-Hellman. Il suffit juste de faire usage du problème de logarithme discret elliptique au lieu de celui utilisé dans $\mathbb{Z}$. Pour les données publiques, cette fois Alice et Bob vont choisir une courbe elliptique particulière (cryptographique) $E(\mathbb{Z}_p)$ ainsi qu'un point commun $P \in E(\mathbb{Z}_p)$. Chacun prendra un scalaire $\alpha, \beta \in \{2, 3, \cdots, \#E - 1\}$ afin de calculer le multiple de P associé qui sera par la suite échangé entre eux, pour qu'ils puissent établir une clé commune [10] comme schématisé dans la Figure [II.12].



$$K = \alpha \cdot \beta P = \alpha\beta \cdot P = \beta\alpha \cdot P = \beta \cdot \alpha P = K$$

Figure II.12 – Échange de clé elliptique de Diffie-Hellman

II.6 Conclusion

Après avoir acquis les bases sur lesquelles repose un cryptosystème conçu par les courbes elliptiques, nous pourrions comprendre les avantages qu'offre un tel système comme la performance malgré la complexité de ses opérations de base, ce qui est un point très fort vis à vis des environnements modernes à sécuriser tel que : les réseaux, les bases de données, le cloud, les objets connectés $\cdots$ qui présentent des contraintes variées, tel que la bande passante et l'énergie, ce qui exige moins de calculs et des clés moins volumineuses sans perte dans le niveau de sécurité.

Dans le chapitre suivant, nous allons abordé une des techniques de la cryptographie moderne qui offre une sécurité aux clés cryptographiques en se basant sur les opérations que les courbes elliptiques mettent à notre portée.

CHAPITRE III

Partage de Secret

Sommaire

III.1 Introduction

La sécurité des cryptosystèmes est un fait qui repose sur des principes. En 1883, *August Kerckhoffs* publia dans son article "La cryptographie militaire" une liste de six exigences connues aujourd'hui sous le nom de *principes de Kerckhoffs*, que doit satisfaire un système de chiffrement pendant un temps illimité. Parmi ces principes, celui considéré comme le plus éminent « Il faut que le système n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi » [1]. En d'autre termes, la sécurité doit reposer sur le secret de la clé et non sur le secret de l'algorithme. Cette clé représente le secret à préserver. Par conséquent, une technique de la cryptographie moderne met à notre disposition une méthode pour assurer un tel niveau de sécurité, il s'agit de répartir le secret sur différentes personnes appelées participants ou candidats et ce après l'avoir partagé en plusieurs parties par un Dealer, de tel sorte qu'il pourrait être reconstitué, une fois suffisamment de parties sont réunies, ce qui résout le problème de perte de la clé et renforce la confidentialité de notre cryptosystème.

III.2 Schémas de partage

Toute technique décrivant une méthode de partage de secret est appelée schéma de partage. Deux catégories de schémas sont à mentionner :

Les schémas N : notés (n, n), où la reconstitution du secret nécessite la totalité des participants à coopérer chacun avec son partage.

Les schémas à seuil : notés (k, n), le secret pourrait être reconstitué si au moins k participants coopèrent, telle que $1 \leq k \leq n$.

III.2.1 Schémas (N,N)

Parmi les méthodes basées sur le schéma N, l'approche naïve, et l'approche du XOR :

III.2.1.1 Méthode naïve

Il s'agit de la technique la plus basique d'un partage, simple approche qui consiste à découper le secret sur autant de parties que de candidats, et pour la reconstitution, il suffit de concaténer tous les morceaux dans le bon ordre (voir Figure [III.1]).

Cette méthode n'est pas sécurisée, son inconvénient réside dans le fait que malgré l'autonomie des partages, chacun d'eux circule une information sur le secret d'origine. Donc, n'importe quelle altération accidentelle ou volontaire pendant l'acheminement ou le traitement des données, pourra facilement affecter la clé reconstituée. En parallèle, une interception divulguera une certaine information sur le secret.

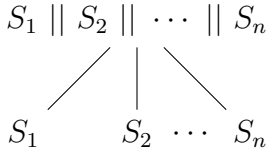


Figure III.1 – Méthode naïve

1. Guillot Philippe, "Auguste Kerckhoffs et la cryptographie militaire", Bibnum. Textes fondateurs de la science, 2013, FMSH-Fondation Maison des sciences de l'homme

Exemple 4: Méthode naïve
Soit le secret $S = CRYPTOGRAPHIE$. Après répartition, on a pu obtenir les partages suivants :

$$S_1 = CRYPT,\ S_2 = OGRA,\ S_3 = PHIE$$

Afin de reconstituer S, il suffit de concaténer toutes les parties dans le bon ordre :

$$S_1 \parallel S_2 \parallel S_3 = CRYPT \parallel OGRA \parallel PHIE = CRYPTOGRAPHIE$$

III.2.1.2 Méthode du XOR

Contrairement à une simple concaténation proposée par la méthode naïve, l'approche du XOR propose une disjonction exclusive entre les différentes parties du secret. Le Dealer génère autant de séquences aléatoires de bits qu'il a de candidats, ayant toutes la même taille du secret à partager, entre lesquelles il effectue une opération XOR (voir Figure [III.2]).

Cette méthode a les mêmes inconvénients que la méthode naïve, même si elle parait présenter plus de sécurité du point de vue théorique. En plus, la perte ou l'altération d'un des partages implique la perte totale du secret.

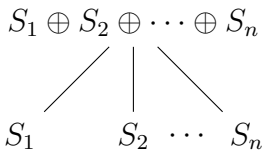


Figure III.2 – Méthode du XOR

Exemple 5: Méthode du XOR
Le Dealer dispose d'un secret S qu'il voudra partager sur 3 participants. Alors il génère 2 séquences de taille $|S|$, P_1, P_2 et afin de générer la troisième, il n'a qu'a effectuer un XOR comme suit :

$$S \oplus P_1 \oplus P_2 = P_3$$

Puis il partage les parties P_1, P_2 et P_3 sur les 3 candidats.

Pour reconstituer le secret, les participants n'ont qu'à coopérer :

$$P_1 \oplus P_2 \oplus P_3 = S$$

Les méthodes de partage basées sur le schéma (n, n) sont rigides mais facile à mettre en œuvre en implémentation. Par contre, ils ne peuvent jamais servir en cas d'une expansion du partage si jamais on aura un jour ou l'autre d'autres participants à rajouter pour le même secret. En plus, ils ne résolvent pas le problème de perte de la clé, car sa reconstitution nécessite la totalité des partages, d'où la nécessité d'un autre schéma présentant plus d'extensibilité et de sécurité.

III.2.2 Les schémas à seuil

Afin de résoudre le problème de perte de la clé secrète, le schéma à seuil apporte des solutions par différentes approches purement mathématiques, où la reconstitution du secret n'exige pas la contribution de la totalité des candidats, mais un certain effectif de participants suffit. Le nombre représentant les participants à coopérer pour cette phase est appelé seuil, qui définit la marge

minimale des partages nécessaires pour l'opération de reconstitution de la clé. Formellement, pour un partage sur n participants, seulement k candidats peuvent reconstituer le secret, tel que $1 \leq k \leq n$.

Outre cela, le schéma à seuil apporte l'option de passage à l'échelle au système de partage, car il permet à tout moment de rajouter d'autres participants sans affecter le système ni refaire les partages déjà effectués.

Un schéma à seuil (k, n) partageant un secret S sur n participants, doit satisfaire les conditions suivantes :

1. la connaissance de k pièces quelconque S_i *ou plus*, facilite le calcul de S ;
2. la connaissance de $k-1$ pièces quelconque S_i *ou moins*, laisse S complètement indéterminé.

Plusieurs schémas ont été proposés dans cette catégories, parmi lesquelles :

III.3 Schéma à seuil de Shamir

Le premier partage de secret par schéma à seuil a été proposé par Adi Shamir en 1979. Basé sur le principe d'interpolation polynomiale de Lagrange dans un plan bidimensionnel, ce schéma a fait l'objet de base de la majorité des travaux de recherche contemporains dans le domaine de partage de secrets cryptographiques.

La notion d'interpolation au sens large, est une méthode d'approximation qui consiste à remplacer une fonction $y = f(x)$ dans un voisinage de points de base $x_0, x_1, \cdots, x_n$ par des fonctions plus simples qui sont des approximations (les meilleurs possibles). En revanche, l'interpolation polynomiale trouve les valeurs d'approximation de ces points de base dans un polynôme. Celle de Lagrange plus précisément, traite la notion de reconstruction d'un polynôme à partir de points de base donnés (voir phase de reconstitution), tel que une fois ce polynôme reconstitué, il sera unique [2].

Le schéma de Shamir partage un secret donné sur un ensemble de n participants, les partages sont représentés par des points d'un polynôme, mais contrairement aux approches vues précédemment, seulement certains k candidats d'entre eux peuvent le reconstituer par interpolation en se servant de k points de base, tel que $1 \leq k \leq n$. La méthode proposé peut être traitée sous deux étapes principales [17] :

III.3.1 Phase de répartition du secret

Soit S un secret à partager sur n personnes en pièces S_i. La première étape consiste à construire un polynôme *aléatoire* $f(x)$ de degrés $k-1$ tel que k représente le seuil minimal exigé de candidats pour la phase de reconstitution :

$$f(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1}$$

avec $a_0 = S$ et $a_1, \cdots, a_{k-1} \in \mathbb{R}$, donc

$$f(x) = S + \sum_{i=1}^{k-1} a_i x^i$$

pour n'importe quel point $(i, f(i))$ du polynôme, on obtient

$$S_1 = f(1), ..., S_i = f(i), ..., S_n = f(n)$$

2. "Petite Encyclopédie des Mathématiques", 1980, Didier, p.680 - 681

Pour un sous ensemble de k points, on pourra déterminer les coefficients du polynôme en le reconstruisant par interpolation polynomiale, et afin de trouver le secret S, il suffit de trouver $f(0)$ car $f(0) = S$.

Exemple 6: Schéma à seuil de Shamir (Préparation des partages)
Soit le secret $S = 1234$, qu'on voudrait partager sur 6 participants ($n = 6$), tel que 3 candidats seulement suffisent pour la reconstruction du secret. Donc, on aura besoin d'un polynôme de degré $k - 1 = 2$ ce qui nécessite 2 coefficients aléatoires, soient 166 et 94

$$a_0 = S = 1234; a_1 = 166; a_2 = 94$$

Ce qui nous donne le polynôme suivant :

$$f(x) = 1234 + 166x + 94x^2$$

En calculant 6 points pour nos partages de la forme $S_i = (i, f(i))$, on obtient

$$\begin{cases} S_1 & = (1, 1494) \\ S_2 & = (2, 1942) \\ S_3 & = (3, 2578) \\ S_4 & = (4, 3402) \\ S_5 & = (5, 4414) \\ S_6 & = (6, 5614) \end{cases}$$

III.3.2 Phase de reconstitution (Interpolation de Lagrange)

Pour la phase de reconstitution du secret partagé par la méthode de Shamir, elle est possible comme suscité par l'interpolation de Lagrange de k points de base, afin de déduire le polynôme initial. Le principe de l'interpolation consiste à déduire une fonction $f(x)$ en partant seulement de quelques points connus. Cette méthode part du principe d'estimation de la valeur de f en un point quelconque $x \in \mathbb{R}$ en construisant un polynôme P tel que $P(x_i) = f(x_i)$ pour $i = 0, \cdots, n$, on obtient par la suite l'approximation $P(x) \approx f(x)$.[12]

On commence par prendre la forme $P_n(x) = L_0(x)y_0 + L_1(x)y_1 + ... + L_n(x)y_n$ donc

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x)$$

tel que L_i est un polynôme de degré au plus n vérifiant l'égalité

$$\forall (i, j) \in [0, n]^2, L_i(x_j) = \delta_{i,j}$$

où $\delta_{i,j}$ est le symbole de *Kronecker* définit par $\delta_{i,j} = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases}$ [13]

$$et\ L_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{k=n} \frac{(x - x_k)}{(x_i - x_k)}$$

Exemple 7: Schéma à seuil de Shamir (Reconstitution du secret)
Reconstruisons le secret de l'exemple précédent depuis 3 partages choisies :

$$\text{soient } \begin{cases}(x_0, y_0) & = (2, 1942)\\ (x_1, y_1) & = (4, 3402)\\ (x_2, y_2) & = (5, 4414)\end{cases}$$

avec ces 3 points, calculons le polynôme de Lagrange :

$$\begin{aligned}L_0 &= \frac{x-x_1}{x_0-x_1}\cdot\frac{x-x_2}{x_0-x_2} = \frac{x-4}{2-4}\cdot\frac{x-5}{2-5} = \frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}\\ L_1 &= \frac{x-x_0}{x_1-x_0}\cdot\frac{x-x_2}{x_1-x_2} = \frac{x-2}{4-2}\cdot\frac{x-5}{4-5} = -\frac{1}{2}x^2 + \frac{7}{2}x - 5\\ L_2 &= \frac{x-x_0}{x_2-x_0}\cdot\frac{x-x_1}{x_2-x_1} = \frac{x-2}{5-2}\cdot\frac{x-4}{5-4} = \frac{1}{3}x^2 - 2x + \frac{8}{3}\end{aligned}$$

$$\begin{aligned}\text{donc } P_n(x) = \sum_{i=0}^{2} y_i L_i(x) &= \left(1942 \times \left(\frac{1}{6}x^2 - \frac{3}{2}x + \frac{10}{3}\right)\right)\\ &+ \left(3402 \times \left(-\frac{1}{2}x^2 + \frac{7}{2}x - 5\right)\right)\\ &+ \left(4414 \times \left(\frac{1}{3}x^2 - 2x + \frac{8}{3}\right)\right)\\ &= 94x^2 + 166x + 1234\end{aligned}$$

III.3.3 Schéma de shamir sur un corps fini

Supposons qu'un attaquant a pu intercepter t points tel que $t < k$, théoriquement il ne devrait pas être en mesure de reconstituer le polynôme pour calculer le secret. Par contre, il pourra largement réduire la taille de l'ensemble des cas possibles à chaque fois que la valeur de t s'approche de celle de k, et ceci en utilisant les informations publiques qu'il tient sur le système de partage et ceux qu'il vient d'intercepter. Avec plusieurs test sur cet ensemble, il sera en mesure de briser la sécurité du système de partage. Ceci est dû au fait que les coefficients ont été choisis sur $\mathbb{R}$.

Afin de remédier à ce problème, Shamir propose d'utiliser l'arithmétique modulaire en prenant des points sur un corps fini, dont le tracé du polynôme sera représenté par un nuage de points disjoints, indépendants et désorganisés au lieu d'une courbe [17]. Plus formellement, il suffit de trouver un nombre premier p tel que $p > S$ et $p > n$, puis de choisir des coefficients uniformément aléatoires $a_1, \cdots, a_{k-1} \in \mathbb{F}_p$, ce qui donne le polynôme

$$f(x) = S + \sum_{i=1}^{k-1} a_i x^i \pmod p$$

puis de calculer chaque $S_i(i, f(i)) \pmod p$.

Exemple 8: Schéma à seuil de Shamir sur un corps fini
Reprenons l'exemple précédent sur un corps fini où $p = 1613$, le polynôme $f(x)$ sera redéfini comme suit :

$$f(x) = 1234 + 166x + 94x^2 \pmod{1613}$$

En recalculant les 6 points $S_i = (i, q(i)) \pmod{1613}$, on obtient

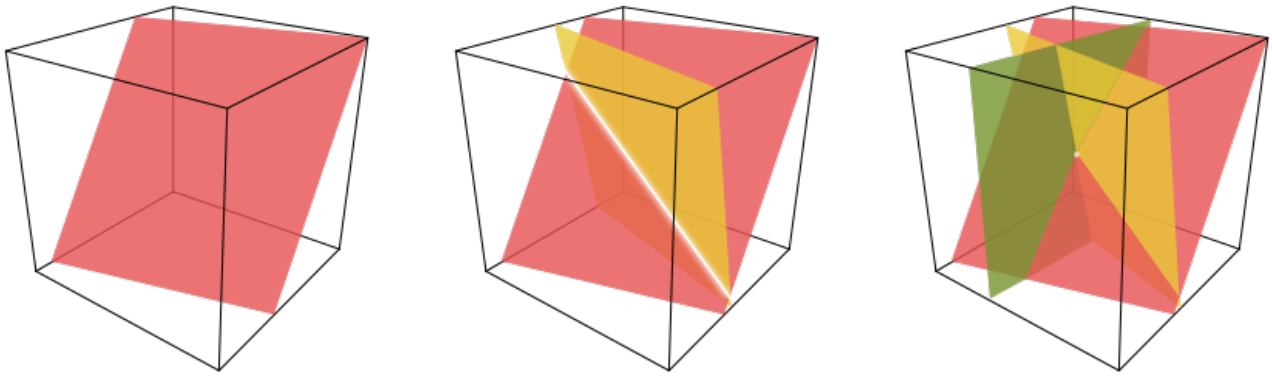
$$\begin{cases} S_1 &= (1, 1494) \\ S_2 &= (2, 329) \\ S_3 &= (3, 965) \\ S_4 &= (4, 176) \\ S_5 &= (5, 1188) \\ S_6 &= (6, 775) \end{cases}$$

Pour la phase de reconstitution, le même procédé est à suivre en calculant le polynôme de Lagrange

$$P_n(x) = \sum_{i=0}^{n} y_i L_i(x) \pmod{p}$$

III.4 Schéma à seuil de Blakley

Le second schéma à seuil publié également en 1979, a été proposé par *G.R.Blakley*, dont la méthode est basée sur la géométrie d'hyperplans et le principe d'intersection des plans non parallèles à un unique point sur un espace donné. Ce point représente le secret à partager sur un certain nombre de participants dont chacun est défini par un plan appelé ombre (*shadow*) Figure III.3.



(a) un seul plan ne révèle aucune information sur le secret répartit

(b) deux plans réduisent l'intervalle de prédictions en une ligne

(c) trois plans se croisent en un point unique qui représente le secret

Figure III.3 – Schéma de Blakley en 3 dimensions [21]

Le seuil k du schéma de partage doit être égal à la dimension de l'espace, où chacun des n plans est défini par une équation linéaire de la forme

$$a_1x_1 + a_2x_2 + \cdots + a_kx_k = b$$

Le secret est obtenu par l'intersection de k plans dont les coordonnées peuvent être représentées sur n'importe quelle équation des plans donnés [3].

III.4.1 Phase de répartition

La méthode commence d'abord par choisir un nombre premier p pour le corps fini $\mathbb{F}_p$. Le Dealer génère un point secret $P \in \mathbb{F}_p$, dont la première coordonnée x_1 est définie suivant la

valeur de la clé, et les valeurs des autres coordonnées sont prises aléatoirement dans le corps. L'utilisateur i obtiendra une équation d'hyperplan

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ik}x_k = y_i$$

par les n équations de degré k, on obtient un système linéaire de $n \times k$. Le Dealer envoi secrètement à chaque candidat i la valeur y_i et publiquement les coefficients $a_{i1}, \cdots, a_{ik}$.

III.4.2 Phase de reconstitution

Pour la reconstitution du secret, elle est effectuée par la résolution du système des équations d'hyperplans des candidats coopérants en nombre égal au seuil, afin de calculer la coordonnées secrète du point d'intersection.[3]

Exemple 9: Schéma de Blakley

Prenons d'abord un nombre premier $p = 37$ et un point secret dans un espace de 3 dimensions $q(x_0, y_0, z_0)$ tel que x_0 représentera notre clé à partager, et $y_0, z_0 \in \mathbb{F}_{37}$.

Soit $q(42, 29, 57)$ qu'on souhaiterait répartir sur 5 participants avec un seuil= 3, tel que chacun des candidats recevra l'équation d'un plan comme suit :

$$\begin{aligned} A : z &= 4x + 19y + 68 \\ B : z &= 52x + 27y + 10 \\ C : z &= 36x + 65y + 18 \\ D : z &= 57x + 12y + 16 \\ E : z &= 34x + 19y + 49 \end{aligned}$$

Afin de reconstituer le secret q, 3 candidats forment un système matriciel de la forme :

$$\begin{pmatrix} a_1 & b_1 & -1 \\ a_2 & b_2 & -1 \\ a_3 & b_3 & -1 \end{pmatrix} \times \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \equiv \begin{pmatrix} -c_1 \\ -c_2 \\ -c_3 \end{pmatrix} \pmod{p}$$

Prenons par exemple les candidats A, B et C

$$\begin{pmatrix} 4 & 19 & -1 \\ 52 & 27 & -1 \\ 36 & 65 & -1 \end{pmatrix} \times \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \equiv \begin{pmatrix} -68 \\ -10 \\ -18 \end{pmatrix} \pmod{73}$$

Tant que le déterminant de la matrice est non nul, sa matrice inverse existera toujours, ce qui permet de calculer le secret.

Après avoir résolu le système, le secret pourrait facilement être reconstitué $(42, 29, 57)$ donc $secret = 42$.

Ce schéma ne présente pas une grande efficacité du point de vue sécurité, car chacun des participants saura toujours que le secret réside dans son plan, ce qui implique la nécessité d'un corps suffisamment grand afin de remédier à toute attaque exhaustive.

III.5 Schéma à seuil basé sur les restes chinois

Vers le troisième siècle, un mathématicien chinois *Sun Tsu* est parvenu à résoudre un système de congruences linéaires de la forme [10] :

$$\begin{cases} x \equiv & a_1 \ (mod\ m_1) \\ x \equiv & a_2 \ (mod\ m_2) \\ & \vdots \\ x \equiv & a_k \ (mod\ m_k) \end{cases}$$

en démontrant que pour un tel système, la solution $x \pmod M$ est unique, ce qui fournit une méthode pour déterminer un nombre secret S modulo k entiers relativement premiers entre eux $m_1, \cdots, m_k$, d'où l'idée du schéma à seuil basé sur les restes chinois qu'a proposé *Mignotte* et *Asmuth-Bloom* en déterminant un message secret S répartit, tel que chaque participant ne pourrait posséder qu'un seul reste $\pmod{m_i}$. Le secret ne pourra être reconstruit que si toutes les parties candidates seront rassemblées, et chaque partie ne révèle aucune donnée sur le secret.

III.5.1 Phase de répartition

La préparation des clés à répartir commence d'abord par le choix d'un secret $S = x$ à partager, ensuite on calcule pour chaque candidat i des n participants sa clé $s_i \pmod{m_i}$, donc pour les k candidats il résulte le système suivant :

$$\begin{cases} x \equiv & s_1 \pmod{m_1} \\ x \equiv & s_2 \pmod{m_2} \\ & \vdots \\ x \equiv & s_k \pmod{m_k} \end{cases}$$

tel que $m_1, m_2, \cdots, m_k$ sont des entiers positifs premiers entre eux, et $s_i \in \mathbb{Z}$;
donc, le système suscité possède une solution unique dans $\mathbb{Z}/n\mathbb{Z}$.

III.5.2 Phase de reconstitution

Afin de reconstituer le secret, en utilisant le théorème des restes chinois, les k parties candidates seront rassemblées sous forme de système de congruences linéaires tel que :

$$M = m_1 \times m_2 \times \cdots \times m_k$$

alors

$$x \equiv s_1 M_1 y_1 + s_2 M_2 y_2 + \cdots + s_k M_k y_k$$

avec

$$M_i = M/m_i \qquad y_i M_i \equiv 1 \pmod{m_i}$$

Exemple 10: Schéma par méthode des restes chinois

$$\begin{cases} x \equiv 1 \pmod 3 \\ x \equiv 2 \pmod 5 \\ x \equiv 3 \pmod 7 \end{cases}$$

on calcule $M = 3 \times 5 \times 7 = 105$

donc

$$\begin{cases} M_1 = 105/3 = 35 & y_1 \times 35 \equiv 1 \pmod 3 & y_1 = 2 \\ M_2 = 105/5 = 21 & y_2 \times 21 \equiv 1 \pmod 5 & y_2 = 1 \\ M_3 = 105/7 = 15 & y_3 \times 15 \equiv 1 \pmod 7 & y_3 = 1 \end{cases}$$

$$x \equiv 1 \times 35 \times 2 + 2 \times 21 \times 1 + 3 \times 15 \times 1 \equiv 157 \equiv 52 \pmod{105}$$

III.6 Options de partage de secret

Depuis la parution de la technique de partage de secret cryptographique, une variété de recherche ont contribué à l'améliorer en ajoutant plus d'options aux méthodes d'origine, produisant par la suite différents protocoles cryptographiques dans le domaine. Dans ce qui suit, nous abordons les plus intéressants de ces protocoles appliqués au schéma de Shamir.

III.6.1 Partage de secret pro-actif

La répartition des partages nécessite toujours une communication à distance en faisant appel aux différentes techniques de réseau, ce qui les expose à une diversité de risques : interception, altération volontaire ou involontaire, destruction ..., cette situation a mené les chercheurs à proposer de réduire la validité des partages à un certain temps avant de les renouveler.

Le partage de secret pro-actif, est une méthode qui permet de limiter le temps de validité du partage, en mettant à jour périodiquement les clés partagés [5]. Le terme pro-actif fait référence au fait qu'il n'est pas nécessaire qu'une violation de la sécurité se produise avant que les secrets ne soient rafraîchis, (rafraîchissement périodiquement donc de manière pro-active).

Un partage de secret pro-actif doit respecter les propriétés suivantes :

— Renouveler les partages existants sans changer le secret, de sorte que l'exposition des actions ne nuira pas au secret (seulement les anciens partages deviennent inutiles).

— Récupérer les partages perdus ou corrompus sans compromettre leur secret.

Cela devrait être effectué bien sûr, en évitant toute fuite d'informations ou altération du secret [5].

La procédure de mise à jour des partages passe par plusieurs étapes que nous exposons comme suite :

(i) Le Dealer génère un polynôme aléatoire de degré $k-1$ tel que k représente le seuil et $f(0) = secret$.

(ii) Chaque participant i reçoit son partage $x_i^0 = f^0(i)$ où $i \in 1, \ldots, n$ tel que n est le nombre des participants et x_i^0 est le partage du candidat i dans le temps 0.

(iii) La reconstitution se fait toujours par interpolation des k clés candidates.

(iv) Afin de mettre à jour les partages, chaque participant calcule un polynôme aléatoire de la forme $\delta_i(z) = \delta_{(i,1)}z^1 + \delta_{(i,2)}z^2 + \cdots + \delta_{(i,k)}z^k$

(v) Chaque participant i envoi à ses confrères j la valeur $u_j^i = \delta_i(j)$ associée

(vi) Chaque participant mettra à jour son partage en calculant $x_i^{t+1} = x_i^t + u_{1,i}^t + \cdots + u_{n,i}^t$

Ces étapes sont à refaire pour chaque intervalle t de temps de validité des partages.

Exemple 11: Partage de secret pro-actif
Soit le système de partage suivant :

— secret : $x = 6$

— nombre de participants : 2

— seuil : 2

— corps : $\mathbb{Z}_{11}$

— exposant 0 pour t_0

Tout d'abord, nous générerons un polynôme aléatoire dans $\mathbb{Z}_{11}$ de degré $seuil - 1 = 1$, soit le polynôme $f^0(x) = 6 + 2x$ (noter que $f^0(0) = secret = 6$).

Les participants vont donc recevoir :

- participant 1 : $x_1^0 = f^0(1) = 6 + 2 \times 1 = 8 \pmod{11}$
- participant 2 : $x_2^0 = f^0(2) = 6 + 2 \times 4 = 10 \pmod{11}$

Afin de mettre à jour ce partitionnement, les participants doivent construire un polynôme aléatoire de degré 1

- participant 1 : $\delta_1^0(z) = \delta_{1,1}^0 \times z^1 = 2 \times z^1$
- participant 2 : $\delta_2^0(z) = \delta_{2,1}^0 \times z^1 = 3 \times z^1$

Chaque participant doit évaluer son polynôme et partager des informations avec les autres :

- participant 1 calcule : $u_{1,1}^0 = \delta_1^0(1) = 2$ et $u_{1,2}^0 = \delta_1^0(2) = 4 \pmod{11}$, et envoi $u_{1,2}^0$ au participant 2.
- participant 2 calcule : $u_{2,1}^0 = \delta_2^0(1) = 3$ et $u_{2,2}^0 = \delta_2^0(2) = 6 \pmod{11}$, et envoi $u_{2,1}^0$ au participant 1.

Les participants, chacun effectue une mise à jour de sa partition en calculant :
$x_i^1 = x_i^0 + u_{1,i}^0 + u_{2,i}^0$

- participant 1 calcule : $x_1^1 = x_1^0 + u_{1,1}^0 + u_{2,1}^0 = 8 + 2 + 3 = 2 \pmod{11}$
- participant 2 calcule : $x_2^1 = x_2^0 + u_{1,2}^0 + u_{2,2}^0 = 10 + 4 + 6 = 9 \pmod{11}$

Pour reconstruire le polynôme $f^1(x)$, on utilise l'interpolation avec x_1^1 et x_2^1

III.6.2 Partage de secret vérifiable

La notion de partage vérifiable est une notion traitant l'intégrité du système et son authenticité, et non pas sa confidentialité, car elle permet de vérifier la validité du partage, puisqu'il peut toujours y avoir des actions incohérentes lors de la répartition du secret ainsi que la distribution. Afin d'empêcher ce comportement malveillant, il est primordiale de proposer un algorithme de vérification de la cohérence du partage pour chaque candidat.

Il est important de noter qu'une telle vérification, ne veut nullement dire une comparaison du secret avec sa reconstitution, car ceci implique la divulgation de notre secret, l'opération de vérification opère juste sur le partage lui même avant la phase de reconstitution du secret.

Il existe deux types de protocoles de partage de secrets vérifiables :

III.6.2.1 Preuve interactive

Représenté pour la première fois par le protocole de *Benaloh* en 1986, c'est un algorithme qui permet aux participants de vérifier l'intégrité et l'authenticité du Dealer (serveur de partage par exemple) sans qu'aucun d'entre eux ne soit tricheur. Dans ce schéma, les participants peuvent

communiquer entre eux lors de la procédure de validation. Par contre, le point faible de cette méthode, c'est qu'en cas de triche, un conflit peut y avoir lieu de sorte qu'il sera impossible de déterminer l'identité du tricheur, d'où l'exigence suscitée sur la nature des acteurs [5].

Pour cette raison, d'autre protocoles interactifs n'exigeant pas l'honnêteté des acteurs. Ces protocoles font appel aux techniques asymétriques de chiffrement/déchiffrement et échange de clé de *Diffie-Hellman*.

III.6.2.2 Preuve non interactive

La première méthode a été présentée par le schéma de *Feldman*, ce protocole est basé sur le fait d'ajouter une information supplémentaire envoyée à chaque participant lui permettant de vérifier son partage du secret. Contrairement au schéma précédent, dans ce protocole aucune communication ne doit avoir lieu entre les différents candidats. Cette méthode utilise des techniques d'échange de clé basées sur le protocole de *Diffie-Hellman* .

III.6.3 Structures d'accès

La notion de structure d'accès discute l'autorisation d'accès d'un groupe à un système, les candidats auxquelles l'accès est accordés sont des parties qualifiées, qui forment la structure d'accès. Dans le domaine de partage de secret, la structure d'accès à un secret partagé est une forme généralisée du schéma à seuil (k, n), elle représente tous les sous ensembles de participants ayant droit d'accéder à la reconstitution du secret.

Soit la structure d'accès d'un schéma de partage de secret Γ. Les éléments de la structure d'accès sont appelés groupes autorisés et les autres sont appelés groupes non autorisés. L'ensemble de tous les groupes non autorisés est appelé la structure adversaire. Cette dernière est notée $\overline{\Gamma}$. A titre d'exemple, pour une structure d'accès de seuil (k, n) [2] :

$$\Gamma = \{A \in 2^{\mathcal{P}} : |A| \geq k\};$$
$$\overline{\Gamma} = \{A \in 2^{\mathcal{P}} : |A| < k\};$$

où $2^{\mathcal{P}}$ est l'ensemble des groupes de participants $\mathcal{P} = \{1, 2, \cdots, n\}$.

Pour chaque ensemble A qualifié, n'importe quel ensemble contenant A sera également qualifié et pourra reconstruire le secret. En outre, si un ensemble A n'est pas qualifié, tout sous ensemble de A ne sera pas qualifié et ne pourra jamais reconstruire le secret.

$$(A \in \Gamma) \wedge (A \subseteq B) \Rightarrow B \in \Gamma \tag{III.1}$$
$$(A \in \overline{\Gamma}) \wedge (B \subseteq A) \Rightarrow B \in \overline{\Gamma} \tag{III.2}$$

La notion d'une structure *monotone* peut être déduite si les deux équations (III.1) et (III.2) sont toutes deux satisfaites [2].

Cette notion va permettre de définir un ou des sous ensembles finis et spécifiques de participants ayant la possibilité de reconstruire le secret d'une manière unique.

III.6.4 Partage de secret multiple

Un partage de secret multiple traite les cas où le secret est défini comme une information composées, ce qui implique qu'au lieu de partager une seule donnée, on est obligé de partager plusieurs valeurs.

Deux approches sont à noter pour ce schéma [1] :

- le cas le plus basique où tous les membres appartenant à la structure d'accès, peuvent reconstituer tous les secrets partagés par le même partage, parallèlement pour l'ensemble des candidats non tolérés, ils ne seront en mesure de reconstituer aucun des secrets.
- Pour la seconde approche, il s'agit du cas où la reconstitution des secrets se fait en groupe par un certain ensemble candidats qualifiés au lieu de la totalité, de tel sorte d'associer à chaque groupe, un secret ou un ensemble de secrets à reconstituer, et la collaboration des groupes donnera le résultat final pour la totalité des clés.

III.7 Conclusion

Dans ce chapitre nous avons entamé la notion du secret réparti appliquée dans la protection des clés cryptographiques, en présentant les deux catégories de protocoles de partage de secret existantes.

Pour chaque catégorie, nous avons cité quelques schémas les plus connus dans le domaine, en détaillant formellement les techniques utilisées pour la préparation des clés réparties et l'opération de partage, ainsi que celles exploitées pour la reconstitution du secret d'origine, suite à quoi nous avons pu remarquer les faiblesses de quelques schémas et la force d'autres.

Comme pour n'importe quel cryptosystème, non seulement la sécurité du schéma de partage est importante, mais également la robustesse de l'algorithme implémenté pour ce schéma, que ce soit du point de vu complexité ou temps d'exécution, qui sont des facteurs très importants pour tout programme en cryptographie. Les travaux de recherche modernes ont tous prouvé l'efficacité des courbes elliptiques dans le domaine de sécurité informatique, sur quoi on a basé toute notre étude.

CHAPITRE IV

État de l'art

Sommaire

IV.1 Introduction

Une variété de travaux de recherche a été effectué sur les méthodes de partage de secret cryptographique depuis la parution de l'article de *Shamir* en 1979. Depuis lors, plusieurs protocoles de partage ont vu le jour, notamment ceux basés sur les courbes elliptiques, visant à minimiser le temps d'exécution des algorithmes tout en réduisant la taille des clés et sans négliger le niveau de sécurité des systèmes établis, que ce soit du point de vue confidentialité ou intégrité du partage.

Dans cet état de l'art, nous présentons plusieurs approches apportant chacune une solution différente à la technique de partage de secret par les courbes elliptiques, qu'on a proposé d'exposer suivant les principaux protocoles abordés.

IV.2 Protocoles d'auto-sélection

Les protocoles d'auto sélection incluent la notion de clé privée/publique au schéma de base de *Shamir*, de telle sorte que chacun des acteurs Dealer/Participant sélectionne une clé privée (gardée secrète) par laquelle il génère une seconde clé rendue publique qui sert dans l'envoi du partage. Donc contrairement au schéma de *Shamir* qui n'impliquait les candidats que dans la phase de reconstitution du secret, dans les schémas d'auto sélection ces derniers jouent un rôle primordial notamment dans la phase de répartition.

IV.2.1 Wang Caimei *et al.* 2009

Wang Caimei *et al.* proposent une méthode basée sur le schéma de shamir avec auto sélection des clés privées, et en ajoutant une couche de sécurité par le chiffrement des communications, effectué par une autre paire de clés générée à l'avance par un Centre d'Authentification (CA) qui jouera le rôle du Dealer [4].

Phase d'initialisation : Afin de partager une clé K sur m candidats, le CA va chiffrer les communications avec sa clé SK_{CA} après avoir rendu publique sa paire PK_{CA} pour les déchiffrements, il génère puis publie les paramètres initiaux du système, cités comme suit :

- Une courbe elliptique sur un corps fini $E(\mathbb{F}_q)$ avec q premier et très grand ;
- Un générateur $G \in E(\mathbb{F}_q)$ d'ordre n ;
- Un polynôme décomposable de la forme $g(x) = g_1(x)g_2(x)\cdots g_k(x)$ avec $g_i(x)$ élémentaire (indécomposable) ;
- Une fonction de hachage $H(x)$ cryptographique ;

Phase de répartition des partages : Chacun des candidats génère un scalaire aléatoire s_i et calcule $P_i = s_i \cdot G$ qu'il envoi au CA après chiffrement $(P_i)_{PK_{CA}}$. Ce dernier de son côté choisi aléatoirement sa propre clé privée r et génère le polynôme de Shamir $f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \cdots + a_{t-1} \cdot x^{t-1} \pmod{g(x)}$ avec $a_0 = K$. En se servant de sa clé privée, le CA calcule des points secrets $r \cdot P_i = (X_i, Y_i) \pmod{g(x)}$ et génère les couples $(X_i, f(X_i))$ qui représentent les partages qu'il distribue $y_i = f(X_i)$ avec un point publique $R = r \cdot G \pmod{g(x)}$. En outre des partages, le CA publie une donnée servant comme information de validation de son authenticité $H_i = H(rP_i)$ pour l'ensemble des candidats.

Phase de reconstitution du secret : En se servant des données publiques et le partage réceptionné, chaque ensemble de candidats atteignant le seuil t pourront reconstituer le secret en passant par les opérations suivantes :

- Reconstruire le point secret pour récupérer son abscisse $s_i \cdot R = s_i \cdot r \cdot G = r \cdot Pi$;

— Générer le polynôme de Lagrange par

$$P(x) = \sum_{i=0}^{t} y_i \prod_{\substack{j=0 \\ j \neq i}}^{t} \frac{x - X_j}{X_i - X_j}$$

IV.3 Protocoles de partage multiple

Les protocoles de partage multiple proposent des solutions pour un partage de secret composé, dont la structure se trouve plus complexe qu'un simple chiffre ou une chaine de caractères. Par conséquent, une variété de types de fichiers pourront être vus comme étant un secret, tels que les images, les vidéos, les fichiers sonores,etc.

IV.3.1 Hua Sun *et al.* 2010

Hua Sun *et al.* proposent une approche de partage de secret multiple avec structures d'accès générales et auto sélection des clés privées, ce qui réduit le nombre de communications effectuées entre les différents acteurs [11].

Phase d'initialisation : Le Dealer prépare les paramètres de base pour le système de partage dans cette étape et les rend publiques. Ces données sont : le corps $E(\mathbb{Z}_p^*)$, le point générateur $P \in E(\mathbb{Z}_p)$, ainsi qu'une fonction de hachage cryptographique H. Les n participants par la suite, chacun d'eux sélectionne sa propre clé privée aléatoirement $s_i \in \mathbb{Z}_p^*$ pour générer une autre clé publique par ECDLP $R_i = s_i \cdot P$ qu'il envoi au Dealer. Ce dernier s'assure de la distinction des n clés publiques sinon il demande aux candidats concernés d'en générer d'autres. Après avoir collecté l'ensemble des clés, le Dealer génère un identifiant $u_i \in \mathbb{Z}_p^*$ pour chaque participant et publie les paires (u_i, R_i).

Phase de distribution des partages : L'approche propose une méthode de partage de m secrets $S_1, \cdot, S_m$ dont chacun sert de coefficient pour la construction du polynôme de partage suivant le schéma de Shamir $f(x) = a + S_1 \cdot x + S_2 \cdot x^2 + \cdots + S_m \cdot x^m \pmod p$ tel que a est un coefficient aléatoire dans le même corps. Le Dealer calcule m couples $(i, f(i))$ avec $i = 1, 2, \cdots, m$ puis génère sa propre clé privée $s_0 \in \mathbb{Z}_p^*$ avec laquelle il calcule le point publique $R_0 = s_0 \cdot P$ ainsi que n points secrets $R'_i = s_0 \cdot R_i$, il hache ces derniers $f_i = f(d_i) \oplus h(R'_{1i} \oplus R'_{2i} \oplus \cdots \oplus R'_{di})$ et forme avec un message :

$$MSG = (f(1), f(2), \cdots, f(m), d_1, d_2, \cdots, d_t, f_1, f_2, \cdots, f_t)$$

tel que $d_1, \cdots d_t \in \mathbb{Z}_p$ représentent les identificateurs des structures d'accès qualifiés. Le message sera rendu publique.

Phase de reconstitution du secret : Dans cette phase, chaque participant pourra trouver la clé R' adéquate en ayant la clé publique R_0 par la formule $R'_i = s_0 \cdot R_i = s_0 \cdot s_i \cdot P = s_i \cdot R_0$ qu'il délivre au Combiner. Ce dernier ayant collecté les j clés de la structure d'accès qualifiée, sera en mesure d'accéder au contenu du message haché afin de récupérer les paires $(i, f(i))$ pour pouvoir générer le polynôme de Lagrange et reconstituer le secret par :

$$f(x) = \sum_{i=1}^{m+1} Y_i \prod_{\substack{j=1 \\ j \neq i}}^{m+1} \frac{x - X_j}{X_i - X_j}$$

IV.3.2 Hisham Dahshan *et al.* 2011

L'approche proposée traite la notion de gestion de clés sur un réseau ad hoc de nœuds mobiles, la particularité des réseaux MANETs est l'absence d'infrastructure, par la suite l'absence d'un nœud central qui pourrait représenter le Dealer, c'est la raison pour laquelle les chercheurs ont introduit un centre d'authentification (CA) comme une partie tierce de confiance dont le rôle est limité à l'initialisation des partages et des pairs de clé (publique PK_i et privée SK_i) à long terme en mode hors ligne. Après le déploiement des nœuds sur le réseau (mode en ligne) les nœuds participants à la session établissent eux même leurs propres paires de clés et effectuent leur propre partage. Le protocole de routage exploité dans l'expérimentation est l'AODV (Ad hoc On demand Distance Vector) [6].

Les chercheurs ont décomposé leur approche en plusieurs étapes comme suit :

Phase d'initialisation : Elle consiste à la préparation des paramètres nécessaires pour le partage. Une courbe elliptique E avec les coefficients a et b pris dans un corps fini $\mathbb{F}_q$ (tel que q est premier) sont à choisir aléatoirement. En outre, un point générateur G du groupe cyclique $GF(q)$ de la courbe.

Phase de préparation des partages : Les n nœuds sont distingués par leur identifiant u_i. La méthode propose un partage de B secrets S_{kh} réarrangés sous forme matricielle de taille $m \times d$ ($B = md$) choisis par le CA. Ce dernier génère un polynôme aléatoire $f_{kh}(x)$ pour chaque secret suivant le schéma de *Shamir* afin de calculer les partages associés pour chaque nœud qui va être représenté par une matrice de même taille.

Les deux dernières phases sont effectuées hors ligne. Par contre, les phases qui suivent seront accomplies après déploiement des nœuds sur le réseau.

Phase de génération des clés de session : Chaque fois que L nœuds parmi ceux déployés veuillent établir une session Figure [IV.1], le nœud u_l ayant la dernière identité dans la session diffuse une requête AODV REQ [1]

$$\{REQ, Nonce, k, u_1, \cdots, u_L, TTL\}_{SK_l}$$

signée par sa clé privée SK_l distribuée en mode hors ligne, tel que :

- $Nonce$: compteur aléatoire utilisé pour caractériser l'identité des nœuds dans chaque session ;
- k : nombre minimum des partages que doit choisir chaque nœud depuis sa matrice ;
- $u_1, \cdots, u_L$: représentent les identités des nœuds membres de la session ;
- TTL : temps de vie du message REQ ;

Dès réception de la requête, le nœud en question u_j vérifie la signature via la clé publique PK_l, une fois authentifiée, chaque nœud membre effectue un hachage récursive de k fois des identifiants avec le $Nonce$ pour obtenir des indices R_i/C_i (pour ligne/colonne)

$$\begin{aligned} R_i &= H^i(u_1|u_2|\cdots|u_L|Nonce) \pmod m \\ C_i &= H^i(Nonce) \pmod d \end{aligned}$$

suivant lesquels il établi un polynôme de *Shamir* sur sa matrice de partages initiale pour un second partage

$$f^{(j)}(x) = s^{(j)}_{R_1C_1} + \sum_{i=2}^{k} s^{(j)}_{R_iC_i} \cdot x^{i-1} \pmod p$$

1. Route Request : Message de demande de route comportant {adresse source, n° séquence source, identificateur de diffusion, adresse destination, n° séquence destination, nombre de sauts}

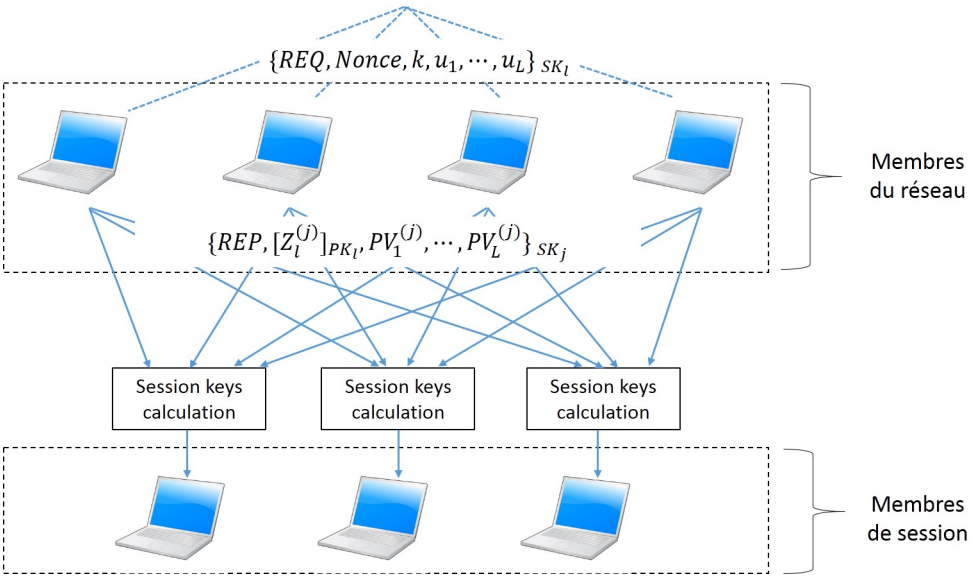


FIGURE IV.1 – Génération des clés de session [6]

avec lequel ce nœud calcule le partage pour chacun de ses voisins de session $Z_l^{(j)} = f^{(j)}(u_l)$ ainsi qu'un point publique qui servira comme information supplémentaire pour la vérification $PV_l^{(j)} = Z_l^{(j)} \cdot G$. Suite à quoi il établit une réponse AODV REP [2]

$$\{REP, [Z_l^{(j)}]_{PK_l}, PV_1^{(j)}, \cdots, PV_L^{(j)}\}_{SK_j}$$

qu'il signe par sa clé privée SK_j. En outre, un point $Pval_i^{(j)} = (a_i^{(j)} \cdot G)$ sera diffusé.

Phase de reconstitution du secret : Le nœud u_l qui jouera également le rôle d'un Combiner, collecte t messages REP représentant le seuil pour reconstituer la clé de session par

$$sk_l = \sum_{j=1}^{t} Z_l^{(j)} \prod_{\substack{i=1 \\ i \neq j}}^{t} \frac{i}{i-j} \pmod p$$

calcule la clé publique associée $pk_l = sk_l \cdot G$ afin de reconstituer le secret initiale

$$SS_{pk} = \sum_{l=1}^{k} pk_l \prod_{\substack{i=1 \\ i \neq l}}^{k} \frac{i}{i-l} \pmod p$$

Vérification des partages : Chaque nœud u_l d'une session sera en mesure de vérifier les seconds partages reçus des voisins par la formule $PV_l^{(j)} = \sum_{i=0}^{t} (Pval_i^{(j)} \cdot u_l^i)$.

Mise à jour des partages : Afin de rafraichir un partage $s_{kh}^{(i)}$, chacun des n nœuds prend un polynôme aléatoire de degré $t-1$:

$$\hat{f}^{(i)}(z) = \hat{a}_1^{(i)} z + \cdots + \hat{a}_{t-1}^{(i)} z^{t-1}$$

Il calcule par la suite la valeur du nouveau partage $\hat{s}_j^{(i)} = \hat{f}^{(i)}(u_j) \pmod p$ et l'envoi au nœuds de la session concernée après l'avoir chiffré avec la clé PK_j. Non seulement le

2. Route Reply : Message de réponse à un message REQ comportant {adresse source, adresse destination, $n°$ séquence destination, nombre de sauts, Temps de vie du message}

partage doit être mis à jour, mais également l'information pour la vérification qui sera diffusée $\hat{A}_l^{(i)} = \hat{a}_l^{(i)} \cdot G$. La nouvelle formule de vérification des partages est donnée par :

$$\hat{s}_j^{(i)} \cdot G = \sum_{l=1}^{t-1} (u_j^l \cdot \hat{A}_l^{(i)})$$

Après avoir reçu les nouveaux partages, chaque nœud met à jour les éléments de sa matrice par :

$$\tilde{s}_{kh}^{(i)} = s_{kh}^{(i)} + \sum_{l=1}^{n} \hat{s}_i^{(l)}$$

IV.4 Protocoles de vérification des partages

Les protocoles de vérification de partage ajoute une option d'authentification à un système de partage qui permet de détecter un intrus dans le système, elle permet également au acteurs communiquant dans le système de s'assurer de l'intégrité des partages.

IV.4.1 Han Yiliang *et al.* 2003

Han *et al.* proposent un cryptosystème de partage vérifiable avec signature numérique et chiffrement/déchiffrement de message basé sur les courbes elliptiques. Les deux algorithmes supplémentaires utilisés sont :

- ECSA : (*Elliptic Curve Signature Algorithm*) un algorithme de signature numérique permettant d'assurer l'intégrité des partages à base du problème de logarithme discret elliptique.
- ECES : (*Elliptic Curve Encryption Scheme*) un algorithme de chiffrement/déchiffrement d'un message via la clé secrète partagée, basé également sur le problème du logarithme discret elliptique.

Les chercheurs ont divisé leur approche en 3 phases [9] :

Phase de répartition du secret : Dans cette étape, un certain secret d est répartit entre n participants dont t d'entre eux forment un groupe qualifié pour l'étape de reconstitution. Un Dealer initialise le système de partage en publiant les paramètres de base et en construisant le polynôme de partage suivant le schéma de *Shamir*. Il calcule les partages $t_i = f(i)$ et les envoi aux candidats. Pour la vérification, Han *et al.* se sont basés sur une méthode de vérification où le Dealer calcule t points $F_j = f_j \cdot G$ et les diffuse sur l'ensemble des candidats pour que chacun d'eux sera en mesure de vérifier l'exactitude des partages par la formule

$$t_i \cdot G = \sum_{j=0}^{t-1} i^j \cdot F_j$$

Phase de vérification avec signature numérique : Pour cette phase, les chercheurs proposent une vérification par signature numérique à base de ECDLP, où un certain utilisateur U ayant un message m signé par un groupe de candidats formant le seuil est capable de vérifier l'authenticité de candidature des membres par leur signature numérique. Pour cela, le Dealer choisit t membres, chacun calcule une clé privée

$$d_i = t_i \prod_{\substack{h=1 \\ h \neq i}}^{t} \frac{h}{h-i}$$

ainsi qu'une autre publique $Q_i = d_i \cdot G$ envoyé au Dealer, et un point $R_i = k_i \cdot G$ avec $k_i \in \{1, 2, \cdots, l-1\}$ aléatoirement choisis, ce point sera diffusé à la totalité des participants pour pouvoir par la suite calculer $R = \sum_{j=1}^{t} R_j$. Le Dealer reçoit les signatures (r, s_i) calculées comme suit :

$$r = (x_R + m) \pmod q$$
$$s_i = (k_i - d_i \cdot r) \pmod l$$

Il vérifie chaque signature par la formule $R_j = s_i \cdot G + r \cdot Q_j$, si le test échoue la candidature du participant j sera rejetée. Après les avoir accepté, Il calcule la signature numérique du système (r, s) avec $s = \sum_{i=1}^{t} s_i$. L'utilisateur qui voudrait vérifier la signature, calcule $R' = s \cdot G + r \cdot Q$ et $r' = (x'_R + mp)$ et vérifie si $r = r'$.

Phase de chiffrement vérifiable : Cette partie donne à l'utilisateur l'opportunité d'envoyer un message m chiffré d'une manière collective. Il choisi $k \in \{1, 2, \cdots, l-1\}$ et génère le point $P = k \cdot G$ ainsi que $k \cdot Q = (x_2, y_2)$ par lequel il chiffre par $c = mx_2$. Le Dealer reçoit (P, c). Pour la partie déchiffrement, le Dealer prend t membres qui composent le groupe qualifié et leur envoi P, chacun d'eux ayant sa clé et son partage pourra par reconstitution du secret calculer $d_i P$ et renvoyer le résultat au Dealer pour le déchiffrement

$$(x_2, y_2) = \sum_{i=1}^{t} d_i P$$

par conséquent, le message en claire pourrait être calculé par $m = cx_2^{-1}$.

IV.4.2 Nisha Patel *et al.* 2016

Patel *et al.* proposent une approche de partage de secret composé en utilisant les courbes elliptiques et en se servant de l'algorithme double Knapsack [3] qui leur permet de sécuriser les communications. Les chercheurs ont exposé leur approche sous forme de 3 phases [14] :

Phase d'initialisation et génération des partages : Le schéma de Patel *et al.* est initialisé par un Dealer qui choisit un grand premier p comme base de construction d'un corps elliptique $E(\mathbb{F}_p)$ dans lequel il détermine un point de base G d'ordre premier q, qu'il transmet publiquement à la totalité des participants. Á leur tour, les n candidats choisissent aléatoirement et indépendamment une clé privée $x_j \in \mathbb{Z}_q^*$ et calculent $R_j = x_j \cdot G \pmod p$ qu'ils transmettent au Dealer. Ce dernier vérifie la distinction entre les différents points reçus $R_j \neq R_i$ sinon il réclame d'autres. Après avoir collecté n points, il attribut à chacun un identifiant et publie les paires (R_j, ID_j).

Phase de Construction du secret et distribution : Afin de construire le polynôme de partage suivant le schéma de *Shamir*, le Dealer prend un nombre aléatoire $b_0 \in \mathbb{Z}_p^*$ et forme $f(x) = b_0 + S_1 x + \cdots + S_k x^k \pmod p$ tel que $S = \{S1, \cdots, S_k\}$ est l'ensemble des secrets à partager, suite à quoi il calcule les images $f(i)$ avec $i = 1, 2, \cdots, k$. Il calcule également le point $R_0 = x_0 \cdot G \pmod p$ et les points $I_j = x_0 \cdot R_j \pmod p$ tel que x_0 est un scalaire aléatoire pris dans le corps fini $\mathbb{Z}_p^*$ ayant comme inverse x_0^{-1} pour lequel il détermine le chiffré y_0 avec double Knapsack, puis publie R_0 et transmet y_0 au Combiner.

Pour la distribution des partages, le Dealer détermine un groupe qualifié $\mathcal{A}$ composé de t participants, et pour chaque paire des coordonnées des points I_t, il prend le minimum

3. Knapsack : Problème du sac à dos, un des problèmes d'optimisation combinatoire utilisé comme algorithme de chiffrement

entre (x, y) et effectue un XOR entre les valeurs prises $d = x_1 \oplus y_2 \oplus \cdots \oplus x_t \pmod p$ et calcule son image $f(d)$. Pour une transmission publique sûre, le Dealer chiffre les identités des candidats qualifiés ainsi que la valeur de $f(d)$ et envoi $\mathcal{A}', f(d)'$ avec les images $f(1), \cdots, f(k)$.

Phase de vérification et reconstitution du secret : Chaque participant envoie au Combiner le chiffré d'un point $W_j = x_j \cdot R_0$. Ce dernier après déchiffrement vérifie si $x_0^{-1} \cdot W_j = R_j$ pour confirmer l'authenticité des candidats avant la reconstitution des secrets qui sera possible par la formule

$$f(x) = \sum_{i=1}^{k+1} Y_i \prod_{\substack{j=1 \\ j \neq i}}^{k+1} \frac{x - X_j}{X_i - X_j} \pmod p$$

IV.5 Synthèse

Les approches étudiées ont toutes réussi à développer un système de partage de secret en utilisant les courbes elliptiques se basant sur le schéma de *Shamir*. Bien que chacune de ces méthodes présente des avantages en ajoutant des options au schéma initial, quelques inconvénients ont été également remarqués.

L'approche de *Wang Caimei et al. 2009* présente différents points positifs tels que la confidentialité des partages assurée par la complexité de résolution du problème de logarithme discret elliptique, ainsi que l'intégrité de la clé privée attribuée par CA qui est affirmée par le hachage. L'algorithme admet l'extensibilité vis à vis le nombre de candidats car l'arrivée d'un nouveau participant ou son départ n'exige aucun changement dans le système. L'approche ajoute une option importante au partage de secret en permettant aux candidats de générer une information qui contribue dans le calcul du partage qu'il leur est destiné contrairement au schéma initial de Shamir, ce qui diminue le risque de plusieurs types d'attaques par écoute de reconstituer le secret.

L'approche proposée par *Hua Sun et al. 2010* traite le partage d'un secret composé et utilise également l'auto sélection des clés par les participants pour la confidentialité des partages. Les chercheurs ont également fait appel à une fonction de hachage pour pouvoir publier les partages et diminuer ainsi le nombre de communications dans leur réseau. Par contre, on a pu soulever une observation importante concernant leur schéma. Les différents composants du secret sont représentés par les coefficients du polynôme de partage, et comme il est indiqué dans le schéma de *Shamir* ce dernier doit avoir un degré de *seuil* -1 suivant le théorème d'interpolation de *Lagrange*, donc une contrainte est exigée concernant la taille du secret pour que le schéma réussisse, elle doit être égale au seuil, donc l'approche fonctionne bien pour un ensemble de secrets générés par le Dealer mais non pas pour un secret quelconque tels que des fichiers choisis à l'avance pour une nécessité bien déterminée. Un autre point positif important est à ne pas négliger, la méthode proposée permet une reconstitution parallèle de différents secrets à la fois, grâce au schéma de structure d'accès généralisé.

L'approche proposée par *Hisham Dahshan et al. 2011* traite un partage proactif dans un réseau de nœuds mobiles MANETs. Ce qu'on remarque de nouveau dans cette méthode est l'utilisation d'un compteur aléatoire *Nonce* qui joue un double rôle dans l'approche, en plus de son utilité vis-à-vis du protocole AODV autant que séquenceur, il permet également de distinguer les sessions l'une de l'autre, donc la validité d'un nœud pour une session précise, n'implique en aucun cas sa validité pour une autre session s'il ne possède pas une clé. Les

chercheurs présument avoir apporté un avantage vis à vis la taille de l'ensemble de clés à générer par une petite matrice choisie de taille $m \times d$, qui pourrait générer T clés possibles avec

$$T = \frac{(m \times d)!}{[(m \times d) - k]!}$$

L'inconvénient à soulever pour cette approche a plus une relation avec le type de réseaux pour lequel elle est conçue, car les MANETs possèdent une contrainte importante vis à vis l'énergie du capteur qui se trouve être limitée, tandis que le schéma proposé génère plusieurs clés et en calcule d'autres par ECDLP qui est reconnu comme étant une opération couteuse, ce qui laisse à penser plus pour sa fiabilité dans un réseau réel de MANETs. En outre, la vitesse de mobilité des capteurs d'une session comparé par la vitesse de calcul des clés pourrait présenter un grand inconvénient, car un nœud mobile dans une session pourrait ne pas être à la portée de ces voisins de session dans la seconde qui suit, alors que les calculs présentés prennent au moins 4 secondes.

Han Yiliang et al. 2003 présente un cryptosystème complet, où il expose un partage vérifiable de secret exploité dans une opération de chiffrement/déchiffrement en utilisant les signature numériques à base de courbes elliptique, ce qui fait face à plusieurs types d'attaque, que ce soit la tentative de falsification de l'un ou de l'autre, car elles sont toutes basées sur la résolution du problème de logarithme discret elliptique.

Nisha Patel et al. 2016 introduit un schéma robuste assuré par l'algorithme ECDLP, l'utilisation du XOR qui donne un aspect aléatoire et uniforme des calculs servant à l'interpolation, sans oublier la sécurité des communication par l'algorithme Double Knapsack qui les a servis à coder toute information transférée via un canal. En revanche, la même contrainte discutée pour l'approche *Hua Sun et al. 2010* est à noter ici concernant la taille du secret.

Toutes les approches ont pu mettre en place une méthode de partage de secret vérifiable en utilisant les courbes elliptiques à base d'interpolation polynomiale, mais elles se distinguent dans d'autres options présentées qu'on synthétise dans le Tableau V.3

Approche	Protocoles					Opérations de base	Critiques
	Shamir	Vérifiable	Auto-sélection	Proactif	Multiple		
Han Yiliang *et al.* 2003	×	×	—	—	—	— ECDLP — ECSA — Interpolation de Lagrange	Partage + Signature + Chiffrement.
Wang Caimei *et al.* 2009	×	×	×	—	—	— ECDLP — Interpolation de Lagrange — Hachage	Extensibilité, confidentialité.
Hua Sun *et al.* 2010	×	×	×	—	×	— ECDLP — Interpolation de Lagrange — Couplage	Reconstitution parallèle. Seuil limité à la taille du secret.
Hisham Dahshan *et al.* 2011	×	×	—	×	×	— ECDLP — Hachage — Interpolation de Lagrange	Taille de l'ensemble des clés. Contrainte d'énergie et mobilité.
Nisha Patel *et al.* 2016	×	×	×	—	×	— ECDLP — Interpolation de Lagrange — Double Knapsack — XOR	Communication sécurisée, aspect uniformément aléatoire. Seuil limité à la taille du secret.

Table IV.1 – Synthèse générale des approches

IV.6 Conclusion

Dans cet état de l'art, nous avons présenté différents types d'approches comportant une variété de protocoles dans le domaine de partage de secret en utilisant les courbes elliptiques. C'est grâce à cette étude qu'on a été inspiré à établir notre propre approche pour développer une application de simulation adéquate.

CHAPITRE V

Contribution

Sommaire

V.1 Introduction

Dans le chapitre précédent nous avons exposé quelques approches dans trois différents types de protocoles de partage de secret : à auto sélection, de multi partage et vérifiable. Pour ce chapitre, nous allons présenter une méthode de partage de secret d'une image numérique dans un réseau Client-Serveur comportant les trois protocoles sus citées. Comme nous présentons les démarches d'implémentation suivies avec les résultats obtenus.

V.2 Approche proposée

Les cryptosystèmes de chiffrement sur un système distribué présente plus de vulnérabilité que sur les systèmes centralisés, et afin de remédier à cela il est préférable de faire appel au partage de secret, d'où l'idée principale sur laquelle on a basé notre approche, avec une clé représentée sous forme d'image numérique qu'on traite comme étant un secret composé.

V.2.1 L'image numérique

Une image numérique est représentée par une matrice de $L \times C$ éléments (lignes/colonnes) [1]. Chaque élément dit pixel (*picture element*) représente l'information stockée sur un seul point formé d'une séquence d'échantillons de 8bits $(0 - 255)$ provenant de 4 canaux différents Figure V.1

- **Alpha :** Canal comportant juste la valeur de transparence du pixel qui varie dans $[0 - 1]$. La valeur 1 indique un pixel opaque par contre 0 désigne un pixel tout à fait transparent, entre les deux le pixel est translucide ;
- **Bleu :** Canal comportant juste la valeur de l'élément en bleu ;
- **Vert :** Canal comportant juste la valeur de l'élément en vert ;
- **Rouge :** Canal comportant juste la valeur de l'élément en rouge ;

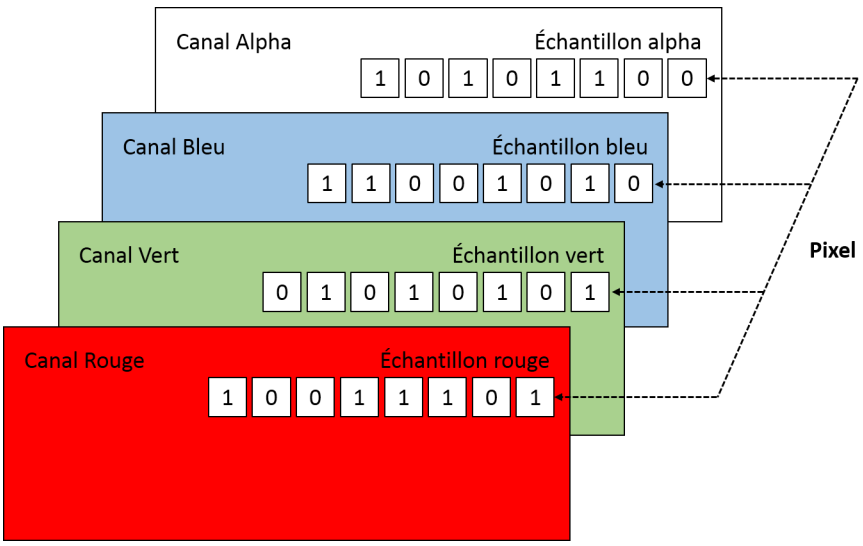


Figure V.1 – La relation entre échantillon, canal et pixel [19]

Dans notre approche, on extrait les valeurs de chaque pixel de l'image comme étant une seule clé afin de formuler le secret composé à partager sous forme matricielle.

1. Contrairement à la définition de la taille d'une matrice, en définissant celle d'une image, on donne généralement la valeur sur l'axe horizontal en premier, puis sur l'axe vertical

V.2.2 Architecture du réseau de partage

L'architecture de base de notre système de partage nécessite deux serveurs, un pour la phase de distribution des partages et l'autre pour la reconstitution du secret. Un client Dealer effectue les opérations d'initialisation des paramètres du système ainsi que le calcul et la distribution des partages, un client pour chaque participant. L'opération de reconstitution est effectuée par le serveur Combiner.

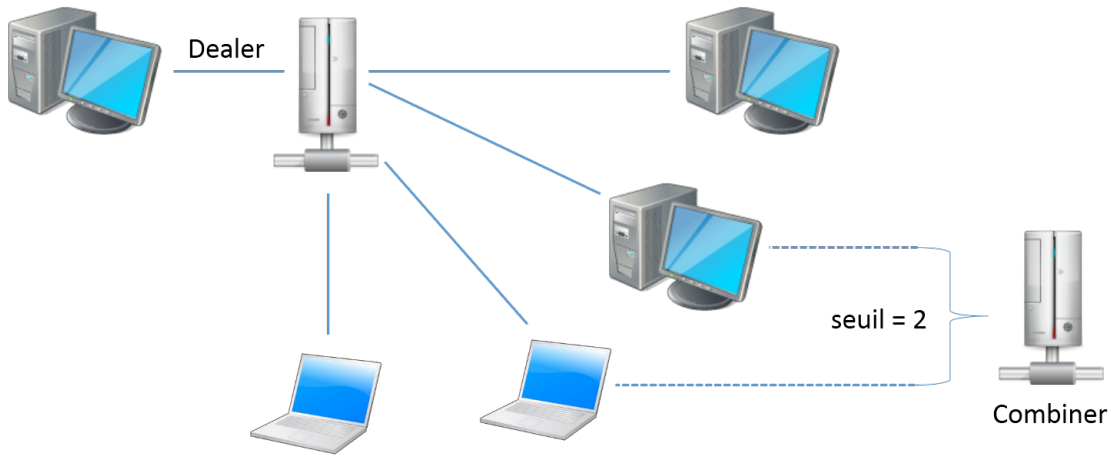


Figure V.2 – Architecture du réseau de partage

V.2.3 Initialisation des paramètres du système de partage

Le système de partage de secret proposé est basé sur les courbes elliptiques, ce qui nécessite au premier lieu d'en déterminer une sur un corps fini. Pour cette raison, le Dealer choisi un grand nombre premier p et une courbe $E(\mathbb{F}_p) : y^2 = x^3 + ax + b \pmod p$ tel que $a, b \in \mathbb{F}_p$, $4a^3 + 27b^2 \neq 0$. Il sélectionne également un point générateur $G(x_G, y_G) \in E(\mathbb{F}_p)$ $(\#E \cdot G = \mathcal{O})$, puis choisi une image (le secret) de taille $T = L \times C$, après quoi il publie $< p, E, G, L, C >$.

Symbole	Description
p	Nombre premier
$\mathbb{F}_p$	Corps fini
$E(\mathbb{F}_p)$	C ellitpique sur un corps fini de la forme $y^2 = x^3 + ax + b \pmod p$
$\mathcal{O}$	Point à l'infini
$\#E$	Cardinal de E (nombre de points)
G	Point générateur
$\mathcal{U}$	Ensemble de participants dans le système
$u_i \in \mathcal{U}$	Un participant
$n = \lvert\mathcal{U}\rvert$	Nombre de participants
k	Seuil pour la reconstitution du secret
$\mathcal{M}$	Matrice du secret
s^i	Clé privée du candidat u_i
T	Nombre de secrets à partager
L	Nombre de lignes de la matrice
C	Nombre de colonnes de la matrice

Table V.1 – Notations mathématiques

Le Dealer attribue à chaque candidat un identifiant unique dès sa connexion en incrémentant un compteur aléatoire *Nonce*. Ensuite, il recevra automatiquement les paramètres nécessaires. Comme il prépare la matrice de secret $\mathcal{M}$ par extraction des valeurs des pixel de l'image.

$$\mathcal{M} = \begin{pmatrix} M_{11} & \cdots & M_{1C} \\ \vdots & \ddots & \vdots \\ M_{L1} & \cdots & M_{LC} \end{pmatrix} \tag{V.1}$$

Cette matrice est obtenu en appliquant l'algorithme suivant :

Algorithme 3 Conversion d'une image en matrice

Entrées : image $C \times L$
Sorties : matrice $\mathcal{M}_{L \times C}$
Algorithme :
  Pour $i = 1$ à L **faire**
    Pour $j = 1$ à C **faire**
      $\mathcal{M}_{ij} \leftarrow RGB(j, i)$
    Fin Pour
  Fin Pour
  Retourner $\mathcal{M}$

Nous allons présenter les démarches du système de partage proposé comme synthétisé dans le diagramme de séquence Figure V.3 :

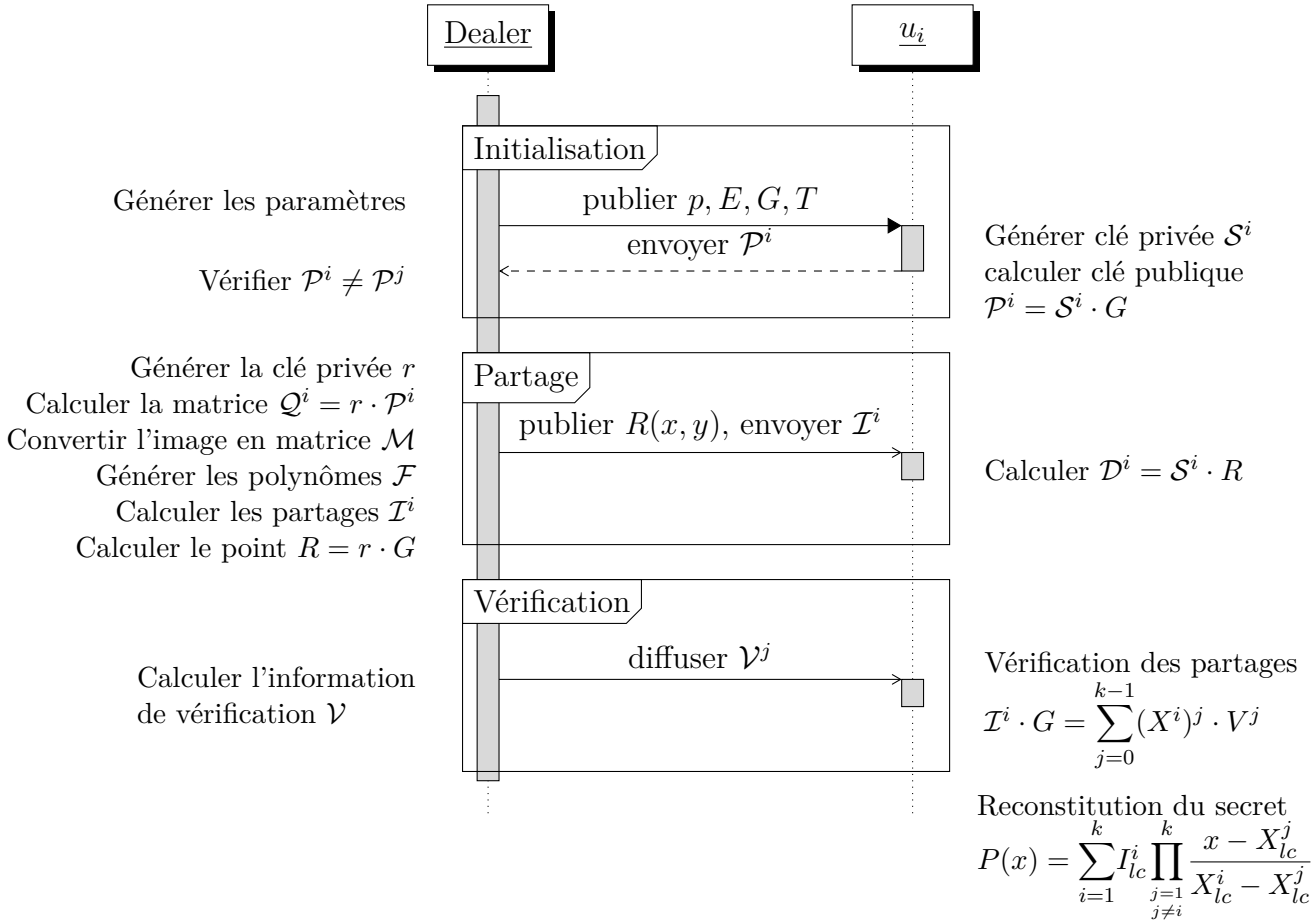


Figure V.3 – Diagramme de séquence de notre approche

V.2.4 Phase de préparation et de distribution des partages

Ayant les paramètres requis, tout candidat u_i sera en mesure de choisir aléatoirement sa propre clé privée $s^i_{lc} \in \mathbb{F}_p$ pour chaque élément du secret, ce qui lui permet de créer la matrice de clés

$$\mathcal{S}^i = \begin{pmatrix} s^i_{11} & \cdots & s^i_{1C} \\ \vdots & \ddots & \vdots \\ s^i_{L1} & \cdots & s^i_{LC} \end{pmatrix} \tag{V.2}$$

Se servant de chacune de ces clés comme un scalaire, u_i génère par ECDLP les clés publiques correspondantes sous forme de points $P^i_{lc} \in E(\mathbb{F}_p)$, et envoie cette matrice au Dealer

$$\mathcal{S}^i \cdot G = \begin{pmatrix} s^i_{11} \cdot G & \cdots & s^i_{1C} \cdot G \\ \vdots & \ddots & \vdots \\ s^i_{L1} \cdot G & \cdots & s^i_{LC} \cdot G \end{pmatrix} = \mathcal{P}^i = \begin{pmatrix} P^i_{11} & \cdots & P^i_{1C} \\ \vdots & \ddots & \vdots \\ P^i_{L1} & \cdots & P^i_{LC} \end{pmatrix} \tag{V.3}$$

Dès réception des clés publiques, le Dealer s'assure de la distinction des matrices clé $\mathcal{P}^i \neq \mathcal{P}^j$, dans le cas contraire il réclame aux participants concernés d'autres matrices. Les opérations V.2 et V.3 sont à itérer jusqu'à ce que le Dealer collecte n matrices différentes.

Après la collecte des clés, le Dealer prépare les partages en passant par les étapes suivantes :

- **Étape 1** Il choisi aléatoirement une seule clé privée $r \in \mathbb{F}_p$, par laquelle il calcule par ECDLP une matrice de points $\mathcal{Q}^i$

$$r \cdot \mathcal{P}^i = \begin{pmatrix} r \cdot P^i_{11} & \cdots & r \cdot P^i_{1C} \\ \vdots & \ddots & \vdots \\ r \cdot P^i_{L1} & \cdots & r \cdot P^i_{LC} \end{pmatrix} = \mathcal{Q}^i = \begin{pmatrix} Q^i_{11} & \cdots & Q^i_{1C} \\ \vdots & \ddots & \vdots \\ Q^i_{L1} & \cdots & Q^i_{LC} \end{pmatrix} \tag{V.4}$$

  où chaque point $\mathcal{Q}^i_{lc}$ a les coordonnées (X^i_{lc}, Y^i_{lc})

- **Étape 2** Notre approche est basée sur le schéma de *Shamir*, par conséquent un polynôme de degré $k = seuil$ est requis pour chaque secret (pixel)

$$f_{lc}(x) = M_{lc} + \sum_{i=1}^{k-1} a_{i,lc} \cdot x^i \pmod p \tag{V.5}$$

  avec $a_{i,lc} \in \mathbb{F}_p$ aléatoirement choisis, le Dealer obtient donc la matrice de polynômes

$$\mathcal{F} = \begin{pmatrix} f_{11} & \cdots & f_{1C} \\ \vdots & \ddots & \vdots \\ f_{L1} & \cdots & f_{LC} \end{pmatrix} \tag{V.6}$$

- **Étape 3** Afin de préparer les partages, il calcule l'image de chaque abscisse des points $\mathcal{Q}^i_{lc}$ par le polynôme adéquat de la même position dans une matrice, le résultat est une autre matrice du partage associée à un candidat précis u_i qui est égale à I^i qu'on appelle également couche ou *shadow* et qui lui sera transmis par la suite

$$\mathcal{I}^i = \begin{pmatrix} I^i_{11} & \cdots & I^i_{1C} \\ \vdots & \ddots & \vdots \\ I^i_{L1} & \cdots & I^i_{LC} \end{pmatrix} \pmod p \tag{V.7}$$

- **Étape 4** Après avoir calculé la clé privée, le Dealer a besoin d'une clé publique correspondante pour pouvoir échanger la matrice des points Q^i avec les participants, il génère donc un point R par ECDLP qu'il publie

$$R = r \cdot G \pmod p \tag{V.8}$$

Afin de mettre en place cette phase, nous avons proposé un algorithme simple pour le calcul des partages suivant le schéma de *Shamir* :

Algorithme 4 Calcul des partages de Shamir

Entrées : $\{a_0, a_1, \cdots, a_{k-1}\}$, X_{Q^i}
Initialisation :
  $polynôme\_somme \leftarrow 0$
Sorties : partage Y^i
Algorithme :
  Pour $j = 0$ à $seuil - 1$ **faire**
    $polynôme\_somme \leftarrow polynôme\_somme + (a_j \times X^j)$
  Fin Pour
  Retourner Y^i

Le Dealer après avoir reçu les points publique, applique l'algorithme d'initialisation et de distribution des partages suivant :

Algorithme 5 Initialisation et distribution des partages

Entrées : matrices : $\mathcal{P}^i$, $clé\_privée$, $\mathcal{F}$
Sorties : partage $\mathcal{I}^i$
Algorithme :
  Pour $l = 0$ à $longueur$ **faire**
    Pour $c = 0$ à $largeur$ **faire**
      $\mathcal{Q}^i_{lc} \leftarrow logarithme\_discret(\mathcal{P}^i_{lc}, clé\_privée)$
      $\mathcal{I}^i_{lc} \leftarrow calcul\_partage(\mathcal{F}_{lc}, \mathcal{Q}^i_{lc})$
    Fin Pour
  Fin Pour
  Retourner $\mathcal{I}^i$

Cette phase diffère de son équivalente dans l'approche originale de *Shamir* par l'auto sélection de clés privées de partage pour chaque participant, ce qui laisse une marge de contribution des candidats dans la phase de préparation des partages, donc n'importe quel attaquant ayant intercepté k couches transmises ne pourra jamais reconstituer le secret puisqu'il ne possède pas les clés privées.

V.2.5 Phase de reconstitution du secret

Afin de reconstituer le secret, le groupe de participants qualifiés qui représente la structure d'accès u_j avec $1 \leq j \leq k$ utilise la matrice réceptionnée $\mathcal{I}^i$ ainsi que le point publié R pour pouvoir générer le polynôme de Lagrange $P(x)$:

- **Étape 1** u_j calcule la matrice des points $\mathcal{D}^i$ pour récupérer les coordonnées par échange de clés en utilisant sa clé privée S^j

$$\mathcal{S}^j \cdot R \pmod p = \mathcal{D}^j = \begin{pmatrix} D^j_{11} & \cdots & D^j_{1C} \\ \vdots & \ddots & \vdots \\ D^j_{L1} & \cdots & D^j_{LC} \end{pmatrix} \tag{V.9}$$

- **Étape 2** Le point D calculé a les même coordonnées que le point Q, par conséquent, le polynôme de Lagrange pourrait être formulé à base de la matrice $\mathcal{I}^i$ reçu ainsi que les

abscisses des points D :

$$P(x) = \sum_{i=1}^{k} I_{lc}^{i} \prod_{\substack{j=1 \\ j \neq i}}^{k} \frac{X_{lc}^{j}}{X_{lc}^{j} - X_{lc}^{i}} \tag{V.10}$$

Pour cette phase, nous avons proposé un algorithme calculant le polynôme de *Lagrange* comme suit :

Algorithme 6 Reconstitution du secret (Interpolation de Lagrange)

Entrées : $\{X_1, X_2, \cdots, X_k\}$, $\{Y_1, Y_2, \cdots, Y_k\}$
Initialisation :
  $somme \leftarrow 0$, $M \leftarrow 0$
Sorties : secret M
Algorithme :
  Pour $i = 0$ à $seuil - 1$ **faire**
    $produit \leftarrow 1$, $multiple \leftarrow 1$
    Pour $j = 0$ à $seuil - 1$ **faire**
      Si $j \neq i$ **alors**
        $fraction \leftarrow X_j (X_j - X_i)^{-1} \pmod{p}$
        $multiple \leftarrow multiple \times fraction$
      Fin Si
    Fin Pour
    $produit \leftarrow produit \times multiple \pmod{p}$
    $somme \leftarrow somme + (Y_i \times produit)$
  Fin Pour
  $M \leftarrow M + somme \pmod{p}$
  Retourner M

Cependant, l'image du secret pourrait être reconstruite en lisant chaque élément de la matrice et le convertir en pixel RGB, chaque canal doit être extrait à part selon sa position, suite à quoi on applique une concaténation des 4 canaux afin de reformuler un pixel, qu'on attribue à la bonne position dans l'image.

Algorithme 7 Conversion d'une matrice en image

Entrées : matrice $\mathcal{M}_{L \times C}$
Sorties : image $C \times L$
Algorithme :
  Pour $i = 1$ à L **faire**
    Pour $j = 1$ à C **faire**
      $RGB(j,i) \leftarrow \mathcal{M}_{ij}$
      $Alpha \leftarrow (RGB(j,i) >> 24) \& 0xff$
      $Rouge \leftarrow (RGB(j,i) >> 16) \& 0xff$
      $Vert \leftarrow (RGB(j,i) >> 8) \& 0xff$
      $Bleu \leftarrow RGB(j,i) \& 0xff$
      $pixel \leftarrow (Alpha << 24) + (Rouge << 16) + (Vert << 8) + Bleu$
      $image_{ji} \leftarrow pixel$
    Fin Pour
  Fin Pour
  Retourner $image$

V.2.6 Phase de vérification non interactive des partages

Dans cette phase, nous proposons une technique de vérification de l'intégrité des partages interceptés où chaque candidat u_i est capable de s'assurer de la validité de la matrice I^i reçu sans passer par la reconstitution du secret, aucune interaction entre les participants n'est requise. Dans le cas où le résultat de cette opération est négatif, le partage ne servira pas à la reconstitution et le candidat établie un avertissement qu'il transmet au Dealer. Cette opération est garantie par une information supplémentaire calculée et diffusée par ce dernier à l'ensemble des participants.

Pour chaque élément polynôme f_{lc} avec les coefficients $a_0, a_1, \cdots, a_{k-1}$, le Dealer calcule par ECDLP k points V_{lc}^j de la forme

$$V_{lc}^j = a_{lc}^j \cdot G \pmod{p} \tag{V.11}$$

avec $0 \leq j \leq k-1$. Donc chaque u_i recevra une matrice de liste de points V_{lc}^j qui lui permettra de vérifier la validité des partages par la formule

$$I_{lc}^i \cdot G = \sum_{j=0}^{k-1} (X_{lc}^i)^j \cdot V_{lc}^j \pmod{p} \tag{V.12}$$

Les points V_{lc}^j seront regroupés dans une matrice $\mathcal{V}^j$, donc k matrices vont être diffusées :

$$\mathcal{V}^j = \begin{pmatrix} V_{11}^j & \cdots & V_{1C}^j \\ \vdots & \ddots & \vdots \\ V_{L1}^j & \cdots & V_{LC}^j \end{pmatrix} \tag{V.13}$$

Afin de partager ces matrices, nous avons mis en place l'algorithme suivant qui les calcule comme étant une grande matrice de liste de points :

Algorithme 8 Préparation des matrice de vérification

Entrées : G, matrice $\mathcal{F}$
Sorties : matrice $\mathcal{V}$
Algorithme :

```
Pour l = 0 à longueur faire
    Pour c = 0 à largeur faire
        Créer{v_0, v_1, ..., v_{k-1}}
        Pour j = 0 à seuil − 1 faire
            v_j ← logarithme_discret(G, F_{lc,j})
        Fin Pour
        V_lc ← {v_0, v_1, ..., v_{k-1}}
    Fin Pour
Fin Pour
Retourner V
```

Ceci était du côté Dealer, pour la vérification du partage côté Candidat, l'algorithme applique seulement le calcule des points par ECDLP et vérifie les deux matrices obtenues.

V.2.7 Preuves de validité

Afin de s'assurer de la validité des formules d'échange et de vérification, nous présentons les preuves ci après :

Preuve 1 *Echange de coordonnées*

Afin de passer les cordonnées des points Q^i au candidats, nous avons fait appel à la méthode d'échange de clés de *Diffie-Hellman*, puisque les deux communicants possèdent chacun une clé privée et une autre publique, les cordonnées pourraient être échangées suivant les égalités

$$Q^i_{lc} = r \cdot P^i_{lc} = r \cdot s^i_{lc} \cdot G = s^i_{lc} \cdot R = D^i_{lc} \tag{V.14}$$

Preuve 2 *Vérification*

N'importe quel candidat u_i possédant un partage et ayant échangé les coordonnées nécessaires, est en mesure de vérifier la validité de sa matrice I^i avant de s'en servir à reconstituer le secret suivant la formule

$$\begin{aligned} I^i_{lc} \cdot G &= f_{lc}(X^i_{lc}) \cdot G \pmod p \\ &= [a_0 + a_1 \cdot X^i_{lc} + \cdots + a_{k-1} \cdot (X^i_{lc})^{k-1} \pmod p] \cdot G \\ &= a_0 \cdot G + a_1 \cdot X^i_{lc} \cdot G + \cdots + a_{k-1} \cdot (X^i_{lc})^{k-1} \cdot G \pmod p \\ &= \sum_{j=0}^{k-1} (X^i_{lc})^j \cdot V^j_{lc} \pmod p \end{aligned} \tag{V.15}$$

V.3 Implémentation

Dans cette section nous présentons notre implémentation de l'approche proposée ainsi que les tests effectués avec une analyse des résultats.

V.3.1 Les outils utilisés

Dans le but d'implémenter une application de test pour notre approche, on avait besoin de quelques outils qu'on présente comme suit :

V.3.1.1 JavaFX

Comme langage de programmation, on a opté pour Java dans sa version JavaFX qui a été introduit à JavaOne 2007 en utilisant la famille de produits JavaFX de Sun Microsystems. L'objectif principal de cette version est de permettre le développement et le déploiement d'applications riches en contenu pour tout genre d'appareils. JavaFX utilise des scripts simples et déclaratifs, avec des fonctions et des attributs typés statiquement [20]. Parmi ses avantages :

- Sa syntaxe déclarative simple qui permet d'exprimer les interfaces utilisateurs y compris un ensemble de widgets simplifiant la mise en page des interfaces indépendamment de leur plate-forme.
- Sa prise en charge du modèle architectural MVC (Model-View-Controller) par défaut qui permet une forte liaison bidirectionnelle entre l'interface et la plate-forme tout en gardant l'indépendance du code de chacun.
- Sa possibilité d'attribuer des déclencheurs et d'invoquer des méthodes de manière directe dans le code déclaratif des composants de l'interface utilisateur.
- Son héritage de tous les avantages natifs du langage Java, tels que son exécution via la machine virtuelle Java (JVM) qui lui permet d'être exécuté partout.

V.3.1.2 Barcode Generator

C'est une application Android qui génère tout genre de codes à barres y compris les codes QR en toutes versions. Cette application développée par *AEIOU*, permet d'encoder différents types d'informations : text, url, email, numéro de téléphone, géo-localisation, configuration wifi, etc. Comme elle donne la possibilité de personnaliser le fichier résultat représenté par le code avant sa sauvegarde en choisissant

- L'emplacement de sauvegarde ;
- Le format du fichier sauvegardé (jpg, png, webp) ;
- La taille de l'image à sauvegarder $(33, 66, 132, 400, 800, 1600$ px) ;
- Les couleurs de l'image ;

L'application permet également la modification de n'importe quel code généré ainsi que son partage sur les différents sites sociaux et applications de conversation installées.

V.3.1.3 QR code lecteur

Développée par *Scan Mobile*, une application Android qui permet de lire les code QR avec une détection automatique rapide et fiable, qui donne la possibilité de sauvegarder les codes lus, de les partager sur les applications installées et même d'en garder un historique.

V.3.2 Analyse des résultats

On a pu effectuer trois types de tests que nous exposons avec une petite analyse dans les sections suivantes :

V.3.2.1 Fiabilité et confidentialité

Afin de prouver la fiabilité de l'opération de reconstitution ainsi que la confidentialité du partitionnement, nous avons effectué un test sur une image de code QR.

Code QR : (*Quick Response Code*)
C'est un type de code à barres à deux dimensions, capable de gérer tous les types de données, numériques, alphabétiques, les symboles, les codes binaires ... pouvant atteindre jusqu'à 7 089 dans un seul symbole. Contrairement au code à barres, le code QR permet de transporter des informations horizontalement et verticalement dans un espace d'environ un dixième, ce qui permet d'obtenir un format d'impression plus petit pour la même taille d'information. Le code QR est composé de plusieurs modules (points noirs et blancs) selon la version (version 1 avec 21×21 modules, jusqu'à la version 40 avec 177×177 modules). Il possède trois motifs de détection qui sont situés dans le coin inférieur gauche, supérieur gauche et supérieur droit, utilisés pour reconnaître le code QR et détecter la position du symbole, ce qui lui permet d'être omnidirectionnel, capable d'être lu à 360 degrés et à haute vitesse [7].

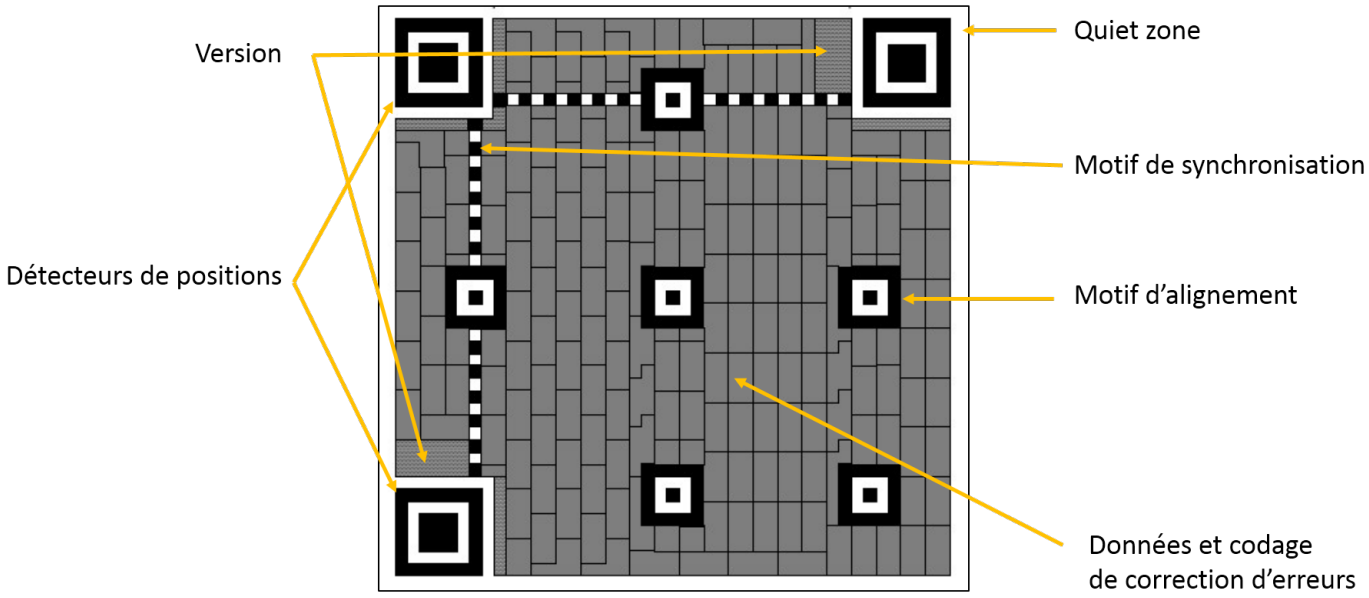
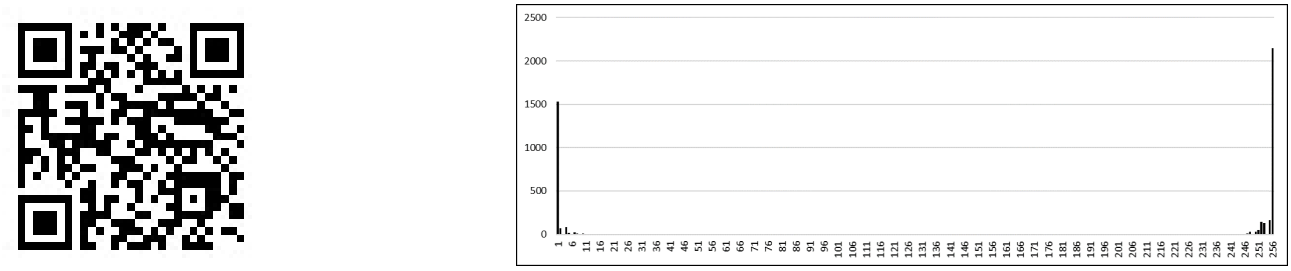


Figure V.4 – Symboles d'un code QR [7]

Pour notre test, on a généré un code QR de 66 × 66px de la phrase "secret sharing using elliptic curves" Figure V.5a qu'on a partagé sur notre schéma avec un seuil = 3, les partages des 3 candidats qualifiés pour la reconstitution sont présentés dans la Figure V.6.



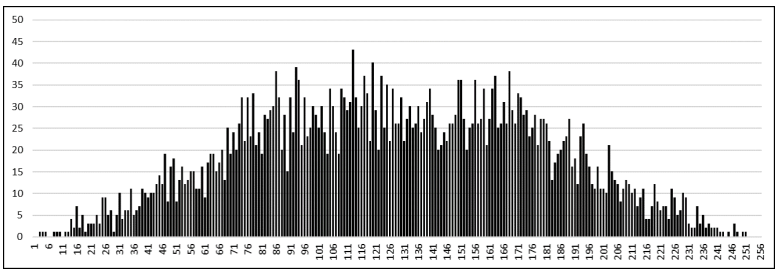
(a) Secret QR

(b) Histogramme de niveau de gris du QR

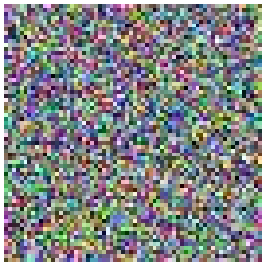
Figure V.5 – L'image du secret à répartir

Dans le but de prouver la confidentialité de notre système de partage, on a utilisé un lecteur de code QR pour essayer d'extraire quelques informations depuis les partages obtenus, mais le lecteur n'arrivait même pas à reconnaitre l'image comme étant une image de code QR. La différence entre l'histogramme du niveau de gris du secret QR et celui des 3 partages montre bien que l'opération du partitionnement ne révèle aucune information sur le secret. Après reconstitution du secret, le test nous a bien donné le même code QR d'origine avec la même phrase fournie lors de sa génération.

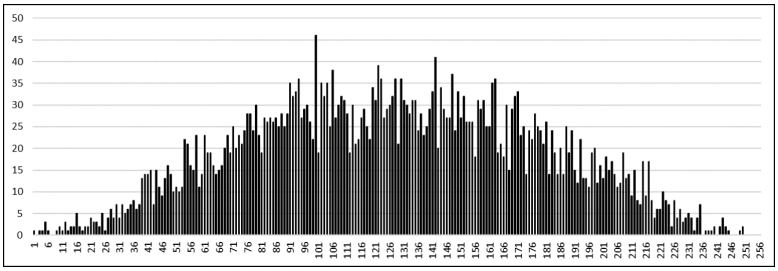
(a) Partage 1



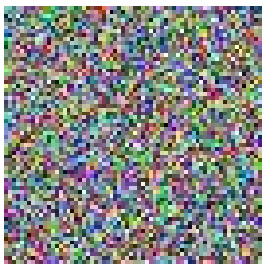
(b) Histogramme de niveau de gris du Partage 1



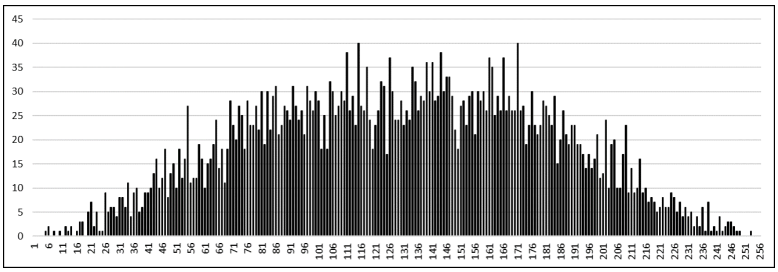
(c) Partage 2



(d) Histogramme de niveau de gris du Partage 2



(e) Partage 3



(f) Histogramme de niveau de gris du Partage 3

Figure V.6 – Partages obtenus

V.3.2.2 Temps d'exécution

Dans cette section, nous présentons les performances du système de partage proposé en notion de temps d'exécution, qui reste un point essentiel pour n'importe quel cryptosystème. Pour cette raison, on a effectué 4 tests différents où on incrémente chaque fois le seuil et en utilisant une image de 33×33px, on a calculé le temps d'exécution de 5 différentes opérations exposées comme suit dans le tableau V.2 :

- *Génér.Vérif :* le temps de calcul de l'opération de génération des matrices de vérification V^j ;
- *ECDLP :* le temps moyen de calcul d'une matrice de points par ECDLP telle que P^i ;
- *Shamir :* le temps moyen de calcul des partages de *Shamir*, représentés par les matrices I^i ;
- *Vérification :* le temps moyen que prend chaque candidat pour vérifier son partage ;
- *Reconstitution :* le temps de calcul pour la construction du polynôme de *Lagrange* et reconstitution du secret ;

Seuil	Génér.Vérif	Moyennes de calcul par participant			Reconstitution
		ECDLP	Shamir	Vérification	
2	8037	7022	4	19835	184
3	14514	7027	13	39775	312
4	21619	6923	15	65343	431
5	28266	7037	18	99543	635

Table V.2 – Temps d'exécution des différentes opérations en ms

D'après les calculs présentés, nous remarquons que l'opération de partage de *Shamir* est presque négligeable et même celle de l'interpolation de *Lagrange*, contrairement au calcul des matrices de points qui se fait par ECDLP, en plus le temps de vérification des partages représente l'opération la plus couteuse car elle comprend plusieurs itérations de ECDLP.

Afin de juger nos résultats, on a effectué une comparaison de l'opération la plus couteuse dans notre approche avec une autre approche déjà étudiée (Tableau V.3), il s'agit du temps d'exécution dans le schéma présenté par *Hichem Dahshan et al. 2011* [6].

Schéma	Taille Matrice	Seuil	Courbe	temps Vérification
Dahshan *et.al.*	24×16	5	192bits	37380
			256bits	67912
Notre approche	24×16	5	192 bits	35526
			256 bits	64271

Table V.3 – Comparaison entre les temps de vérification en ms

Pour ces tests, nous avons partagé une matrice de la même taille que celle présentée dans les résultats de l'approche étudiée 24×16 avec un seuil égal à 5 en utilisant deux types de courbes elliptiques. Bien que nos résultats ont été pris d'un vrai partage d'image, on remarque que le temps d'exécution enregistré est un peu diminué de celui de *Dahshan et al.* donné par simulation.

V.3.2.3 Sécurité

À partir de différents histogrammes analysés, on a pu conclure que le schéma n'est pas sensible à la valeur du seuil, que ce soit des partages générés avec seuil de 2 ou de 5, nous obtenons toujours des histogrammes pareils, aucune information circulée sur l'image originale.

Dans notre approche, les communications avec le serveur Dealer ne requis pas un grand niveau de sécurité, le partage est considéré comme une opération sécurisée puisque les communications Dealer/Candidats sont soit pour le transfert de données publiques soit pour les diffusions, juste la transmission du partage qui est sensible, mais il faut noter que les candidats sont indépendants l'un de l'autre, aucune interaction entre eux n'est requise. Par conséquent, une attaque externe demande une connaissance préalable du temps exact de transmission de partage pour chaque participant, sinon n'importe quelle interception de $k-1$ partages ne servira jamais à reconstituer le secret.

Un autre type d'attaque est possible à signaler dans un système de partage de secret, il s'agit des acteurs malicieux qui modifient les partages. C'est pour ce type d'attaques que les options de vérifications ont été proposées, où n'importe quel candidat est en meure de vérifier l'intégrité de sa matrice reçue avant, et sans reconstitution du secret d'une manière indépendante des autres.

V.3.3 Présentation de l'application

Comme indiqué dans l'architecture sus citée, afin de simuler l'approche proposée on avait besoin de développer 3 applications : une pour le partage représenté par le Dealer, une deuxième pour les candidats, et une troisième représentant le serveur Combiner. Dans ce qui suit nous exposons les différentes interfaces des 3 acteurs :

Dans toutes les interfaces que nous allons présenter ci dessous, nous avons inclut le temps d'exécution pour les opérations de base discutées précédemment, ainsi que les graphes en histogramme RGB pour les images, afin de donner plus de lisibilité des résultats à notre audience. Comme nous avons laisser apparaitre les matrices obtenues pour cette version de simulation pour pouvoir bien remarquer la complexité des calculs effectués.

V.3.3.1 Dealer partageant un secret

On génère d'abord grâce à l'application android sus citée, un code QR cachant la phrase "secret sharing using elliptic curves" qu'on considère comme étant notre secret à partager (voir Figure V.7).

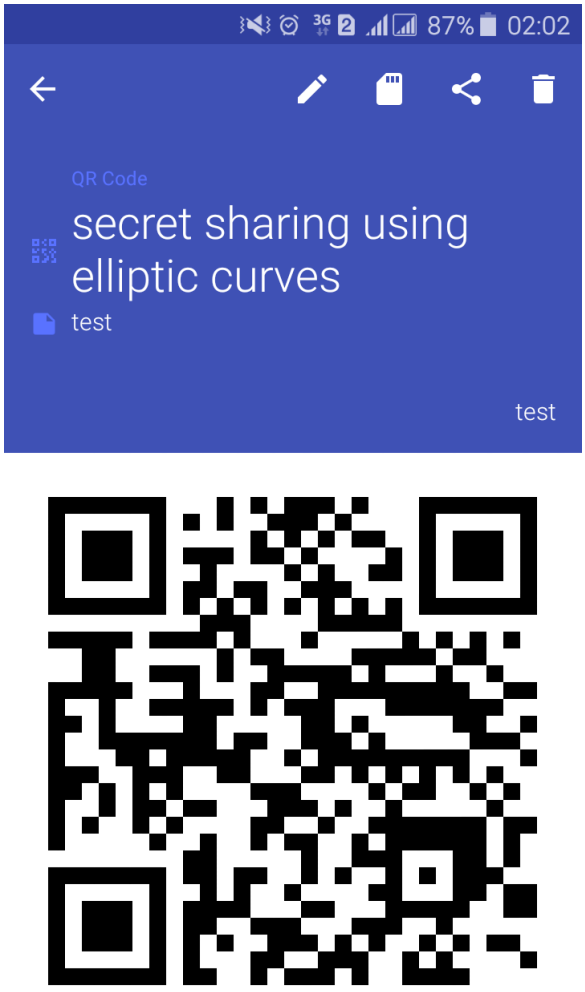


Figure V.7 – Code QR généré (secret)

Le dealer, dans l'interface (Figure V.8) initialise les paramètres. On a envisagé d'utiliser des courbes elliptiques cryptographiques du standard $NIST$[2] qui nous propose des courbes dans une variété de corps finis de longueurs différentes avec leur générateur [15].

2. National Institute of Standards and Technology

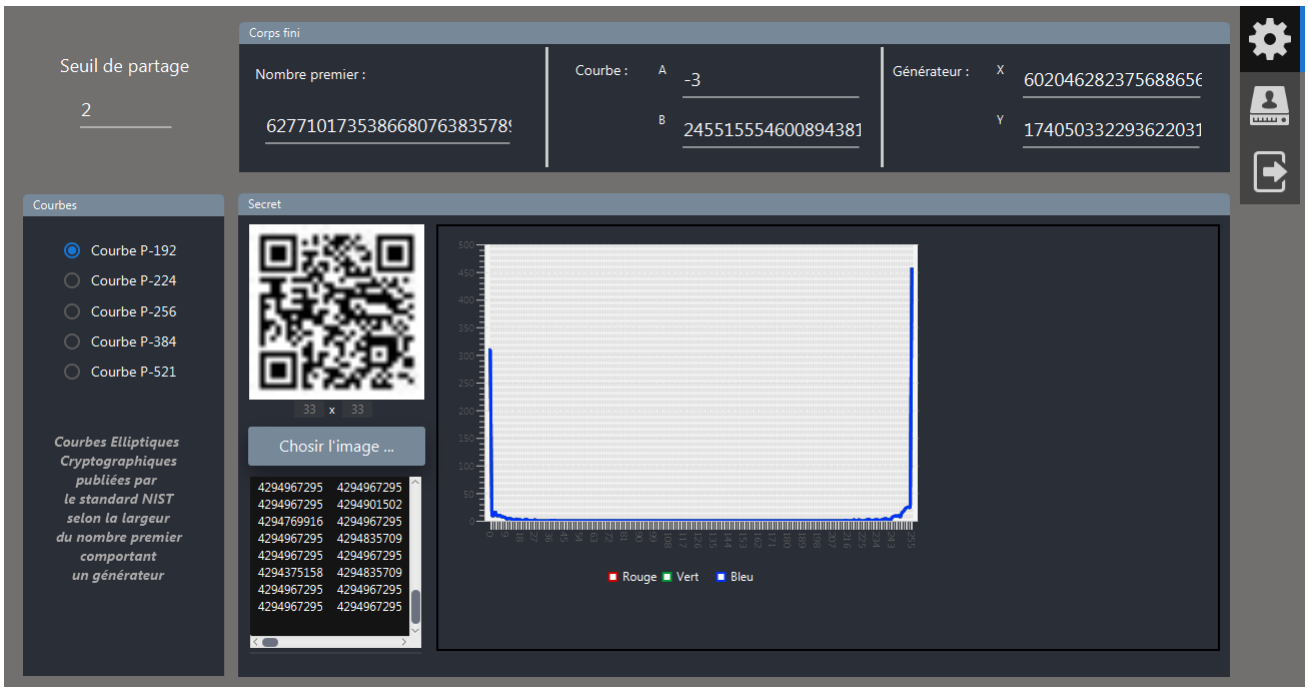


Figure V.8 – Initialisation des paramètres du système de partage

Dans ce deuxième onglet Figure V.9, le Dealer lance le serveur de partage afin de transmettre les paramètres requis au candidat ainsi qu'au Combiner. Cette opération est effectuée pour une version de simulation, car pour un système réel, ces paramètres seront rendus publiques. Nous pouvons également remarquer les candidats connectés au serveur de partage et le résultat des communications effectuées.

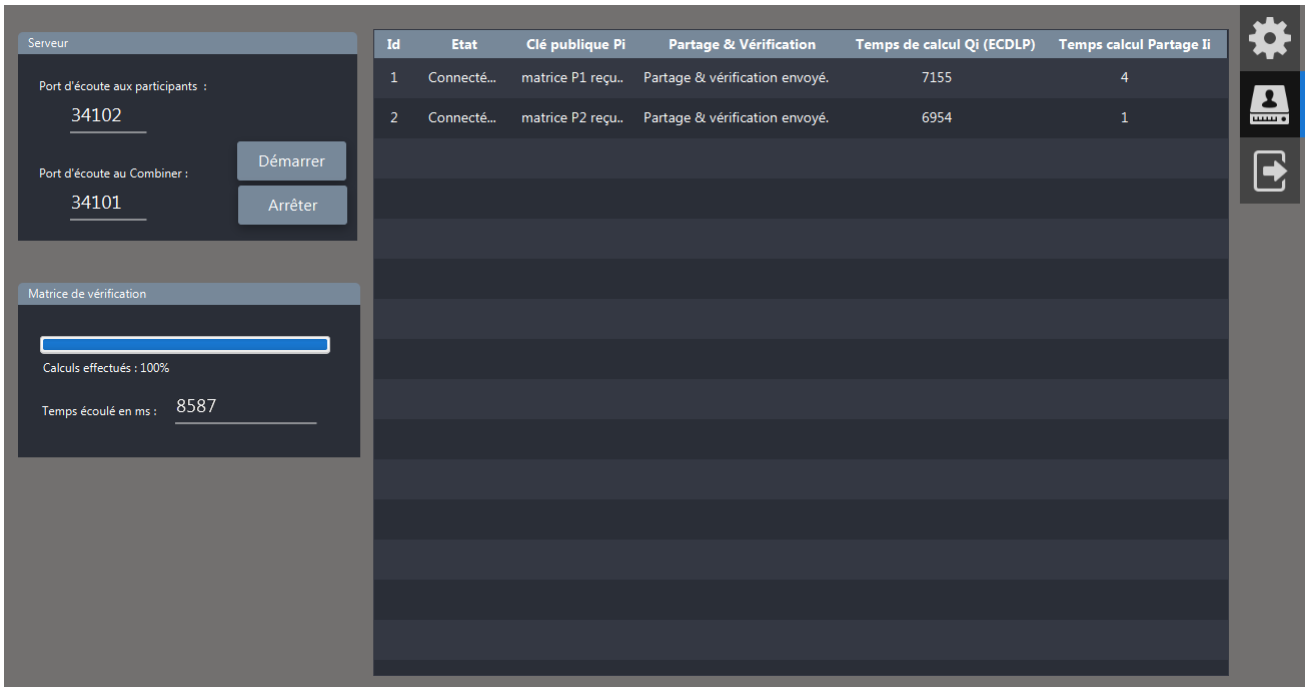


Figure V.9 – Distribution des partages

V.3.3.2 Candidat réceptionnant un partage

Dès connexion, le candidat reçoit les paramètres publiques du Dealer (Figure V.10).

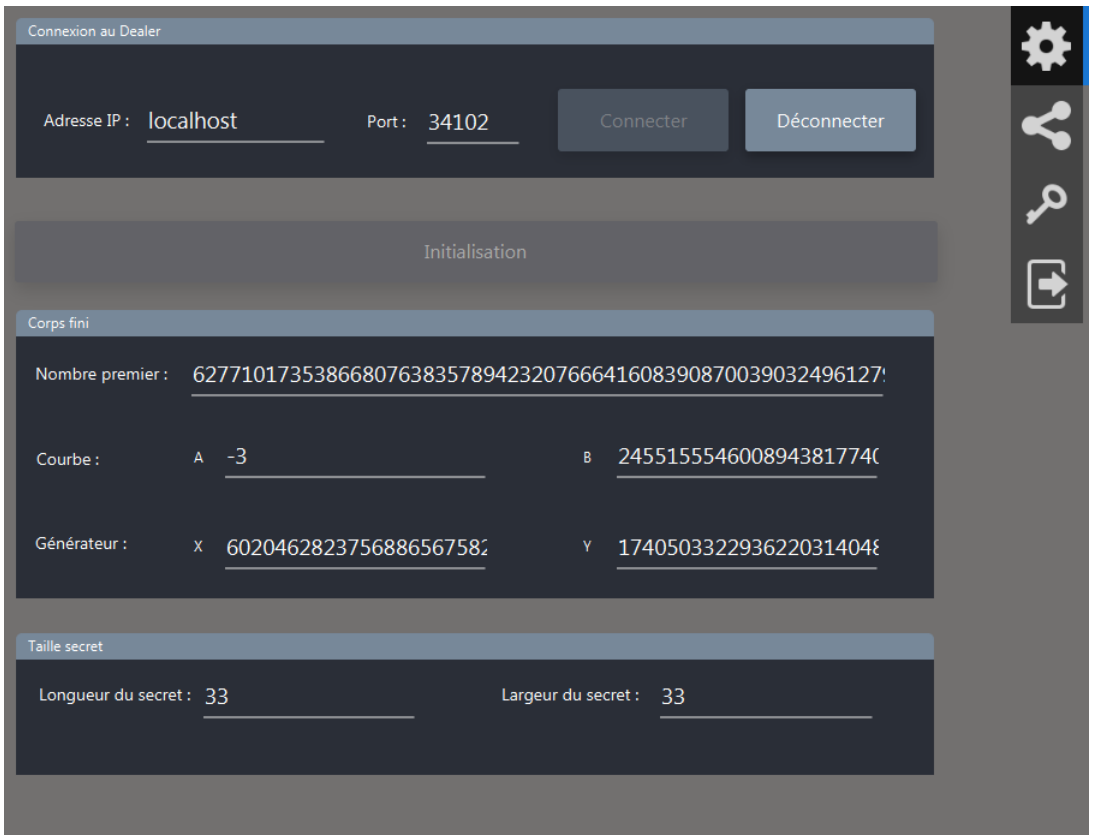


Figure V.10 – Réception des paramètres

Dans l'interface (Figure V.11), le candidat génère sa paire de clé privée et publique, envoi la clé publique au Dealer, suite à quoi il réceptionne son partage.

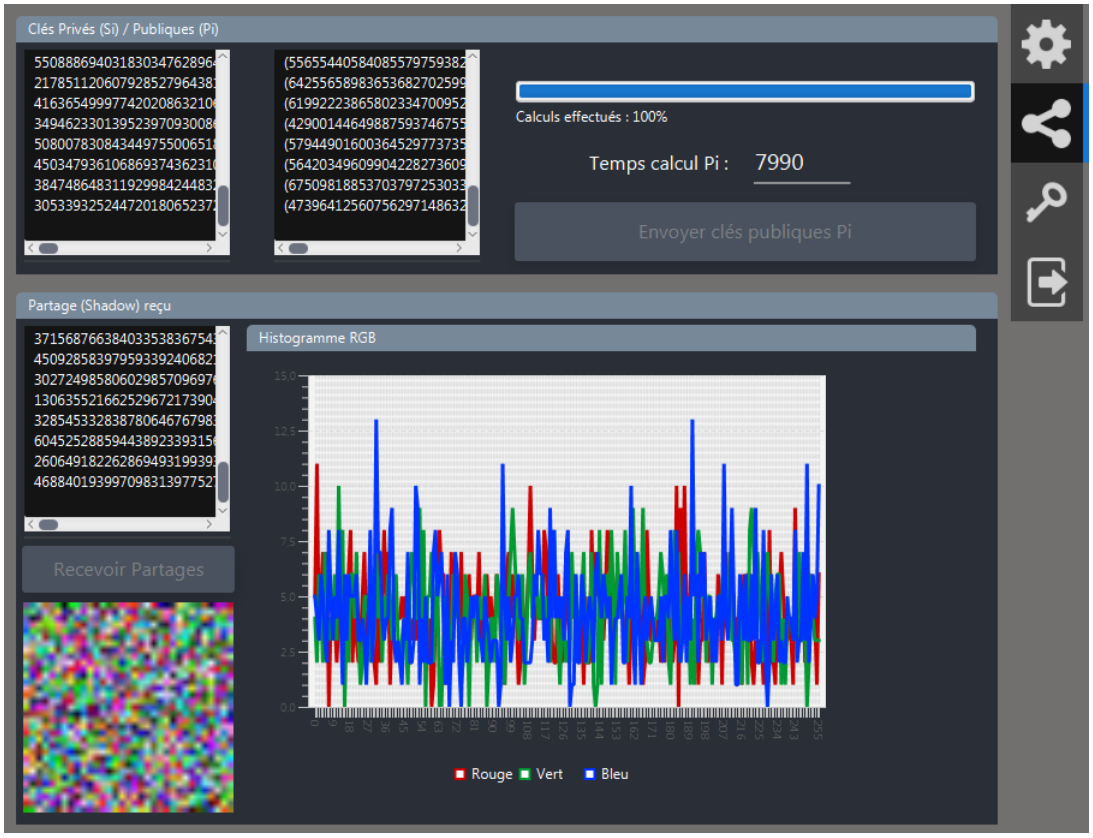


Figure V.11 – Génération des clés Privée/Publique et réception du partage (shadow)

Un candidat est capable de vérifier son shadow après avoir calculé la matrice d'échange de points, ce qui lui permet de demander une reconstitution du secret en se connectant au serveur Combiner.

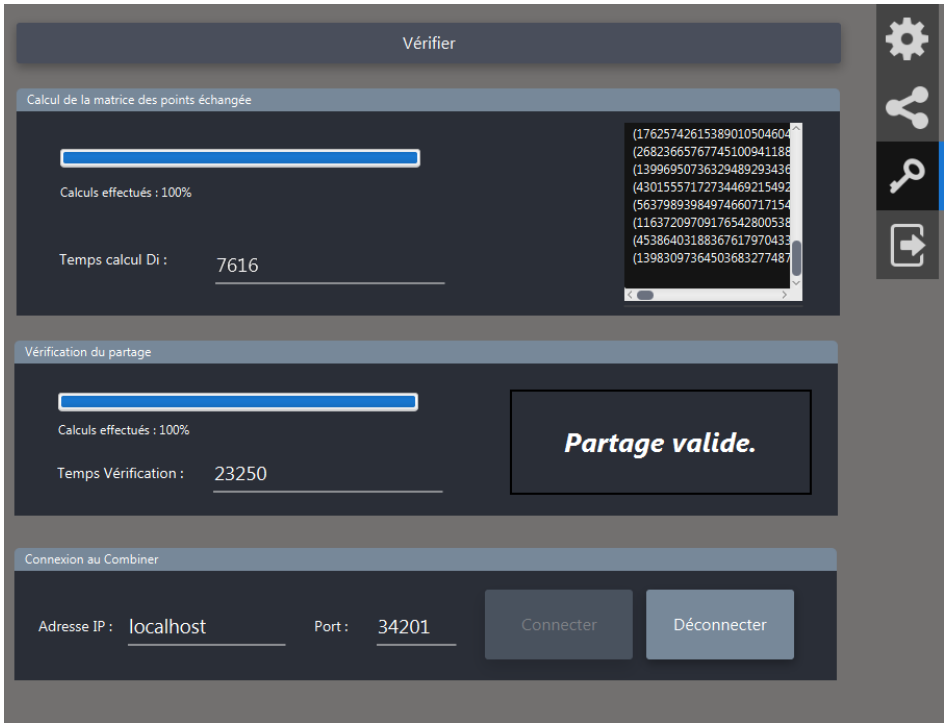


Figure V.12 – Vérification du partage et demande de reconstitution

V.3.3.3 Combiner reconstituant le secret

Le combiner a besoin de quelques paramètres seulement pour générer le polynôme de Lagrange, qu'il reçoit en se connectant au Dealer (Figure V.13).

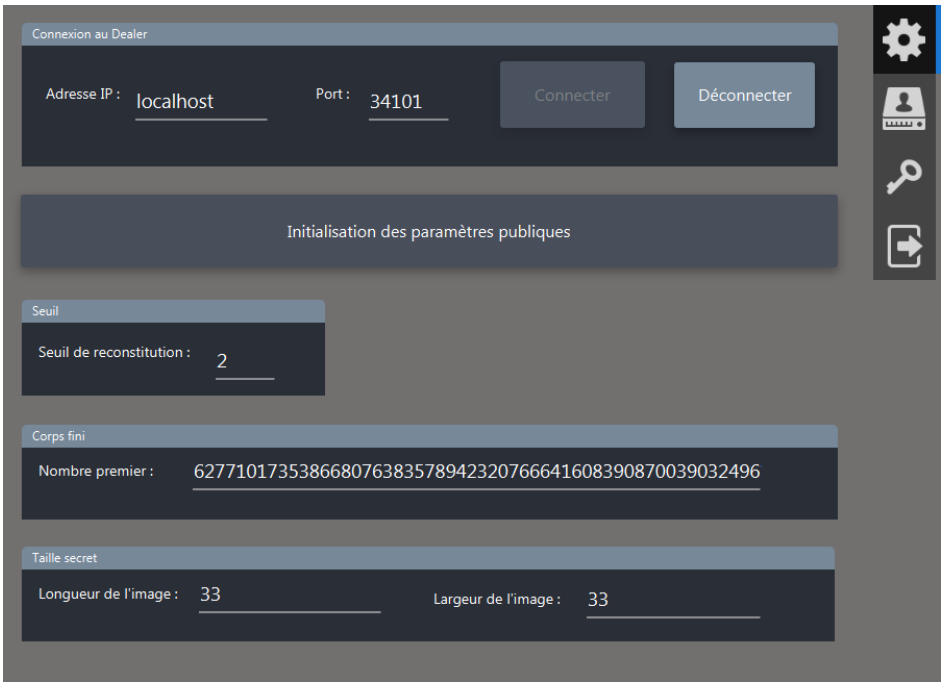


Figure V.13 – Réception des paramètres publiques

Les participants demandant une reconstitution du secret, se connectent au serveur et envoient leur partage comme indiqué dans le tableau de l'interface (Figure V.14).

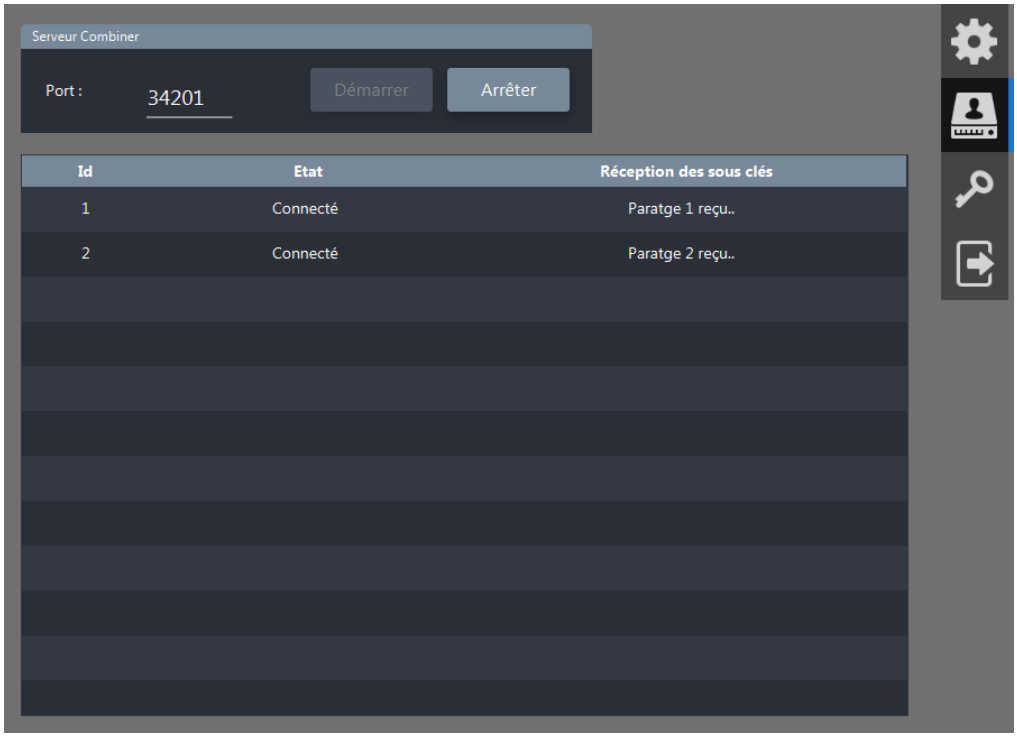


Figure V.14 – Lancement du serveur et connexion des candidats

Enfin, dès collecte de $k = seuil$ partages, on donne la main à l'utilisateur de lancer la reconstitution du secret (voir Figure V.15).

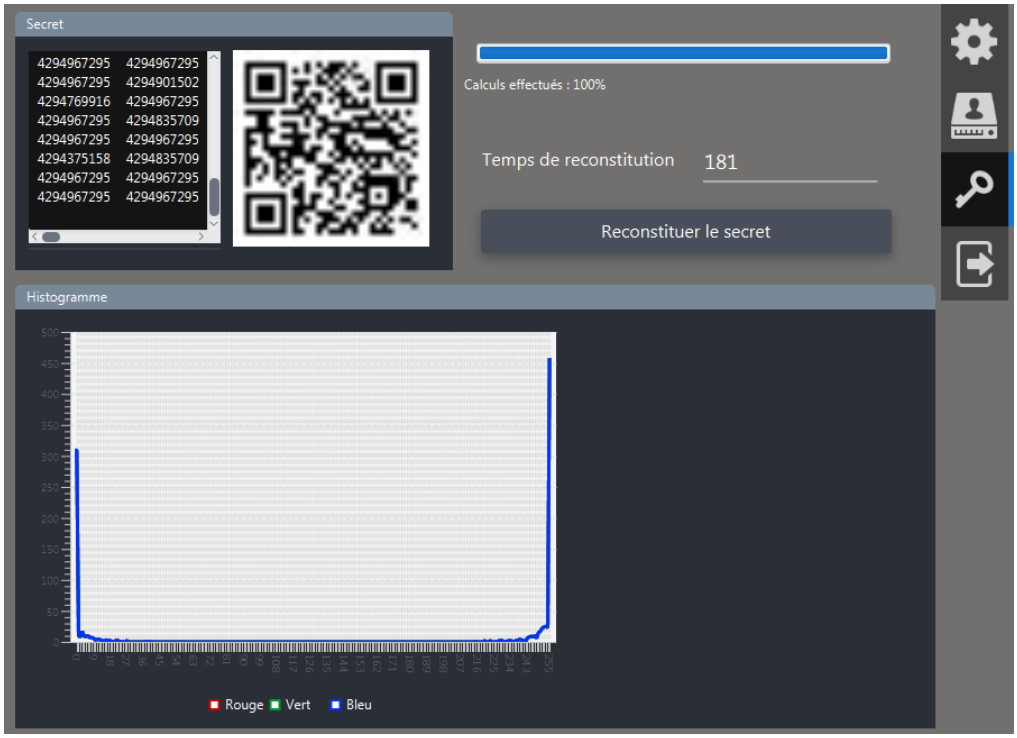


Figure V.15 – Lancement du serveur et connexion des candidats qualifiés

Pour tester la validité de cette opération, on a qu'à lire le code QR généré par notre application android (Figure V.16).

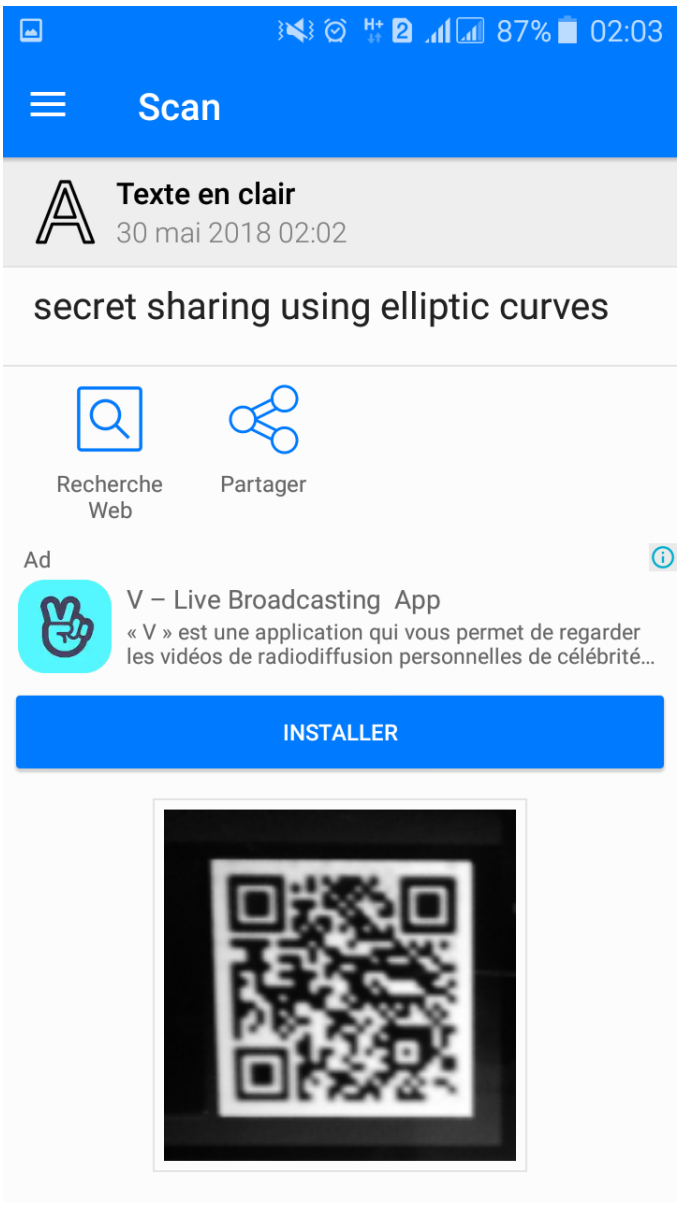


Figure V.16 – Lecture du code QR généré

V.4 Conclusion

Dans ce chapitre on a pu exposer l'approche proposée pour notre sujet d'étude, en présentant les formules théoriquement prouvées avec les différentes étape et outils requis pour leur implémentation. On a également présenté une analyse des résultats de quelques tests effectués, afin de prouver son exactitude et sa fiabilité.

CHAPITRE VI

Conclusion générale

Les cryptosystèmes basés sur le problème du logarithme discret elliptique se développent rapidement au cours de ces dernières années, tâchant de minimiser le temps d'exécution ainsi que la taille de la clé sans provoquer une perte au niveau de la sécurité du système, et ce grâce à la complexité de résolution du problème malgré la simplicité des opérations de base.

Dans ce modeste travail, nous avons abordé une des techniques de la cryptographie moderne représentée par le partage de secrets en utilisant les courbes elliptiques, plus précisément le problème du logarithme discret elliptique. Notre mémoire a été élaborée suivant un plan composé en quatre chapitres.

Le premier chapitre, consacré aux bases mathématiques requises explique les différentes opérations sur les courbes elliptiques ainsi que le problème du logarithme discret elliptique. Tandis que dans le deuxième chapitre, nous avions abordé quelques techniques de partage de secrets les plus connues et utilisées, ce qui nous a permis par la suite d'analyser quelques travaux de recherche dans le domaine qui composent notre état de l'art dans un troisième chapitre, grâce auquel on a enfin pu mettre en place notre propre approche dont on a exposé le côté théorique et pratique dans un dernier chapitre, et présenté également une analyse des résultats obtenus.

Les perspectives de notre travail visent à ajouter plus d'options au schéma proposé, en concevant un protocole proactif qui permettra de mettre à jour les partages d'une manière périodique. En outre, et du point de vue mathématique, nous aimerions entamer la notion de couplage par les courbes elliptiques pour pouvoir rivaliser les dernières recherches de la cryptologie en générale. Cependant, en ce qui concerne le côté pratique, nous prévoyons de perfectionner les résultats obtenus du point de vue temps d'exécution, en faisant appel à un langage plus léger et plus rapide que Java tel que le langage Python dont l'efficacité a été prouvée dans une variété de travaux dans le domaine de la sécurité.

Bibliographie

[1] Carlo Blundo, Alfredo De Santis, Giovanni Di Crescenzo, Antonio Giorgio Gaggia, and Ugo Vaccaro. Multi-secret sharing schemes. In *Annual International Cryptology Conference*, pages 150–163. Springer, 1994.

[2] Ilker Nadi Bozkurt, Kamer Kaya, and Ali Aydın Selçuk. Secret sharing for general access structures. In *4th International Conference on Information Security and Cryptology, Ankara, Turkey*, 2010.

[3] Kaya Bozkurt and Guloglu Selcuk. Threshold cryptography based on blakely secret sharing. *Information Sciences*, pages 1–4, 2008.

[4] Wang Caimei and Li Zhengmao. Self-selecting sub-secret keys sharing scheme based on polynomials over elliptic curve. In *Information Assurance and Security, 2009. IAS'09. Fifth International Conference on*, volume 2, pages 734–737. IEEE, 2009.

[5] Malkhi Dahlia. *Secret Sharing – Lecture Notes*. The Hebrew University of Jerusalem, 2002.

[6] Hisham Dahshan and James Irvine. An elliptic curve secret sharing key management scheme for mobile ad hoc networks. *Security and Communication Networks*, 4(12) :1405–1419, 2011.

[7] Wave Denso. Answers to your questions about the qr code. `http://www.qrcode.com/en/`.

[8] Lafon Guillaume. *Les courbes elliptiques pour les nuls*. Université de Caen,France, 2003.

[9] Yiliang Han, Xiaoyuan Yang, Jun Sun, and Delong Li. Verifiable threshold cryptosystems based on elliptic curve. In *Computer Networks and Mobile Computing, 2003. ICCNMC 2003. 2003 International Conference on*, pages 334–337. IEEE, 2003.

[10] Jeffrey Hoffstein, Jill Catherine Pipher, Joseph H Silverman, and Joseph H Silverman. *An introduction to mathematical cryptography*, volume 1. Springer, 2008.

[11] Sun Hua and Wang Aimin. A multi-secret sharing scheme with general access structures based on elliptic curve. In *Advanced computer theory and engineering (ICACTE), 2010 3rd international conference*, volume 2, pages V2–629. IEEE, 2010.

[12] Dabounou Jaouad. *Analyse numérique, Interpolation polynomiale*. Université Hassan Premier, 2014.

[13] Rouget Jean-Louis. *Polynomes d'interpolation de Lagrange*. maths-france, 2007.

[14] Manish Panchal Nisha Patel, Prakash D. Vyavahare. A novel verifiable multi-secret sharing scheme based on elliptic curve cryptography. In *The Tenth International Conference on Emerging Security Information, Systems and Technologies*. IARIA, 2016.

[15] NIST Nist. Recommended elliptic curves for federal government use, 1999.

[16] Christof Paar and Jan Pelzl. *Understanding cryptography : a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[17] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11) :612–613, 1979.

[18] Theon. Associativite addition courbe elliptique. `http://commons.wikimedia.org/wiki/File:Associativite_Addition_Courbe_Elliptique.svg`.

[19] W3C. Portable network graphics (png) specification (second edition). `http://www.w3.org/TR/PNG/index-noobject.html`.

[20] James Weaver. *JavaFX Script : Dynamic Java Scripting for Rich Internet/Client-Side Applications*. Apress, 2007.

[21] Wikipedia. Secret sharing. `http://en.wikipedia.org/wiki/Secret_sharing`.