MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE SAAD DAHLEB DE
BLIDA 1
Faculté des Sciences

Département d'Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER en informatique

THEME :

Comparaison des performances de déchiffrements RSA et ECC

Spécialité : Sécurité des systèmes d'informations

Réalisé par : Berrani Narimane

Promoteur : Mr Benyahia Mohamed
Jury : Mr kameche a Abdellah
Mr Oueld Aissa Ahmed

Année universitaire : 2022-2023

Remerciement

Nous tenons tout d’abord à remercier Allah le tout puissant et miséricordieux qui nous a donné la force et la patience d’accomplir ce modeste travail.

En second lieu, nous tenons à remercier très chaleureusement notre encadreur monsieur Benyahia Mohamed pour la confiance placé en nous et pour avoir accepté de diriger ce travail. Aussi, nos vifs remerciements aux membres du jury pour l’intérêt accordé à notre travail en l’examinant minutieusement et avec attention. Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenus pour la réussite dans nos études.

A nos familles et nos amis qui par leurs prières et leurs encouragements, on a pu surmonter tous les obstacles.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de cz travail.

Merci

Table de matière

Résume ……...……
Chapitre 1 la cryptographie
1.1 Introduction . . . . . .……..... . .1
1.2 Définition de la cryptographie . . . . . .…… 1
1.3 Terminologie . . . . . .……. . .. 1
1.4 Objectif de la cryptographie . . . . . .…….. 2
1.5 Classification de cryptographie . . . . . .…. ……. . . .2
1.5.1 Cryptographie symétrique . . . . . .…..……...3
1.5.1.1 Les catégories de la cryptographie symétrique . . . . . .…….. . …..3
1.5.1.1.1 Chiffrement par flux . . . . . .…….. . . 4
1.5.1.1.2 Chiffrement par bloc . . . . . .……. 4
1.5.1.2 Les algorithmes de la cryptographie symétrique . . . .…….. ……4
1.5.1.2.1 DES . . . . . .…... ……. . ..4
1.5.1.2.2 AES . . . . . .…….. . ..5
1.5.1.3 Avantages et inconvénients de la cryptographie symétrique . . ……. . . .5
1.5.2 Cryptographie Asymétrique . . . .……..…. 5
1.5.2.1 Principe de fonctionnement . . . . . .…….... 6
1.5.2.2 Caractéristiques de la cryptographie asymétrique . . . . . .……. . …... . 6
1.5.2.3 Les algorithmes de la cryptographie asymétrique . . . . . .…….. 7
1.5.2.3.1 RSA . . . . . .…….. 7
1.5.2.3.2 ECC . . . . . .……. 8
1.5.2.4 Les fonctions de hachage . . . . . .…..... 9
1.5.2.4.2 L’utilisation des fonctions de hachage cryptographique. . …….....9
1.5.2.5 Avantages et inconvénients de la cryptographie asymétrique . . ……. ..9
1. 6 Conclusion . . . .…. . . .……. . . 9
2Advanced Encryption Standard & Elliptic Curve Cryptography . . . . . .……..… .10
2.1 Introduction . . . .…..... 10
2.2 AES (Advanced Encryption Standard) . . . . . .…….. 10
2.2.1 Présentation de l’AES . . . . . .…….. . . …..10
2.2.2 Fonctions de transformation AES . . . .……....11
2.2.3 Les avantages et les inconvénients de l’AES . . . . . .…. . . . . .……...11
2.3 La cryptographie à base des courbes elliptique (ECC) . . . . . .……. . . .11
2.3.1 Généralités . . . . . .…… …. 11
2.3.1.1 Groupe . . . . . .……. …….11
2.3.1.2 Corps . . . . . .…... . . . . .…... . ……. . 12
2.3.2 Présentation des courbes elliptiques . . . . . .……. 13
2.3.2.1 Equation de Weierstrass . . . . . .…….. …… . 13
2.3.2.2 Définition des courbes elliptiques . . . . . .……. . . 14

3.3.2.3 Problème du logarithme discret sur les courbes elliptique . . ……. . 19
3.3.5.3 Des applications qui utilisent ECC . . . . . .……. . . . 23

3.4 Conclusion . . . . . .……..... . . …..... . . .25
Chapitre 3 implémentation et réalisation
3.1 Introduction……...…….27
3.2 Environnement de développement…….... . …..27
3.3 Ressource logiciel…….……...…….28
3.4 Présentation et interface…… . . . . ……..…..…28
3.5 Discussion…….....…….33
3.6Conclusion…….....33
Conclusion générale ……..…… . …. …….. ……...37
Bibliographie ……..…….…….....39

Tables de figure

Figure 1: une representation d'un cryptoststeme.....…….....14
Figure 2: les methodes de la cryptographie moderne….....15
Figure 3 : principe d'un chiffrement a clé secrete…….....…....15
Figure 4: schéma de chiffrement par flux……..…….17
Figure 5 :principe d'un chiffrement a clé publique……..……..17
Figure 5 :principe de fonctionement de hachage……...……...18
Figure 7 :hiearchie des opération sur les courbes elliptiques en cryptographie…...23
Figure 8 :exemples de courbe elliptique……..23
Figure 9 :addition des points sur des courbes…….……..25
Figure 10 :comparaison des tailles de clé entre ECC et RSA……26
Figure 11 : interface de cryptage…….37
Figure 12 :interface de cryptage de fichier……..……..38
Figure 13 :interfce pour choisir le fichier à crypter……..38
Figure 14 :code de cryptage…….……39
Figure 15 :resultat de cryptage……39
Figure 1 :interface de decryptage……40
Figure 17 :interface de selection le fichier à decrypte…….….41
Figure 18 :interface de selection de fichiers à decrypte……...41
Figure 19 :code de decryptage…….41
Figure 20 :resultat de decryptage…….42

Listes des tableaux

Tableau1 :avantage et inconvinients de chiffrement symetrique……18
Tableau 2 :avantage et inconvénients de chiffrement asymétrique……..…...33
Tableau 3 :comparaison entre ECC et RSA ……..…….33
Tableau 5 : comparaison des tailles de clé ECC et RSA……..26
Tableau 5: avantage et inconvénients ans la cryptographie asymétrique RSA et ECC……..…26
Tableau 7 : temps nécessaires pour crypté un fichier texte……..42
Tableau 8 : temps nécessaires pour décrypté un fichier texte…….43

Tables des acronymes

DES: Data Encryption Standard.
AES: Advanced Encryption Standard.
RSA: Rivest.R, Shamir.A, Adleman.L.
ECC: Elliptic Curve Cryptography.
ECDSA: Elliptic Curve Digital Signature Algorithm.

Résume

. Si le chiffrement est un moyen efficace de garantir la confidentialité, la signature numérique assure quant à elle l'authenticité des données. Son utilisation découle des procédés de la cryptographie asymétrique. Elle est une alternative crédible pour garantir l'authenticité, la non falsification, la non réutilisation, l'inaltérabilité et l'irrévocabilité des données. Le chiffrement des données et la signature électronique par la cryptographie sur les courbes elliptiques sont maintenant très répandus et il est important d'en saisir les différents avantages. Dans cet article, nous avons évalué les performances de la cryptographie asymétrique sur les courbes elliptiques par rapport à celle de l'algorithme RSA. Les cryptosystèmes élaborés à cet effet (en utilisant les protocoles ci-après : ECNR, ECDSA, ECIES et RSA) nous ont permis d'effectuer des tests qui ont montré que la plupart des algorithmes de cryptographie à courbe elliptique sont plus avantageux en termes de consommation de mémoire et de vitesse de calcul que le cryptosystème RSA.

ce travail est divisé en 3 chapitres et une annexe.

Le chapitre 1, présente les principes et notions de la sécurité et de la cryptographie, quelques algorithmes cryptographiques et la cryptanalyse en général.

Le chapitre 2, présente le fonctionnement de cryptosystème RSA et ECC, le principe de son chiffrement, du déchiffrement et de la signature, ainsi que la cryptanalyse de RSA et quelques attaques.

Le chapitre 3, présente la conception, explique et expose l'application réalisée dans ce sens, en montrant ses différentes fonctionnalités et les scénarios possibles.

Enfin nous terminerons par une conclusion générale.

Abstract

Encryption method is an effective way to guarantee the confidentiality and in other ways (digital signature) the authenticity of data. The use of the digital signature derives from the methods of asymmetric cryptography. It appears as credible alternatives to guarantee the authenticity, non forgery, non reuse, inalterability and irrevocability of data. Data encryption and electronic signature by elliptic curve cryptography are now widespread and it is important to highlight the comparative advantages. In this paper, we evaluated the performance of asymmetric cryptography through elliptic curve cryptography versus that with the RSA algorithm. Then, we achieved some cryptosystems by using elliptic curve cryptography protocols : ECNR, ECDSA, ECIES, and RSA. Therefore, we perform tests that showed most of the elliptic curve cryptography algorithms are more advantageous in terms of memory consumption and computing speed over the RSA cryptosystemIt is divided into 3 chapters and an appendix.

Chapter 1 presents the principles and concepts of security and cryptography, some cryptographic algorithms and cryptanalysis in general.

Chapter 2 presents the operation of RSA and ECC cryptosystem, the principle of its encryption, decryption and signature, as well as the cryptanalysis of RSA and some attacks.

Chapter 3, presents the design, explains and exposes the application carried out in this direction, showing its different functionalities and possible scenarios.

Finally, we will conclude with a general conclusion.

ملخص

في حين أن التشفير هو وسيلة فعالة لضمان السرية ، فإن التوقيع الرقمي
يضمن صحة البيانات. ينبع استخدامه من عمليات التشفير غير المتماثل. إنه بديل موثوق لضمان الأصالة وعدم التزوير عدم إعادة استخدام البيانات وعدم قابليتها للتغيير وعدم قابليتها للإلغاء. تشفير البيانات والتوقيع الإلكتروني عن طريق التشفير على منحنيات بيضاوية الانتشار الآن ومن المهم أن نفهم الفوائد المختلفة. في هذه الورقة ، قمنا بتقييم أداء التشفير غير المتماثل على المنحنيات الإهليلجية مقارنة بأداء الخوارزمية
، ECDSA، ECNR :أنظمة التشفير المطورة لهذا الغرض (باستخدام البروتوكولات التالية آر إس إيه
سمح لنا بإجراء اختبارات أظهرت أن معظم الخوارزميات (RSA و ECIES
تشفير المنحنى الإهليلجي أكثر فائدة من حيث استهلاك الذاكرة
RSA والسرعة الحسابية من نظام تشفير
الهدف من هذا العمل هو دراسة تحليل الشفرات.

RSA

وهي مقسمة إلى 3 فصول .
وملحق
يعرض الفصل 1 مبادئ ومفاهيم الأمن والتشفير، وبعض خوارزميات التشفير وتحليل الشفرات بشكل عام.

، ومبدأ تشفيره وفك تشفيره وتوقيعه ، بالإضافة إلى تحليل تشفير ECC و RSA يعرض الفصل 2 تشغيل نظام تشفير.
وبعض الهجمات RSA

الفصل 3، يعرض التصميم ويشرح ويعرض التطبيق الذي تم تنفيذه في هذا الاتجاه، موضحا وظائفه المختلفة والسيناريوهات المحتملة.

وأخيرا، سنختتم باستنتاج عام.

.

Chapitre 1

La cryptographie

1.1 Introduction :

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

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

1.2 Définition de la cryptographie :

Le terme cryptographie vient en effet de deux mots grecs : Kruptus qu'on peut traduire comme secret et Graphein pour écriture.
La cryptographie est l'art de cacher l'information pour qu'elle soit incompréhensible, elle désigne l'ensemble des techniques qui permettent de chiffrer les messages, son objectif principale est de permettre à deux personnes **Alice** et **Bob** de communiquer à travers un canal peu sécurisé de telle sorte qu'un opposant Eve ne puisse pas comprendre ce qui est échangé, on utilise une clé appelée clé de chiffrement pour le processus de chiffrement. Pour rendre l'information à nouveau compréhensible on utilise une clé appelée clé de déchiffrement pour le processus de déchiffrement. [1]

1.3 Terminologie [2] [3]

Les principaux termes utilisés dans la cryptographie sont :

Cryptologie : C'est une science mathématique regroupant la cryptographie et la cryptanalyse.
Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.
Cryptanalyse : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.
Cryptosystème : un Cryptosystème est constitué d'un algorithme cryptographique ainsi que toutes les clés possibles et tous les protocoles qui le font fonctionner.

Cryptogramme : Texte chiffré : Ciphertext : est le résultat de l'application d'un chiffrement d'un texte clair.
Texte clair : Plaintext : le message à chiffrer.
Chiffrement: la fonction permettant de transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a crée le message et ainsi que le
destinataire.
Déchiffrement : la fonction permettant de retrouver le texte clair à partir du texte chiffré.
Clé : une clé est un paramètre utilisé en entrée d'une opération cryptographique (chiffrement, Déchiffrement).On distingue généralement deux types de clefs :
Clés symétriques : il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou à clé secrète.
Clés asymétriques : il s'agit de clés utilisées dans le cas du chiffrement asymétrique ou à clé

publique. Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

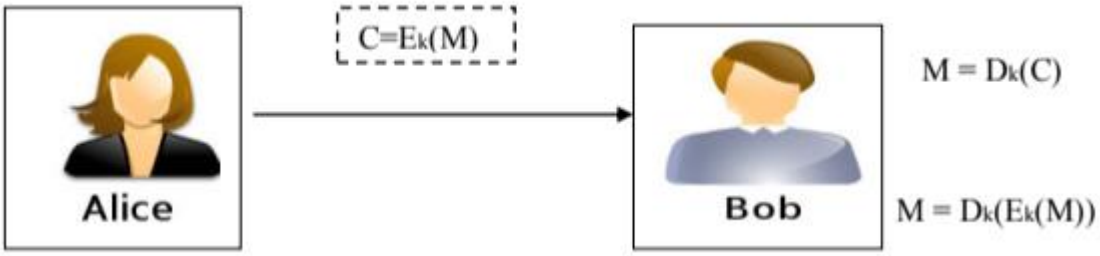


Figure 1 : Une représentation d'un Cryptosystème. sont [18]

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

1.4 Objectif de la cryptographie

Les principaux objectifs garantis par l'application de la cryptographie sont [4]:

- **La confidentialité**:
  Le message chiffré ne doit pas être compréhensible que par les destinataires légitimes. Il ne peut pas être déchiffré par un intrus.

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

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

- **La non répudiation:**
  Un expéditeur ne peut pas nier d'avoir envoyé un message et le destinataire ne peut pas nier de l'avoir reçu.

1.5 Classification de cryptographie

Dans la cryptographie moderne toute la sécurité est basée sur la clé et non dans les détails des algorithmes utilisés. On trouve principalement deux grandes familles de cryptographie moderne : la cryptographie symétrique ou à clé secrète et la cryptographie asymétrique ou à clé publique. [5]

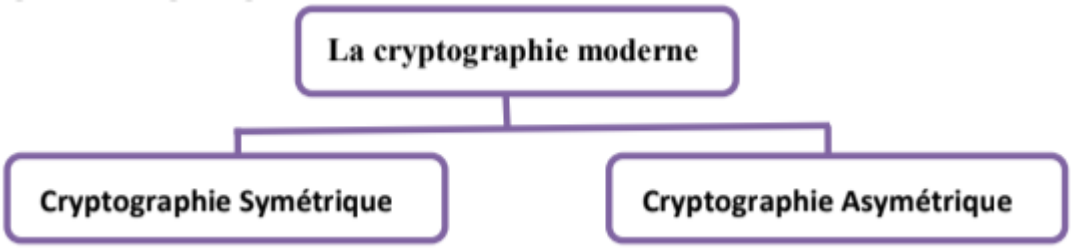


Figure .2: Les méthodes de la cryptographie moderne. [19]

1.5.1 Cryptographie symétrique

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

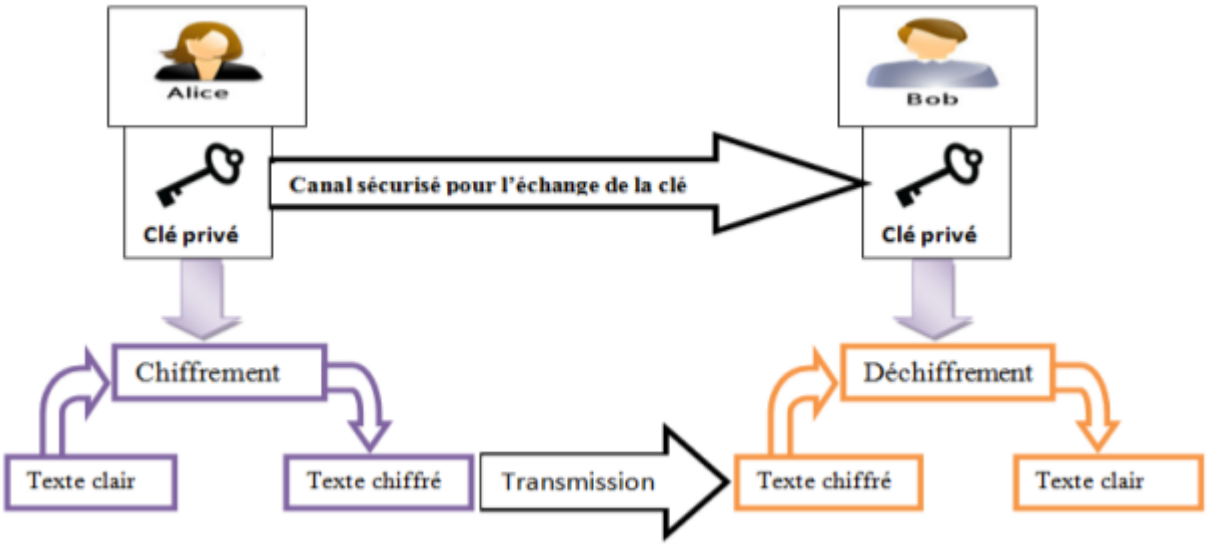


Figure 3 : Principe d'un chiffrement à clé secrète (symétrique). [21]

1.5.1.1 Les catégories de la cryptographie symétrique

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

1.5.1.1.1 Chiffrement par flux : [7] [8]

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

Figure 4 : Schéma de chiffrement par flux **[22]**

Exemples d'algorithmes de chiffrement par flux :

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

1.5.1.1.2 Chiffrement par bloc :

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

1.5.1.2 Les algorithmes les plus connus dans la cryptographie symétrique

1.5.1.2.1 DES

Jusqu'à l'introduction de la norme de chiffrement avancé (AES) en 2001, la norme de chiffrement des données (DES) était le schéma de chiffrement le plus utilisé. Il a été conçu dans les années 1970 par la firme américaine IBM pour être un système de chiffrement par blocs à clé secrète, et fut adopté comme standard de chiffrement en 1977. Comme illustre la figure 2.12, DES commence par une permutation IP de bits initiale à l'aide de la matrice de permutation.

DES traite des messages de 64 bits avec des clés de 56 bits, et il fournit un cryptogramme de 64 bits en sortie. Il consiste alors à réitérer la structure Feistel 16 tours. [9]

1.5.1.2.2 AES

AES est le chiffre le plus utilisé dans l'univers, ce chiffre symétrique est normalisé par le NIST en 2000 en remplacement du DES, de nos jours, les processeurs presque modernes intègrent des instructions AES telles que les processeurs Intel, les processeurs AMD, plutôt1 que de nombreux autres produits logiciels tels que OpenSSL.[10]

contrairement à la majorité des algorithmes de chiffrement asymétriques dont la sécurité repose sur des problèmes mathématiques difficiles tels que le logarithme discret dans le cas d'ECC, ou la factorisation entière dans le cas de RSA, AES tire sa force de la combinaison entre permutation et

substitution, plus communément connu sous le nom de réseaux de substitution-permutation (SPN), nous pouvons dire que l'AES est lui-même un problème difficile, car de nombreuses personnes qualifiées ont tenté de briser le cryptage AES et ont échoué. [11]

AES est un chiffrement par bloc symétrique prend une taille de bloc de texte en clair de 128 bits ou 16 octets. La longueur de clé peut être de 128, 192 ou 256 bits. [10]

1.5.1.3 Avantages et inconvénients de la cryptographie symétrique :

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

Table 1 : Avantages et Inconvénients de chiffrement symétrique.

1.5.2 Cryptographie Asymétrique

La cryptographie asymétrique, ou cryptographie à clé publique repose sur l'utilisation d'une clé publique (qui est diffusée) et d'une clé privée (gardée secrète), l'une permettant de chiffrer le message et l'autre de le déchiffrer. Ainsi, l'expéditeur peut utiliser la clé publique du destinataire pour chiffrer un message que seul le destinataire (en possession de la clé privée) peut le déchiffrer, garantissant la confidentialité du contenu. Inversement, l'expéditeur peut utiliser sa propre clé privée pour chiffrer un message, le destinataire peut déchiffrer avec la clé publique ; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message. Dans la cryptographie asymétrique impossible de trouver la clé privée à partir de la clé publique. [12]

1.5.2.1 Principe de fonctionnement

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

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

1.5.2.2 Caractéristiques de la cryptographie asymétrique

- DSK (EPK(M)) = M
- PK la clé publique.
- SK la clé privée secrète.

- La connaissance de PK ne permet pas de déduire SK.
- La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire.
- Les algorithmes de chiffrement Asymétrique se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le problème des logarithmes discrets (El Gamal, courbe elliptique), ou encore le problème du sac à dos (Merkle-Hellman). [14]

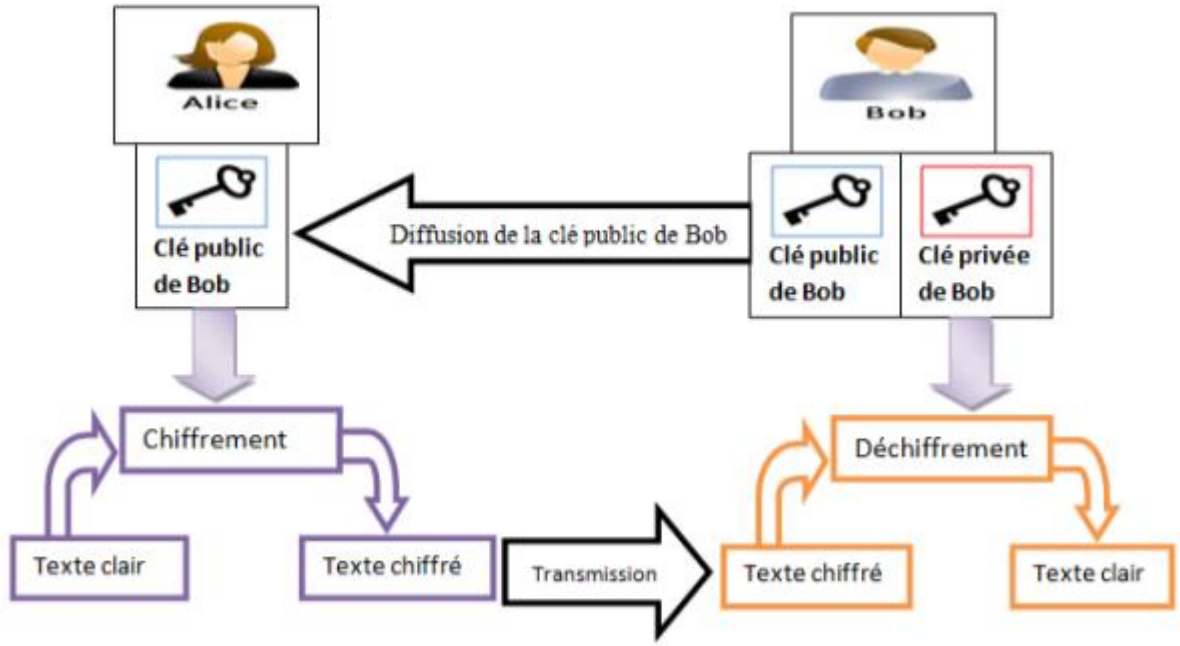


Figure 5 : Principe d'un chiffrement à clé publique (Asymétrique). [24]

1.5.2.3 Les algorithmes les plus connus dans la cryptographie asymétrique

1.5.2.3.1 RSA [9]

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

1.5.2.3.1 Principe de fonctionnement

 Générer aléatoirement deux nombres premiers p et q, puis les multiplier pour générer le nombre n.
 Déterminer **φ (n) : φ (n) = (p-1)*(q-1).**
 Choisir un entier naturel e premier tel que **1 < e <φ (n) et PGCD(e,φ (n))=1.**
 Calculer l'entier naturel d tel que **e*d modφ(n)=1.**
 Le couple **(e, n)** est la clé publique du chiffrement, alors que le couple **(n, d)** est sa clé privée.
 Pour chiffrer un texte, nous calculons c avec : **c=memod n.**
 Pour déchiffrer un texte chiffré, nous calculons **m** avec **: m=cdmod n.**

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

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

Algorithme 1 : Génération de clés avec RSA

1 Sortie : Clé publique (n, e) et clé privée d générées

2 **Début**

3 Sélection au hasard de deux nombres premiers p et q ;

4 Calculer n=p * q et φ (n) = (p-1)*(q-1) ;

5 Choisir un entier e tel que 1<e< φ (n) et le PGCD (e, φ (n))=1 ;

6 calculer d tel que 1<d<φ (n) et e*d _1 (modφ (n)) ;

7 **Fin**

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

Algorithme 2 : Chiffrement avec RSA

1 Entée :(n, e) et m // la clé publique et le texte clair m∈ [0,n-1] ;

2 Sortie : c //le texte chiffré ;

3 **Début**

4 Calculer c=memod n ;

5 **Fin**

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

Algorithme 2 : Déchiffrement avec RSA

1 Entée :(n, d) et c // la clé privé et le texte chiffré

2 Sortie : m // le texte clair ;

3 Début

4 Calculer m=cdmod n ;

5 Fin

1.5.2.3.2 ECC

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

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

Nous donnons un exemple, considérons le groupe E23 (9,17), c'est le groupe défini par l'équation y2mod 23 = (x3 + 9x + 17) mod 23. Quel est le logarithme discret k de Q = (4, 5) à la base P=(16,5) ? La méthode par force brute consiste à calculer des multiples de P jusqu'à ce que Q soit trouvé.

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

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

Dans le prochain chapitre, nous allons bien expliquer la cryptographie à base des courbes elliptique. **[15]**

1.5.2.4 Les fonctions de hachage

Une bonne cryptographie doit pouvoir offrir une garantie de l'intégrité des informations. En effet, il ne doit pas être possible de pouvoir modifier les

informations cryptées de façon totalement transparente, un processus de vérification de l'intégrité du message doit être mise en place, Ce processus est réalisé par une fonction de hachage.[28]

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

Figure 6 : principe des fonctions de hachage. [27]

1.5.2.4.2 L'utilisation des fonctions de hachage cryptographique :

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

- ✓ Authentification des messages: est obtenue à l'aide du code d'authentification des messages.
- ✓ Signatures numériques: la valeur de hachage du message est chiffrée avec la clé privée de l'utilisateur, afin que le destinataire connaisse la clé publique de l'utilisateur, le destinataire puisse

vérifier l'intégrité du message.

- ✓ Fichiers de mots de passe unidirectionnels: dans la plupart des cas, la valeur de hachage des mots de passe est stockée au lieu du mot de passe lui-même, cela protège les mots de passe même si quelqu'un accède aux fichiers qui contiennent des mots de passe.
- ✓ Détection de virus: une valeur de hachage de chaque fichier est stockée en toute sécurité, plus tard, il est possible de déterminer si un fichier a été modifié ou non en recalculant simplement la valeur de hachage du fichier et en comparant la nouvelle valeur avec celle stockée en toute sécurité.

1.5.2.5 Avantages et inconvénients de la cryptographie asymétrique : [17]

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

Table .2 : Avantages et Inconvénients de chiffrement asymétrique.

1. 6 Conclusion :

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

Chapitre 2

Le cryptosystème RSA

2.1 Introduction :

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

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

2.2 AES (Advanced Encryption Standard) [4]:

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

2.2.1 Présentation de l'AES

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

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

Promu par le gouvernement américain, à devenir un standard pour l'échange sécurisé des informations, aux États-Unis et entre les États-Unis et leurs partenaires. Le champ d'application de l'AES a évolué et devient, à compter du premier octobre 2015, l'algorithme de chiffrement des informations jusqu'au niveau TOP SECRET aux États-Unis. De même, il est, aujourd'hui, l'algorithme symétrique de chiffrement par bloc le plus couramment utilisé en occident. L'AES est un algorithme de chiffrement symétrique par bloc. Il chiffre et déchiffre des blocs de données à partir d'une seule clef. Contrairement au DES, basé sur un réseau de Feistel, l'AES s'appuie sur un réseau de substitutions et de permutations (SP-network). Ce dernier est constitué de fonctions de substitutions non linéaires contenues dans une S-Box et de fonctions de permutation linéaire.

Chaque boîte prend un bloc de texte et la clé en entrée puis fournit un bloc de texte chiffré en sortie.

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

Pour son fonctionnement interne, chaque bloc de données est organisé en tableau de quatre colonnes et quatre lignes, chaque case contenant un octet, soit $4 \times 4 \times 8 = 128$ bits par tableau.

Les opérations de chiffrement et de déchiffrement sont effectuées sur ce tableau, puis, le résultat est copié dans un tableau de sortie. [29]

2.2.3 Les avantages et les inconvénients de l'AES

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

- **Les avantages**

- ✓ Un chiffrement rapide en termes de temps d'exécution
- ✓ Fournit un très bon niveau de sécurité.
- ✓ Ses besoins en ressources mémoires sont également très faibles.
- ✓ Utilise peu de ressources systèmes.

Les inconvénients

- ✓ Problème d'échange de la clé secrète.
- ✓ Problème de distribution des clés n (n-1)/2 Clés pour n partenaires.
- ✓ Tables statiques : SBOX INVSBOX RCON prédéfinie.

2.3 la cryptographie à base des courbes elliptique (ECC)

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

Un système concurrent met au défi RSA: la cryptographie à courbe elliptique (ECC).

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

2.3.1 Généralités

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

2.3.1.1 Groupe

En mathématique, un groupe est un couple (E, ·) où E est un ensemble et · est une loi de composition interne qui combine deux éléments a et b de E pour obtenir un troisième élément

$a \cdot b$. Il faut que la loi satisfasse les quatre axiomes ci-dessous. [30]

- Fermeture :$\forall (a, b) \in E : a \cdot b \in E$
- Associativité : $\forall (a, b) \in E : (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- Élément neutre : $\exists e \in E : a \cdot e = e \cdot a = a$
- Symétrique : $\forall a \in E, \exists b \in E : a \cdot b = b \cdot a = e$
- Groupe abélien

Un groupe abélien, est un groupe dont la loi de composition interne est commutative.

Un ensemble E est un groupe commutatif lorsque. [30]

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

 Groupe cyclique

Un groupe fini G est cyclique si tout élément du groupe peut s'exprime sous forme d'une puissance ou d'un multiple d'un élément particulier g, appelé le générateur du groupe, c'est-à-dire G =>g<= {gn| n ϵZ*}. Par exemple si G = {g0, g1, g2, g3, g4, g5} et g6 = g0, alors G est un groupe cyclique. Tout groupe cyclique est abélien car gngm = gn+m = gm+n = gmgn

. L'ordre d'un élément e d'un groupe cyclique est le nombre entier n positif le plus petit tel que ne = 0 (en notation additive) ou en = 1 (en notation multiplicative). Reprenons le même groupe G du paragraphe précédent, par exemple l'ordre l'élément g2 est 3 car l'élément neutre du groupe est g0 = 1 et (g2)3 = g6 = 1. [31]

2.3.1.2 Corps

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

- Le couple (E, +) forme un groupe abélien, il existe un élément neutre, noté 0, tel que ∀a ϵE :a + 0 = 0 + a = a.
- Le couple (E\{0}, ·) forme aussi un groupe abélien dont l'élément neutre est 1 : ∀a ϵE :a · 1 = 1 · a = a.
- La multiplication · est distributif pour l'addition, c'est-à-dire : ∀ (a, b, c) ϵE| a · (b + c) = a · b + a · c et (b + c) · a = b · a + c · a.

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

 Corps finis

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

 Corps premier

Un corps est un corps premier, noté Fp lorsque l'ordre du corps q = pn et p est un nombre premier. Le corps est constitué des nombres entiers {0, 1, 2, ..., p − 1}, et ∀a ϵZ, a mod p donne le reste unique r qui est compris entre [0, p − 1]. [30]

 Corps binaire

Un corps fini de l'ordre 2n est un corps binaire, noté F2 n , qui peut être construit en utilisant une représentation polynomiale. Les éléments du corps sont des polynômes binaires dont les coefficients aiϵ {0, 1} et les degrés sont inférieurs à n. C'est-à-dire : F2

n = {an−1zn−1 + an−2zn−2 + ... + a1z + a0 : aiϵ {0, 1}}.

2.3.2 Présentation des courbes elliptiques

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

pouvons effectuer sur elles.

2.3.2.1 Equation de Weierstrass [32]

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

E : F (x ; y ; z) = y2z + a1xyz + a3yz2 – x3 – a2x2z – a4xz2-a6z3

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

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

E : Y2+a1XY + a3Y = X3 + a2X2 + a4X + a6.

2.3.2.2 Définition des courbes elliptiques

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

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

Y2+a1XY + a3Y = X3 + a2X2 + a4X + a6

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

Δ = -d22d8 – 8d43- 27d62 + 9d2d4d6

d2= a12+ 4a2

d4 = 2a4+ a1+a3

d6= a32+4a6

d8= a12+4a2a6 – a1a3a4+ a2a32 – a42

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

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

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

E : Y2= X3 + AX + B

Où les constantes A et B doivent satisfaire :

4A3+ 27B2≠ 0

La courbe elliptique E est l'ensemble des points (x,y) єk2 satisfaisant cette équation.[30] La forme de la courbe peut varier en fonction des paramètres choisis, dans la figure 3.9 nous avons deux exemples des courbe elliptiques.

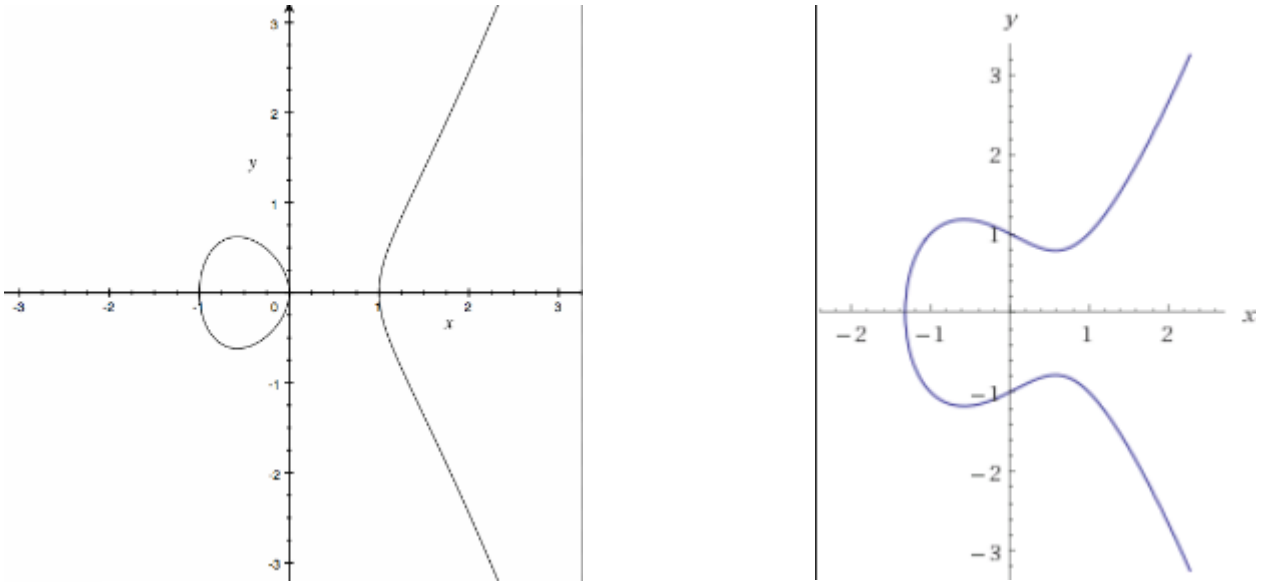


Figure 7. Exemples de courbes elliptiques.

Dans le domaine cryptographique, nous utilisons les courbes elliptiques qui sont définies dans un corps fini dont l'ordre q. ce cors peut être soit premier, soit binaire et le choix de corps n'a pas une influence importante sur la performance du cryptosystèmes. Dans la littérature, il existe différents algorithmes et techniques qui nous permettent d'optimiser les performances de calcul sur les courbes qui sont définies sue les 2 types de corps. Cependant pour une raison de simplicité d'implémentation et de présentation, durant les travaux de recherche de cette mémoire, nous avons utilisé que des courbes qui sont définies sur un corps premier.

L'équation Weierstrass d'une courbe elliptique peut être simplifiée, si la courbe est définie sur un corps premier F dont les caractéristique est différente de 2 et 3. Nous pouvons transformer la formule 1.1 à l'équation de Weierstrass simplifiée 1.2.

$$E/F_{2^m} : y^2 + xy = x^3 + ax^2 + b \quad (1.2)$$

C'est aussi la forme de courbe que nous avons utilisons dans la suite de cette mémoire.

2. 6 Opération arithmétique de ECC

Pour obtenir le point symétrique de P, il suffit de changer le signe de sa coordonnée y, si $P = (x, y)$, alors $-P = (x, x+ y)$ et $P + (-P) = \infty$.

Le calcul de l'addition et le doublement de point est montré dans les formules 1.3 et 1.4. Supposons que $P = (x1, y1) \in E$, $Q = (x2, y2) \in E$ et $P, \pm Q$, alors $P + Q = (x3, y3)$ où

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a \quad \text{et} \quad y_3 = \lambda(x_1 + x_2) + x_3 + y_1$$

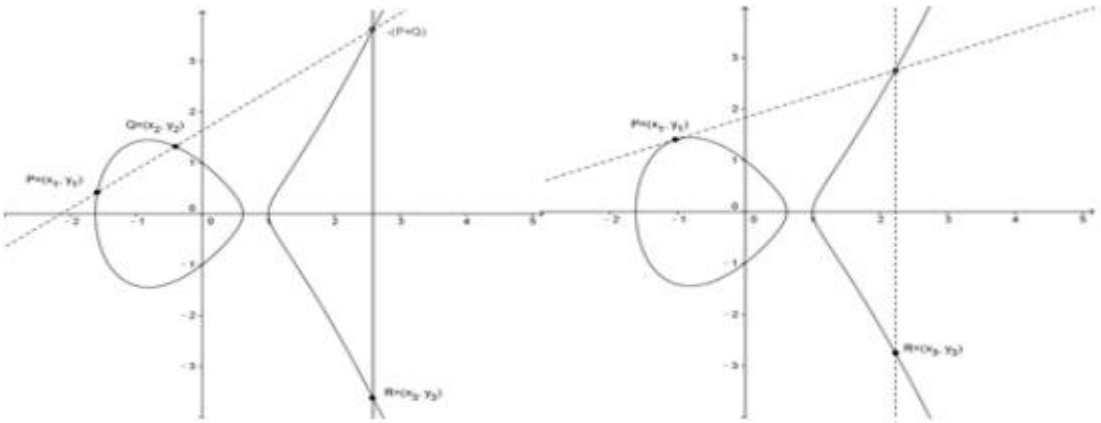
$$\lambda = (y_1 + y_2)/(x_1 + x_2) \qquad (1.3)$$

Si $P = (x1, y1) \in E$ et $P \neq -P$, alors $2P = (x3, y3)$ où

$$x_3 = \lambda^2 + \lambda + z = x_1^2 + b/x_1^2 \quad \text{et} \quad y_3 = x_1^2 + \lambda x_3 + x_3$$

$$\lambda = x_1 + y_1/x_1 \qquad (1.4)$$

Une représentation géométrique est donnée la figure pour additionner les points P et Q, nous tranchons une droite qui passe par ces deux points, le résultat de l'addition est le point symétrique par rapport à l'axe abscisse du 3eme point d'intersection avec la courbe. Pour doubler le point P, il suffit de trouver la tangente à la courbe au point P et le résultat du doublement est le point symétrique par rapport à l'axe abscisse de deux points d'intersection avec la courbe.



Addition : $P + Q = R$ Doublement : $P + P = R$

Figure 8.addition des points sur des courbes elliptiques. [30]

2.3.2.3 Equivalence entre RSA et ECC

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

2.4 Comparaison de performance entre RSA et ECC

RSA fait partie des premiers cryptosystème asymétriques qui sont largement appliqués et aujourd'hui, il est toujours considéré comme le cryptosystème asymétrique le plus utilisé, notamment pour échanger des données confidentielles sur internet. Il est proposé en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman. L'algorithme est basé sur l'hypothèse qu'il est extrêmement difficile d'effectuer la factorisation d'un nombre entier en facteurs premiers. Certes, la factorisation n'est pas la seul méthode pour casser RSA, il n'existe pas d'autre attaque qui soit suffisamment efficace.

RSA est proposé avec trois algorithmes qui sont conçus respectivement pour la génération des clés, la signature numérique et la vérification de signature.

ECC peut avoir le même niveau de sécurité que RSA avec une clé beaucoup plus courte. Les longueurs de clé utilisées sont suggérées dans le tableau.

Pour être honnête nous avons comparé seulement la taille des clés, il y'a d'autre facteur de comparaison. [26]

Bits de sécurité	RSA	ECC
80	1024	10
112	2048	224
128	3072	256
192	7680	384
256	15360	512

Tableau 3. Permet de comparer facilement les tailles de clé entre ECC et RSA.

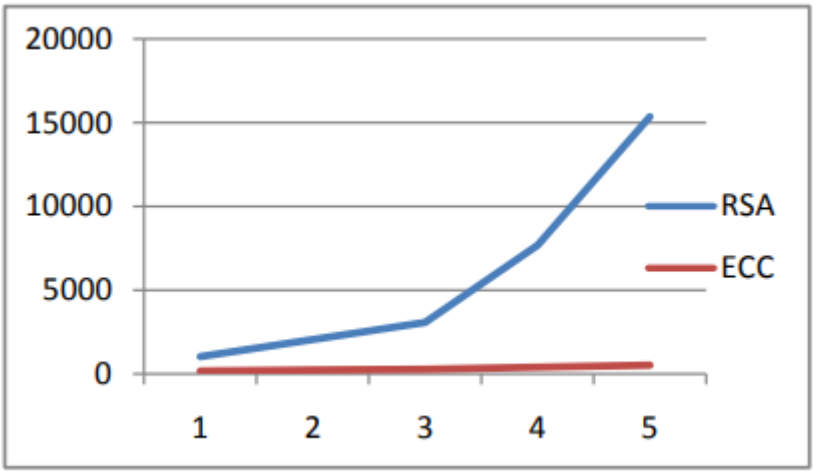


Figure 9. Comparaison des tailles de clé entre ECC et RSA.

	Asymétrique (RSA)	Asymétrique (ECC)
Avantages	- Cryptosystème largement répandu - Nombreuses études au sujet de sa sécurité	- Taille de clé inférieure pour une sécurité égale - La taille des clés croit moins vite que le RSA si on souhaite une meilleure sécurité - Utilisation pour systèmes embarqués - Calculs moins lourds que l'exponentiation - Utilisation mémoire moindre - Cryptanalyse par algorithme exponentiel
Inconvenients	- Opérations de dé/chiffrement très inégales en termes de temps de calcul - Cryptanalyse par algorithme sous exponentiel	- Complexe - Peu de développement sur des systèmes à grande échelle (mais tend à changer) - Travaux d'optimisation essentiellement destinés aux systèmes mobiles

Tableau 4. Avantage et inconvénients dans la cryptographie à clé asymétrique dans le cas de RSA et ECC.

2.3.5.3 Des applications qui utilisent ECC

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

 TLS Le protocole de sécurisation d'échange sur internet utilise la ECC dans la phase "Handshake" pour l'agrément sur la clé à utiliser pour chaque session. [32]

 Smart Cards A coté de RSA, les nouvelles smart-card support la ECC.

 Passeport biométrique La ECC est utilisée pour construire la clé de session.

 Bitcoin : ECC est utilisée pour la signature Transport Layer Security électronique pendant les transactions des paiements avec Bitcoin, la courbe utilisée par Bitcoin c'est Secp256k1 avec l'équation :y2 = x3 + 7 [38].

2.8 Conseils d'utilisation du RSA

Pour garantir une sécurité adéquate lors du chiffrement RSA, il est essentiel de respecter plusieurs règles :[27]

Éviter le chiffrement de blocs trop courts, qui peuvent être vulnérables à des attaques.

- Utiliser des nombres premiers de grande taille pour p et q, afin que leur produit (p-1) x (q-1) soit également grand.
- Utiliser des valeurs de N très grandes pour compliquer la tâche des attaquants.
- Éviter l'utilisation de la même valeur de N pour plusieurs clés RSA, car cela peut faciliter les attaques en partageant des informations entre les clés.
- Si une paire (N, d) est compromise, il est important de ne plus l'utiliser Pour la cryptographie RSA. sont [37]:

2.9 Catégories d'attaques sur RSA

La communauté cryptographique étudie la solidité du RSA face à une panoplie d'attaques. Ces attaques ont été réparties en trois 03 catégories selon la faille qu'elles exploitent : [30]

2.8.1Attaques mathématiques

Les attaques mathématiques sont des attaques qui cherchent à trouver une faille dans les fondements mathématiques mêmes de l'algorithme.

- ***Attaque de Wiener***

Proposée par le cryptologue Michael J. Wiener. Cette attaque permet de retrouver facilement la clé privée à partir de la clé publique (, N) , lorsque les conditions(d trop petit) et $q<p<2q$ (p et q trop proches) sont remplies, il est facile de retrouver d. $\frac{1}{3} N^{\frac{1}{4}}$

Cette attaque a été améliorée par Dan Boneh et Glen Durfee pour tous les exposants d inférieurs ou égaux à $N^{0.292}$. [29]

- ***Factorisation de modulo N***

Le RSA est basé sur la difficulté de factoriser un entier ayant plusieurs milliers de chiffres décimaux, on peut facilement démontrer qu'un tel nombre est composé, pour autant les méthodes utilisées ne fournissent en général aucune information sur ses diviseurs premiers. Il existe certaines méthodes permettant de les déterminer, nous

allons décrire l'une de ces méthodes qui est la division successive.

La méthode des divisions successives, qui est la plus simple, consiste à déterminer les diviseurs premiers de N plus petits que $\sqrt{N}$, ensuite il suffit de diviser N successivement par tous les nombres premiers retrouvés jusqu'à atteindre p, qui est le plus petit diviseur premier de N ($\leq \sqrt{N}$) .

Cette méthode est une technique d'attaque par force brute, bien qu'elle ne permette pas de factoriser des entiers n'ayant que des grands facteurs premiers et par rapport aux autres méthodes elle est inefficace, elle est néanmoins incontournable, et c'est la première méthode que l'on doit essayer afin de factoriser N. [28]

➢ ***Attaque en connaissant quelques bits de la clé privée***

Une autre technique d'attaque apparait lorsqu'on possède quelques bits de la clé privée d. D. Boneh et al ont montré que si la taille de la clé d est de k bits alors la connaissance de $\frac{k}{4}$ bits de poids faible est largement suffisante pour récupérer la clé d. [31]

2.8.2 Attaques de protocole

Même si RSA est solide, la façon dont on l'utilise n'est pas neutre. Par exemple si on envoie le même message à 3 personnes différentes, chiffrés avec 3 clés RSA de ces personnes, on peut facilement retrouver le message en clair à partir des 3 messages chiffrés en utilisant la propriété de multiplicativité de la fonction RSA **:** $f(x \times y) = f(x) \times f(y)$.
Il est également risqué de chiffrer plusieurs messages liés au moyen de la même clé publique RSA. [32]

2.8.3 Attaques physiques

Si l'on suppose que la fonction RSA est mathématiquement solide, on peut alors construire des protocoles surs pour le chiffrement et qui admettent des preuves de sécurité. Malgré ces preuves, certaines attaques restent possibles. Celles-ci ne s'attaquent pas au problème mathématique, mais elles utilisent le fait que le dispositif électronique qui fait les calculs de chiffrement, n'est pas une abstraction mathématique : dans le monde physique, il met un certain temps à calculer, il consomme du courant électrique, il peut faire des erreurs de calcul…etc. Avec ces paramètres supplémentaires, l'attaquant parvient parfois à ses fins. [35]

➢ ***Attaque sur le temps de calcul***

L'algorithme de calcul de la fonction RSA est toujours le même. En particulier il fait intervenir une boucle pour le calcul de $C^d mod\ N$ (ou d est la clé secrète).d =
$d_1, d_2, d_3, \ldots, d_n$, chaque bit $d_i$est égal à 0 ou 1. Or dans la boucle de calcul, on trouve une instruction du type 'si d_i = 1 alors faire tel calcul sinon ne pas le faire'.
En mesurant les temps de calcul pour de nombreuses valeurs initiales de C

(message crypté), on peut déduire si le calcul qui est fait lorsque $d_i = 1$ a été réellement effectué et de retrouver la clé secrète d bit par bit. On peut contourner ce type d'attaque en masquant les différences de temps de calcul. [35]

➢ ***Attaque sur la consommation électrique***

Le même type d'attaque peut être effectué avec la consommation électrique pour chacun des calculs, on mesure la courbe de consommation électrique du composant qui fait le calcul. En analysant la statistique de la consommation électrique à chaque étape du calcul, on déduit au fur et à mesure la valeur de la clé secrète d.
Des méthodes existent pour fausser la consommation électrique et rendre cette information inutilisable. [35]

➢ ***Attaque par injection de faute***

Les attaques par faute sont une famille de techniques qui consistent à produire volontairement des erreurs dans le cryptosystème. Ces attaques peuvent porter sur des composants matériels ou logiciels. Elles ont pour but de provoquer un comportement inhabituel des opérations cryptographiques dans le but d'en extraire des informations secrètes (comme une clé de chiffrement). Une attaque par faute peut être couplée à d'autres attaques comme l'attaque sur la consommation électrique et l'attaque sur le temps de calcul .

Les attaques sont possibles sous l'hypothèse que l'attaquant peut affecter l'état interne du système en écrivant des valeurs par exemple en mémoire ou sur un bus informatique. [35]

2.10 Outils mathématiques

2.9.1 Factorisation des entiers

La factorisation des entiers est un problème crucial en cryptographie, car elle permet de casser le RSA. La méthode la plus élémentaire pour factoriser un entier N consiste à prendre tous les entiers inférieurs à N, et à tester s'ils divisent N (algorithme de force brute). C'est bien sûr un algorithme inutilisable si N est grand. Un premier raffinement consiste à ne prendre que les entiers inférieurs à racine de N (si N n'est pas premier, N admet forcément un diviseur inférieur à racine de N). C'est beaucoup mieux, mais encore insuffisant pour les entiers de 1000 chiffres que l'on souhaite factoriser.

L'idée utilisée par les algorithmes modernes est due à l'arithméticien français Pierre de Fermat : [36]

si on trouve deux entiers x et y, non égaux, non opposés, tels que $x^2 = y^2 \ mod\ N$, alors$(x - y)(x + y) = 0 \ mod\ N$, et $pgcd\ (x + y, N)$ou $pgcd\ (x - y, N)$ donne un diviseur non trivial de N Il reste à trouver de tels nombres x et y. Posons x un nombre juste supérieur à racine de N. x^2 est juste supérieur à N, et $x^2 = a \ mod\ N$, avec a petit. Il est donc facile de factoriser a, et avec un peu de chances, en essayant plusieurs x, on peut espérer trouver un a tel que $a = y^2$.

Exemple : Soit à factoriser $N = 3337$, sa racine carrée vaut 57. On teste :

– $58^2 = 27 = 3^3 \ mod\ 3337$: ne convient pas.
– $59^2 = 144 = 12^2 \ mod\ 3337$: convient !

Alors, (59 + 12,3337) = 71 donne un diviseur non trivial de N. Le second facteur premier est 49.

2.9.2 Petit théorème de Fermat amélioré

Nous connaissons le petit théorème de Fermat :

Théorème: (Petit théorème de Fermat).
[37] Si p est un nombre premier,
Si a est un nombre premier avec p (c'est-à-dire que $(a, p) = 1$) alors

$$a^{p-1} \equiv 1 \ mod\ p$$

Théorème: (Petit théorème de Fermat amélioré). [37] Soient p et q deux nombres premiers distincts et soit $N = pq$. Pour tout $a \in Z$ tel que $(a, N) = 1$ alors :

$$^{(p-1)(q-1)} \equiv 1 \ mod\ N$$

on note $(N) = (p-1)(q-1)$ la fonction d'Euler l'hypothèse $pgcd(a, N) = 1$ équivaut ici à ce que a ne soit divisible ni par p, ni par q.

Par exemple pour $p = 5$ et $a = 7$:
$(N) = 4 \times 6 = 24$ Alors pour $a = 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, ...$ on a bien :

$$a^{24} \equiv 1 \ mod\ 35$$

Démonstration : Notons $C \equiv {}^{(p-1)(q-1)}$ Calculons C modulo p:

$$C \equiv {}^{(p-1)(q-1)} \equiv (a^{(p-1)})^{(q-1)} \equiv 1^{(q-1)} \equiv 1\ mod\ q$$

Où l'on a appliqué le petit théorème de Fermat : ${}^{(p-1)} \equiv 1\ mod\ p$ car p ne divise pas a. Calculons ce même C mais cette fois modulo q :

$$C \equiv {}^{(p-1)(q-1)} \equiv (a^{(q-1)})^{(p-1)} \equiv 1^{(p-1)} \equiv 1\ mod\ p$$

Où l'on a appliqué le petit théorème de Fermat : ${}^{(q-1)} \equiv 1\ mod\ q$ car q ne divise pas a. Conclusion partielle : $C \equiv 1\ mod\ p$ et $C \equiv 1\ mod\ q$.
Nous allons en déduire que $C \equiv 1\ mod\ pq$.
Comme $C \equiv 1\ mod\ p$ alors il existe $\alpha\ \epsilon\ Z$ tel que $C = 1 + \alpha$. [37]

2.9.3 Algorithme d'Euclide

L'algorithme d'Euclide sur deux nombres entiers positifs a et b avec $a > b \geq 0$ procédé comme suit :

Si $b = 0$ l'algorithme termine et rend la valeur a ;

Sinon l'algorithme calcule le reste r de la division euclidienne de a par b puis recommence avec $a \coloneqq b$ et $b \coloneqq r$.

Formellement l'algorithme d'Euclide construit deux suites finies d'entiers par récurrence : la suite (q_n) des quotients et la suite (r_n) des restes :

$r_0 =$, $r_1 = b$;

Pour $n \geq 1$, r_{n+1} et $q_n$sont le reste et le quotient de la division euclidienne de r_{n-1} par r_n , Définis par les deux conditions :

$r_{n-1} = r_n q_n + r_{n+1}$ et $0 \leq r_{n+1} < r_n$.

Par définition la suite r_n est une suite strictement décroissante d'entiers positif : elle est donc finie et s'arrête au premier n tel que $r_n = 0$. [38]

Exemple

Soit à calculer le $pgcd$ de 21 et 15.

On réalise la division euclidienne de $a = 21$ et $b = 15$,

Le quotient est 1 et le reste 6.L'algorithme continue avec a: = b est b: = le précédent reste, jusqu'à trouver un reste nul. L'algorithme s'arrête alors est retourne le dernier reste non nul trouvé, Ici $r_3 = 3$ qui est bien le $pgcd$ de 21 et 15.

2.9.4 L'algorithme d'Euclide étendu

L'algorithme d'Euclide étendu est une variante de l'algorithme d'Euclide qui permet, à partir de deux entiers a et b, de calculer non seulement leur plus grand commun diviseur ($pgcd$), mais un couple de coefficients de Bézout, c'est-à-dire deux entiers u et v tel que

$au + bv = pgcd(a, b)$.

Cet algorithme est particulièrement utilisé lorsqu'on souhaite calculer l'inverse multiplicatif d'un entier. La question importante est comment calcule-t-on les coefficients u et
v. L'idée principale de l'algorithme est d'effectuer les mêmes étapes que pour l'algorithme d'Euclide, mais en exprimant à chaque itération le reste comme une combinaison linéaire de a et b. Puisque le dernier reste est le $pgcd$, celui-ci sera alors exprimé comme une combinaison linéaire de a et b. [39]

2.10 Impacts des attaques RSA

Les attaques RSA peuvent avoir un impact significatif sur la sécurité des communications qui utilise cet algorithme de chiffrement. Si un attaquant a réussi à casser la clé privée RSA, il peut déchiffrer tous les messages chiffrés avec la clé publique correspondante, ce qui peut porter atteinte à la confidentialité des données sensibles.

De plus, l'attaque de Hastad, qui est une attaque par canal auxiliaire, peut être utilisée pour récupérer le message en clair lorsque le même message est chiffré avec plusieurs clés publiques différentes. Cela peut être particulièrement dangereux si des messages sensibles sont envoyés à plusieurs destinataires à la fois.

En général, les attaques contre RSA peuvent affaiblir la confiance dans la sécurité des systèmes de cryptographie à clé publique, ce qui peut avoir des répercussions sur les secteurs tels que les services financiers, les systèmes de paiement en ligne et les communications sécurisées. Il est donc important de prendre des mesures pour renforcer la sécurité de l'algorithme RSA et des systèmes qui l'1utilisent.

2.12 Limites des attaques RSA :

Les attaques RSA ont des limites qui sont principalement dues à la taille des clés utilisées. Les attaques de force brute sont pratiquement impossibles car la complexité de l'algorithme de factorisation augmente de façon exponentielle avec la taille de la clé. Les attaques par faute et par canal auxiliaire nécessitent un accès physique à l'appareil et sont donc difficiles à réaliser. Les attaques par analyse de temps et par analyse de consommation d'énergie nécessitent des équipements spécialisés et sont également difficiles à réaliser. Enfin, les attaques en connaissant quelques bits de la clé privée ou les attaques par oracle nécessitent un accès au système ou un comportement particulier de celui-ci, ce qui les rend également difficiles à mettre en œuvre. Cependant, la sécurité de RSA dépend également de la qualité de l'implémentation et de la gestion des clés, ce qui peut introduire des vulnérabilités potentielles.

2.13 Conclusion :

Dans ce chapitre, nous avons mis l'accent sur l'algorithme RSA, nous avons détaillé son fonctionnement, puis nous nous sommes intéressés à sa cryptanalyse et ses trois catégories d'attaques qui sont les attaques mathématiques, de protocole et attaques physiques, et nous avons introduit les outils mathématiques utilisés pour la simulation de nos deux attaques (par factorisation et de Wiener) dans ce qui suit nous allons présenter cette simulation.

CHAPITRE 3

Implémentation et réalisation

3.1 Introduction

Dans ce chapitre, nous allons présenter dans un premier lieu l'environnement de travail ainsi que les outils utilisés. En second lieu, nous allons présenter quelque capture d'écran de l'interface graphique de notre application.

3.2 Environnement de développement:

Durant la réalisation de notre application, nous avant utilisé une machine ayant les caractéristiques suivantes :

- Un micro lenevo core i5-2520M
- Une mémoire vive d'une capacité de **4Go.**
- Système d'exploitation **64 bits**.
- Une carte graphique de **512 Mo.**

3.3 Ressources logicielles

Php

3.4 Présentation des interfaces de notre application :

Nous allons présenter les différentes interfaces de l'application

a) interface de cryptage :

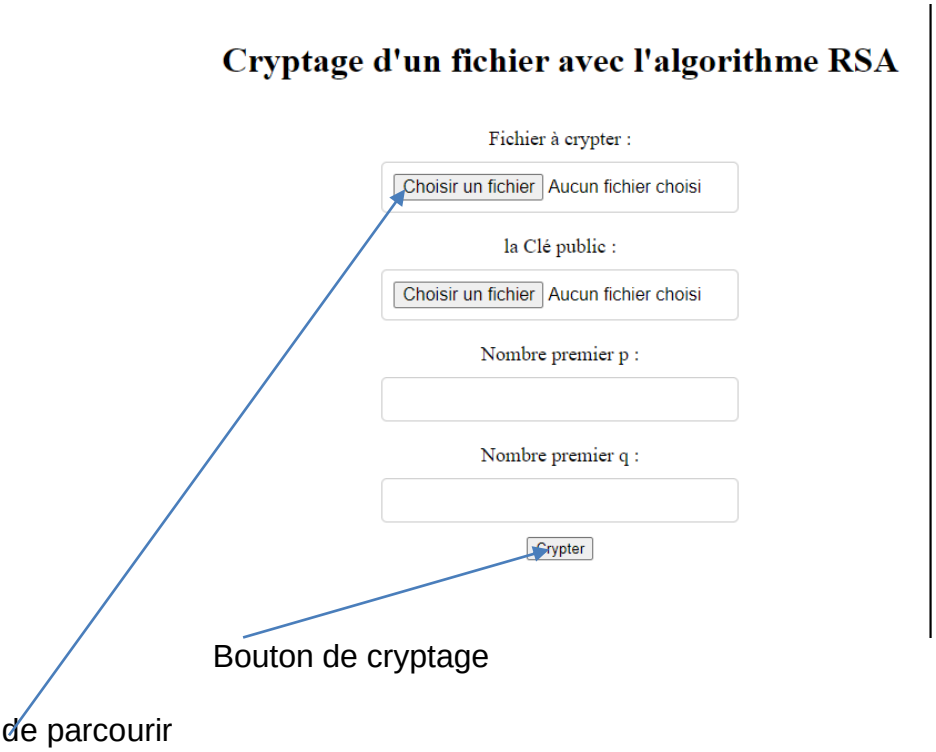


Figure 10. Interface de cryptage

Bouton de « Cryptage » : permet à l'utilisateur de lancer l'opération de cryptage.

b) Fenêtre de sélection de fichier a crypté :

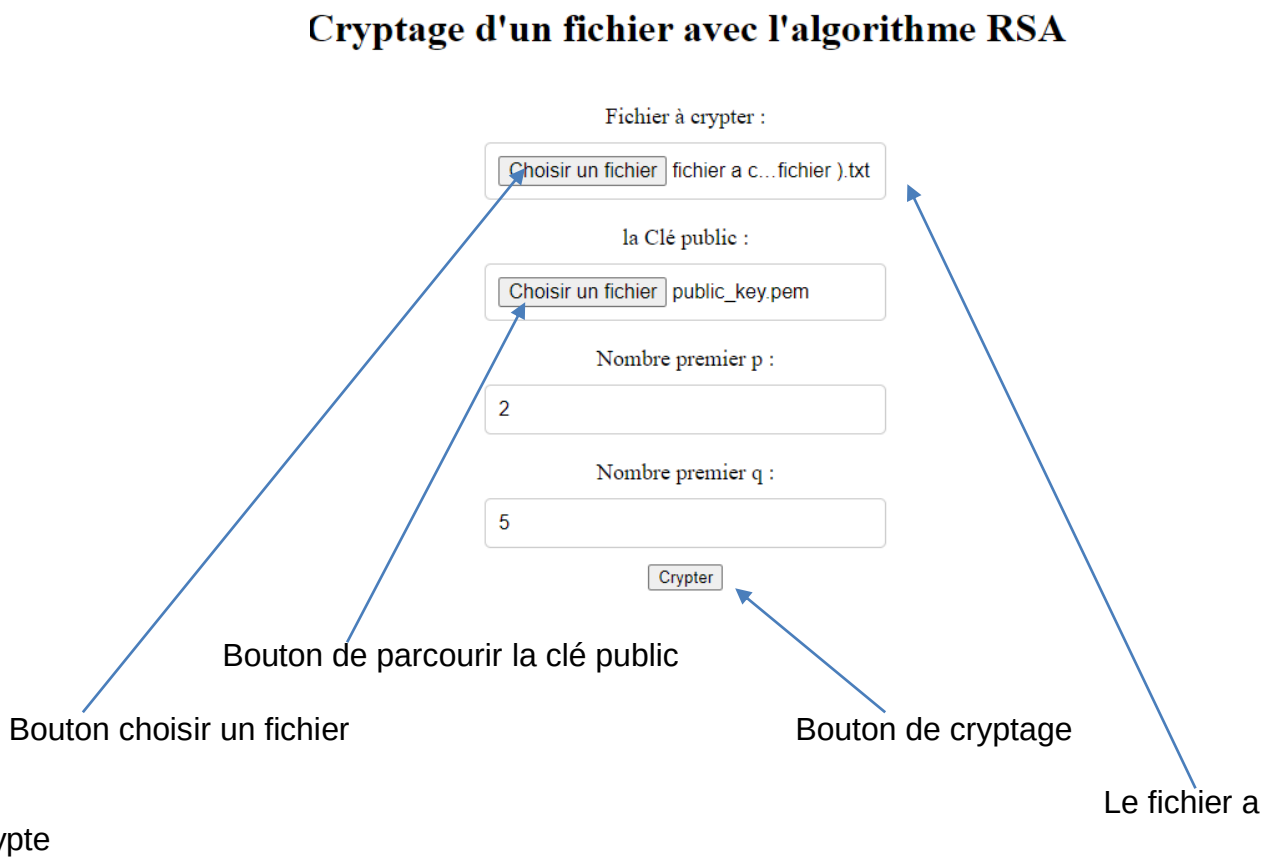


Figure 11: interface de cryptage de fichier.

L'utilisateur sélectionne le fichier à crypter en cliquant sur le bouton choisir un fichier qui lui permet de choisir un fichier et la clé public dans un répertoire. La figure représente l'interface procréée par ce bouton.

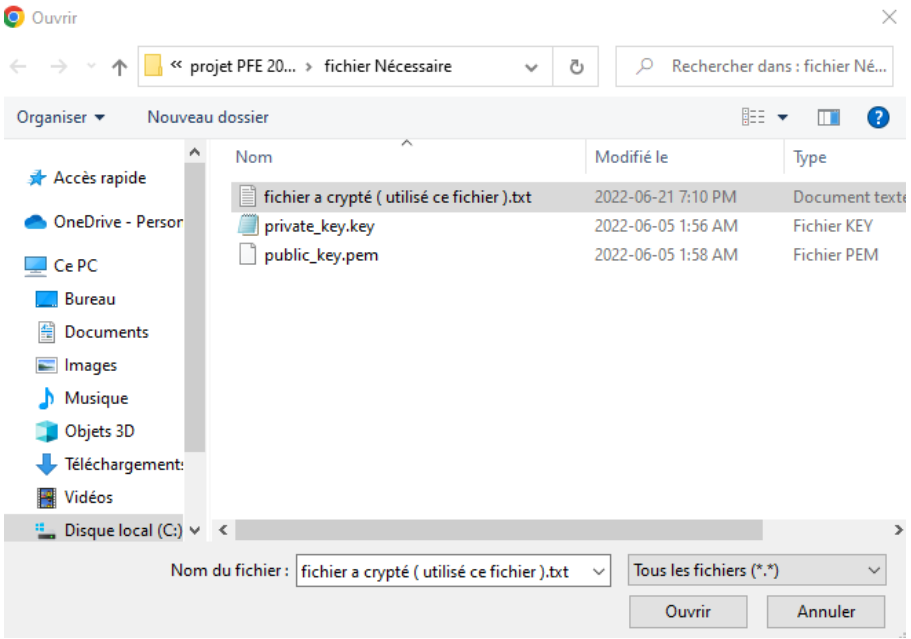


Figure 12: interface pour choisir le fichier à crypter

Après avoir sélectionné le fichier qu'on veut crypter et la clé publique, il faut cliquer sur le bouton « crypter »qui fait appel à l'algorithme de cryptage.

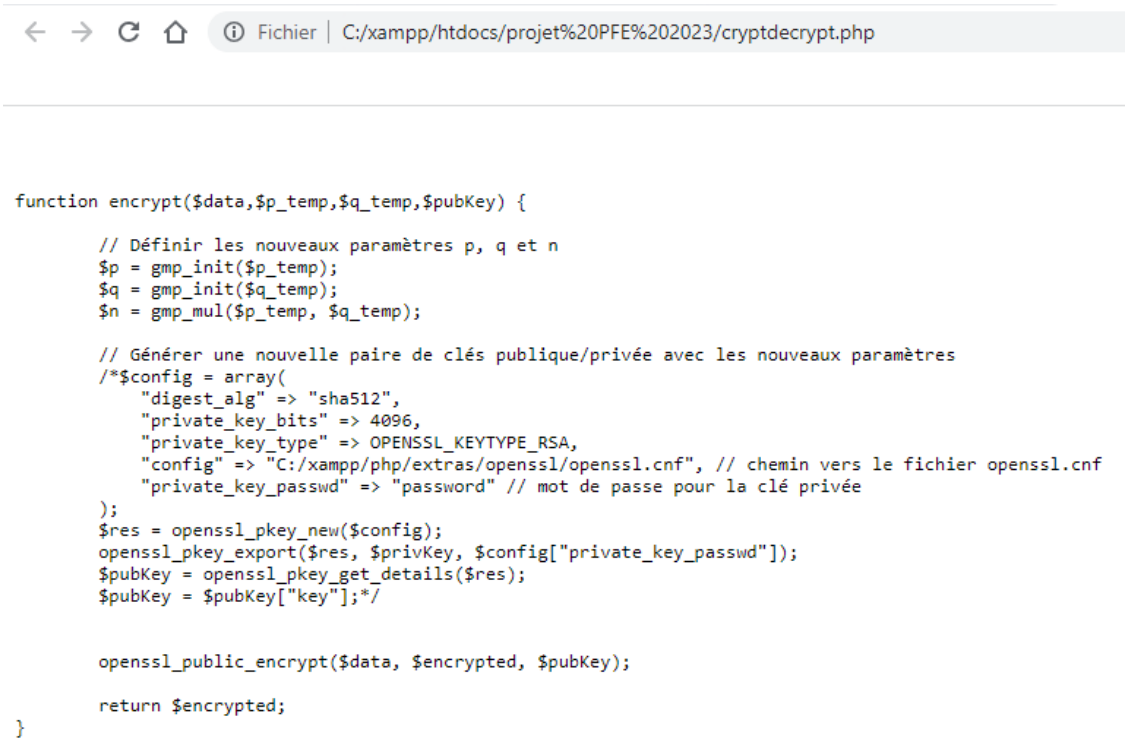


Figure 13. Code de cryptage.

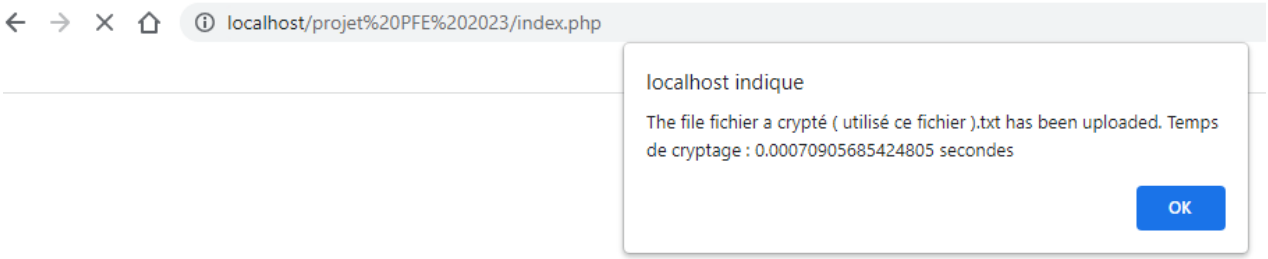


Figure 14. Résultat de cryptage.

c) Interface de décryptage :

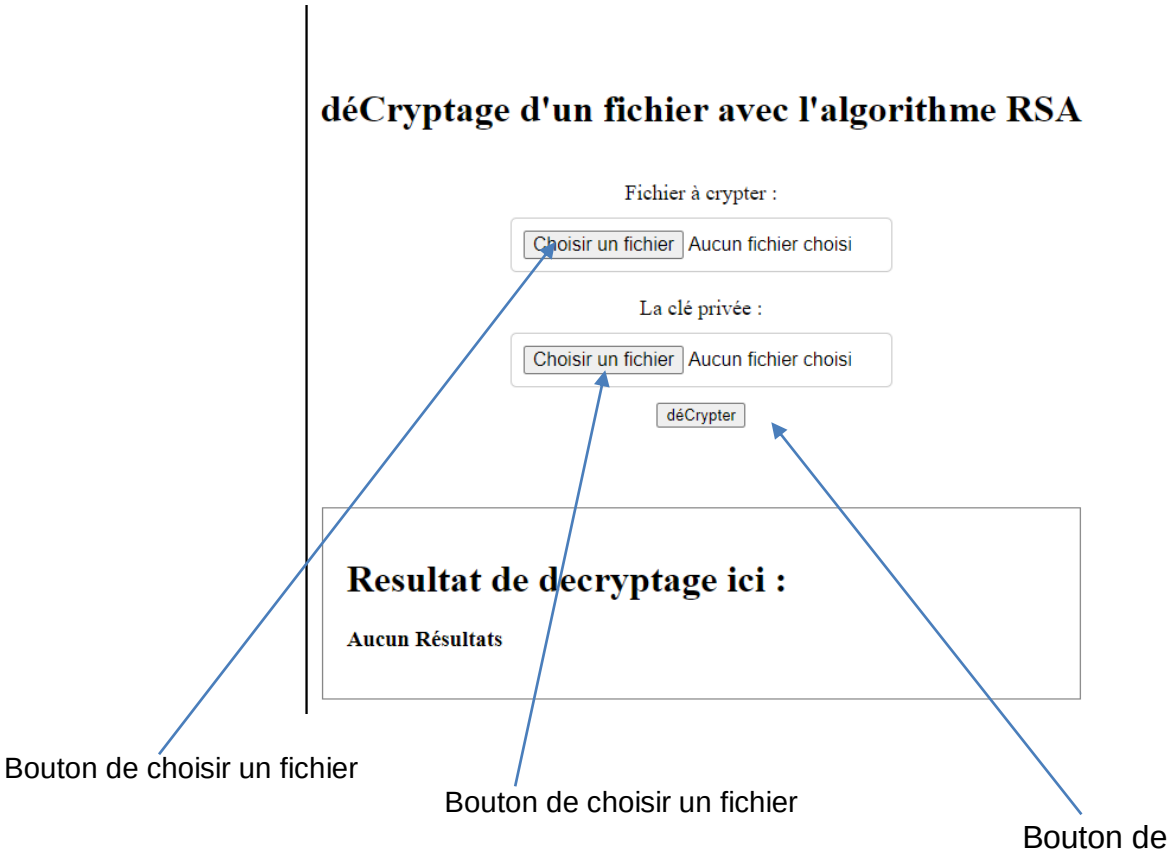


Figure 16. Interface de décryptage

L'utilisateur a choisi le décryptage par sélection de fichier il aura la figure pour choisir le fichier souhaité.

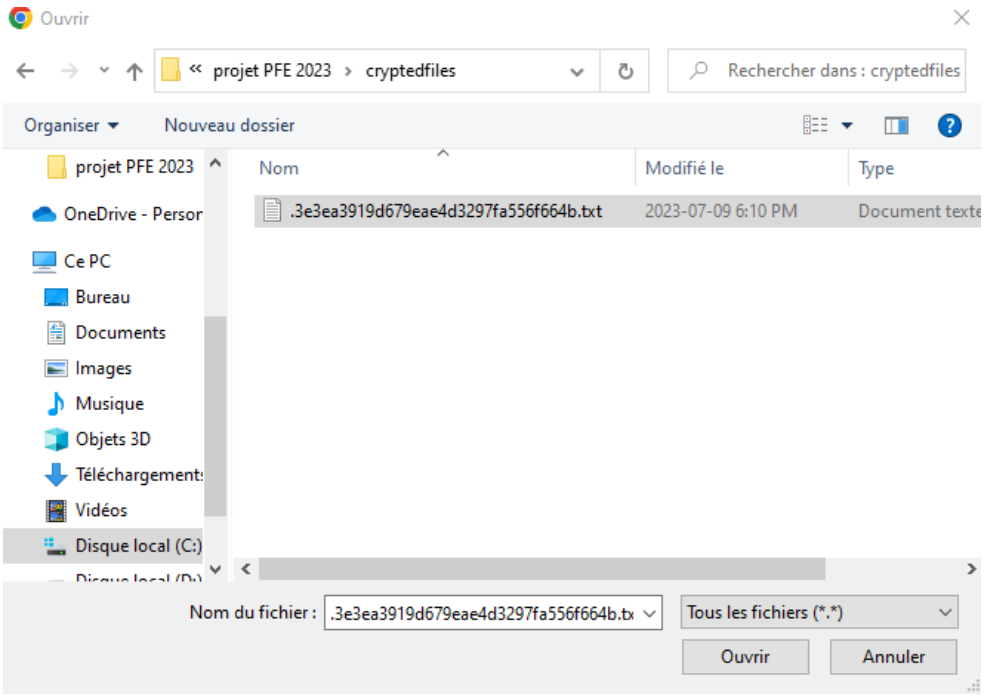


Figure 17: interface de sélection de fichier a décrypte.

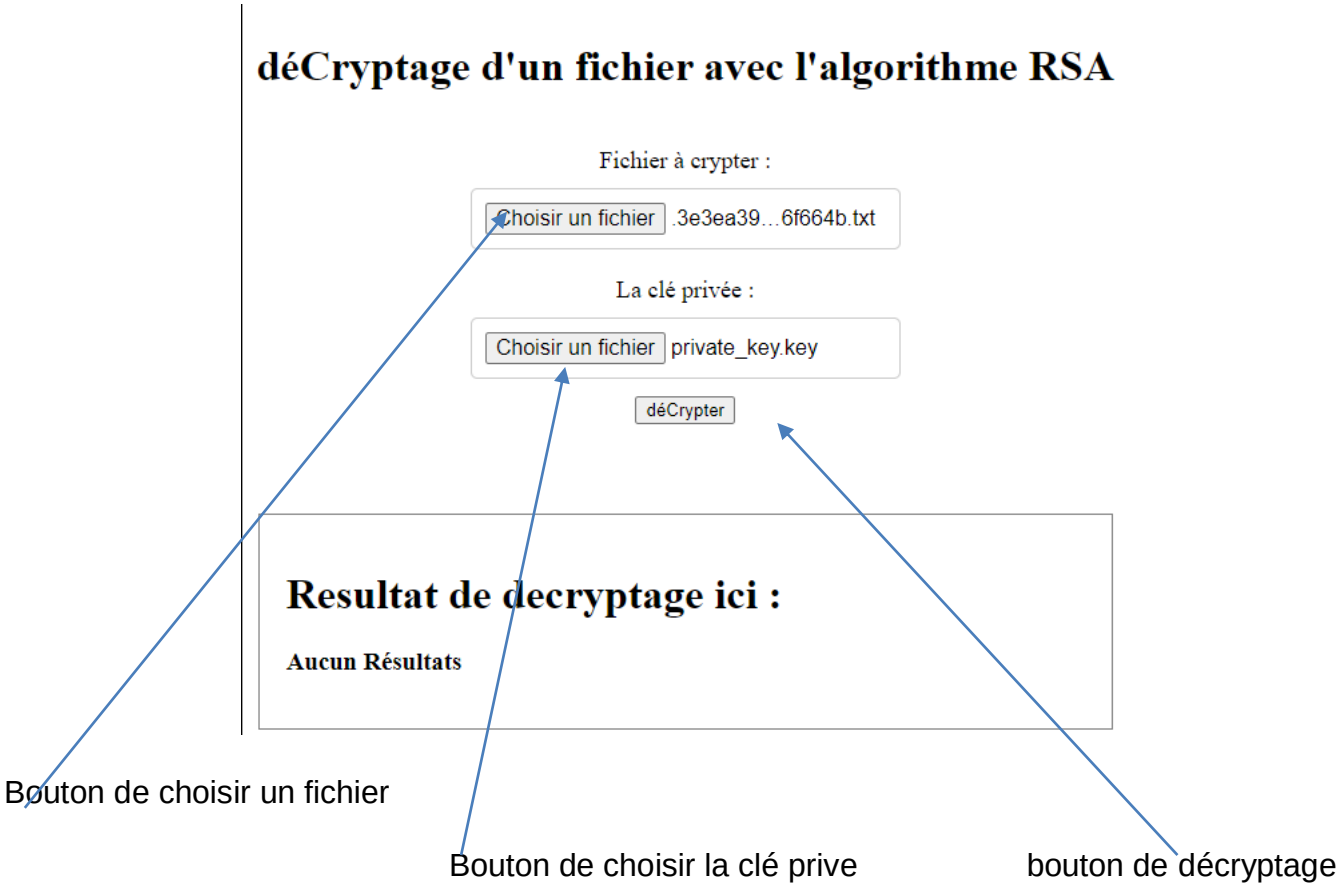


Figure 18: interface de sélection de fichiers à décrypter

- Bouton « choisir un fichier » : permet de sélection le fichier à crypter.
- Bouton « choisir un fichier » : permet de sélection la clé prive.
- Bouton « Décrypter » : fait appel à l'algorithme de décryptage.

```
function decrypt($encrypted,$privKey){

        // Déchiffrer la chaîne de caractères avec la clé privée
        openssl_private_decrypt($encrypted, $decrypted, $privKey);
        return $decrypted;
}
```

Figure 19. Code de décryptage.

Figure 20. Résultat de décryptage

Le tableau suivant montre le temps nécessaire pour crypte un fichier texte de différentes tailles à savoir 8 bits, 16 bits respectivement.

P	Q	Temps (s)
2	5	0,0003440711975097
211	41	0,000704978103377
59659	24889	0,0009939565277095

Tableau 5. Temps nécessaire pour crypte un fichier texte.

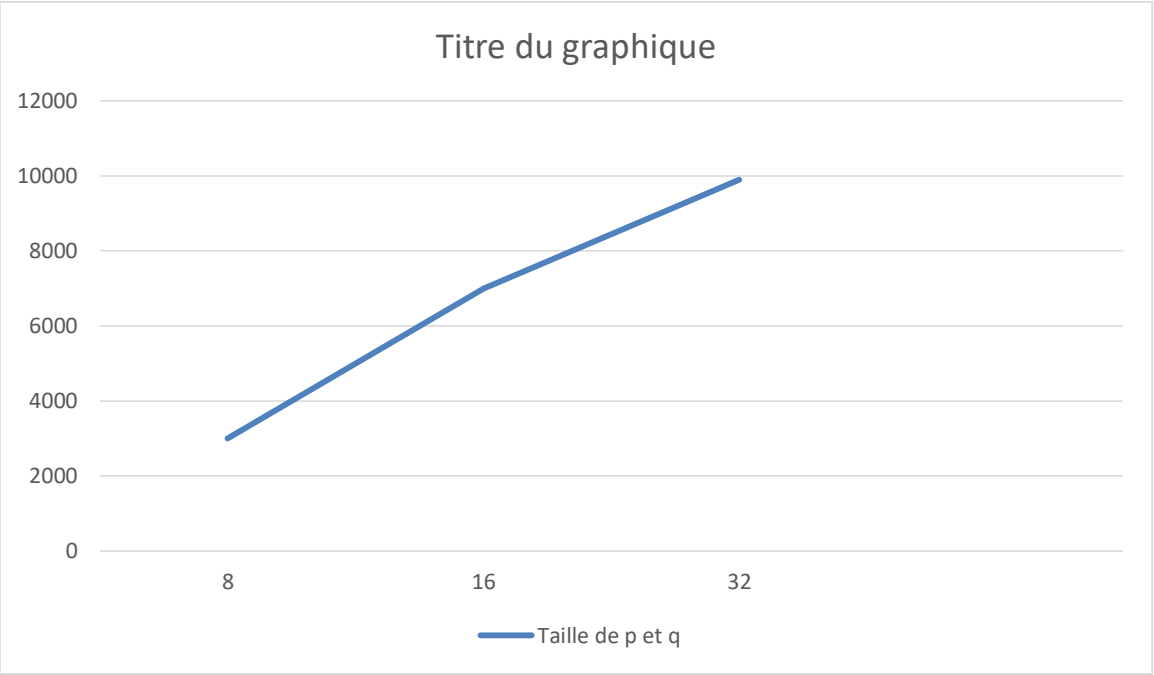


Figure 21. Graphe du temps calculé pour le cryptage de fichier texte.

Le tableau suivant montre le temps nécessaire pour décrypte un fichier texte de différentes tailles à savoir 8 bits, 16 bits respectivement.

P	Q	**Temps (s)**
211	41	0,001749992370055
59659	24889	0,003854181289729
2731321963	428540977	0,007709938049314

Tableau 5. Temps nécessaire pour décrypté un fichier texte.

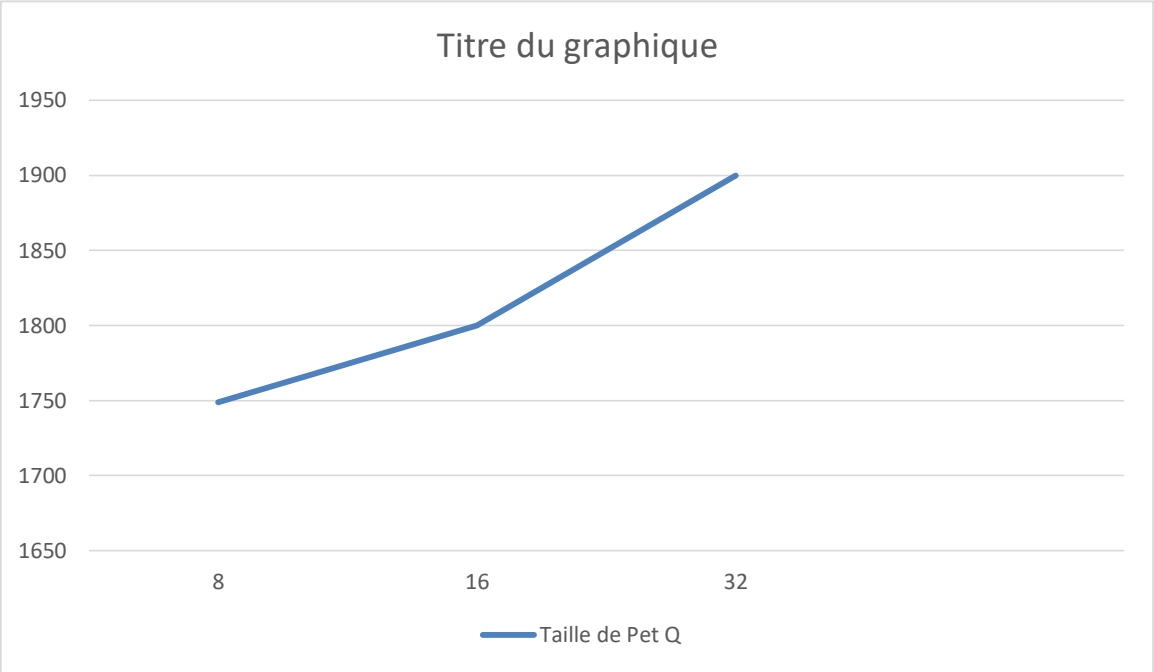


Figure 22. Graphe du temps calculé pour le décryptage de fichier texte.

3.5 Discussion

D'après le graphe du temps calculé pour le cryptage et le décryptage du fichier texte, on remarque que dans les deux cas, plus la taille de p et de q qui sont les deux facteurs de N qui constitue le module de RSA est grande, plus le temps de calcul pour réaliser.

3. 6 Conclusion :

Dans ce chapitre, nous avons présenté notre application qui nous permet de dérouler l'algorithme RSA. Les résultats obtenus par les différents scénarios testés montrent que la sécurité du cryptosystème RSA est liée à la taille de la clé.

Conclusion générale

Les cryptographes n'ont cessé de redoubler d'ingéniosité, faisant se succéder des dizaines de systèmes de chiffrement plus recherchés les uns que les autres, mais chaque système a ses limites et ne doit pas être un substitut aux autres mesures de sécurité.

La cryptographie profite de l'évolution de la technologie, mais elle est en même temps victime de cette évolution, car les intrus peuvent utiliser ces nouvelles technologies dans la cryptanalyse, en mettant en œuvres des attaques contre les différents algorithmes de chiffrement, citons parmi ces algorithmes, le RSA qui est l'algorithme le plus populaire de nos jours et sur lequel nous avons mené notre étude, plus exactement sur sa cryptanalyse.

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser une application qui nous permet de crypte et décrypté un fichier texte.

De nos jours, avec la perpétuelle augmentation des domaines d'utilisation des systèmes embarqués et avec la multiplication des objets connectés, un chiffrement des échanges est nécessaire ainsi qu'une accélération des performances et bien sur une protection contre les attaques et pour satisfaire ces objectifs les systèmes embarqués s'intéressent aux courbes elliptiques qui sont principalement utilisées dans les environnements à faibles ressources car elles ne nécessitent pas de longues clés pour assurer un haut niveau de sécurité contrairement aux autres algorithmes de chiffrement à clé publiques tel que le RSA qui utilise des clés trop longues.

Bibliographie

[1] S.H.Nawal, Conception et réalisation d'un système collaboratif pour les experts métier à base d'agent et des algorithmes de cryptage. Université Ahmed ben Bella d'Oran, thèse de doctorat. 2017.
[2] R.Dumont, Cryptographie et Sécurité informatique, livre informatique, 2009 - 2010.
[3] B.Rabab, Sécurité des images Numériques compressées JPEG. Université Djillali Liabès de Sidi Bel Abbes, thèse de doctorat.03 juin 2019.
[4] A.Nassima et CH.Hamida, Etude sur l'Applicabilité de la Cryptographie Asymétrique aux Réseaux de Capteurs sans Fil. Université Abderrahmane Mira de Béjaïa, thèse de master.2012.
[5] W.Stallings, Cryptography and Network Security : Principles and Practice, Sixth Edition, livre informatique, 2014.
[6] J.-P.Aumasson, Serious Cryptography A Practical Introduction to Modern Encryption, No Starch Press, Article, 2018.
[7] D.Stinson, Cryptography: Theory and Practice. CRC Press, livre informatique, 2005.
[8] B.Mohamed Kamal, Approche Cryptographique basé sur les algorithmes génétique pour la sécurité des réseaux Adhoc. PhD thesis, Université d'Oran, thèse de doctorat.

[9] M.Dubois, Conception, développement et analyse de systèmes de fonction booléennes décrivant les algorithmes de chiffrement et de déchiffrement de l'Advanced Encryption Standard. ParisTech, l'école Nationale Supérieure de Paris, thèse de doctorat. 2018.

[10] Culture Informatique, Comment ça marche le cryptage ?, 18 mars 2016

[11] Preuve de sécurité https://fr.wikipedia.org/wiki/Preuve_de_sérurité

[12] Explication, Décryptage https://chiffrer.info

[13] Travail de Bachelor, Haute École de Gestion de Genève (HEG-GE) Filière Informatique de Gestion, 5 juin 2015, Daniel LAMAS, « La cryptographie ». https://core.ac.uk

[14] Mémoire de fin d'études en recherche opérationnelle, Université HOUARI BOUMEDIENNE, 2007, Louiza REZALLAH, « De la cryptographie classique a la cryptographie moderne théorie et application ». http://repository.usthb.dz

[15] D'après un cours de Daniel Barsky & Ghislain Dartois, Cryptographie, Paris 13 le 1 octobre 2010 https://www.math.univ-paris13.fr

[17] Généralité sur la cryptographie http://dspace.univ-tlemcen.dz/bitstream/112/6836/1/Etude-comparative-entre-la-cryptographie.pdf

[18] Houda FERRADI, Chiffrement par bloc (AES), Université Paris 13 Villetaneuse 01/02/2016

[19] Chiffrement RSA https://www.lemagit.fr

[20] Thèse de doctorat de l'université Pierre et Marie Curie Spécialité Informatique, 2012, Stéphane Jacob « Protection cryptographique des bases de données : conception et cryptanalyse »

[21] Renaud Dumont, Cryptographie et Sécurité informatique, Université de Liège Faculté des Sciences Appliquées, 2009 – 2010.

[22] M.Dubois, Conception, développement et analyse de systèmes de fonction booléennes décrivant les algorithmes de chiffrement et de déchiffrement de l'Advanced Encryption Standard. ParisTech, l'école Nationale Supérieure de Paris, thèse de doctorat. 2018.

[23] Y.SHOU, Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les

réseaux de capteurs. PhD thesis, Université de Franche-Comté, thèse de doctorat,2014.

[24] G.Seroussi I.Blake, G.Seroussi and N.Smart, Elliptic curves in cryptography. Cambridge

university press, Article, 1999.

[25] V. Gayoso Martinez, C. Sanchez Avila J. Espinosa Garcia et L. Hernandez Encinas,Elliptic curve cryptography. java implementation issues, Article, 2005.

[27] Vincent Verneuil, Courbes elliptiques et attaques par cannaux auxiliaires. Science et

Technologie, Atricle, 2009.

[28] A.Singh et R.Singh, Various attacks over the elliptic curve-based cryptosystems.International Journal of Engineering and Innovative Technology (IJEIT), 2015.

[29] S.Pontié, Sécurisation matérielle pour la cryptographie à base de courbes elliptiques.PhD thesis, Université Grenoble Alpes, thèse de doctorat, 2016.

[30] B.Hichem. Sur la sécurité de l'information par le biais des courbes

elliptiques. PhD thesis, Université Djillali Liabes Faculté Des Sciences exactes, Sidi Bel Abbés, thèse de doctorat, 2018.

[31] C.Gonçalves, Cryptographie Avancée Courbes elliptiques, livre informatique, 2015.

[33] I.Lotfi, Cryptographie à base de courbes elliptiques. PhD thesis, Ecole Nationale

Supérieure d'Informatique, thèse de doctorat, 2017.

[35] https://www.secg.org/SEC2-Ver-1.0.pdf.

[37] https://fr.wikipedia.org/wiki/Pretty_Good_Privacy.

[38] https://en.wikipedia.org/wiki/GNU_Privacy_Guard.

[39] P.Crescenzo. OFL : Un modèle pour paramétrer la sémantique opérationnelle des langages à objets application aux relations inter-classes. PhD thesis, l'Université de Nice-Sophia Antipolis, thèse de doctorat, 2001.