الجمهوريـة الجزائـريـة الديمقراطيـة الشعبيـة

République Algérienne Démocratique et Populaire

وزارة التعليـم العالـي والبحـث العلمـي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Réf……

Centre Universitaire de Mila

Institut des sciences et de la technologie **Département de Mathématiques et Informatique**

Mémoire préparé En vue de l'obtention du diplôme de licence

En :-Filère : Informatique Général

Thème

Sécurité des données par le système cryptographique à clé mixte : PGP

Préparé par: **- Dib Nasreddin**

- Boudebane Hamza

- Belharebi Sami

- Encadré par : Mme Deffas Zineb M.A.A.

Année universitaire : 2013/2014

-

Remerciement

C'est avec l'aide de Dieu qu'a vu les jours ce présent travail.

Ensuite, il n'aurait pas pu être achevé sans le soutien, les conseils,

Les encouragements de certaines personnes auxquelles nous tenons ici

À exprimer nos sincères remerciements.

En première lieu nous exprimons toute notre gratitude pour notre encadreur Mme DEFFAS ZINEB pour ses précieux conseils, ses disponibilités,

La confiance qu'elle nous a toujours témoigné et la sollicitude dont elle nous

A entouré, et ce tout au long de l'élaboration du présent travail.

Nous n'oublions pas non plus nos enseignants qui tout au long du cycle d'étude au centre universitaire de Mila, nous ont transmis leur savoir.

Nous tenons enfin à remercier tous ceux qui ont collaborés de prés

Ou de loin à l'élaboration de ce travail. Qu'ils acceptent nos humbles remerciements.

NASREDDINE & SAMI & Hamza

Dédicaces

Je dédie ce travail :

- ✓ A mon père
- ✓ A ma mère
- ✓ A mes sœurs
- ✓ A mon binôme Sami et hamza.
- ✓ A mes collègues d'étude surtout Sami, Hamza, Mourad, Fayçal, et mon ami Zaki
- ✓ A mon cousin Adèle
- ✓ A toute ma famille

NASREDDINE

Dédicaces

A mes très chers parents qui m'ont beaucoup

Soutenu, encouragé et qui ont fait de moi ce que je suis

Aujourd'hui ;

A mes frères : Hamza, Bessam, Bilal

Ames sœurs et le marie de ma grande sœur : Nasreddine

A mes binômes :Nasreddine Dib, Boudebane Hamza

A tous mes amis en particulier

Khalid, Houssin, Zakaria, Ilyasse, Issam,Abd elhamide

A tous mes amis (es) 3emanneé informatique en

Particular Salem, Nasser, Khaled, Abd elazize, Mohamed

N'oublier pas les meilleurs amis durant 3 dernières années

A toute les familles

Beharbi et Bouali

A tous ceux que j'aime tant et que je n'ai pas cités ;

BELHARBI SAMI

Dédicaces

A mes très chers parents qui m'ont beaucoup

Soutenu, encouragé et qui ont fait de moi ce que je suis

Aujourd'hui ;

A mes frères : Oussama, Kamel, Faysal, souffian

A mes sœurs Samia, Alima, Masika

A mes binômes : Nasreddine Dib, Belharbi Sami,

A tous mes amis en particulier

Ahmed, Zaki, Mounaime, Souffian, Dani, Salah,

A tous mes amis (es) 3emanneé informatique en

Particular Farid, Zakariya, Mohamed

N'oublier pas les meilleurs amis durant 3 dernières années ,

A toute les familles

Boudebane hamza

Résumé

Le problème de sécurité ne se posait pas, lorsque les entreprises et les universités n'avaient qu'un seul centre d'ordinateurs. Il suffisait de placer un garde à l'entrée de la salle. Maintenant avec la venue des réseaux, l'emploi des liaisons satellites et l'utilisation de l'Internet, la situation a radicalement changé, dans la mesure où un même message transite par plusieurs machines avant d'atteindre son destinataire. A chaque étape, il peut être copié, perdu ou altéré. Le cryptage (sécurité des données) est donc nécessaire pour que les données soit non-intelligibles sauf à l'auditoire voulu. Dans cette communication , on va étudier le système PGP (Pretty Good Privacy) utilisé par la communauté d'Internet, PGP est un l'algorithme à clé mixte ou hybride qui est constitué de deux algorithmes de types différents, l'un à clé publique RSA et l'autre à clé secrète IDEA, combinés de façon à exploiter les avantages du premier algorithme pour minimiser les inconvénients du deuxième, afin de mieux Sécuriser l'information émise.

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION GÉNÉRAL

Introduction générale

La cryptologie, « science du secret » est l'une des sciences les plus antiques (plus de 3000 ans). Elle a été réservée pendant longtemps, aux milieux diplomatiques et militaires. Grâce au Développement de la société de l'information et l'évolution des réseaux de communications sa Démocratisation s'est installée et elle s'est imposée dans tous les domaines. De nouvelles Exigences se sont alors apparues : assurer la confidentialité des messages ne suffit plus, il faut Également assurer leur intégrité et leur authenticité.

Dans la cryptologie, on distingue la cryptographie et la cryptanalyse. La première définit Et étudie les systèmes utilisés, alors que la seconde cherche à valider ou à casser ces systèmes.

La cryptographie (sécurité des données) est l'ensemble des processus de verrouillage visant à Protéger l'accès à certaines données afin de les rendre incompréhensible aux Personnes non autorisées, Autrement dit garantir la confidentialité, l'intégrité de ces informations , ainsi que leur imputabilité. L'émetteur d'une Information doit être certain de l'identité du Destinataire et inversement.

Puisque cette science est très large, nous essayons d'implémenté un système de Cryptographies PGP (Pretty Good Privacy)

- **Le PGP** de Phil Zimmermann utilise deux algorithmes distincts pour crypter et décrypter Les données dans un réseau de téléinformatique, l'un est à clé publique et l'autre à clé secrète.

Pour aboutir à ce but nous divisons notre travail en 3 chapitres:

Le chapitre 1 présente une généralité sur la cryptographie, et un panorama sur les Algorithmes Cryptographique existent avec des exemples, et termine par quelques concepts sur la Cryptanalyse et Les Différentes techniques utilisent dans cette opération.

Le chapitre 2 explique en détaille le système de cryptographie que nous avons choisi (système PGP).

Le chapitre 3 commence par la description du langage et l'environnement de développement Utiliser pour l'implémentation. Puis présente l'interface graphique de notre application suivie Par des exemples

CHAPITRE I GÉNÉRALITÉS SUR LA CRYPTOGRAPHIE

1. Introduction

Dans ce chapitre , nous présentons un aperçu général sur la notion de la cryptographie nous commençons par l'historique, puis , nous donnons quelques définitions de la cryptographie et leurs Fonctions. Par la suite, nous présentons une classification des algorithmes cryptographique, et enfin Nous terminons ce chapitre par la cryptanalyse et les différentes attaques utiliser.

2. Les fondements de la cryptographie

2.1 Historique

Avant de décrire la cryptographie en soi , passons en revue son histoire , son apparition et son évolution au fil du temps ; les aspects techniques brièvement présentés dans cet Élément. La science de chiffrement n'est pas une science moderne mais L'histoire de Cryptage est plus long est classé sur trois grandes étapes :

2.1.1 L'âge artisanal (part des origines)

La cryptographie serait apparue pour la première fois il y a 4000 ans, au bord du Nil : un scribe aurait tracé d'une façon particulière des hiéroglyphes sur la tombe de son maître. Le but n'était néanmoins pas de rendre le texte illisible, mais de le rendre plus solenne

Mais le cryptage développé avec Jules César utilisait, semble-t-il un mécanisme de confidentialité rudimentaire, où chaque Lettre d'un message était remplacée par celle située trois positions plus loin dans l'alphabet. La méthode se généralise en opérant une permutation quelconque de l'alphabet, et prend le nom De substitution. Une autre méthode, dite de transposition, change l'ordre des lettres; elle a été mise en œuvre au Moyen Age notamment par un dispositif appelé " grille de Cardan ". De façon générale, Jusqu'au début du vingtième siècle, la cryptographie était affaire de substitutions et de transpositions. On opérait d'ailleurs fréquemment des substitutions non seulement sur des lettres mais sur des mots, en s'aidant d'une sorte de dictionnaire à double entrée, nommé code ou répertoire [1].

2.1.2 L'âge technique

De nombreux dispositifs mécaniques furent mis en place afin de faciliter le Chiffrement. La Plupart De ces dispositifs se basaient sur des rotors (des disques où Sont imprimées Les lettres de l'alphabet). C'est le cas de la machine Enigma,

Destinée initialement aux civils (inventée en 1919 par Arthur Scherbius et Commercialisée en 1923), mais très vite Utilisée par l'armée allemande à partir des années 30. La plupart des communications Allemandes seront chiffrées via la machine Enigma, d'où L'intérêt pour les alliés de pouvoir Déchiffrer leurs messages.

La machine Enigma effectue une substitution qui change à chaque lettre, son fonctionnement Est basé sur un assemblage d'un tableau de connexion (qui ne fait que permuter Quelques lettres, afin de rendre la cryptanalyse plus difficile), de plusieurs rotors qui Effectuent les permutations des lettres, et d'un réflecteur qui renvoie le courant sur le Panneau lumineux Où la lettre chiffrée s'allume. À chaque Lettre tapée, le premier Rotor avance d'une Position. Lorsque le premier rotor a fait un tour complet, c'est le Second qui tourne, et ainsi De suite. Grâce à la combinaison de ces dispositifs, on peut obtenir plus de 1017 Clés Possibles pour une machine À 5 rotors.

2.1.3 L'âge paradoxal (utilise les algorithmes de cryptage)

Après la guerre 40-45, il faudra attendre une trentaine d'années avant de nouvelles Avancées Dans le domaine de la cryptologie. En 1971, un cryptographe d'IBM, Horst Feistel met au point un Algorithme de chiffrement par bloc, nommé Lucifer Qui possède de nombreuses variantes. Deux ans plus tard, la ***NSA*** modifie Lucifer pour Sortir le Data Encryptions Standard, ***DES***, en 1977 ; il fut encore amélioré par la Suite et longuement utilisé et reste utilisé encore aujourd'hui (bien qu'il Soit moins Répandu), notamment Avec ***le Triple DES***, qui consiste simplement à opérer trois ***DES*** Consécutifs avec deux ou trois clés. La même année, le chiffrement à clé Publique (ou asymétrique) est présenté pour la première fois dans une publication de ***W. Diffie*** et ***M. Hellman***. Ce concept est mis en pratique avec le chiffrement ***RSA***, Inventé par ***Ron Rivest***, ***Adi Shamir*** et ***Leonard Adleman*** et présenté en 1978 Dans une Publication. Grâce aux algorithmes à clé Publique, le problème de Distribution des clés est résolu via l'utilisation de deux clés : une Pour chiffrer, rendue publique, et une pour déchiffrer, gardée secrète Par la personne Censée déchiffrer le Message [2].

2.2 Définition de la cryptographie

En informatique, la littérature fournit un tas de définitions du mot cryptographie.

Ces Définitions, dans Leur diversité, offrent des points de vue à la fois différents et Complémentaires. Dans ce qui suite, nous Décrivons quelque saunes.

- La cryptographie est un moyen de sauvegarder le caractère confidentiel des Informations. Elle ne protège pas les communications en tant que telles mais plutôt leur contenu pour protéger l'accès à certaines données afin de les rendre incompréhensible aux Personnes non autorisées. [3]
- La cryptographie est l'ensemble des techniques qui permet de rendre un message Inintelligible. L'action de coder le message initial (plain texte) en un message inintelligible, Appelé «cryptogramme» (Cipher texte), se nomme « chiffrement » (ou cryptage). L'opération inverse est le « Déchiffrement » (Ou décryptage) [4].
- La cryptographie est la science qui utilise les mathématiques pour le cryptage et le Décryptage De données. Elle vous permet ainsi de stocker des informations confidentielles ou de les transmettre sur des réseaux non sécurisés (tels que l'Internet), afin qu'aucune personne autre que le destinataire ne puisse les lire [5].

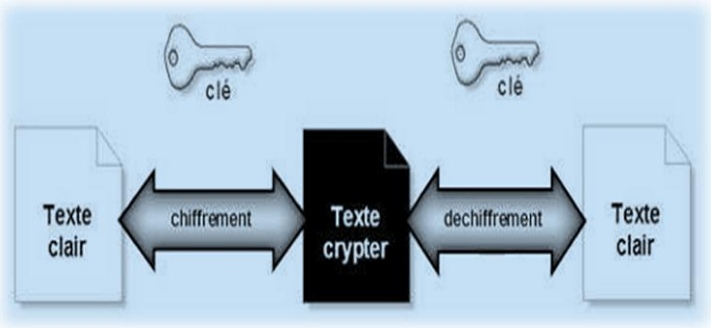


Figure I.1: schéma générale de cryptographie

2.3 Les fonctions de la cryptographie

La cryptographie est traditionnellement utilisée pour dissimuler des messages Aux yeux de certains Utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus Grand que les communications via internet circulent dans des infrastructures dont on ne Peut garantir La Confidentialité. Désormais, la cryptographie sert non seulement à

Préserver la confidentialité des données mais aussi à garantir Leur intégrité et leur Authenticité.

2.3.1 L'authenticité:

Garantit l'identité d'une entité donnée ou l'origine d'une communication ou D'un Fichier. Lorsqu'il s'agit d'un fichier et que l'entité qui l'a créé est la Seule à Avoir pu apporter la garantie D'authenticité, on parle de non -répudiation.

2.3.2 Non-répudiation:

Mécanisme pour enregistrer un acte ou un engagement d'une personne ou d'une entité De telle sorte que celle-ci ne puisse pas nier avoir accompli cet acte ou pris Cet Engagement.

2.3.3 L'intégrité:

Garantit que le contenu d'une communication ou d'un fichier n'a pas été modifié. Par exemple, on peut souhaiter vérifier qu'aucun Changement du contenu d'un disque Dur n'a Eu lieu: des produits commerciaux, mettant en jeu des méthodes Cryptologiques, sont Disponibles à cet effet.

2.3.4 La confidentialité:

Garantit que le contenu d'une communication ou d'un fichier n'est pas accessible aux Tiers. Des Services de confidentialité sont offerts dans de nombreux contextes

- en téléphonie mobile , pour protéger les communications dans la partie "aérienne";
- en télévision à péage pour réserver la réception des données aux abonnés;
- dans les navigateurs, par l'intermédiaire du protocole SSL (Secure Socket Layer), dont est souvent indiquée par un cadenas fermé représenté En bas de la fenêtre .

2.4 .A quoi sert la cryptographie ?

L'application la plus évidente de la cryptographie est la protection de la Confidentialité d'une Information, qu'elle soit stockée localement sur une machine, Ou transmise sur un Réseau. Le besoin De confidentialité n'est pas l'apanage des Militaires ou de certain gros Industriels. Tous les individus, toutes les organisations Ont, à des degrés divers, un tel Besoin :

- Confidentialité des transactions bancaires.
- Protection de secrets industriels ou commerciaux.

- Protection des sessions de télétravail.
- Protection du secret médical.
- Protection des systèmes informatique contre les intrusions.
- Protection de la confidentialité de communications dans le cadre d'une association, d'un parti Politique, d'un syndicat…
- Protection de la vie privée.
- jeux

2.5 Termes de la cryptographie

. Le chiffrement : est la démarche effectuée afin de rendre le message clair illisible, Chiffré. On utilise parfois le terme « cryptage », qui est un anglicisme ; nous éviterons donc de l'utiliser.

- **Le déchiffrement :** est la démarche inverse du chiffrement, qui retrouve le Message Clair à partir du message chiffré en ayant connaissance de la clé, du Secret ou de L'algorithme utilisé (Contrairement à la cryptanalyse). Le mot décryptage Peut aussi être Utilisé.
- **Cryptologie :** Regroupe cryptographie et cryptanalyse.
- **La cryptanalyse :** vise à retrouver le message clair à partir du message chiffré Sans Avoir Connaissance de la clé ou du secret.
- **La sténographie:** est une discipline semblable à la cryptographie mais qui consiste A Cacher un message (dans une image. . .) et non pas à la rendre inintelligible.
- **La clé de chiffrement:** est une donnée (mot, suite d'opération, nombre. . .) utilisée Pendant le Chiffrement afin de rendre le déchiffrement plus difficile sans la Connaissance de celle-ci. Il existe plusieurs types de clés :

Certaines qui doivent être gardées secrètes (clés privées) et d'autres qui peuvent Être Diffusées (Clés publiques).

- **Les données claires:** sont les données dans leur forme initiale, non chiffrées. Ces données peuvent être du texte (un message) ou d'autres données informatiques (Un fichier,. . .).
- **Les données chiffrées :** sont les données qui sont chiffrées via un certain

Algorithme Déchiffrement.

- **Un cryptogramme :** est un message chiffré dans le but d'être déchiffré par des Amateurs De Cryptographie (on en retrouve par exemple dans les journaux). Ils sont La plupart du temps Assez Simples.
- **La clé :** la clé c'est une valeur utilisée dans un algorithme de cryptographie, afin De générer un texte chiffré.
- **L'algorithme de cryptage :** un algorithme de cryptographie ou un chiffrement est une fonction Mathématique utilisée lors du processus de cryptage et de décryptage.

3. Classification des algorithmes cryptographique

On peut effectuer une classification des algorithmes cryptographiques selon déférents critères, par exemple en peut les classifier selon l'apparition (algorithmes cryptographiques classiques et modernes), ou selon la nature de la clé (publique ou secrète); dans ce chapitre nous essayons de les classifier en prenant en compte tous ces critères.

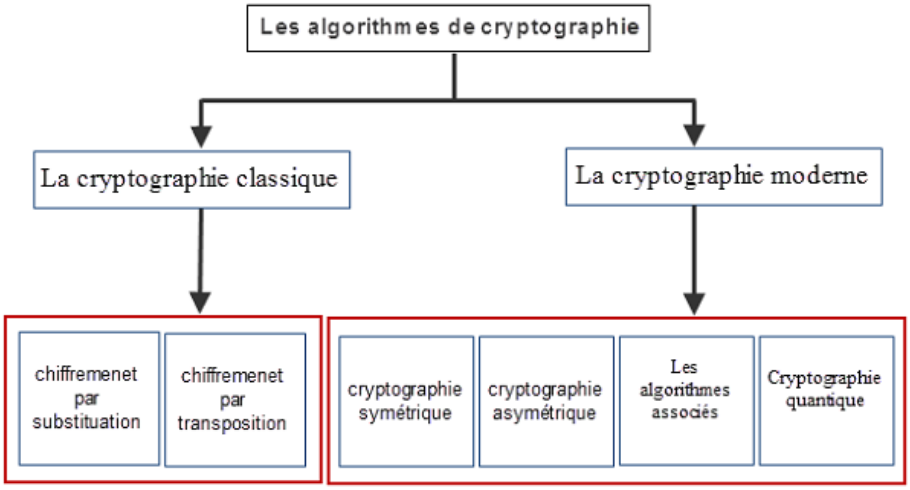


Figure I.2: Classification les algorithmes de la cryptographie

3.1 La cryptographie classique

La cryptographie classique décrit la période avant les ordinateurs. Elle traite Des Systèmes Reposant sur les lettres et les caractères d'une langue naturelle. Les principaux outils utilisés remplacent des caractères par des autres et les Transposent dans des ordres différents .Les meilleurs systèmes (de cette classe D'algorithmes) répètent ces deux opérations de base Plusieurs Fois. Cela suppose que

Les procédures (de chiffrement ou déchiffrement) soient Gardées secrètes ; et sans cela Le système est complètement inefficace (n'importe qui peut Déchiffrer le message Codé).

3.1.1 Chiffrement par substitution

3.1.1.1 Substitution mono alphabétique : « cryptage de césar »

Chaque caractère du texte e en clair est remplacé par un caractère correspondant Dans Le texte Chiffré. Les exemples les plus célèbres sont les algorithmes de César et Rot13. Ils Sont encore utilisés aujourd'hui pour cacher le sens de certains messages (Par exemple la solution de certains jeux Dans des journaux), mais bien sûr elles Sont très peu sûrs. En effet avec ce principe, les lettres les plus fréquentes dans le texte en clair Restent les plus fréquentes dans le texte chiffré, il ne cache Donc pas les fréquences D'apparition des caractères. C'est une faiblesse importante Puisque des techniques statistiques peuvent être utilisées pour associer aux lettres les Plus fréquentes, Les algorithmes à base de substitutions mono-alphabétique sont Facilement Cassés par les spécialistes. Pour l'instant, nous avons vu les chiffrements à un alphabet, ou mono alphabétiques, et constaté comme l'utilisation d'un ordre alphabétique et non alphabétique a un effet sur la complexité.

Exemple :

Chaque texte clair A est remplacé par le texte chiffré B, chaque texte claire B est remplacé Par le texte chiffré C, etc... Utiliser un seul alphabet pour substituer des lettres dans un Message Ne suffit pas à stopper un cryptanalyse.

A B C D E F G H I J K L M N OP Q R S T U V W X Y Z

B C D E F G H I J K L M N O PQ R S T U V W X Y Z A

On a le mot "CRYPTAGE" qui sera "DSZQUBHF" après la substitution.

Text en clair	C	R	Y	P	T	A	G	E
Text crypté	D	S	Z	Q	U	B	H	F

Tableau I.1: table de substitution

3.1.1.2 Substitution poly alphabétique : « cryptage de vigenère »

Le chiffrement de vigénère ressemble beaucoup au chiffrement de césar, à la Différence près Qu'il utilise une clef plus longue afin de pallier le principal problème du Chiffrement de césar. Le fait qu'une lettre puisse être codé d'une seule façon. Pour cela On utilise un mot clef au lieu d'un simple Caractère. Pour crypter on choisit une clef (Mot ou phrase). Chaque lettre du texte clair on fait Correspondre une lettre de clef (la Clef étant répétée autant de fois que nécessaire).

La lettre de chiffré cela prise dans la colonne correspondante à la lettre du texte clair , et dans la Ligne correspondante à la lettre de la clef. En posant **C** le texte codé, **T** le Texte et **K** la clé, on peut Traduire ceci par la formule : **C=T+K [mod26]** Il consiste à coder un texte avec un mot en ajoutant à chacune de ses lettres la lettre d'un autre Mot appelé clé. La clé est ajoutée indéfiniment en vis à vis avec le texte à chiffrer. Pour déchiffrer le message, il suffit de faire l'opération inverse, on prend la ligne Correspondante à La lettre de clé, et on la suit jusqu'à rencontrer le caractère codée, La lettre décodée est alors la première Cette colonne, ce qui se traduit par la formule :

T=C-K [mod26]

Exemple: le texte " texte chiffré " avec la clé « BIEN » sera codé de la manière suivante :

Texte Clair:

T	**E**	**x**	**E**	**e**	**c**	**H**	**I**	**f**	**F**	**R**	**E**
24	**20**	**05**	**03**	**08**	**09**	**06**	**06**	**18**	**05**	**05**	**20**

Clé :

B	I	E	N
02	09	05	14

Texte crypté :

t + B	e +I	x +I	t + N	e +B	c +I	h +E	i +N	f +B	f +I	r +E	e +N
20+02	05+09	24+05	20+14	05+02	03+09	08+05	09+14	06+02	06+09	18+05	05+14

V	N	C	X	G	L	M	W	H	O	W	S
22	14	29=3	24	7	12	13	23	8	15	23	19

''Texte chiffré'' en écrit donc**: VNCXGLMWHOWS**

3.1.2 Chiffrement par transposition

Jusqu'ici, tous les chiffrements que nous avons abordés sont des systèmes Substitution dans lesquels les lettres du texte clair sont remplacées par celles du Texte chiffré. Changer les positions des lettres du texte clair est une autre Technique de chiffrement On 'appelle transposition, beaucoup de journaux présentent Des puzzles par transposition. Un simple chiffrement par transposition de FIVE AM Déplace chaque lettre d'une position vers la gauche.

FIVE AM devient ainsi IVEA MF. Bien que les lettres ayant été déplacées, les lettres du texte chiffré Soient toutes identiques à celles du texte clair, il n'y a ni remplacement, ni Substitution de lettre.

Pour illustrer cette technique de chiffrent, voyons une transposition plus complexe :

LA CRYPTOGRAPHIE CLASSIQUE. (Voir la table I.2 ci-dessous)

Texte clair

LA CRYPTOGRAPHIE CLASSIQUE

Texte chiffré **LT HSAOISCGEIRRC QYALUPPAE**

L	A	S	R	Y	P
T	O	G	R	A	P
H	I	E	C	L	A
S	S	I	Q	I	E

Tableau I.2: Table de transposition

Chaque lettre en texte clair est transférée vers une nouvelle position. La **Table I.2** ressemble à un Traitement de texte à six colonnes, muni de retour à la ligne la méthode Employée pour lire les lettres de la grille est le chiffrement. Ce système ci lit les lettres de Première colonne vers le bas, puis les lettres de la deuxième colonne vers le bas, etc. Les lettres cryptées sont les mêmes que les lettres en texte Clair, mais elles sont Positionnées pour former un nouveau schéma.

Le destinataire prévu doit connaître deux choses : la longueur et la largeur de la

Grille et la manière dont Les lettres sont à partir de la grille.

3.2 La cryptographie moderne

Les méthodes utilisées de nos jours sont plus complexes, cependant la philosophie Reste-la même. La Différence fondamentale est que les méthodes modernes manipulent Directement des bits contrairement Aux anciennes méthodes qui opéraient sur des caractères Alphabétiques. Ce n'est donc qu'un changement de taille (ou de représentation), puisque L'on utilise plus que deux éléments au lieu des 26 lettres de L'alphabet. La plupart Des bons systèmes de cette catégorie combinent toujours des substitutions et des Transpositions, et les règles sont connues de tous, c'est pourquoi on appelle cette classe : le Chiffrement à Usage général. La sécurité de ces méthodes repose maintenant sur un Nouveau concept clé.

3.2.1 Chiffrement symétrique

3.2.1.1 Principe

Ce type de chiffrement est aussi appelé chiffrement à clé privée, du fait que le chiffrement se fait via une clé qui doit rester secrète et qui ne doit être connue que par les personnes devant chiffrer et déchiffrer les messages. La sécurité de la méthode de chiffrement réside donc dans la difficulté de

Trouver cette clé.

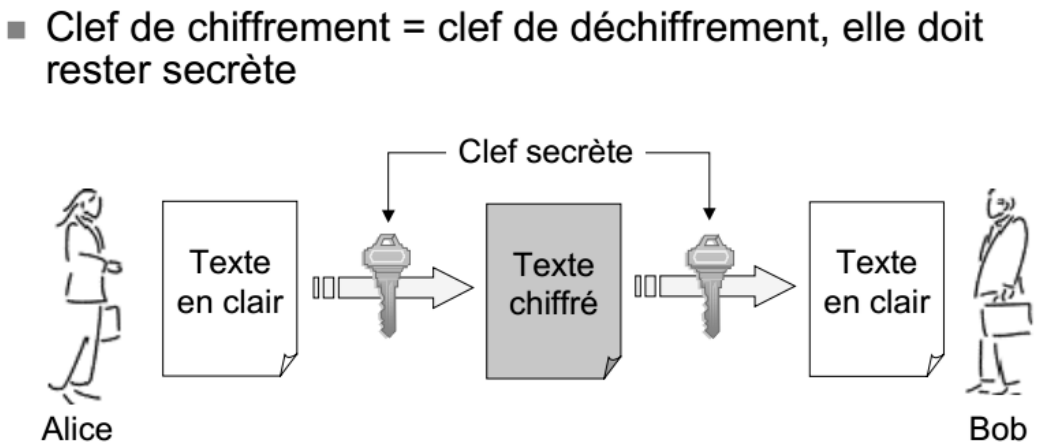


Figure I.3: La cryptographie symétrique

3.2.1.2 Modes de chiffrement

Il y a deux moyens de chiffrer les données avec les algorithmes à clé privée : le chiffrement Par Bloc et Le chiffrement par flot aussi appelé chiffrement par flux:

a) Chiffrement par bloc

Dans ce monde de cryptage, le texte clair est fractionné en blocs de même langueur à L'aide d'une clef unique. Les algorithmes de chiffrement par blocs sont en général Construit sur un modèle emploie une Fonction **F** qui prend en paramètre une clef **K** et Un message de **n** bits. **F** est répétée un certain nombre de Fois, en parle de ronde. A chaque Ronde, la clef **K** utilisée est changée et le message que l'on chiffre est Le résultat de L'itération précédente.

1- Electronique code book (ECB)

Il s'agit du mode le plus simple. Le message à chiffrer est subdivisé en plusieurs blocs qui Sont chiffrés Séparément les uns après les autres. Le gros défaut de cette méthode est que Deux blocs avec le même en cherchant les séquences identiques. On obtient dès lors un « dictionnaire de codes » avec les correspondances entre le clair et le chiffré d'où Le terme *code book*.

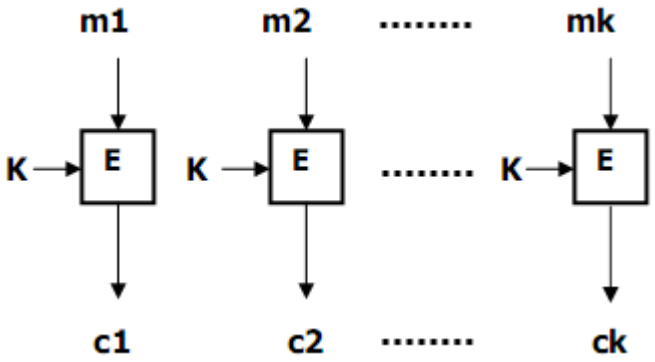


Figure I.4: Chiffrement en mode ECB

2- Cipher Block Chaining (CBC)

Dans ce mode, on applique sur chaque block un 'OU exclusif' avec le chiffrement du bloc précédent Avant qu'il soit lui-même chiffré. De plus, afin de rendre chaque message unique, un vecteur D'initialisation (IV) est utilisé.

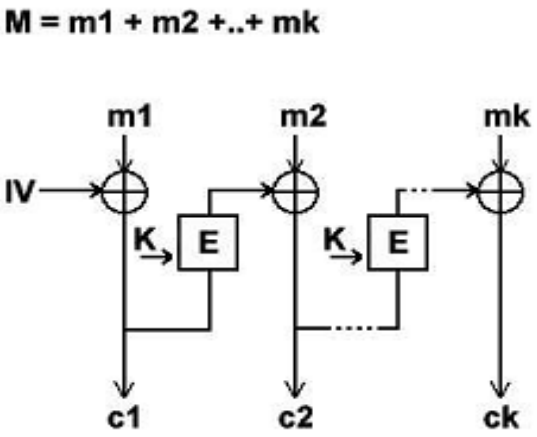


Figure I.5: Chiffrement en mode CBC.

3- Cipher Feedback (CFB)

Ce mode et les suivants agissent comme un chiffrement par flux. Ils génèrent un Flux de clés qui Est ensuite appliqué au document original. Dans ce mode, le flux de clé est Obtenu en chiffrant précédent Bloc chiffré. CFB est un chiffrement par flot. Son grand Intérêt est qu'il ne nécessite que la fonction de Chiffrement, ce qui le rend moins cher à Câbler ou programmer pour les algorithmes ayant une fonction De chiffrements différents de la fonction de déchiffrement.

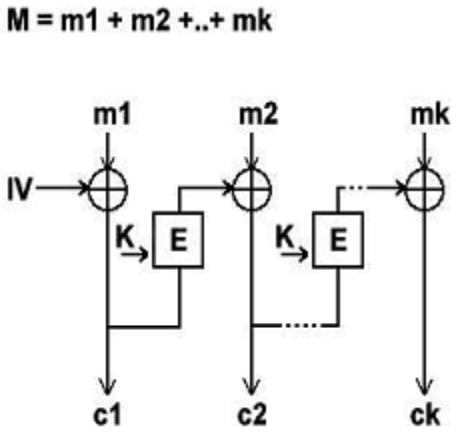


Figure I.6 : Chiffrement en mode CFB.

4- Output Feedback (OFB)

C'est un mode de chiffrement de flot qui possède les mêmes avantages que CFB. De Plus, il est Possible du pré calculé en chiffrant successivement le vecteur d'initialisation. Il n'est donc sûr que si la fonction de chiffrement alliée à la clé forment une bonne Suite pseudo-aléatoire . Il présente beaucoup de problèmes de sécurité et il est peu conseillé sauf dans le cas où sa longueur est égale à Celle de l'algorithme utilisé.

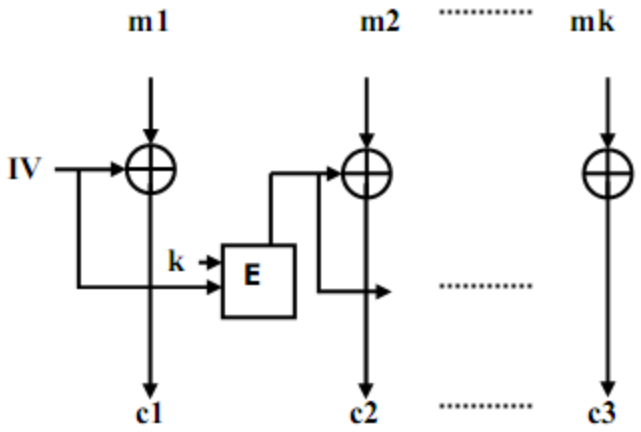


Figure I.7 : Chiffrement en mode OFB

b) Chiffrement par flux

Le chiffrement par flux traite les données comme elles arrivent, élément par élément. Sur Les systèmes Informatiques et électroniques, un élément est composé d'un bit, chaque bit est donc traité séparément.

Le chiffrement par flux le plus simple à comprendre est sûrement le chiffrement via le ou Exclusif. Ce ou exclusif, ou **XOR**, est un opérateur logique défini dans l'algèbre de Boole, qui, pour deux valeurs Données pouvant être soit vraies (1 en binaire) soit fausses (0 En binaire), renvoie vrai dans le cas où une Et une seule des valeurs données est vraie ; Dans les autres cas, le ou exclusif nous renverra faux.

Cet opérateur est représenté par le symbole (+) et on peut le définir simplement Comme suit, en considérant A et B comme des bits différents (valant 0 ou 1).

A (+) B = 1 B (+) A = 1 A (+) A = 0

Leurs avantages principaux sont :

- Du fait que la transformation (méthode de chiffrement) peut être changée à chaque symbole du texte clair et du fait qu'ils soient extrêmement rapides.
- De plus, ils sont utiles dans un environnement où les erreurs sont fréquentes car ils ont l'avantage de ne pas propager les erreurs (diffusion).
- Ils sont aussi utilisés lorsque l'information ne peut être traitée qu'avec de petites quantités de symboles À la fois, comme par exemple si l'équipement n'a pas de mémoire physique ou une mémoire tampon très limitée.

3.2.1.3 DES (Data Encryptions Standard)

Au début des années 70, la NBS (National Bureau of Standards) a lancé un appel Dans le Fédérale Registre pour la création d'un algorithme de cryptage répondant aux Critères suivants :

♦ ayant un haut niveau de sécurité lié à une clé

♦ Compréhensible

♦ Ne devant pas dépendre de la confidentialité de l'algorithme

♦ Adaptable et économique

♦ Efficace et exportable

Fin 1974, IBM propose Lucifer, qui grâce à la NSA (National Security Agency) est modifié Pour donner Le DES (Data Encryptions Standard) en 1976. Le DES a finalement été Approuvée 1978 parle NBS DES, Est le système de cryptage utilisé aujourd'hui en Etats-Unis Pour certaines banques. Il est réactualisé tous les 5 ans.

C'est un système de chiffrement par blocs de 64 bits, dont le dernier octet (8 bits) sert de Test de parité (Pour vérifier l'intégrité des données). Il consiste à faire des combinaisons, Des substitutions et des Permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans Les deux sens (pour le décryptage). La clé est Codée sur16 bits et formée de 16 blocs de 4 bits, Généralement notés k0 à k15. Etant donné que 8 bits de la clé sont réservés pour le test de la parité,« Seulement » 56 bits servent Réellement à chiffrer, ce qui représente tout de même 256 possibilités, C'est-à-dire 72*1015 clés possibles...

Les grandes lignes de l'algorithme sont les suivantes :

- ✓ Fractionnement du texte en blocs de 64 bits
- ✓ Permutation des blocs
- ✓ Découpage des blocs en deux parties : gauche et droite
- ✓ Etapes de permutation et de substitution répétées 16 fois (appelées rondes)
- ✓ Recollement des parties gauche et droites puis permutation initiale inverse.

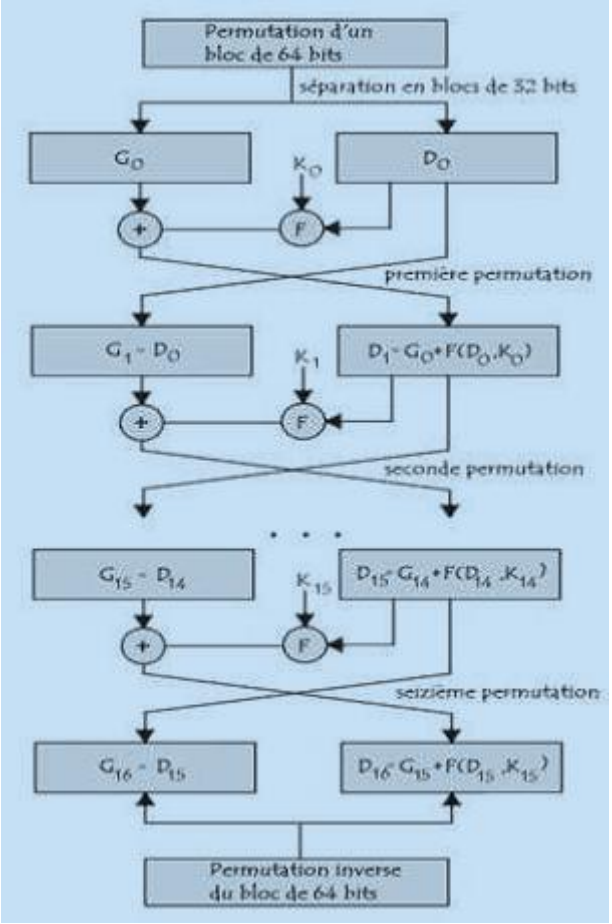


Figure I.8: chiffrement par DES

3.2.1.4 Le Triple DES

Le Triple DES (aussi appelé 3DES) est un algorithme de chiffrement symétrique par bloc, Enchaînant 3 applications successives de l'algorithme DES sur le même bloc de données de 64 bits, avec 2 ou 3 clés DES différentes.

Cette utilisation de trois chiffrements DES a été développée par Walter Tuchman (chef du projet DES chez IBM), il existe en effet d'autres manières d'employer trois fois DES mais elles ne sont pas Forcément sûres. La version de Tuchman utilise un Chiffrement, suivi d'un déchiffrement pour Se conclure à nouveau par un chiffrement. Le Triple DES est généralement utilisé avec seulement deux clés différentes. Le mode D'usage standard

Est de l'utiliser en mode EDE (Encryptions, Décryptions, Encryptions, c'est-à-dire Chiffrement, Déchiffrement, Chiffrement) ce qui le rend compatible avec DES quand on Utilise trois fois la même clé. Dans le cas d'une implémentation matérielle cela permet D'utiliser le même composant pour respecter le Standard DES et le standard Triple DES.

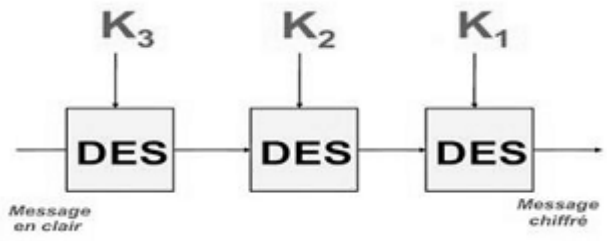


Figure I.9 : L'algorithme TDES (192 bits).

3.2.1.5 IDEA (International Data Encryptions Algorithme)

Développé à Zurich en Suisse par Xuejia Lai et James Massey en 1992, le Chiffrement par Blocs IDEA (International Data Encryptions Algorithme) utilise des Blocs de 64 bits et est Contrôlé par une clé de 128 bits. Malgré le fait qu'IDEA 'est pas un chiffrement Feistel, le déchiffrement est effectué avec La même Fonction que celle utilisée dans le chiffrement. L'algorithme a innové dans son Domaine en utilisant des opérations de trois groupes algébriques Différents. Le processus De chiffrement est le même Que pour le déchiffrement, à moins d'une Utilisation de Différentes sous-clés, ce qui est rare. En gros, Le processus se résume

à huit étapes de Chiffrement identiques, les rounds, suivis d'une transformation Au bloc de sortie. Contrairement au DES et à Blow Fish, IDEA n'utilise aucune S-Box. L'algorithme peut être utilisé avec les modes d'opération ECB, CBC, CFB et OFB. Sa Vitesse est environ la même que le DES.

3.2.1.6 AES (Advanced Encryptions Standard)

AES est le sigle d'Advanced Encryptions Standard. Il s'agit d'un algorithme de Chiffrement Symétrique, choisi en octobre 2000 par le NIST pour être le nouveau Standard de chiffrement pour Les organisations du gouvernement des Etats-Unis. Ce Faisant, l'AES remplace le DES (choisi comme standard dans les années 1970) qui de Nos jours devenait obsolète. L'AES a été adopté par le NIST (National Institute of Standards and Technologie) en 2001. De plus, son utilisation est très pratique Car il Consomme peu de mémoire et n'étant pas basé sur des schémas de Feistel, sa Complexité est moindre et Il est plus facile à implémenter. Utilise des clés de Tailles 128, 192 et 256 bits.

3.2.2 Chiffrement asymétrique

3.2.2.1 Principe

Inventé en 1977 par Diffie et Hellman, le chiffrement asymétrique, aussi connu sous le nom de Chiffrement à clé publique permet de résoudre le problème de communication des clés du chiffrement Symétrique, et permet aussi l'authentification, en s'aidant des fonctions de hachage. Chaque utilisateur possède alors deux clés : une clé publique qu'il distribue à tout le monde Ainsi qu'une Clé privée, qu'il garde secrète. La clé publique permettra de chiffrer un message, Que l'utilisateur pourra Déchiffrer grâce à sa clé privée. Ainsi, seule la personne en possession de la clé de déchiffrement (Privée), le destinataire donc, pourra déchiffrer le message.

Figure I.10 : La cryptographie asymétrique

3.2.2.2 RSA (Rivest Shamir Adleman)

RSA est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et Plus Généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. RSA a été breveté par le Massachusetts Institute of Technologie (MIT) en 1983 aux Etats-Unis. Le brevet a expiré le 21 septembre 2000.

Cet algorithme est fondé sur l'utilisation d'une paire de clés composée d'une clé Publique pour chiffrer (Respectivement vérifier) et d'une clé privée pour déchiffrer (Respectivement signer) des données Confidentielles.

La clé publique correspond à une clé qui est accessible par n'importe quelle personne Souhaitant chiffrer des informations, la clé privée est quant à elle réservée à la personne Ayant créé la paire de clés. Lorsque deux personnes, ou plus, souhaitent échanger des données confidentielles, Une personne, nommée par convention Alice prend en charge la création de la paire de clés, Envoie sa clé Publique aux autres personnes Bob, Carole… qui peuvent alors chiffrer les données confidentielles a l'aide de celle-ci puis envoyer les données chiffrées à la personne ayant Créé la paire de clés, Cette Dernière peut alors déchiffrer les données confi Clé publique clé Secrète(**e**, **n**).

3.2.2.3 El Gamal

L'algorithme El Gamal est un algorithme de cryptographie asymétrique basé sur les Logarithmes Discrets. Il a été créé par Taher El Gamal. Cet algorithme est utilisé par le logiciel libre GNU Privacy Giard, de récentes versions de PGP, et d'autres systèmes de chiffrement, et n'a jamais été sous la Protection d'un brevet contrairement à RSA. Il

Peut être utilisé pour le chiffrement et la signature Électronique. L'algorithme DSA du NIST est basé sur El Gamal [6].

L'algorithme fonctionne comme suit **:**

Il s'agit d'un système à clé publique dont la sécurité repose, comme le protocole de Diffie et Hellman, Sur la difficulté de calculer le logarithme discret.

- Alice calcule h= g^x avec g, h∈ Z_p pour un grand nombre premier p, et divulgue sa clé publique (p, g, h). La valeur x est sa clé privée.
- Si Bob veut envoyer un message à Alice, il convertit d'abord son message sous la forme d'un nombre /m ∈ Z_p
- Bob génère un nombre entier k aléatoirement et calcule c1=g^k et c2=$m.h^k$.Il envoie (c1, c2) à Alice.
- Alice peut reconstruire le message initial m en calculant c2 /c1.$c1^x$

$$\frac{c_2}{c_1^x} = \frac{m \cdot h^k}{g^{xk}} = \frac{m \cdot g^{xk}}{g^{xk}} = m$$

Il n'est pas obligatoire d'utiliser Z_p . Tout groupe cyclique convient.

3.2.3 Comparaison entre la cryptographie symétrique et asymétrique

Attribut	Cryptographique Symétrique	Cryptographique asymétrique
Année d'existence	Des milliers	Moins de 50
Utilisation actuelle Principale	Chiffrement des données en gros	Echange des clés, signature numérique
Standard actuel	DES, IDEA et Rijndeal	RSA, diffie - Hellman, DSA (la technologie ECC est une nouvelle venue prometteuse)
Vitesse de chiffrement/ Déchiffrement	Rapide	Lent
Clés	Secrète partagée par au moins deux personnes	Privée : gardée cachée par un personne Publique : largement distribuée
Echange des clés	Transfert risqué et difficile pour une clé secrète	Simple, moins risqué de remettre une clé publique Clé privée jamais partagée
Longueur de la clé	56 bits obsolètes 128 bits considérés comme sûr	1024suggérée (RSA) certains utilisateur exigent 2048~172(courbe elliptique)
Confidentialité, Authentification, intégrité	Oui	Oui

du message		
Non-répudiation	Non besoin d'un tiers de confiance servant de témoin	Attaques signatures numériques : pas besoin de tiers
Attaques	Oui	Oui

Tableau I.3 : Comparaison entre la cryptographie symétrique et asymétrique

3.3 Les algorithmes associés

3.3.1 Cryptographie hybride

La cryptographie asymétrique est intrinsèquement lente à cause des calculs complexes qui y sont associés, alors que la cryptographie symétrique brille par sa rapidité. Toutefois, cette dernière souffre d'un grave lacune, on doit transmettre les clés de manière sécurisée (sur un canal authentifié). Pour Pallier ce défaut, on recourt à la cryptographie asymétrique qui travaille avec une paire de clés : la clé procèdent de la manière suivante. Une clé aléatoire est générée pour l'algorithme symétrique (3DES, IDEA, AES et bien d'autres encore), Cette clé fait généralement entre 128 Et 512 bits selon les algorithmes. L'algorithme De Chiffrement symétrique est ensuite utilisé pour Chiffrer Le message. Dans le cas d'un Chiffrement par blocs, on doit utiliser un mode d'opération Comme par Exemple CBC, Cela permet de chiffrer un message de taille supérieure à celle d'un bloc. La Clé Aléatoire quant à elle, se voit chiffrée grâce à la clé publique du destinataire, c'est ici Qu'intervient la Cryptographie Asymétrique (RSA ou Diffie -Hellman). Comme la clé est courte, ce chiffrage prend Peu de temps. Chiffrer l'ensemble du message avec un Algorithme asymétrique serait bien plus lourd, c'est pourquoi on préfère passer par un algorithme symétrique. Il suffit ensuite d'envoyer le message chiffré avec L'algorithme symétrique et accompagné de la clé chiffrée correspondante. Le destinataire déchiffre la clé Symétrique avec sa clé privée et via un déchiffrement symétrique, retrouve le message.

- **PGP** *(* Pretty *Good* Privacy*)* est un crypto système inventé par Philip Zimmermann , un Analyste informaticien. Philip Zimmermann a travaillé de 1984 à 1991 sur un programme permettant de faire fonctionner RSA sur des Ordinateurs personnels (PGP).Il est très rapide et sûr ce qui le rend Quasiment impossible à cryptanalyse. PGP est une combinaison des meilleures fonctionnalités de la Cryptographie asymétrique et symétrique. Il Fonctionne suivant le principe suivant :

• **Compression :** Le texte à envoyer est compressé. Cette étape permet de réduire le temps de transmission des données, d'économiser l'espace de disque et améliore Également la sécurité.

• **Chiffrement du message :** Une clé de session aléatoire est générée. Le message est Chiffré par un Algorithme symétrique à l'aide d'une clé de session. L'algorithme utilisé A varié au cours de temps : Il s'agissait au début d'IDEA, puis de CAST.

• **Chiffrement de la clé de session :** La clé de session est chiffrée en utilisant la clé publique du Destinataire et l'algorithme RSA.

• **Envoi et réception du message :** L'expéditeur envoie couple (message chiffré, Clé) de session Chiffrée au destinataire. Celui-ci récupère d'abord la clé de session, en Utilisant sa clé privée, puis il déchiffre le Message grâce à La clé de session.

3.3.2 Les fonctions de hachage

Les fonctions de hachage sont des fonctions à sens unique, et produisent à partir d'un Texte d'une Longueur quelconque, une somme de contrôle (aussi appelée empreinte ou hash) de longueur fixe, qu'on peut voir comme un résumé du texte (à partir duquel on ne Peut pas retrouver le texte d'origine, ou Du moins très difficilement).

Les fonctions de hachage doivent être résistantes aux collisions, c'est-à-dire qu'on ne peut pas trouver Facilement deux messages différents ayant la même empreinte.

Fonctions de hachage usuelles :

• MD4 et MD5 (Message Digest) furent développées par Ron Rivest. MD5 produit des hachés de 128 Bits en travaillant les données originales par blocs de 512 bits.

• SHA-1(Secure Hash Algorithme 1), comme MD5, est basé sur MD4.Il fonctionne également à partir de Blocs de 512 bits de données et produit par contre des condensés de 160 bits en sortie. Il nécessite donc plus de ressources que MD5.

• SHA-2 (Secure Hash Algorithme 2) a été publié récemment et est destiné à remplacer SHA-1. Les Différences principales résident dans les tailles de hachés possibles : 256, 384 ou 512 bits. Il sera bientôt La nouvelle référence en termes de fonction de hachage.

• RIPEMD-160 (Ripe Message Digest) est la dernière version de l'algorithme RIPEMD. La version Précédente produisait des condensés de 128 bits mais présentait des failles de sécurité importantes. La Version actuelle reste pour l'instant sure; elle produit comme son

nom l'indique des condensés de 160 Bits. Un dernier point la concernant est sa relative gourmandise enfermes de ressources et en Comparaison Avec SHA-1 qui est son principal concurrent.

• **Tigre:** Tigre est une fonction de hachage cryptographique conçue par Ross Anderson et Eli Biham en 1996. Tigre fournit une empreinte sur 192 bits mais des versions sur 128 et 160 bits existent aussi. Ces versions raccourcies prennent simplement les premiers bits de la signature de 192 bits.

3.3.3 Signature numérique

La signature numérique (parfois appelée signature électronique) est un mécanisme permettant de Garantir l'intégrité d'un document électronique et d'en authentifier l'auteur, par analogie avec la signature Manuscrite d'un document papier. Un mécanisme de signature numérique doit présenter les propriétés Suivantes :

- Il doit permettre au lecteur d'un document d'identifier la personne ou l'organisme qui a apposé sa Signature.
- Il doit garantir que le document n'a pas été altéré entre l'instant où l'auteur l'a signé et le moment où le Lecteur le consulte.

  Pour cela, les conditions suivantes doivent être réunies :

• **Authentique:** L'identité du signataire doit pouvoir être retrouvée de manière certaine.

• **Infalsifiable:** La signature ne peut pas être falsifiée. Quelqu'un ne peut se faire passer pour un autre.

• **Non réutilisable:** La signature n'est pas réutilisable. Elle fait partie du document signé et ne peut être déplacée sur un autre document.

• **Inaltérable:** Un document signé est inaltérable. Une fois qu'il est signé, on ne peut plus le modifier.

• **Irrévocable:** La personne qui a signé ne peut le nier. La signature électronique n'est Devenue possible qu'avec la cryptographie asymétrique. Elle se différencie de la signature Écrite par le fait qu'elle n'est pas visuelle, mais correspond à une suite de nombres

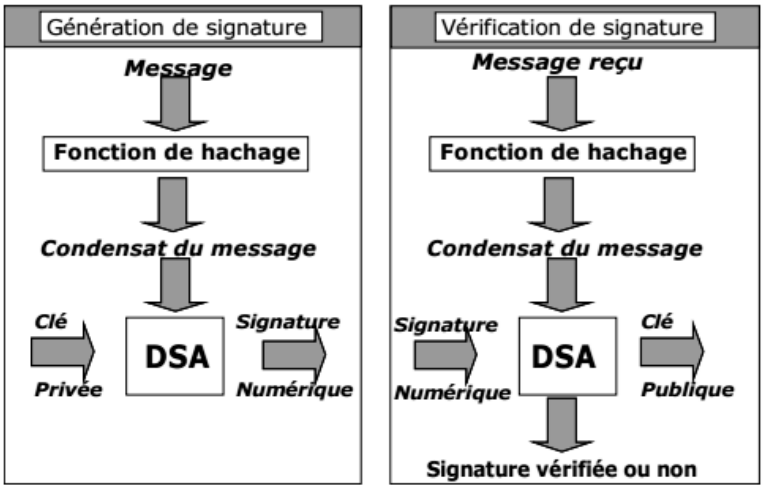


Figure I.11: schéma de hachage et signature numérique

3.3.4 Certificat numérique

C'est un certificat qui permet à une entreprise de s'authentifier de manière certaine, unique Et sécurisée Pour transmettre ou recevoir des informations numériques sensibles. Un certificat numérique est délivré par un organisme agréé par le ministère chargé Des finances. Cet Organisme est un Prestataire de Service de Certification Électronique (PS Ce). Un certificat Numérique Autorise son possesseur à signer des Engagements au nom de son entreprise dans les Procédures pouvant Être télétransmises : Immatriculation de véhicules, déclaration de TVA, marchés Publics, …Il peut se Présenter sous forme de clé USB ou de carte à puce. Il existe aussi des certificats Numériques sous forme logicielle, mais ils ne sont pas autorisés pour le SIV (Système D'Immatriculation des Véhicules) [6].

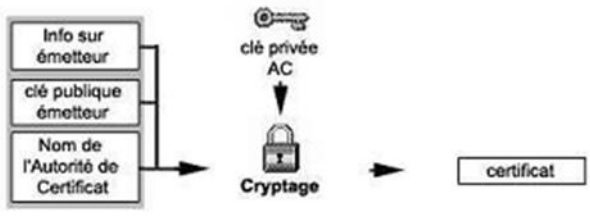


Figure I.12: certificat numérique

3.4 .Cryptographie quantique

La communication de données confidentielles par un canal de transmission classique (Par exemple Internet) nécessite l'utilisation d'algorithmes de cryptographie classiques : Algorithmes de chiffrement asymétrique tels que RSA, ou de chiffrement symétrique (Triple DES, AES). Dans le cas du chiffrement symétrique, les deux interlocuteurs doivent posséder a priori une clé secrète, C'est-à-dire qui ne soit connue que d'eux deux. Se pose alors la question suivante : comment transmettre une clé de cryptage entre deux interlocuteurs à Distance, à la demande, et avec une sécurité démontrable? Actuellement, la technique se Rapprochant au Mieux de ces trois critères est une transmission physiquement sécurisée, de Type valise diplomatique. La cryptographie quantique cherche à répondre à ces trois critères En transmettant de l'information entre Les deux interlocuteurs en utilisant des objets Quantiques, et en utilisant les lois de la physique quantique et de la théorie de l'information Pour détecter tout espionnage de cette information. S'il n'y a pas eu Espionnage, une Clé parfaitement secrète peut être extraite de la transmission, et celle-ci peut être utilisée dans tout algorithme de chiffrement symétrique afin de transmettre un message. Pourquoi utiliser le système de cryptographie quantique pour transmettre une clé, et non le Message en Lui-même ?

Pour deux raisons essentielles :

- Les bits d'informations communiqués par les mécanismes de la cryptographie quantique ne peuvent Être aléatoires. Ceci ne convient pas pour un message, mais convient Parfaitement bien à une clé secrète, qui doit être aléatoire.
- Même si le mécanisme de la cryptographie quantique garantit que l'espionnage de la communication sera toujours détecté, il est possible que des bits d'informations entrent En possession de l'espion Avant que celui-ci ne soit détecté. Ceci est inacceptable pour un Message, mais sans importance pour une clé aléatoire qui peut être simplement jetée En cas d'interception.

4. La cryptanalyse

4 .1. Définition

Si le but de la cryptographie est d'élaborer des méthodes de protection, le but de la Cryptanalyse est au contraire de casser ces protections. On appelle attaque une

Tentative de cryptanalyse. Un algorithme de chiffrement est considéré comme sûr si ce dernier résiste à des attaques dont le besoin en temps ou en espace de mémoire est Raisonnable. En d'autres termes, ces derniers ne permettent Pas de découvrir la clé Secrète, ni le message en clair. Dans la plupart des cas, le chiffrement est D'autant plus Sûr que la clé a une grande taille. Par conséquent, la taille de la clé est un paramètre Essentiel de la sureté d'un algorithme de chiffrement. Dans la littérature de la cryptanalyse, Il existe un Vaste choix de méthode d'attaque pour retrouver la clé secrète. Ces Attaques différentes par le coût, Le temps de calcul, la connaissance du matériel, L'algorithme de chiffrement utilisé.

4 .2. Classification des attaques en fonction des données disponible

Les techniques d'attaques peuvent être classifiées dans 4 catégories :

a)Attaque à texte chiffré connu :

Cette famille d'attaques fait comme hypothèse de base que l'attaquant a uniquement Les Messages chiffrés.

b) Attaque à texte clair connu :

Cette famille fait comme supposition qu'Eve a les textes clairs et les chiffrés de ces clairs.

c) Attaque à texte chiffré choisi :

Cet ensemble d'attaques fait comme hypothèse que l'attaquant peut choisir les messages chiffrés et recevoir leur clair.

d) Attaque à texte clair choisi:

Cet ensemble d'attaques suppose qu'Eve peut choisir les textes clairs Et obtenir leur Chiffré. Cette Attaque est largement réalisable dans les algorithmes asymétriques où la clé de Chiffrement est public. Ainsi, l'attaquant peut chiffrer l'ensemble des messages qu'il Choisit. Le but final étant de choisir les Bons couples de clair/chiffré pour retrouver une Information utile.

4.3 Les différentes attaques

a) L'analyse des fréquences

Exposée pour la première fois par Al-Kindi au IXe siècle, l'analyse fréquentielle permet De déchiffrer des chiffrements simples. Dans le cas d'une substitution, les caractères sont Représentés par autre chose, Mais leurs fréquences d'apparition ne changent pas ; on peut

Donc alors facilement retrouver le message initial, connaissant les fréquences normales D'apparition des caractères dans la langue du message.

b) L'indice de coïncidence

L'indice de coïncidence, inventé en 1920 par William F. Friedman, permet de Calculer la probabilité de Répétitions des lettres du message chiffré. Il est souvent Couplé avec l'analyse fréquentielle. Cela permet de savoir le type de chiffrement d'un Message (chiffrement mono-alphabétique ou poly-alphabétique) ainsi que la longueur Probable de la clé.

c) L'attaque par mot probable

L'attaque par mot probable consiste à supposer l'existence d'un mot probable dans Le message Chiffré. Il est donc possible d'en déduire la clé du message si le mot Choisi est correct. Ce type D'attaque a été mené contre la machine Enigma durant la Seconde Guerre mondiale.

d) L'attaque par force brute

L'attaque par force brute consiste à tester toutes les solutions possibles de mots de Passe ou de clés. C'est le seul moyen de récupérer la clé dans les algorithmes les plus Modernes et encore inviolés comme AES. Il est peu utilisé pour des mots de passe Possédant un très grand nombre de caractères car le temps Nécessaire devient alors Trop important. De même plusieurs brevets rendent cette méthode inefficace, Comme celui De Bell ou d'IBM.

e) Attaque par paradoxe des anniversaires

Le paradoxe des anniversaires est un résultat probabiliste qui est utilisé dans les Attaques contre les Fonctions de hachage. Ce paradoxe permet de donner une borne Supérieure de résistance aux collisions D'une telle fonction. Cette limite est de l'ordre de la racine de la taille de la sortie, ce qui signifie que, pour un algorithme comme MD5 (Empreinte sur 128 bits), trouver une collision quelconque avec 50% de chance Nécessite 2^{64} hachages d'entrées distinctes.

f) Cryptanalyse différentielle

Découverte à la fin des années 1980, la cryptanalyse différentielle analyse comment Les différences dans les messages clairs influencent les différences dans les messages Chiffrés. L'étude de ces différences Renseigne alors l'attaquant de certains bits de la clé. Il faut Donc alors être dans une situation d'attaque à Texte clair choisi (l'attaque à texte clair Connu est aussi possible, mais nécessite un plus grand nombre De textes connus). Nous N'allons pas plus développer cette attaque, qui serait bien trop longue à décrire en Détail dans ce travail.

g) Cryptanalyse linéaire

La cryptanalyse linéaire, due à Mitsuru Matsui, consiste à faire une approximation Linéaire de la structure interne de la méthode de chiffrement. Elle remonte à 1993 et S'avère être l'attaque la plus Efficace sur DES. Les algorithmes plus récents sont Insensibles à cette attaque.

h) Autres attaques

Il existe encore d'autres types d'attaque non abordées ici. Le secteur de la Cryptanalyse est en perpétuelle évolution, chacun peut trouver de nouvelles méthodes De cryptanalyse avec beaucoup de travail et de connaissances.

Les attaques présentées ici ne sont faisables que sur des algorithmes de chiffrements Symétriques, Les systèmes de chiffrements utilisant un lien entre la clé publique et la clé Privée, on essayera plutôt d'exploiter ce lien pour trouver la clé privée. Dans le cas de RSA, il faut factoriser le nombre e, afin De trouver ses deux diviseurs, p et q, on pour alors Retrouver facilement la clé privée d. La sécurité de Cet algorithme réside alors dans la Difficulté de factoriser un nombre qui est le produit de deux grands Nombres premiers [7].

4 .4 Réussite de l'attaque

Il existe plusieurs découvertes qui permettent de dire qu'u algorithme déchiffrement Est cassé. la liste suivante regroupe des exemples :

1. Si nous retrouvons la clé de chiffrement
2. Si nous retrouvons le message clair
3. Si nous retrouvons un sous-ensemble du message clair

4. Si nous retrouvons une information du message clair

Il existe un ensemble très vaste d'attaques. En effet, il existe autant de techniques D'attaque que d'algorithmes de chiffrement et certaines de ces méthodes sont Généralisées pour une famille d'algorithmes de chiffrement. Ni la liste exhaustive De ces attaques, ni de leur contremesure ne pourrait être donnée ici. Pour les lecteurs Désirant apprendre des contremesures sur les attaques présentes dans ce document, Est un bon choix. Toutefois, essayons de détailler quelques-unes de ces attaques pour Voir d'où a débuté la cryptanalyse jusqu'à arriver à l'attaque utilisée dans ce Travail Avec l'aide de l'apprentissage automatique [8].

5. Conclusion

Dans ce chapitre, nous avons tenté de dresser une généralité sur la cryptographie, Et son Importance Dans la protection des informations pendant le transfert. Ensuite, Nous avons Abordé la Classification des algorithmes cryptographique Avec des exemples sur chaque classe D'algorithme. Enfin, nous avons vu quelques Concepts sur la cryptanalyse et les différentes Techniques Utilisent dans cette Opération.

CHAPITRE II
SYSTÈME PGP

1. Introduction

Après l'étude des différentes classes des algorithmes cryptographique dans le chapitre précédent. Nous prenons dans ce chapitre le système PGP (Pretty Good Privacy) en détail, qui est un standard de cryptage symétrique et a symétrique largement adopté dans l'industrie du matériel cryptographique. Son principal avantage réside dans sa rapidité du chiffrement et du déchiffrement qui est nettement plus vite sur des cartes électroniques dédiées du point de vue hardware (cartes à puces, systèmes électroniques de communication).

2. Historique de système PGP

L'histoire de PGP est très complexe et les sources , bien que nombreuses, apparaissent souvent incomplètes , voire contradictoires. Toutefois 1983, quatre ans après la création de l'algorithme RSA (Rivest Shamir Adeleman), est considérée comme le point de départ de PGP.

L'informaticien Charlie Merrit, qui travaille sur une implémentation de RSA, prend contact avec Philip Zimmermann et lui demande d'effectuer des recherches concernant la possibilité d'implémenter RSA pour des ordinateurs privés. Charlie Meritt présente alors Philip Zimmermann à Jim Bidzos le président de RSA Data Security. On sait peu de choses de cet entretien, mais on peut affirmer qu'il fut primordial pour l'avenir de PGP : Zimmermann prétend que lors de cette réunion , Bidzos lui offrit des licences gratuites sur l'utilisation de l'algorithme RSA. Lorsque la première version de PGP est achevée, Zimmermann contacte Bidzos pour obtenir ces licences, mais ce dernier refuse, prétextant que celles-ci vont à l'encontre de la politique de la société. Zimmermann se retrouve confronté à un grave problème. Son logiciel utilise l'algorithme RSA alors qu'il n'en a pas obtenu les droits commerciaux[9].

De plus, peu de temps après ce refus de Bidzos, le Sénat des États-Unis vote une loi anti criminalité obligeant tout fabricant de crypto systèmes à inclure dans son produit une « porte dérobée » c'est-à-dire un moyen pour le gouvernement de retrouver de façon systématique le texte en clair depuis un texte crypté. Zimmermann , qui se refuse à ajouter une porte dérobée à PGP, se retrouve confronté à une double difficulté : un litige financier avec RSA Data Security et un risque imminent de voir son produit devenir illégal. Il s'empresse ainsi de mettre PGP à disposition du public en l'offrant en libre

téléchargement sur son site Internet. La première version publiquement connue de PGP (1.0) voit le jour en 1991.

Bidzos menace alors Zimmermann de poursuites judiciaires et ce dernier accepte de ne plus mettre PGP à disposition du public tant qu'il n'a pas obtenu explicitement l'accord de RSA Data Security. Cette décision arrive trop tard, PGP a déjà fait son chemin sur Internet.

L'algorithme original de PGP (appelé Bass-O-Matic) développé par Zimmermann lui-même présente des défauts de sécurité et est rapidement abandonné au profit d'un autre algorithme développé en Suisse et connu sous le nom d'IDEA (International Data Encryptions Algorithm) qui est inclus dès la version 2.0 de PGP.

En 1993, le gouvernement des États-Unis d'Amérique lance des poursuites judiciaires contre Zimmermann pour une violation des restrictions liées à l'export des produits cryptographiques, PGP pouvant être téléchargé librement partout dans le monde.

Après trois ans d'enquête, l'investigation est abandonnée sans que le gouvernement en explique le motif.

Dans le même temps, Zimmermann , qui cherche à rendre PGP légitime , fonde la société Viacrypt et produit Viacrypt PGP qui devient la première version légale de PGP. Afin de compenser le prix des licences RSA, il est vendu aux alentours de 150 $ US. Toutefois , une version gratuite de PGP pour usage non commercial est systématiquement mise à disposition par Viacrypt.

En 1996, la société PGP Inc. Est créée et absorbe Viacrypt.

En janvier 1998 c'est au tour de Network Associates d'acquérir PGP Inc. Mais ceux-ci rompent avec la tradition de PGP voulant que le code source soit mis à disposition du public afin qu'il puisse s'assurer de la qualité du produit et de l'absence de portes dérobées dans le logiciel.

En 1998 est proposé un standard de l'IETF (Internet Engineering Task Force) nommé Open-PGP et décrit dans la RFC 2440. Il décrit les formats des messages, signatures ou clefs, qui peuvent être envoyés par des programmes crypto systèmes en se basant sur PGP.

On peut désormais considérer que PGP implémente cette norme tout en offrant des fonctionnalités supplémentaires. Mais également, que d'autres logiciels peuvent dès lors implémenter la même norme que PGP et ce, gratuitement, à condition ne pas utiliser d'algorithme breveté payant.

En 2002, Network Associates vend ses activités dans PGP à PGP Corporation qui, la même année, rend de nouveau le code source de PGP accessible pour des revues par les pairs.

En 2006, PGP Corporation est toujours propriétaire de PGP, et la version actuelle PGP 9.0 n'a plus beaucoup de points communs avec PGP 1.0. D'un système de cryptage en ligne de commande offrant peu de possibilités , PGP est devenu une offre logicielle complexe . Elle se compose toujours du logiciel, mais il est désormais muni d'une interface graphique avancée et de diverses possibilités telles que le cryptage de disque dur, de mails via des plug-ins pour les différents clients de messagerie, etc. des bibliothèques de fonctions sont également vendues et permettent d'incorporer les fonctionnalités de PGP au sein des développements logiciels[9]

3. principe de fonctionnement de PGP

Le système de PGP est un système hybride que l'on peut classer dans les systèmes « à clef de session », C'est-à-dire un système qui utilise à la fois le principe du chiffrement à clef Privée et le principe du Chiffrement à clef publique.Considérons les différentes étapes du transfert d'un message crypté avec PGP de

L'expéditeur X Vers le destinataire Y.

(i) X doit envoyer le message crypté à Y.

(ii) Y crée une paire de clefs via l'algorithme RSA. Il transmet sa clef publique à X.

(iii) X saisit le texte en clair à envoyer. Ce texte est tout d'abord compressé ce qui offre un double

Avantage :

- la taille des données à transférer est réduite
- les risques de décryptage sont minimisés (la plupart des techniques de cryptanalyse se Basent sur le texte en clair obtenu. Si le texte obtenu est un texte compressé, il est, par exemple, Plus difficile de Calculer la probabilité de

retrouver telle ou telle lettre).Il faut noter que la compression n'est pas systématique. Si le taux de compression d'un fichier n'est pas Satisfaisant ou si le fichier est trop petit, cette étape n'est pas réalisée.

(iv) puis, X crée aléatoirement une clef secrète IDEA. (Nous reviendrons sur les clefs IDEA dans la partie Pratique). L'expéditeur chiffre le texte avec cette clef IDEA. Le Texte ainsi chiffré pourra être déchiffré Avec la même clef. Dans PGP, le message est alors crypté selon un système symétrique (à clef secrète) .

(v) Le destinataire Y ne connaissant pas cette clef, elle va lui être envoyée avec le message Crypté . toute fois pour éviter qu'elle soit interceptée , la clef sera également cryptée à L'aide de la Clef publique de Y. La clef privée IDEA est cryptée avec la clef publique de Y selon un système asymétrique (À clef Publique).

Finalement, le résultat obtenu contient :

- le texte chiffré avec la clef IDEA
- la clef IDEA chiffrée avec la clef publique RSA du destinataire.

À réception du message, le destinataire utilise sa clef privée RSA pour retrouver la valeur de la clef IDEA. Il utilise la clef obtenue pour déchiffrer le message reçu.

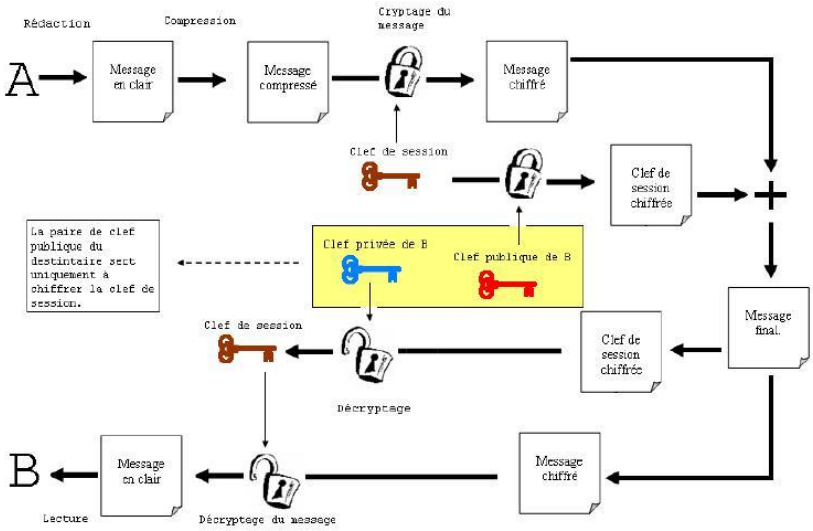


Figure II.1: Chiffrement &déchiffrement par système PGP

PGP possède plusieurs avantages :

- La rapidité : le message est chiffré par un cryptage symétrique. La clef IDEA est chiffrée de façon asymétrique. Toutefois le volume de données que représente cette clef est négligeable par rapport au volume de données que représente le message. Par conséquent, le temps de chiffrage global est proche de celui d'un système symétrique .
- Une plus haute sécurité qu'un système à clef symétrique. En effet, si l'on peut considérer que le niveau de sécurité du système PGP est celui de son maillon le plus faible - le système à clef privée servant à coder le message - il faut toutefois nuancer cette conclusion. Dans un système à clef privée standard, le canal d'échange de la clef est le point faible du système. Si l'on désire changer de clef afin de minimiser les risques, cela nécessite de définir un moyen d'échanger cette nouvelle clef, opération difficile à mettre en pratique. Dans PGP en revanche, la clef utilisée pour coder le message est nouvelle pour chaque message. Ce qui implique que pour effectuer une attaque il est nécessaire de casser au choix :
    - autant de clefs privées que de messages,
    - le système de clefs RSA, ce qui rend finalement PGP plus résistant qu'un système à clef privée classique [9] .

4. Présentation de les 'algorithmes utilisés par PGP

4.1. Algorithme RSA

4.1.1Présentation général.

Le principe du RSA est relativement simple. Utilisons deux utilisateurs classiques, Alice et Bob, Avec Bob qui veut envoyer un message à Alice mais en utilisant le RSA.

Alice va générer deux clés :

Une clé publique qu'elle diffusera aux personnes voulant lui parler. Cette clé sert à crypter Et uniquement crypter les messages, comme nous le verront plu tard on ne peut pas décrypter les messages avec la clé publique.

Une clé privée qu'Alice gardera bien cacher des autres utilisateurs. Cette clé sert à décrypter tous les messages qui ont été crypté avec sa clé publique.

Alice envoie donc sa clé publique a Bob pour qu'il puisse lui envoyer en message crypter. Puis Alice récupère le message crypté de Bob et le décrypter à l'aide de sa clé Privé.

Maintenant que nous avons nos couples de clés (publique et privé) nous pouvons encrypter nos message et les décrypter.

Pour cela on effectue les opérations suivantes :

Pour encrypter le message : c = m^e mod n

Pour décrypter le message : m = c^d mod n

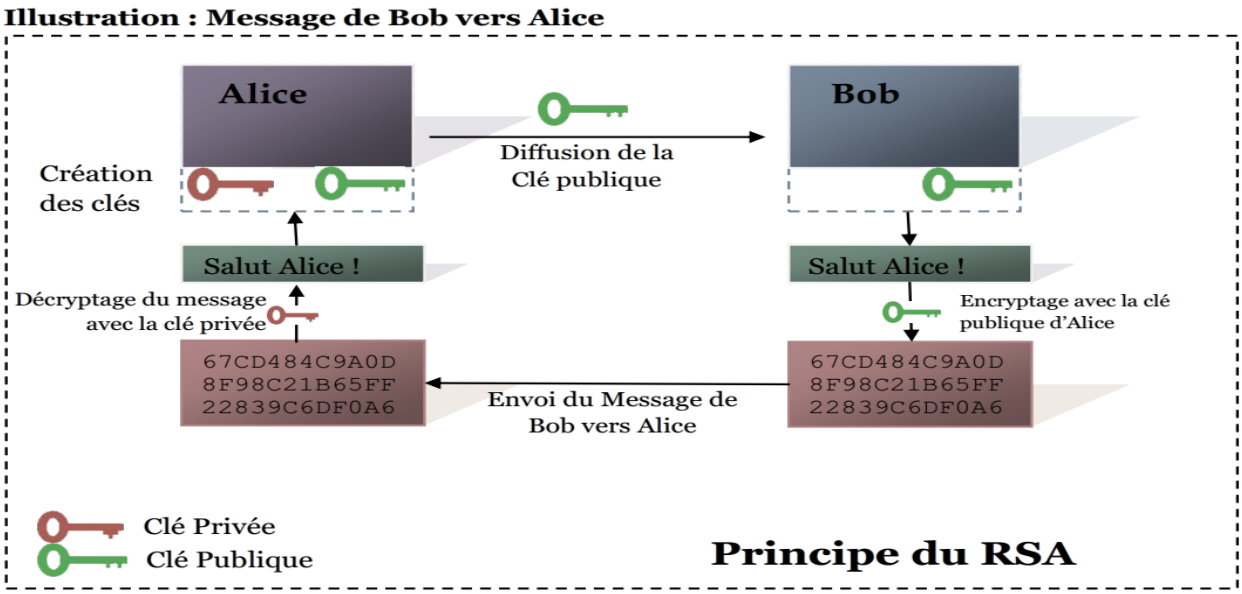


Figure II.2: chiffrement & déchiffrement par algorithme RSA

4.1.2. Base mathématique de RSA

4.1.2.1. Algorithme d'Euclide.

Avant d'aborder le théorème de Bachet – Bézout il faut déjà avoir compris le l'algorithme d'Euclide , Nous proposons donc de l'étudier rapidement ici. L'algorithme d'Euclide sert à calculer le PGCD (Plus Grand Commun Diviseur) mais aussi à calculer les différents coefficients dans la formule de Bézout.

Le calcul du PGCD de a et b , deux nombres entiers naturels utilise la division euclidienne de a par b, tel Que **a=b*q +r** avec **r<b**. Tout diviseur de et **b** divise **r=a-b .q**.

Par réciproque , tout diviseur commun De **b** Et **r** divise aussi **a=p*q +r**. Le calcul du PGCD de **a** et **b** se ramène au calcul de **b** et **r**. N'en Réitérant, Le dernier reste non nul est le PGCD recherché [10] .

Exemple :

Prenons deux nombres : 383 et 127

383 > 127 donc : 383 = 127 *3 + 2

127 > 2 donc 127 = 2*63 + 1

Donc PGCD (383 ; 127) = 1

On rappelle que si le PGCD de deux nombre est égale à 1, alors ils sont premiers entre eux.

4.1.2.2. Théorème de Bachet - Bézout.

Le théorème de Bachet – Bezout est très important dans le RSA car il permet de calculer **e** facilement. Il prouve l'existence d'une solution a*u + b*v = PGCD (a, b) Pour Le RSA on cherche à calculer **e** tel qu'il n'est aucun facteur commun avec (p-1) (q-1), c'est à dire que le PGCD est égale à 1 (Premier entre eux).

Exemple :

Reprenons l'exemple précédent avec 383 et 127.

On cherche à calculer u et v tel que 383*u + 127*v = 1

(A) 1 = 127 - 2 * 63 (Démontré précédemment avec Euclide)

(b) 2 = 383 - 127 * 3

On remplace (b) dans (A) :

1 = 127 - (383 - 127 * 3) * 63 (on développe)

1 = 127 - 383*63 + 127*189

1 = 127 (1 + 189) + 383 * (-63)

Donc on a bien 383*u + 127*v = 1 avec u= -63 et v= 190Pour le RSA, on prendra e = v = 190 car u<0.

4.1.2.3. L'opération Modulo (Mod)

L'opération modulo est une fonction fondamentale dans l'arithmétique modulaire en Mathématique Utilisé par exemple dans le RSA également très utilisé en informatique car Une très grande partie des Calculs réalisés en informatique sont des calculs d'arithmétique

Modulaire.

L'opération Modulo est tout simplement le reste d'une division euclidienne. Par exemple, En divisant 5 Par 2 l'on obtient 5= 2*2+1 ceux qui correspond à 5 = 1[9] .

4.1.3. Chiffrement

a)Première étape: Création des clés

Destinataire Construit un quadruplet de nombres **(p, q, e, d)**

- L'utilisateur choisit deux grands nombres **p** et **q** Premiers et les multiplie pour obtenir **n = p * q**
- Il choisit un nombre grand e et premier avec **(p–1) * (q - 1)**
- On calcul **d** inverse de **e** , tel que **(ed-1)** est un multiple de **(p-1)(q-1),** c'est-à-dire tel que **ed = l mod (p-l)(q-1)**. Celle-ci peut être résolue grâce à une version étendue de l'algorithme D'Euclide.
- Si **A** est un entier quelconque, alors : $\mathbf{A^{ed} = A (mod\ n)}$, et c'est cette identité qui va tout fair Fonctionner.

b) Deuxième étape : Publication la clef publique.

Destinataire rend publics **(n, e),** qui constituent la clef publique. Il la publie dans un annuaire ou La Communique à Emetteur, qui la lui demande. Il ne communique sur tout pas **p** , **q** ou **d** . Les nombres **p** et **q** peuvent être oubliés , car ils ne serviront plus à personne. Le nombre **(n, d)** constitue la clef Secrète de destinataire.

c) Troisième étape : Transmission d'information.

Emetteur, qui veut transmettre une information secrète à Destinataire, transforme son information En un nombre entier **A ,** inférieur à **n** (ou en plusieurs si nécessaire) , en utilisant des conventions Connues de tous comme par exemple le code **ASCII** Une personne voulant envoyer le message **A** au Propriétaire des paramètres **(n, e)** va décomposer **A** en blocs aide longueur strictement inférieure à La Taille (en nombre de chiffres) de **n**.

d) quatrième étape : Chiffrage de l'information.

Emetteur calcule, grâce à la méthode d'exponentiation rapide :

- Calculer pour tout $\mathbf{i}$: $\mathbf{b_i = a_i^e\ Mod\ n}$

- Former le message **B** en regroupant les blocs b_i
- **B** = A^e **(mod n),** envoie **B** à destinataire par un canal qui n'a pas besoin d'être protégé (par exemple, le courrier électronique) [10] .

4.1.4. Déchiffrement

Destinataire, pour décoder B, calcule B^d(mod n), ce qui lui redonne A, car , d'après le théorème du **RSA**, On a: $\mathbf{B^d = A^{ed} = A \ (mod\ n).}$

Exemple :

– On choisit : p = 5 et q = 11

– Ce qui implique :

✓ n= p x q = 5 x 11 = 55

✓ z = Φ (n) = (p-1) x (q-1) = 4 x 10 = 40

– On peut choisir e = 7 (7 est premier avec 40)

✓ et on calcule d tel que:

dx e= 1 mod z => dx 7= 1 mod 40 => d= 23

– On obtient donc:

✓ Clé publique: {7, 55}

✓ Clé privée : {23, 55}

Chiffrement						
Clair		Clé publique : 7	Chiffré			
		P^7	$P^7 \bmod 55$			
I	9	4782969	4	70368744177664	9	I
U	21	1801088541	21	25765808751082182919290758696 61	21	U
P	16	268435456	36	62367382520429325666908919788312 9856	16	P
M	13	62748517	7	27368747340080916343	13	M
I	9	4782969	4	70368744177664	9	I
A	1	1	1	1	1	A
G	7	823543	28	19259043800372760688541191131627 52	7	G
E	5	78125	25	14210854715202003717422485351562 5	5	E
					C^{23}	$C^{23} \bmod 55$
			Chiffré	Clé privé : 23	Déchiffré	
			Déchiffrement			

Figure II.3: Exemple de chiffrement & déchiffrement par RSA

4.2. Algorithme IDEA

IDEA est un système de chiffrement par blocs de 64 bits, avec une clé de 128 bits, qui tourne sur 8 Rondes . Cet algorithme , utilisé par PGP , n'utilise que trois opérations simples :

- le XOR notée $\oplus$,
- L'addition modulo 2^{16} notée [+]
- La multiplication modulo $2^{16}+1$ notée $\otimes$.

Le principe est basé sur la difficulté d'inverser les nombres dans un corps intègre Abélien Z //pz /.

4.2.1. Chiffrement

1. Le message en clair est partagé en plusieurs blocs.

2. Chacun de ces blocs étant chiffré à l'aide d'une même clé K de longueur r qui permet par un procédé à préciser, de fabriquer dessous clés k1, k2, …, kd.

3. Cette clé, choisie au hasard par l'utilisateur, est divulguée aux personnes concernées, qui Doivent aussi S'en servir pour lire les données protégées.

4. Le bloc subit un certain nombre (égal à d) de rondes.

Pour commencer le texte clair est divisé en un nombre de blocs comportant chacun 64 bits. L'algorithme va être appliqué à chacun de ces blocs jusqu'à ce que le texte soit Entièrement transformé en un texte codé.

Le bloc de données de 64 bits est divisé en 4 sous blocs de 16 bits : X1, X2, X3 et X4. Ces 4 sous Blocs Deviennent les entrées de la première ronde de l'algorithme . Il y a 8 Rondes au total . A chaque Ronde , Les 4 sous blocs sont combinés par ou exclusif , Additionnés , multipliés entre eux et avec 6 Sous blocs De 16 bit dérivés de la clef. Entre chaque ronde, le deuxième et le troisième sous bloc Sont échangé

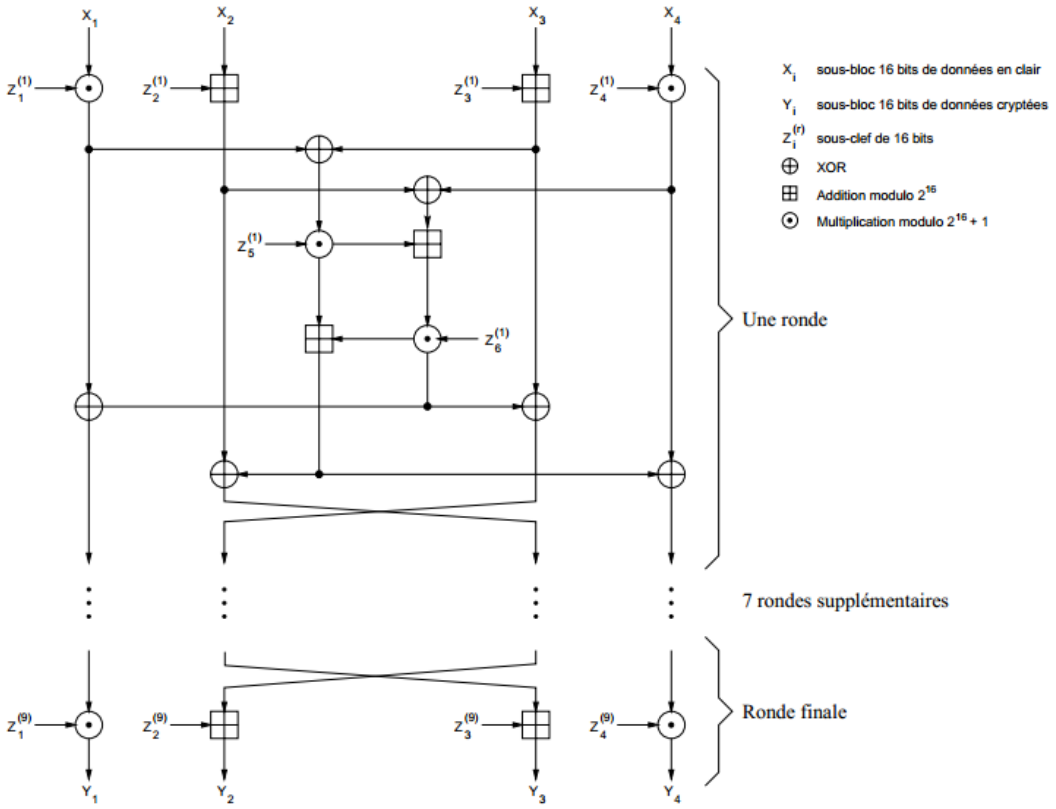


Figure II.4 : chiffrement par algorithme IDEA

Liste des 14 étapes de chaque ronde :

1) X1*Z1

2) X2+Z2

3) X3+Z3

4) X4*Z4

5) (Etape 1) XOR (Etape 3)

6) (Etape 2) XOR (Etape 4)

7) (Etape 5)*Z5

8) (Etape 6) + (Etape 7)

9) (Etape 8)*Z6

10) (Etape 7) + (Etape 9)

11) (Etape 1) XOR (Etape 9) →X1 de la ronde suivante

12) (Etape 3) XOR (Etape 9) →X3 de la ronde suivante

13) (Etape 2) XOR (Etape 10) →X2 de la ronde suivante

14) (Etape 4) XOR (Etape 10) →X4 de la ronde suivante

Il y aura en plus 4 étapes supplémentaires après la 8 me ronde :

1) X1*Z1 : **2)** X2+Z2

3) X3+Z3 : **4)** X4*Z4

Enfin, les 4 sous blocs ré assemblés pour former le texte chiffré [10] .

4.1.2. Génération des clés

IDEA crée 52 sous clefs à partir de la clef principale de l'utilisateur qui est de 128 bits (16 Octets) soit 6 Sous clefs pour chacune des 8 rondes (IE. chaque ronde l'algorithme utilise 6 sous clefs) et 4 autres Sous Clefs pour la transformation finale. Le processus de génération Des sous clefs est décrit dans les étapes suivantes :

1. D'abord, la clef secrète K de l'utilisateur qui est de 128 bits est divisée en 8 sous-clés De 16 bits. Ce Sont les 8 premières sous-clés pour l'algorithme (les 6 de la première Ronde et les 2 premières de la deuxième ronde).

2. Ensuite la clef secrète K est décalée circulairement de 25 bits vers la gauche.

3. Cette nouvelle clef de 128 bits est à nouveau divisée en 8 sous -clefs. Les 4 premières Sont utilisées Lors de la deuxième ronde et les 4 autres lors des autres lors de la troisième ronde.

4. Cette clef est à nouveau décalée circulairement de 25 bits vers la gauche pour être Diviser en 8 Sous -clefs suivantes de 16 bits, et ainsi de suite jusqu'à la fin de l'algorithme.

Le tableau suivant indique les sous-blocs de clef de chiffrement correspondant [10] .

Ronde	Sous blocs de clef de chiffrement
1 :	$Z_1^{(1)}$ $Z_2^{(1)}$ $Z_3^{(1)}$ $Z_4^{(1)}$ $Z_5^{(1)}$ $Z_6^{(1)}$
2 :	$Z_1^{(2)}$ $Z_2^{(2)}$ $Z_3^{(2)}$ $Z_4^{(2)}$ $Z_5^{(2)}$ $Z_6^{(2)}$
3 :	$Z_1^{(3)}$ $Z_2^{(3)}$ $Z_3^{(3)}$ $Z_4^{(3)}$ $Z_5^{(3)}$ $Z_6^{(3)}$
4 :	$Z_1^{(4)}$ $Z_2^{(4)}$ $Z_3^{(4)}$ $Z_4^{(4)}$ $Z_5^{(4)}$ $Z_6^{(4)}$
5 :	$Z_1^{(5)}$ $Z_2^{(5)}$ $Z_3^{(5)}$ $Z_4^{(5)}$ $Z_5^{(5)}$ $Z_6^{(5)}$
6 :	$Z1^{(6)}$ $Z_2^{(6)}$ $Z_3^{(6)}$ $Z_4^{(6)}$ $Z_5^{(6)}$ $Z_6^{(6)}$
7 :	$Z_1^{(7)}$ $Z_2^{(7)}$ $Z_3^{(7)}$ $Z_4^{(7)}$ $Z_5^{(7)}$ $Z_6^{(7)}$
8 :	$Z_1^{(8)}$ $Z_2^{(8)}$ $Z_3^{(8)}$ $Z_4^{(8)}$ $Z_5^{(8)}$ $Z_6^{(8)}$
Finale	$Z_1^{(9)}$ $Z_2^{(9)}$ $Z_3^{(9)}$ $Z_4^{(9)}$

Tableau II.1 : les sous-blocs de clef de chiffrement

4.2.2. Déchiffrement

Le déchiffrement est exactement le même que le chiffrement, accepté que les sous blocs de Clefs soient inversés et légèrement différents. La conséquence est que le même processeur, la même Carte ou Le même logiciel sert à chiffrer et à déchiffrer. La sécurité de ce système repose donc Entièrement sur la Clé personnelle de l'utilisateur. Le tableau suivant indique les sous blocs de clef de déchiffrement correspondant.

Ronde	Sous blocs de clef de déchiffrement
1 :	$Z_1^{(9)-1}$ $-Z_2^{(9)}$ $-Z_3^{(9)}$ $Z_4^{(9)-1}$ $Z_5^{(8)}$ $Z_6^{(8)}$
2 :	$Z_1^{(8)-1}$ $-Z_3^{(8)}$ $-Z_2^{(8)}$ $Z_4^{(8)-1}$ $Z_5^{(7)}$ $Z_6^{(7)}$
3 :	$Z_1^{(7)-1}$ $-Z_3^{(7)}$ $-Z_2^{(7)}$ $Z_4^{(7)-1}$ $Z_5^{(6)}$ $Z_6^{(6)}$
4 :	$Z_1^{(6)-1}$ $-Z_3^{(6)}$ $-Z_2^{(6)}$ $Z_4^{(6)-1}$ $Z_5^{(5)}$ $Z_6^{(5)}$
5 :	$Z_1^{(5)-1}$ $-Z_3^{(5)}$ $-Z_2^{(5)}$ $Z_4^{(5)-1}$ $Z_5^{(4)}$ $Z_6^{(4)}$
6 :	$Z_1^{(4)-1}$ $-Z_3^{(4)}$ $-Z_2^{(4)}$ $Z_4^{(4)-1}$ $Z_5^{(3)}$ $Z_6^{(3)}$
7 :	$Z_1^{(3)-1}$ $-Z_3^{(3)}$ $-Z_2^{(3)}$ $Z_4^{(3)-1}$ $Z_5^{(2)}$ $Z_6^{(2)}$
8 :	$Z_1^{(2)-1}$ $-Z_3^{(2)}$ $-Z_2^{(2)}$ $Z_4^{(2)-1}$ $Z_5^{(1)}$ $Z_6^{(1)}$
Finale	$Z_1^{(1)-1}$ $-Z_2^{(1)}$ $-Z_3^{(1)}$ $Z_4^{(1)-1}$

Tableau II.2 : les sous blocs de clef de déchiffrement

Les sous blocs de clef de chiffrement sont inverses par rapport à l'addition ou par Rapport à La multiplication des sous blocs de clefs de chiffrement. (Pour les besoins D'IDEA, L'inverse de la Multiplication de 0 est 0.) [10] .

5. autres fonctionnalité de PGP

5.1. La signature des données

En matière de signature des données, PGP utilise un scellement de données. Il applique une Fonction de hachage au texte en clair à signer. Puis le condensé obtenu, de taille fixe, est signé avec la clef Privée de l'expéditeur. Le sceau ainsi obtenu est joint au texte en clair. A la réception du message, le destinataire (i) applique la fonction de hachage au texte en clair, (ii) utilise la clef publique de l'expéditeur pour retrouver la valeur du condensé joint au texte en clair et (iii) compare les deux condensés. Il s'agit d'un système de scellement des plus classiques, seule la fonction de hachage utilisée est propre à PGP [9] .

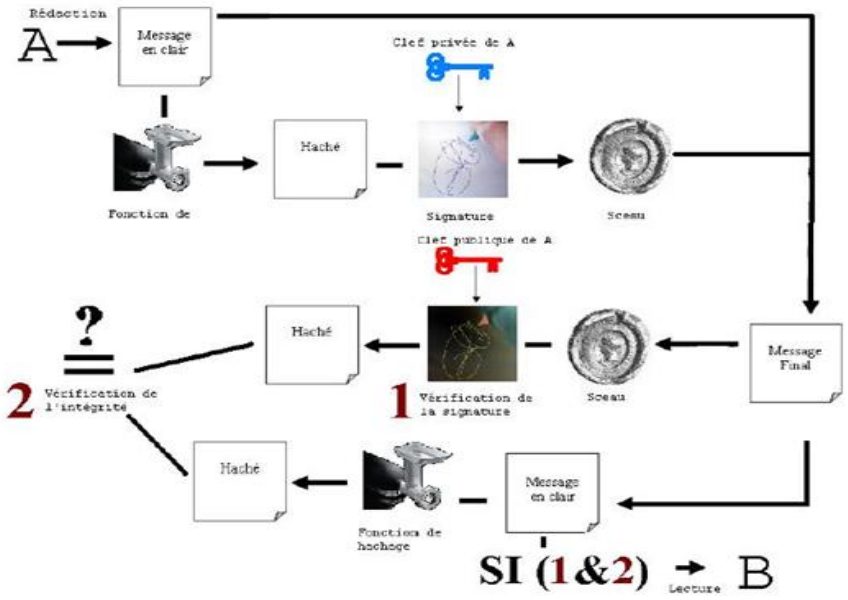


Figure II.5 : la signature de donne dans le système PGP

5.2. Les certificats

Rappelons que l'un des points-clefs des crypto-systèmes est la vigilance que L'utilisateur apporte À la vérification de l'appartenance de la clef au bon propriétaire. Ces systèmes sont en effet sensibles aux attaques dites de « l'homme au milieu » qui Consiste pour L'attaquant à créer une clef qu'il fait passer pour la clef d'une autre personne, s'accordant ainsi la possibilité de lire tous les messages sensément destinés à La personne dont il a usurpé l'identité. Pour Éviter cela, on crée des certificats numériques dont le but est d'apporter la preuve de la validité d'une clef et de son appartenance à un Propriétaire donné. Ces certificats peuvent être considérés comme les cartes d'identité Des clefs et se composent de trois

Parties **:**

- la clef publique pour laquelle le certificat est généré
- des informations sur le détenteur de la clef (nom d'utilisateur, mail, etc.) et sur le certificat lui-même (durée de validité…)
- d'une ou plusieurs signatures de personnes attestant de la validité de ces informations et de la clef.

En règle générale, une autorité appelée « autorité de certification » est seule habilitée à Produire des Certificats et les signer à l'aide de sa clef privée. Dans PGP, le système Retenu pour les certifications Numériques ne se base pas sur une autorité de certification Centralisée, mais sur un système de confiance. Il est possible à chaque personne de signer De sa clef privée un certificat. Contrairement à un système Centralisé, un certificat PGP Pourra contenir une multitude de signatures numériques.

Le format d'un certificat PGP comprend entre autres les informations suivantes :

- le numéro de la version de PGP utilisée pour créer la clef associée à ce certificat
- la clef publique sur laquelle porte ce certificat ainsi que l'algorithme employé pour la générer
- des informations sur le propriétaire de cette clef (nom, mail, nom d'utilisateur ; etc.)
- la signature numérique du propriétaire effectuée à l'aide de la clef privée qui correspond à la clef publique du certificat
- la période de période de validité du certificat
- les éventuelles signatures effectuées par d'autres utilisateurs.

PGP reconnaît également les certificats X. 509. Il est possible de produire ses propres Certificats PGP. Pour les certificats X. 509, il faut effectuer la demande auprès d'une autorité De certification. La structure Des certificats est normalisée par le standard X.509v3 de L'hyperlink « http://www.itu.int » UIT **(** Union Internationale des Télécommunications**)** [9].

5.3. Les niveaux de confiance

Avec le système des certificats à signatures multiples se pose le problème de Déterminer si les Signatures apportées aux certificats sont dignes de confiance. Pour Pallier cet inconvénient, un système Basé sur les niveaux de confiance et de validité est Mis en place.

De base, il existe dans PGP trois niveaux de validité :

- valide.
- semi valide.
- invalide.

Ainsi que trois niveaux de confiance :

- confiance totale.
- confiance moyenne.

- aucune confiance.

Lorsque l'utilisateur génère sa paire de clefs dans PGP, celle-ci se voit implicitement attribuer Les niveaux maximums pour ces deux domaines.

Considérons maintenant que l'utilisateur importe la clef de X dans son système. Il valide La clef de X en signant son certificat et lui accorde une confiance maximum.

La clef de X a désormais valeur d'autorité de certification dans le système, ainsi lorsqu'une Clef signée par X est importée, elle sera considérée comme valide dans le système. Si, Désormais, une confiance moyenne est accordée à X et Y et que l'utilisateur importe une Clef, deux cas

De figure se présentent :

- si la clef est seulement signée par X ou Y, elle ne sera pas valide dans le système.
- si la clef est signée par X et Y, elle sera valide.

En résumé, pour qu'une clef soit validée dans le système PGP de l'utilisateur, elle devra avoir reçu soit :

- la signature de ce dernier.
- la signature d'une clef à laquelle une confiance totale aura été accordée.
- la signature d'au moins deux clefs auxquelles une confiance moyenne aura été apportée.

Les signatures effectuées par des clefs qui ont le niveau de confiance « Aucune Confiance » ne sont tout simplement pas prises en compte [9] .

5.4. Les empreintes

Les niveaux de confiance permettent d'automatiser la validation de clefs grâce à des Clefs qui sont déjà valide. Reste qu'en haut de cette pyramide, il faut commencer par valider Des clefs. C'est pourquoi il est nécessaire de pouvoir s'assurer de l'intégrité de celles-ci. C'est Le rôle des empreintes. Une empreinte est un condensé obtenu en appliquant une fonction de hachage à un certificat. Bien entendu, comme tout condensé, il est unique.

Dans PGP, les empreintes de certificats peuvent être affichées sous forme hexadécimale ou Sous la forme d'une série mots biométriques (« biome tric Word »). Les empreintes sont utilisées pour s'assurer auprès du détenteur de la clef que l'on veut valider L'authenticité du certificat qui la contient, en comparant cette empreinte à son homologue dans le Système du propriétaire de la clef.

5.5. La révocation

Un certificat possède une durée de validité. À l'issue de celle-ci, la clef de ce certificat N'est plus Considérée comme valide. Il peut toutefois être nécessaire d'invalider une clef Avant la fin de son Certificat :

Si celle-ci a été cassée ou le mot de passe lié à cette clef a été perdu. Dans X509 , révoquer sa Signature sur un certificat consiste à enlever sa signature d'un certificat d'authenticité, C'est-à-dire à indiquer qu'on n'apporte plus son crédit à ce certificat. La révocation PGP va Plus loin en apportant la possibilité d'invalider entièrement un certificat (et Plus seulement D'en enlever sa signature).

Toutefois, une telle révocation n'est possible que pour :

- le propriétaire de ce certificat.
- une personne considérée par ce propriétaire comme une autorité de certification, c'est-à-dire une Personne à laquelle il a attribué un niveau de confiance totale [9].

6. Conclusion

Après l'analyse de fonctionnement de PGP, on peut dire qu'il est rapide, simple D'accès, sûr et inattaquable (du moins inattaquée jusqu'à présent). De plus certains aspects De PGP sont très intéressants, comme par exemple sa gestion des clés distribuées. En effet, Les utilisateurs peuvent engendrer et distribuer leurs propres clés publiques. Ils Peuvent aussi Signer les clés des autres, ajoutant ainsi un niveau de confiance dans le Système.

Chapitre III Implémentation de système PGP

1. Introduction

Dans le but de bien réaliser notre application qui intègre tous les algorithmes déjà Vus dans le chapitre précédent, mais aussi afin de bien les implémenter on avait besoin d'utiliser Un langage à la fois simple à manipuler, qu'il soit un langage orienté objet, qu'il puisse Effectuer toutes les taches d'un langage de haut niveau (bureautique, graphique, multimédia, base de donné, environnement de développement, etc.). Notre choix porte sur le Langage java.

2. Pour quoi java ?

Nous avons choisi Java comme langage de programmation pour réaliser notre système Pour les raisons suivantes:

- Maitriser un nouveau langage de programmation et comprendre bien le concept orienté objet.
- Et les exceptions qui distingue ce langage par rapport les autres langages comme

La facilité (leur syntaxe est une extension de langage C)…etc

3. présentation général de java

Le langage Java sont issus d'un projet de sun Microsystems datant de 1990. Généralement, on attribut sa paternité à trois de ses ingénieurs : James Gosling, Patrick Naughton et Mike Sheridan. Il a été créé avec l'objectif de pouvoir exécuter les Programmes sans recompilation sur n'importe quelle machine et il est devenu aujourd'hui l'un des langages de programmation les plus utilisés. Il est incontournable Dans plusieurs domaines : Systèmes dynamiques (chargement dynamique de classes), Internet (les Applets Java), les systèmes Communicants (RMI, Corba, EJB) [11] .

3.1 Le langage java

Le langage Java est un langage de programmation informatique orienté objet. On définit un objet comme un modèle de programmation, il a un état et un comportement. Dans l'implémentation d'un objet, son état est défini par ses variables d'instance,

elles sont propres à l'objet. Le comportement d'un objet est défini par ses méthodes. La classe est une structure qui définit les variables d'instance et les méthodes d'un objet.

Java est un langage hybride, à la fois compilé et interprété. On dit qu'il est semi-compilé. Pour simplifier, disons qu'un programme Java est compilé dans un langage qui devra ensuite être Interprété. Le résultat de la compilation n'est pas du langage machine directement exécutable (propre au processeur), mais un code intermédiaire appelé byte code. Le byte code est intermédiaire entre le code source et le langage machine. Pour exécuter le programme, le byte code est interprété par un interpréteur appelé machine virtuelle Java (JVM). Toutes les machines actuelles possèdent une JVM. Ainsi, le byte code D'un programme peut être exécuté sur n'importe quel ordinateur (possédant une JVM) Alors qu'un programme compilé en langage machine n'est exécutable que sur un seul Type de processeur. C'est pour cela que le langage Java est un langage portable.

La syntaxe de Java se fonde très fortement sur le langage C++, dont il est par ailleurs Un sous-ensemble par certains aspects, alors que d'autres caractéristiques du langage Sont réellement originales. C++ est un langage à objets, Java ne reprend pas tous les aspects

De ce langage, certains choix ont été faits. Voici quelques exemples :

- La notion de pointeur (dans C++) n'existe pas en Java, il n'y a que la notion de référence.
- Il n'y a plus de destructions explicites des objets, tout est géré par un ramasse-miettes (Garbage collector). Ce ramasse-miettes détruit les objets qui ne sont plus référencés.
- Une nouvelle notion apparaît dans Java :

La notion de package. Un package permet de Regrouper plusieurs définitions de classes ou d'interfaces[11] .

3.2. La plate-forme java

La plate-forme Java est une plate-forme produite par Sun Microsystems permettant de développer et d'exécuter des programmes écrits en langage Java indépendants de tout Processeur et de tout système d'exploitation. Elle est constituée de plusieurs programmes, Chacun d'entre eux apportant une fonctionnalité de l'ensemble de ces capacités.

3.3. Pourquoi NetBeans ?

On a distingué l'environnement de développement NetBeans parce que il est faciles à Comprendre et à utiliser, très riche, et disponible gratuitement, pour compiler et exécuter Les programmes Java, les environnements de développement intégrés peuvent nécessiter Des ressources importantes et être lourds à utiliser pour de petits programmes. Comme Solution intermédiaire, vous disposez des éditeurs de texte appelant le compilateur Java Et exécutant les programmes Java. Lorsque vous aurez maîtrisé les techniques présentées.

4. Présentation général de NetBeans

NetBeans est un environnement de développement intégré, ou IDE, pour la création de Programmes d'ordinateur dans un certain nombre de langues différentes. Développement NetBeans Se réfère au processus de l'utilisation de NetBeans pour créer, Editer et organiser Votre code que Vous développez un programme informatique. NetBeans fut développé à l'origine par une équipe D'étudiants à prague, racheté ensuite par Sun Microsystems. quelque part en 2002, développement NetBeans peut également se référer à l'utilisation de La plate-forme NetBeans comme un cadre logiciel pour créer de nouvelles applications. NetBeans soutient le développement en Java , PHP , HTML ,JavaScript , CSS , Groovy et C + +. Dans ce contexte, le développement se réfère à un codage à l'appui, le débogage Et la compilation Du code dans ces langues. Dans la phase de codage, NetBeans vérifie Votre code en temps réel pour assurer syntaxe correcte. NetBeans comprend un certain nombre d'outils de débogage pour Vous aider à isoler et corriger les bogues non liées À la syntaxe. NetBeans peut aussi compiler votre code dans un programme de travail [11].

5. Description de l'application

5.1 Structure de l'application

Notre application est regroupe en deux groupe

5.1.1 Les classe

- ✓ IDEAtxt: cette classe est une méthode pour crypter et décrypter un texte.
- ✓ PGPcrypto: cette classe est une méthode pour crypter et décrypter par le système PGP.
- ✓ RSAtxt : cette classe est une méthode pour crypter et décrypter un texte par RSA.

5.1.2 Les interfaces graphiques

Notre logiciel possède une interface graphique qui facilite son utilisation, représentée par la figure suivant :

Figure III.1 Interface graphique principale

5.2 L'interface graphique de l'application

5.2.1 Barre de Menu

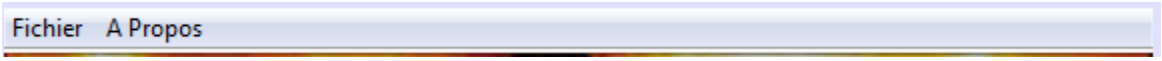
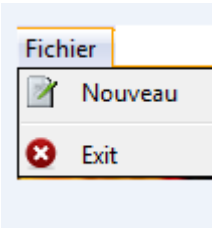


Figure III.2 Barre de Menu

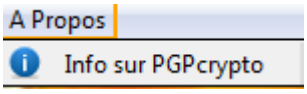
Cette bar est contient deux menu comme suivant :



a) Menu Fichier

1- **Nouveau :** ouvrir nouvelle une zone de texte

2- **Exit :** fermer le logiciel.



b) Menu A propos

3- Info sur PGPcrypto : information sur l'application.

5.2.2 Buttons raccourcis

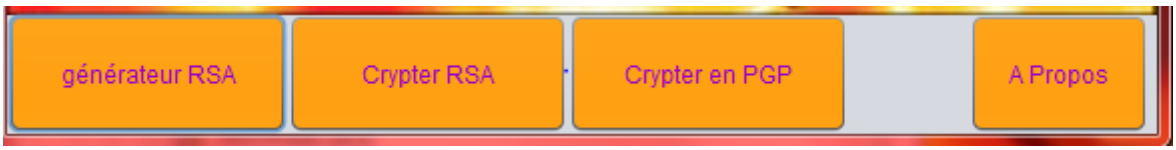


Figure III.3 Buttons raccourcis

1.Button générateur RSA: pour générer les clés publiques et clés priver) de RSA un

Nouvelle texte

2- Button crypter RSA: pour crypter et décrypter un texte par RSA.

3- Button cryptér PGP : pour crypter et décrypter un texte par system PGP

4-Button A propos: information sur l'application

5.3. Exemple d'application sur un texte RSA:

a) Le chiffrement :

1. généré la clé publique et clé prive :

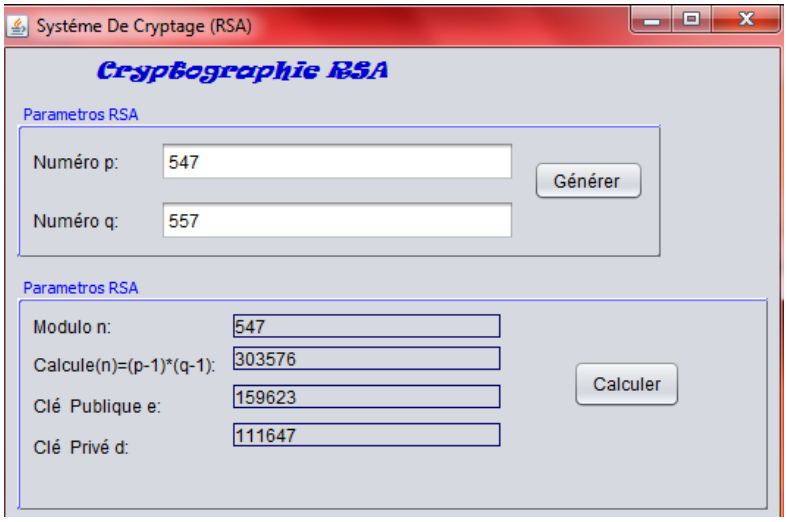


Figure III.4 générer les clés publiques et clés priver

2. Saisir le texte clair et cocher le bouton radio chiffrer et cliquer le bouton accepter

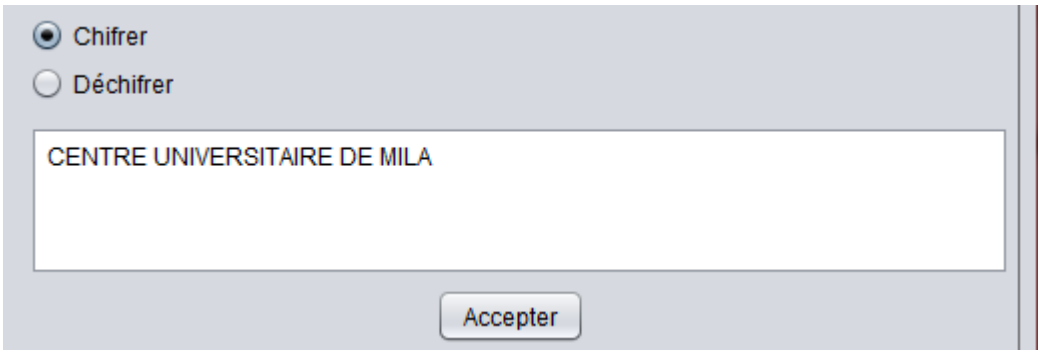


Figure III.5 Texte Clair

3. Le résultat de cryptage de texte est :

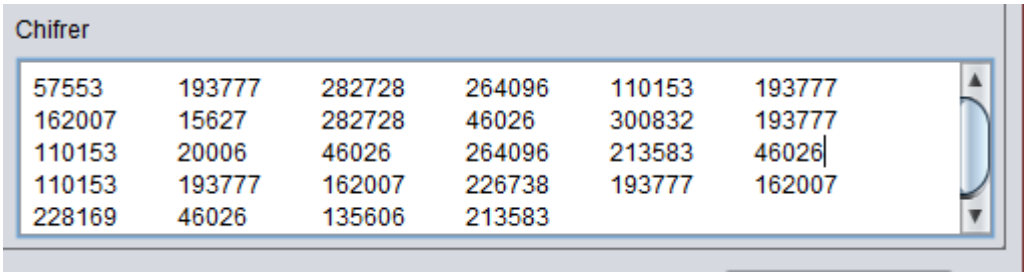


Figure III.6 Texte crypté

b) Le déchiffrement

1. Pour le décryptage nous suivons les mêmes étapes précédentes mais nous choisissons Le command décrypté au lieu de crypter.

5.4. Exemple d'application sur un texte par PGP :

1. Saisir le texte clair :

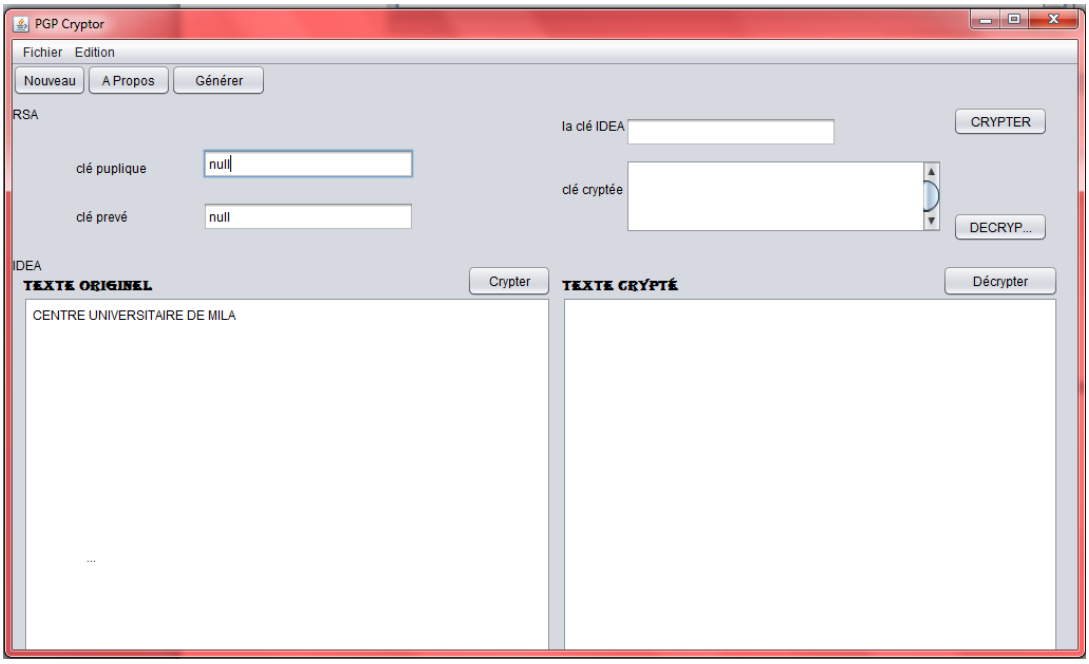


Figure III.7 Texte Clair

2. choisir le bouton crypter (Crypter : texte).

3. entrer la clé de cryptage.

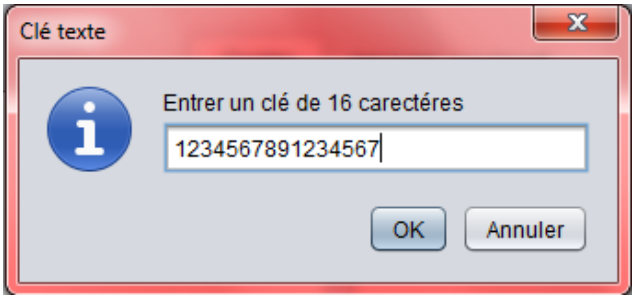


Figure III.8 entrer la clé

4. Le résultat de cryptage de texte est :

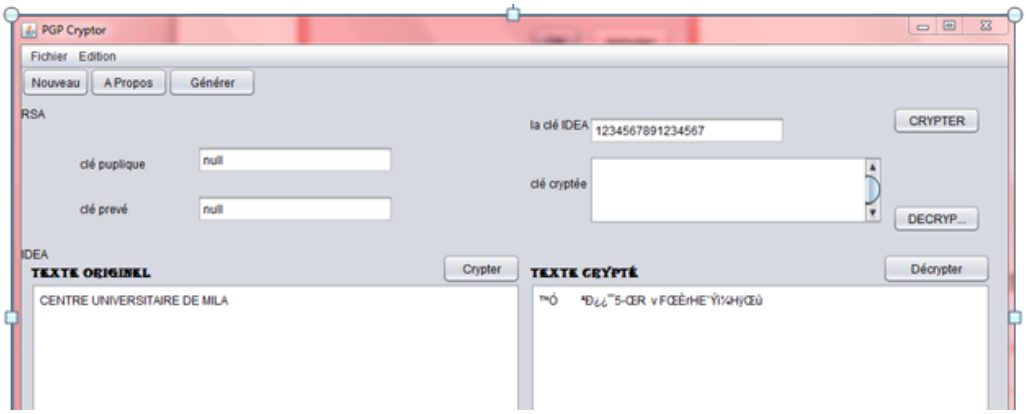


Figure III.9 Texte crypté

5 .Crypter la clé IDEA

6. entrer la clé publique et clé privé

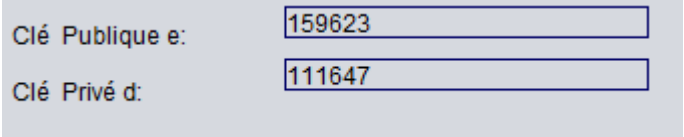


Figure III.10 entré la clé publique et clé priver

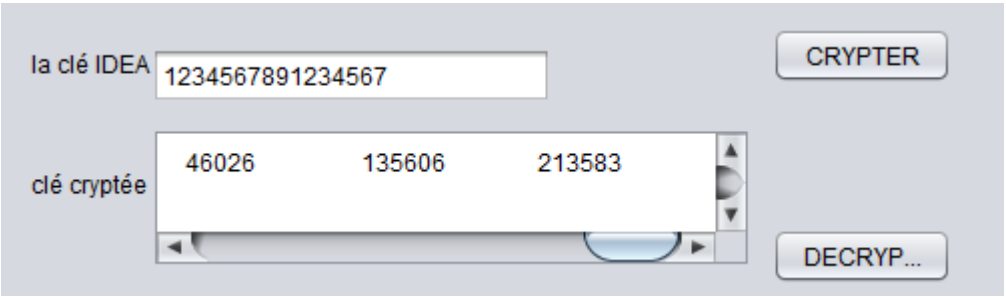


Figure III.11 clé crypté

1. Pour le décryptage nous avant décrypter la clé la clé IDEA par RSA

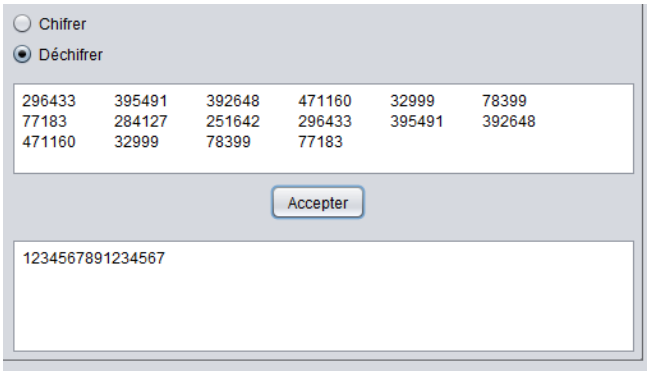


Figure III.12 décrypté la clé IDEA par RSA

3. décrypter le texte crypté par la c IDEA

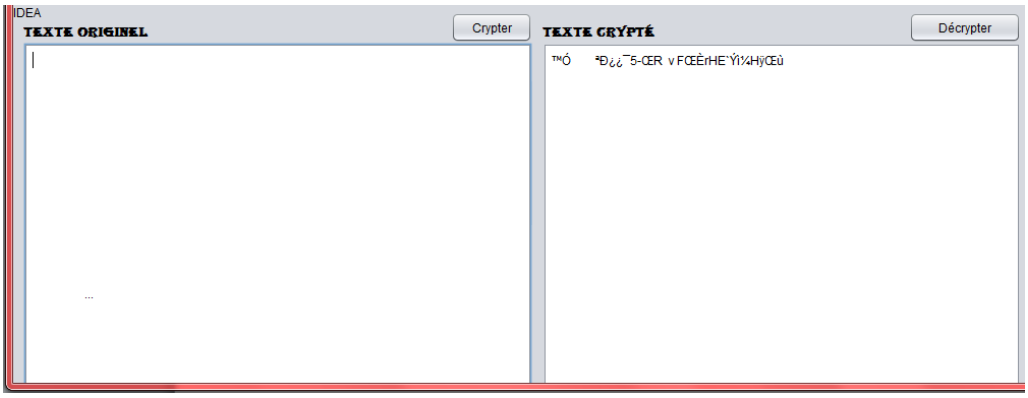


Figure III.13 entré le texte crypter

3. entrer la clé IDEA

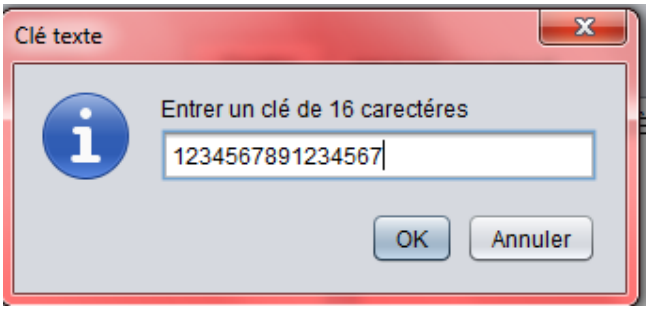


Figure III.14 entrer la clé

4. Le résultat de décryptage de texte est :

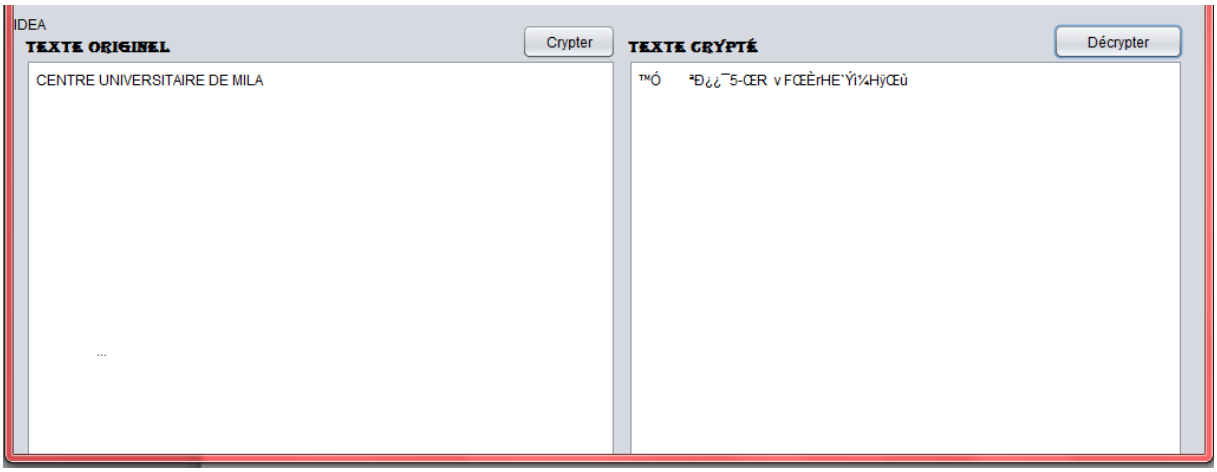


Figure III.15 : texte décrypté

6. Conclusion

Ce dernier chapitre était consacré pour la réalisation de système PGP. Nous avons commencé par la description du langage et l'environnement de développement choisi. Puis nous avons présenté l'interface graphique de notre application suivie par des exemples de cryptage et décryptage d'un texte et d'une image.

Conclusion générale

La cryptographie est une science très vaste, elle a un lien direct avec la sécurité des données Et la communication. De nos jours il y a déplus en plus d'informations qui doivent rester Secrètes ou confidentielles, par exemple les informations échanger par les banques, ou un mot de passe ne doivent pas divulgues et personne ne doit pouvoir les déduire. C'est Pourquoi ce genre d'information est crypté. Le cryptage est donc, nécessaire pour que Les données soient non Intelligibles sauf à l'auditoire voulu. Dans notre travail, nous avons élaboré une généralité Sur tous Les concepts en relation Avec la cryptographie à partir des notions fondamentales Jusqu'aux Mécanismes avancées, Suivi par un panorama sur les différentes classes des Algorithmes Cryptographiques existantes, Et un aperçu global sur la cryptanalyse. Enfin, nous Avons présenté en détail un standard de chiffrement qu'est Système PGP, suivi par une description des classes d'objets utilisées pour le cryptage et décryptage des textes en JAVA, Ainsi que l'interface graphique de l'application Et les résultats obtenus.

Bibliographie

[1] NGUYEN phong quang : Théorie et pratique de la cryptanalyse à clef publique, 2000.

[2] Quentin Stiévenart : La cryptologie, 1998.

[3] Adda ALI PACHA - Naima HADJ-SAID. La cryptographie et ses principaux systèmes de Références, 2002.

[4] Bourgeois Morgan. Initiation à PGP : Gnu PG, 2006.

[5] Hervé Schauer.Introduction à la cryptographie, 1999-2001.

[6] www.commentcamarche.net.

[7] www.wikipedia.org.

[8] Liran Lerman.Cryptanalyse par analyse de consommation, 2009-2010.

[9] www.developpez.com

[10] http://www.ifi.uio.no/~staalesc/pgp/home.html

[11] Wikipédia en français.