REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GÉNIE ÉLECTRIQUE
ET D'INFORMATIQUE

Mémoire

De fin d'études

En vue de l'obtention du diplôme de Master en Informatique

Option : Systèmes Informatiques

Thème

Cryptographie, cryptosystèmes et leurs applications
Cas : Machine ENIGMA

Dirigé par :
Mr CHAIEB Yazid

Réalisé par :
Mlle KAMEL Thinhinane
Mlle REMIDI Sonia

Jury composé de :
Président : . . . . . .
Examinateurs : . . . . . .
. . . . . .
. . . . . .

Promotion 2012/2013

Remerciements

Nous remercions en premier le Bon Dieu pour nous avoir donné le courage et la volonté pour réaliser ce travail et qui nous a éclairé les chemins par la lumière de son immense savoir.

Gratitude et sincères remerciements vont à notre promoteur Mr Y. CHAIEB de nous avoir encadré, pour son soutien tout au long de ce travail, son suivi et son orientation.

Notre profonde reconnaissance pour le temps précieux qu'il nous a consacré, ses remarques pertinentes et précieuses, ses conseils et ses encouragements.

Nous tenons aussi à exprimer toute notre gratitude envers les membres du jury qui nous font l'honneur de juger notre travail.

Nos remerciements vont également aux enseignants qui nous ont assistés durant tout le cursus universitaire.

Enfin, il nous est agréable d'exprimer nos remerciements à nos familles pour leur soutien et leur compréhension toute au long de notre cycle universitaire.

Dédicaces

A ma chère mère et à la mémoire de mon père malheureusement décédé, source d'affection et d'amour, qui n'ont pas cessé de m'encourager et de prier pour moi, qu'ils trouvent ici l'expression de ma gratitude qui, si grande qu'elle puisse être, ne sera jamais à la hauteur de leur dévouement. Aucune dédicace ne saura leur exprimer la profondeur de mes sentiments,

A mes adorables, sœur MELHA, beau-frère BRAHIM et leurs petites perles ILLY et NARA,

A tous les membres de ma famille tantes, oncles, en particulier NADIR et ZAHIA, et a tous mes cousins et cousines,

A ma binôme et chère amie SONIA et à toute sa famille,

A MASSILVA et KOUCI au près de qui j'ai trouvé beaucoup de soutien durant la réalisation de ce projet,

A LYNDA, SABER, ANIS et SLIMANE, et à tous ceux et celles dont les noms n'ont pas pu être cités…

Thinhinane

Dédicaces

Je dédie ce modeste travail

A ceux qui m'ont tout donné sans rien attendre en retour mis à part ma réussite, à ceux qui m'ont appris à aller au bout de mes ambitions, à ceux qui ont toujours cru en moi : à mes très chers parents, aucune autre personne ne m'est aussi trop chère que vous, je vous aime, que Dieu vous bénisse, vous assiste et vous soit en aide,

A la mémoire de mes grands-parents,

A mes adorables chères sœurs Djoummana et Lilia,

A mon très cher frère Boudjemaa,

A tous les membres de ma famille tantes, oncles, ainsi à tout mes cousins et cousines,

A ma binôme et chère amie Hinane et à mon adorable amie Massilva et leurs familles, vous qui m'avez accompagné tout au long de ce projet dans la joie et la "douleur". Merci pour les nombreux fous rires et tous les bons moments !

A Lynda, et tous mes ami(e)s et à toutes les personnes qui m'ont aidé de près ou de loin…

Sonia

Sommaire

Table des matières

Partie I

Partie II

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

La pérennité de toute Entreprise passe aujourd'hui par une disponibilité permanente de son système d'information. L'information nécessaire au bon fonctionnement de l'entreprise englobe aussi bien les données stratégiques que les données de tous les jours. Le système d'information doit être vu comme un ensemble qui inclut aussi bien l'information elle-même que les systèmes nécessaires pour la mettre en œuvre.

La continuité de l'activité de l'entreprise appelle à la continuité de son système d'information. Cette continuité ne peut être assurée que par la mise en place de moyens de protection permettant d'apporter un niveau de sécurité adapté aux enjeux spécifiques de l'entreprise. Ces derniers peuvent varier d'une entreprise à l'autre, mais la mise en place de la protection des systèmes d'information répond à des critères communs à adapter en conséquence.

Pour atteindre un niveau de protection satisfaisant, il convient de définir une politique de sécurité correspondant aux besoins. En effet, toute démarche de sécurité rigoureuse doit être inscrite dans une politique claire et documentée. Sa conception est donc une étape primordiale, qui consiste à identifier les objectifs de sécurité et à élaborer un ensemble de règles en fonction d'une analyse des risques. Ceci permet de minimiser le risque de dommages indésirables ou de pallier leurs effets et conduit à protéger les informations et les ressources identifiées comme sensibles

La cryptographie a donc pour but d'assurer la sécurité de ces ressources et la sécurité des communications et des données stockées en présence d'un adversaire. Elle propose un ensemble de techniques permettant d'offrir des services de confidentialité, d'authentification et d'intégrité. La cryptographie demeure la technique indispensable pour, d'une part, protéger la confidentialité des informations transmises sur les réseaux ou stockées dans les serveurs de données et pour, d'autre part, assurer l'intégrité d'un document ou pour prouver l'authenticité d'une opération ou d'une transaction. Elle applique des concepts mathématiques et met en place des paradigmes informatiques afin de résister aux attaques potentielles d'assaillants ou de prouver, de manière quasi sûre, qu'une procédure est incorruptible. La cryptographie participe à la sécurité informatique en proposant des primitives qui permettent d'atteindre les objectifs d'authentification, de confidentialité et de protection en intégrité.

Dans cette étude, nous avons essayé d'analyser l'environnement de la sécurité informatique, et nous avons cité des différentes solutions qui mèneront à augmenter la sécurisation des données. Aussi, nous avons étudié les techniques de cryptographie en les illustrant par quelques algorithmes de chiffrement, et nous avons proposé un logiciel adoptant un algorithme de cryptage symétrique, et portant sur différentes applications.

Chapitre I
Sécurité Informatique

I.1. Introduction

À l'heure du "tout disponible partout tout de suite", le transport des données en dehors du domicile d'un particulier ou d'une entreprise est une réalité qui mérite que l'on s'interroge sur la sécurité des transmissions pour ne pas compromettre un système d'information. Que ce soit à l'échelle d'une multinationale, d'une entreprise ou à plus petite échelle, la sécurité d'un système d'information prend plus ou moins d'importance selon la valeur que l'on confère à ces données.

Avec le développement d'Internet, chacun a accès au réseau où de plus en plus d'informations circulent. De plus en plus, les entreprises communiquent et diffusent via le media, que ce soit dans leurs liens avec leurs fournisseurs ou leurs partenaires ou en interne, dans les relations entre les employés eux mêmes. Nous sommes face non seulement à une augmentation de la quantité, mais aussi et surtout celle de l'importance des données.

L'ensemble formé par tout le réseau d'utilisateurs du système d'information se doit d'être connu pour être sûr. Les ressources qui y circulent doivent absolument être protégées en mettant en place une politique de sécurité efficace répondant aux besoins en sécurité des utilisateurs.

I.2. Environnement de la sécurité

I.2.1. Généralité sur les systèmes d'informations

Le système d'information définit l'ensemble des données et des ressources matérielles et logicielles de l'entreprise. Ce système permet de stocker et de faire circuler les ressources qu'il contient. Il représente également le réseau d'acteurs qui interviennent dans celui-ci, qui échangent les données, y accèdent et les utilisent. Ce système représente la valeur de l'entreprise, il est essentiel de le protéger. Le compromettre revient à compromettre l'entreprise. Il convient donc d'assurer sa sécurité en permanence, et surtout dans des conditions d'attaque, d'espionnage ou de défaillance. Il faut s'assurer que les ressources servent uniquement dans le cadre prévu, par les personnes accréditées et surtout pas dans un autre but.

Bien qu'elles n'en aient pas toujours conscience, les entreprises détiennent de nombreuses informations ayant une valeur économique et stratégique qui composent le capital immatériel de l'entreprise. Dans un contexte concurrentiel mondialisé, avec des relations commerciales souvent difficiles, ce capital immatériel permet à l'entreprise de perdurer, de se démarquer de la concurrence, de s'adapter aux besoins multiformes et évolutifs du marché.

Compte tenu de leur importance, ces informations sont exposées à de nombreuses menaces, parmi lesquelles figurent les risques de divulgation ou d'usages non autorisés provenant tant de l'intérieur de l'entreprise que de l'extérieur. Ces risques peuvent avoir de graves incidences sur la compétitivité de l'entreprise et peut, dans certains cas, aboutir à la paralysie de son activité voir à son anéantissement.

La sécurité engendre généralement le déploiement de moyens techniques, mais également et surtout, de solutions de prévention, qui doivent absolument prendre en compte la formation et la sensibilisation de tous les acteurs du système. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, dont la mise en œuvre se fait selon les quatre étapes suivantes :

- Identifier les besoins en terme de sécurité, les risques informatiques pesant sur l'entreprise et leurs éventuelles conséquences ;
- Elaborer des règles et des procédures à mettre en œuvre dans les différents services de l'organisation pour les risques identifiés ;
- Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés ;
- Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une menace ;

A cet égard, il ne revient pas aux seuls administrateurs informatiques de définir les droits d'accès des utilisateurs mais aux responsables hiérarchiques de ces derniers. Le rôle de l'administrateur informatique est donc de s'assurer que les ressources informatiques et les droits d'accès à celles-ci sont en cohérence avec la politique de sécurité définie par l'organisation.

Les mécanismes de sécurité mis en place peuvent néanmoins provoquer une gêne au niveau des utilisateurs et les consignes et règles deviennent de plus en plus compliquées au fur et à mesure que le réseau s'étend. Ainsi, la sécurité informatique doit être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils puissent utiliser le système d'information en toute confiance.

I.2.2. Les aspects des systèmes informatiques

Plusieurs facteurs engendrent l'insécurité des systèmes informatiques et parmi eux on peut citer :

I.2.2.1. Vulnérabilité [1]

La vulnérabilité est une faille dans les actifs[1], les contrôles techniques de sécurité ou les procédures d'exploitation ou d'administration utilisés dans un réseau. Elle consiste en une

[1] Peuvent représenter : les équipements, les matériels, les logiciels, les processus, etc.

faiblesse dans la protection du système, sous la forme d'une menace qui peut être exploitée pour intervenir sur l'ensemble du système, ou d'un intrus qui s'attaque aux actifs.

On peu distinguer deux principales familles de vulnérabilités :

1. **Vulnérabilité liées aux domaines physiques : [1]**

- Manque de ressources au niveau équipement.
- Accès aux salles informatiques non sécurisé.
- Absence ou mauvaise stratégie de sauvegarde de données.

2. **Vulnérabilité liés aux domaines technologiques :**

- Failles nombreuses dans les services et applicatifs Web et les bases de données.
- Pas de mises à jour des systèmes d'exploitation et des correctifs.
- Pas de contrôles suffisants sur les logiciels malveillants.
- Récurrence des failles et absence de supervision des événements.
- Réseaux complexes, non protégés.
- Mauvaise utilisation de la messagerie.

I.2.2.2. Les menaces [2]

Pour mettre en place une politique de sécurité, il faut d'abord commencer par identifier la menace, le risque potentiel. Il faut connaître son ennemi, ses motivations et prévoir la façon dont il procède pour s'en protéger et limiter les risques d'intrusion.

La menace existe en corrélation avec des vulnérabilités, il peut y avoir aussi plusieurs menaces pour chaque vulnérabilité. La connaissance des différents types de menaces peut aider dans la détermination de leurs dangerosités et des contrôles adaptés permettant de réduire leur impact potentiel.

Les principales menaces effectives auxquelles un système d'information peut être confronté sont :

- **Un utilisateur du système :** l'énorme majorité des problèmes liés à la sécurité d'un système d'information est l'utilisateur, généralement insouciant.
- **Une personne malveillante (Hackers et crackers) :** une personne parvient à s'introduire sur le système, légitimement ou non, et à accéder ensuite à des données ou à des programmes auxquels elle n'est pas censé avoir accès en utilisant par exemple des failles connues et non corrigées dans les logiciels.
- **Un programme malveillant :** un logiciel destiné à nuire ou à abuser des ressources du système est installé par mégarde ou par malveillance sur le système, ouvrant la porte à des intrusions ou modifiant les données. Des données personnelles peuvent être collectées à l'insu de l'utilisateur et être réutilisées à des fins malveillantes ou commerciales.

I.2.2.3. Les principales attaques

Les attaques se divisent, selon leurs types sur quatre catégories :

1. Les attaques par programmes malveillants :

Un logiciel malveillant (malware en anglais) est un logiciel développé dans le but de nuire un système informatique. Voici les principaux types de programmes malveillants :

- **Le virus :** programme se dupliquant sur d'autres ordinateurs.
- **Le ver** (**Worm** en anglais) **:** exploite les ressources d'un ordinateur afin d'assurer sa reproduction.
- **Le wabbit :** programme qui se réplique par lui-même (mais qui n'est ni un virus, ni un ver)
- **Le cheval de Troie** (**trojan** en anglais) **:** programme à apparence légitime qui exécute des routines nuisibles sans l'autorisation de l'utilisateur.
- **La porte dérobée** (**backdoor** en angalais) **:** ouvreur d'un accès à distance frauduleux sur un système informatique.
- **Le logiciel espion** (**spyware** en anglais) **:** collecteur d'informations personnelles sur l'ordinateur d'un utilisateur sans son autorisation, et en envoyant celles-ci à un organisme tiers.
- **Le keylogger** (**enregistreur de touches**) **:** programme généralement invisible installé sur le poste d'un utilisateur et chargé d'enregistrer à son insu ses frappes clavier.
- **L'exploit :** programme permettant d'exploiter une faille de sécurité d'un logiciel.
- **Le rootkit :** ensemble de logiciels permettant généralement d'obtenir les droits d'administrateur sur une machine, d'installer une porte dérobée, de truquer les informations susceptibles de révéler la compromission, et d'effacer les traces laissées par l'opération dans les journaux système.

2. Les attaques par messagerie :

En dehors des nombreux programmes malveillants qui se propagent par la messagerie électronique, il existe des attaques spécifiques à celle-ci :

- **Le Pourriel** (**spam** en anglais) **:** un courrier électronique non sollicité, la plupart du temps de la publicité. Ils encombrent le réseau, et font perdre du temps à leurs destinataires.
- **L'Hameçonnage** (**phishing** en anglais) **:** un courrier électronique dont l'expéditeur se fait généralement passer pour un organisme financier et demandant au destinataire de fournir des informations confidentielles.
- **La Canular informatique** (**hoax** en anglais) **:** un courrier électronique incitant généralement le destinataire à retransmettre le message à ses contacts sous divers prétextes. Ils encombrent le réseau, et font perdre du temps à leurs destinataires. Dans certains cas, ils incitent l'utilisateur à effectuer des

manipulations dangereuses sur son poste (suppression d'un fichier prétendument lié à un virus par exemple).

3. Les attaques sur le réseau :

- **Intrusion :** L'intrusion dans un système informatique a généralement pour but la réalisation d'une menace et est donc une attaque. Les conséquences peuvent être catastrophiques : vol, fraude, incident diplomatique, chantage…

  Le principal moyen pour prévenir les intrusions est le pare-feu (« firewall »). Il est efficace contre les fréquentes attaques de pirates amateurs, mais d'une efficacité toute relative contre des pirates expérimentés et bien informés. Une politique de gestion efficace des accès, des mots de passe et l'étude de fichiers « log » (traces) est complémentaire.

- **Ecoute du réseau** (**sniffing**) **:** il existe des logiciels qui permettent d'intercepter certaines informations qui transitent sur un réseau local, en retranscrivant les trames dans un format plus lisible (Network packet sniffing). C'est l'une des raisons qui font que la topologie en étoile autour d'un hub n'est pas la plus sécurisée, puisque les trames qui sont émises en « broadcast » sur le réseau local peuvent être interceptées.

  De plus, l'utilisateur n'a aucun moyen de savoir qu'un pirate a mis son réseau en écoute. L'utilisateur de switchs (commutateur) réduit les possibilités d'écoute mais en inondant le commutateur, celui-ci peut se mettre en mode « HUB » par sécurité.
- **Le déni de service** (**Denial of service**) **:** l'attaquant n'obtient pas un accès au système informatique sur le réseau mais il parvient à mettre en panne certains composants stratégiques (le serveur de messagerie, le site web, etc.). Le but d'une telle attaque n'est pas de dérober des informations sur une machine distante, mais de paralyser un service ou un réseau complet. Les utilisateurs ne peuvent plus alors accéder aux ressources. Les deux exemples principaux, sont le « Ping flood » ou l'envoi massif de courriers électroniques pour saturer une boîte aux lettres (mailbombing). La meilleure parade est le firewall ou la répartition des serveurs sur un réseau sécurisé.
- **IP Spoofing :** Usurpation d'adresse IP, on fait croire que la requête provient d'une machine autorisée. Une bonne configuration du routeur d'entrée permet d'éviter qu'une machine extérieure puisse se faire passer pour une machine interne.
- **DNS Spoofing :** Pousse un serveur DNS à accepter l'intrus. Pour l'éviter, il est intéressant de séparer le DNS du LAN de celui de l'espace public.

4. Les attaques sur les mots de passe :

Les attaques sur les mots de passe peuvent consister à faire de nombreux essais jusqu'à trouver le bon mot de passe. Dans ce cadre, notons les trois méthodes suivantes :

- **L'attaque par dictionnaire :** le mot testé est pris dans une liste prédéfinie contenant les mots de passe les plus courants et aussi des variantes de ceux-ci (à l'envers, avec un chiffre à la fin, etc.). Ces listes sont généralement dans toutes les langues les plus utilisées, contiennent des mots existants, ou des diminutifs (comme par exemple « powa » pour « power », ou « G0d » pour « god »).
- **L'attaque par force brute :** toutes les possibilités sont faites dans l'ordre jusqu'à trouver la bonne solution (par exemple de « aaaaaa » jusqu'à « ZZZZZZ » pour un mot de passe composé de six caractères alphabétiques).
- **L'attaque hybride :** C'est une combinaison d'attaque par force brute et d'attaque par dictionnaire qui permet au pirate de retrouver les mots de passe constitués d'un nom significatif suivi d'une lettre ou d'un chiffre (tel que « marechal6 »).

I.2.3. La sécurité des systèmes

I.2.3.1. Les critères de sécurité [3]

Pour protéger ses systèmes d'information, l'entreprise doit - en fonction d'une évaluation des risques - mettre en place une politique de sécurité informatique, des systèmes d'information, afin d'assurer les mesures de protection des données qui sont habituellement effectives dans au moins un des domaines suivants :

1. La confidentialité

Seules les personnes habilitées doivent avoir accès aux données. Toute interception ne doit pas être en mesure d'aboutir, les données doivent être cryptées, seuls les acteurs de la transaction possédant la clé de compréhension. Pour que ce service soit efficace, il doit fonctionner avec les services de responsabilité afin d'identifier correctement les personnes.

Grâce à cette fonction, le service de confidentialité protège des attaques d'accès. Il doit tenir compte du fait que l'information peut se trouver sous forme physique (dossiers, papier), sous forme électronique (fichiers) et en circulation sur le réseau.

2. L'intégrité

De manière générale, l'intégrité des données désigne l'état de données qui, lors de leur traitement, de leur conservation ou de leur transmission, ne subissent aucune altération ou destruction volontaire ou accidentelle, et conservent un format permettant leur utilisation. L'intégrité des données comprend quatre éléments : l'intégralité, la précision, l'exactitude/authenticité et la validité.

En fait, l'expression intégrité des données correspond à deux notions légèrement différentes, selon que le contexte est celui des télécommunications ou de la cryptographie.

Si le principe général est le même - les données ne doivent pas avoir été modifiées depuis leur création, à comprendre au sens large (écriture sur un support de stockage, transmission...) - la cryptographie veut pouvoir affirmer que les données ont ou n'ont pas été modifiées, ce qui se fait souvent via une fonction de hachage ou, mieux, un MAC (*Message authentication code*) qui ajoute l'usage d'une clé secrète, tandis qu'en télécommunication, on souhaite simplement pouvoir détecter et souvent corriger ces modifications.

Une autre façon de présenter cette différence est de dire que la cryptographie cherche à prouver qu'il n'y a pas eu de falsification, alors que les télécommunications cherchent à vérifier qu'il n'y a pas eu d'erreurs de " copie ".

3. La disponibilité [WEB 01]

Il faut s'assurer du bon fonctionnement du système, de l'accès à un service et aux ressources à n'importe quel moment. La disponibilité couvre aussi les systèmes de communication qui transmettent les informations entre sites ou entre systèmes, c'est à l'information et aux services électroniques que nous pensons le plus souvent. Cependant, la disponibilité des fichiers papiers peut aussi être assurée.

La disponibilité d'un équipement ou d'un système est une mesure de performance qu'on obtient en divisant la durée durant laquelle ledit équipement ou système est opérationnel par la durée totale durant laquelle on aurait souhaité qu'il le soit. On exprime classiquement ce ratio sous forme de pourcentage.

Il ne faut pas confondre la disponibilité avec la *« rapidité de réponse »*, que l'on appelle aussi « performance ».

La disponibilité est aussi à prendre de manière relative. Les systèmes n'ont pas la même importance suivant les moments, l'impact n'est pas le même suivant qu'on a absolument besoin du système à ce moment ou alors qu'on est dans une période de moins grand besoin.

4. L'authentification

L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...). L'authentification permet donc de valider l'authenticité de l'entité en question.

L'authentification, indissociable de la notion d'identification, est un élément essentiel de la sécurité informatique. Authentifier est en effet une condition indispensable :

- pour garantir la confidentialité et l'intégrité des données transmises,
- pour protéger contre les Hackers et autres utilisateurs malveillants.

L'authentification consiste donc à vérifier qu'une personne possède bien l'identité qu'elle affirme avoir.

Le service d'authentification limite l'accès aux personnes autorisées. Il faut s'assurer de l'identité d'un utilisateur avant l'échange de données. L'authentification garantit à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

I.2.3.2. Les étapes de la sécurité

De nombreuses techniques peuvent être mises en œuvre pour assurer la sécurité des informations. Il convient de choisir des étapes nécessaires, suffisantes, et justes, pour la bonne mise en œuvre de ces techniques, parmi elles on peut citer :

1. Conception d'une approche globale [4]

La sécurité d'un système informatique fait souvent l'objet de métaphores. En effet, on la compare régulièrement à une chaîne en expliquant que le niveau de sécurité d'un système est caractérisé par le niveau de sécurité du maillon le plus faible. Ainsi, une porte blindée est inutile dans un bâtiment si les fenêtres sont ouvertes sur la rue.

Cela signifie que la sécurité doit être abordée dans un contexte global et notamment prendre en compte les aspects suivants :

- La sensibilisation des utilisateurs aux problèmes de sécurité.
- La sécurité logique, c'est-à-dire la sécurité au niveau des données, notamment les données de l'entreprise, les applications ou encore les systèmes d'exploitation.
- La sécurité des télécommunications : technologies réseau, serveurs de l'entreprise, réseaux d'accès, etc.
- La sécurité physique, soit la sécurité au niveau des infrastructures matérielles : salles sécurisées, lieux ouverts au public, espaces communs de l'entreprise, postes de travail des personnels, etc.

2. Analyses des besoins

La phase de définition des besoins en termes de sécurité est une étape indispensable pour la mise en œuvre d'une politique de sécurité.

L'objectif consiste à déterminer les besoins de l'organisation en faisant un véritable état des lieux du système d'information, puis d'étudier les différents risques et la menace qu'ils représentent afin de mettre en œuvre une politique de sécurité adaptée

La phase de définition comporte ainsi deux étapes :

- L'identification des besoins.
- L'analyse des risques.

i. Identification des besoins

La phase d'identification des besoins consiste dans un premier temps à faire l'inventaire du système d'information, notamment pour les éléments suivants :

- Personnes et fonctions (cas d'entreprise).
- Matériels, serveurs et les services qu'ils délivrent.
- Cartographie du réseau (plan d'adressage, topologie physique, topologie logique, etc.).
- Liste des noms de domaine de l'entreprise.
- Infrastructure de communication (routeurs, commutateurs, etc.).
- Données sensibles.

ii. L'analyse des risques

L'étape d'analyse des risques consiste à répertorier les différents risques encourus, d'estimer leur probabilité et enfin d'étudier leur impact.

La meilleure approche pour analyser l'impact d'une menace consiste à estimer le coût des dommages qu'elle causerait (par exemple attaque sur un serveur ou détérioration de données vitales pour l'entreprise).

Sur cette base, il peut être intéressant de dresser un tableau des risques et de leur potentialité, c'est-à-dire leur probabilité de se produire, en leur affectant des niveaux échelonné selon un barème à définir, par exemple :

- Sans objet (ou improbable) : la menace n'a pas lieu d'être ;
- Faible : la menace a peu de chance de se produire ;
- Moyenne : la menace est réelle ;
- Haute : la menace a de grandes chances de se produire.

3. Politique de sécurité [4]

La sécurité des systèmes d'information se charge généralement de garantir les droits d'accès aux données et ressources d'un système, en mettant en place les mécanismes d'authentification et de contrôle. Ces mécanismes permettent d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leurs ont été octroyés.

La sécurité informatique doit toutefois être étudiée de telle manière à ne pas empêcher les utilisateurs de développer les usages qui leur sont nécessaires, et de faire en sorte qu'ils

puissent utiliser le système d'information en toute confiance. C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, c'est-à-dire :

- Elaborer des règles et des procédures, installer des outils techniques dans les différents services de l'organisation (autour de l'informatique).
- Définir les actions à entreprendre et les personnes à contacter en cas de détection d'une intrusion.
- Sensibiliser les utilisateurs aux problèmes liés à la sécurité des systèmes d'informations.
- Préciser les rôles et responsabilités.

La politique de sécurité est donc l'ensemble des orientations suivies par une entité en termes de sécurité. A ce titre, elle se doit d'être élaborée au niveau de la direction de l'organisation concernée, car elle concerne tous les utilisateurs du système.

I.2.3.3. Les techniques de sécurisation des données

Sécuriser des données, consiste à les rendre imperméables aux intrus qui veulent s'emparer des informations que vous jugez secrètes ou bien confidentielles. Pour cela, différents techniques de sécurisation ont été misent en œuvre pour atteindre ce but.

Deux types de sécurité englobent ces techniques :

1. La sécurité physique [5]

Par sécurité physique on désigne les mesures destinées à empêcher l'accès physique non autorisé aux ressources du réseau, à ses équipements, installations, matériels et documents, et visant à les protéger contre les dommages, le vol et la modification.

La sécurité n'est pas associée à un service de sécurité particulier. En fait, les services de sécurité physique regroupent tous les services de sécurité susmentionnés, qu'il s'agisse de la confidentialité, de l'intégrité, de l'imputabilité et, surtout, de la disponibilité.

Pour contrôler l'accès aux ressources du réseau, on peut recourir à divers moyens de sécurité physique, notamment des verrous, des gardes, des laissez-passer, des alarmes et d'autres dispositifs similaires.

2. La sécurité logicielle

La sécurité logicielle est considérée comme indispensable pour assurer la confidentialité des données. Les logiciels, on le sait, souffrent de défauts de conception, ou bugs ce qui les rend facilement exploitables à des fins malveillantes. Et ces défauts deviennent des failles de sécurité quand ils permettent à de petits programmes tiers (des « exploits ») de s'immiscer dans le logiciel pour en modifier le comportement et récupérer des informations confidentielles, voire d'ouvrir une porte d'accès dérobée pour se propager sur le réseau.

Une faille, qui affecte un logiciel se connectant au réseau, le navigateur Internet par exemple, peut être exploitée à distance. Avec le développement d'Internet, ce type de failles

de sécurité est devenu le plus courant. Les failles de sécurité sont corrigées par les mises à jour des logiciels de l'ordinateur. Mais plus le temps se prolonge entre le moment de la découverte d'une faille par des personnes malintentionnées (tels les hackers) et celui de l'installation du correctif proposé par l'éditeur du logiciel incriminé, plus le risque de contamination par un logiciel malveillant est grand.

I.3. Conclusion

Le système d'information d'une entreprise peut être vital à son fonctionnement. Il est donc nécessaire d'assurer sa protection, afin de lutter contre les menaces qui pèsent sur l'intégrité, la confidentialité et la disponibilité des ressources.

La malveillance informatique est souvent à l'origine de ces menaces, qu'il s'agisse de vol d'information ou de sabotage, n'importe qui pouvant s'improviser pirate informatique avec des outils adaptés.

Beaucoup de compétences sont nécessaires pour assurer une sécurité optimale, mais il est impossible de garantir la sécurité de l'information à 100%. Malgré tout, il existe des moyens efficaces pour faire face à ces agressions.

C'est pour cela qu'il est utile de bien savoir gérer les ressources disponibles et comprendre les risques liés à la sécurité informatique, pour pouvoir construire une politique de sécurité adaptée aux besoins de la structure à protéger. La mise en place d'un dispositif de sécurité efficace ne doit cependant jamais dispenser d'une veille régulière au bon fonctionnement du système.

Chapitre II
La Cryptographie

II.1. Introduction à la cryptographie

Les cryptographes n'ont cessé de redoubler d'ingéniosité, faisant se succéder des dizaines de systèmes de chiffrement plus recherchés les uns que les autres. Se livrant bataille pour la gloire ou l'argent, ils n'ont cessé de faire évoluer cette science qu'est la cryptographie. Avec d'abord une mécanisation (notamment la machine Enigma), puis grâce à l'avènement des ordinateurs, et avec eux une puissance de calcul surpassant de loin le niveau humain, la cryptographie a su trouver son chemin dans les dédales du progrès.

Dans ce chapitre, nous nous tournerons dans un premier temps vers les techniques cryptographiques qui ont marqué l'histoire, suivis par les techniques actuelles du monde de l'informatique. Par ailleurs, nous nous arrêterons également sur les notions de signatures, clé, certificats et fonctions de hachage.

II.2. Définition et terminologie

II.2.1. Définition [6]

La cryptologie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse.

Étymologiquement, le mot cryptographie est composé du grec κρυπτο *caché* et de γραφν *lire*, ce qui nous amène à concevoir la cryptographie comme l'art de communiquer en langage secret.

La cryptographie traditionnelle est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce qu'on appelle le chiffrement, qui, à partir d'un texte en clair, donne un texte chiffré ou cryptogramme. Inversement, le déchiffrement est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré. Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées algorithmes cryptographiques, qui dépendent d'un paramètre appelé clé.

La cryptanalyse, à l'inverse, est l'étude des procédés cryptographiques dans le but de trouver des faiblesses et, en particulier, de pouvoir décrypter des textes chiffrés. Le décryptement est l'action consistant à retrouver le texte en claire sans connaitre la clé de déchiffrement.

II.2.2. Terminologie

Les intervenants des opérations de cryptologie sont toujours désignés par les mêmes prénoms. Bruce Shneier a recensé les plus utilisés [7]. Alice et Bob sont les opérateurs de bonne foi du système cryptologique, et Ève en est l'attaquante. Ainsi, le plus souvent Alice

souhaite envoyer un message à Bob sans qu'un tiers, Ève, ne puisse apprendre une quelconque information.

II.3. Historique

II.3.1. Première utilisation de communication sécurisée

La sécurisation des transmissions d'information a été très tôt reconnue comme très importante. Le plus ancien usage de cet art nous est rapporté par Hérodote [8] qui relate entre autres les différends entre la Perse et la Grèce au V^{e} siècle av J-C. Démarate, roi de Sparte exilé en Perse a réussi à prévenir la Grèce, au moyen d'un ingénieux procédé, de l'imminence d'une invasion perse et a permis ainsi d'éviter une catastrophe pour les grecs.

II.3.2. Chiffre de César

Le chiffre de César est la méthode de cryptographie la plus ancienne communément admise par l'histoire. Il consiste en une substitution mono-alphabétique: chaque lettre est remplacée ("substitution") par une *seule* autre ("mono-alphabétique"), selon un certain décalage dans l'alphabet ou de façon arbitraire. D'après Suétone, César avait coutume d'utiliser un décalage de 3 lettres : A devient D, B devient E, C devient F, etc. Il écrivait donc son message normalement, puis remplaçait chaque lettre par celle qui lui correspondait :

CLAIR	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-> décalage = 3																										
CODE	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Exemple : d'après cette méthode, "VIVE LES MATHS" devient donc "YLYH OHV PDWKV" !

La sécurité d'un tel chiffrement est assez limitée puisqu'il n'existe que 26 façons différentes de crypter un message avec ce code. Dès lors, des attaques exhaustives (tester toutes les décalages un à un) ne demanderaient que très peu de temps.

II.3.3. Chiffre de Vigenère

Malgré quelques faiblesses, aucune amélioration n'est survenue entre César et le XVIème siècle en matière de procédé cryptographique, à la fois sûr et facile à utiliser ! Un chiffrement intéressant a été développé par Battista aux alentours de 1467 après Jésus-Christ, bien qu'il fût plus tard attribué à Vigenère. Son idée, que l'on peut qualifier de révolutionnaire pour l'époque, à propos de cette nouvelle substitution consistait à utiliser un chiffre de César différent pour chaque lettre, suivant une clé choisie préalablement par les protagonistes de la communication. On modifie donc le décalage pour chaque lettre. Il constitue donc un chiffre par substitution poly alphabétique.

Pour ce faire, on utilise la table suivante :

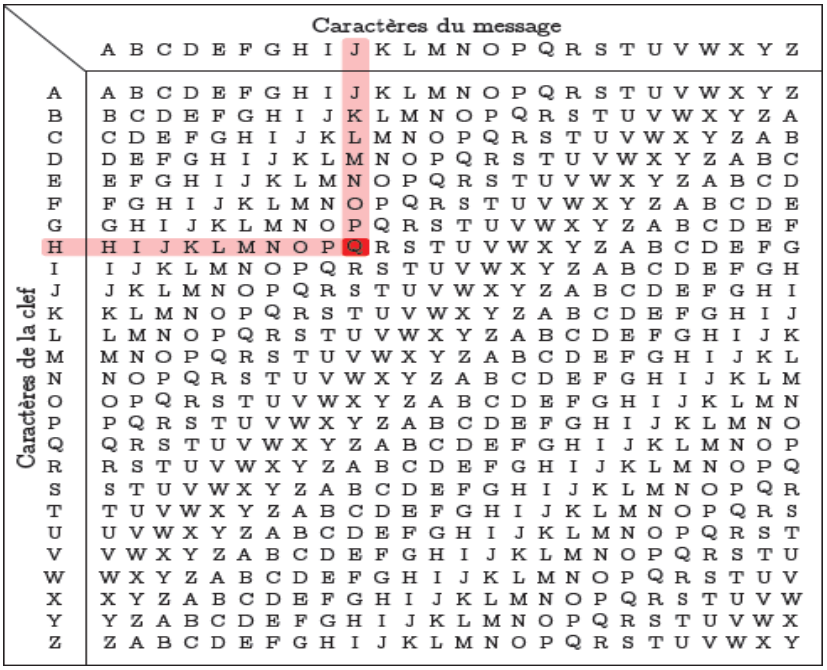


Tableau 2-1 Carré de Vigenère

La méthodologie pour coder un message est la suivante : on commence par choisir une clé (un mot de longueur arbitraire). On reproduit ensuite ce mot sous la totalité du texte à coder autant que nécessaire de façon à ce que toutes les lettres du texte soit associée à une lettre de la clé. Enfin, on regarde dans le tableau la lettre qui se trouve à l'intersection de la lettre du texte et de celle de la clé.

Exemple, on va coder le texte « La cryptographie » avec la clé « SRETI ». On commence donc par écrire la clé sous le texte à coder :

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

Pour coder la lettre H, on regarde l'intersection de la colonne issue de H avec la ligne issue de E et l'intersection donne la lettre L. Puis on continue et on obtient au final : DR GKGHKSZZSGLBM

Ce code n'est certes pas infaillible mais résout l'un des problèmes qui se posait précédemment. Le E a été codé en E, R, I et X. Ainsi, une analyse statistique simple ne peut pas permettre de découvrir où se trouvent les E. Par ailleurs, on peut produire une infinité de clés qui peuvent être des mots voire des phrases. Pour toutes ces raisons, cette méthode s'est imposée durant plus de 3 siècles.

Le défaut principal de cette méthode de chiffrement, est la longueur limitée de la clé. Comme la clé est plus courte que le message, elle se trouve répétée. Ainsi l'attaquant peut essayer de déterminer la longueur de la clé grâce à des répétitions de certains motifs dans le

message chiffré. En effet, lorsque dans le texte chiffré, on peut retrouver plusieurs fois le même motif, il est fort probable que ce soit le même texte clair qui en soit à l'origine. Cela permet de déduire un multiple du nombre de caractères de la clé. Une fois la longueur de la clé connue, une attaque en fréquence analogue au chiffre de César est possible.

II.3.4. La machine Enigma

La machine allemande Enigma a joué un grand rôle pendant la guerre de l'Atlantique, et son décryptement par les Alliés leur a assuré bon nombre de victoires (notamment parce que les Allemands ne se doutaient pas que leurs messages étaient déchiffrés).

Enigma ressemble à une machine à écrire : on frappe le clair sur un clavier, et des petites lampes s'allument pour éclairer les lettres résultant du chiffrement.

Le principe de chiffrement qu'utilise Enigma est à la fois simple et astucieux. Simple, car il ne s'agit ni plus ni moins d'une substitution de lettres : par exemple, A devient Q, P devient N, etc. Et astucieux, parce que la substitution change d'une lettre à une autre : si la lettre A correspond à Q la première fois qu'on la saisit, elle pourrait correspondre à M, K, H, ou tout autre lettre différente de Q à la fois suivante (ce principe est possible grâce à un système de rotors).

De plus, un autre avantage non négligeable que possède Enigma est la réversibilité : si on tape le message clair, on obtient le message code, et avec le message codé, on obtient le message clair.

Figure 2-1 La machine Enigma

Notre étude portera sur la machine Enigma, les détails du fonctionnement de cette machine seront présentés dans le chapitre qui suit.

II.4. Intégrité et authentification

La cryptographie peut assurer l'intégrité et l'authentification des données en proposant plusieurs moyens parmi ces moyens on peut trouver :

II.4.1. Hachage [WEB 02]

Le système décrit précédemment comporte certains inconvénients. Il est lent et produit un volume important de données (au moins le double de la taille des informations d'origine).

L'ajout d'une fonction de hachage à sens unique (ie : qu'il n'existe pas de fonction mathématique ni d'algorithme pour effectuer l'opération inverse de celle effectuée par la fonction de hachage) dans le processus permet d'améliorer la figure ci-dessous. Cette fonction traite une entrée de longueur variable (dans ce cas, un message pouvant contenir indifféremment des milliers ou des millions de bits), afin d'obtenir en sortie un élément de longueur fixe, par exemple 128bits, appelée « valeur hash » En cas de modification des données (même d'un seul bit), la fonction de hachage garantit la production d'une « valeur hash » différente. On utilise le résumé (hash) et la clé privée pour générer la « signature ».

Le logiciel de cryptographie transmet en même temps la signature et le texte en clair. A la réception du message par le destinataire, son logiciel traite à nouveau le message, vérifiant ainsi la signature.

Si une fonction de hachage sécurisée est utilisée, il est impossible de récupérer la signature d'un document pour la joindre à un autre document ou d'altérer un message signé.

En effet, la moindre modification apportée à un document signé entraîne l'échec du processus de vérification de la signature numérique.

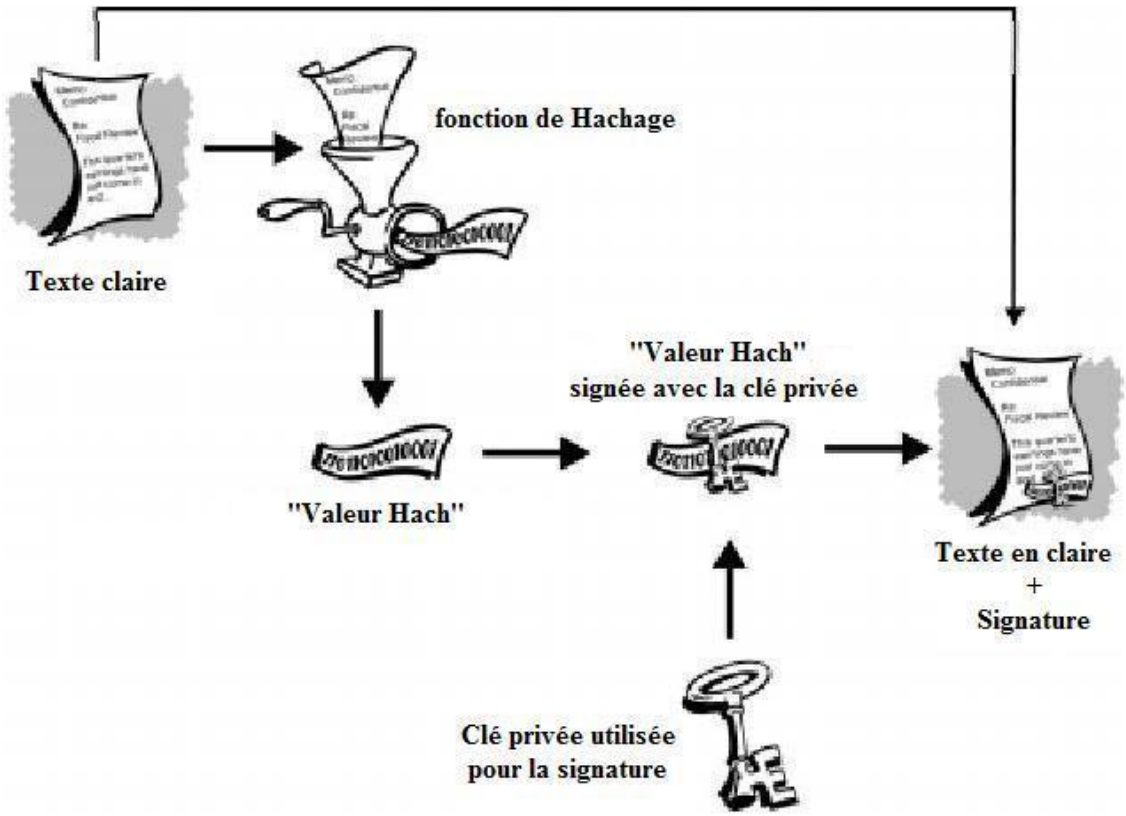


Figure 2-2 Fonction de Hachage.

Par exemple si Alice souhaite envoyer un fichier par mail à Bob, mais que ce fichier est de taille importante. Elle souhaite de plus rassurer Bob sur la provenance de ce fichier (Alice) et sur son contenu. Plutôt que de chiffrer son fichier directement avec sa clé privée, elle va hacher son fichier et chiffrer le condensé obtenu avec sa clé privée. Elle enverra ensuite son fichier original ainsi que le condensé chiffré (la signature) à Bob.

Celui-ci va, lors de la réception, hacher d'une part le fichier reçu et d'autre part déchiffrer le condensé reçu (au moyen de la clé publique d'Alice).

S'il n'y a pas égalité entre les 2 résultats, cela signifiera :

- soit que la signature n'est plus celle d'Alice, donc que quelqu'un a intercepté le fichier (pour le modifier ou le remplacer, …etc.)
- soit que le fichier n'est plus le même que l'original (mais la signature n'a pas été remplacée); dans ce cas, le hachage ne peut plus donner le même condensé ce qui conduit au rejet lors du test de comparaison.

Dans les 2 cas, ni l'intégrité ni l'authentification du fichier n'ont été vérifiées. Il ne faut donc pas faire confiance au fichier.

Nous voyons comment dans ce cas simple, l'utilisation d'une fonction de hachage permet de s'assurer de l'intégrité des données et indirectement de les authentifier. Il existe bien sûr de nombreuses autres applications pour les fonctions de hachage, comme les MACs (message authentication code), certificats, …etc.

II.4.2. La clé [9]

Une clé est une valeur utilisée dans un algorithme de cryptographie, afin de générer un texte chiffré. Les clés sont en réalité des nombres extrêmement importants. La taille d'une clé se mesure en bits et le nombre correspondant à une clé de 1 024 bits est gigantesque.

Dans la cryptographie de clé publique, plus la clé est grande, plus la sécurité du texte chiffré est élevée. Cependant, la taille de la clé publique et de la clé secrète de cryptographie conventionnelle sont complètement indépendantes. Une clé conventionnelle de 80 bits est aussi puissante qu'une clé publique de 1 024 bits. De même, une clé conventionnelle de 128 bits équivaut à une clé publique de 3 000 bits. Encore une fois, plus la clé est grande, plus elle est sécurisée, mais les algorithmes utilisés pour chaque type de cryptographie sont très différents.

Même si les clés publiques et privées sont liées par une relation mathématique, il est très difficile de deviner la clé privée uniquement à partir de la clé publique. Cependant, la déduction de la clé privée est toujours possible en disposant de temps et de puissantes ressources informatiques. Ainsi, il est très important de sélectionner des clés de tailles correctes, suffisamment grandes pour être sécurisées, mais suffisamment petites pour être utilisées assez rapidement.

Plus la clé est grande, plus sa durée de sécurisation est élevée. Si les informations que l'on souhaite crypter doivent rester confidentielles pendant plusieurs années, on peut utiliser une clé correspondant à un nombre de bits extrêmement élevé.

Les clés sont stockées sous forme cryptée. Par exemple PGP conserve les clés sur le disque dur, dans deux fichiers : l'un est destiné aux clés publiques, l'autre aux clés privées. Ces fichiers s'appellent des trousseaux de clés. Lors de l'utilisation de PGP, on doit généralement ajouter les clés publiques des destinataires sur le trousseau de clés publiques. Les clés privées sont stockées sur le trousseau de clés privées. En cas de perte du trousseau de clés privées, il sera impossible de décrypter les informations cryptées vers les clés de ce trousseau.

II.4.3. Signature numérique [6]

L'un des principaux avantages de la cryptographie de clé publique est qu'elle offre une méthode d'utilisation des *signatures numériques*. Celles-ci permettent au destinataire de vérifier leur authenticité, leur origine, mais également de s'assurer qu'elles sont intactes. Ainsi, les signatures numériques de clé publique garantissent l'*authentification* et l'*intégrité* des données. Elles fournissent également une fonctionnalité de *non répudiation*, afin d'éviter que l'expéditeur ne prétende qu'il n'a pas envoyé les informations. Ces fonctions jouent un rôle tout aussi important pour la cryptographie que la confidentialité.

Une signature numérique a la même utilité qu'une signature manuscrite. Cependant, une signature manuscrite peut être facilement imitée, alors qu'une signature numérique est pratiquement infalsifiable. De plus, elle atteste du contenu des informations, ainsi que de l'identification du signataire.

Supposons, par exemple, qu'Alice veut signer numériquement un message destiné à Bob. Pour ce faire, elle utilise sa clé privée pour chiffrer le message, puis elle envoie le message accompagné de sa clé publique (habituellement, la clé publique est jointe au message signé). Étant donné que la clé publique d'Alice est la seule clé qui puisse déchiffrer ce message, le déchiffrement constitue une vérification de signature numérique, ce qui signifie qu'il n'y a aucun doute que le message ait été chiffré à l'aide de la clé privée d'Alice.

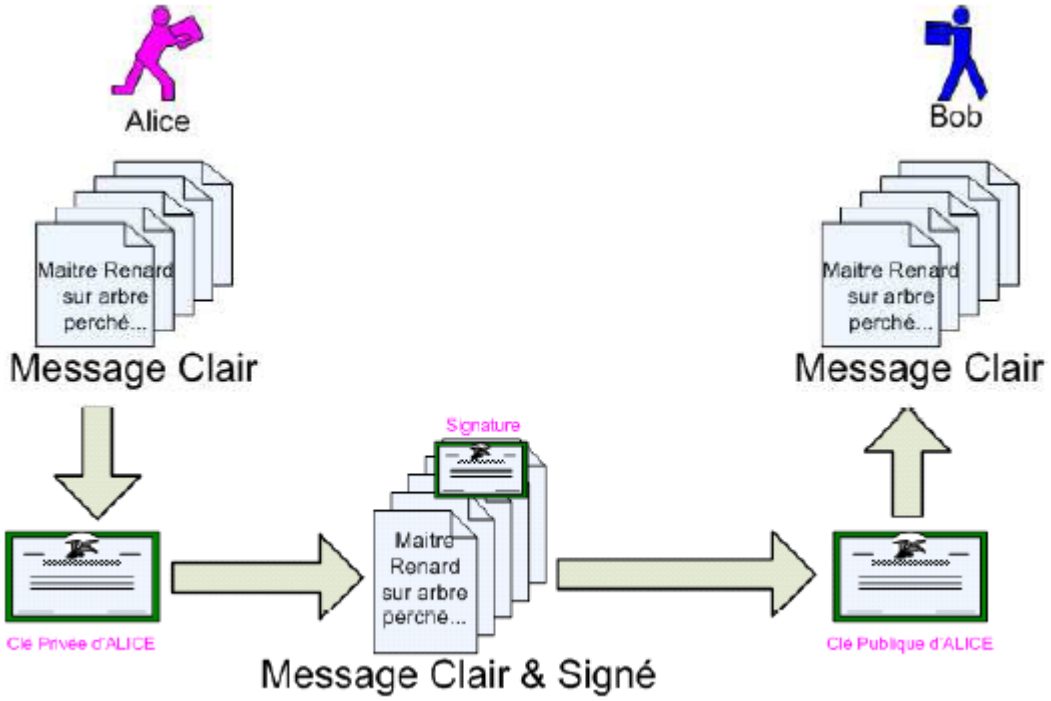


Figure 2-3 Cryptage avec signature.

II.4.4. Certificat [9]

Un certificat est un élément d'information qui prouve l'identité du propriétaire d'une clé publique. Ainsi un certificat permet d'associer une clé publique à une entité (une personne, une machine, ...) afin d'en assurer la validité. Les certificats sont signés et transmis de façon sécuritaire par un tiers de confiance appelé autorité de certification (AC). Tant que Bob et Alice ont confiance en ce tiers, ils peuvent être assurés que les utilisateurs de ces clés en sont bel et bien les propriétaires.

Les certificats sont des petits fichiers divisés en deux parties :

- La partie contenant les informations
- La partie contenant la signature de l'autorité de certification

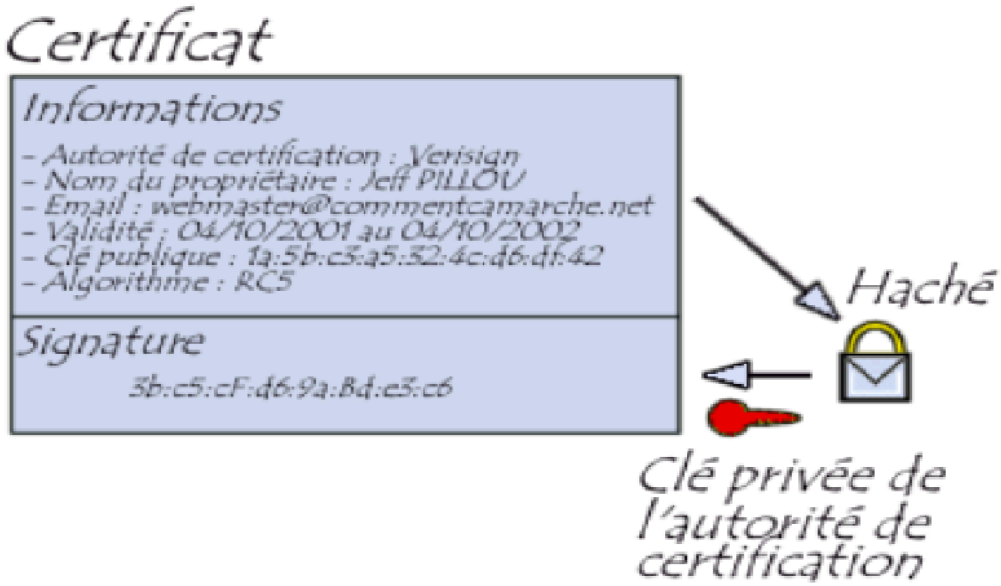
La structure des certificats est normalisée par le standard **X.509** de l'UIT [2](plus exactement X.509v3), qui définit les informations contenues dans le certificat :

- La version de X.509 à laquelle le certificat correspond ;
- Le numéro de série du certificat ;
- L'algorithme de chiffrement utilisé pour signer le certificat ;
- Le nom (DN, pour *Distinguished Name*) de l'autorité de certification émettrice ;
- La date de début de validité du certificat ;
- La date de fin de validité du certificat ;
- L'objet de l'utilisation de la clé publique ;

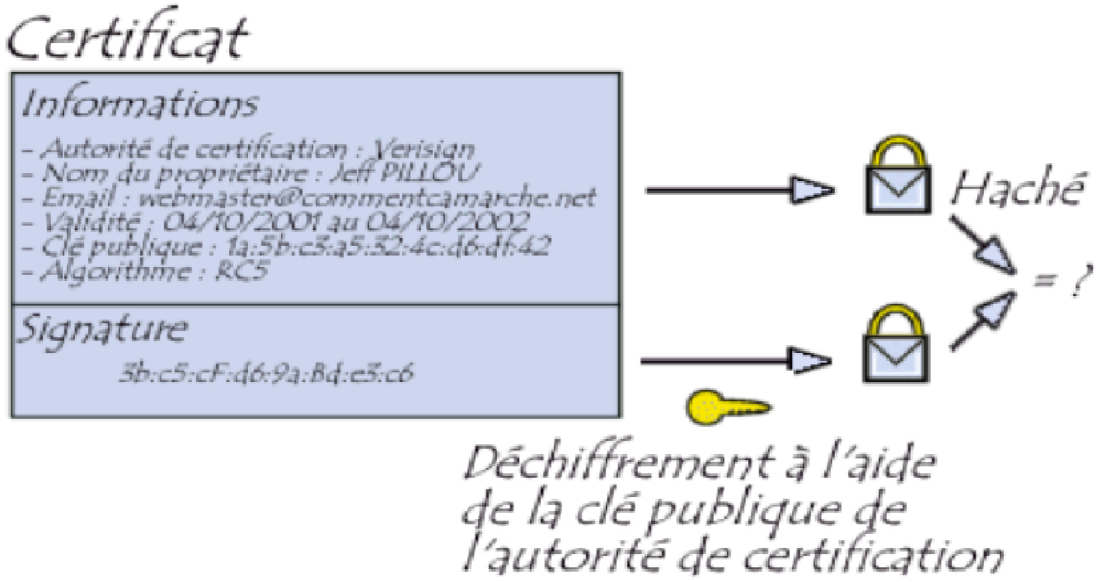
[2] L'UIT (Union internationale des télécommunications) est l'institution spécialisée des Nations Unies pour les technologies de l'information et de la communication (TIC).

- La clé publique du propriétaire du certificat ;
- La signature de l'émetteur du certificat (*thumbprint*).

L'ensemble de ces informations (informations + clé publique du demandeur) est signé par l'autorité de certification, cela signifie qu'une fonction de hachage crée une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé privée de l'AC; la clé publique ayant été préalablement largement diffusée afin de permettre aux utilisateurs de vérifier la signature avec la clé publique de l'AC.



Par exemple lorsqu'Alice désire communiquer avec Bob, il lui suffit de se procurer le certificat de Bob. Ce certificat contient le nom de Bob, ainsi que sa clé publique et est signé par l'autorité de certification. Il est donc possible de vérifier la validité du message en appliquant d'une part la fonction de hachage aux informations contenues dans le certificat, en déchiffrant d'autre part la signature de l'autorité de certification avec la clé publique de cette dernière et en comparant ces deux résultats.



Notez que les certificats sont signés par une AC, ce qui signifie qu'ils ne peuvent être altérés. La signature de l'AC peut, à son tour, être vérifiée à l'aide du certificat de cette AC.

II.5. Conclusion

Nous avons vu un panel de méthodes de chiffrement de l'antiquité à nos jours, les attaques existantes sur les cryptosystèmes actuels les plus utilisées et les moyens inventés pour s'assurer de l'intégrité, de l'authentification de l'expéditeur et du destinataire d'un message.

Ainsi, la cryptographie est une science en perpétuelle évolution, la cryptanalyse aidant à trouver les failles d'un système pour toujours avancer. Cette évolution est importante car la cryptographie joue un grand rôle dans la sécurité internationale, tout étant aujourd'hui informatisé.

Pourtant, même si la cryptanalyse permet de faire avancer la cryptographie avec des méthodes de chiffrement et une technologie toujours plus poussées, elle représente aussi un danger à l'échelle internationale.

Toute la sécurité informatique serait remise en question et ce que nous connaissons aujourd'hui tels que les sites de vente en ligne basées sur ces algorithmes ne fonctionneraient plus. Par conséquent, c'est non seulement la sécurité internationale qui serait touchée, mais aussi toute une économie.

Chapitre III
Les Cryptosystèmes

III.1. Introduction

La révolution d'Internet et l'utilisation de plus en plus massive d'informations sous forme numérique facilitent les communications et rendent fragiles les informations que l'on détient. De ce fait, les primitives cryptographiques ont toujours attiré l'attention des attaquants. Le domaine de la cryptologie voit donc une lutte permanente entre les cryptographes, qui conçoivent ces mécanismes, et les cryptanalystes, qui cherchent à les mettre en défaut. Donc pour mieux assurer la sécurité et la confidentialité des informations, il est important de faire appel à des outils qui sont à la pointe de l'état de l'art dans le domaine.

Dans ce chapitre on verra les différentes méthodes de cryptage tout en citant quelques exemples de cryptage avec des algorithmes déjà existants. Et enfin, nous apporterons plus de détails sur le fonctionnement de la machine Enigma sur laquelle se base notre travail.

III.2. Mécanismes de la cryptographie [6]

Un *algorithme de cryptographie* ou un *chiffrement* est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clé (un mot, un nombre ou une phrase), afin de crypter le texte en clair. Avec des clés différentes, le résultat du cryptage variera également.

La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clé.

Un *système de cryptographie* est constitué d'un algorithme de cryptographie, ainsi que de toutes les clés et tous les protocoles nécessaires à son fonctionnement.

III.3. Les cryptosystèmes actuels

III.3.1. Cryptographie symétrique [10]

En cryptographie *symétrique*, également appelée cryptage de *clé secrète* ou cryptographie *conventionnelle*, une seule clé suffit pour le cryptage et le décryptage. Dans ce système, la clé qui sert au chiffrement des informations sert également à leur déchiffrement. Il est ainsi généralement qualifié d'algorithme « symétrique ».
La sécurité de cette solution repose sur le fait que la clé qui sert aussi bien pour l'encryptage que pour le décryptage est connue uniquement par l'émetteur et le récepteur du message.

III.3.1.1. Le principe de base

Un expéditeur et un destinataire souhaitant communiquer de manière sécurisée à l'aide du cryptage conventionnel doivent convenir d'une clé et ne pas la divulguer. La Figure 3-1 est une illustration du processus de cryptage conventionnel.

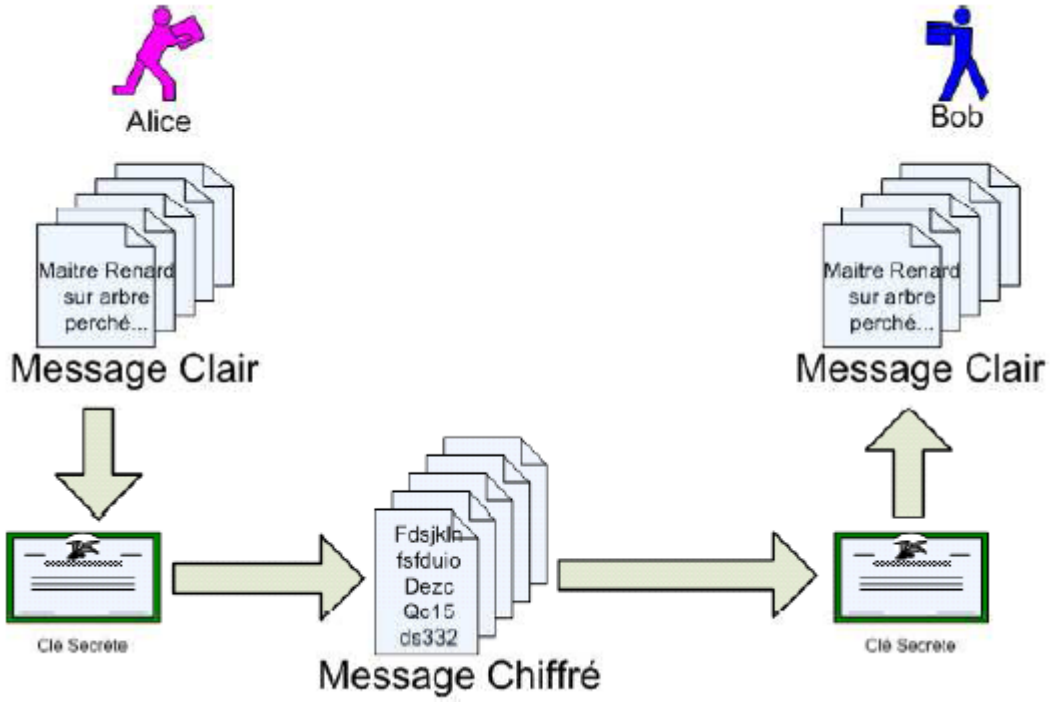


Figure 3-1 Cryptage conventionnel.

III.3.1.2. Le chiffrement par flots

Dans un cryptosystème par flots, le cryptage des messages se fait caractère par caractère ou bit à bit, au moyen de substitutions de type César générées aléatoirement : la taille de la clé est donc égale à la taille du message. L'exemple le plus illustratif de ce principe est le chiffre de Vernam. Cet algorithme est aussi appelé « One Time Pad » (masque jetable), c'et à dire que la clé n'est utilisée qu'une seule fois.

Voici un exemple simple de l'application du chiffre de Vernam :

Exemple :

Message en clair: "**SALUT**"

=> (conversion en binaire)

01010011 01000001 01001100 01010101 01010100

XOR

Clé (générée aléatoirement)

01110111 01110111 00100100 00011111 00011010

=

00100100 00110110 01101000 01001010 01001110

=> (conversion en caractère)

"Message chiffré: **$6jJM**"

Il a été démontré par le mathématicien Claude Elwood Shannon qu'il était impossible de retrouver un message crypté par le principe de Vernam sans connaître la clé. Ce qui ferait en théorie du chiffre de Vernam un cryptosystème incassable. Mais dans la pratique, le cryptosystème par flots pose des problèmes délicats : canaux sûrs de distribution des clés, taille des clés encombrantes car de même taille que le message et surtout caractère aléatoire des générateurs de bits de clés utilisés. En revanche, un des avantages du système est qu'il est insensible aux phénomènes de propagation d'erreurs : un bit erroné donne une erreur à la réception ou à l'émission, mais est sans incidence sur les bits suivants.

III.3.1.3. Le chiffrement par blocs

Les primitives ("combinaison d'algorithmes" ou "pratiques") de chiffrement par flot permettent d'effectuer un chiffrement rapide en traitant les données bit par bit mais cela n'a conduit à aucun cryptosystème avec des garanties suffisantes en termes de sécurité à ce jour. À la place, c'est le principe de chiffrement par blocs qui a permis l'élaboration de la plupart des primitives utilisées de nos jours.

Le principe est simple : Emil découpe le message à traiter en blocs de longueur fixe, et on traite ces blocs de manière successive. On va donc transformer un message clair de taille fixe **N** en plusieurs blocs de taille **n** que l'on va chiffrer en plusieurs messages chiffrés de taille **n** que l'on va ensuite réassembler pour obtenir le message chiffré final (de taille **N**). Cette opération doit être inversible.

III.3.1.4. Exemples d'algorithmes symétriques

1. DES (Data Encryption Standard) [WEB 03]

Publié en 1977 par le NBS (National Bureau of Standards), le DES est un algorithme de chiffrement de données recommandé pour les organisations à caractère fédéral, commercial ou privé. Le DES tire son origine des travaux menés par le groupe cryptographique d'IBM dans le cadre du projet LUCIFER. Le DES a été l'objet de nombreuses implémentations, à la fois en matériel et en logiciel, depuis sa publication. Après une décennie de succès, pendant laquelle les moyens et techniques de cryptanalyse mis en œuvre pour en étudier les caractéristiques n'ont pas permis d'en découvrir des faiblesses rédhibitoires, le DES a, depuis peu, révélé des sensibilités à des attaques nouvelles et puissantes, parfois réalisées sur un simple micro-ordinateur. Aussi l'ISO (International Organization for Standardization) a-t-il récemment refusé la normalisation du DES, ce qui n'empêche pas cet algorithme d'être, de loin, aujourd'hui encore comme le moyen de chiffrement le plus sûr (et le plus largement utilisé) pour des données non militaires.

Le DES est un algorithme de chiffrement symétrique par blocs qui permet de chiffrer des mots de 64 bits à partir d'une clé de 56 bits (56 bits servant à chiffrer + 8 bits de parité servant à vérifier l'intégrité de la clé en réalité).

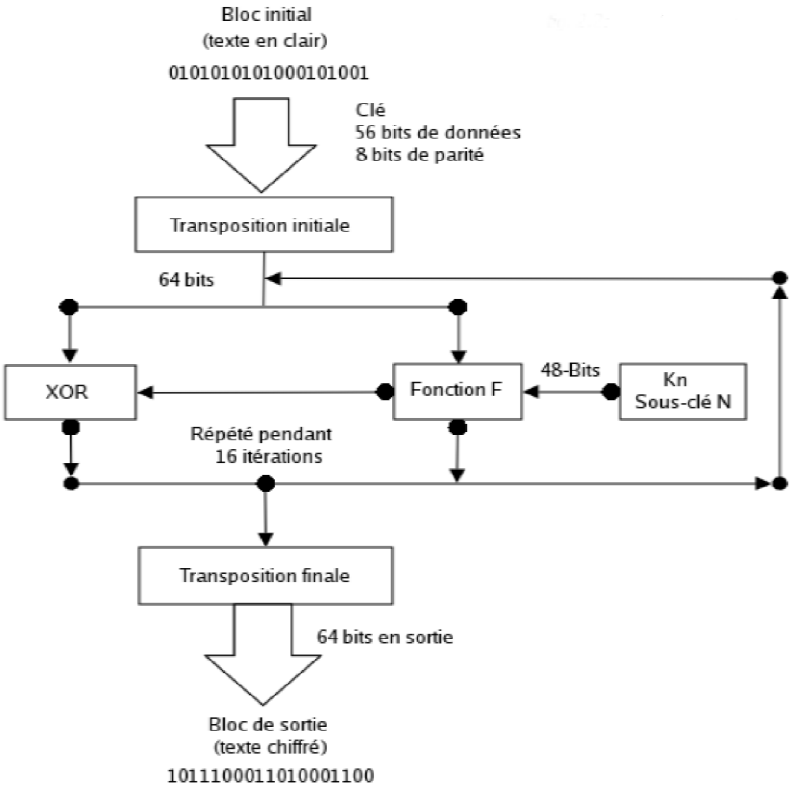


Figure 3-2 Les étapes de l'algorithme DES.

Voici les différentes étapes de l'algorithme du DES :

- ✓ **Fractionnement du message :**

Dans un premier temps le message en clair est découpé en blocs de 64 bits.

- ✓ **Transposition initiale**

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

Chaque bit d'un bloc subit une permutation selon l'arrangement du tableau ci-contre c'est-à-dire que le 58ème bit du bloc se retrouve en 1ère position, le 50ème en seconde position, …etc.

- ✓ **Scindement en bloc de 32 bits**

Le bloc de 64 bits est scindé en deux blocs de 32 bits notés G et D. On notera G0 et D0 l'état initial de ces deux blocs.

G_0

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

D_0

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

On remarque que G0 contient tous les bits pairs du message initial et D0 tous les bits impairs.

✓ **Rondes**

Les blocs G_i et D_i sont soumis à un ensemble de transformations appelées rondes. Une ronde est elle-même composée de plusieurs étapes :

- **Fonction d'expansion :**

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Les 32 bits du bloc D0 sont étendus à 48 bits grâce à une table d'expansion dans laquelle 32 bits sont mélangés et 16 d'entre eux sont dupliqués.

Ainsi, le 32ème bit devient le premier, le premier devient le second… Les bits 1,4,5,8,9,12,13,16,17,22,21,24,25,28,29 et 32 sont dupliqués et disséminés pour former un bloc de 48 bits que l'on nommera D'0 .

- **OU exclusif (XOR) avec la clé :**

DES procède ensuite à un OU exclusif entre D'0 et la première clé k1 générée à partir de la clé K (que doivent se partager émetteur et destinataire) par l'algorithme de cadencement des clés que nous décrirons plus bas. Nous appellerons D''0 le résultat de cette opération.

- **Boîtes de substitution :**

D''0 est découpée ensuite en 8 blocs de 6 bits, noté $D''0_i$. Chacun de ces blocs passe par des boîtes de substitution (S-boxes), notées généralement S_i. Les premier et dernier bits de chaque $D0_i$ déterminent la ligne de la fonction de substitution, les autres bits déterminent la colonne. Grâce à cela la fonction de substitution « choisit » une valeur codée sur 4 bits (de 0 à 15).

Voici la première boîte de substitution :

S_1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Soit $D0_i$ égal à 010101, les premiers et derniers bits donnent 01, c'est-à-dire 1 en binaire. Les bits autres bits donnent 1010, soit 10 en binaire. Le résultat de la fonction de substitution est donc la valeur située à la ligne n°1, dans la colonne n°10. Il s'agit de la valeur 6, soit 0110 en binaire.

Chacun des 8 blocs de 6 bits est passé dans la boîte de substitution correspondante. Voici les autres S-Boxes :

S_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	5	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

On obtient donc en sortie 8 blocs de 4 bits. Ces bits sont regroupés pour former un bloc de 32 bits.

- **Permutation :**

Le bloc de 32 bits subit une permutation dont voici la table :

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

- **OU exclusif :**

Le bloc de 32 bits ainsi obtenu est soumis à un OU exclusif avec le G0 de départ pour donner D1 et le D0 initial donne G1.

L'ensemble de ces étapes est itérée seize fois.

✓ **Transposition initiale inverse**

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

Au bout des seize itérations, les deux blocs G16 et D16 sont « recollés » pour reformer un seul bloc de 64 bits puis subit la transposition initiale inverse selon l'arrangement du tableau ci-contre.

On obtient alors le bloc initial chiffré.

✓ **Reconstruction du message chiffré**

Tous les blocs sont collés bout à bout pour obtenir le message chiffré.

✓ **Algorithme de cadencement des clés**

Nous allons décrire l'algorithme qui permet de générer à partir d'une clé de 64 bits, 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES.

De prime abord les clés de parité sont éliminées pour obtenir une clé de 56 bits.

Ce bloc subit une permutation puis est découpée en deux pour obtenir 2 blocs de 28 bits décrits par les matrices ci-dessous :

40	8	48	16	56	24	64
39	7	47	15	55	23	63
38	6	46	14	54	22	62
37	5	45	13	53	21	61

40	8	48	16	56	24	64
39	7	47	15	55	23	63
38	6	46	14	54	22	62
37	5	45	13	53	21	61

Ces deux blocs subissent une rotation à gauche, c'est-à-dire que les bits en seconde position prennent la première position, ceux en troisième position la seconde, celle en première position la dernière…

14	17	11	24	1	5	3	28	15	6	21	10
13	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	57	45
44	49	39	56	34	53	46	42	50	36	29	32

Les 2 blocs sont regroupés pour faire un bloc de 56 bits qui passe par une permutation fournissant un bloc de 48 bits représentant la clé k_i. :

Des itérations de l'algorithme permettent de donner les 16 clés utilisées dans l'algorithme de DES.

Bien que sur le plan de la cryptologie, on constate que la cryptanalyse (l'art et la manière de casser les codes secrets) n'a pas réussi à faire "craquer" le DEA (Data Encryption Algorithm), l'algorithme de chiffrement employé par le DES.

Par contre sur le plan informatique, l'évolution fulgurante de ces dernières années a révélé les limitations de construction du DES. L'algorithme DEA traite les données à chiffrer par blocs de 64 bits (8 octets) et combine ces blocs avec une clé secrète de même longueur. Sur ces 8 octets de longueur de clé, seuls 7 sont réellement utiles, le dernier servant au contrôle de parité. Il existe donc 2^{56} clés secrètes possibles pour découvrir le sens caché d'un message chiffré par le DEA. L'attaque dite à force brute consistant à tester de manière exhaustive l'ensemble des clés possibles est désormais à portée de processeurs, surtout si ceux-ci sont utilisés en parallèle. L'Electronic Frontier Foundation a construit un "DES-Cracker" qu'elle a présenté au congrès Crypto'98 à Santa Barbara. Cette machine qui avoisinerait les 2 millions de francs serait en mesure de trouver une clé secrète DES en 4 jours. Il parait clair que de telles machines existent déjà et sont utilisées par certains gouvernements et, pourquoi pas, par le crime organisé. Les conséquences économiques,

politiques et diplomatiques sont considérables. Pour le présent et le futur, il est désormais nécessaire de considérer comme peu sûrs tous les systèmes basés sur DES.

2. Rijndael (AES) [WEB 03]

En Janvier 1997, la NIST (National Institute of Standards and Technology) lance un appel d'offre international pour remplacer le vieillissant DES : il en résulte 15 propositions.

Parmi ces 15 algorithmes, 5 furent choisis pour une évaluation plus avancée en avril 1999 : MARS, RC6,Rijndael, Serpent et Twofish. Finalement, en octobre 2000 la NIST élit Rijndael comme nouveau standard qu'on nomme aussi AES (Advanced Encryption Standard).

Rijndael, du nom condensé de ses concepteurs Rijmen et Daemen, est un algorithme de chiffrement par blocs à plusieurs tours similaire à DES mais avec une taille de blocs et de clés supérieures et variables, choisis entre 128, 196 et 256 bits.

✓ Le Corps GF(28)

Un octet b composé des 8 bits $b_7,b_6,b_5,b_4,b_3,b_2,b_1,b_0$ peut être vu comme un polynôme de degré inférieur ou égal à 7 avec des coefficients dans {0,1} :

$$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$

L'addition de deux polynômes de ce type revient à additionner modulo 2 les coefficients de chacun. Cette addition correspond au OU exclusif.

Pour la multiplication, c'est la multiplication usuelle suivie d'une réduction modulo un polynôme binaire irréductible de degré 8.

Dans Rijndael, ce polynôme est $\mathbf{m(x) = x^8 + x^4 + x^3 + x + 1}$. Le résultat sera à nouveau un polynôme de degré inférieur ou égal à 7.

Pour tout polynôme binaire de degré inférieur ou égal à 8, l'algorithme d'Euclide étendu permet de calculer b(x) tel que a(x) b(x) mod m(x) soit égal à 1, autrement de calculer l'inverse de a(x) : $a^{-1}(x)$.

On peut voir que l'ensemble des 256 éléments possibles, avec l'addition et la multiplication ci-dessus, ont la structure du corps $GF(2^8)$: le corps fini de polynômes de degré ≤ 7 avec des coefficients dans {0,1}.

✓ Structure d'état dans l'AES

On appelle état un bloc vu comme un tableau de 4 x Nb octets où Nb est égal à Taille du bloc / 32. On représente la clé de la même façon, le nombre de colonnes étant :

Nk = longueur de la clé / 32.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Exemple d'état (avec des blocs de 128 bits, Nb = 4) et de clé (de longueur 128 bits, Nk = 4).

✓ **Nombre de tours**

Le nombre de tours dans l'AES dépend à la fois de la taille des blocs et de la clé. Le nombre r de tours est donné par le tableau :

N_r	$N_b = 4$ (128 bits)	$N_b = 6$ (192 bits)	$N_b = 8$ (256 bits)
$N_k = 4$ (128 bits)	10	12	14
$N_k = 6$ (192 bits)	12	12	14
$N_k = 8$ (256 bits)	14	14	14

Chaque tour utilise une sous-clé différente et qui est composée de quatre étapes : ByteSub, ShiftRow, MixColumn et AddRoundKey.

1) ByteSub

ByteSub est une substitution qui agit isolément sur tous les octets $a_{i,j}$ d'un état en 2 étapes :

1. on regarde $a_{i,j}$ comme polynôme dans GF(2^8), et on prend son inverse $a^{-1}_{i,j}$

2. on calcule l'image du résultat par la fonction y = f(x) suivante :

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}$$

2) ShiftRow

ShiftRow effectue un décalage des lignes de l'état courant. La ligne 0 n'est pas décalée, la ligne 1 l'est de **C1** octets, la 2 de **C2** octets et la ligne 3 de **C3** octets. Les valeurs de C1, C2 et C3 dépendant de la taille du bloc, selon la table suivante :

N_b	C_1	C_2	C_3
4	1	2	3
6	1	2	3
8	1	3	4

3) MixColumn

La transformation MixColumn consiste à prendre chaque colonne de l'état et à la multiplier par la matrice suivante:

$$\begin{pmatrix} b_{0,x} \\ b_{1,x} \\ b_{2,x} \\ b_{3,x} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} = \begin{pmatrix} a_{0,x} \\ a_{1,x} \\ a_{2,x} \\ a_{3,x} \end{pmatrix}$$

4) AddRoundKey

AddRoundKey consiste en un OU exclusif de l'état courant et de la clé du tour.

III.3.1.5. Les avantages et inconvénients du cryptage symétrique

- ✓ **Les avantages :**

- la rapidité d'exécution (une seule clé utilisée)

- la simplicité d'implémentation (gestion d'une seule clé)

- ✓ **Les inconvénients :**

- la complexité de fonctionnement : une obligation d'avoir le nombre de clés privées égal au nombre de destinataires.

- la sécurisation de la chaîne de transmission de la clé.

III.3.2. Cryptographie asymétrique

La cryptographie asymétrique est un procédé asymétrique utilisant une paire de clés pour le cryptage : une clé publique qui crypte des données et une clé privée ou *secrète* correspondante pour le décryptage. Le créateur peut ainsi publier sa clé publique tout en conservant sa clé privée secrète. Tout utilisateur possédant une copie de sa clé publique peut ensuite crypter des informations que le créateur sera le seul à pourvoir lire. Même les personnes qu'il ne connait pas personnellement peuvent utiliser sa clé publique. D'un point de vue informatique, il est impossible de deviner la clé privée à partir de la clé publique.

III.3.2.1. Le principe de base [WEB 02]

Le principe est donc de distribuer la clé publique tout en conservant la clé privée secrète. Tout utilisateur possédant une copie de la clé publique pourra ensuite crypter des informations que seul le propriétaire de la clé privée pourra déchiffrer.

Il faut également noter que si le cryptage est bien possible à l'aide de la clé publique, l'opération inverse (le décryptage) ne sera pas possible au moyen de la clé publique mais exigera l'utilisation de la clé privée correspondante.

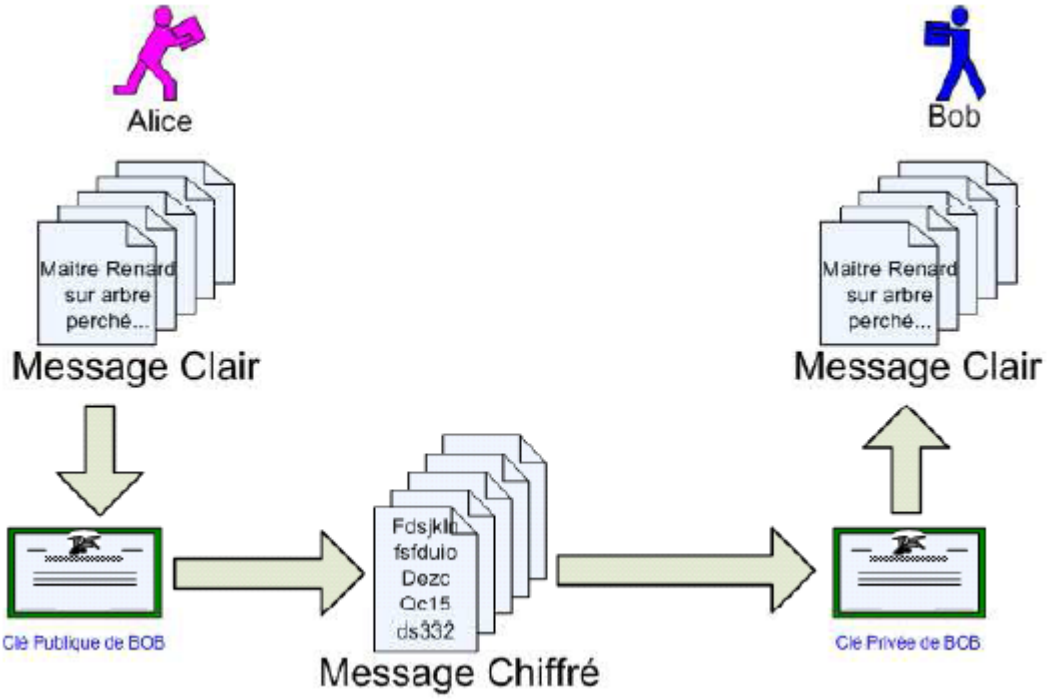


Figure 3-3 Cryptage de clé publique.

L'application de la cryptographie asymétrique permet aussi des fonctions d'authentification de documents et d'expéditeurs. L'authentification de documents consiste à générer une signature à partir du document original et de la clé privée de l'expéditeur. Le destinataire en possession de la clé publique de l'expéditeur peut alors s'assurer de l'identité de celui qui a signé le message.

III.3.2.2. Exemple d'algorithmes asymétriques

Elgamal (d'après le nom de son inventeur, Taher Elgamal), RSA (d'après le nom de ses inventeurs, Ron Rivest, Adi Shamir et Leonard Adleman), Diffie-Hellman (également d'après le nom de ses inventeurs) et DSA, l'algorithme de signature numérique (élaboré par David Kravitz), sont des exemples de systèmes de cryptographie de clé publique.

1. RSA [11]

La méthode de cryptographie RSA a été inventée en 1977 par Ron Rivest, Adi Shamir et Len Adleman, à la suite de la découverte de la cryptographie à clé publique par Diffie et Hellman [WEB 04]. Le RSA est encore le système cryptographique à clé publique le plus utilisé de nos jours. Il est intéressant de remarquer que son invention est fortuite : au départ, Rivest, Shamir et Adleman voulaient prouver que tout système à clé publique possède une faille.

Cet algorithme repose sur la difficulté de factoriser des grands nombres. Le principe est décrit par la séquence d'étapes suivantes :

1. On commence par choisir deux grands nombres premiers, p et q, et on calcule n = p*q.

n est rendu public ; p et q doivent rester secrets et sont donc détruits une fois les clés générées.

2. On choisit ensuite aléatoirement une clé publique e telle que e et (p-1)*(q-1) soient premiers entre eux.
3. La clé privée d est obtenue grâce à l'algorithme d'Euclide : e*d ≡ 1 mod (p-1)*(q-1).

Soit m la valeur en binaire du message en clair et c le cryptogramme. La fonction de chiffrement est, de façon simplifiée, $c = m^e \bmod n$ (si m est plus grand que n, il est séparé en morceaux de valeur inférieure à n et chaque morceau est chiffré séparément suivant cette formule). Du fait de la relation entre e et d, la fonction de déchiffrement correspondante est $m = c^d \bmod n$. La signature se fait de manière similaire, en inversant e et d, c'est-à-dire en chiffrant avec une clé privée et en déchiffrant avec la clé publique correspondante : $s = m^d \bmod n$ et $m = s^e \bmod n$.

Pour un cryptanalyste, retrouver la clé privée à partir de la clé publique nécessite de connaître (p-1)*(q-1) = p*q-p-q+1 = n+1-p-q, donc de connaître p et q. Pour cela, il doit factoriser le grand nombre n. Donc n doit être suffisamment grand pour que cela ne soit pas possible dans un temps raisonnable par rapport au niveau de sécurité requis.

Actuellement, la longueur du module n varie généralement de 512 à 2048 bits suivant les utilisations. Compte tenu de l'augmentation des vitesses de calcul des ordinateurs et des avancées mathématiques en matière de factorisation des grands nombres, la longueur minimale des clés doit augmenter au cours du temps.

Clés			
Clef publique	$n = p*q$, où p et q sont deux grands nombres premiers tenus secrets e telle que e et $(p-1)*(q-1)$ soient premiers entre eux		
Clef privée	$d \equiv e^{-1} \bmod (p-1*)(q-1)$		
Algorithme			
Chiffrement	$c = m^e \bmod n$	Déchiffrement	$m = c^d \bmod n$

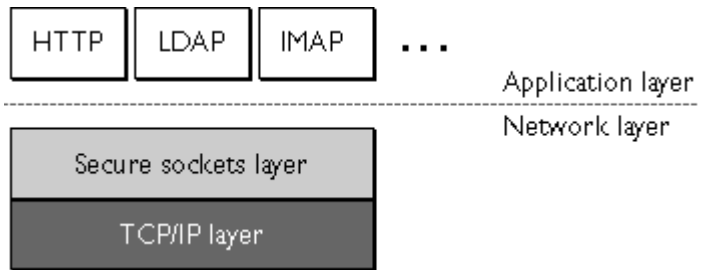
Le crypto-système RSA est très lent comparé aux autres crypto-systèmes symétriques, 1000 fois plus lent que le DES. Cette lenteur est montrée dans le tableau 2.2.

	512 bits	768 bits	1024 bits
Chiffrement	0,03 s	0,05 s	0,08 s
Déchiffrement	0,16 s	0,48 s	0,93 s
Signature	0,16 s	0,52 s	0,97 s
Vérification	0,02 s	0,07 s	0,08 s

Tableau 3-1 Vitesses du RSA pour différentes longueurs clés publiques (sur une station SPARC II) [12]

2. SSL [WEB 05]

Le protocole SSL (*Secure Sockets Layers*, que l'on pourrait traduire par « *couche de transport sécurisé* »), est un procédé développé par Netscape en collaboration avec *Mastercard*, *Bank of America*, *MCI* et *Silicon Graphics*, ayant pour but de sécuriser les transactions effectuées sur Internet. De nombreux sites de commerces de nos jours sont sécurisés avec SSL (afin de communiquer sûrement avec leurs clients et d'obtenir le paiement de leurs ventes). Ce système repose à la fois sur les algorithmes à clé publique, sur les algorithmes à clé privée et sur les certificats électroniques afin de garantir au maximum la sécurité de la transmission de données avec un tel site. Cependant un utilisateur quelconque ignore en principe totalement qu'il utilise SSL, car celui-ci agit de manière transparente. SSL est indépendant des protocoles de communications, il agit directement entre la gestion des commandes et la gestion du transport des données (il agit comme une couche supplémentaire de protection).



De cette façon un utilisateur se connectant à un site de commerce sécurisé via un navigateur Internet enverra des données chiffrées (code de Carte Bleue..) par SSL sans même s'en préoccuper. Seule l'apparition d'un petit cadenas s'affichant dans le navigateur et l'url commençant par https:// (le «s» signifiant secured) pourront permettre à un utilisateur averti de se rendre compte de l'intervention de SSL dans un tel échange.

L'utilisation de SSL est donc assez simple étant donné qu'elle se fait seule : son fonctionnement l'est aussi. La transaction par protocole SSL est en fait basée sur un échange de clé entre un client et un serveur. La transaction sécurisée se fait selon un schéma bien défini. Pour commencer, le client se connecte à un site de commerce (grâce à un navigateur utilisant SSL) en lui demandant de s'identifier (ce qui évitera de nombreux problèmes comme nous le verrons dans le titre concernant les attaques sur SSL). Le client envoie aussi une liste des cryptosystèmes qu'il connaît. Ensuite le serveur s'identifie en envoyant un certificat d'authentification contenant la clé publique du serveur ainsi que le nom du cryptosystème qui lui convient (souvent celui dont la longueur de clé est la plus longue).

A la réception, le client doit alors vérifier la validité du certificat envoyé par le serveur. Il choisi ensuite une clé secrète (de manière aléatoire) et l'envoie au serveur sous forme cryptée grâce à la clé publique du serveur (c'est la que RSA intervient). Cette nouvelle clé est appelée clé de session. Le serveur est enfin capable de déchiffrer le reste des transactions par le biais de la clé de session. Le serveur ainsi que le client sont donc tous deux en possession d'une clé commune, permettant alors la confidentialité des données échangées. Une dernière authentification est possible, celle du client. Elle permettrait encore une plus grande sécurité, mais elle est en fait très rarement utilisée dans les utilisations courantes de SSL.

L'utilisation du système SSL ne cesse de s'accroître. En outre il semble important de rappeler que SSL est indépendant du protocole utilisé (couche supplémentaire), c'est-à-dire qu'il peut non seulement sécuriser des transactions effectuées sur le Web mais aussi des connexions par FTP, POP, TELNET, IMAP, SMTP,…etc.

III.3.2.3. Les avantages et inconvénients du cryptage asymétrique

1. Les avantages :

- l'élimination de la problématique de la transmission de clé

- la possibilité d'utiliser la signature électronique

- l'impossibilité de décrypter le message dans le cas de son interception par une personne non autorisée.

2. Les inconvénients :

- le temps d'exécution : plus lent que le cryptage symétrique

- le danger des attaques par substitution des clés (d'où la nécessité de valider les émetteurs des clés).

III.3.3. Le chiffrement hybride [6]

Comme on l'a vu, lors d'un chiffrement asymétrique, chaque utilisateur génère une clé privée, et sa clé publique est diffusée. Cela permet de résoudre les problèmes de gestion des clés dans les réseaux de taille importante. Le principal problème de l'asymétrie par rapport à la symétrie va être le problème du débit, en effet les mécanismes mis en œuvre sont moins performants. C'est donc à partir de ce constat qu'est né le chiffrement hybride. Ce principe fait appel aux deux branches de la cryptographie.

- ✓ **Fonctionnement de PGP**

PGP est une combinaison des meilleures fonctionnalités de la cryptographie de clé publique et de la cryptographie conventionnelle. PGP est un système de cryptographie hybride.

Lorsqu'un utilisateur crypte du texte en clair avec PGP, ces données sont d'abord compressées. Cette compression des données permet de réduire le temps de transmission par

modem, d'économiser l'espace disque et, surtout, de renforcer la sécurité cryptographique. La plupart des cryptanalyses exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement. La compression réduit ces modèles dans le texte en clair, améliorant par conséquent considérablement la résistance à la cryptanalyse. Toutefois, la compression est impossible sur les fichiers de taille insuffisante ou supportant mal ce processus.

PGP crée ensuite une *clé de session* qui est une clé secrète à usage unique. Cette clé correspond à un nombre aléatoire, généré par les déplacements aléatoires de votre souris et les séquences de frappes de touches. Pour crypter le texte en clair, cette clé de session utilise un algorithme de cryptage conventionnel rapide et sécurisé. Une fois les données codées, la clé de session est cryptée vers la clé publique du destinataire. Cette clé de session cryptée par clé publique est transmise avec le texte chiffré au destinataire.

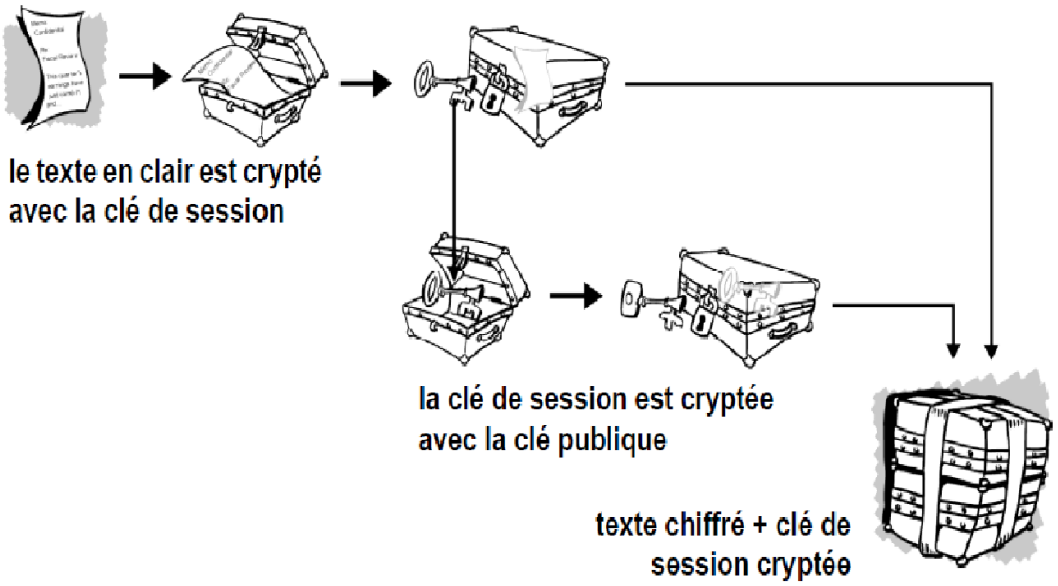


Figure 3-4 Fonctionnement du cryptage PGP.

Le processus de décryptage est inverse. La copie de PGP du destinataire utilise sa clé privée pour récupérer la clé de session temporaire qui permettra ensuite de décrypter le texte crypté de manière conventionnelle.

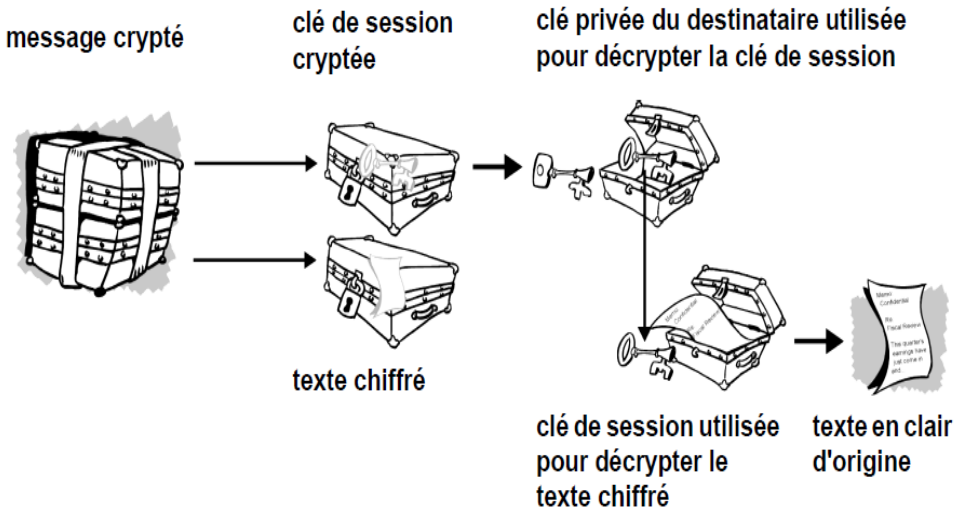


Figure 3-5 Fonctionnement du décryptage PGP.

Ces deux méthodes de cryptage associent la facilité d'utilisation du cryptage de clé publique à la vitesse du cryptage conventionnel. Le cryptage conventionnel est environ 1 000 fois plus rapide que le cryptage de clé publique. De plus, le cryptage de clé publique résout non seulement le problème de la distribution des clés, mais également de la transmission des données. Utilisées conjointement, ces deux méthodes améliorent la performance et la distribution des clés, sans pour autant compromettre la sécurité.

III.3.4. Cryptographie quantique [WEB 04]

III.3.4.1. Présentation

Un des procédés cryptographiques qui a révolutionné la fin du XXème siècle est la cryptographie quantique qui se base sur des propriétés de la physique quantique afin d'atteindre des niveaux de sécurité inaccessibles pour des méthodes classiques. Longtemps considérée comme une méthode quasiment invulnérable, ce procédé est surtout utilisé pour chiffrer les clés lors d'une communication entre deux personnes distantes. Jusqu'à maintenant le moyen de transmettre des clés secrètes entre deux personnes distantes afin que celles-ci puissent communiquer, était la valise diplomatique.

III.3.4.2. Fonctionnement et particularités

Afin de bien comprendre le fonctionnement de la cryptographie quantique, posons quelques bases. Pour cela, introduisons 3 personnages : Alice et Bob qui communiquent entre eux, et Eve qui les espionne. Voici ce dont disposent nos deux interlocuteurs :

- des objets quantiques, comme des impulsions lumineuses: des photons.
- un canal quantique qui va permettre aux objets quantiques de se déplacer d'un point à un autre, comme la fibre optique.
- un canal classique de communication, comme Internet.

Chacun des photons peut être polarisé sur des angles qui vont varier de 0° à 180°. Les protocoles de cryptographie quantique les plus connus sont le BB84 et le E90 mis au point par

les canadiens CH.Bennett et G.Brassard. Dans ces protocoles les champs magnétiques des photons prennent les directions suivantes: 0°, 45°, 90° et 135° soit 4 positions possibles.

Figure 3-6 Illustration des 4 positions possibles

Pour les polarisations de 0° et 90° on parle de **polarisation rectiligne** et pour 45° et 135°, de **polarisation diagonale**.

Afin de détecter l'orientation du photon, le receveur va utiliser un filtre. Dans le cas d'un filtre orienté à 0°, si un photon orienté à 0° est transmis, il va traverser le filtre et donc être enregistré par le receveur qui dispose d'un détecteur. Maintenant si l'émetteur transmet un photon orienté à 45° ou 135°, le photon va traverser le filtre une fois sur deux. Il est donc possible de distinguer une polarisation de 0° d'une polarisation de 90° mais les polarisations 45° et 135° sont indifférenciables. C'est le protocole de sécurité que nous allons expliquer par la suite qui va permettre de les différencier. Il faut noter que dans le cas d'un filtre orienté à 45°, il laissera passer les photons orientés à 45°, stoppera ceux à 135° et posera le même problème pour les photons à 0° et 90°.

Nos deux personnages Alice et Bob peuvent maintenant commencer à s'échanger la clé secrète via le canal quantique. Alice va émettre une série de photons où ceux orientés à 0° et 45° représentent 0, et les autres 1. De l'autre côté, Bob va recevoir les photons et mesurer leur polarité avec un filtre rectiligne ou diagonal, et considérera 0 si le photon traverse le filtre, 1 sinon.

Lors de la mesure avec par exemple un filtre rectiligne, si un photon est orienté diagonalement, il passera une fois sur deux, donc la mesure de Bob pourra être faussée. C'est là qu'intervient le canal classique de communication entre les deux interlocuteurs. Bob va pouvoir indiquer à Alice le filtre qu'il a utilisé et Alice va alors pouvoir confirmer ou non la mesure de Bob. Les bits qui seront connus d'Alice et Bob constitueront la clé secrète.

Le canal classique va être primordial pour la sécurité du système. En effet, si Eve était en train d'écouter la transmission d'Alice vers Bob, elle effectuerait le même travail que Bob. Eve intercepte donc le photon, et afin que son action reste invisible, renvoie immédiatement un photon vers Bob en essayant de transmettre le même qu'Alice. Cependant il y a une chance sur deux pour que le photon renvoyé soit le même photon, donc Bob a maintenant une chance sur quatre d'avoir une mesure fausse. Alice est Bob vont donc sacrifier quelques-uns des bits communs afin de vérifier la sécurité :

1. Alice émet un photon à 45° de valeur en bit 0
2. Eve l'intercepte avec un filtre rectiligne, lit 1 et transmet un photon à 90°
3. Bob reçoit le photon avec un filtre diagonal, le photon passe, Bob note 1
4. Alice et Bob sacrifient la mesure, Bob annonce un filtre diagonal

Alice et Bob peuvent donc en conclurent qu'Eve les a espionnés.

Grâce à ce procédé, la cryptographie quantique demeure très sûre, bien que quelques expériences commencent à prouver le contraire, notamment certains travaux au sein de l'université des sciences et technologies de Norvège, où des chercheurs sont parvenus à trouver un moyen d'écouter une communication sans utiliser le canal direct de communication afin de ne pas laisser de traces. Les chercheurs sont allés directement aveugler la machine émettrice grâce à des flashs laser interceptant ainsi 4% des communications sécurisées.

III.4. Enigma

Comme nous l'avons cité avant nous allons détailler le fonctionnement de la machine Enigma.

III.4.1. Fonctionnement d'Enigma

Le codage Enigma effectué par la machine Enigma est à la fois simple et astucieux. Chaque lettre est remplacée par une autre, et la substitution opérée change d'une lettre à l'autre, un peu comme dans le chiffre de Vigenère. La machine Enigma est alimentée par une pile électrique. Quand on appuie sur une touche du clavier, un circuit électrique est fermé, et une lampe s'allume qui indique par quelle lettre on doit remplacer la lettre que l'on vient de frapper.

Concrètement, le circuit électrique est constitué de plusieurs éléments en chaîne :

- **le tableau de connexions :** il permet d'échanger des paires de l'alphabet, deux à deux, au moyen de fiches. Il y a 6 fiches qui permettent donc d'échanger 12 lettres. Un tableau de connections est donc une permutation très particulière où on a échangé au plus 6 paires. Par exemple, dans le tableau suivant (avec simplement 6 lettres), on a échangé A et C, D et F, tandis que B et E restent invariants.

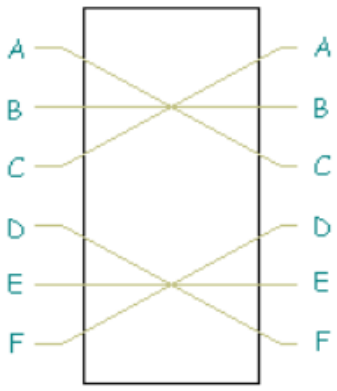


Figure 3-7 Tableau de connexions.

- **les rotors :** un rotor est également une permutation, mais cette fois quelconque. A chaque lettre en entrée correspond une autre lettre.

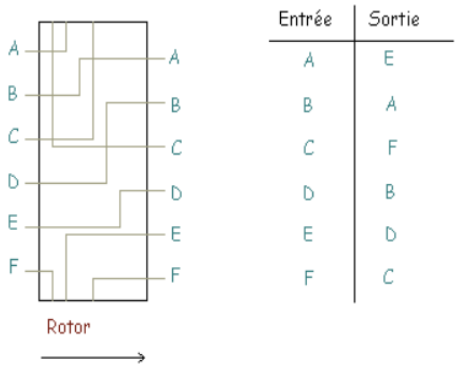
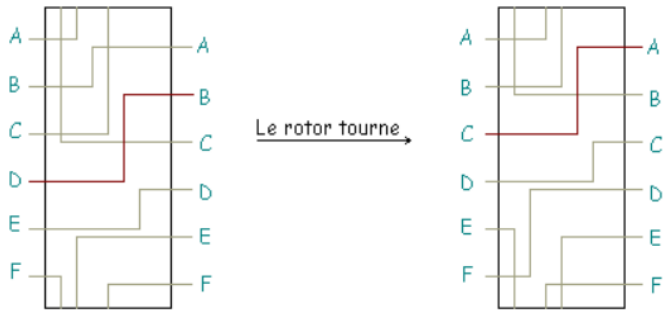


Figure 3-8 Image représentant un rotor.

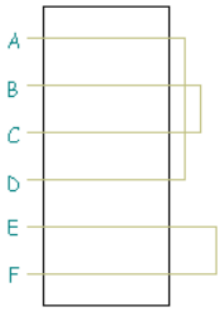
On peut composer les rotors, c'est-à-dire les mettre les uns à la suite des autres. La machine Enigma disposera, au gré de ses évolutions successives, de 3 à 6 rotors. Parmi ces rotors, seuls 3 sont utilisés pour le codage, et on a le choix de les placer dans l'ordre que l'on souhaite (ce qui constituera une partie de la clé).

Surtout, les rotors sont cylindriques, et ils peuvent tourner autour de leur axe. Ainsi, à chaque fois qu'on a tapé une lettre, le premier rotor tourne d'un cran, et la permutation qu'il engendre est changée. Observons ce changement sur la figure suivante : le rotor transforme initialement D en B. Lorsqu'il tourne d'un cran, cette liaison électrique D--->B se retrouve remontée en C--->A et, lorsque la prochaine lettre sera tapée, le rotor transformera cette fois D en C.



Chaque rotor possède donc 26 positions. A chaque fois qu'une lettre est tapée, le premier rotor tourne d'un cran. Après 26 lettres, il est revenu à sa position initiale, et le second rotor tourne alors d'un cran. On recommence à tourner le premier rotor, et ainsi de suite... Quand le second rotor a retrouvé sa position initiale, c'est le troisième rotor qui tourne d'un cran. Par ce procédé, on utilise donc $26^3 = 17\ 576$ alphabets différents, ce qui rend l'analyse de fréquence hors de portée.

- **Le réflecteur :** Pour utiliser une telle machine à chiffrer, une des difficultés pratiques vient du fait qu'il est indispensable de disposer de deux machines différentes : l'une pour chiffrer, l'autre qui correspond à la transformation inverse, pour déchiffrer. Pour pallier cet inconvénient, son concepteur, Arthur Scherbius, eut l'idée d'ajouter un réflecteur, similaire à un rotor fixe. Son rôle est de transformer la lettre obtenue après passage dans les trois rotors puis de faire subir au résultat la même transformation dans le sens inverse. Le résultat du chiffrement d'une lettre correspond ainsi à un circuit entre une lettre du clavier et une lettre du panneau lumineux.



A est permuté avec D, B est permuté avec C, et E avec F

Figure 3-9 Le réflecteur.

III.4.2. Une machine à chiffrer et à déchiffrer [WEB 06]

Résumons sur la machine simplifiée suivante (6 lettres, 3 rotors) comment est codée la lettre B :

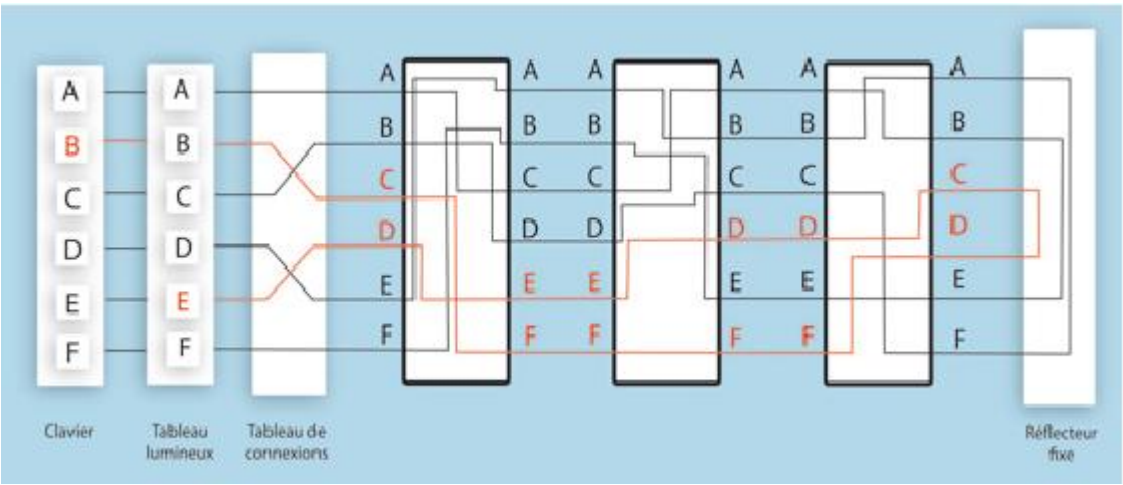


Figure 3-10 Machine à trois rotors avec réflecteur et tableau de connexions.

- on traverse le tableau de connexions : on obtient C.
- on traverse les 3 rotors : on obtient successivement F, F et D.
- on traverse le réflecteur où on obtient C, puis on renvoie dans les rotors pour obtenir D, E, D et finalement un E s'affiche sur le tableau lumineux après le passage au tableau de connexions.

Comme c'est ce même circuit qui est utilisé pour déchiffrer, on voit que si la lettre « b » est chiffrée en « e », alors pour une position identique des rotors, la lettre « e » sera chiffrée en « b ». Grâce à cette astuce, une seule machine suffit à chiffrer et à déchiffrer, car il s'agit exactement de la même transformation. Les rotors doivent avoir la même position de départ pour le chiffrement et pour le déchiffrement ; cette position fait partie du secret partagé par l'expéditeur et le destinataire.

Avec l'étrange conception d'un hasard visiblement bien ordonné, l'armée allemande poussa cependant le raffinement jusqu'à choisir un réflecteur qui ne transformait aucune lettre en elle-même, assurant ainsi qu'une lettre ne pouvait jamais être son propre chiffré. Loin d'être une garantie de sécurité supplémentaire, cette propriété fut amplement exploitée par Alan Turing dans sa cryptanalyse.

- **Nombre de clés possibles**

Si l'on suppose que le câblage des rotors est connu de l'attaquant (ce qui fut le cas, car les plans de la machine furent communiqués aux Français par un espion dès le début des années trente), la sécurité d'Enigma repose sur un triple secret :

1. La position de départ des trois rotors offre 26^3, soit 17 576 possibilités.
2. le choix de ces trois rotors parmi cinq rotors différents multiplie ce nombre de clés possibles par 10.
3. Enfin l'ordre des rotors le multiplie par factorielle de 3, soit $3 ! = 1 \times 2 \times 3 = 6$.

Le nombre de clés possibles était donc $26^3 \times 10 \times 6$, ce qui fait plus d'un million de clés à essayer.

Cependant, c'est pour augmenter encore ce nombre qu'Arthur Scherbius avait ajouté le tableau de connexions dont le rôle est de transposer 10 paires de lettres immédiatement en sortie du clavier et avant affichage sur le tableau lumineux. Le choix de ces dix couples faisait partie de la clé secrète, multipliant ainsi le nombre de possibilités par 10^{14}, ce qui conduit à un nombre total de presque 10^{20} clés, c'est-à-dire des centaines de milliards de milliards, ce qui est énorme pour l'époque!

III.5. Conclusion

Des notions de cryptographie nécessaire pour la compréhension des cryptosystèmes de chiffrement ont été présentées dans ce chapitre. Nous avons parlé des deux grandes familles de méthodes de chiffrement et leurs modes de fonctionnement, leurs applications et les notions qui vont avec la cryptographie de manière générale. Nous avons abordé quelques exemples d'algorithmes symétriques et asymétriques tels que DES et RSA, et quelques applications comme le chiffrement par flot ou encore le chiffrement hybride.

Et enfin, nous avons fait une description de la machine Enigma et expliqué le principe de son fonctionnement.

Avant de clore ce chapitre, voici un récapitulatif des différentes méthodes et leurs complexités :

	Le nom de la méthode	**Sa complexité**
	Le chiffrement par flots	• propagation d'erreurs (problème de synchronisation) • sécurité difficile à atteindre (pas de preuve)
	Le chiffrement par blocs	• un même bloc de texte en clair sera toujours chiffré en un même bloc de texte chiffré • facilité de manipulation des messages chiffrés en retirant, répétant ou interchangeant les blocs

Cryptographie symétrique		• l'amplification d'erreur : si un bit du texte chiffré est modifié pendant le transfert, tout le bloc de texte en clair correspondant sera faux [11]
	Enigma	• L'une des failles de la machine Enigma est que jamais la lettre A ne sera codée par un A. Cela élimine un certain nombre de cas à inspecter. • Une des autres faiblesses dépend plutôt du protocole utilisé par les allemands : certains opérateurs (par exemple, ceux qui informaient de la météo) prenaient peu de précautions et commençaient toujours leurs messages par les mêmes mots (typiquement "Mon général..."). Les anglais connaissaient ainsi pour une partie du message à la fois le texte clair et le texte codé, ce qui aide à retrouver la clé. Et comme c'est la même clé qui sert pour toutes les machines Enigma de l'armée allemande pour un jour donné, une erreur de protocole dans un message peut compromettre la sécurité de tous les autres!
	DES	• Attaque en 2^{56} en temps et 2^{56} couples (chiffré, clé) en mémoire. • Taille de clé (recherche exhaustive en 2^{56} est réaliste) → utilisation du Triple-DES • Taille du bloc (attaques avec 2^{32} messages) • Cryptanalyse linéaire et différentielle
	AES	• le décryptage est plus difficile à implanter en "Smart Card" • code et tables différents pour l'encryptage et le décryptage • dans une réalisation en matériel, il y a peu de réutilisation • des circuits d'encryptage pour

		effectuer le décryptage
Cryptographie asymétrique	RSA	• Algorithme de Shor dès que l'on aura un ordinateur quantique • Difficulté d'implémentation : comment choisir des premiers de grande taille • On peut tenter de s'attaquer non plus à la factorisation mais à la façon dont sont générés p et q [13]
	SSL	• absence de vérification de l'identité du client • l'efficacité de la protection en cours de transmission dépend essentiellement de la clé de cryptage retenue [WEB 05]
Cryptographie hybride	PGP	• risque de falsification des clés publiques • Suppression de fichiers incomplète • Virus et chevaux de Troie [6]
Cryptographie quantique	Cryptographie quantique	• la difficulté à transmettre sur de longues distances des particules et leurs états quantiques, donc des particules intriquées. Des photons véhiculés par fibres optiques ne pourront, dans l'état des connaissances, que franchir une petite centaine de kilomètres

Tableau 3-11 Tableau représentant différentes méthodes de cryptographie et leurs complexités

Chapitre IV
Analyse et conception

IV.1. Introduction

Apres avoir vu, dans les chapitres précédents les différents concepts nécessaires à l'accomplissement de notre travail, nous passons maintenant à la partie conception.

Aujourd'hui, le développement d'un logiciel s'appuie sur une prospective orientée objet. Tout simplement parce qu'elle a démontré son efficacité lors de la construction de systèmes dans les domaines les plus divers et qu'elle englobe toutes les dimensions de tous les degrés de complexité.

IV.2. Présentation du projet

IV.2.1. Description

Ce projet consiste à la réalisation d'un logiciel de cryptage, qui assure le chiffrement et le déchiffrement de données, en proposant un algorithme de cryptage symétrique, qui repose sur des formules mathématiques appliquées sur la codification octale des caractères qui constituent le fichier(ou bien le texte) à crypter.

L'objectif principal de ce projet, est de sécuriser les données à stocker sur une machine ou bien à transmettre sur un réseau, pour faire face à ceux qui ont un but de falsifier ou de détruire ces données en utilisant la technique de cryptographie vue précédemment.

L'application à réaliser doit garantir les critères de la sécurité informatique tel que :

- La confidentialité des données cryptées.
- L'intégrité des informations décryptées.
- La disponibilité de l'information (les données peuvent être cryptées ou décryptées par le logiciel de façon indépendante du moment et de la machine sur laquelle il est utilisé).

En plus de ces critères, notre application doit offrir une interface conviviale et simple d'utilisation.

IV.2.2. Principe de fonctionnement

La cryptographie à algorithmes symétriques utilise généralement la même clé (formule de cryptage) pour les processus de chiffrement et de déchiffrement, cette clé est appelée secrète car toute la sécurité de l'ensemble est directement liée au fait que cette clé reste secrète, car on considère l'algorithme de chiffrement connu de tous.

L'algorithme de cryptage que nous proposons repose sur la modification et la conversion qui sont des méthodes de cryptage très utilisées, et leur mixage permet d'atteindre un degré de sécurité largement élevé. Il est a noté que ce même principe est utilisé pour l'algorithme de chiffrement et celui de déchiffrement.

L'algorithme utilisé reçoit en entré un bloc de données clair ou chiffré de taille variable, il le divise en octets, et traites ces derniers indépendamment. Ce traitement consiste à appliquer la clé de cryptage (formule mathématique générée aléatoirement) sur chaque octet, il ajoute le résultat de cette formule à la valeur de l'octet lors du chiffrement, et l'enlève lors du déchiffrement.

A la fin de ce traitement, le bloc de données en sortie est considéré comme étant crypté en cas de chiffrement, et décrypté en cas de déchiffrement.

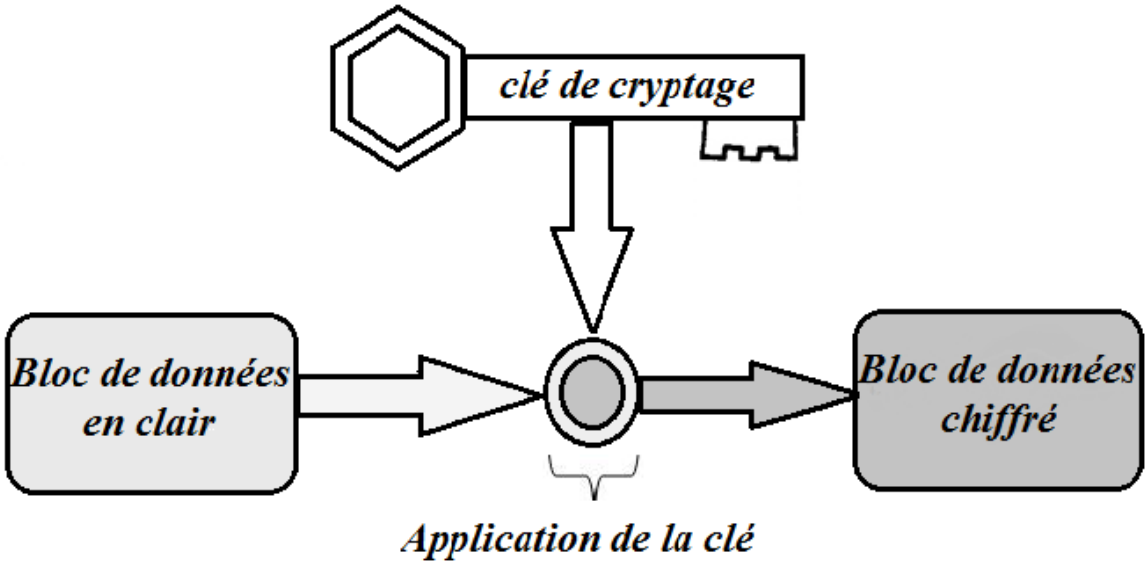


Figure 4-1 Fonctionnement de l'algorithme.

IV.2.3. Méthodes de traitement des données

Notre algorithme de cryptage permet de crypter et de décrypter tout type de fichier (quelque soit l'extension du fichier en entrée) et aussi tout texte saisi qui peut faire objet d'un message à envoyer une fois crypté.

Notons que l'algorithme adopté par notre application reste le même dans les deux cas que nous venons de citer, et que seulement la façon de traitement des données reçus change, et ce, selon le format de ces données.

IV.2.3.1. Traitement de fichiers

Le principe de ce fonctionnement réside dans le fait que l'extension du fichier en clair reçu en entrée sera extraite de ce dernier, puis on applique l'opération de cryptage sur ce fichier.

A la fin de cette opération, on ajoute l'extension extraite au début, dans le fichier crypté portant l'extension (.crypt).

La figure 4-2 suivante illustre ce fonctionnement :

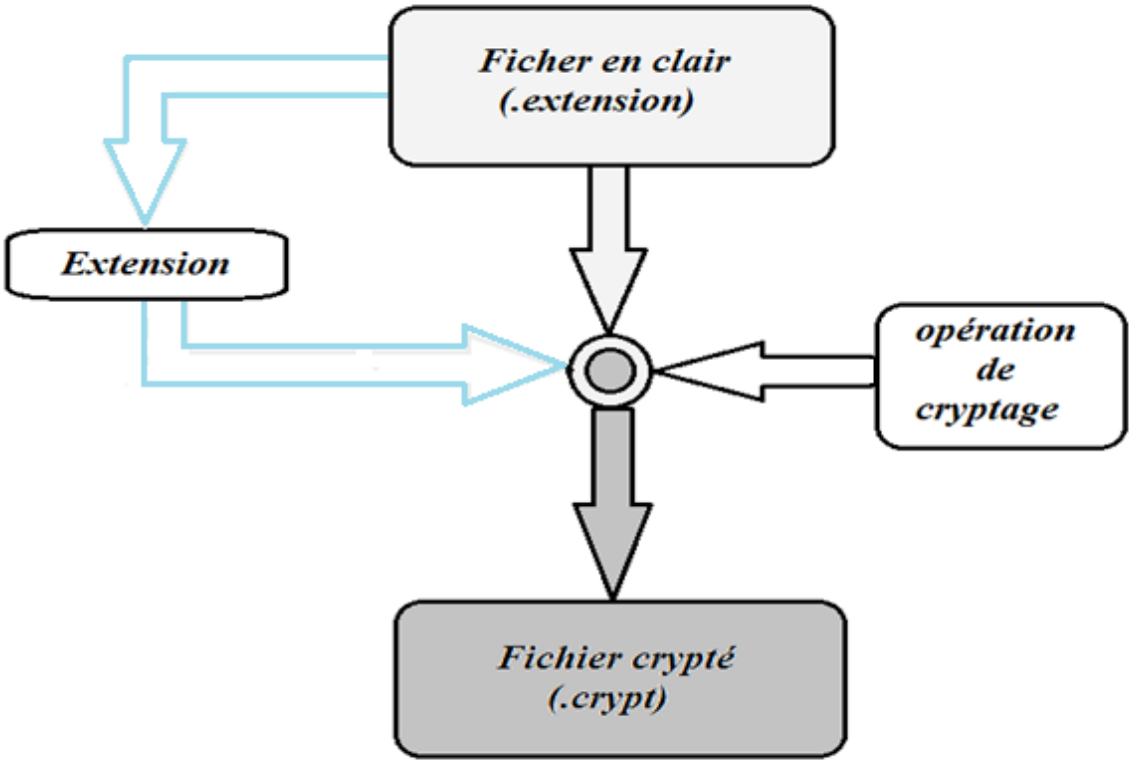


Figure 4-2 Cryptage de fichiers.

Pour décrypter un fichier crypté, on lui applique l'opération de décryptage, et on extrait son extension.

A la fin de cette opération, on rajoute l'extension extraite au nom du fichier décrypté.

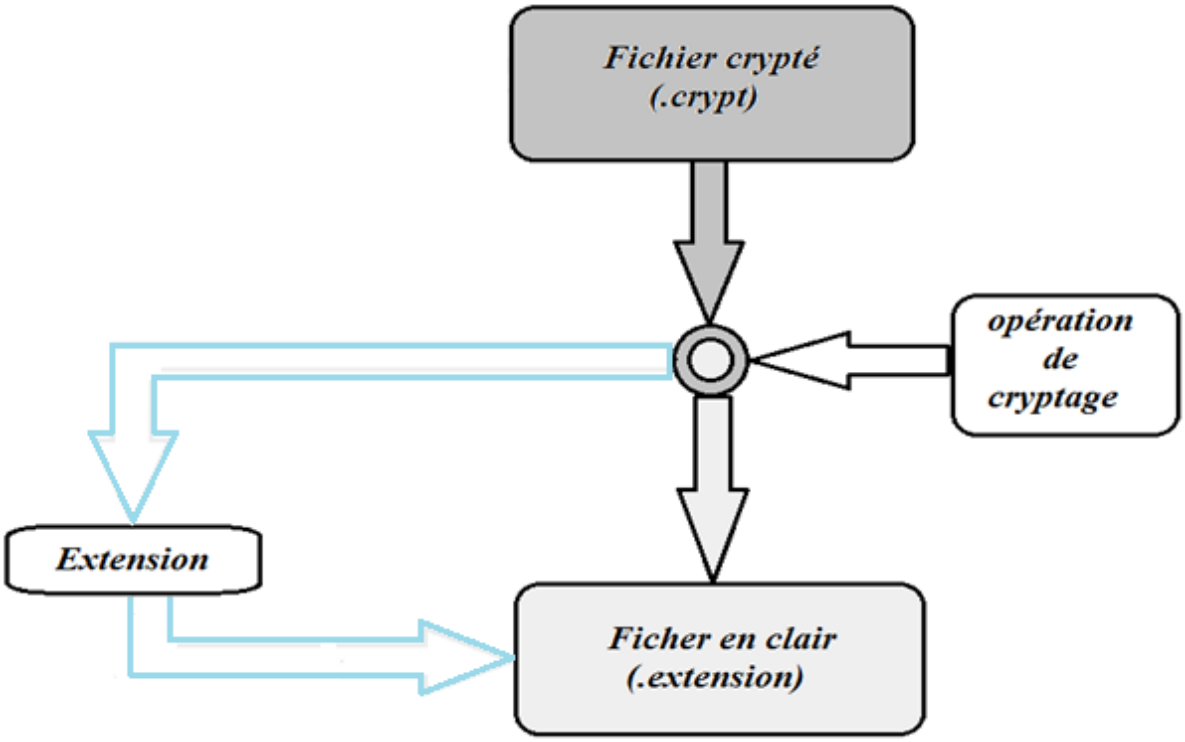


Figure 4-3 Décryptage de fichiers.

IV.2.3.2. Traitement de textes

Pour crypter un texte saisi par un utilisateur, on lui applique la fonction de cryptage, et en sortie on aura le texte équivalent crypté.

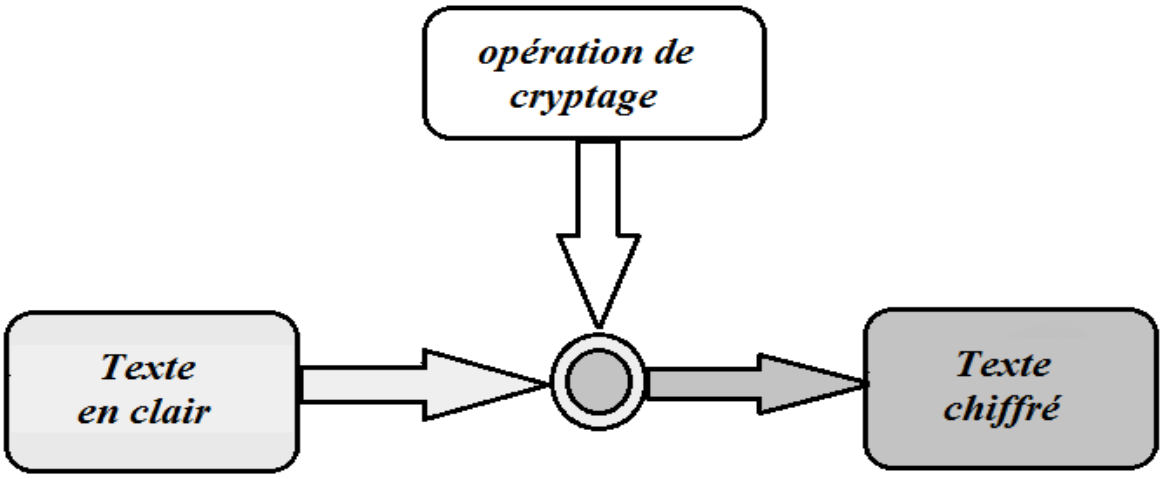


Figure 4-4 Cryptage de textes.

Le processus de décryptage se fait inversement, en entrée, on aura un texte crypté, et en appliquant sur ce dernier la fonction décryptage, le résultat sera un texte identique à celui en clair.

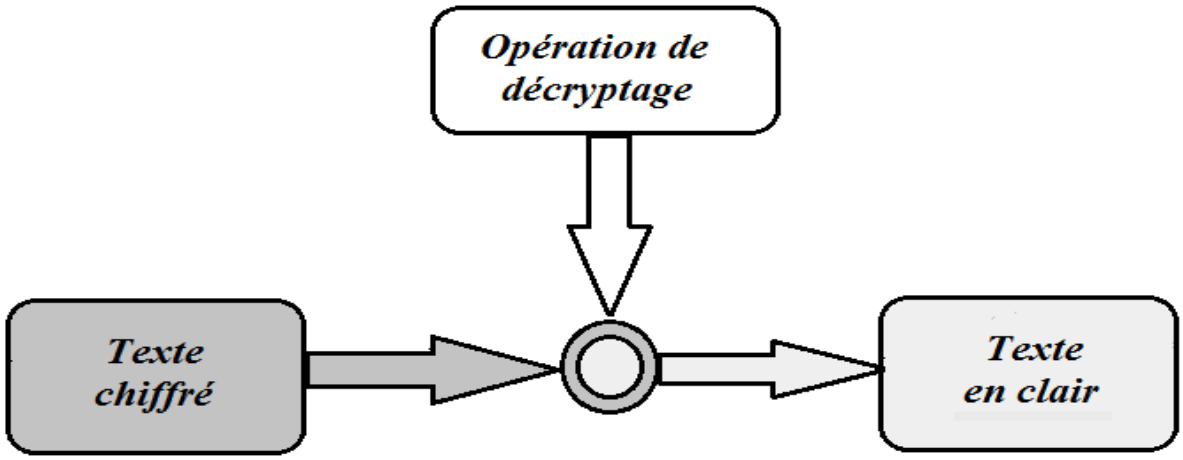


Figure 4-5 Décryptage de textes.

IV.3. Description de la clé de cryptage

Comme tout algorithme de cryptage, notre algorithme repose sur une clé secrète. Et puisque cet algorithme est symétrique, la clé de cryptage et celle de décryptage doivent être identiques.

La clé que nous avons proposée est une formule mathématique générée aléatoirement, ce qui nous donne la possibilité d'avoir un nombre de clés différentes pour autant d'opérations de cryptage effectuées.

Etant donné qu'en cryptographie, la taille de la clé utilisée est importante en termes de sécurité, du fait que, plus la taille de la clé est importante, plus le système cryptographique est considéré comme sûr. On a choisi une taille de **13** bits pour notre clé.

- ✓ **La formule de cryptage**

Soient :

- **x** : un nombre fractionnaire aléatoire dont sa valeur est comprise entre **]0,0** et **1,0[**.
- ***f*(x)** : la formule de cryptage.
- **n** : un nombre positif inférieur ou égale à **7936**.
- **P** : un nombre entier, tel que **(x*p) + n ≤ 7936**

  La formule de la clé est la suivante : « ***f*(x) = la valeur entière de [(x*p) + n]**»

Explications :

Comme la taille de notre clé est de **13** bits, la valeur maximale est $\mathbf{2^{13}-1}$ qui est égale à **8191**.

Et puisque la valeur d'un octet non signé est comprise entre **0** et **255,** donc sa plus grande valeur possible est égale à **255**.

Afin d'éviter tout dépassement de capacité (la taille du résultat du cryptage de chaque octet doit être représentée sur 13 bits) le résultat **(*f*(x) + 255)** doit être inferieur ou égale à $\mathbf{(2^{13}-1)}$, ce qui nous donne : ***f*(x) ≤ 7936.**

IV.4. Description de l'algorithme

L'activité de conception consiste à enrichir la description du logiciel de détails d'implémentation afin d'aboutir à une description très proche d'un programme, pour cela nous allons détailler le fonctionnement des deux opérations: cryptage et décryptage.

IV.4.1. Algorithme de cryptage

La figure suivante, illustre le processus de chiffrement adopté par notre application dont les étapes sont décrites en détail dans le paragraphe qui suit la figure :

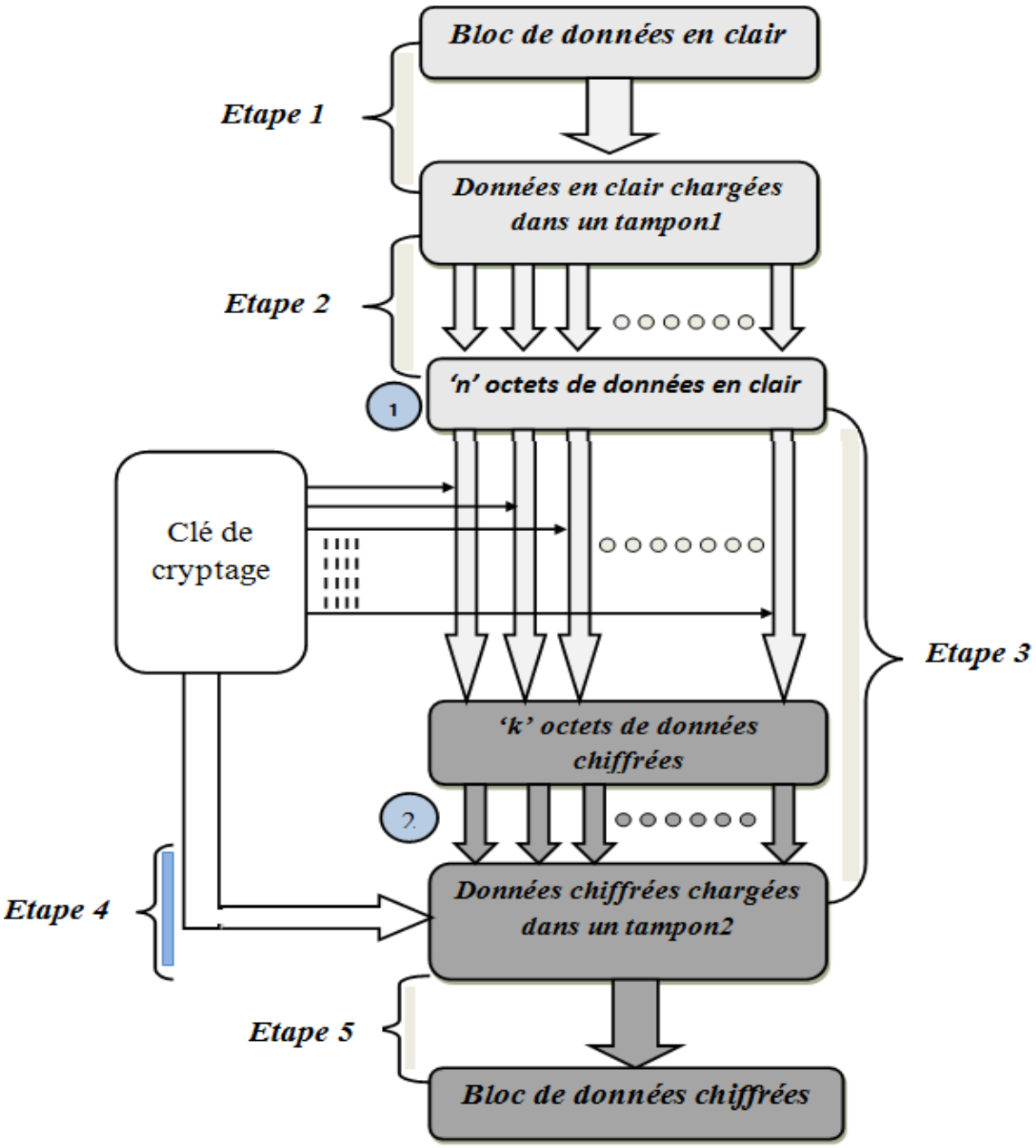


Figure 4-6 Les étapes de chiffrement.

Pour crypter des données, notre algorithme de cryptage suit 5 étapes, qui sont :

- **Etape 1 : Chargement des données dans le tampon mémoire :**

Apres sélection des données à chiffrer¸ ces dernières seront chargées dans un tampon mémoire, appelé tampon d'entrée, pour permettre au processus de chiffrement de modifier (crypter) ces données.

- **Etape 2 : Extraction des données Octet par Octet :**

Du moment que le cryptage se fait au niveau d'Octet, l'algorithme extrait à partir du tampon mémoire les données sous forme d'Octets, en utilisant une fonction qui lit un Octet mémoire à partir d'un Buffer.

- **Etape 3 : Application de la clé de cryptage :**

Après séparation des octets, On applique sur chacun de ces derniers les procédures suivantes :

1. On ajoute la valeur de la clé de cryptage à la valeur de l'octet sélectionné.
2. On ajoute le résultat du chiffrement à un tampon mémoire appelé tampon de sortie.

A la fin de cette étape, on aura des données cryptées correspondantes aux données en entrée.

- **Etape 4 : Rangement de la clé de cryptage :**

Dès que la 3eme étape se termine, et qu'on a toutes les données cryptées complètes dans le tampon de sortie, on rajoute à la fin de ces dernières la valeur de la clé, pour pouvoir effectuer l'opération de décryptage par récupération de cette valeur.

- **Etape 5 : Chargement des données du tampon de sortie :**

C'est la dernière étape du processus, elle consiste à récupérer les données cryptées du tampon de sortie, et les transforme en un bloc de données cryptées.

Nous venons de décrire toutes les étapes du processus de cryptage, et comme notre application adopte un système de cryptage symétrique, la clé de chiffrement doit être conservée secrètement afin de pouvoir décrypter les données en sécurité.

Pour des fins de disponibilité, nous avons rangé cette clé dans le même bloc que les données cryptées, et cela offre une certaine indépendance au niveau du décryptage, et même la possibilité d'appliquer une clé différente a chaque opération de cryptage.

IV.4.2. Algorithme de décryptage

Ayant en entrée un bloc de données crypté, le processus de décryptage repose sur les mêmes étapes décrites précédemment sauf que l'ordre d'exécution est inversé. La figure suivante illustre le fonctionnement de ce processus :

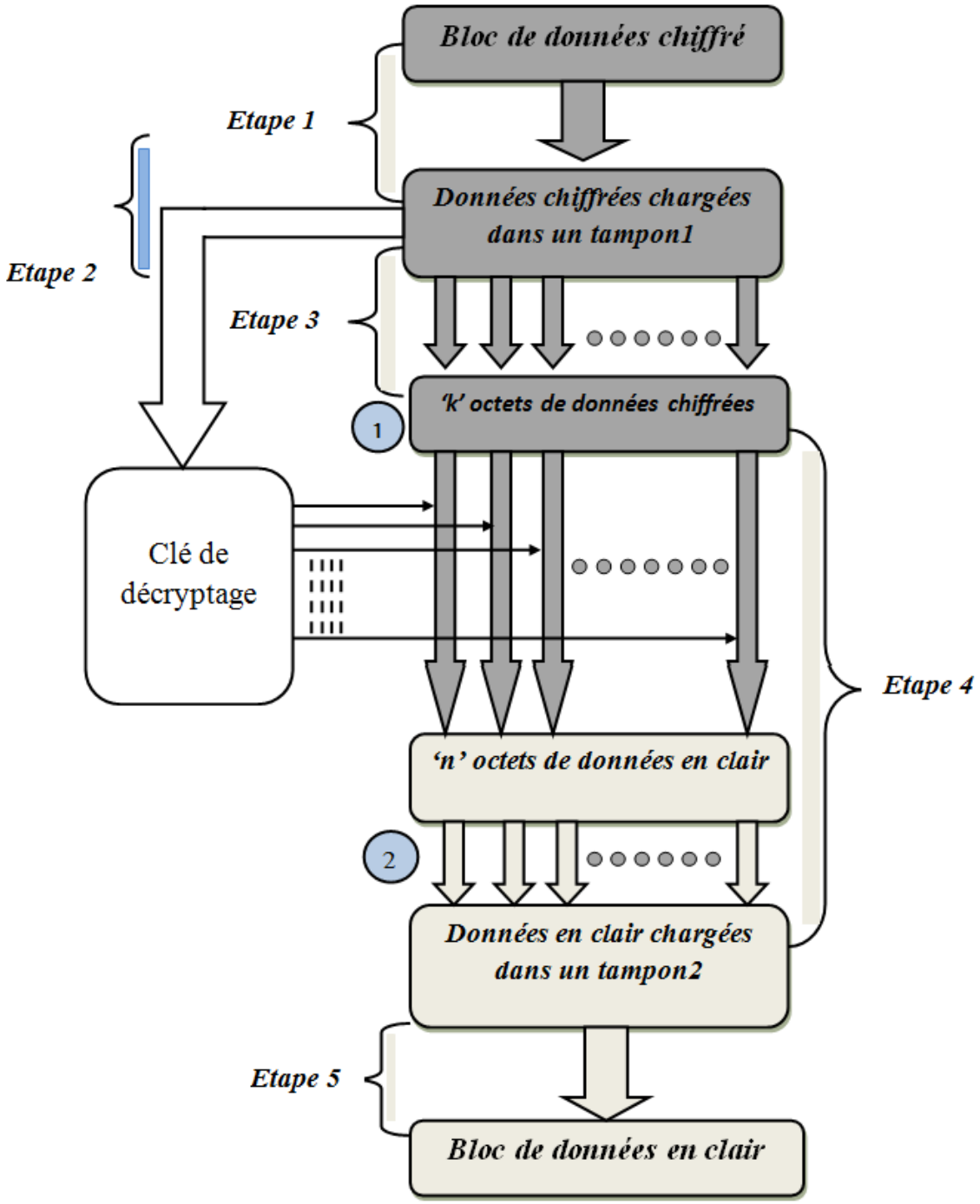


Figure 4-7 Les étapes de déchiffrement.

- **Etape 1 : Chargement des données dans le tampon mémoire :**

Après sélection des données à déchiffrer¸ ces dernières seront chargées dans un tampon mémoire, appelé tampon d'entrée, pour permettre au processus de chiffrement de modifier (décrypter) ces données.

- **Etape 2 : Extraction de la clé de décryptage :**

Le processus récupère la clé de déchiffrement à partir d'un emplacement spécifié dans le tampon d'entrée, cette clé est la valeur rangée avec les données cryptées.

- **Etape 3 : Extraction des données paquet par paquet :**

Une fois la clé extraite, l'algorithme récupère à partir du tampon mémoire les données sous forme de paquets de caractères, la taille de chaque paquet doit être égale à celle de la clé utilisée, en utilisant une fonction qui lit un flux de caractères à partir d'un Buffer.

- **Etape 4 : Application de la clé de décryptage :**

Après séparation des paquets, On applique sur chacun de ces derniers les procédures suivantes :

1. On soustrait la valeur de la clé de cryptage à la valeur du paquet sélectionné.
2. On ajoute le résultat du déchiffrement à un tampon mémoire appelé tampon de sortie.

A la fin de cette étape, on aura des données décryptées correspondantes aux données préalablement cryptées.

- **Etape 5 : Chargement des données du tampon de sortie :**

C'est la dernière étape du processus, elle consiste à récupérer les données décryptées du tampon de sortie, et les transforme en un bloc de données décryptées.

IV.5. Conclusion

Ce chapitre est d'une grande importance pour la suite du travail, du fait qu'il projette les notions théoriques vues dans notre étude, et qu'il traite de la conception du logiciel à réaliser, sans laquelle la réalisation ne pourra se faire.

Nous y avons d'abord décrit le fonctionnement général de notre algorithme, puis expliqué son fonctionnement détaillé en faisant appel à des schémas illustrant le déroulement de ces processus.

Donc à ce stade on est assez armé pour mettre sur pied notre application, ce qui va être l'objet du chapitre suivant toute en exposant l'environnement de développement.

Chapitre V
Implémentation et réalisation

V.1. Introduction

Après avoir présenté dans le chapitre précédent les différentes étapes de fonctionnement de notre algorithme de cryptage, dans le présent chapitre nous allons décrire son implémentation.

Tout au long de ce chapitre, nous allons, tout d'abord, commencer par la description de l'environnement de développement et d'implémentation de notre application, puis nous nous focaliserons sur la présentation de cette dernière, tout en illustrant les différentes interfaces qu'elle contient.

V.2. Environnement de développement

- ✓ **Matériel utilisé**

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

- Un microprocesseur Intel(R) Core™ i3
- Fréquence d'horloge 2.53 GHz
- RAM 2 GB DDR3
- Disque dure de 250 320 GB HDD

Nous avons développé notre application sous un système d'exploitation Microsoft Windows Seven.

V.3. Présentation du logiciel

SéTigma est un logiciel de cryptage et de décryptage de données, il offre deux types de cryptage/décryptage :

- **Cryptage/décryptage de fichiers :**

SéTigma permet à l'utilisateur de crypter des fichiers ayant n'importe qu'elle extension (.txt, .docx, .pdf, .jpg, png, .exe, .mp3, …), le fichier en résultat portera l'extension «.crypt».

Apres décryptage, le fichier portera son extension initiale.

- **Cryptage/décryptage de textes :**

Ce logiciel offre aussi la possibilité de crypter/décrypter un texte saisi directement par l'utilisateur.

V.3.1. Pourquoi SéTigma ?

Nous avons attribué ce nom à notre logiciel en s'inspirant de la célèbre machine de cryptographie «Enigma» qui a été utilisée par l'armée allemande durant la seconde guerre mondiale.

Il s'agit d'une machine mécanique à chiffrer et à déchiffrer qui allie à la fois les méthodes de substitution et de transposition.

Figure 5-1 La machine Enigma.

V.3.2. Présentation des interfaces de notre application

Nous allons maintenant présenter les différentes interfaces du logiciel ainsi que son fonctionnement.

V.3.2.1. Interface d'authentification

Le but de cette interface est d'authentifier l'utilisateur du logiciel pour offrir une certaine privatisation de l'utilisation de ce logiciel, et ainsi empêcher toute tentative d'utilisation illégale des personnes ne possédant pas le droit de l'utiliser.

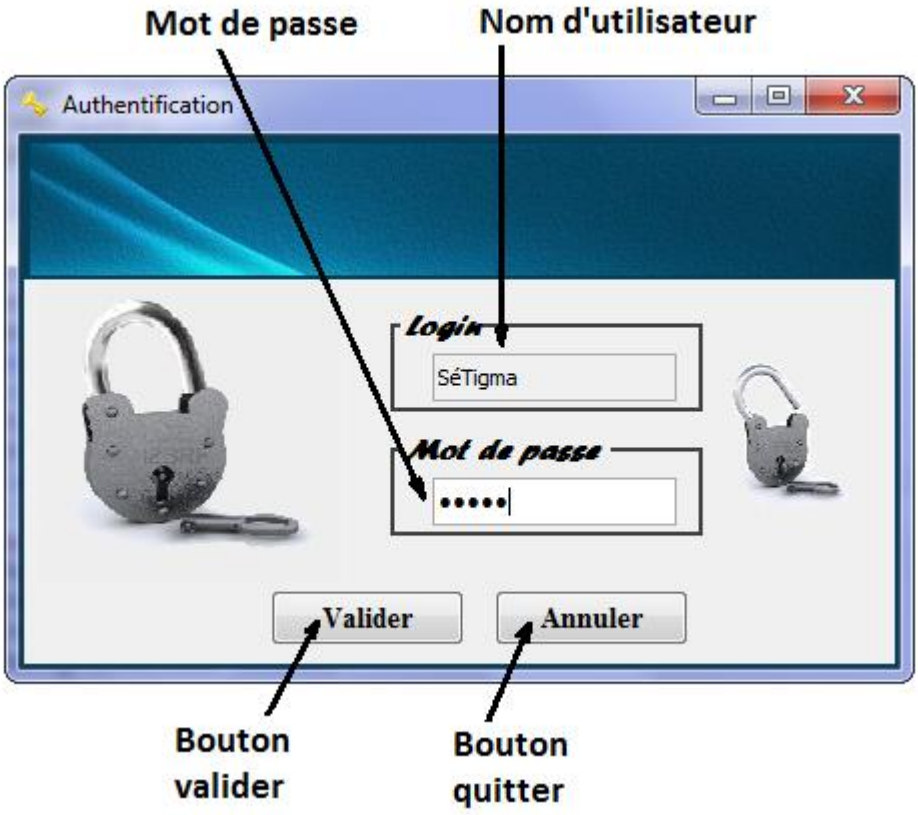


Figure 5-2 Interface Authentification.

V.3.2.2. Interface d'accueil

C'est l'interface principale de notre logiciel, elle est conçue de sorte qu'elle offre une simplicité d'utilisation. Cette interface permet l'accès aux différentes opérations du logiciel (cryptage, décryptage, changement de mot de passe,…).

En plus de la barre du menu principal, quatre boutons caractérisent cette interface :

1- Le bouton «Crypter» : permet à l'utilisateur de lancer une opération de cryptage.

2- Le bouton «Décryptage» : permet à l'utilisateur de lancer une opération de décryptage.

3- Le bouton «À propos» : permet le lancement de la fenêtre à propos.

4- Le bouton «Quitter» : pour quitter l'application.

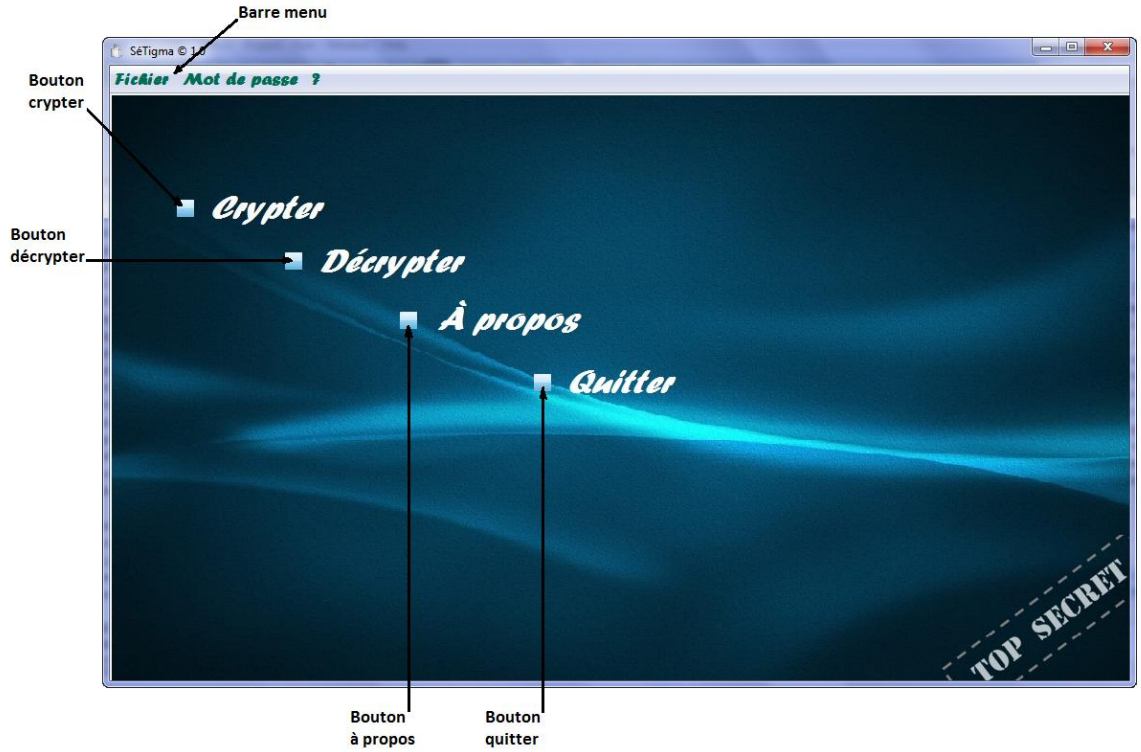


Figure 5-3 Interface accueil.

V.3.2.3. Interfaces de cryptage

Figure 5-4 Interface choix du type de cryptage.

Dans cette interface, l'utilisateur choisit le type de cryptage qui lui convient, notre application offre deux types qui sont :

- ✓ Cryptage de fichiers par sélection.
- ✓ Cryptage de texte.

Pour crypter des fichiers, l'utilisateur doit sélectionner le bouton «Cryptage de fichier par sélection», puis il clique sur le bouton «Valider».

Si l'utilisateur veut crypter un texte en le saisissant, il n'a qu'à sélectionner le bouton «Cryptage de texte par saisie», puis il clique sur «Valider».

Pour quitter l'interface cryptage, l'utilisateur clique sur le bouton «Quitter».

1. Cryptage de fichiers par sélection :

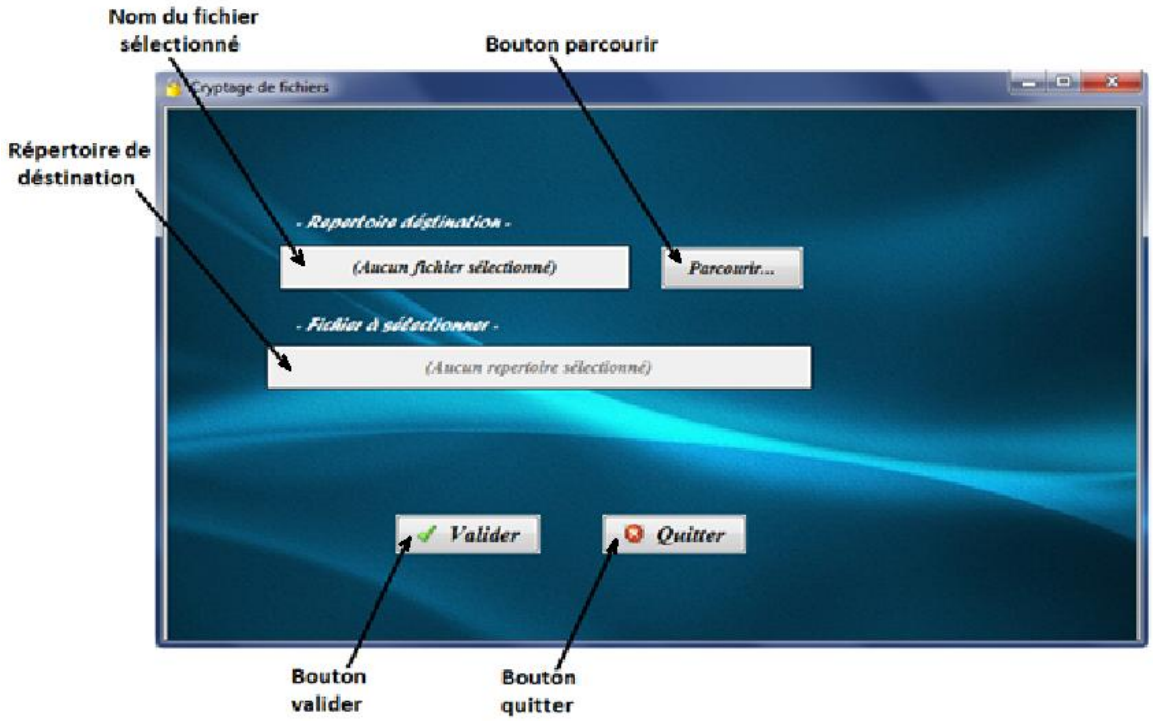


Figure 5-5 Interface cryptage de fichier.

Arrivé à ce stade, l'utilisateur sélectionne le fichier à crypter en cliquant sur le bouton «Parcourir…», qui lui permet de choisir un fichier à partir d'un répertoire. La Figure 5-7 suivante représente l'interface engendrée par ce bouton.

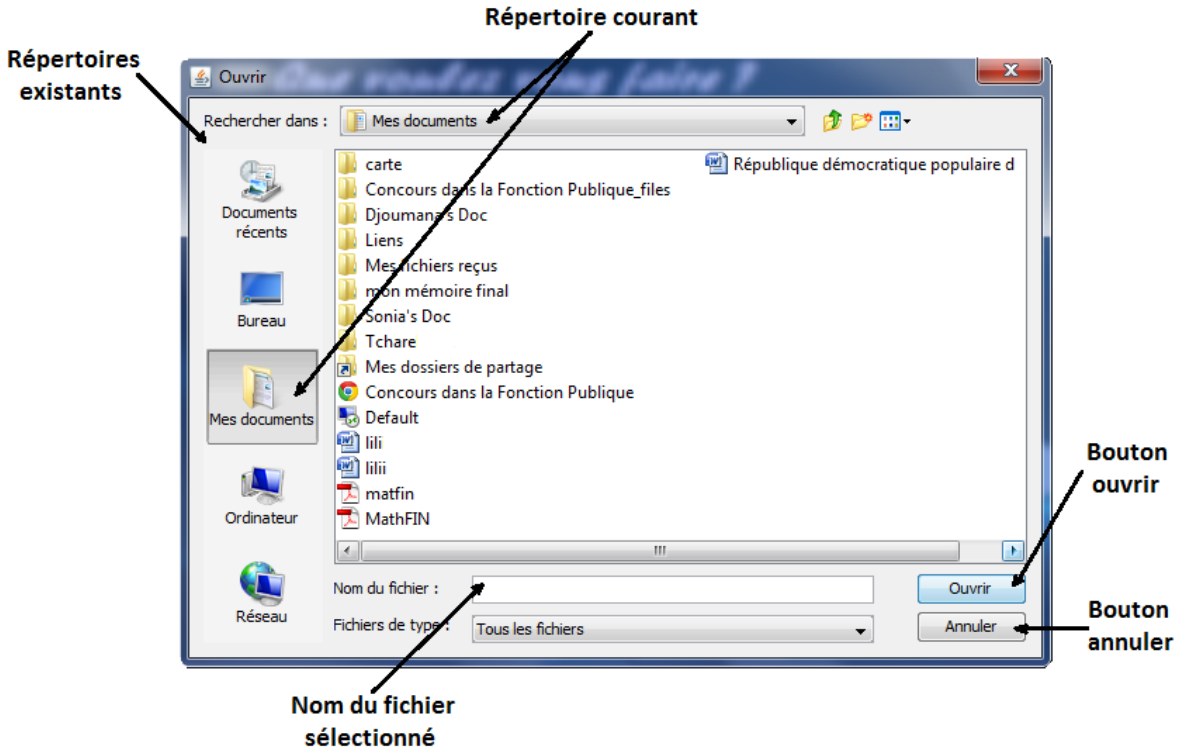


Figure 5-6 Interface pour choisir le fichier à crypter.

Après la sélection du fichier que l'on désire crypter, le nom du fichier apparaît dans le champ «Fichier à sélectionner» de l'interface «Cryptage de fichier», ensuite l'utilisateur passe à la dernière étape du cryptage en cliquant sur le bouton «Valider».

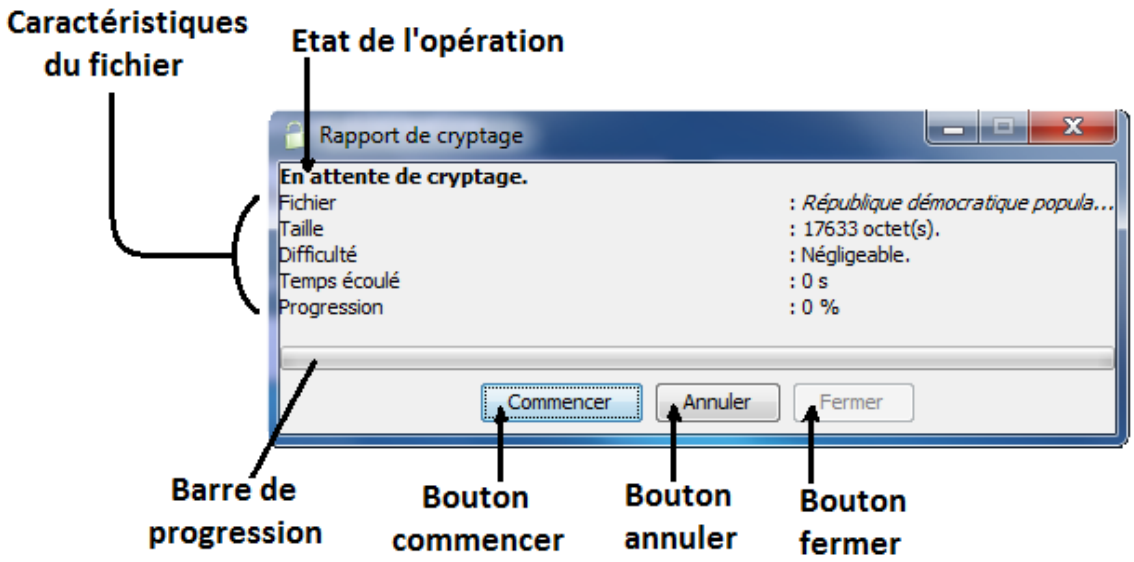


Figure 5-7 Interface rapport de cryptage.

Arrivant à cette étape, l'utilisateur aura toutes les informations concernant le fichier à crypter (Nom du fichier, sa taille, la difficulté de cryptage,…), il aura aussi à sa disposition trois bouton, dont deux activés a l'état initial, qui sont :

- Bouton «Commencer» : permet le déclanchement de l'opération de cryptage.
- Bouton «Annuler» : permet d'annuler l'opération, et retourner à l'étape précédente.

A la fin de l'opération, le bouton «Fermer» s'active, permettant à l'utilisateur de retourner à l'interface précédente.

2. Cryptage de textes par saisie :

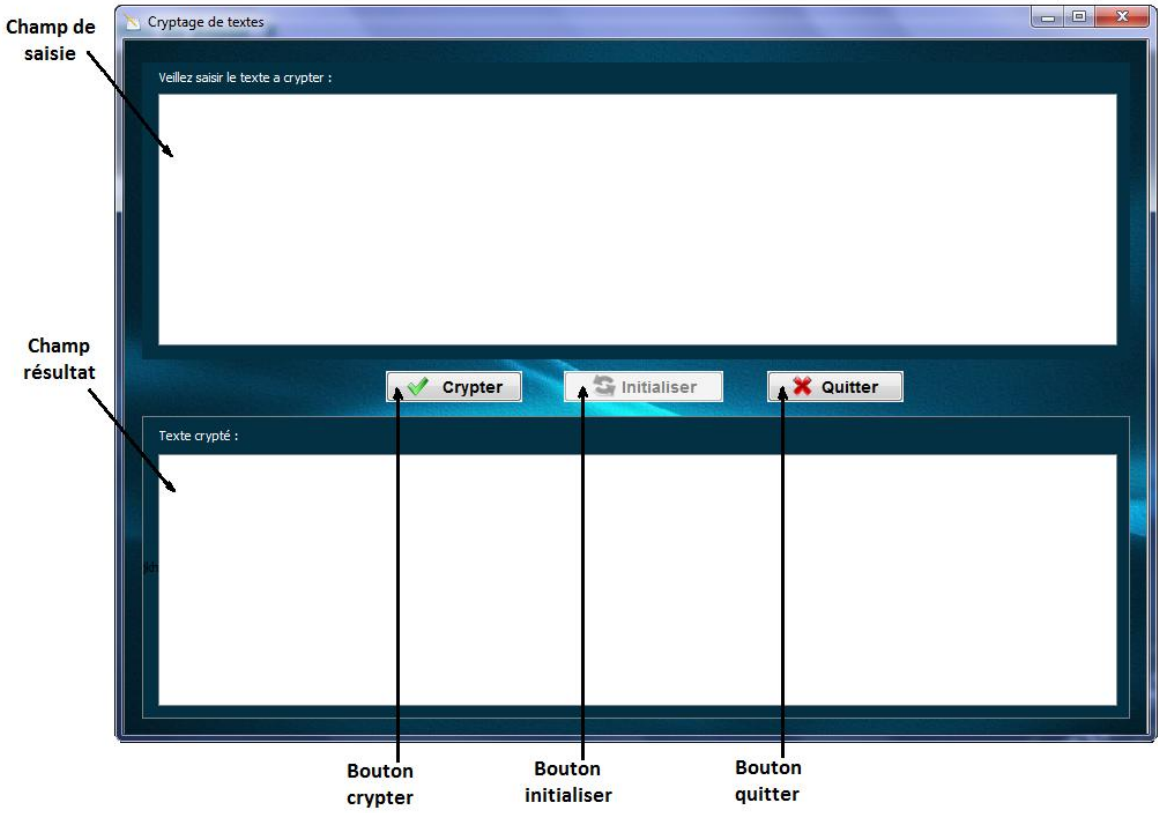


Figure 5-8 Interface cryptage de texte.

Cette interface offre la possibilité de saisir un texte dans le champ de «saisie», et de le crypter en cliquant sur le bouton «Crypter». Le texte crypté apparaît dans le champ «résultat».

Pour crypter un autre texte, il suffit de cliquer sur le bouton «Initialiser» qui efface le contenu des deux champs.

V.3.2.4. Interfaces de décryptage

Figure 5-9 Interface choix du type de décryptage.

Cette interface permet à l'utilisateur d'effectuer un choix entre le décryptage de fichiers par sélection ou, le décryptage de textes par saisie. Pour le décryptage d'un fichier, l'utilisateur doit sélectionner le bouton «Décryptage de fichier par sélection», puis il clique sur le bouton «Valider».

Quant au décryptage d'un texte par saisie, il n'a qu'à sélectionner le bouton «Décryptage de texte par saisie», puis il clique sur «Valider».

Pour quitter l'interface décryptage, l'utilisateur clique sur le bouton «Quitter».

1. Décryptage de fichiers par sélection :

C'est l'interface qui permet de sélectionner un fichier à décrypter en cliquant sur le bouton «Parcourir…», qui lui permet de choisir un fichier à partir d'un répertoire.

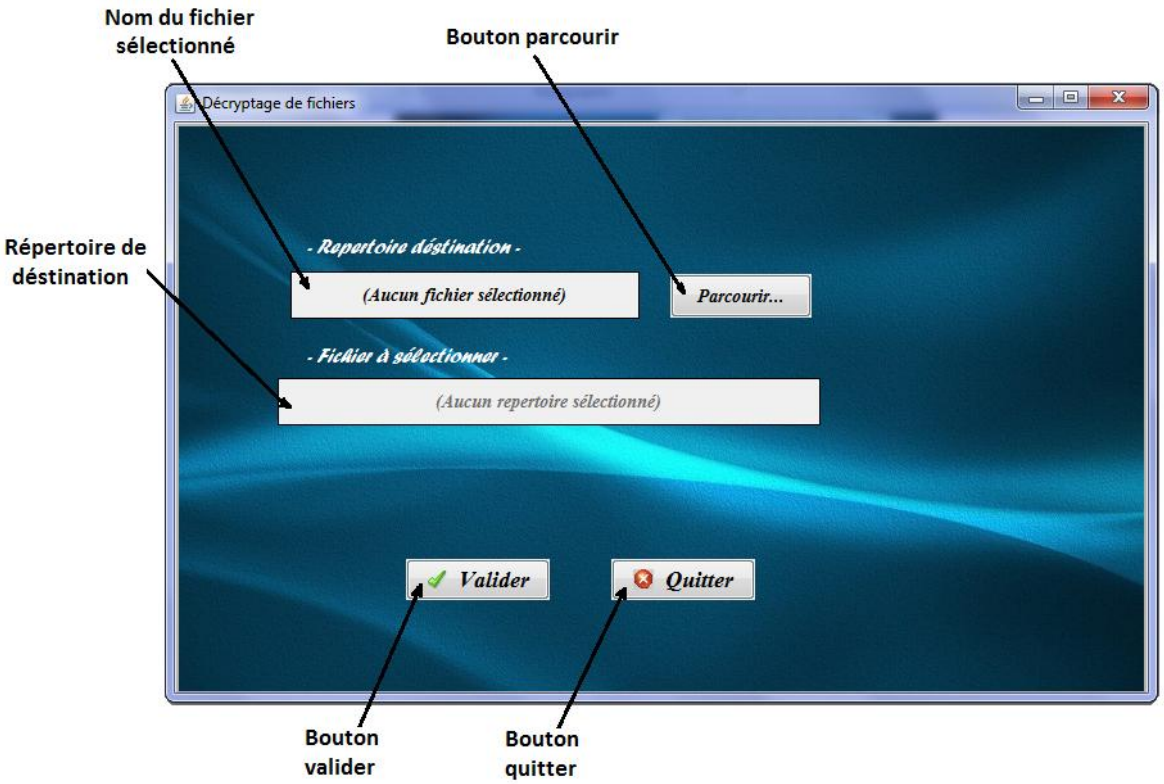


Figure 5-10 Interface décryptage de fichier.

Apres la sélection d'un fichier conforme (qui porte l'extension «.crypt») à décrypter, l'utilisateur passera à la dernière étape où il aura toutes les informations concernant ce fichier (Nom du fichier, sa taille, la difficulté de décryptage,…), il aura aussi à sa disposition trois bouton, dont deux activés à l'état initial, qui sont :

- Bouton «Commencer» : permet le déclanchement de l'opération de décryptage.
- Bouton «Annuler» : permet d'annuler l'opération, et retourne à l'étape précédente.

A la fin de l'opération, le bouton «Fermer» s'active, permettant à l'utilisateur de fermer la fenêtre.

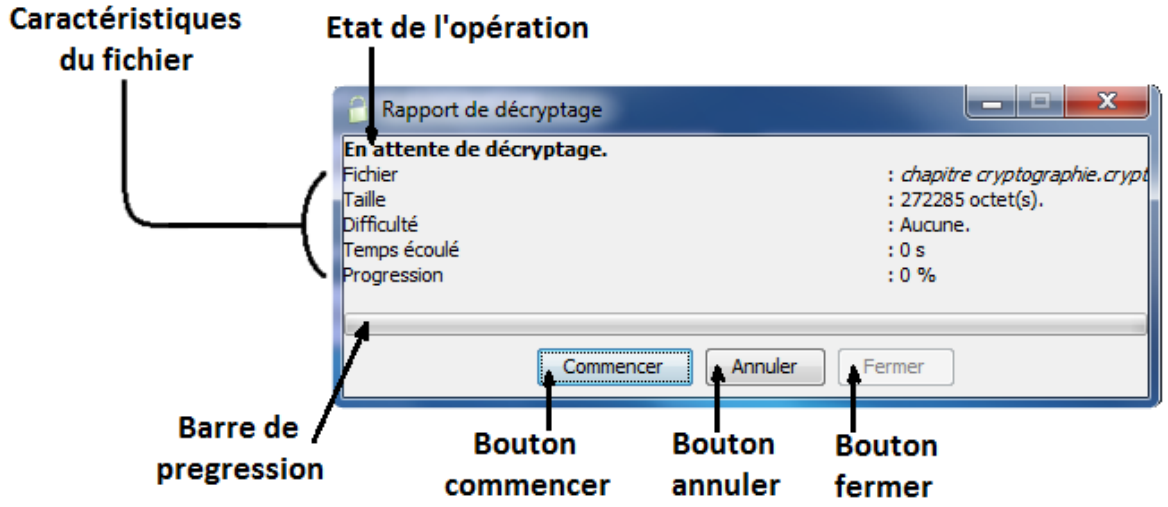


Figure 5-11 Interface rapport de décryptage.

2. Décryptage de textes par saisie :

Cette interface offre la possibilité de saisir un texte dans le champ «saisie», et de le décrypter en cliquant sur le bouton «Décrypter». Le texte décrypté apparaît dans le champ «résultat».

Pour décrypter un autre texte, il suffit de cliquer sur le bouton «Initialiser» qui efface le contenu des deux champs.

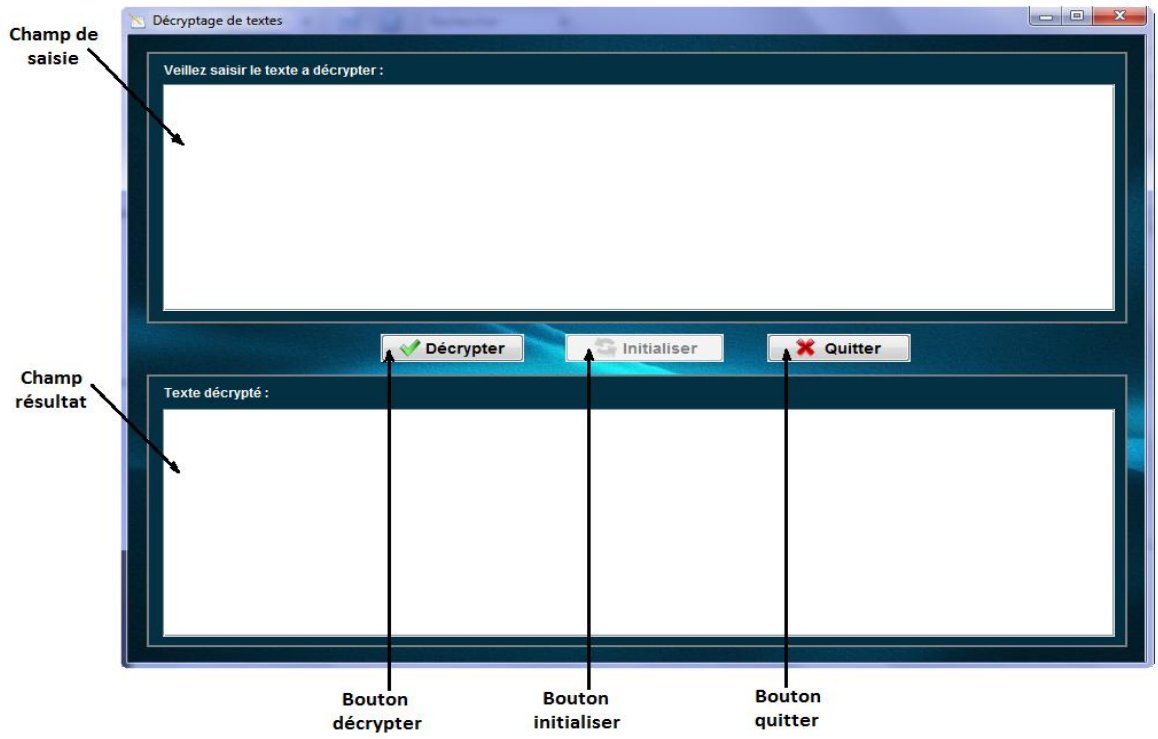


Figure 5-12 Interface décryptage de texte.

V.3.2.5. Interface de changement de mot de passe

Cette interface permet à l'utilisateur de changer le mot de passe d'accès au logiciel, pour ce faire, l'utilisateur entre l'ancien mot de passe, ainsi que le nouveau mot de passe et la confirmation.

A la fin l'utilisateur clique sur le bouton «Valider» pour enregistrer le nouveau mot de passe.

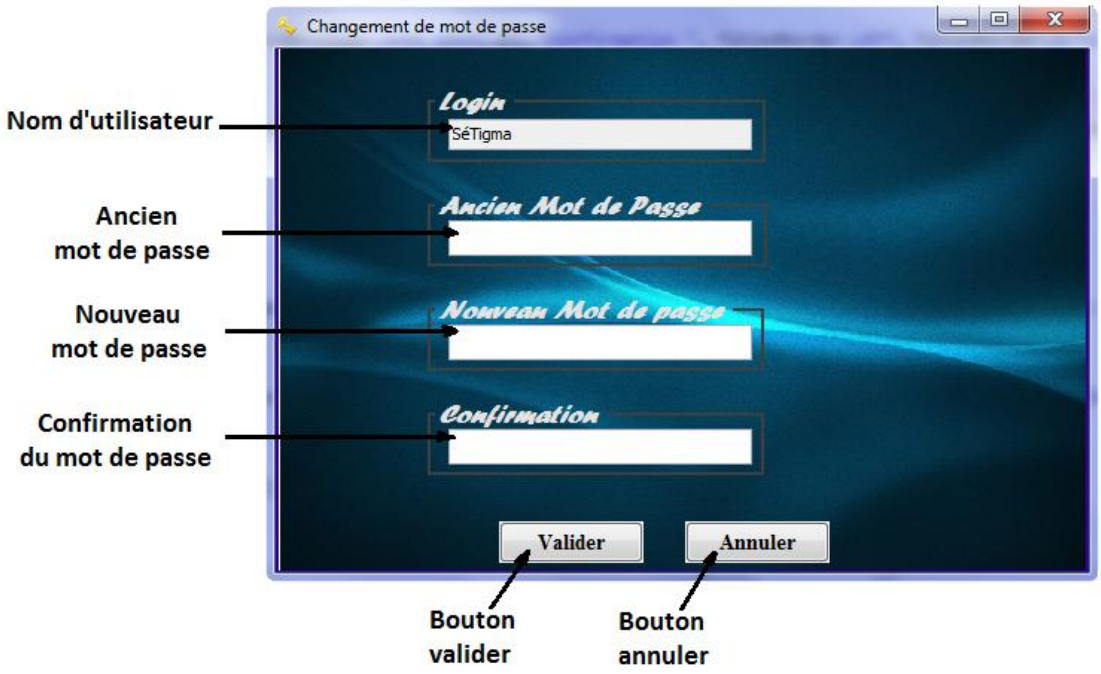


Figure 5-13 Interface changement de mot passe.

V.3.2.6. Interface Aide

Cette interface joue le rôle d'un guide pour l'utilisateur de ce logiciel, elle décrit chaque opération de façon explicite en présentant ses étapes par des illustrations.

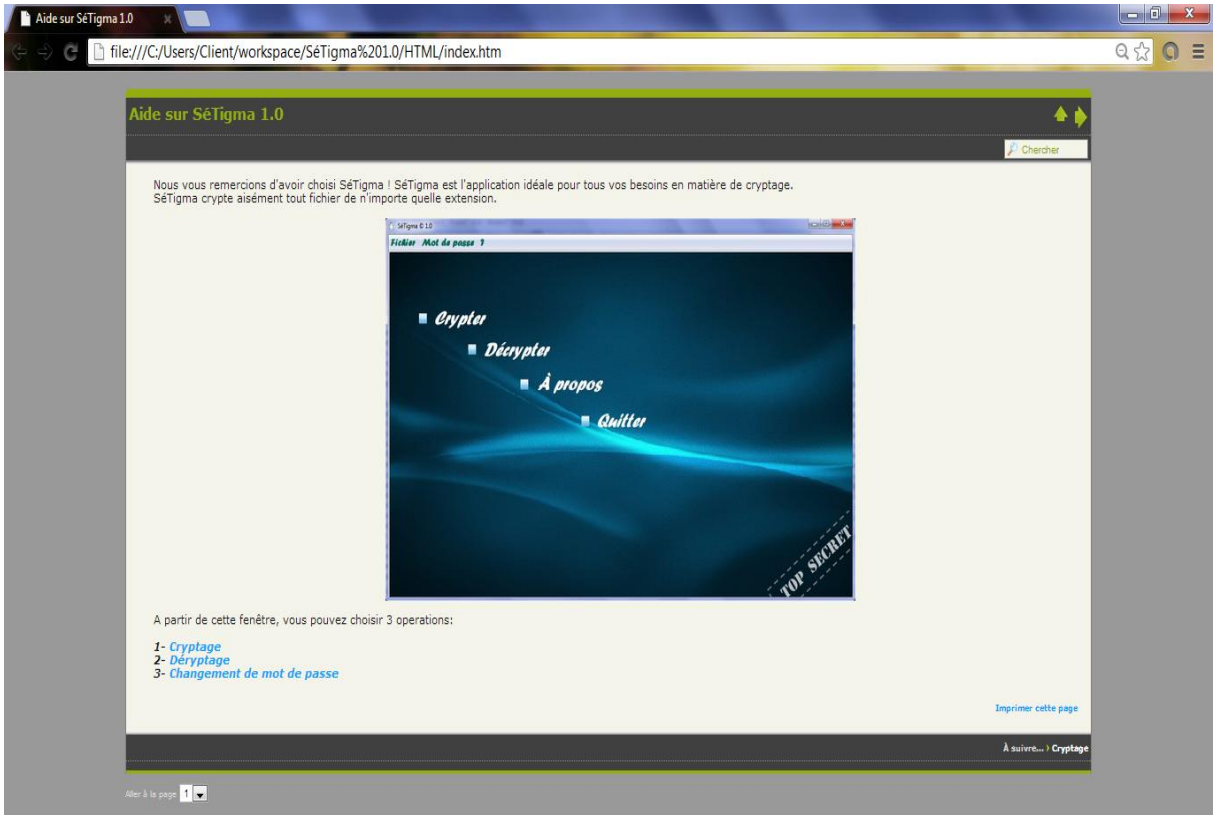


Figure 5-14 Interface aide.

Cette interface héberge trois liens :

- Cryptage.
- Décryptage.
- Changement de mot de passe.

Le lien «Cryptage» permet d'ouvrir une fenêtre expliquant les étapes à suivre pour faire une opération de cryptage.

Le lien «Décryptage» relie une fenêtre qui présente à l'utilisateur le chemin à suivre pour effectuer un décryptage.

Le lien «Changement de mot de passe» pour expliquer à l'utilisateur comment procéder a un changement de mot de passe.

V.3.2.7. Interface A propos

Figure 5-15 Interface à propos.

Permet d'afficher les informations générales concernant le logiciel.

V.3.2.8. Interface de chargement

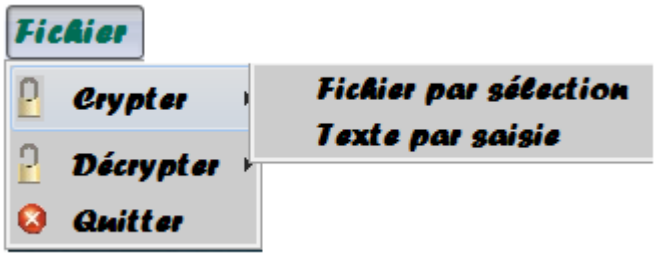
C'est la première interface qui apparaît lors de l'exécution de l'application.

Figure 5-16 Interface Chargement.

V.3.3. Présentation du menu principal

Le menu principal de ce logiciel comporte trois sous-menus différents :

- **Le sous-menu «Fichier» :**



Permet à l'utilisateur de choisir une opération, ainsi que son type.

- **Crypter :** pour choisir le type de cryptage, et lancer la fenêtre liée a ce type.
- **Décrypter :** pour choisir le type de décryptage, et lancer la fenêtre liée à ce type.
- **Quitter :** permet de quitter l'application.

- **Le sous menu «Mot de passe» :**

- **Changer mot de passe :** Permet à l'utilisateur de changer le mot de passe de l'utilisateur.

- **Le sous menu «?» :**

- **Aide :** offre différents conseils pour mieux maitriser le logiciel.
- **À propos :** Affiche la fenêtre donnant les informations générales sur le logiciel.

V.4. Exemples d'utilisation

Afin d'évaluer la performance de notre logiciel, nous vous proposons deux exemples d'utilisation :

- Cryptage d'un fichier.
- Décryptage d'un texte.

V.4.1. Cryptage d'un fichier

Pour dérouler cet exemple, nous allons utiliser le fichier « crypter.docx » dont le contenu est le suivant :

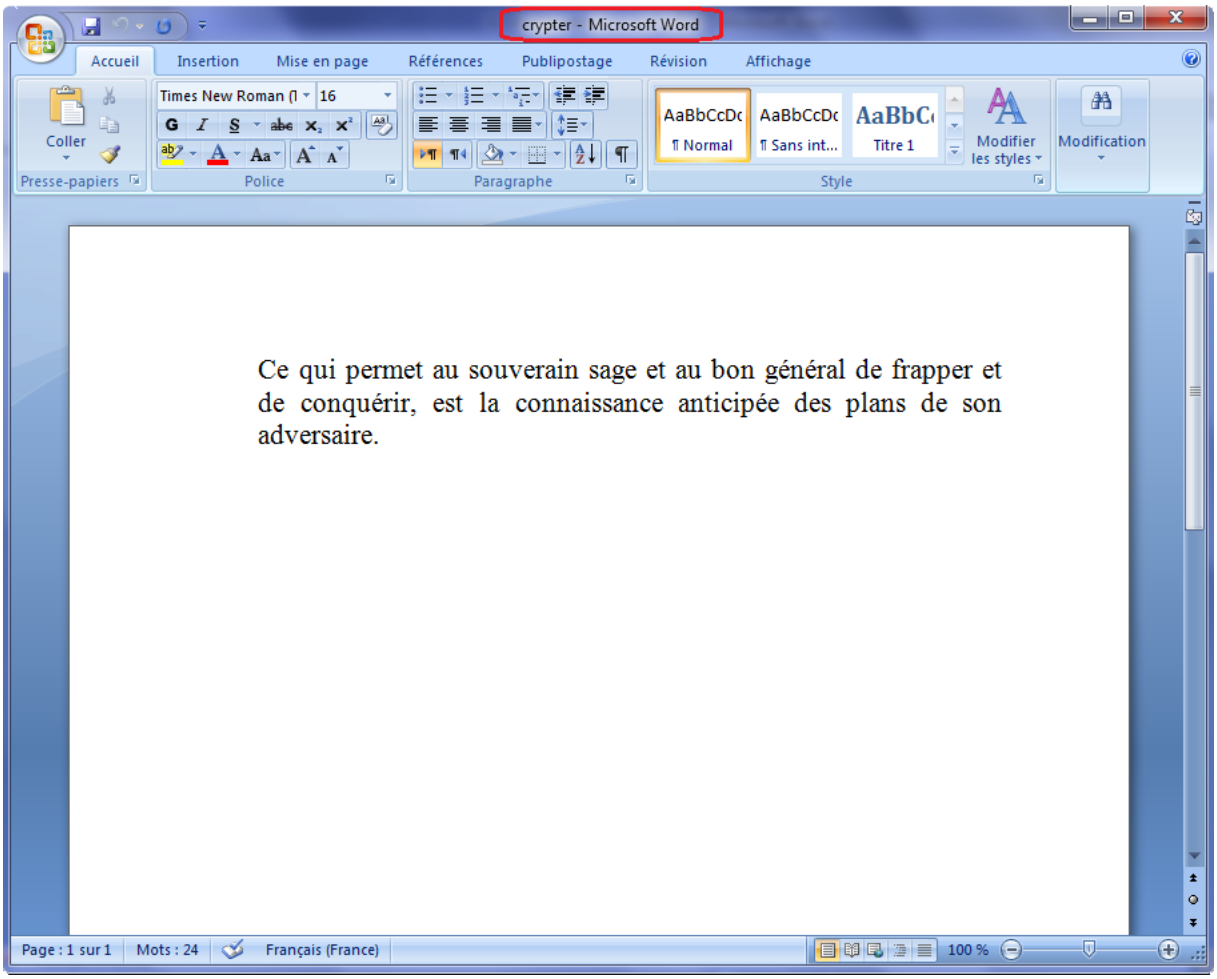


Figure 5-17 Contenu du fichier à crypter.

Pour commencer l'opération de cryptage de ce fichier, on clique sur le bouton «Cryptage» à partir de la fenêtre principale.

Figure 5-18 Sélection d'opération de cryptage.

Après avoir choisi l'opération à faire, on choisi le type de cryptage à utiliser dans la fenêtre qui suit :

Figure 5-19 Choix de cryptage par sélection.

Arrivé a ce niveau, on choisi le fichier à crypter à partir de la fenêtre « Cryptage de fichier » en cliquant sur le bouton « Parcourir »

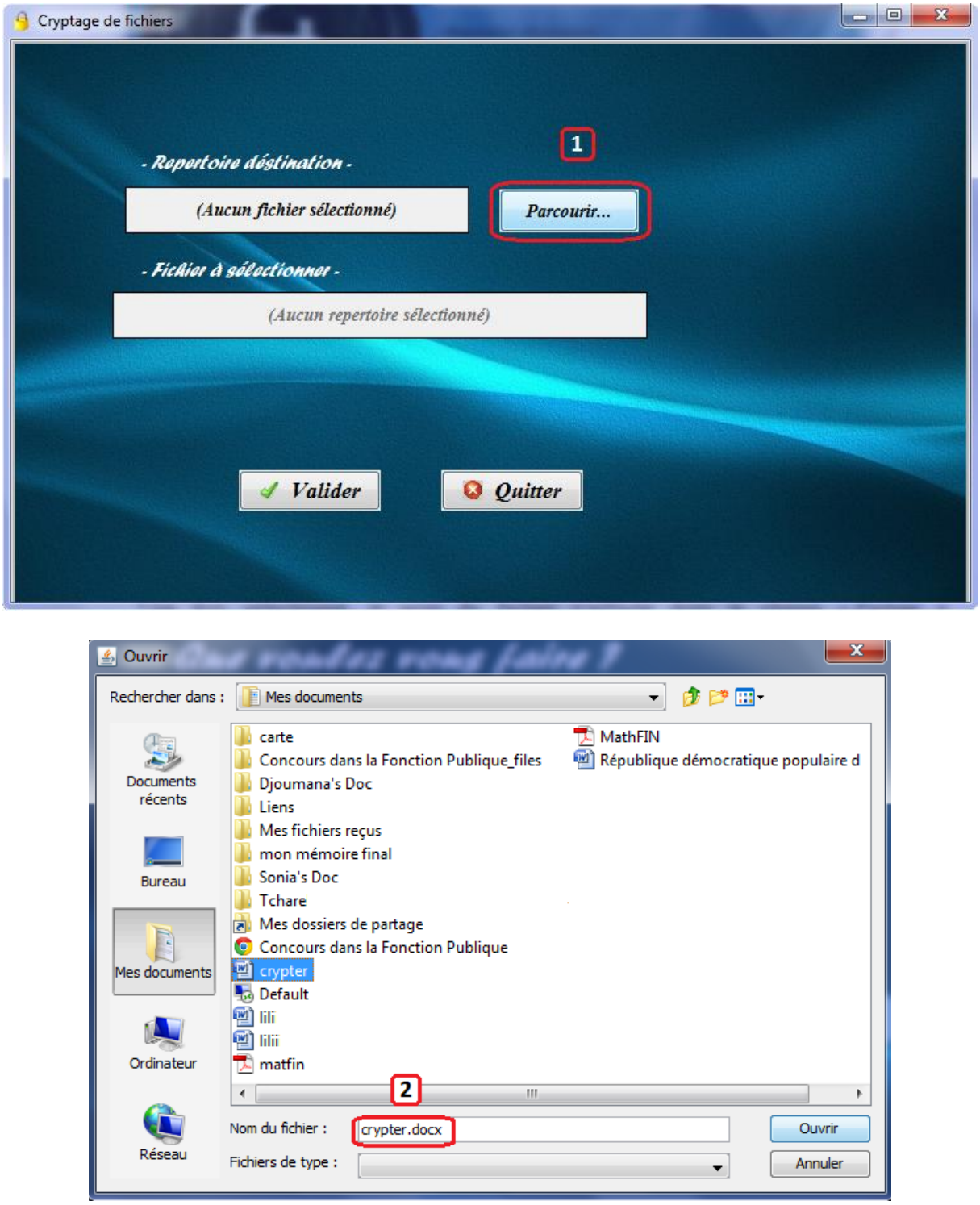


Figure 5-20 choix du fichier à crypter.

Une fois sélectionné, le nom du fichier s'affiche dans le champ « Fichier à sélectionner », et le répertoire du fichier qui sera crypté s'affiche dans le champ « Répertoire destination ».

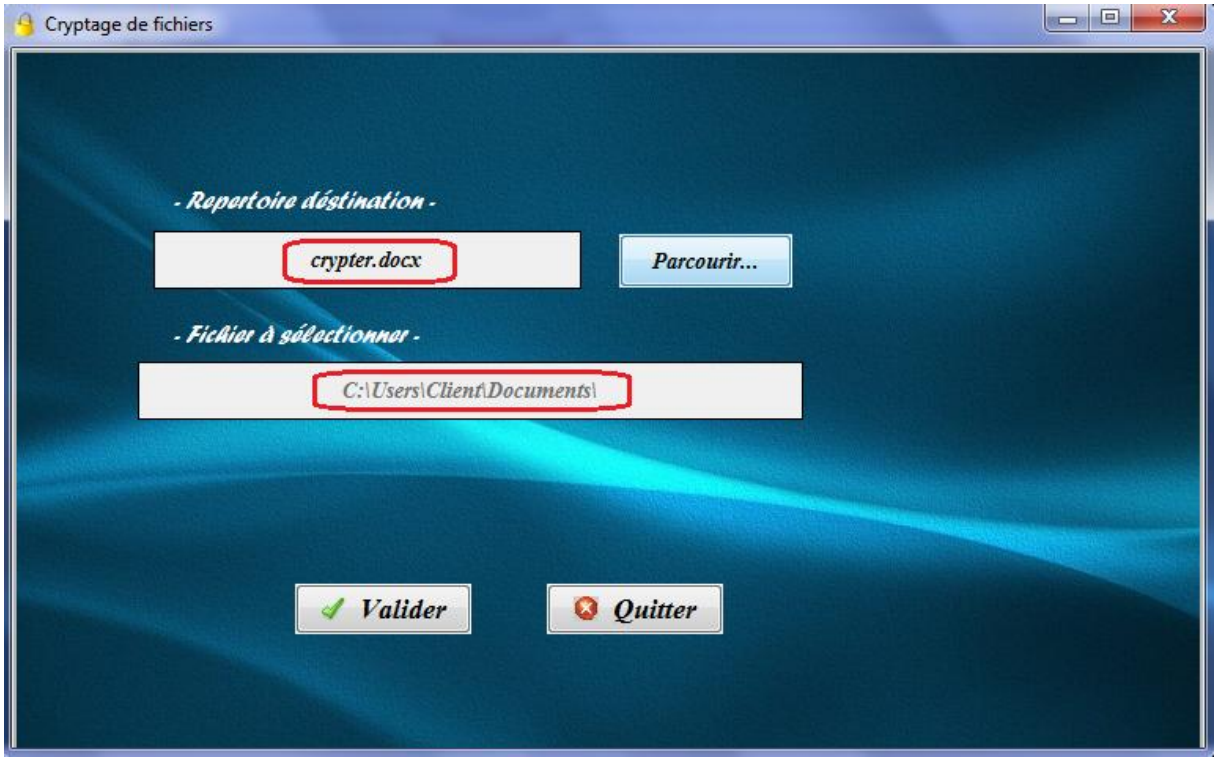


Figure 5-21 Validation de l'opération de cryptage.

Après validation, on aura une fenêtre «Rapport de cryptage» qui contient les caractéristiques du fichier à l'état initial.

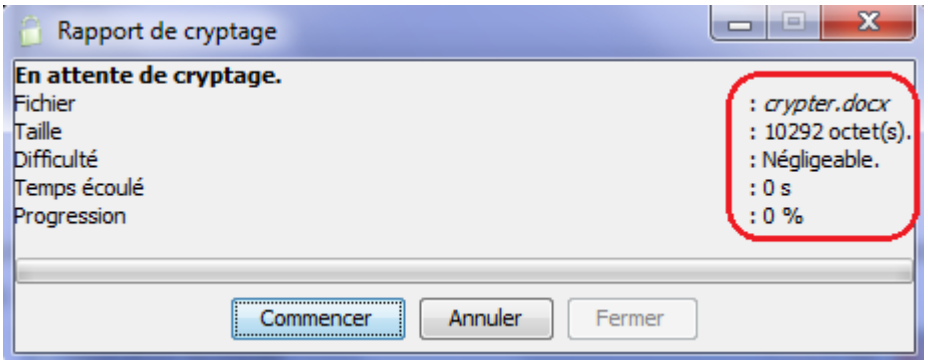


Figure 5-22 Rapport initial de cryptage.

Pour commencer le cryptage, on clique sur le bouton « Commencer », et à la fin de l'opération on aura :

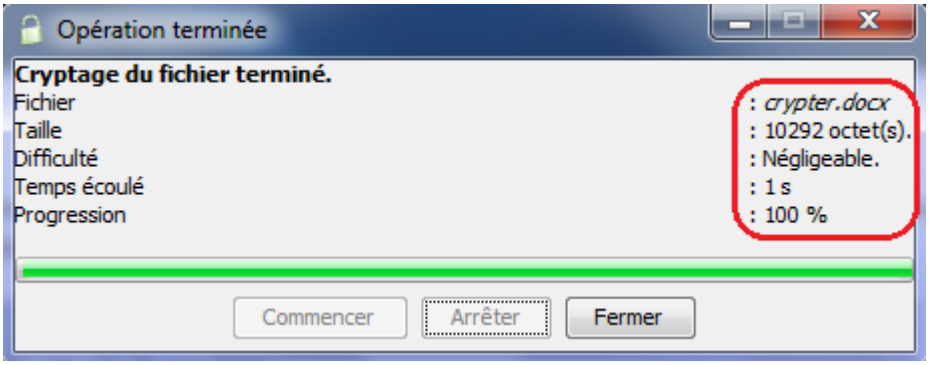


Figure 5-23 Rapport final de cryptage

A la fin du cryptage, on ouvre le fichier crypté avec l'éditeur de texte de Windows «Bloc note», et on trouve le contenu suivant :

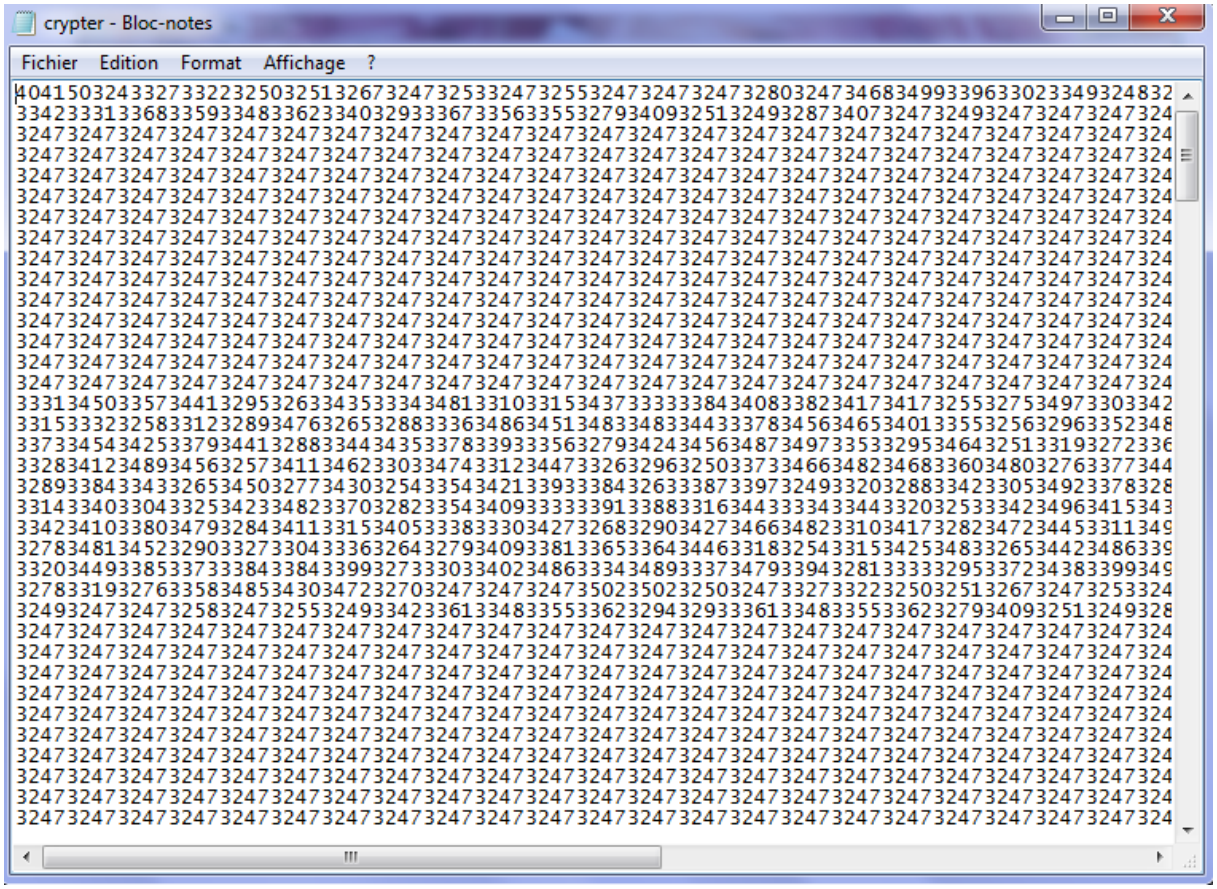


Figure 5-24 Contenu du fichier crypté.

V.4.2. Décryptage d'un texte

Pour décrypter un texte, on clique sur le bouton «Décryptage» à partir de la fenêtre principale.

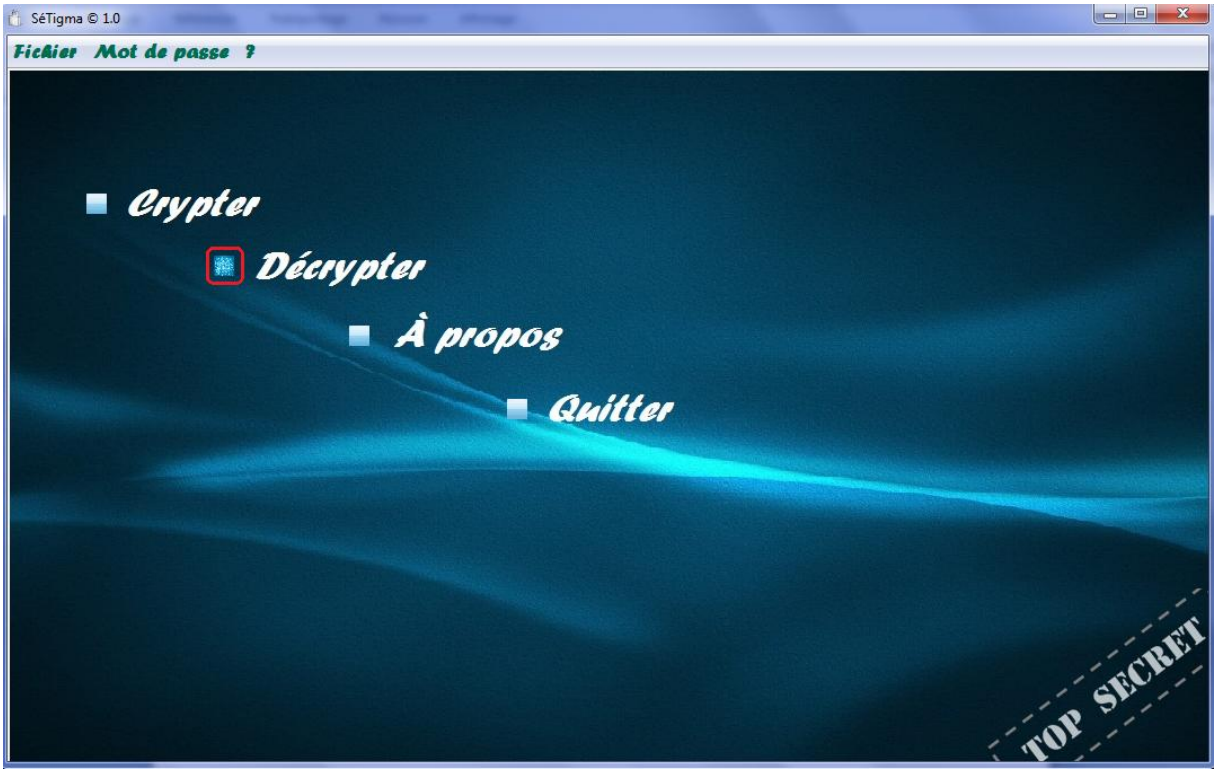


Figure 5-25 Sélection d'opération de décryptage.

Après avoir choisi l'opération à faire, on choisi le type de décryptage à utiliser dans la fenêtre qui suit :

Figure 5-26 Choix de décryptage par saisie.

Après cette étape, on saisi le texte à décrypter dans le champ de saisie, et on clique sur le bouton « Décrypter » :

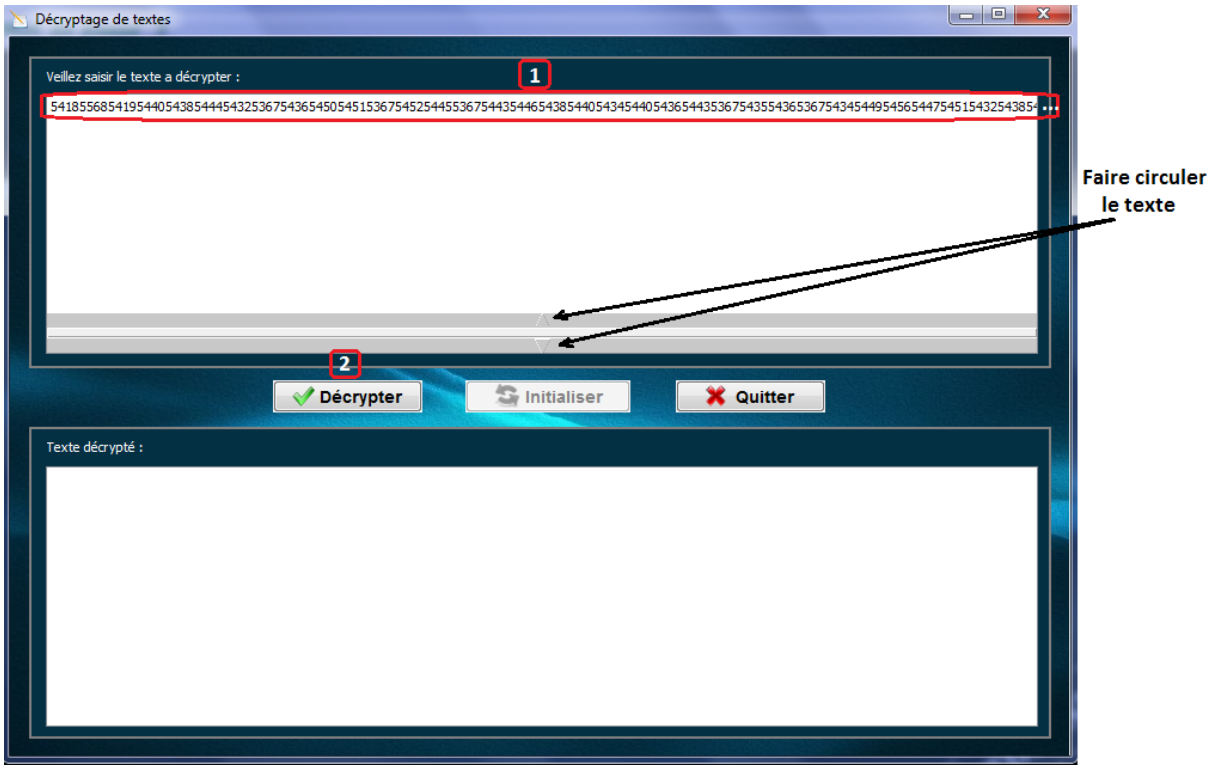


Figure 5-27 Saisie du texte à décrypter.

A la fin, nous aurons le texte décrypté suivant :

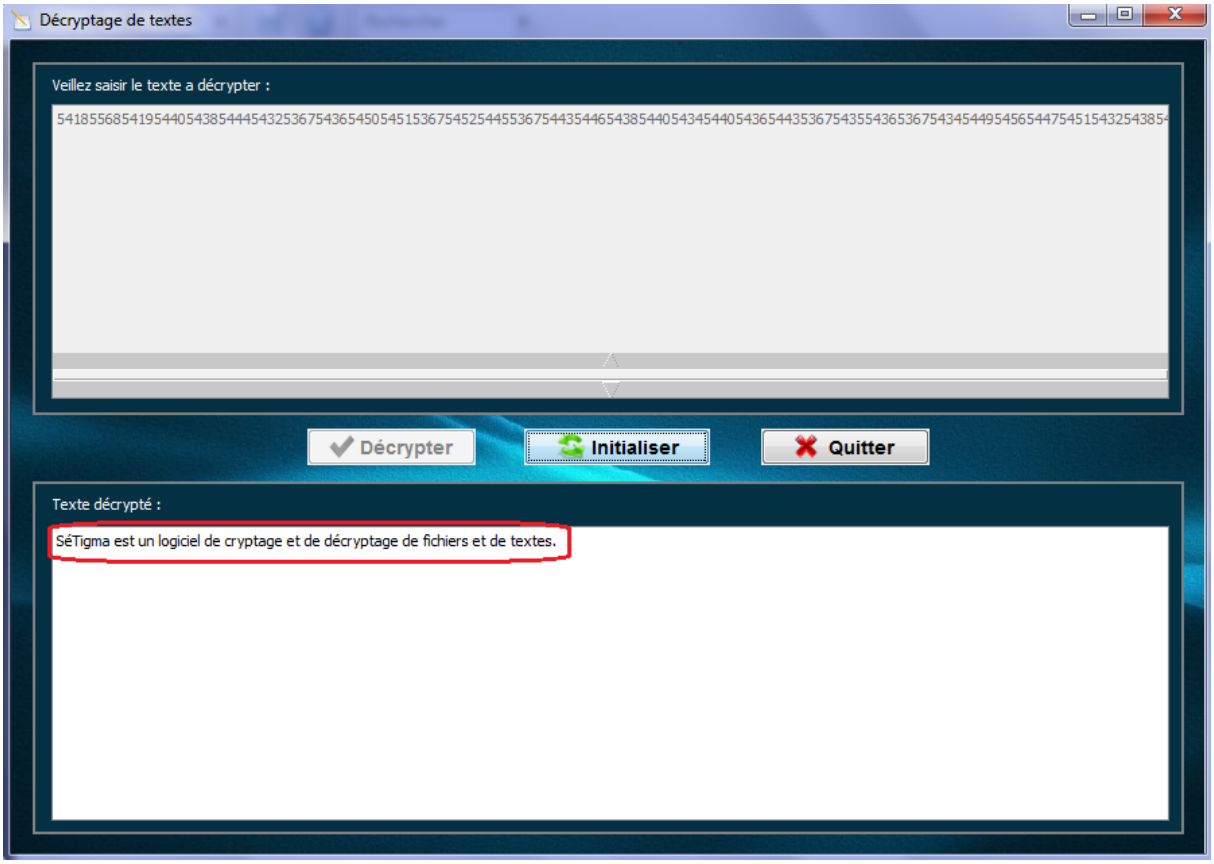


Figure 5-28 Décryptage du texte saisi.

V.5. Conclusion

Dans ce chapitre, nous avons présenté l'environnement d'implémentation et de développement de notre application, ainsi que les outils utilisés pour réaliser notre travail.

La description de notre application s'est faite en présentant les interfaces essentielles de notre logiciel et en suivant des scénarios décrits schématiquement, ainsi que des exemples d'utilisation pour évaluer la performance de notre logiciel.

Conclusion générale

Conclusion générale

Dans notre travail, nous avons commencé par revenir aux racines de la cryptographie en parlant des plus emblématiques et des plus anciennes, voire très anciennes, méthodes de cryptographie. C'est ainsi que nous avons « découvert » le principe de substitution mono-alphabétique avec la célèbre application proposée par César : le code/chiffre de César. Nous avons également abordé le chiffre de Vigenère qui constituait une application de la substitution polyalphabétique. Une constatation évidente fut que la substitution mono-alphabétique possédait de nombreuses lacunes, du fait de sa simplicité et des caractéristiques de la langue française, que son homologue ne présentait plus.

Nous avons ensuite pris un peu de recul pour parler des deux grandes familles de méthodes de chiffrement et expliquer leurs modes de fonctionnement, leurs applications et les notions qui vont avec la cryptographie de manière générale. Nous avons par exemple abordé quelques applications comme le chiffrement par flot ou encore le chiffrement hybride.

Après cela, nous avons présenté quelques logiciels actuels permettant de chiffrer des données avec plus ou moins de succès.

Le travail que nous avons mené, nous a permis d'étudier quelques solutions pour contrer les menaces aussi nombreuses que variées, qui pèsent sur le réseau, et également de nous initier dans le domaine de la cryptographie et sur la programmation orientée objet.

En effet, notre application s'est portée sur le chiffrement des informations. Le logiciel que nous avons réalisé permet de chiffrer et déchiffrer des fichiers, et des textes saisis, avec une clé secrète grâce à un algorithme symétrique que nous avons proposé, et ainsi de sécuriser les informations et les mettre à l'abri des indiscrets.

La cryptographie n'a pour limite que son temps, chaque méthode finit par être déchiffré au fur et à mesure du temps, ce qui prouve qu'une cryptographie sans faille n'existe pas tant que des personnes se donneront la peine de les décoder.

Nous avons donc défini qu'une cryptographie, aussi brillante soit elle, atteindra tôt ou tard une limite dans son fonctionnement, mais nous avons vu aussi qu'il n'y a pas de limite à de nouvelles cryptographies puisque ces méthodes existent depuis des siècles et même si leurs méthodes ont été trouvés de nouvelles sont mises en place.

Références bibliographique

Bibliographie

[01] : **Jean François Carpentier**, La sécurité informatique dans la petite entreprise, Edition ENI, Avril 2009.

[02] : **Jean-Marc ROYER**, Sécuriser l'informatique de l'entreprise : Enjeux, Menace, Prévention et parades, Edition ENI- 2007.

[03] : **Eric Maiwald**, Sécurité des réseaux, Edition CompusPress, 2001.

[04] : **Guy Pujolle**, Les Réseaux, Edition EYROLLES, 1995-1997.

[05] : Sécurité des réseaux : analyse et mise en œuvre, Janvier 1996.

[06] : **Phil Zimmermann**, Une Introduction à la Cryptographie, Edition NAI, 1998.

[07] : B. Schneier. Applied Cryptography. John Wiley and Sons, 1996.

[08] : Hérodote. Histoires. Hachette, 1958.

[09] : Azzouzi Oussama & Haddadi Ferhat, Plateforme de chiffrement/déchiffrement pour la sécurisation du stockage et de la transmission de l'information, 2012.

[10]: **Claude Lecommandeur**, Chiffrement des données et sécurité informatique, Revue Flash informatique, numéro 7, 26 septembre 1995.

[11]: **Ghislaine Labouret**, Introduction à la cryptologie, Edition *HSC (Hervé Schauer Consultants)*, Novembre 1998.

[12] : Bruce Schneier. Cryptographie appliquée. Vuibert, Collection « Vuibert informatique ». 1994.

[13] : Jean-Baptiste Campesato, L'algorithme RSA, 2010

[14] : **Claude Delannoy**, Programmer en java (5eme édition / Java 5 et 6), EYROLLES, 2008.

Webographie

[WEB 01] : www.wikipedia.org

[WEB 02] : www.cases.public.lu (/fr/publications/dossiers/cryptographie)

[WEB 03] : www.commentcamarche.net (/Introduction au chiffrement)

[WEB 04] : www.bibmath.net (/crypto/)

[WEB 05] : www.authsecu.com (/ssl/historique)

[WEB 06] : www.interstices.info (/turing-enigma)

Annexe

A. Présentation de l'environnement

A.1. Langage de programmation utilisé

Langage JAVA [14]

Nous avons choisi le langage Java pour écrire notre programme, ce choix est motivé par les raisons suivantes :

- **Java est indépendant de toute plate-forme :** Une application en java fonctionne sur n'importe quel environnement (Unix, Windows, …) disposant d'une Java Virtual Machine JVM (la machine virtuelle java).
- **Java est extensible à l'infini :** Idéalement, toutes les catégories d'objets (appelées classes) existantes en java sont définies par extension d'autres classes, en partant de la classe de base la plus générale : la classe Object. Pour étendre le langage il suffit donc de développer de nouvelles classes.
- **Java est un langage de haute sécurité :** Java a été développé dans un souci de sécurité maximale. L'idée maitresse est qu'un programme comportant des erreurs ne doit pas pouvoir être compilé. Ainsi les erreurs ne risquent pas d'échapper du programmeur et de passer les procédures de tests. En détectant les erreurs à la source, on évite qu'elles se propagent en s'amplifiant.
- **Java est un langage compilé :** C'est-à-dire qu'avant d'être exécuté, il doit être traduit dans le langage de la machine sur laquelle il doit fonctionner. Cependant, contrairement à de nombreux compilateurs, java traduit le code source dans le langage de sa JVM. Le code traduit appelé byte code, ne peut pas être exécuté directement par le processeur d'une machine.
- **Java est doté de standard de bibliothèques de classes :** ces classes sont très riches et elles comprennent la gestion des interfaces graphiques (fenêtres, boites de dialogue, contrôles, menus, graphisme), la programmation multi-threads (multitâches), la gestion des exceptions, les accès aux fichiers et au réseau… l'utilisation de ces bibliothèques facilite grandement la tâche du programmeur lors de la construction d'applications complexes.

A.2. Présentation d'Eclipse

Eclipse est un projet open source fondé par SUN Microsystems. L'IDE Eclipse est un environnement de développement permettant d'écrire, de compiler, de déboguer et de déployer des programmes. Il est écrit en java, et il y'a un grand nombre de modules pour étendre l'IDE Eclipse.

L'IDE Eclipse est un produit gratuit, sans aucune restriction quant à son usage.

L'installation de L'IDE Eclipse nécessite l'installation de la JDK (Java Développement Kit), le kit de développement java compatible avec la version d'IDE.

Pour concevoir notre application, nous avons utilisé la version Eclipse JUNO. Son interface principale est donnée dans la Figure 5-1 suivante :

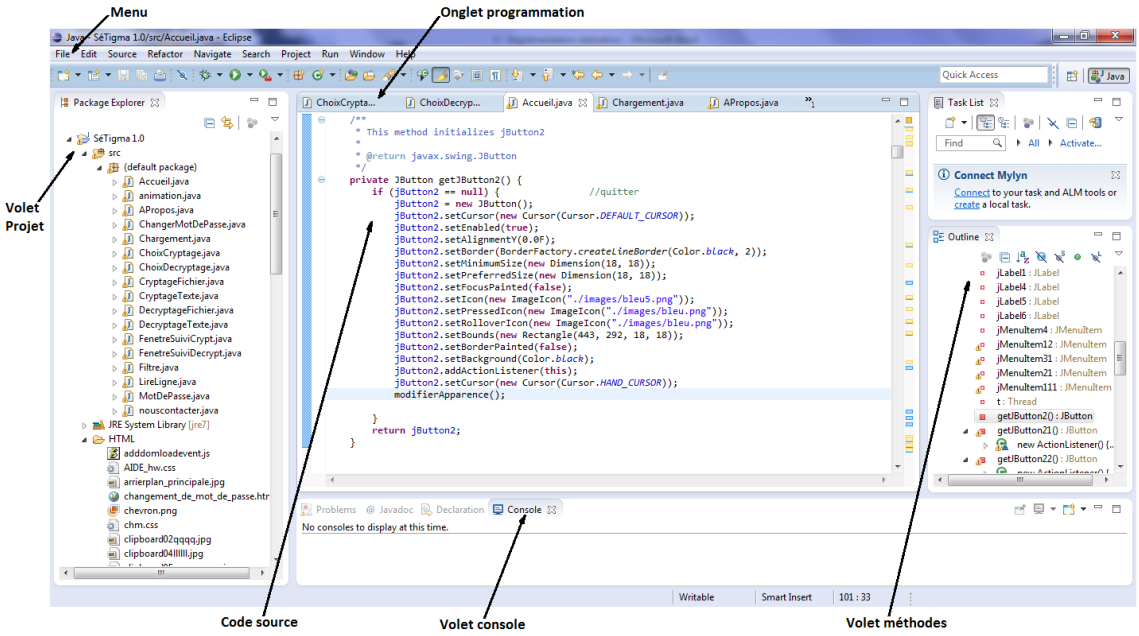


Figure A-1 Interface JAVA Eclipse.