



N° d'ordre :

UNIVERSITE DE M'SILA
FACULTE DE MATHEMATIQUES ET D'INFORMATIQUE
Département d'Informatique

MEMOIRE

Présenté pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique

Filière : Informatique

Spécialité : système d'information avancée

Par :

Mourad ALILI

SUJET

Mise au point d'un système de
cryptage/décryptage pour un client de
messagerie électronique

Soutenu publiquement le : 27/06/2012 devant le jury composé de:

Mr.LAMICHE Chaabane	Université de M'sila
Mr.HEMMAK Allaoua	Université de M'sila
Mr.CHIKOUCHE Nourddine	Université de M'sila
Mr.BOUBAKIR Mouhamed	Université de M'sila

Président
Rapporteur
Examineur
Examineur

Promotion : 2011 /2012

DEDICACE

Pour l'amour de mon dieu, je dédie ce modeste travail

A mes très chers parents qui n'ont cessé de m'encourager durant toute l'année.

Et qui m'ont fait bénéficier de leur générosité et de leur soutien à tout moment.

A mes frères .A mes sœurs.

A mes très chers amis, pour leurs qualités humaines, leurs optimismes,

Leur amitié et leur soutien appréciable au cours de ces années.

...et à toute ma famille.

Mourad ALILI

REMERCIEMENTS

*Je voudrais très sincèrement remercier monsieur **Mr. HEMMAK Allaoua** pour avoir dirigé l'encadrement de ce travail, sa disponibilité, son expérience, son savoir scientifique et sa qualité humaine qui ont été des plus déterminants dans l'aboutissement de ce travail.*

Je remercie également :

Messieurs les membres du jury pour l'intérêt qu'ils ont porté pour mon travail.

Tous ceux qui m'ont aidé de près ou de loin.

*Je voudrais terminer en saluant la **promotion d'Informatique 2012**.*

Mourad ALILI

Table des matières

Listes des Figures	1
Listes des Tables	2
INTRODUCTION	3
 CHAPITRE 1 : LA CRYPTORAPHIE	
1. Introduction	5
2. Définitions de bases	5
3. Les objectifs de Chiffrement	6
4. Systèmes à clé privée	7
4.1. Cryptosystèmes par flots	8
4.1.1. Système de Vigenère	8
4.2. Cryptosystèmes par blocs	9
4.2.1. Rijndael	11
5. Le chiffrement asymétrique	17
5.1. Le système RSA	18
5.1.1. Présentation du cryptage RSA	18
5.1.2. Principe mathématique	18
5.1.3. Génération des clés	19
5.1.4. Cryptage et décryptage	19
5.1.5. Exemple.....	19
6. Les fonctions de hachage	20
7. Le codage des informations	21
7.1. Pourquoi coder ?	21
7.2. Codage ASCII étendu.....	21
7.3. Le codage Base64	21
7.4. Unicode	22
7.5. Autres codes existant	22
8. Conclusion	22

CHAPITRE 2 : LES MESSAGERIE ELECTRONIQUE ET LEUR CHIFFREMENT

1.Introduction	23
2.Qu'est-ce qu'Internet	23
2.1. Comprendre les termes d'internet	23
2.2. Structure des adresses	23
3.Les courriers électroniques	25
3.1. Définition de courrier électronique	25
3.2. Origines	25
3.3. Différents types des clients des courriers électroniques	26
3.4. Quelques exemples du webmails	26
3.5. Appellations des clients de messagerie	27
3.6. Composants d'un service de courrier électronique	27
3.7. Structure d'un système de messagerie	27
3.7.1. Agent utilisateur (user agent ou UA)	27
3.7.2. Agent de routage des messages	28
3.7.3. Agent de transport des messages	28
3.7.4. Les boîtes aux lettres	28
4.Architecture Client/Serveur	29
4.1. Notion de base	29
5.Architecture logicielle d'une messagerie	29
5.1. Les logiciels de messagerie	30
5.2. Les protocoles de messagerie	31
5.2.1. SMTP pour la gestion du courrier	31
5.2.2. Le protocole ESMTP (Extended Simple Mail Transfer Protocol)	33
5.2.3. POP3 et IMAP pour interroger la BAL	33
5.2.4. MIME pour la mise en forme des messages	35
5.3. Processus d'envoi et de réception d'un courrier électronique	35
6.L'état de l'art	36

6.1. Pourquoi crypter ?	36
6.2. Les applications du chiffrement.....	36
6.2.1. Le protocole SSL.....	36
6.2.2. Le protocole HTTPS	37
6.2.3. PGP (Pretty Good Privacy)	37
6.3. Le cryptage dans les clients de messageries actuels.....	40
6.3.1. Le chiffrement dans Microsoft Outlook 2010	40
6.3.1. Mozilla Thunderbird et Enigmail	41
7.La contribution	41
7.1. Evaluation des résultats.....	43
8.Conclusion	43

CHAPITRE 3 : ANALYSE ET CONCEPTION

1. Introduction	44
2. Analyse des besoins	44
2.1. Fonctionnalités attendues	44
2.2. Public visé	44
3. L'outil de conception utilisé	45
4. Présentation du système en UML	45
4.1. Scénarios de chaque cas d'utilisation.....	46
4.2. Diagramme de cas d'utilisation.....	47
4.3. Scénarios de cas d'utilisation «Crypté les messages envoyés»	49
4.3.1. Diagramme de séquence.....	49
4.3.2. Diagramme de classe	49
4.3.3. Diagramme d'état transition.....	50
4.3.4. Diagramme d'activité	51
4.4. Scénarios de cas d'utilisation «Décrypté les messages reçues»	52
4.4.1. Diagramme de séquence.....	52
4.4.2. Description des classes associées.....	52
4.4.3. Diagramme d'état transition.....	53
4.4.4. Diagramme d'activité	53
4.5. Scénarios de cas d'utilisation «Envoyer message».....	54

4.5.1. Diagramme de séquence.....	54
4.5.2. Description des classes associées.....	54
4.5.3. Diagramme d'état transition.....	54
4.5.4. Diagramme d'activité	55
4.6. Scénarios de cas d'utilisation «Consulter les messages reçus»	56
4.6.1. Diagramme de séquence.....	56
4.6.2. Description des classes associées.....	56
4.6.3. Diagramme d'état transition.....	57
4.6.4. Diagramme d'activité	57
5. Les algorithmes utilisés	58
5.1. Rijndael.....	58
5.2. RSA (Ron Rivest, Adi Shamir et Leonard Adleman)	58
6. Les protocoles utilisés.....	58
6.1. POP3.....	58
6.2. SMTP	58
7. Les bases de données utilisées.....	58
8. Conclusion	60

CHAPITRE 4 : REALISATION ET IMPLEMENTATION

1. Introduction	61
2. Environnement de travail.....	61
2.1. Environnement matériel	61
2.2. Environnement logiciel.....	61
3. L'accès à l'application	62
4. Test du programme	69
5. Conclusion	73

CONCLUSION	74
-------------------------	-----------

NOMENCLATURE	75
---------------------------	-----------

BIBLIOGRAPHIE	77
----------------------------	-----------

Listes des Figures

Chapire 01

Figure 1.1 : Principe de chiffrement/déchiffremnt	4
Figure 1.2 : Chiffrement symétriqu	5
Figure 1.3 : carre de vigenere.....	6
Figure 1.4 : Schéma bloc de l'algorithme Rijndael avec une clé de 128 bits.....	9
Figure 1.5 : Schéma de déchiffrement d'algorithme Rijndael.....	14
Figure 1.6 : Chiffrement asymétrique	16

Chapire 02

Figure 2.1 : Structure d'un Système de messagerie	26
Figure 2.2 : Processus d'envoi et de réception d'un courrier.....	34
Figure 2.3 : Chiffrement de PGP	36
Figure 2.4 : Déchiffrement de PGP	36
Figure 2.5 : Conception de nouveau chiffrement.....	40
Figure 2.6 : Conception de nouveau déchiffrement	41

Chapire 03

Figure 3.1: Diagramme de cas d'utilisation	46
Figure 3.2: diagramme de séquence associe au scénario «Crypté les messages envoyés»	47
Figure 3.3 : Diagramme de classe associe à « crypté les messages envoyés »	48
Figure 3.4 : Diagramme d'état transition associe à « crypté les messages envoyés »	48
Figure 3.5 : Diagramme d'activité associe à « crypté les messages envoyés »	49
Figure 3.6: diagramme de séquence associe au scénario «Décrypté les messages reçues»	50
Figure 3.7 : Diagramme de classe associe à «Décrypté les messages reçues »	50
Figure 3.8 : Diagramme d'état transition associe à «Décrypté les messages reçues »	51
Figure 3.9 : Diagramme d'activité associe à «Décrypté les messages reçues»	51
Figure 3.10 : Diagramme de séquence associe à «Envoyer message»	52
Figure 3.11 : Diagramme de classe associe à «Envoyer message»	52
Figure 3.12 : Diagramme d'état transition associe à «Envoyer message».....	52
Figure 3.13 : Diagramme d'activité associé à «Envoyer message»	53
Figure 3.14 : Diagramme de classe associe à «Consulter les messages reçus»	54
Figure 3.15 : Diagramme de classe associe à «Consulter les messages reçus »	54
Figure 3.16 : Diagramme d'état transition associe à «Consulter les messages reçus »	55
Figure 3.17 : Diagramme d'activité associé à «Envoyer message»	55
Figure 3.18 : tables de la base de données clefs.sdf	56

Figure 3.19 : tables de la base de données contacte_compte.sdf	57
Figure 3.20 : tables de la base de données emails.sdf	57
Figure 3.21 : La table Login.....	58

Chapire 04

Figure 4.1 : La fenêtre d'authentification.....	60
Figure 4.2 : La fenêtre de modification de mot de passe et le nom d'utilisateur	60
Figure 4.3 : La forme principale.....	61
Figure 4.4: la liste des comptes	61
Figure 4.5 : la liste des comptes	62
Figure 4.6 : la liste des comptes	62
Figure 4.7 : la liste des comptes	63
Figure 4.8 : la liste des contacts indésirables	64
Figure 4.9 : la liste des contacts	64
Figure 4.10: Générât pair de clef RSA.....	65
Figure 4.11 : Générât une clef AES	65
Figure 4.12 : Gestion des clefs publiques	65
Figure 4.13 : Gestion des clefs privées	66
Figure 4.14 : Ecrire un message.....	66
Figure 4.15 : les mails cryptés	67
Figure 4.16 : les mails cryptés	68
Figure 4.17 : envoi un mail crypté	69
Figure 4.18 : envoi un mail crypté	69
Figure 4.19 : Déchiffrer un mail reçu.....	70
Figure 4.20: Envoi un mail clair	70
Figure 4.21: recevoir un mail clair	71

Listes des Tables

Tableau 1.1 : S-Box	10
Tableau 2.1 : Champs Normalisés du protocole SMTP	30
Tableau 2.2 : Commandes d'envoi du protocole SMTP	30
Tableau 2.3 : Réponses Aux commandes du protocole SMTP	30
Tableau 2.4 : commandes du protocole POP3.....	32

INTRODUCTION

GENERALE

Introduction générale

Les communications ont toujours constitué un aspect important dans l'acquisition de nouvelles connaissances et l'essor de l'humanité. Le besoin d'être en mesure d'envoyer un message de façon sécuritaire est probablement aussi ancien que les communications elles-mêmes.

D'un point de vue historique, c'est lors des conflits entre nations que ce besoin a été le plus vif. Dans notre monde moderne, où diverses méthodes de communication sont utilisées régulièrement, le besoin de confidentialité est plus présent que jamais à une multitude de niveaux. Par exemple, il est normal qu'une firme désire protéger ses nouveaux logiciels contre la piraterie, que les institutions bancaires veuillent s'assurer que les transactions sont sécuritaires et que tous les individus souhaitent que l'on protège leurs données personnelles.

Le besoin de communications sécuritaires a donné naissance à la science que nous appelons cryptologie.

Le cryptage (ou chiffrement) est une opération mathématique qui permet de coder le contenu d'un message afin de garantir que seule votre correspondant pourra le déchiffrer (ou le décrypter).

Le courrier électronique, ou courriel par contraction, est un service de transmission de messages envoyés électroniquement via un réseau informatique (principalement l'Internet) dans la boîte aux lettres électronique d'un destinataire choisi par l'émetteur.

Lors de son acheminement, un email est relayé par un certain nombre de serveurs où il se retrouve donc copié. Et derrière ces serveurs, ce sont autant d'entreprises commerciales ou d'administratrices curieuses qui peuvent, bien que la loi défende les correspondances privées, fouiner dans vos courriers. Les mails peuvent également être interceptés lors de leurs transferts.

Le chiffrement est réalisé par l'émetteur en utilisant la clé publique du destinataire. En réalité le chiffrement du message est réalisé par un algorithme symétrique, beaucoup plus rapide. La clé publique du destinataire est utilisée pour chiffrer la clé secrète de chiffrement du message. Dans ce cas, seul le destinataire peut récupérer la clé secrète, mais celle-ci est connue de l'émetteur du message.

A partir de ce mécanisme nous proposons une petite amélioration sur ce mécanisme afin d'augmenter le niveau de sécurité et difficulté l'attaque, cette amélioration réside d'ajouter une nouvelle clef d'algorithme symétrique (Rijndael dans notre cas) pour protéger la clef de session avons le crypté avec la clef publique de destinataire.

Pour ce but, nous allons concevoir un logiciel de messagerie électronique permettant de crypter les emails sortant et décrypter les emails entrants selon l'amélioration précédente.

Le besoin de communications sécurisés sur l'internet, et le vaste stade d'utilisation des messageries électroniques, étaient les motivations majeures à aborder ce travail (client de messagerie électronique) pour le but de sécuriser la communication au cours de l'envoi et de la réception des emails.

L'objectif de ce travail était d'étudier et concevoir un logiciel de messagerie électronique pour la sécurité des courriers électroniques, fonctionne sur le réseau internet, et aussi permettre de la génération des clefs (publiques et privés) des cryptages.

La structure de ce mémoire s'articule autour de quatre chapitres :

Chapitre 01 : présente les définitions et les fondements théoriques de la cryptographie, passe sur les algorithmes modernes de cette dernière et on terminera ce chapitre par la définition de fonction de hachage et le codage de l'information.

Chapitre 02 : concerne la résolution du problème sujet de l'étude : traite les courriers électroniques et sa structure ainsi que les protocoles utilisés lors de l'envoi et la réception des mails, en passant sur l'état de l'art et notre modeste contribution.

Chapitre 03 : la conception de notre logiciel en utilisant UML comme langage de modélisation.

Chapitre 04 : concerne la réalisation de notre logiciel par des interfaces homme machine avec des tests dans le but de faciliter l'utilisation de notre logiciel.

CHAPITRE 1

Chapitre 1

La Cryptographie

1. Introduction

Le cryptage est historiquement l'une des premières applications de l'informatique. Ce domaine, qui était il y a encore quelques années, réservé aux militaires et aux grandes entreprises, concerne aujourd'hui tous ceux qui souhaitent transmettre des données protégées, qu'ils soient professionnels ou particuliers.

Dans ce chapitre, nous allons présenter quelques définitions de bases puis présenter les deux grandes catégories de cryptographie (symétrique et asymétrique) et quelque exemple de chaque catégorie passant sur les fonctions de hachage et le codage des informations.

2. Définitions de bases

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

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 **clef**.

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 clair sans connaître la clef de déchiffrement. [11]

Le principe de ce qui vient d'être énoncé peut se résumer ainsi :

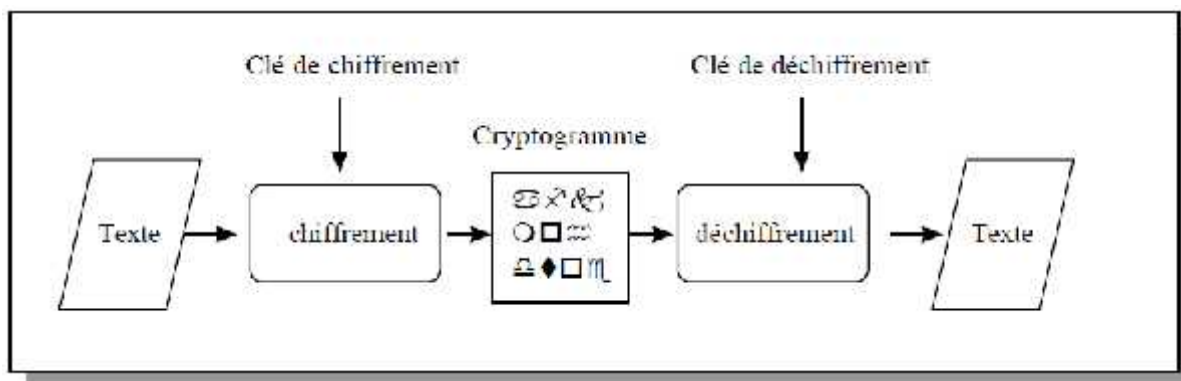


Figure 1.1 : Principe de chiffement/déchiffement.

On conçoit donc facilement que dans un environnement non sûr où les usagers peuvent être multiples, le chiffement apporte un niveau de sécurité supplémentaire.

La sécurité des communications, pourvu que la qualité du chiffement soit suffisante, est alors garantie, même si une tierce personne intercepte le trafic échangé. [5]

Note : Les termes “cryptage” et “crypter” sont des anglicismes, dérivés de l’anglais to encrypt, souvent employés incorrectement à la place de chiffement et chiffrer. En toute rigueur, ces termes n’existent pas dans la langue française. Si le “cryptage” existait, il pourrait être défini comme l’inverse du décryptage, c’est-à-dire comme l’action consistant à obtenir un texte chiffré à partir d’un texte en clair sans connaître la clef. [11]

3. Les objectifs de Chiffement

- assurer la confidentialité des données: «La confidentialité est la propriété qu’une information n’est ni disponible ni divulguée aux personnes, entités ou processus non autorisés ». En général, l’information n’est disponible et partagée qu’entre les deux parties de confiance que sont l’émetteur et le récepteur définis dans le cadre d’un échange,
- assurer l’intégrité des données : « L’intégrité est la prévention d’une modification non autorisée de l’information ».
- assurer l’authentification : consiste à vérifier l’identité des différentes parties impliquées dans le dialogue. L’authentification préserve de l’usurpation d’identité et participe de la confidentialité dans le sens où elle assure que celui qui émet est bien l’entité attendue.
- assurer le non répudiation : permet de prouver qu’un message a bien été émis par son initiateur. Le message ne peut donc plus ensuite être dénié par celui qui l’a émis. [5]

4. Systèmes à clé privée

Les systèmes de cryptage à clé secrète, appelés aussi systèmes de cryptage symétrique ou cryptage conventionnel, sont utilisés depuis plusieurs siècles déjà. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique.

Voici quel en est son 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.

Dans la majorité des systèmes de cryptage symétrique la clé de chiffrement et la clé de déchiffrement sont identiques. La taille des clés utilisées varient selon le besoin et font en standard 64 ou 128 bits.

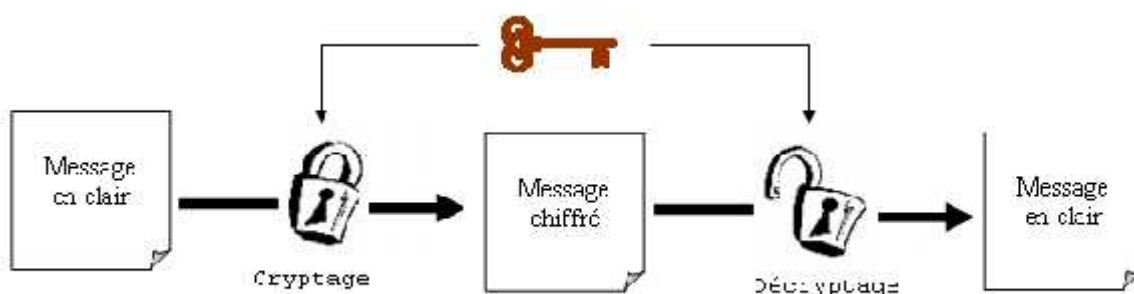


Figure 1.2 : Chiffrement symétrique

Le cryptage conventionnel comporte un avantage majeur : sa rapidité. Il est particulièrement adapté à la transmission par des moyens de transmission sécurisés. Il comporte par contre aussi des faiblesses.

Ce système nécessite la connaissance de la clé par l'émetteur et par le destinataire. C'est la transmission de cette clé entre les intervenants qui représente la faiblesse inhérente au système. S'ils se trouvent à des emplacements géographiques différents, ils devront faire confiance à une tierce personne ou un moyen de communication sécurisé. Toute personne interceptant la clé lors d'un transfert peut ensuite lire, modifier et falsifier toutes les informations cryptées ou authentifiées avec cette clé.

De la norme de cryptage de données DES au code secret de Jules César, la distribution des clés reste le problème majeur du cryptage conventionnel. (Autrement dit, comment faire parvenir la clé à son destinataire sans qu'aucune personne ne l'intercepte ?) Les moyens à déployer pour garantir la distribution sécurisée des clés entre les correspondants sont très onéreux, ce qui constitue un inconvénient supplémentaire.

Les principaux algorithmes à clé privée utilisés actuellement sont : Blowfish, DES / 3DES, IDEA, RC2, RC5, RC6, Rijndael. [2,3]

Les principaux types de cryptosystèmes à clefs privés utilisés aujourd'hui se répartissent en deux grandes catégories : les cryptosystèmes par flots et les cryptosystèmes par blocs.

4.1. Cryptosystèmes 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 clef est donc égale à la taille du message.

4.1.1. Système de Vigenère

Un autre système de cryptographie des plus anciens est cette fois-ci, la substitution polyalphabétique, qui utilise plusieurs alphabets décalés pour crypter un message.

L'algorithme de substitution polyalphabétique le plus connu est le chiffre de Vigenère, mis au point par Blaise de Vigenère en 1586, qui fut utilisé pendant plus de 3 siècles. Son chiffre consiste à utiliser le chiffre de César, mais en changeant le décalage à chaque fois. Il utilise alors un carré composé de 26 alphabets alignés, décalés de colonne en colonne d'un caractère.

Il place également au-dessus de ce carré, un alphabet pour la clef et à sa gauche un autre alphabet pour le texte à coder. Il suffit alors, pour chiffrer un message, de choisir un mot de longueur quelconque, de l'écrire sous le message à coder (de façon répétée s'il le faut) et de regarder dans le tableau l'intersection de la lettre à coder et de la lettre de la clef.

Pour mieux comprendre le fonctionnement du Carré de Vigenère nous vous proposons cet exemple :

Supposons que nous voulons coder le texte « CARRE DE VIGENERE » avec la clef « MALICE ». On commence par écrire la clef sous le texte à coder :

C	A	R	R	E		D	E		V	I	G	E	N	E	R	E
M	A	L	I	C		E	M		A	L	I	C	E	M	A	L

Figure 1.3 : carre de vigenere

Pour chiffrer la lettre C, la clef est donnée par la lettre M. On regarde dans le tableau l'intersection de la ligne donnée par le C, et de la colonne donnée par le M. On trouve O. Puis on continue, jusqu'à ce qu'on ait fini de chiffrer notre texte. En chiffrant le texte « Carre de Vigenere », on obtient donc le texte « OAUZG HG VTOGRQRP ».

Cet algorithme de cryptographie ainsi que celui de César sont les premiers des algorithmes à clef privée. [4]

Note : L'avantage de cet algorithme est cependant que le nombre de clés est en principe infini, et qu'un espion éventuel ne connaît pas la longueur de la clé. [12]

4.2. Cryptosystèmes par blocs

La deuxième classe de cryptosystèmes utilisée aujourd'hui est celle des cryptosystèmes par blocs. Dans ce mode de cryptage, le texte clair est fractionné en blocs de même longueur à l'aide d'une clef unique. Les algorithmes de chiffrement par blocs sont en général construits sur un modèle itératif. Ce modèle emploie une fonction F qui prend en paramètres une clef k et un message de n bits. F est répétée un certain nombre de fois, on parle de ronde. A chaque ronde, la clef k utilisée est changée et le message que l'on chiffre est le résultat de l'itération précédente.

$$C_1 = F(k_1, M)$$

$$C_2 = F(k_2, C_1)$$

...

$$C_r = F(k_r, C_{r-1})$$

Emetteur et destinataire se partagent une clé K secrète. L'algorithme qui engendre les clefs k_i à partir de K se nomme l'algorithme de cadencement des clefs.

La fonction F doit être inversible, ce qui veut dire qu'il faut pour toute clef k et message M pouvoir recalculer M à partir de $F(k, M)$, sinon le déchiffrement est impossible et on ne dispose pas d'un algorithme utilisable. C'est-à-dire qu'il existe une fonction G vérifiant $G(k, F(k, M)) = M$ et que F est une permutation.

La sécurité d'un algorithme de chiffrement par blocs réside principalement dans la conception de l'algorithme de cadencement des clefs et la robustesse de la fonction F. Si l'algorithme de cadencement est mal élaboré, les k_i peuvent être déductibles les unes des autres. La fonction F doit donc être difficile à inverser sans connaître la clef k ayant servi dans le calcul de $C=F(k,M)$. En d'autres termes, connaissant seulement C, F et G, on ne doit pouvoir retrouver le message M seulement en effectuant une recherche exhaustive de la clef.

Les caractéristiques de ces systèmes sont en général liées à leur très forte sensibilité à la dépendance inter-symboles, ainsi qu'à leur mécanisme de propagation d'erreurs. Toute erreur commise sur un bloc de texte clair ou chiffré peut perturber gravement le chiffrement/déchiffrement de ses voisins. [4]

4.2.1. Rijndael

En 1997, le NIST (National Institute of Standards and Technology) fait un appel d'offre pour l'élaboration d'un nouveau système cryptographique car le 3DES ne peut pas constituer une solution à long terme. La sécurité du nouvel algorithme doit être supérieure ou égale à 3DES mais plus efficace. Le chiffrement se fait par blocs de 128 bits, au lieu des 64 octets dans DES. Les clés supportées auront 128, 192 ou 256 bits. Le 15 juin 1998, date de la fin des candidatures, 15 projets sont retenus. Pendant deux ans, les algorithmes sont évalués par des experts, avec un forum de discussion sur Internet et des organisations de conférences. Au deuxième tour, en août 1999, il ne reste plus que 5 finalistes : MARS (IBM), RC6 (RSA Laboratories), Rijndael (Contraction des noms des deux inventeurs belges : Dr. Joan Daemen, Dr. Vincent Rijmen de l'Université Catholique du Louvain, Belgique), Serpent, Twofish.

Le 2 octobre 2000, le NIST retient l'algorithme de Rijndael à cause du bon compromis entre sécurité, performance, efficacité, facilité d'implémentation et flexibilité. Rijndael travaille par blocs de 128 bits et est symétrique (Alice et Bob utilisent la même clé qui doit rester secrète). La taille de la clé est généralement de 128 bits avec les variantes de 192 et de 256 bits. [1]

I. Présentation de l'algorithme

L'algorithme se présente en deux temps, tout d'abord une procédure d'expansion de la clef, puis la fonction principale de chiffrement.

La fonction de chiffrement se divise en trois : une transformation initiale avec la clé, une série de tours puis une transformation finale.

Le nombre de tours s'établit en fonction de la taille des blocs et de la clé :

- 9 tours si la taille des blocs et de la clé sont de 128 bits,
- 11 tours si la taille des blocs ou de la clé est de 192 bits,
- 13 tours si la taille des blocs ou de la clé est de 256 bits.

Pour comprendre le fonctionnement de Rijndael, il paraît judicieux de commencer par l'envisager dans son ensemble (schéma bloc). Les différentes opérations seront détaillées successivement par la suite. [7]

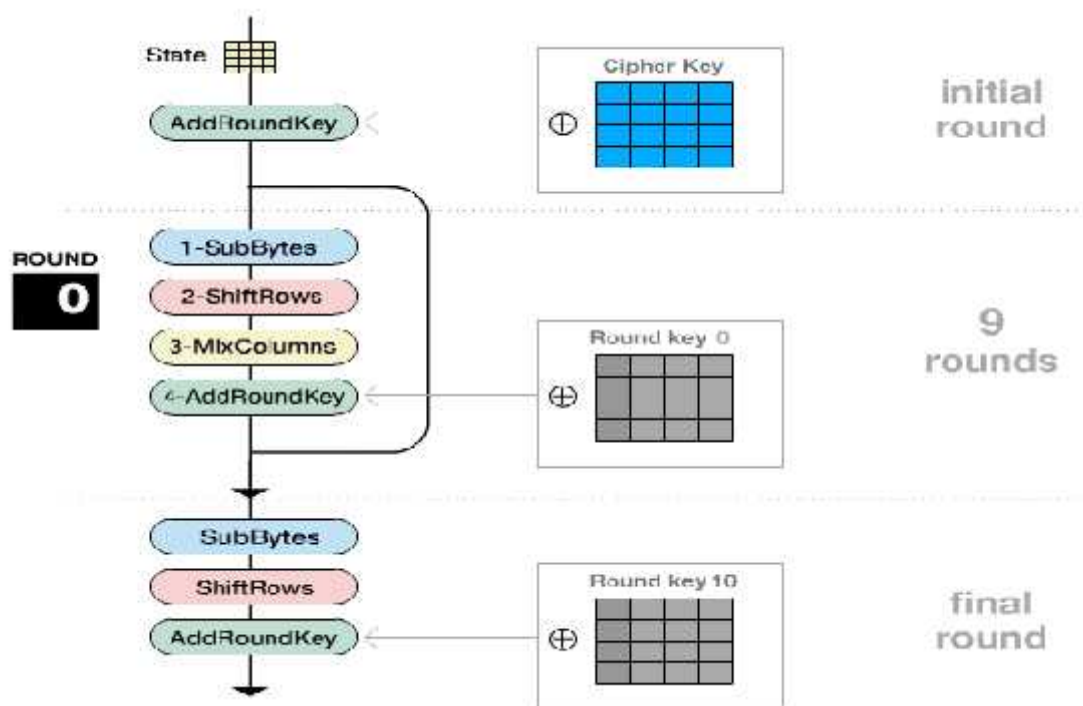


Figure 1.4 : Schéma bloc de l'algorithme Rijndael avec une clé de 128 bits

Initial Round : C'est la première étape et la plus simple car elle ne compte qu'une seule opération, AddRoundKey. Le texte à chiffrer est découpé en blocs de 128 bits et est disposé sous forme matricielle. Les matrices ont 4*4 éléments et chaque élément comporte 8 bits (1 octet), donc chaque bloc comporte 4*4*8=128 bits.

Nous allons représenter les chiffres sous forme hexadécimale, donc il suffit de représenter la suite de chiffres sous forme matricielle :

32	88	31	E0
43	5A	31	37
F6	30	98	07
A8	8D	A2	34

Qui représente le texte en clair (State). Maintenant prenons une clé codée sur 128 bits, en notation hexadécimale que nous représenterons sous forme de matrice (key):

2B	28	AB	09
7E	AE	F7	CF
15	D2	15	4F
16	A6	88	3C

Round 0 – Round 9 (Rondes 0 à 9)

Cette partie du processus de chiffrement dépend de la taille de la clé utilisée. Comme on envisage la version 128 bits de l'algorithme, cette deuxième étape compte 9 itérations.

Chacune de ces 9 itérations effectue successivement les quatre opérations détaillées ci-dessous.

1-SubBytes

Cette opération consiste à remplacer chaque octet de la matrice State par une autre valeur. La substitution se fait à l'aide d'une table appelée S-Box. Les octets que cette table contient sont les octets de remplacement.

hex		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	cc	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	a1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tableau 1.1 : S-Box

La substitution se fait de la manière suivante :

1. Prendre la matrice state trouvée à la sortie de la ronde initiale (initial round).
2. Pour chaque élément de cette matrice, procéder comme suit :
 - Le premier caractère hexadécimal de l'élément indique une ligne de la S-Box tandis que le deuxième indique une colonne.
 - L'octet se trouvant à l'intersection ligne-colonne dans la S-Box est celui qui doit être substitué à celui de la matrice State.

Exemple : on a 19 comme premier élément donc 1 représente la ligne et 9 la colonne. La valeur de remplacement du premier élément sera donc d4. Le deuxième élément vaut A0 et sera remplacé par E0. Ainsi la matrice state issue de cette opération (SubBytes) est la suivante:

4A	E0	B8	1E
27	BF	B4	41
11	98	5D	52
AE	F1	E5	30

2-ShiftRows

Cette opération consiste à décaler des lignes dans la matrice State. De faibles changements dans le texte clair impliquent de grands changements dans le texte chiffré. Les décalages ne modifient pas les valeurs des bytes, mais changent leur ordre.

Les décalages se font comme suit :

- La première ligne n'est pas décalée.
- La deuxième ligne est décalée de 1 octet vers la gauche.
- La troisième ligne est décalée de 2 octets vers la gauche.
- La quatrième ligne est décalée de 3 octets vers la gauche.

La matrice State issue de cette opération (ShiftRows) est la suivante :

D4	E0	B8	1E
BF	B4	41	27
5D	52	11	98
30	AE	1F	5E

3-MixColumns

Cette opération consiste à multiplier une matrice constante avec la matrice State :

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} D4 & E0 & B8 & 1E \\ BF & B4 & 41 & 27 \\ 5D & 52 & 11 & 98 \\ 30 & AE & F1 & E5 \end{bmatrix} = \begin{bmatrix} 04 & E0 & 48 & 28 \\ 66 & CB & F8 & 06 \\ 81 & 19 & D3 & 26 \\ E5 & 9A & 7A & 4C \end{bmatrix}$$

4- AddRoundKey

Lors du processus de chiffrage, Rijndael transforme la clé (matrice Key). A chaque itération (Round), une matrice Key différente est utilisée (RoundN Key). Ceci permet d'éliminer les attaques liées à la clé en faisant disparaître la symétrie. Pour obtenir les 10 nouvelles clés nécessaires (puisque la version 128 bits de l'algorithme compte en tout 11 rondes (Rounds)), Rijndael procède à une opération appelée Key Scheduling ou Key Expansion.

Même si la clé change à chaque ronde, l'opération AddRoundKey reste simple puisqu'elle consiste, comme celle de la première étape, à additionner modulo 2 (XOR) la matrice State et la matrice Key de la ronde en cours (Round N Key).

Final round

Cette étape est quasiment identique à l'un des neuf rondes de la deuxième étape. La seule différence est que, dans cette dernière ronde, l'opération MixColumns n'est pas effectuée.

II. Extension de la clé – Key Expansion

Il s'agit ici de voir comment Rijndael opère pour déduire les 10 clés secondaires dont il a besoin pour chiffrer le texte.

Matrice Key (clé)

2B	28	AB	09
7E	AE	F7	CF
15	D2	15	4F
16	A6	88	3C

On considère encore une matrice *Rcon* (Round constant word array) construite de la façon suivante : $Rcon[j] = (RC[j], 0, 0, 0)$ avec $RC[1] = 1$, $RC[j] = 2 \cdot RC[j-1]$

Les valeurs $RC[j]$ en hexadécimal sont les suivantes :

J	1	2	3	4	5	6	7	8	9	10
RC(j)	01	02	04	08	10	20	40	80	1B	36

Ainsi la matrice *Rcon* vaut

01	02	04	08	10	20	40	80	1B	36
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Les quatre vecteurs composant la matrice Key sont appelés Words (ce sont des mots de 32 bits).

On commence par calculer le cinquième vecteur (W_i) :

W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
2B	28	AB	09	
7E	AE	F7	CF	
15	D2	15	4F	
16	A6	88	3C	

On prend le vecteur W_{i-1} auquel on applique une opération appelée *RotWord* qui consiste en un simple décalage des quatre bytes du vecteur vers le haut.

W_{i-1}	$RotWord(W_{i-1})$
09	CF
CF	4F
4F	3C
3C	09

Au résultat on applique encore l'opération *SubBytes* (avec le S-Box):

RotWord(W_{i-1})	Subbyte(RotWord(W_{i-1}))
CF	8A
4F	84
3C	EB
09	01

Le vecteur obtenu doit encore être additionné (mod 2) avec le vecteur W_{i-4} ainsi que le premier vecteur de la matrice $Rcon$: $W_{i-4} \oplus Rcon(\text{colonne1}) \oplus \text{SubBytes}\{\text{RotWord}(W_{i-1})\}$:

$$\begin{bmatrix} 2B \\ 7E \\ 15 \\ 16 \end{bmatrix} \oplus \begin{bmatrix} 01 \\ 00 \\ 00 \\ 00 \end{bmatrix} \oplus \begin{bmatrix} 8A \\ 84 \\ EB \\ 01 \end{bmatrix} = \begin{bmatrix} A0 \\ FA \\ FE \\ 17 \end{bmatrix}$$

Il constitue le premier des quatre vecteurs de la deuxième clé :

W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
2B	28	AB	09	A0
7E	AE	F7	CF	FA
15	D2	15	4F	FE
16	A6	88	3C	17

Le deuxième vecteur de la deuxième clé s'obtient plus simplement que le premier. En effet, il est donné par $W_i = W_{i-4} \oplus W_{i-1}$:

W_{i-5}	W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
2B	28	AB	09	A0	88
7E	AE	F7	CF	FA	54
15	D2	15	4F	FE	2C
16	A6	88	3C	17	B1

Le troisième vecteur de la deuxième clé s'obtient de la même manière que le deuxième. On a donc à nouveau $W_i = W_{i-4} \oplus W_{i-1}$:

W_{i-6}	W_{i-5}	W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
2B	28	AB	09	A0	88	23
7E	AE	F7	CF	FA	54	A3
15	D2	15	4F	FE	2C	39
16	A6	88	3C	17	B1	29

Le dernier vecteur de la deuxième clé s'obtient de la même manière que les deuxième et troisième vecteurs, $W_i = W_{i-4} \oplus W_{i-1}$:

W_{i-7}	W_{i-6}	W_{i-5}	W_{i-4}	W_{i-3}	W_{i-2}	W_{i-1}	W_i
2B	28	AB	09	A0	88	23	2A
7 ^E	AE	F7	CF	FA	54	A3	6C
15	D2	15	4F	FE	2C	39	76
16	A6	88	3C	17	B1	29	05

Les quatre vecteurs de la deuxième clé sont maintenant définis. La matrice Key a doublé de taille. Elle contient pour l'instant les clés de la ronde 0 et 1. Les vecteurs W_{i-7} à W_{i-4} constituent la clé de la ronde 0 et les vecteurs W_{i-3} à W_i ceux de la clé de la ronde 1. Il reste à faire 9 fois l'intégralité de cette démarche d'extension de la clé pour trouver les 9 autres clés secondaires.

Remarque :

En termes décimaux, ces différentes tailles possibles signifient concrètement que:

3.4×10^{38} clés de 128-bit possibles

6.2×10^{57} clés de 192-bit possibles

1.1×10^{77} clés de 256-bit possibles [21]

III. Déchiffrement

Toutes les opérations réalisées lors du chiffrement sont réversibles. Le générateur de clés de ronde fonctionne exactement de la même manière. Il suffira de reprendre l'algorithme en sens inverse, avec les fonctions inverses.

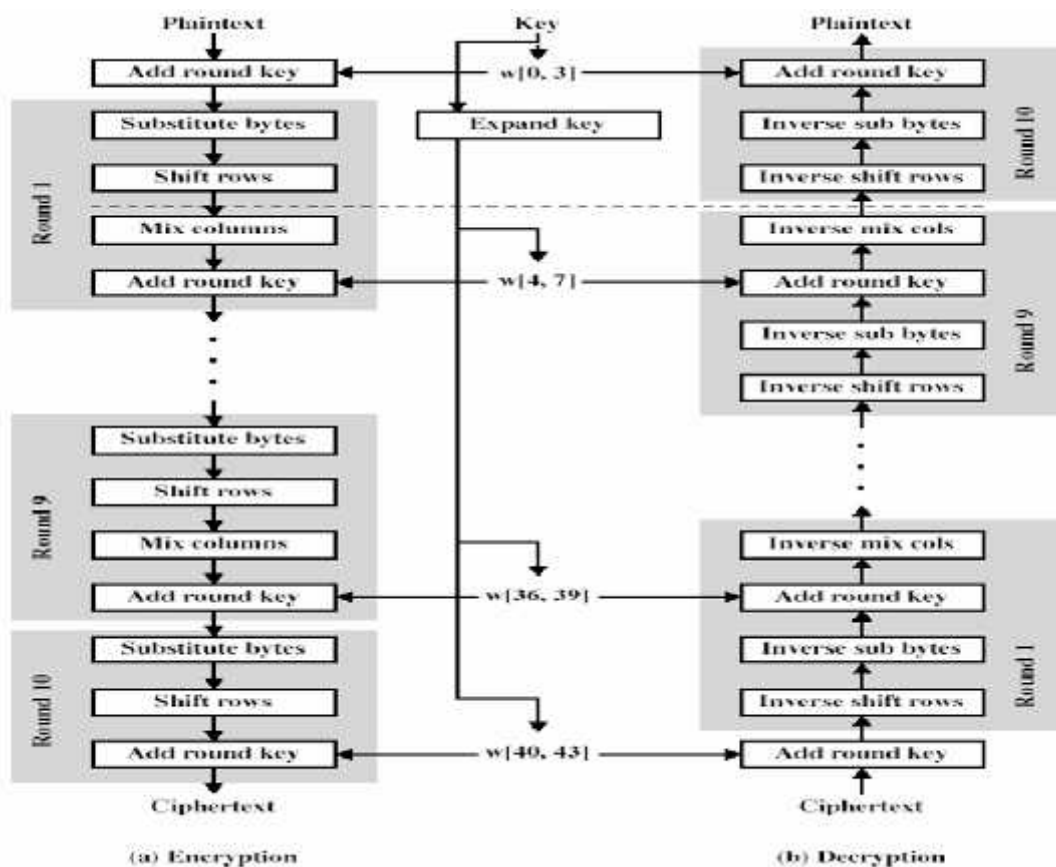


Figure 1.5 : Schéma de déchiffrement d'algorithme Rijndael

- Bob procède à l'extension de la clé de la même manière qu'Alice l'a fait lors du chiffrement.
- Les additions modulo 2, ou \oplus , effectuées lors de l'opération AddRoundKey sont réversibles (en effet, $(A \oplus B) \oplus B = A$).
- L'opération SubBytes est inversée en utilisant la table S-Box inverse (Inverse S-Box). Si par exemple la S-Box indique le byte F7 (ligne 2, colonne 6), alors la Inverse S-Box restituera le byte 26 (ligne F, colonne 7).
- Les décalages de l'opération ShiftRows sont inversés, c'est-à-dire effectués vers la droite.
- La multiplication matricielle de l'opération MixColumns nécessite une inversion de la matrice (qui rend le déchiffrement plus lent que le chiffrement).

Une fois la matrice inverse obtenue, la manipulation est la même que pour l'opération MixColumns faite lors du chiffrement.

En conclusion on peut dire que l'utilisation de la S-Box constitue une réelle difficulté pour les cryptanalystes. L'opération MixColumns combinée avec ShiftRows fait que, après les nombreuses rondes, tous les bits de sortie dépendent de tous les bits d'entrée. Ceci rend la cryptanalyse difficile. L'utilisation des clés secondaires construites par extension de la clé originale, quant à elle, complique les attaques liées à la clé en cassant les symétries. Rijndael est un algorithme sûr et va probablement le demeurer encore une vingtaine d'années. [6,7]

5. Le chiffrement asymétrique (ou chiffrement à clefs publiques)

Les chiffrements asymétriques utilisent un système de paires de clefs (des bi-clefs). L'utilisateur génère une clef aléatoire dite clef privée qu'il est seul à connaître. De cette clef est déduite la seconde, appelée clef publique, que l'utilisateur distribue par exemple via un serveur de clefs.

L'expéditeur qui souhaite envoyer un message chiffré au destinataire doit récupérer la clef publique du destinataire sur le serveur. Puis il chiffre le document avec cette clef comparable à un cadenas. Lorsque le destinataire reçoit le document, il le déchiffre grâce à sa clef privée, la seule clef capable d'ouvrir le cadenas.

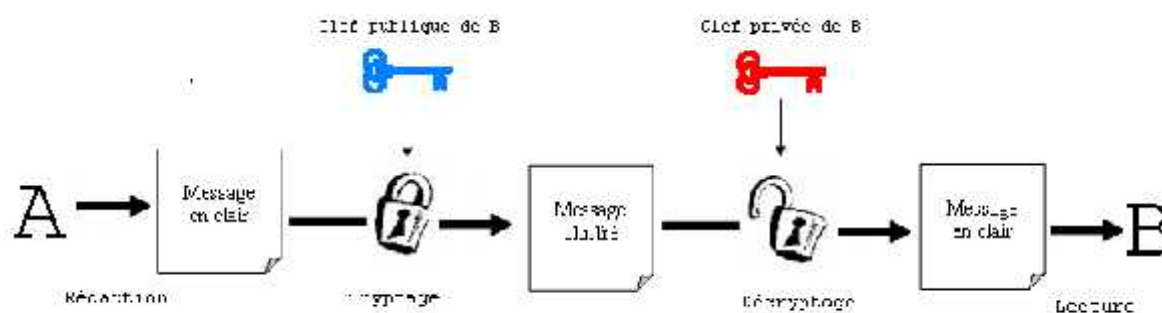


Schéma 1.6 : Chiffrement asymétrique. [3]

Avec le déchiffrement asymétrique le problème de la distribution de la clef est résolu. Cependant cette technique est moins rapide et nécessite de s'assurer que la clef récupérée est bien celle du destinataire du message. [3]

5.1. Le système RSA

5.1.1. Présentation du cryptage RSA

Le cryptage RSA, du nom de ses concepteurs, *Ron Rivest*, *Adi Shamir* et *Leonard Adleman*, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 au Massachusetts Institute of Technology.

Un chiffrement asymétrique est un cryptage où l'algorithme de chiffrement n'est pas le même que celui de déchiffrement, et où les clés utilisées sont différentes. L'intérêt est énorme : il n'y a plus besoin de transmettre la clé à son destinataire, il suffit de publier librement les clés de cryptage. N'importe qui peut alors crypter un message, mais seul son destinataire, qui possède la clé de décodage, pourra le lire. En quelques années, RSA s'est imposé pour le cryptage comme pour l'authentification et a progressivement supplanté son concurrent, le DES.

Le RSA est basé sur la théorie des nombres premiers, et sa robustesse tient du fait qu'il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers. Alors qu'il est facile de multiplier deux nombres premiers, il est très difficile de retrouver ces deux entiers si l'on en connaît le produit. [8]

5.1.2. Principe mathématique

La preuve de l'existence de tels nombres repose sur un résultat de Léonard Euler, établi vers 1760 et généralisant le petit théorème de Fermat :

Soit m un entier strictement positif, produit de nombres premiers distincts (m est sans carré). Si $\phi(m)$ est le nombre des entiers a premiers avec m et tels que $0 \leq a < m$, alors pour tout multiple r de $\phi(m)$ et pour tout entier x on a $x^{r+1} \equiv x[m]$.

Ainsi, si l'on peut décomposer $r+1$ en deux facteurs différents ($r+1=k \cdot u$), on peut alors construire deux fonctions réciproques l'une de l'autre $f(x) \equiv x^k[m]$ et $g(y) \equiv y^u[m]$. En outre, si $j(m)$ n'est pas connu, nous allons voir qu'il est difficile de trouver X seulement en connaissant N et P . [9]

5.1.3. Génération des clés

Le RSA fonctionne à partir de deux nombres premiers, que l'on appellera p et q . Ces deux nombres doivent être très grands, car ils sont la clé de voûte de notre cryptage.

Aujourd'hui, on utilise des clés de 128 à 1024 bits, ce qui représente des nombres décimaux allant de 38 à 308 chiffres !

Une fois ces deux nombres déterminés, multiplions-les. On note n le produit $n = p \times q$, et $z = (p - 1) \times (q - 1)$.

Cherchons maintenant un nombre e (inférieur à z), qui doit nécessairement être premier avec z . Calculons ensuite l'inverse de e modulo z , que nous noterons d .

$$d \equiv e^{-1} \pmod{(p-1)(q-1)}$$

Le couple (e, n) est la clé publique, et (d, n) est la clé privée.

5.1.4. Cryptage et décryptage

Pour crypter un nombre, il suffit de le mettre à la puissance e . Le reste modulo n représente le nombre une fois crypté.

$$c = t^e \pmod{n}$$

Pour décrypter, on utilise la même opération, mais en mettant à la puissance d :

$$t = c^d \pmod{n}$$

Une fois e , d et n calculés, on peut détruire p , q et z , qui ne sont pas nécessaires pour crypter et décrypter. Pire encore, on peut calculer très rapidement la clé privée d à partir de p et q , il ne faut donc pas conserver ces nombres.

Note : En général, la clé privée est ensuite cryptée à l'aide d'un cryptage symétrique.

Cela permet de la conserver de façon sûre, car la clé utilisée par le cryptage symétrique n'a pas à être transmise, et donc ne risque pas d'être interceptée. [8]

5.1.5. Exemple

Bob choisit $p = 17$ et $q = 19$, $n = p \times q = 323$ et $e = 5$.

Sa clé privée est alors $d=173$ car $173 \times 5 = 1 \pmod{(17 \times 19)}$

Supposons qu'Alice veuille lui envoyer le message « BONJOUR » en se servant du tableau suivant pour transformer les lettres en nombre :

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

Cela donne :

B	O	N	J	O	U	R
2	15	14	10	15	21	18

Après avoir chiffré en remplaçant chaque nombre b par $(b^e \bmod n)$ on obtient :

32	2	29	193	2	89	18
----	---	----	-----	---	----	----

Qu'Alice envoie à Bob.

Bob réalise pour chaque nombre b du message $b^d \bmod n$ pour trouver :

2	15	14	10	15	21	18
B	O	N	J	O	U	R

Qui est bien le message initial.

RSA est basé sur la difficulté de factoriser n . En effet celui qui arrive à factoriser n peut retrouver facilement la clef secrète de Bob connaissant seulement sa clef publique. C'est pourquoi dans la pratique la taille des clefs est au minimum de 512 bits.

6. Les fonctions de hachage

Une fonction de hachage est aussi appelée fonction de hachage à sens unique ou "*oneway hash function*" en anglais. Ce type de fonction est très utilisé en cryptographie, principalement dans le but de réduire la taille des données à traiter par l'algorithme de chiffrement. En effet, la caractéristique principale d'une fonction de hachage est de produire un haché des données, c'est-à-dire un condensé de ces données. Ce condensé est de taille fixe, dont la valeur diffère suivant la fonction utilisée.

Fonctions de hachage usuelles

- MD4 et MD5 (Message Digest) furent développées par *Ron Rivest*. MD5 produit des hachés de 128 bits en travaillant les données originales par blocs de 512 bits.
- SHA-1 (Secure Hash Algorithm 1), comme MD5, est basé sur MD4. Il fonctionne également à partir de blocs de 512 bits de données et produit par contre des condensés de 160 bits en sortie. Il nécessite donc plus de ressources que MD5.
- SHA-2 (Secure Hash Algorithm 2) a été publié récemment et est destiné à remplacer SHA-1. Les différences principales résident dans les tailles de hachés possibles : 256, 384 ou 512 bits. Il sera bientôt la nouvelle référence en termes de fonction de hachage.
- RIPEMD-160 (Ripe Message Digest) est la dernière version de l'algorithme RIPEMD. La version précédente produisait des condensés de 128 bits mais présentait des failles de sécurité importantes. La version actuelle reste pour l'instant sûre; elle produit comme son nom l'indique des condensés de 160 bits.

Un dernier point la concernant est sa relative gourmandise en termes de ressources et en comparaison avec SHA-1 qui est son principal concurrent.

– Tiger : Tiger est une fonction de hachage cryptographique conçue par Ross Anderson et Eli Biham en 1996. Tiger fournit une empreinte sur 192 bits mais des versions sur 128 et 160 bits existent aussi. Ces versions raccourcies prennent simplement les premiers bits de la signature de 192 bits. [10]

7. Le codage des informations

7.1. Pourquoi coder ?

La communication nécessite la compréhension entre les deux entités communicantes. L'émetteur envoie de l'information au récepteur qui doit savoir l'interpréter pour la comprendre. Le codage de l'information est la première étape de toute communication. Ainsi pour comprendre les 0 et les 1 de nos machines, il a bien fallu un codage et décodage pour que l'information arrive jusqu'à nos écrans de manière compréhensible, l'information a été traduite.

7.2. Codage ASCII étendu

Le code ASCII a été mis au point pour la langue anglaise, il ne contient donc pas de caractères accentués, ni de caractères spécifiques à une langue. Pour coder ce type de caractère, le code ASCII à 7 bits ne suffit pas, il faut recourir à un autre codage. Il a donc été étendu à 8 bits (un octet) pour pouvoir coder plus de caractères, d'où son nom code ASCII étendu. Ce code attribue les valeurs 0 à 255 aux lettres majuscules et minuscules, aux chiffres, aux marques de ponctuation et aux autres symboles.

Le code ASCII étendu n'est pas unique et dépend fortement de la plateforme utilisée, d'où tous les problèmes de caractères changés lors d'un passage d'une plateforme à une autre.

7.3. Le codage Base64

Le principe du codage Base 64 consiste à utiliser des caractères US-ASCII (caractères non accentués) pour coder tout type de données codé sur 8 bits.

Les protocoles de courrier électronique ont en effet été prévus à l'origine pour transporter des messages en texte seulement. Or, étant donné la diversité des systèmes de courrier électronique, l'échange de données binaires se traduit la plupart du temps par des transformations du contenu rendant illisible le document original.

Le format Base64, utilisé massivement dans les échanges de courrier électronique, permet ainsi de transmettre n'importe quel document binaire (application, vidéo, fichier audio, etc.) en pièce jointe d'un courrier électronique en les codant à l'aide de caractères classiques.

Le principe du codage Base64 consiste à utiliser 4 caractères imprimables (au format US-ASCII) pour coder un groupe de 3 octets quelconques ($3 \times 8 \text{ bits} = 24 \text{ bits}$). [22]

7.4. Unicode

Le code Unicode est un système de codage des caractères sur 16 bits mis au point en 1991. Le système Unicode permet de représenter n'importe quel caractère par un code sur 16 bits, indépendamment de tout système d'exploitation ou langage de programmation.

Il regroupe ainsi la quasi-totalité des alphabets existants (arabe, arménien, cyrillique, grec, hébreu, latin, ...) et est compatible avec le code ASCII. [25]

7.5. Autres codes existant

Malgré le fait que le code ASCII soit le standard pour le codage de caractères, il en existe bien d'autres. Les plus connus sont le code EBCDIC, développé par IBM, qui permet de coder des caractères sur 8 bits et le code Unicode sur 16 bits mis au point en 1991 qui permet de représenter n'importe quel caractère indépendamment de tout système d'exploitation ou langage de programmation mais qui a le gros désavantage de doubler la taille de n'importe quel fichier de texte. [2]

8. Conclusion

Dans ce chapitre, nous avons présenté quelques définitions de cryptographie, en passant sur l'objectif de chiffrement et les deux grandes catégories de cryptographie (symétrique et asymétrique) et terminé le chapitre par le codage des informations.

En particulier, nous avons présenté l'algorithme de Rijndael par des exemples, et l'algorithme du RSA, en utilisant les deux algorithmes pour le cryptage de messagerie électronique comme présenté dans le chapitre suivant.

CHAPITRE 2

Chapitre 2

Le chiffrement des messageries électroniques

1. Introduction

Internet constitue le plus grand réseau du monde. Beaucoup d'entreprises choisissent de relier leur réseau à Internet pour pouvoir tirer de ses ressources extrêmement intéressantes et des multiples services qu'il offre.

Le courrier électronique est le service le plus connu d'Internet, dans le cadre de l'entreprise il offre l'opportunité fabuleuse de prendre contact avec d'autres utilisateurs, clients et dirigeants. Lors de son acheminement, un email est relayé par un certain nombre de serveurs où il se retrouve donc copié. Et derrière ces serveurs, ce sont autant d'entreprises commerciales ou d'administratrices curieuses qui peuvent, bien que la loi défende les correspondances privées, fouiner dans vos courriers. Les mails peuvent également être interceptés lors de leurs transferts.

Dans ce chapitre nous allons parler sur la définition d'internet puis les courriers électroniques et leur structure passant sur les clients des messageries électroniques et leurs architectures et le défient protocoles utilisées pour l'envoi et la réception des mails, terminer ce chapitre par la précision de l'état de l'art et la contribution qui nous avons ajoutée.

2. Qu'est-ce qu'internet

Un immense réseau de réseaux : Un ensemble d'ordinateurs, interconnectés entre eux par des câbles, liaisons téléphoniques, infrarouges ..., et communiquant grâce à un même langage de communication représente un réseau informatique. L'interconnexion de réseaux est la suite logique de l'évolution de la technologie appliquée à l'informatique.

Tous les ordinateurs, quel que soit leur marque ou leur puissance, ont la possibilité d'échanger des informations en toute liberté. Au cours du temps, le nombre d'ordinateurs interconnectés et leur position sur le globe évoluent, la topologie des réseaux ressemble au

maillage d'un immense filet, ou à une immense toile d'araignée. Une machine (ordinateur ou périphérique de connexion), branchée au réseau, s'appelle un nœud du réseau, ou hôte. Une machine qui offre des services : temps de calcul, documents, base de données, est un serveur. La machine de l'utilisateur qui utilise un service de l'hôte à un moment donné, c'est à dire qui est « servie » est dite cliente.

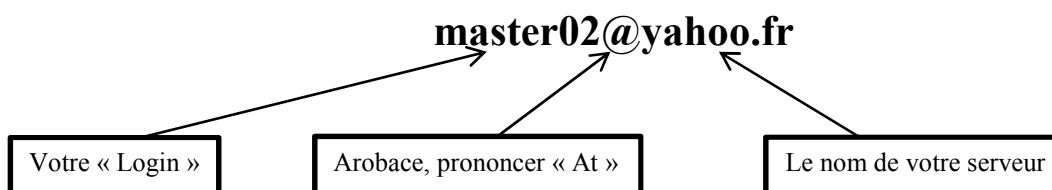
2.1. Comprendre les termes d'internet

Naviguer sur Internet, visiter des sites, envoyer des messages électroniques, dialoguer sur le Chat, etc., nécessitent de comprendre les termes employés sur Internet. Vous trouverez ci-dessous une liste qui propose les termes généraux d'Internet.

- **Autoroutes de l'information** est un terme qui désigne l'ensemble des technologies et des services nécessaires pour acheminer des bases de données, des images, des conversations, des fichiers multimédias, etc., entre les multiples utilisateurs.
- **Cookies** qui sont des petits fichiers placés automatiquement sur votre disque dur lors des visites dans certains sites.
- **Cybercafés** sont des bistrots, des restaurants, etc., qui mettent à votre disposition des ordinateurs et proposent, pour d'accéder à Internet.
- **Internauts** est le nom des personnes naviguant sur Internet.
- **Naviguer** (ou surfer) est un terme qui signifie circuler sur le Web. Cela consiste à suivre des liens hypertexte pour se promener de site en site.
- **Nétiquette** est le « règlement » sur Internet. En effet, les Internauts forment une communauté de personnes qui partagent un même intérêt pour Internet et, qui dit convivialité et échange, dit règles de bienséance. De ce fait, un certain nombre de principes moraux, de réglementations et de mises en forme ont été définis: c'est la netiquette, contraction de net et d'éthique. Par exemple, la nétiquette pose pour principe qu'un message écrit en majuscules signifie que vous criez ou encore qu'il est de bon ton, lorsque l'on utilise l'image d'un site, de citer son adresse Web. [17]

2.2. Structure des adresses

Tout comme le courrier « classique », le courrier électronique répond à certaines normes dans le libellé des adresses. Lorsqu'un administrateur vous crée une boîte aux lettres électroniques, il vous attribue un mot de passe (l'équivalent de votre clé) et un login, c'est à dire un nom. Vous disposerez dès lors d'une adresse électronique qui sera du type :



- ⇒ Dans cet exemple « master02 » correspond au login, c'est à dire le nom qui vous identifie sur votre serveur, en général il correspond à votre nom, votre prénom ou un mélange des deux. Il vous est attribué par l'administrateur du serveur.
- ⇒ Le caractère « @ » est américain, il s'appelle « Arobace » et se prononce "At", il sépare le login de l'adresse du serveur.
- ⇒ « yahoo.fr » correspond au nom d'une machine connectée en permanence à l'Internet et sur laquelle vous êtes identifié. Cette machine est votre serveur de mail, il est enregistré mondialement de telle sorte que n'importe qui peut vous envoyer du courrier depuis n'importe quelle autre machine connectée au réseau mondial. [13]

3. Les courriers électroniques

3.1. Définition de courrier électronique

Le courrier électronique, ou courriel par contraction, est un service de transmission de messages envoyés électroniquement via un réseau informatique (principalement l'Internet) dans la boîte aux lettres électronique d'un destinataire choisi par l'émetteur. L'expression, qui désigne aussi le message, concurrence, sous sa forme abrégée, les emprunts à l'anglais e-mail, email et mail (formes abrégées de « electronic mail »), dont l'emploi, s'il est proscrit dans les documents administratifs en France et au Québec, est largement répandu dans la langue courante. [16]

3.2. Origines

Le courrier électronique existait avant Internet et fut un outil précieux durant la création de celui-ci.

Il prit forme en 1965 en tant que moyen de communication entre utilisateurs d'ordinateurs à exploitation partagée. Le Q32 du SDC et le CTSS du MIT furent les premiers systèmes de messagerie électronique. Ils s'étendirent rapidement en réseau, permettant aux utilisateurs de transmettre des messages à travers différents ordinateurs. Le système AUTODIN pourrait avoir été le premier, en 1966, à autoriser l'échange de courriels entre ordinateurs, le système SAGE avait des fonctionnalités similaires quelque temps auparavant.

Le réseau ARPANET fut une contribution majeure à l'évolution du courrier électronique. Un rapport y indique des transferts de messages inter systèmes peu après sa création, en 1969. En 1972, Ray Tomlinson proposa l'utilisation du signe @ pour séparer le nom d'utilisateur de celui de la machine.

Ses premiers programmes de courriel SNDMSG et READMAIL jouèrent un rôle important dans le développement du courrier électronique, lequel vit sa popularité fortement augmenter grâce à ARPANET. [16]

3.3. Différents types des clients des courriers électroniques

Pour gérer son courrier électronique, un utilisateur pourra utiliser :

- ❖ soit un **webmail** :
 - une application web, c'est à dire un logiciel dont l'interface est un **site web**. Cet interface propose des formulaires pour s'authentifier, consulter sa boîte à lettres, rédiger et envoyer des courriels... : il faudra un navigateur web (Firefox, Internet Explorer...) pour se servir du webmail ;
 - Les Webmails reposent en général sur des protocoles d'accès à des serveurs de messagerie, et appelés aussi Les **Clients légers**.
- ❖ soit un **logiciel client de courrier électronique**. Ce logiciel doit être installé sur son ordinateur. On peut utiliser les deux types parallèlement, par exemple un logiciel de courrier électronique chez soi et le webmail en dehors de chez soi. les logiciels des courriers électroniques appelés aussi **Clients lourds**.

3.4. Quelques exemples du webmails

- *AOL Mail*, maintenant ouvert à tous les internautes. Le Webmail intègre la messagerie instantanée AIM.
- *ContactOffice*, compatible avec les courriels en UTF-8, et en HTML.
- *Free*, le fournisseur d'accès à Internet utilise trois Webmails.
- *Gmail*, compatible avec les courriels en Unicode utf 8, et en HTML, autant en émission qu'en réception. Le Webmail intègre la messagerie instantanée Google Talk et permet de lire directement les documents attaches aux formats les plus courants, même propriétaires.
- *GMX*, webmail leader en Allemagne ayant pris le relais de Caramail suite à la fermeture de ce dernier. Il propose l'import de plusieurs messageries, la synchronisation POP3 et IMAP ainsi que l'envoi de pièces jointes de 50 Mo.
- *Windows Live Hotmail* (ex-MSN Hotmail), avec intégration du service de messagerie instantanée Windows Live Messenger (ex-MSN Messenger).
- *LaPoste.net* est compatible avec l'encodage UTF-8.
- *Opera Web Mail* le service de messagerie d'Opéra, qui utilise le système de messagerie d'Outblaze.
- *Voila*, compatible avec les courriels en Unicode.
- *Yahoo!*, avec intégration du service de messagerie instantanée Yahoo! Messenger.

Avantages web mail:

- Accessible depuis tout ordinateur connecté à Internet.
- Ne nécessite aucune configuration sur l'ordinateur.

Inconvénients :

- Consultation plus lente qu'avec un logiciel client ;
- Consultation d'une seule boîte à lettres ;

- Nécessite d'être connecté en permanence à Internet pour toute opération :
 - Consultation de la boîte
 - Organisation des dossiers de réception
 - Rédaction d'un courrier
 - Gestion du carnet d'adresse
 - Gestion des filtres
- Gestion du carnet d'adresse moins élaborée.
- Gestion des filtres moins élaborée, parfois inexistante.
- Pas de sauvegarde personnelle des données courriers, carnet d'adresses...[18]

3.5.Appellations des clients de messagerie

D'autres appellations couramment utilisées sont : « logiciel de messagerie », « client de courrier électronique », « client courriel », « courrielleur », « client e-mail », ou « MUA » (abréviation de l'anglais Mail User Agent).

3.6.Composants d'un service de courrier électronique [18]

Le service de courrier électronique nécessite :

1. des **serveurs de courrier** électronique capables :
 - de transférer et d'acheminer des messages vers les boîtes à lettres,
 - de recevoir et de stocker ces messages,
2. un **client de courrier** électronique capable :
 - d'envoyer des messages,
 - de relever le contenu de boîtes à lettres.

Pour communiquer, les serveurs et les clients utiliseront différents **protocoles** de communication :

- ❖ protocole de **transfert** de courrier électronique : **SMTP**,
- ❖ protocoles de **réception** de courrier électronique : **POP3** ou **IMAP**.

3.7.Structure d'un système de messagerie

Un système de messagerie contient les composants suivants :

3.7.1. Agent utilisateur (user agent ou UA)

Un agent utilisateur «UA » est un programme que l'utilisateur emploie pour composer son message (logiciel client) et l'envoyer à l'agent de routage (voir ci-dessous) pour l'injecter dans le système de messagerie.

Un **UA** permet également la lecture du courrier, c'est la phase finale. Il y a un UA à chaque extrémité du système de messagerie.

3.7.2. Agent de routage des messages

Un agent de routage reçoit un message. En fonction de l'adresse du destinataire, il décide de faire appel à un agent de transport de messages, dont le but est d'acheminer le message dans la direction du destinataire.

3.7.3. Agent de transport des messages

Un agent de transport reçoit un message et une direction. Sa tâche est d'acheminer ce message à l'endroit indiqué. Un agent de transport de messages est spécialisé pour un type de transmission, par exemple : il y aura un agent de transport pour l'envoi (SMTP : protocole utilisé sur Internet), et un autre pour la réception (POP3: Post Office Protocol version 3)...etc.

3.7.4. Les boîtes aux lettres

Une boîte aux lettres est matérialisée par un fichier (un répertoire ou un fichier de données), La boîte aux lettres est utilisée pour stocker les messages qui arrivent. Chaque abonné possède une adresse de courrier : c'est la boîte aux lettres.

Lorsqu'on interroge une boîte aux lettres, on rapatrie tous les messages qui se trouvent sur le serveur. Lorsqu'on expédie un message à quelqu'un, il sera envoyé dans sa boîte aux lettres.

Remarque :

Dans la terminologie, on utilise le terme agent de transfert de messages **MTA**, pour une notion qui regroupe, les agents de routage et de transport.

La figure suivant montre les différents composants d'un système de messagerie. [19]

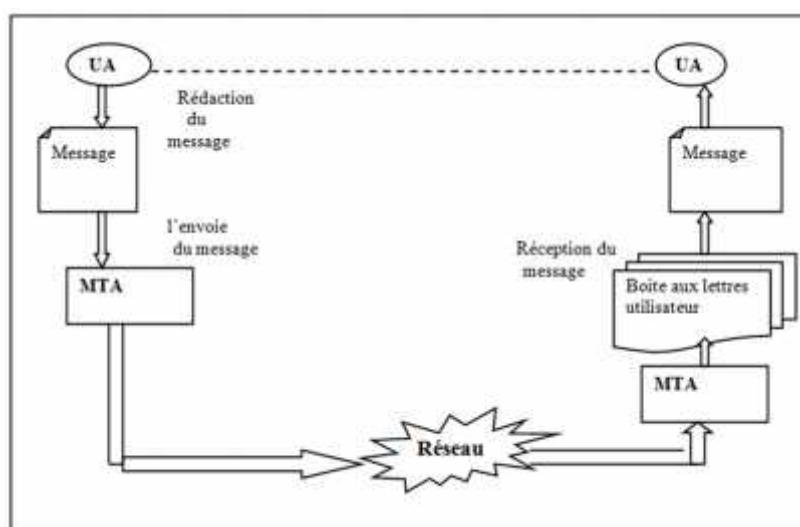


Figure 2.1 : Structure d'un Système de messagerie.

4. Architecture Client/Serveur

De nombreuses applications fonctionnent selon un environnement Client/Serveur, cela signifie que des machines clientes (faisant partie du réseau) contactent un serveur, une machine généralement très puissante en termes de capacités d'entrée-sortie, qui leur fournit des services.

Ces services sont des programmes fournissant des données telles que l'heure, des fichiers et une connexion. Les services sont exploités par des programmes, appelés programmes clients, s'exécutant sur des machines clientes. On parle ainsi de client FTP, de messagerie...etc. Lorsqu'on désigne un programme tournant sur une machine cliente capable de traiter les informations qu'il récupère auprès du serveur (dans le cas du client FTP il s'agit de fichiers, tandis que pour le client de messagerie il s'agit de courrier électronique).

Dans un environnement purement Client/Serveur, les ordinateurs du réseau (les clients) ne peuvent contacter que le serveur, c'est l'un des principaux atouts de ce modèle.

4.1. Notion de base

Le modèle de communication client-serveur est orienté vers la fourniture de service par un processus serveur à un processus client. Un échange consiste donc en la transmission d'une requête à un serveur, qui exécute l'opération demandée et envoie en retour la réponse.

Nous définissons ci-dessous plus précisément ces concepts de bases :

- **Client** : processus demandant d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Serveur** : Processus accomplissant une opération sur demande d'un client et transmettant la réponse à ce client.
- **Requête** : Message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte du client.
- **Réponse** : Message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'opération.

5. Architecture logicielle d'une messagerie

Une application de messagerie va installer sur un poste client cinq modules :

- une interface utilisateur fournissant les commandes permettant l'édition, l'envoi et la réception des messages.
- une interface de service (Service Provider Interface) chargée d'exécuter les commandes transmises par l'interface utilisateur via le système d'exploitation. Ce module contient les

paramètres de fonctionnement de l'outil de messagerie (nom du serveur, protocoles utilisés, planification des tâches).

- un module de stockage des messages contenant les messages des boîtes d'envoi et de réception.
- un module de transport mettant en forme les messages en fonction des protocoles utilisés. Ce module dialogue avec le module TCP/IP des couches 3 et 4 du modèle OSI.
- un annuaire permettant de stocker une liste de destinataires (carnet d'adresses, contacts).

Chaque outil de messagerie installe ses propres modules de service. Il faut noter que les formats de stockage des messages et informations ne sont pas normalisés.

Chaque constructeur choisit son format, avec d'éventuels problèmes de compatibilité avec les autres applications.

Côté serveur de messagerie donne les éléments logiciels installés:

- l'agent de transport des messages (Message Transport Agent) chargé de la réception et de l'envoi des messages avec d'autres serveurs.
- le module de traitement des messages gère l'envoi et la réception des messages des clients, avec la mise au format de stockage.
- l'annuaire contient la liste des BAL gérées par le serveur.
- le module de passerelles intervient pour la mise au format des messages lors de l'envoi vers d'autres serveurs utilisant des protocoles différents.
- le module de stockage des messages des clients (BAL).

On retrouve bien évidemment les modules du système d'exploitation assurant le transport des données (Remote Protocol Control) et les protocoles des couches 3 et 4 du modèle OSI. [14]

5.1. Les logiciels de messagerie

Il faut distinguer les logiciels de serveurs de messagerie et ceux destinés aux postes clients. Il n'est pas indispensable d'utiliser un outil de messagerie du même éditeur que le logiciel utilisé sur le serveur. Les protocoles d'échange normalisés rendent compatibles les différentes versions. Des problèmes tels que l'affichage de certains caractères ou la gestion des pièces jointes ne sont toutefois pas à exclure.

Côté serveur, les logiciels les plus courants sont Exchange Server de Microsoft, Netscape Messaging Server, ou encore Domino Mail Server de Lotus.

Côté client, les outils de messagerie les plus utilisés sont Netscape, Eudora, Mozilla, bien évidemment Outlook Express, Outlook ou Exchange de Microsoft, mais également Notes de Lotus ou Groupwise de Novell.

A noter que des serveurs de liste de diffusion (serveur SYMPA par exemple) permettent de gérer des groupes de destinataires et d'assurer la diffusion d'un message à tous les membres.

L'abonnement et le désabonnement à la liste ou sa consultation s'effectuent automatiquement par l'abonné lui-même ou par l'administrateur de la liste grâce à l'envoi de commandes par courrier électronique. Un gestionnaire est souvent utile pour s'assurer que la liste est à jour.

5.2. Les protocoles de messagerie [14]

Chaque internaute possède une « boîte aux lettres » (BAL) identifiée par une adresse électronique du type « adresse_perso@[sous_domaine].domaine ».

L'acheminement de l'e-mail se fait en plusieurs étapes. Tout d'abord, le courrier est envoyé à un serveur de mail qui va se charger de l'acheminer à bon port.

Il va donc transmettre le message au serveur destinataire qui le stocke en attendant que l'internaute destinataire le récupère à partir de sa boîte aux lettres personnelle.

Contrairement au courrier postal, l'acheminement d'un message électronique est beaucoup plus rapide, et il peut être distribué automatiquement à plusieurs destinataires à la fois.

Différents protocoles applicatifs sont utilisés au-dessus des couches réseaux et transport (TCP/IP) d'Internet (SMTP, POP, IMAP) :

- Le protocole SMTP permet de transmettre les messages envoyés par les postes clients.
- Les protocoles POP3 ou IMAP permettent de dialoguer à partir de postes clients afin d'assurer l'interrogation des boîtes aux lettres et le rapatriement des messages du serveur sur le poste de travail.

5.2.1. SMTP pour la gestion du courrier

Le protocole SMTP (Simple Mail Transfer Protocol) est le plus couramment utilisé pour la gestion du courrier entre serveurs sur Internet, reliés en permanence.

Le format des messages SMTP utilise le caractère «@» comme séparateur du nom de la BAL de celui du serveur de messagerie. Ce dernier utilise le format commun des serveurs sur Internet tel que « mail.yahoo.fr » pour le serveur « mail » du domaine « yahoo.fr ». Ainsi l'adresse de Bernard sur ce serveur aura la syntaxe bernard@mail.yahoo.fr. Cette adresse devra apparaître dans le champ « destinataire » de l'éditeur de messages.

Les éditeurs proposent généralement les champs:

- Champ CC (Carbone Copy ou Copie Conforme) pour l'envoi d'une copie du message aux destinataires dont l'adresse se trouve dans ce champ. Les noms des destinataires apparaîtront sur tous les messages transmis.
- Champ BCC (Blind Carbone Copy) ou CCI (Copie Conforme Invisible) pour l'envoi aux destinataires indiqués dans ce champ. Les destinataires indiqués dans les autres champs ne verront pas ces destinataires dans la liste des destinataires.

Le tableau suivant donne une liste des principaux champs normalisés dans le format de messages SMTP, tels que définis par la RFC 822

Champ	Signification	Champ	Signification
To	Adresse destinataire principal	Date	Date et heure d'émission
Cc	Adresse destinataire secondaire	Reply To	Adresse de réponse
Bcc	Adresse destinataire caché	Message-Id	Numéro d'identification
From	Créateur du message	References	Numéros de messages liés
Received	Adresse émetteur	Subject	Résumé de message
Recived :	Adresse des agents de transfert	Keywords	Mots clés de l'émetteur
Return-Path	Identifie le chemin de retour		

Tableau 2.1 : Champs Normalisés du protocole SMTP.

Les serveurs dialoguent en utilisant des commandes. Les deux tableaux suivant montrent quelques commandes normalisées constituées:

- D'un code de 4 lettres ou plus pour les commandes d'envoi;
- D'un code de 3 chiffres pour les commandes de réponse;
- Le premier chiffre signifie une exécution réussie (1, 2 ou 3) ou non (4 ou 5);
- Les chiffres suivants précisent le code de retour de commande ou la nature de l'erreur.

Commande	Fonction
HELO « exp »	Requête de connexion provenant d'un expéditeur SMTP
MAIL FROM : « adr_exp »	Lancer une transaction de courrier vers une ou plusieurs boîtes aux lettres
RCPT TP : « adr_dest »	Spécifie un destinataire du courrier. Pour plusieurs destinataires, la commande est répétée
DATA	Marque le début des données d'un message La fin est marquée par la séquence <CRLF>.<CRLF>
QUIT	Demande au récepteur l'envoi d'une réponse OK et de fermer la connexion
RESET	Annulation du mail en cours
NOOP	Demande au récepteur l'envoi d'une réponse OK

Tableau 2.2 : Commandes d'envoi du protocole SMTP

Commande	Fonction
250	Action demandée bien effectuée (réponse OK)
251	Utilisateur non-local, message retransmis
354	Commencer à transmettre le mail (fin d'envoi par <CRLF>.<CRLF>)
450	Action demandée non-effectuée, BAL occupée
550	Action demandée non-effectuée, BAL inaccessible
451	Action demandée annulée : erreur pendant le traitement
551	Utilisateur non-local : rediriger le message
452	Action demandée non-effectuée : espace de stockage insuffisant
552	Action demandée non-effectuée : dépassement de quota disque
553	Action demandée non-effectuée : nom BAL illégal
554	Echec de transaction

Tableau 2.3 : Réponses Aux commandes du protocole SMTP

5.2.2. Le protocole ESMTP (Extended Simple Mail Transfer Protocol)

Ce protocole est une amélioration du protocole SMTP avec lequel il est compatible. Il permet de passer davantage de commandes.

La commande d'ouverture « HELO » est remplacée par « EHLO » (Extended HELO). Le destinataire ne répond plus seulement « OK », mais donne aussi la liste des mots-clefs qu'il est capable de traiter. Dans le cas où le destinataire ne supporte pas le protocole ESMTP, il retourne un message d'erreur et la communication continue dans un mode SMTP. [15]

5.2.3. POP3 et IMAP pour interroger la BAL

Ce protocole POP3 (Post Office Protocol) est destiné à récupérer le courrier sur un serveur pour un utilisateur non connecté en permanence à Internet, mais se connectant à travers un réseau d'opérateur de télécommunication tel que le RTC ou le RNIS. Il gère:

- l'authentification du client (vérification du nom et du mot de passe);
- la réception des courriers et fichiers attachés à partir du serveur de messagerie;
- la réception de messages d'erreur ou d'acquiescement.

Ce protocole ne permet pas l'envoi de messages. Il ne permet pas non plus la lecture des messages « en ligne ». Il est nécessaire de télécharger l'intégralité du message et des pièces jointes avant sa lecture. Il ne permet donc pas de manipuler les messages sur le serveur.

Pour lire le courrier « en ligne », il faut utiliser un protocole comme IMAP (Interactive Mail Access Protocol). Il permet également la manipulation sur les messages tels que les recherches selon critères, le tri, l'effacement, ainsi que la création sur le serveur de dossiers publics et privés pour le classement des messages. Les dossiers privés ne sont accessibles qu'à leur créateur; les dossiers publics sont accessibles à tous ou à un groupe de clients. Le protocole IMAP4 utilise le port 143 par défaut.

Les serveurs POP3 dialoguent par le port 110 (port TCP par défaut). Ils utilisent des commandes normalisées définies par la RFC 1939, comportant quatre lettres. Les réponses sont transmises sous forme d'une chaîne de caractères précédée des caractères +OK ou —RR suivant que celle-ci est positive ou négative. Le tableau suivant donne la liste des commandes disponibles sous POP3.

Commande	Fonction
STAT	Récupère le nombre et la taille des messages en attente
LIST (msg)	Demande d'information sur le message spécifié en paramètre (msg)
RETR msg	Récupère une liste de messages
DELE msg	Supprime le message spécifié
USER nom	Spécifie une boîte aux lettres
PASS password	Spécifie un mot de passe
QUIT	Supprime les messages lus et fermer la connexion

Tableau 2.4 : commandes du protocole POP3

Détaillons les particularités de chacun de ces deux protocoles:

- Avec le protocole POP, les messages sont en général effacés du serveur après le téléchargement. L'espace disque nécessaire à chaque client sur le serveur peut être limité, et surtout reste à peu près constant, ce qui simplifie l'administration. Toutefois, en l'absence d'une commande d'effacement (DELE), un double du message est conservé sur le serveur après son téléchargement sur le poste client. Les messages sont rangés sur le poste client dans des dossiers créés localement. Le client ne crée donc pas de dossiers sur le serveur, ni n'effectue de manipulation de fichier. Ceci est vu comme une sécurité par beaucoup d'administrateurs. Pour envoyer un message à plusieurs destinataires, le message est dupliqué en autant d'exemplaires que de destinataires. C'est le cas par exemple dans les équipes de projet dont les membres veulent diffuser une information ou un document.
- Le protocole IMAP laisse les messages sur le serveur de messagerie. L'espace disque de chaque client risque donc de croître, si celui-ci ne fait pas le « ménage » dans ses messages. Les messages sont rangés sur le serveur par le client. Celui-ci a donc la possibilité de créer des dossiers sur le serveur. Un aspect intéressant du protocole consiste en la possibilité de créer des dossiers publics accessibles par un groupe de clients.

Trois cas orientent le choix vers un serveur IMAP:

- la nécessité pour des collaborateurs de consulter leurs messages de plusieurs ordinateurs dans l'entreprise ou hors de celle-ci. Ils doivent alors trouver sur le serveur l'intégralité de leurs messages;
- le souhait de permettre la consultation des messages à partir d'un navigateur web sur les postes clients. Le serveur de messagerie doit alors être interfacé à un module logiciel pour réaliser un Webmail.
- la volonté d'assurer une transmission cryptée des messages. Cette solution permet de bénéficier du mode de transmission sécurisé SSL.

5.2.4. MIME pour la mise en forme des messages

Pendant longtemps, le codage des caractères était laissé au libre choix des éditeurs de logiciels de messagerie. Il s'ensuivait des affichages peu fiables lorsque le message était lu sur un logiciel d'un éditeur différent de celui utilisé pour la création du message. Aujourd'hui, les éditeurs proposent aux utilisateurs plusieurs choix de protocoles. Le plus utilisé actuellement est le protocole MIME (Multipurpose Internet Mail Extension). Ce protocole assure le codage du texte et l'insertion de fichiers joints, qu'ils soient de texte formaté, d'image ou de son. Le protocole MIME reprend le codage ASCII sur 7 bits ou 8 bits (caractères accentués) de la RFC 822 et définit des règles pour le codage de messages non ASCII. Le codage utilisé pour la transmission peut être:

- un codage base64 pour les messages binaires (groupe de 24 bits segmentés en 6 bits et ASCII légal: A pour 0, B pour 1...);
- un codage QP (Quoted Printable) pour les messages texte (codage ASCII sur 7 bits et une séquence spécifique composée du signe égal et de la valeur du caractère pour les codes supérieurs à 127 - par exemple =E9 pour le caractère « é ») ;
- un codage permettant de spécifier le type de fichier et sous-type contenu dans le message (texte, son ou vidéo...).

Ce codage permet la transmission de nombreux types de fichiers.

5.3. Processus d'envoi et de réception d'un courrier électronique

Soit un expéditeur appartenant au **domaine A** qui veut envoyer un courriel à un destinataire appartenant au **domaine B** :

1. L'expéditeur, à l'aide de son **client de courrier**, rédige son courriel et s'adresse à son serveur de courrier (domaine A) pour **envoyer** son courriel ;
2. Le serveur du domaine A **transfère** le courriel au serveur de courrier du domaine B : le message est stocké sur ce serveur ;
3. Le destinataire veut relever son courrier électronique : à l'aide de son **client de courrier** électronique, il **interroge son serveur** de courrier (Domaine B) ;
4. Le client de courrier électronique **récupère le courrier** depuis son serveur de courrier (Domaine B). [18]

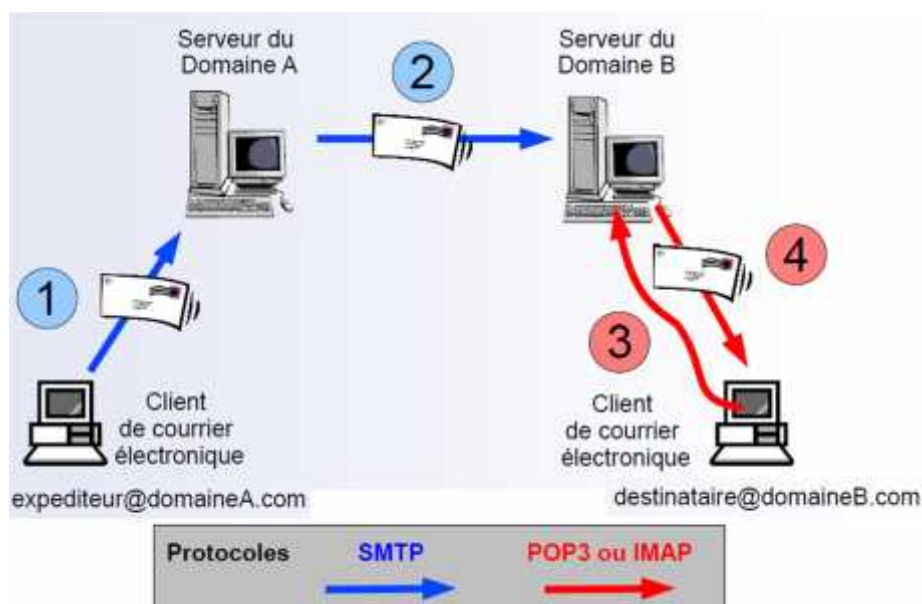


Figure 2.2 : Processus d'envoi et de réception d'un courrier

6. L'état de l'art

6.1. Pourquoi crypter ?

Le cryptage n'est pas une nécessité, sauf si vous avez des informations vraiment confidentielles à protéger, et êtes prêt à organiser votre travail.

- L'interception d'un courrier n'est pas à la portée de tous. Pour pirater votre courrier, il faut disposer d'équipements et de logiciels et avoir une très forte motivation.
- Le cryptage exige du temps, une certaine connaissance technique, et une procédure relativement compliquée, non seulement pour vous, mais aussi pour vos correspondants. Ce type de protection ne s'impose donc que dans les cas où le secret est impératif. Crypter dans d'autres circonstances est franchement une perte de temps. [20]

6.2. Les applications du chiffrement

6.2.1. Le protocole SSL

Le protocole SSL a été développé par Netscape Communication. De façon à améliorer la confidentialité des échanges de tout protocole basé sur TCP/IP. L'application la plus développée est bien sûr la sécurisation des sessions HTTP (*Hyper Text Transfer Protocol*, [HTTP]) Entre clients et serveurs WWW.

La première version diffusée est SSL v2.0 [SSLV2] en 1994.

Netscape a proposé en 1996 une amélioration du protocole, appelée SSL v3.0. La description a fait l'objet d'une proposition de standard Internet. Elle a toutefois servi de base au

développement du protocole TLS (*Transport Layer Security*), en cours de normalisation par l'IETF (*Internet Engineering Task Force*).

6.2.2. Le protocole HTTPS

Le protocole HTTPS est l'implémentation du protocole HTTP au-dessus de SSL. Du fait que le démarrage d'une session SSL se fait à l'initiative du client, l'utilisation de HTTPS pour le transfert de pages WWW doit être spécifié dans l'URL (*Uniform Resource Locator*). Ainsi, pour une URL de la forme `http://www.domaine.fr`, le client établira une connexion HTTP classique, alors que pour une URL de la forme `https://www.domaine.fr`, le client ouvrira une session SSL avec le serveur, et utilisera ensuite le protocole HTTP au-dessus de cette session.

Le protocole HTTPS est communément utilisé dans les applications de commerce électronique, au moins dans la phase de transfert des coordonnées bancaires (numéros de cartes bancaires, notamment). Ce n'est bien sûr pas la seule application, comme on le verra dans la suite.

6.2.3. PGP (Pretty Good Privacy)

1) Chiffrement de PGP

Le système de PGP est un système hybride que l'on peut classer dans les systèmes "à clef de session", c'est-à-dire un système qui utilise à la fois le principe du chiffrement à clef privée et le principe du chiffrement à clef publique.

Considérons les différentes étapes du transfert d'un message crypté avec PGP de l'expéditeur X vers le destinataire Y.

- ❖ X doit envoyer le message crypté à Y.
- ❖ Y crée une paire de clef via l'algorithme RSA. Il transmet sa clef publique à X.
- ❖ X saisit le texte en clair à envoyer. Ce texte est tout d'abord compressé ce qui offre un double avantage :
 - la taille des données à transférer est réduite,
 - les risques de décryptage sont minimisés (la plupart des techniques de cryptanalyse se base sur le texte en clair obtenu. Si le texte obtenu est un texte compressé il est, par exemple, plus difficile de calculer la probabilité de retrouver telle ou telle lettre).
- ❖ Puis, X crée aléatoirement une clef secrète IDEA. L'expéditeur chiffre le texte avec cette clef IDEA. Le texte ainsi chiffré pourra être déchiffré avec la même clef. Dans PGP, le message est alors crypté selon un système symétrique (à clef secrète).
- ❖ Le destinataire Y ne connaissant pas cette clef, elle va lui être envoyée avec le message crypté. Toutefois pour éviter qu'elle soit interceptée, la clef sera également

cryptée à l'aide de la clef publique de Y. La clef privée IDEA est cryptée avec la clef publique de Y selon un système asymétrique (à clef publique).

Finalement, le résultat obtenu contient :

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

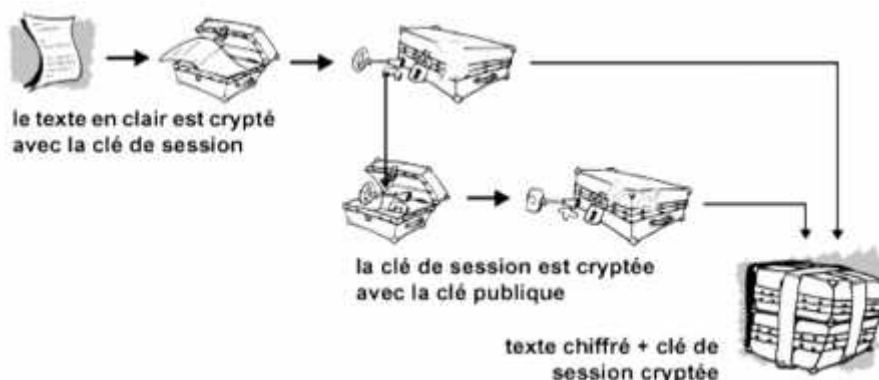


Figure 2.3 : Chiffrement de PGP [3]

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

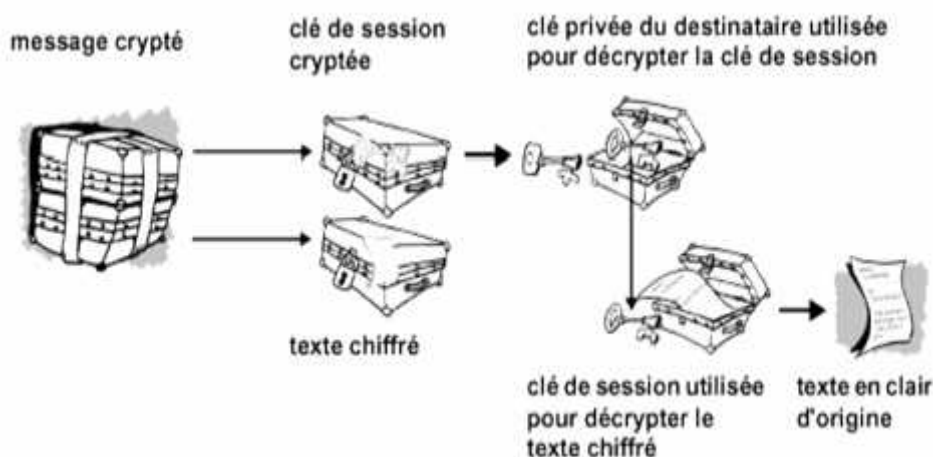


Figure 2.4 : Déchiffrement de PGP [3]

2) Avantage de PGP

PGP possède plusieurs **avantages** :

- La rapidité : le message est chiffré par un cryptage symétrique. La clef IDEA est chiffrée de façon asymétrique. Toutefois le volume de données que représente cette clef est négligeable par rapport au volume de données que représente le message. Par conséquent, le temps de chiffrement global est proche de celui d'un système symétrique.
- Une plus haute sécurité qu'un système à clef symétrique. En effet, si l'on peut considérer que le niveau de sécurité du système PGP est celui de son maillon le plus faible - le système à clef privée servant à coder le message - il faut toutefois nuancer cette conclusion. Dans un système

à clef privée standard, le canal d'échange de la clef est le point faible du système. Si l'on désire changer de clef afin de minimiser les risques, cela nécessite de définir un moyen d'échanger cette nouvelle clef, opération difficile à mettre en pratique. Dans PGP en revanche, la clef utilisée pour coder le message est nouvelle pour chaque message. Ce qui implique que pour effectuer une attaque il est nécessaire de casser au choix :

- autant de clefs privées que de messages.
- le système de clefs RSA, ce qui rend finalement PGP plus résistant qu'un système à clef privée classique.

3) La signature des données

En matière de signature des données, PGP utilise un scellement de données. Il applique une fonction de hachage au texte en clair à signer. Puis le condensé obtenu, de taille fixe, est signé avec la clef privée de l'expéditeur. Le sceau ainsi obtenu est joint au texte en clair.

A la réception du message, le destinataire (i) applique la fonction de hachage au texte en clair, (ii) utilise la clef publique de l'expéditeur pour retrouver la valeur du condensé joint au texte en clair et (iii) compare les deux condensés. Il s'agit d'un système de scellement des plus classiques, seule la fonction de hachage utilisée est propre à PGP.

4) Les certificats

Rappelons que l'un des points clefs des cryptosystèmes est la vigilance que l'utilisateur apporte à la vérification de l'appartenance de la clef au bon propriétaire.

Ces systèmes sont en effet sensibles aux attaques dites de 'l'homme au milieu' qui consiste pour l'attaquant à créer une clef qu'il fait passer pour la clef d'une autre personne, s'accordant ainsi la possibilité de lire tous les messages sensément destinés à la personne dont il a usurpée l'identité. Pour éviter cela, on crée des certificats numériques dont le but est d'apporter la preuve de la validité d'une clef et de son appartenance à un propriétaire donné.

Ces certificats peuvent être considérés comme les cartes d'identité des clefs et se composent de trois parties :

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

En règle générale, une autorité appelée « autorité de certification » est seule habilitée à produire des certificats et les signer à l'aide de sa clef privée. Dans PGP, le système retenu pour les certifications numériques ne se base pas sur une autorité de certification centralisée mais sur un système de confiance.

Il est possible à chaque personne de signer de sa clef privée un certificat. Contrairement à un système centralisé, un certificat PGP pourra contenir une multitude de signatures numériques.

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

- Le numéro de la version de PGP utilisée pour créer la clef associée à ce certificat.
- La clef publique sur laquelle porte ce certificat ainsi que l'algorithme employé pour la générer.
- Des informations sur le propriétaire de cette clef (nom, mail, nom d'utilisateur etc.)
- La signature numérique du propriétaire effectuée à l'aide de la clef privée qui correspond à la clef publique du certificat.
- La période de validité du certificat.
- Les éventuelles signatures effectuées par d'autres utilisateurs. [3]

5) Utilisation pratique

Dans cette partie, nous traiterons de l'utilisation pratique de PGP. A cette fin, nous n'utiliserons pas le logiciel PGP de la PGP Corporation (cf. www.pgp.com) mais une autre implémentation du standard Open PGP appelée GPG ou GnuPG (Gnu Privacy Guard) libre de droit et reconnue par Philip Zimmermann. De plus ce logiciel s'avère plus éducatif car il fonctionne entièrement en ligne de commande.

IDEA est un algorithme de génération de clef privée propriétaire qui est utilisé dans le logiciel PGP de la PGP Corporation, à ce titre, il n'est pas utilisé dans GnuPG. La clef qui sert à crypter le message dans GnuPG n'est donc pas une clef IDEA, mais reste une clef privée répondant à la spécification de la recommandation Open-PGP.

6.3. Le cryptage dans les clients de messageries actuels

6.3.1. Le chiffrement dans Microsoft Outlook 2010

Le chiffrement d'un message électronique dans Microsoft Outlook 2010 permet de protéger la confidentialité de ce message en le convertissant à partir d'un texte brut, lisible, en texte chiffré. Seul le destinataire disposant de la clé privée (clé privée : clé secrète conservée sur l'ordinateur de l'expéditeur et utilisée par celui-ci pour signer numériquement les messages qu'il envoie et déchiffrer ceux qu'il reçoit. Les clés privées doivent être protégées par mot de passe) Qui correspond à la clé publique (clé publique : clé qu'un expéditeur donne à un destinataire de sorte que ce dernier puisse vérifier la signature de l'expéditeur et confirmer que le message n'a pas été modifié. Les destinataires utilisent également la clé publique pour chiffrer les messages électroniques renvoyés à l'expéditeur) que vous avez utilisée pour chiffrer le message peut déchiffrer celui-ci et le lire. Tout destinataire ne possédant pas la clé privée ne voit que du texte indéchiffrable.

Remarques

- L'envoi et l'affichage de messages électroniques chiffrés requièrent que l'expéditeur et le destinataire partagent leur identification numérique (identification numérique : contient une clé privée qui reste sur l'ordinateur de l'expéditeur et un certificat (avec une clé publique). Le certificat est envoyé avec les messages signés numériquement. Les destinataires enregistrent le certificat et utilisent la clé publique pour chiffrer les messages destinés à l'expéditeur.) ou certificat de clé publique. Cela signifie que le destinataire et vous-même devez-vous envoyer un message signé numériquement, ce qui vous permet d'ajouter le certificat de l'autre personne à vos contacts. Vous ne pouvez pas chiffrer de messages électroniques sans identification numérique.
- Si vous envoyez un message chiffré à un destinataire dont la configuration de messagerie ne prend pas en charge le chiffrement, Outlook vous avertit et vous offre la possibilité d'envoyer le message sous format non chiffré.
- Ce processus chiffre également les pièces jointes envoyées avec des messages chiffrés. [23]

6.3.2. Mozilla Thunderbird et Enigmail

La fondation Mozilla offre des pré-versions de son logiciel de messagerie Mozilla Thunderbird. Bien que Thunderbird n'en soit qu'à la version 0.8, ses fonctionnalités et sa qualité dépassent de loin ce qu'on peut espérer retrouver dans un logiciel avec ce numéro de version. Cela s'explique en partie par le fait que Thunderbird soit basé sur le logiciel de messagerie de la Suite Mozilla, qui est développée depuis 1998, dérivée des produits de Netscape Corporation. Enigmail est une extension disponible pour Thunderbird qui permet le chiffrement asymétrique basé sur la norme OpenPGP. Enigmail utilise le logiciel libre GnuPG, (prononcer Gnou-Pé-Gé) pour arriver à ses fins. Vous avez peut-être déjà entendu parler d'un logiciel qui s'appelle PGP. GnuPG est l'équivalent en logiciel libre. La norme OpenPGP est basée sur des outils employés dans les domaines militaires, financiers et autres depuis le début des années 90. [24]

7.La contribution

Dans notre travail nous avons conçu un petit client de messagerie électronique permet d'envoyer et de recevoir des messages électronique sur internet, et gérer le chiffrement de ces messages, selon une amélioration nous avons ajouté sur le mécanisme de PGP, traitons un petit point faible :

Pour faire une attaque sur le mécanisme PGP nécessite :

- ❖ *Hacker une clef pour le déchiffrement* : le point vie de PGP est la clef publique, donc l'attaqueur fait un attaque pour cette clef pour obtenu la clef privé de destinataire.

Aussi les messages envoyés ne possèdent pas le même degré d'importance, pour cela en va ajouter une petite amélioration pour augmenter le niveau de sécurité, et pour assurer l'attaque encore plus difficile.

Comme solution nous allons proposer d'ajouter une autre clef secrète d'un algorithme symétrique (Cette clé est un secret entre l'émetteur et l'avenir) pour protéger la clef de session avants l'envoi de message, en utilisons pour cette clé l'algorithme de Rijndael comme un algorithme à clef privé, alors les étapes de cryptages devenait :

- crée aléatoirement une clef secrète (dans notre cas Rijndael).
- L'expéditeur chiffre le texte avec cette clef.
- Crypté Cette clef (clef de session) avec la clé secrète qui nous avons ajoutée (proposé).
- Puis chiffreée cette clef (clef de session cryptée) avec la clef publique de destinataire (RSA dans notre cas).
- Envoyer cette clef cryptée et le texte chiffré vers le destinataire.

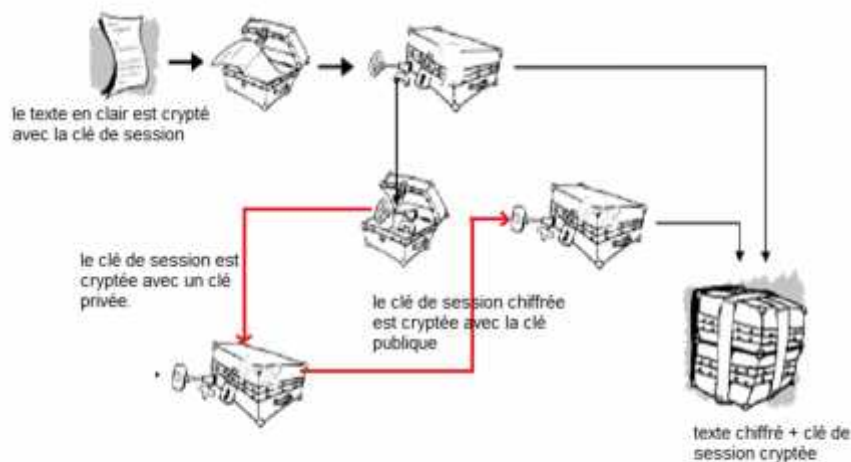


Figure 2.5 : Conception de nouveau chiffrement

A la réception du message, le destinataire utilise sa clef privée RSA pour retrouver la valeur de la clef de session crypté puis utilisé la clé secrète pour trouver la clé de session. Il utilise la clef obtenue pour déchiffrer le message reçu.

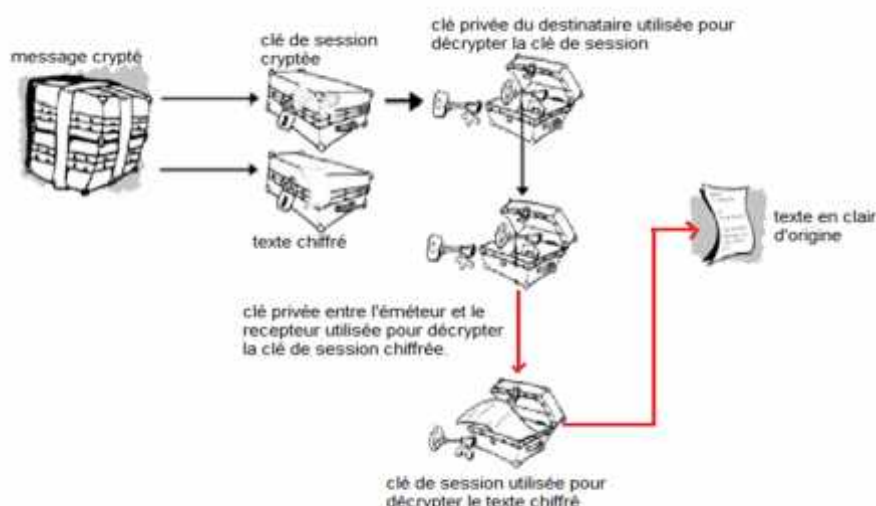


Figure 2.6 : Conception de nouveau déchiffrement

Alors maintenant pour faire une attaque il faut obtenir les deux clefs, la clef privé d'algorithme asymétrique RSA, et la clef secret d'algorithme symétrique Rijndael, donc l'attaque maintenant nécessite les deux clefs pour le déchiffrement du mail et obtenu le texte originale.

7.1. Evaluation des résultats

Avantages

- Augmenter le niveau de sécurité.
- Pour attaquer un message il faut attaquer les deux clés.
- Supposons un attaqueur essayés des clefs alors, pour chaque clef RSA il nécessite d'essayer $(1.1 \times 10^{77} \text{ clés})$ pour un taille 256-bit d'algorithme Rijndael.

Inconvénients

Le majeur inconvénient dans cette amélioration est le suivant :

- Difficulté de changement la clé secrète.

8. Conclusion

Dans ce chapitre, nous avons présenté la définition d'internet et quelques termes, puis nous avons présenté un état de l'art sur les techniques utilisées aujourd'hui pour sécuriser les données lors des échanges d'informations.

En particulier, nous avons parlé sur le PGP et quelque fonctionnalités de ce système, puis nous avons exposé notre modeste contribution que nous avons intéree dans notre logiciel, ce dernier sera modélisé avec UML dans le chapitre suivant.

CHAPITRE 3

Chapitre 3

Analyse et conception

1. Introduction

Dans ce chapitre, nous allons présenter l'analyse et la conception de notre application, cette application offre à des clients la possibilité d'envoi et de recevoir des mails, cryptés et non cryptés à travers l'Internet, ainsi la possibilité de déchiffrée les mails cryptés.

Nous avons suivi une approche orientée objet pour analyser et concevoir notre système, en appliquant la notation UML.

2. Analyse des besoins

2.1. Fonctionnalités attendues

Notre logiciel est un client de messagerie électronique, fonctionne sur le réseau Internet, gérer l'envoi et le recevoir des emails, d'une compte ou plusieurs, enregistrer dans une base de données, et gérer les fonctionnalités suivantes :

- ✦ Envoi les emails depuis quelque compte enregistrées dans le logiciel.
- ✦ Recevoir les emails reçus de chaque compte.
- ✦ Crypter les emails envoyé à partir de ce logiciel.
- ✦ Décrypter les emails reçus.

2.2. Public visé

Ce logiciel est destiné à toutes les personnes pouvant crypter mails, ainsi on particulier le militaire, car il utilise les mails crypté à travers le transfert des informations pour protéger les informations changées, et aussi on peut utiliser ce logiciel dans les communications sécurisés sur l'internet.

3. L'outil de conception utilisé

UML :

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue. [19]

UML unifie à la fois les notations et les concepts orientés objets. Il ne s'agit pas d'une simple notation graphique, car les concepts transmis par un diagramme ont une sémantique précise sont porteurs de sens au même titre que les mots d'un langage.

UML permet de représenter un système selon différentes vues complémentaires : les diagrammes.

Un diagramme UML 2.0 est une représentation graphique, qui s'intéresse à un aspect précis du modèle.

4. Présentation du système en UML

Les deux **acteurs** de notre système sont :

- L'émetteur
- Le récepteur

Les cas d'utilisation

- ❖ Ajouter un compte
- ❖ Générer pair de clef RSA
- ❖ Générer une clef privée (AES)
- ❖ Gestion des clefs publiques
- ❖ Gestion des clefs privées
- ❖ Crypté les messages envoyés
- ❖ Décrypté les messages reçues
- ❖ Consulter les messages reçus
- ❖ Consulter le carnet d'adresse
- ❖ Gestion des indésirables
- ❖ Envoyer message
- ❖ Modifier le mot de passe

4.1. Scénarios de chaque cas d'utilisation**a) Ajouter un compte**

Un seul scénario caractérise ce cas d'utilisation :

- Remplir le questionnaire par le client pour ajouter un compte.
- Supprimer un compte.

b) Générer pair de clef RSA

Deux scénarios caractérisent ce cas d'utilisation :

- Choisir la taille de clef RSA
- Générer les deux clefs (publique et privé)

c) Générer une clef privée (AES)

Deux scénarios caractérisent ce cas d'utilisation :

- Choisir la taille de clef AES
- Générer la clef privée (Rijndael)

d) Gestion des clefs publiques

Deux scénarios caractérisent ce cas d'utilisation :

- Ajouter une clef pour un contact
- Supprimer une clef associée à un contact

e) Gestion des clefs privées

Deux scénarios caractérisent ce cas d'utilisation :

- Ajouter une clef pour un compte
- Supprimer une clef associée à un compte

f) Crypter les messages envoyés

Deux scénarios caractérisent ce cas d'utilisation :

- Créer un message et entrer l'adresse de la récepteur
- Faire le cryptage de mail

g) Décrypter les messages reçus

Deux scénarios caractérisent ce cas d'utilisation :

- Sélectionner un mail pour décrypter
- Décrypter le mail reçue

h) Consulter les messages reçus

Trois scénarios caractérisent ce cas d'utilisation :

- Supprimer un message
- Imprimer

- Consulter les messages

i) Consulter le carnet d'adresses

Un seul scénario caractérise ce cas d'utilisation :

- Consulter le carnet d'adresse

j) Gestion des indésirables

Deux scénarios caractérisent ce cas d'utilisation :

- Ajouter un contact à la liste des indésirables
- Supprimer un contact parmi la liste des indésirables

k) Envoyer message

Quatre scénarios caractérisent ce cas d'utilisation :

- Ecrire le message envoyé
- Envoyer message
- Garder copie (sauvegardes)
- Pièce jointe

l) Modifier le mot de passe

Deux scénarios caractérisent ce cas d'utilisation :

- Remplir le formulaire
- Changer le mot de passe

4.2. Diagramme de cas d'utilisation

Représente les fonctions du système du point de vue de l'utilisateur, et se compose des Eléments suivants :

- **Acteur** : un rôle joué par une personne, un service... etc, qui interagit avec le système étudié.
- **Cas d'utilisation** : il est destiné pour représenter les fonctions du système et leurs interactions avec les différents acteurs (utilisateurs et autres systèmes qui interagissent avec le système étudié).
- **Relations** : entre cas d'utilisations et acteurs, mais simplement des relations d'utilisation (uses ou include).

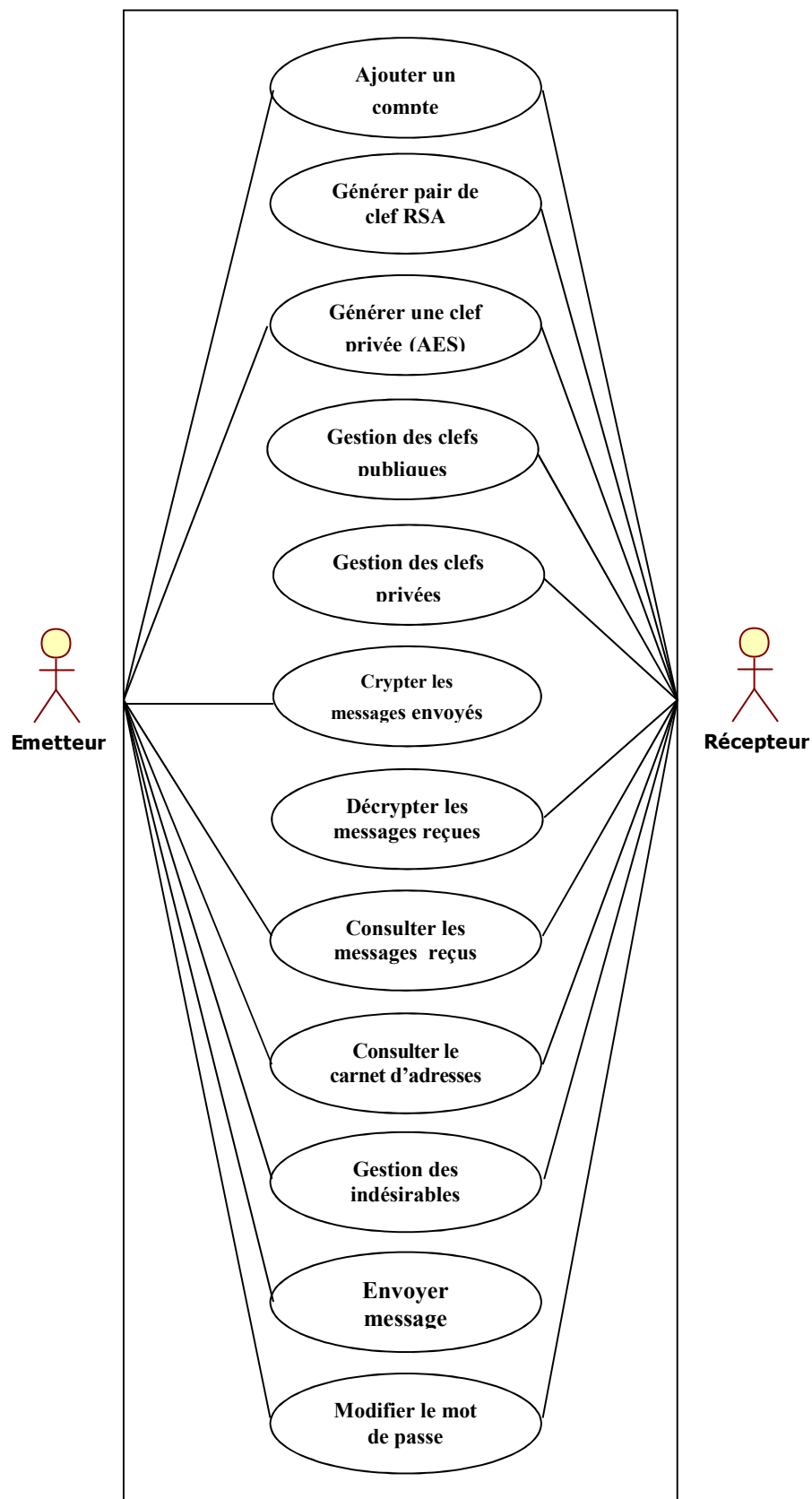


Figure3.1: Diagramme de cas d'utilisation

4.3. Scénarios de cas d'utilisation «Crypté les messages envoyés»

4.3.1. Diagramme de séquence

Les diagrammes de séquences permettent de représenter les interactions entre objets selon un point de vue temporel. L'accent est mis sur la chronologie des envois de messages.

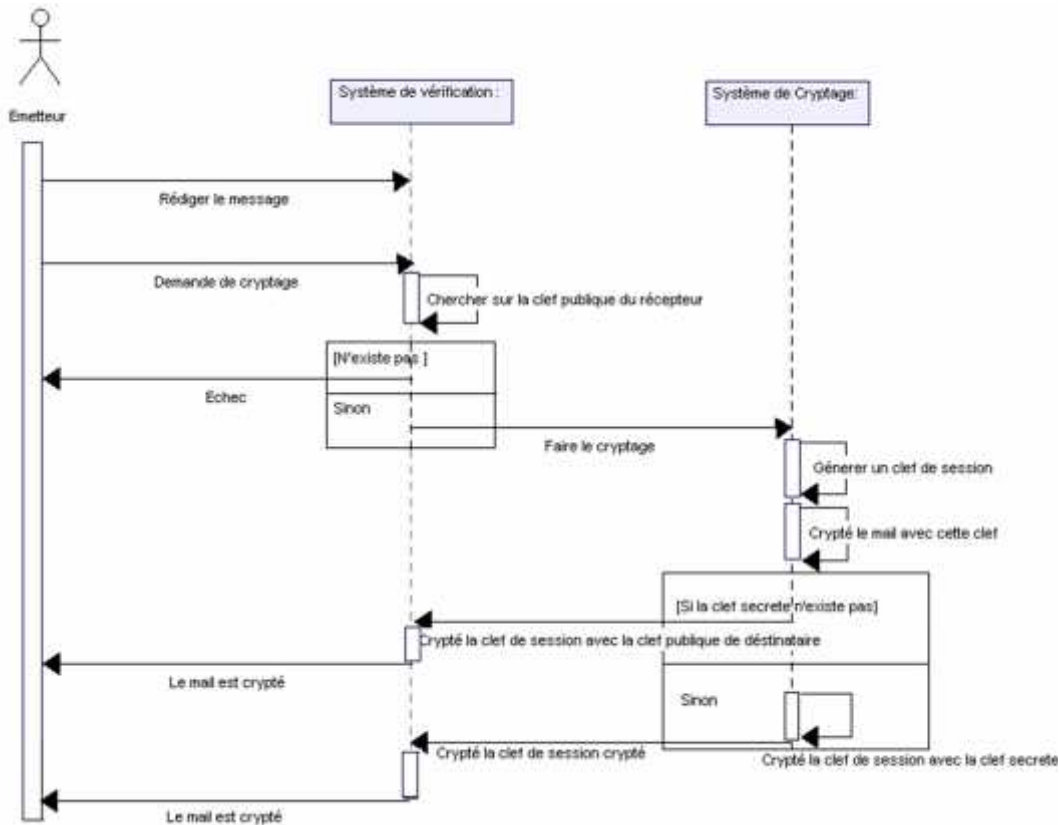


Figure 3.2: diagramme de séquence associé au scénario «Crypté les messages envoyés»

4.3.2. Diagramme de classe

Le diagramme de classes exprime la structure statique du système en termes de classes et de relations entre ces classes.

L'intérêt du diagramme de classe est de modéliser les entités du système d'information.

Le diagramme de classe permet de représenter l'ensemble des informations finalisées qui sont gérées par le domaine. Ces informations sont structurées, c'est-à-dire qu'elles ont regroupées dans des classes.

Le diagramme met en évidence d'éventuelles relations entre ces classes.

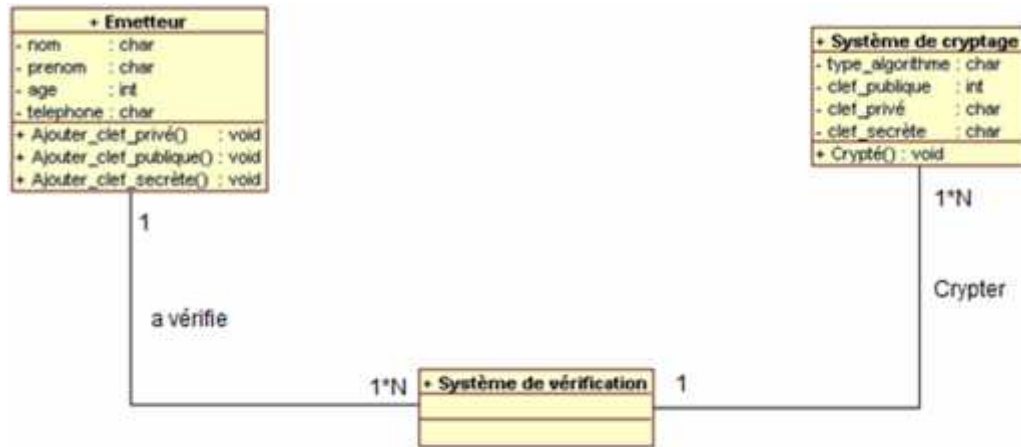


Figure 3.3 : Diagramme de classe associé à « crypté les messages envoyés »

4.3.3. Diagramme d'état transition

Ils ont pour rôle de représenter les traitements (opérations) qui vont gérer le domaine étudié. Ils définissent l'enchaînement des états de classe et font donc apparaître l'ordonnancement des travaux.

Le diagramme d'états-transition est associé à une classe pour laquelle on gère différents états : il permet de représenter tous les états possibles ainsi que les événements qui provoquent les changements d'état.

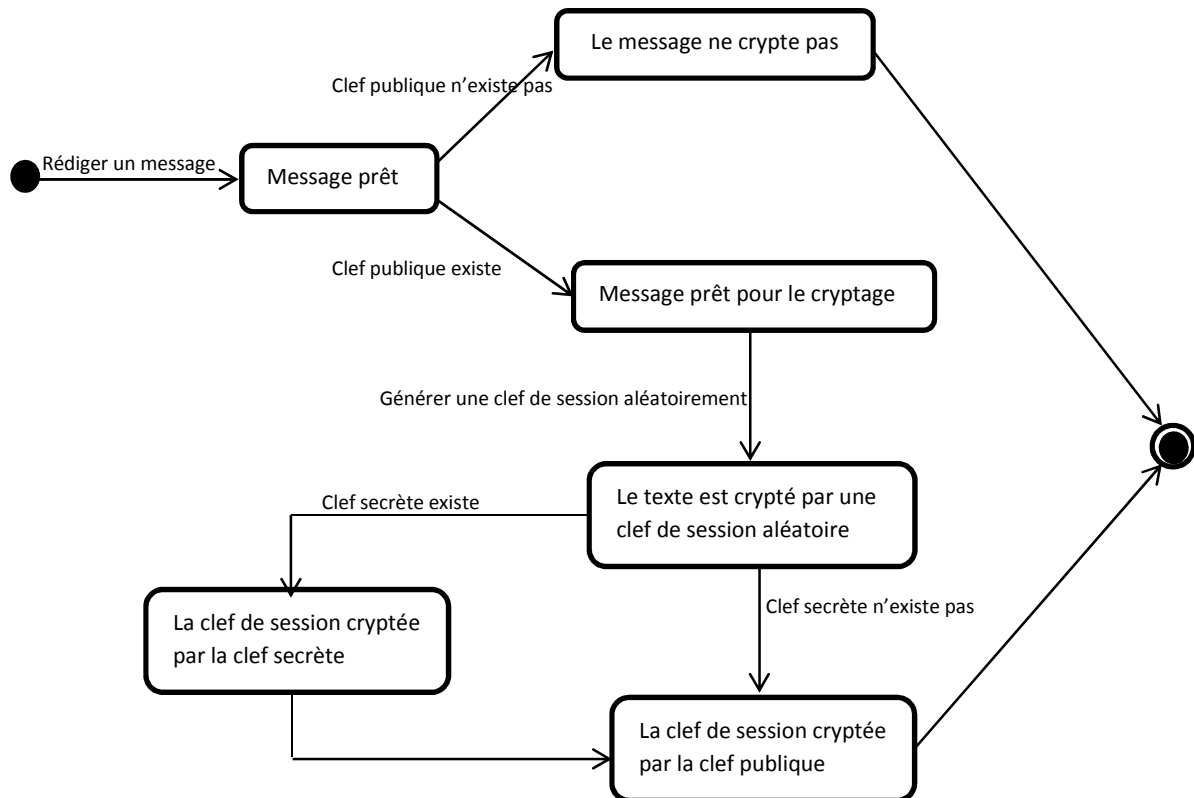


Figure 3.4 : Diagramme d'état transition associé à « crypté les messages envoyés »

4.3.4. Diagramme d'activité

Le diagramme d'activité est attaché à une catégorie de classe et décrit le déroulement des activités de cette catégorie. Le déroulement s'appelle "flot de contrôle". Il indique la part prise par chaque objet dans l'exécution d'un travail. Il sera enrichi par les conditions de séquencement.

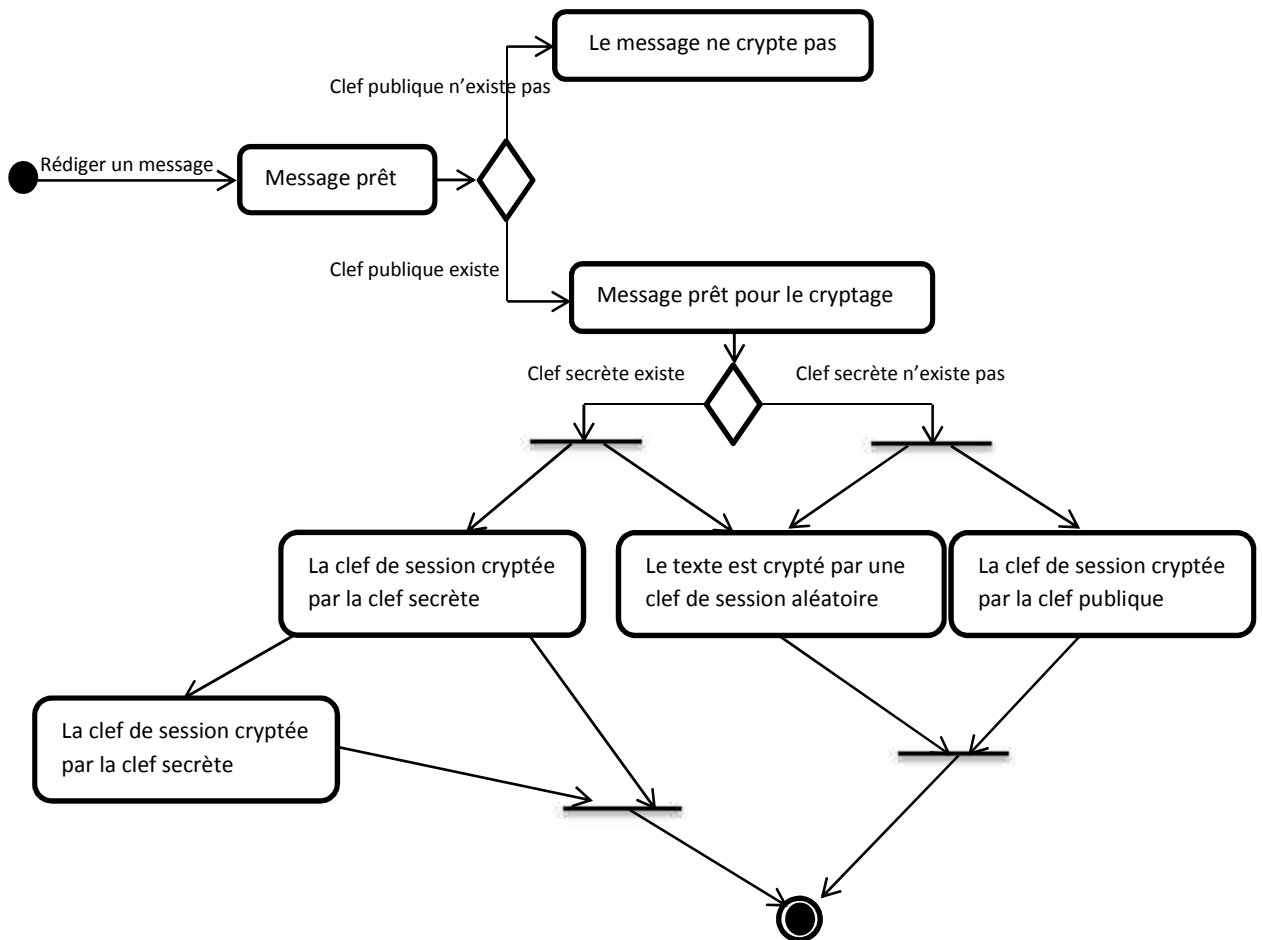


Figure 3.5 : Diagramme d'activité associé à « crypté les messages envoyés »

4.4. Scénarios de cas d'utilisation «Décrypté les messages reçues»

4.4.1. Diagramme de séquence

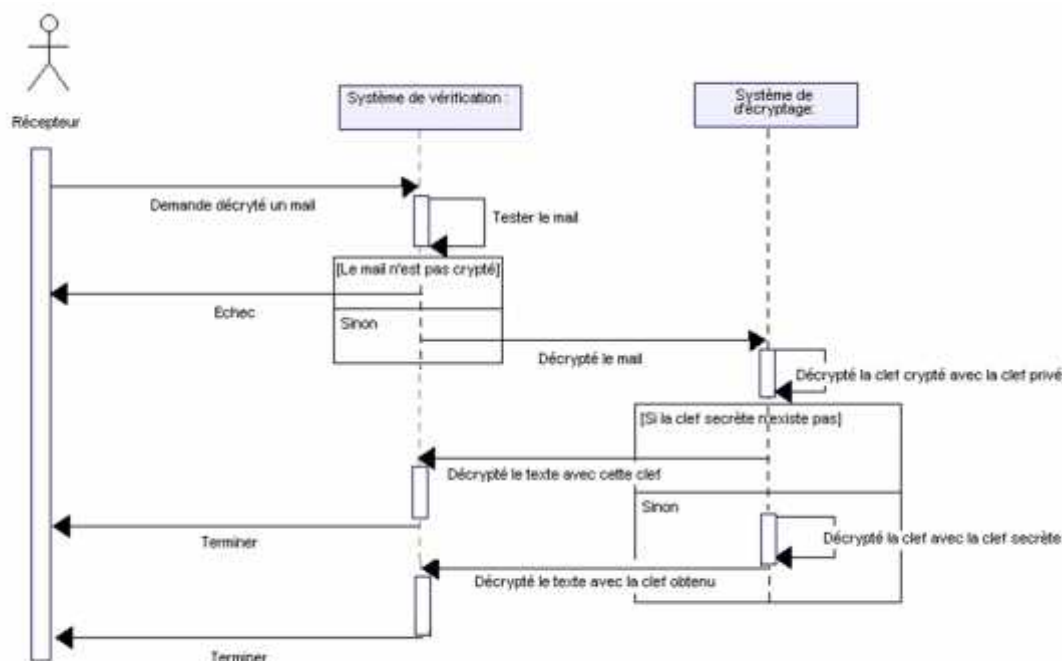


Figure 3.6: diagramme de séquence associé au scénario «Décrypté les messages reçues»

4.4.2. Description des classes associées

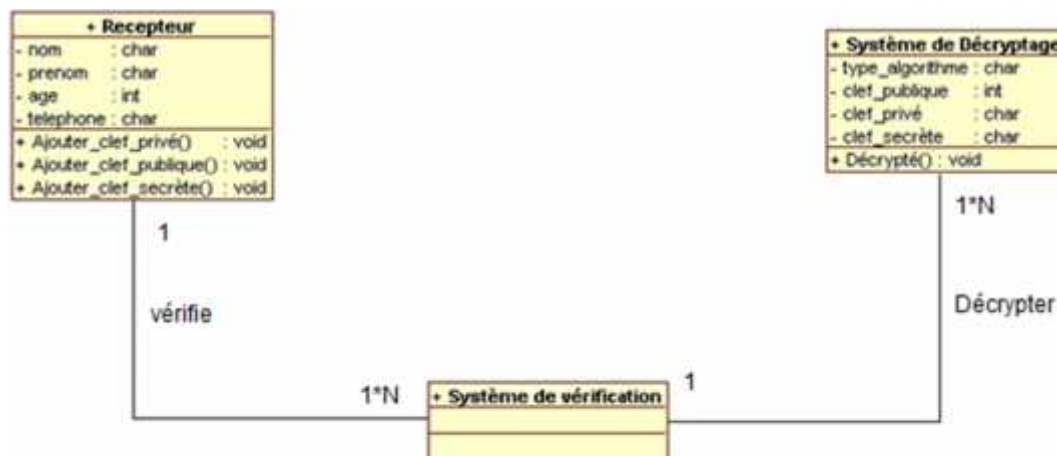


Figure 3.7 : Diagramme de classe associé à «Décrypté les messages reçues »

4.4.3. Diagramme d'état transition

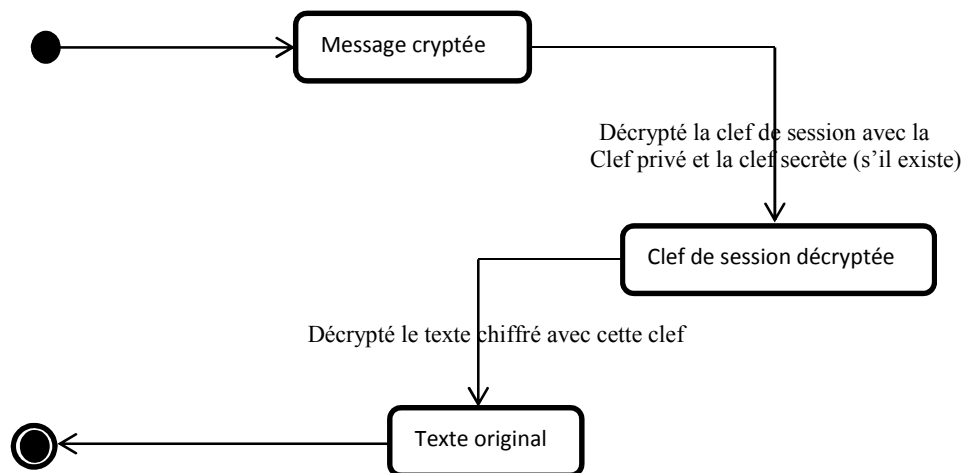


Figure 3.8 : Diagramme d'état transition associe à «Décrypté les messages reçues »

4.4.4. Diagramme d'activité

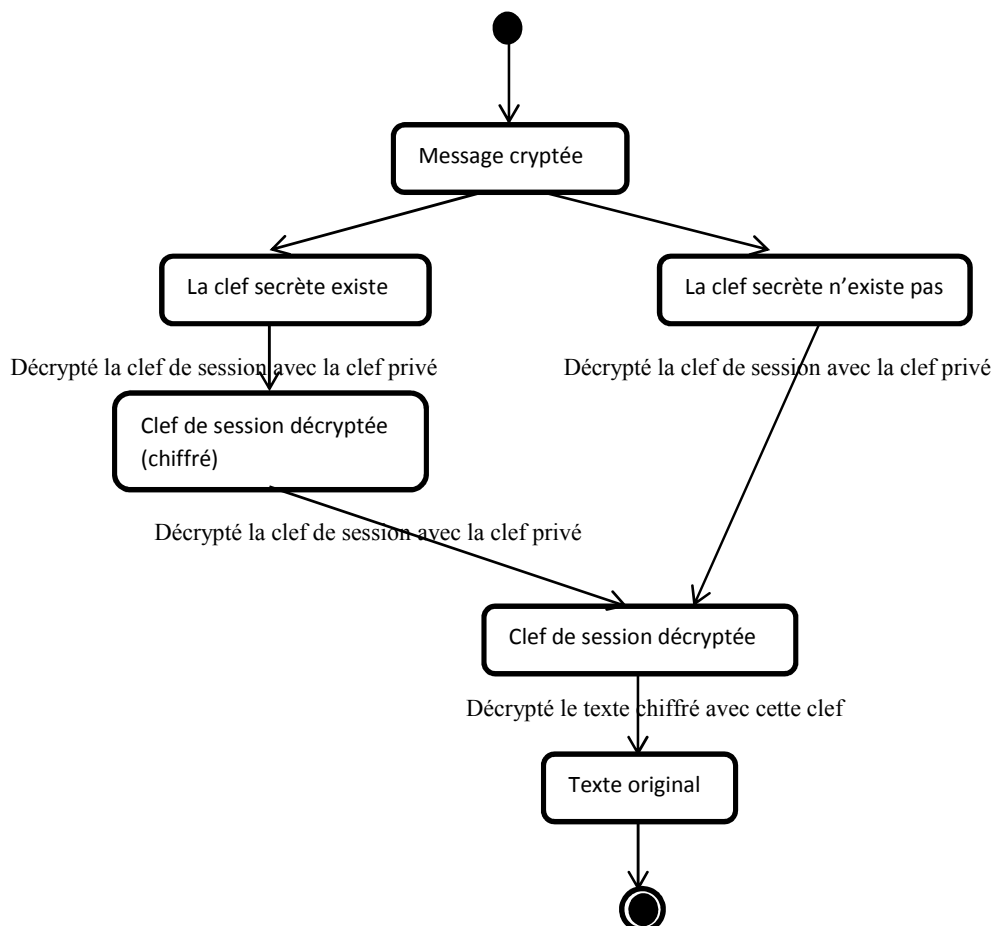


Figure 3.9 : Diagramme d'activité associe à «Décrypté les messages reçues »

4.5. Scénarios de cas d'utilisation «Envoyer message»

4.5.1. Diagramme de séquence

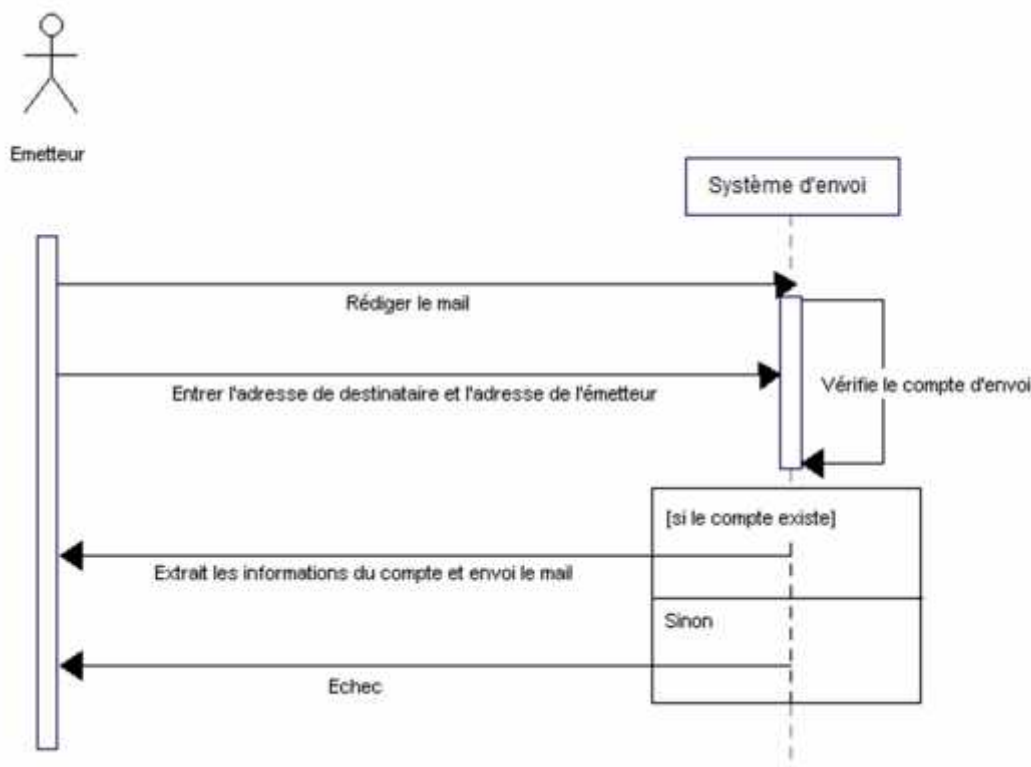


Figure 3.10 : Diagramme de séquence associé à «Envoyer message»

4.5.2. Description des classes associées

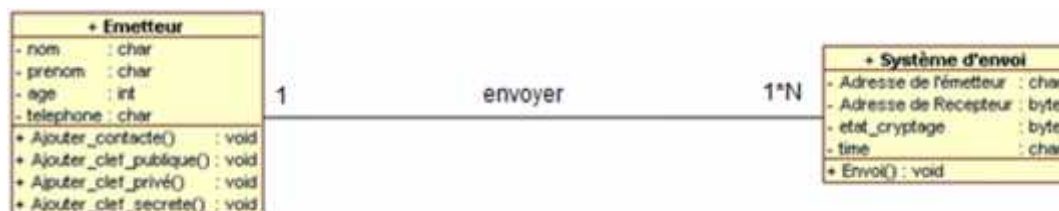


Figure 3.11 : Diagramme de classe associé à «Envoyer message»

4.5.3. Diagramme d'état transition

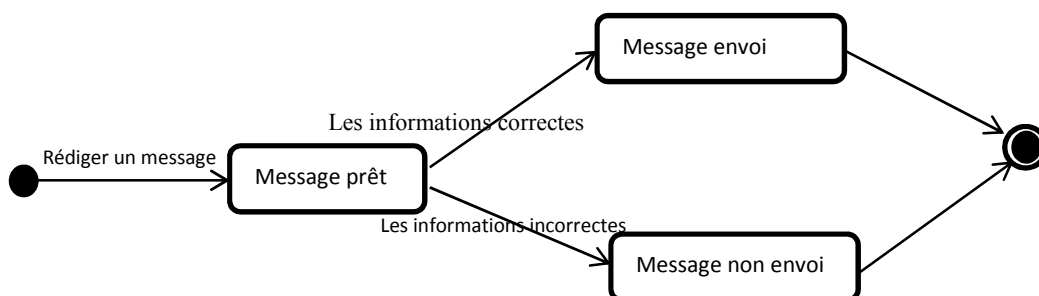


Figure 3.12 : Diagramme d'état transition associé à «Envoyer message»

4.5.4. Diagramme d'activité

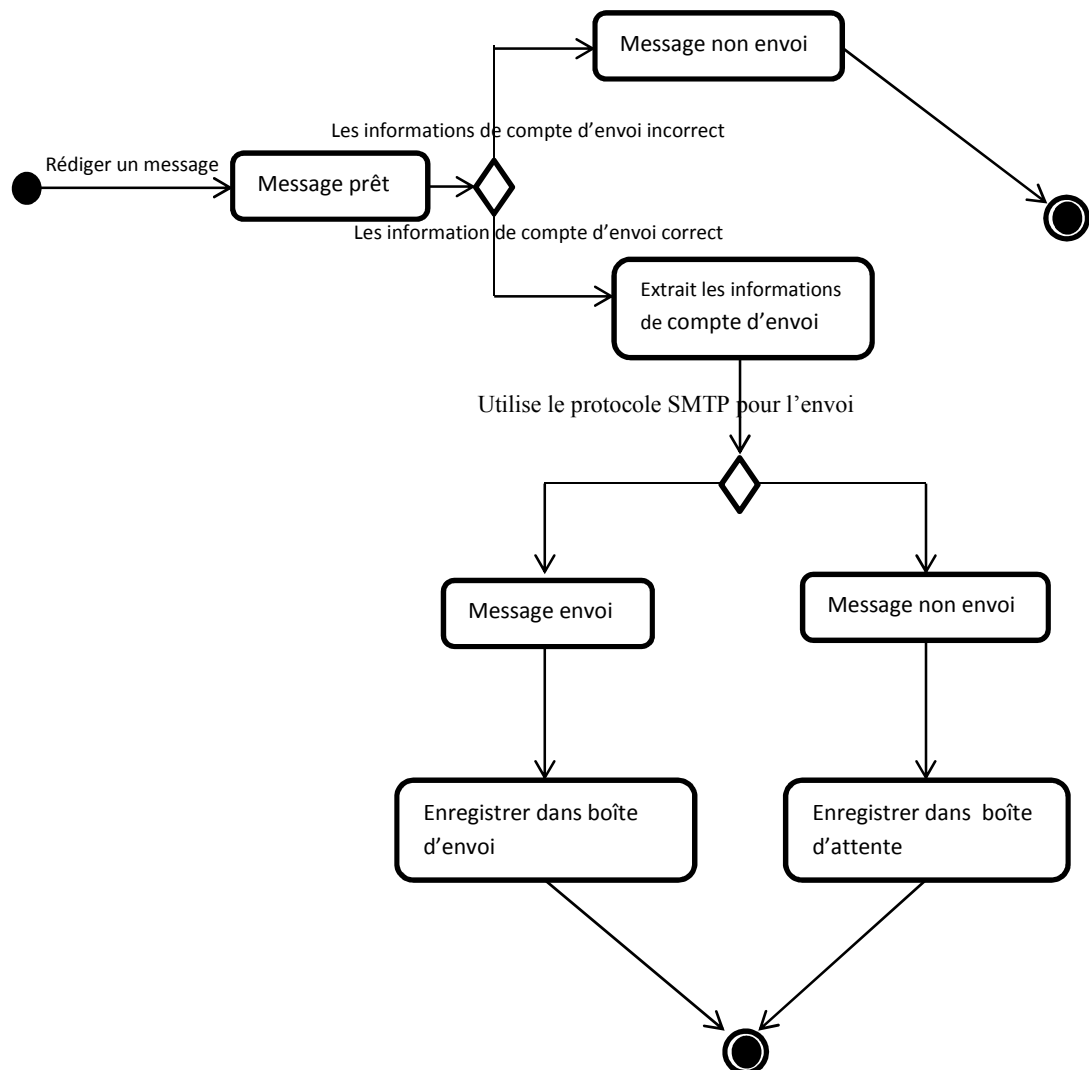


Figure 3.13 : Diagramme d'activité associé à «Envoyer message»

4.6. Scénarios de cas d'utilisation «Consulter les messages reçus»

4.6.1. Diagramme de séquence

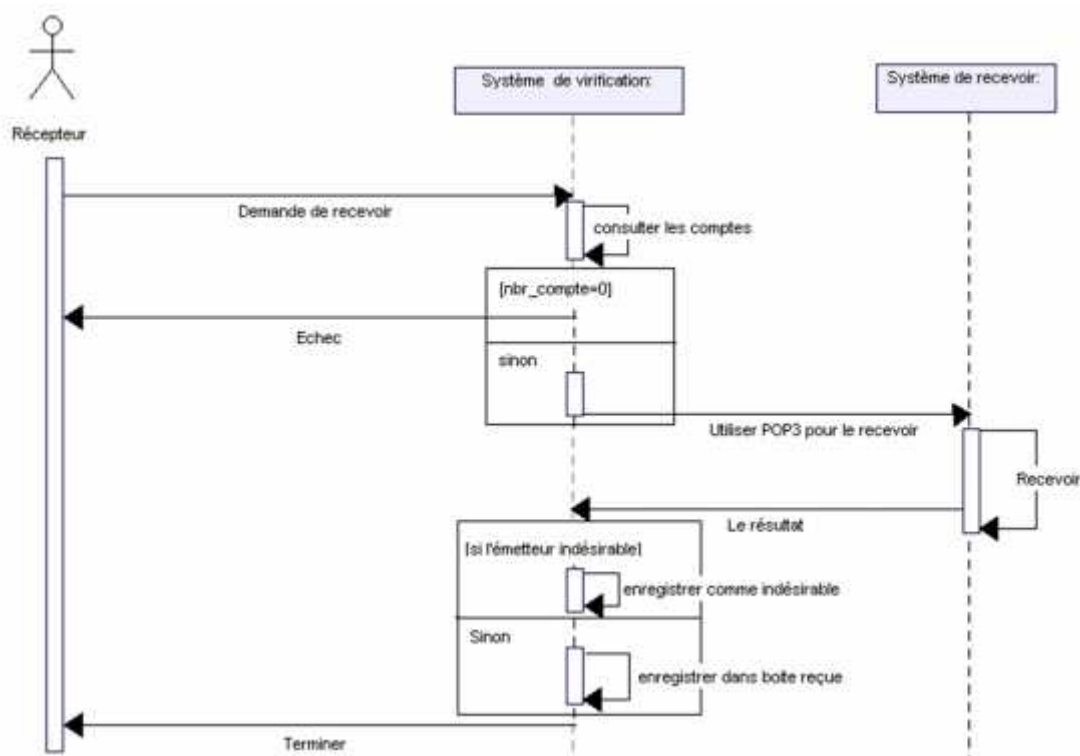


Figure 3.14 : Diagramme de classe associé à «Consulter les messages reçus »

4.6.2. Description des classes associées

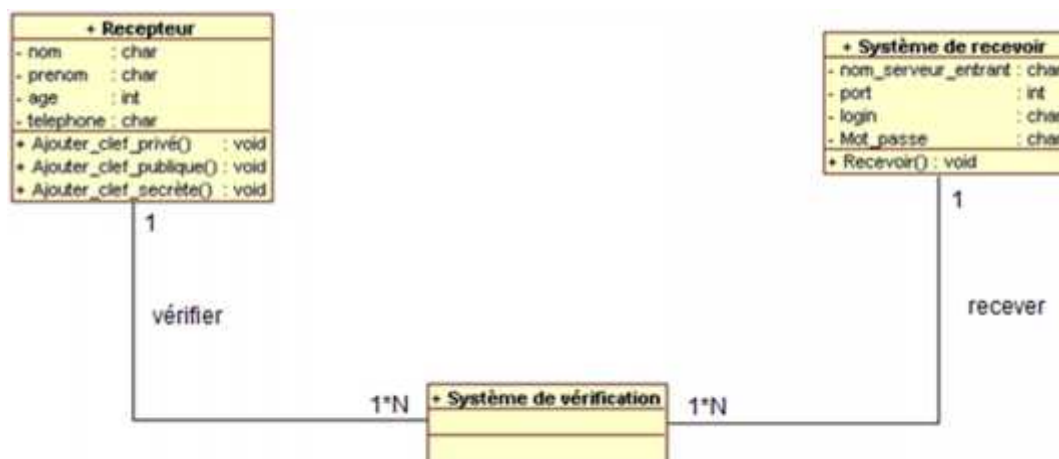


Figure 3.15 : Diagramme de classe associé à «Consulter les messages reçus »

4.6.3. Diagramme d'état transition

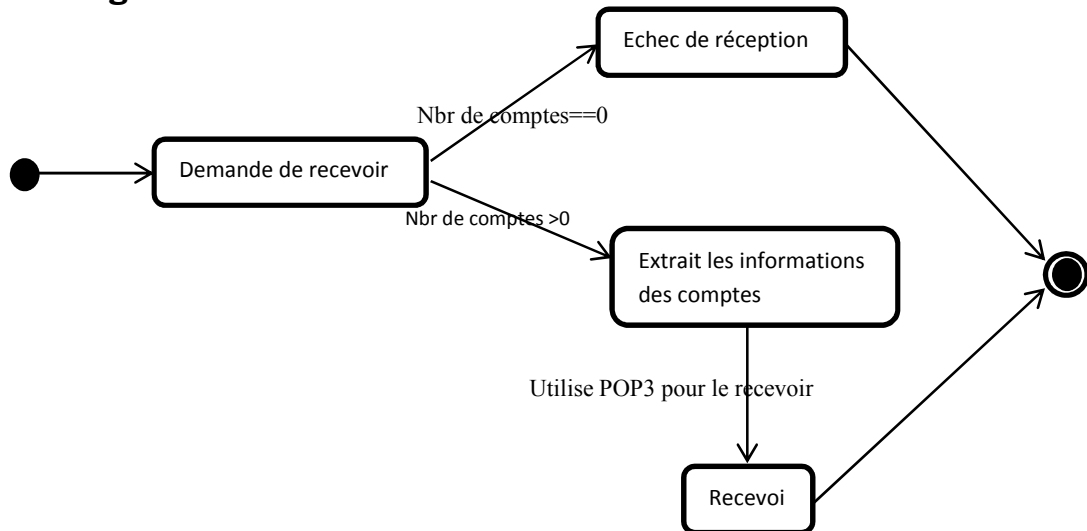


Figure 3.16 : Diagramme d'état transition associé à « Consulter les messages reçus »

4.6.4. Diagramme d'activité

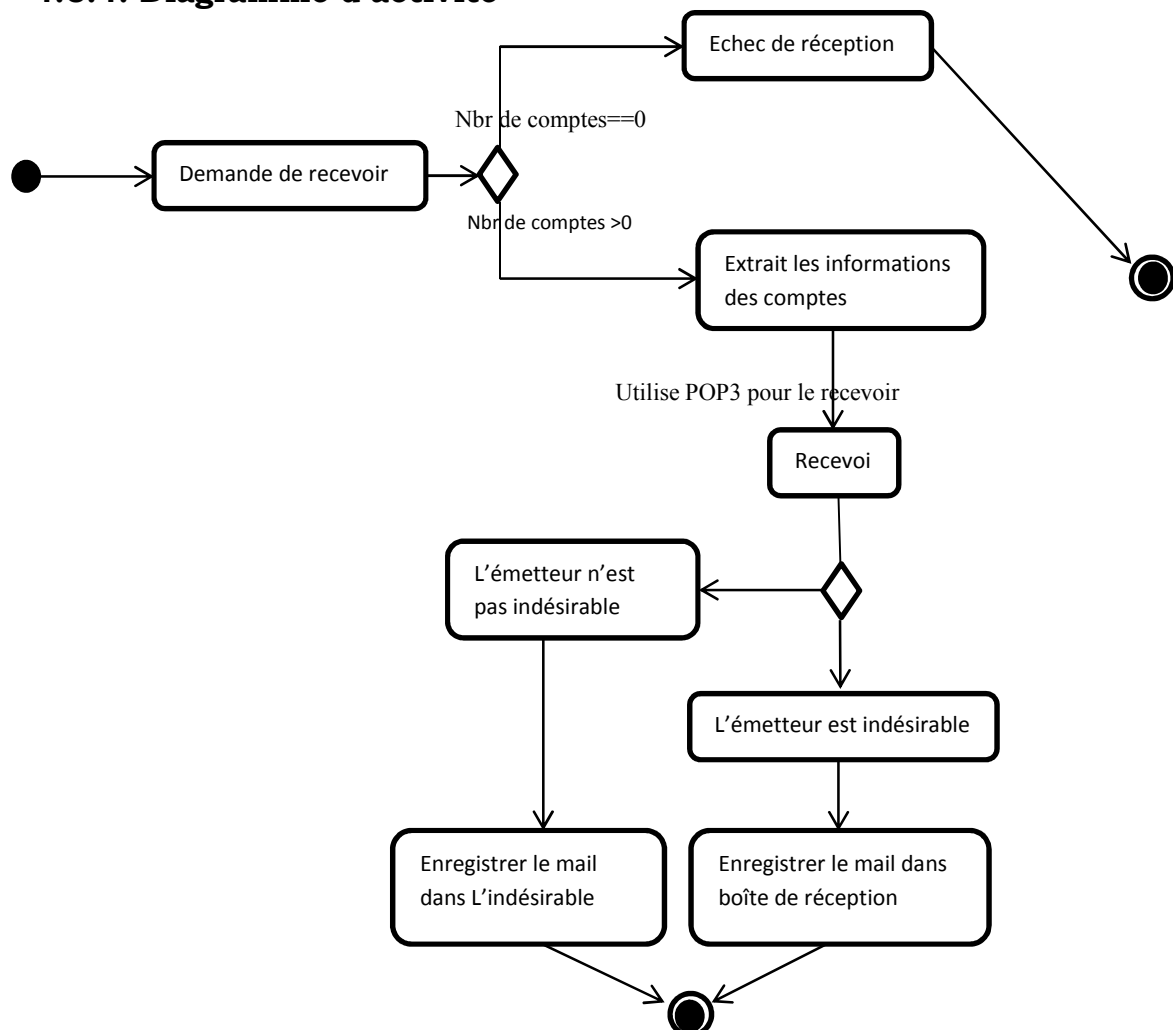


Figure 3.17 : Diagramme d'activité associé à « Envoyer message »

5. Les algorithmes utilisés

5.1. Rijndael

Nous avons utilisé l'algorithme de Rijndael (Contraction des noms des deux inventeurs belges : Dr. Joan Daemen, Dr. Vincent Rijmen de l'Université Catholique du Louvain, Belgique) comme un algorithme de cryptographie symétrique, pour crypté le texte de mail, et aussi pour crypté la clef de session (Key) et sa matrice initiale (IV) par une clef secrète.

5.2. RSA (*Ron Rivest, Adi Shamir et Leonard Adleman*)

Nous avons utilisés l'algorithme RSA comme un algorithme de chiffrement asymétrique pour crypter la clef de session.

6. Les protocoles utilisés

6.1. POP3

Nous avons utilisé le protocole POP3 pour la réception des emails, ce protocole intégrer dans l'environnement de Visual Studio avec le composant *Openpop.dll*, pour le but de facilité la réception des mails.

Ce protocole utilise le port 995, et le serveur pop.gmail.com pour le serveur GMAIL et aussi utilise le port 995, et le serveur pop.mail.yahoo.fr pour le serveur YAHOO.

6.2. SMTP

Le protocole SMTP (Simple Mail Transfer Protocol) est le plus couramment utilisé pour la gestion du courrier entre serveurs sur Internet, reliés en permanence. Nous utilisons ce protocole pour l'envoi des mails.

Ce protocole utilise le port 587 et le serveur smtp.gmail.com pour le serveur GMAIL, et aussi le port 25 avec le serveur smtp.mail.yahoo.fr pour le serveur YAHOO.

7. Les bases de données utilisées

Dans notre logiciel nous utilisons SqlServer pour créer les bases de données qui utilisent. Les bases de données utilisées sont :

- 1- **Clefs.sdf** : Cette base de données contient deux tables :



Figure 3.18 : tables de la base de données clefs.sdf

2- Contacte_compte.sdf : contient trois tables :

Table	Fields
compte	nom_compte, adresse_electronique, nom_apparait_adresse, serv_entr, port_serv_ent, serv_sort, port_ser_sort, nom_utilisateur, mot_passe
contacte	nom, prenom, date_naissance, age, adresse, adresse_electronique, telephone
contacte_indesirable	nom_officher, adresse_messagerie

Figure 3.19 : tables de la base de données contacte_compte.sdf

3- Emails.sdf : Contient six tables :

Table	Fields
envoyé	De, A, Objet, Date, Contenu, Crypté
Boite_reception	lire, De, A, Objet, Date, Contenu, Chiffree
attente	De, A, Objet, Date, Contenu, Crypté
Corbeille	lire, De, A, Objet, Date, Contenu, Crypté
Brouillons	De, A, Objet, Date, Contenu, Crypté
indesirable	lire, De, A, Objet, Date, Contenu, Crypté

Figure 3.20 : tables de la base de données emails.sdf

4- Password.sdf : contient une seule table

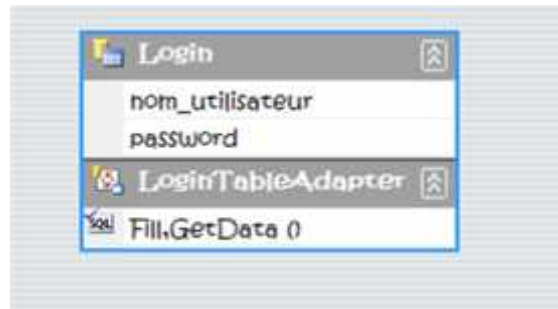


Figure 3.21 : La table Login.

8. Conclusion

Dans ce chapitre nous avons élaboré les diagrammes de cas utilisation, séquence, classes, état transition et le diagramme d'activité pour faire la modélisation des majeures fonctionnalités de notre logiciel avec le langage de modélisation UML, ce logiciel sera implémenté avec le langage de C# (C sharp) comme sera décrit dans le chapitre suivant.

CHAPITRE 4

Chapitre 4

Réalisation & Implimentation

1. Introduction

Dans ce chapitre nous allons faire la réalisation de notre application, cette application permettra à ses utilisateurs de mieux gérer leurs e-mails, Nous consacrons la première partie à la présentation de l'environnement de l'application. Par la suite, nous exposerons quelques interfaces homme machine qui concordent avec les fonctionnalités du système.

2. Environnement de travail

2.1. Environnement matériel

Pour le développement nous avons utilisé une machine à puissance moyenne : un PC « lap top » core (TM) 2 Duo 2.10GHz et 2Go de RAM.

2.2. Environnement logiciel

Le long de la phase de développement, nous avons utilisé l'environnement logiciel suivant :

Système d'exploitation : Windows 7 Professionnel.

Outils de développement : Pour réaliser notre application nous avons adopté le langage C SHARP (c#) comme un langage de programmation dans l'environnement de Visual Studio 2010.

Les protocoles utilisés : on utilise dans notre travail le protocole POP3 pour la réception des mails et SMTP pour l'envoi.

Les DLL ajoutées : nous avons intégré deux composants au Visual studio :

Openpop.dll pour permet l'utilisation du POP3 et SMTP, et aussi en utilise **irisskin.dll** pour améliorer l'affichage.

Conception et modélisation en UML : Open ModelSphere 3.0 : est un outil de conception.

3. L'accès à l'application

La première fenêtre qui apparaît, en lançant notre logiciel est la suivante. Elle contient deux boutons, le premier pour entrer à l'application et l'autre pour l'annuler, pour entrer à l'application il faut saisir le nom d'utilisateur et leur mot de passe :



Figure 4.1 : La fenêtre d'authentification

Ces informations peuvent être changées à partir du programme comme montrer dans la fenêtre suivante :



Figure 4.2 : La fenêtre de modification de mot de passe et le nom d'utilisateur

La Forme principale contient un « *Listview* » pour afficher les emails, soit dans la boîte de réception ou bien dans les éléments en attentes, les éléments envoyées, Brouillons, les éléments supprimés ou dans les courriel indésirables, et contient aussi une barre de tâche pour faciliter l'utilisation de ce logiciel, la forme principale apparaît dans la figure suivante :

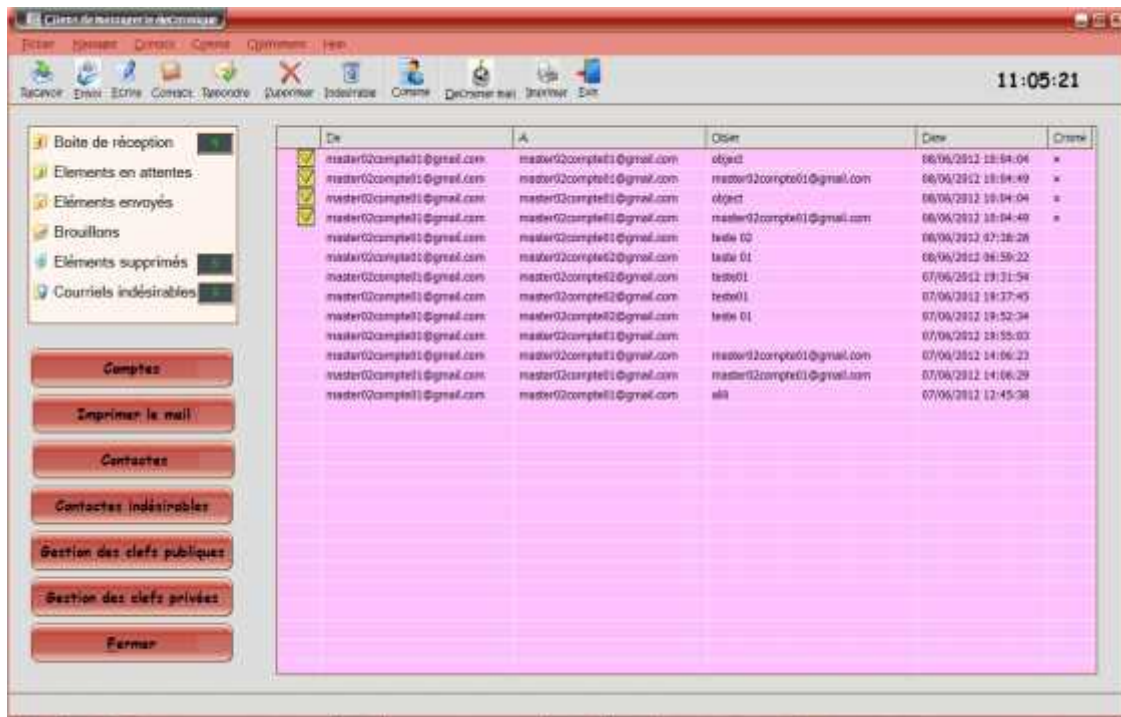


Figure 4.3 : La forme principale

Pour ce logiciel fonctionne il faut configurer cette application par ou moins un compte pour l'envoi des mails et la réception, pour configurer un compte il nécessite: un nom d'utilisateur et un mot de passe de cette e-mail, et aussi le nom de serveur sortent et sa port (SMTP dans notre cas) et le serveur entrant et sa port (POP3). La liste des comptes illustrés dans le logiciel à l'aide d'une liste comme la figure suivante :

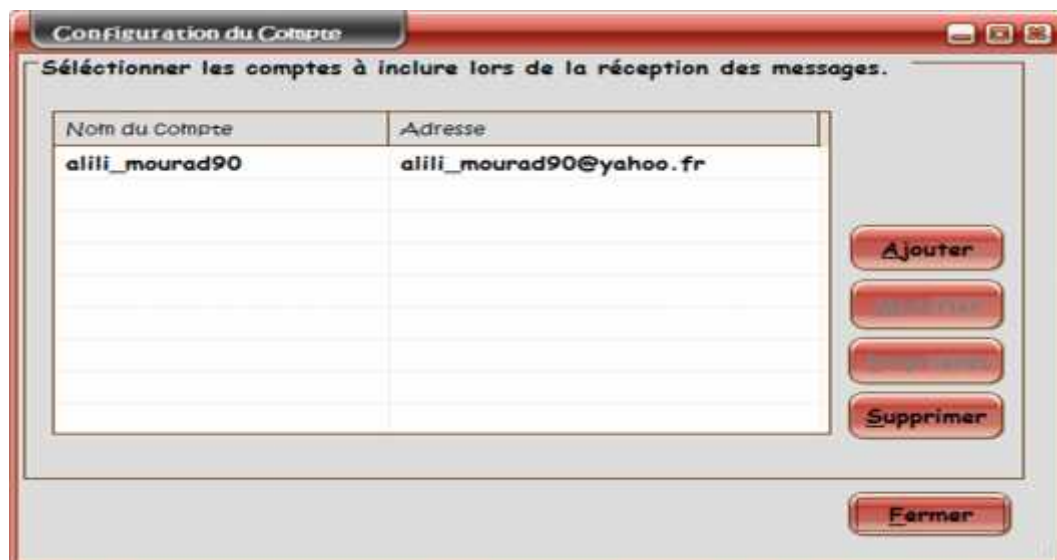


Figure 4.4: la liste des comptes

Pour configurer un compte appuyer sur le Botton *Ajouter*, les fenêtres apparait sont les suivantes :



Figure 4.5 : la liste des comptes

Puis la fenêtre suivante :

A screenshot of a software window titled "Configuration du Compte". The window has a red title bar with standard window controls. The main content area has a light gray background. The title "Configuration : mon compte" is centered at the top. Below it, there are four input fields, each preceded by a label: "Votre nom (Apparaîtra dans l'entête du mail) :", "Votre adresse de messagerie :", "Votre nom d'utilisateur : (Login) :", and "Votre mot de passe :". At the bottom right, there are two red buttons with white text: "Suivant" and "Annuler".

Figure 4.6 : la liste des comptes

Et terminer par la fenêtre suivante :

The screenshot shows a window titled 'Configuration du Compte' with a sub-header 'Configuration : serveur'. It contains the following fields:

- Type de serveur entrant : POP3 (dropdown menu)
- Nom du serveur entrant (ex pop.xxx.fr) : [empty text box]
- Port : 995 (text box)
- Nom du serveur sort (ex smtp.xxx.fr) : [empty text box]
- Port : 587 (text box)

At the bottom right, there are two buttons: 'Suivant' and 'Annuler'.

Figure 4.7 : la liste des comptes

Ces comptes incluent lors de la réception des messages, mais il y a des personnes considéré comme indésirables, son message dirigé vers la boîte des indésirable, ses personnes illustrer dans le logiciel à l'aide d'une liste, et peuvent être ajouté ou supprimé des contacts à partir cette liste, comme la figure suivante :

The screenshot shows a window titled 'Listes des contacts qui sont enregistrés en indésirables'. It contains a table with the following data:

Nom de ContaCts	Adresse du ContaCt
Ahmed	ahmed88@gmail.com
Mouhamed	mouhamed@yahoo.fr

At the bottom right, there are three buttons: 'Ajouter', 'Supprimer', and 'Exit'.

Figure 4.8 : la liste des contacts indésirables

Mais les autres contacts situés dans une autre liste comme la figure suivante :

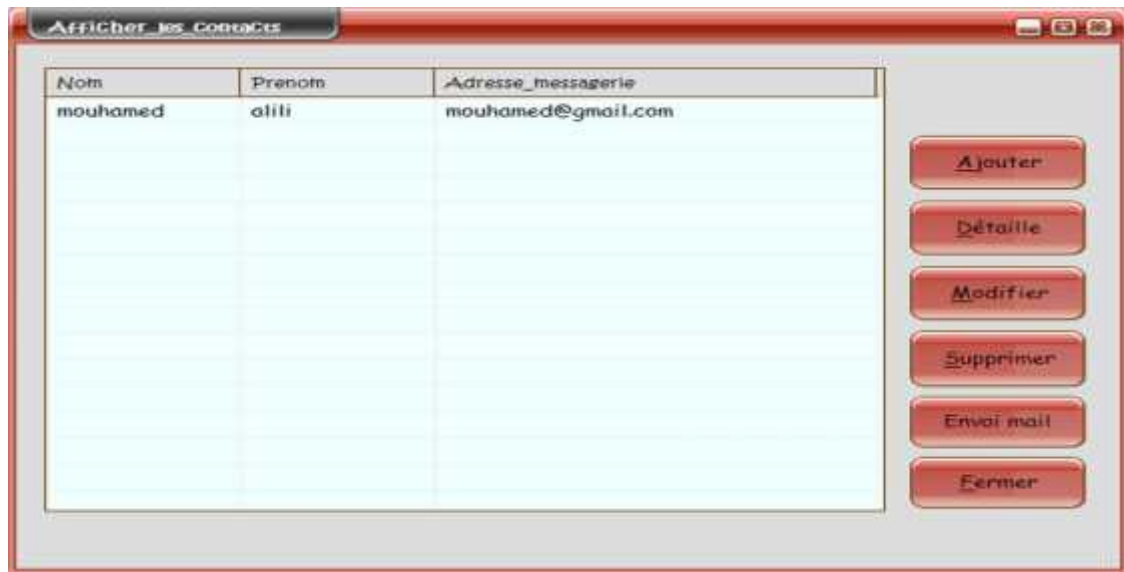


Figure 4.9 : la liste des contacts

On parle maintenant sur le plus important, alors pour générer un pair de clef RSA la fenêtre suivante donné un choix pour la taille de la clef, et si appuyons sur le boutons *Generate keys* le système donne deux choix, le premier la place de sauvegarde de la clef privé (pour le décryptage) et le deuxième la place de sauvegarde de la clef publique (pour le cryptage) :



Figure 4.10: Générât pair de clef RSA

Et pour générer une clef privée (secrète) d'algorithme AES, la fenêtre suivante donne un choix pour la taille de la clef (128, 192 ou 256 bit), comme la figure suivante:



Figure 4.11 : Génération d'une clé AES

Et pour associer une clé publique à un contact, la fenêtre suivante permet de gérer la gestion des clés publiques (l'ajout et la suppression) :

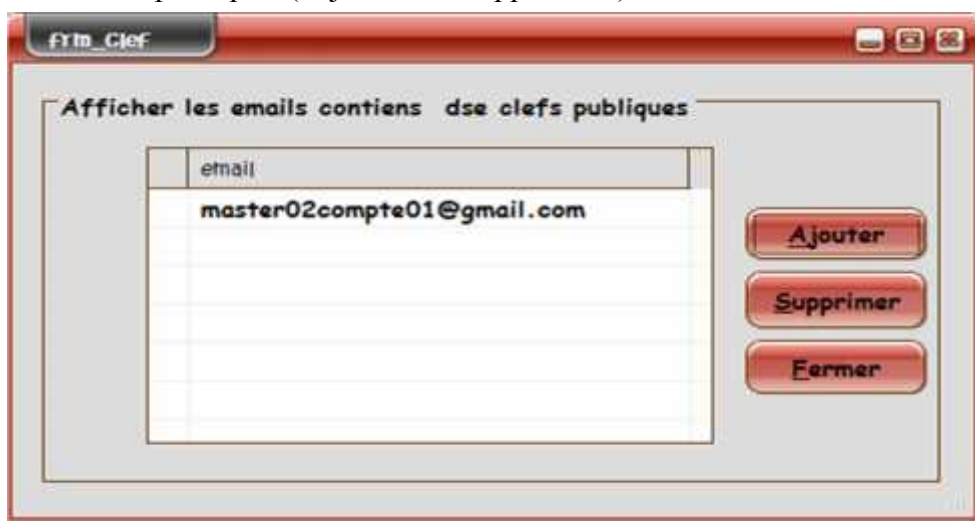


Figure 4.12 : Gestion des clés publiques

Et la fenêtre suivante permet de gérer les clés privées :

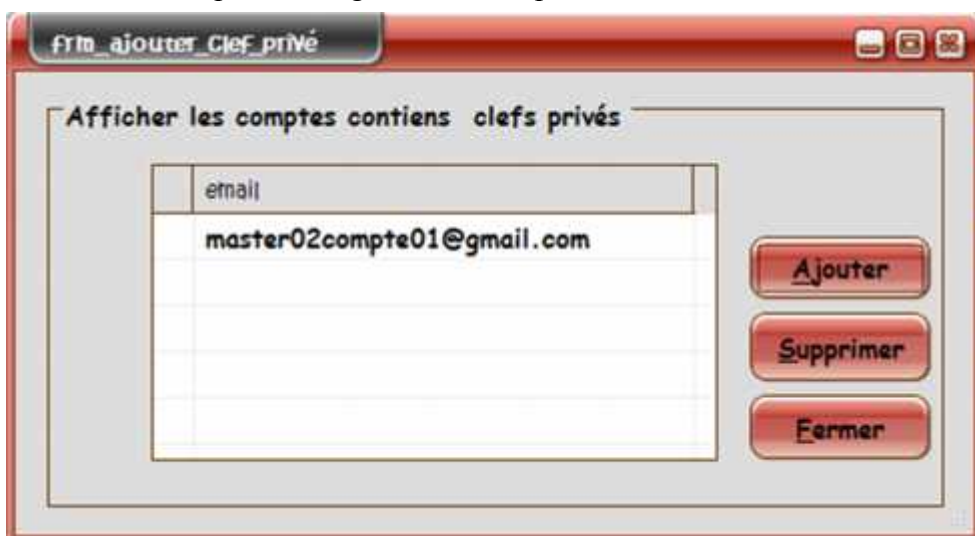



Figure 4.13 : Gestion des clés privées

Maintenant on parle comment crée et envoi un mail soi crypté ou non ? Alors pour rédiger un mail, appuyer sur l'icône  dans la barre des taches ou bien choisir *Ecrire nouveau message* dans le menu message :

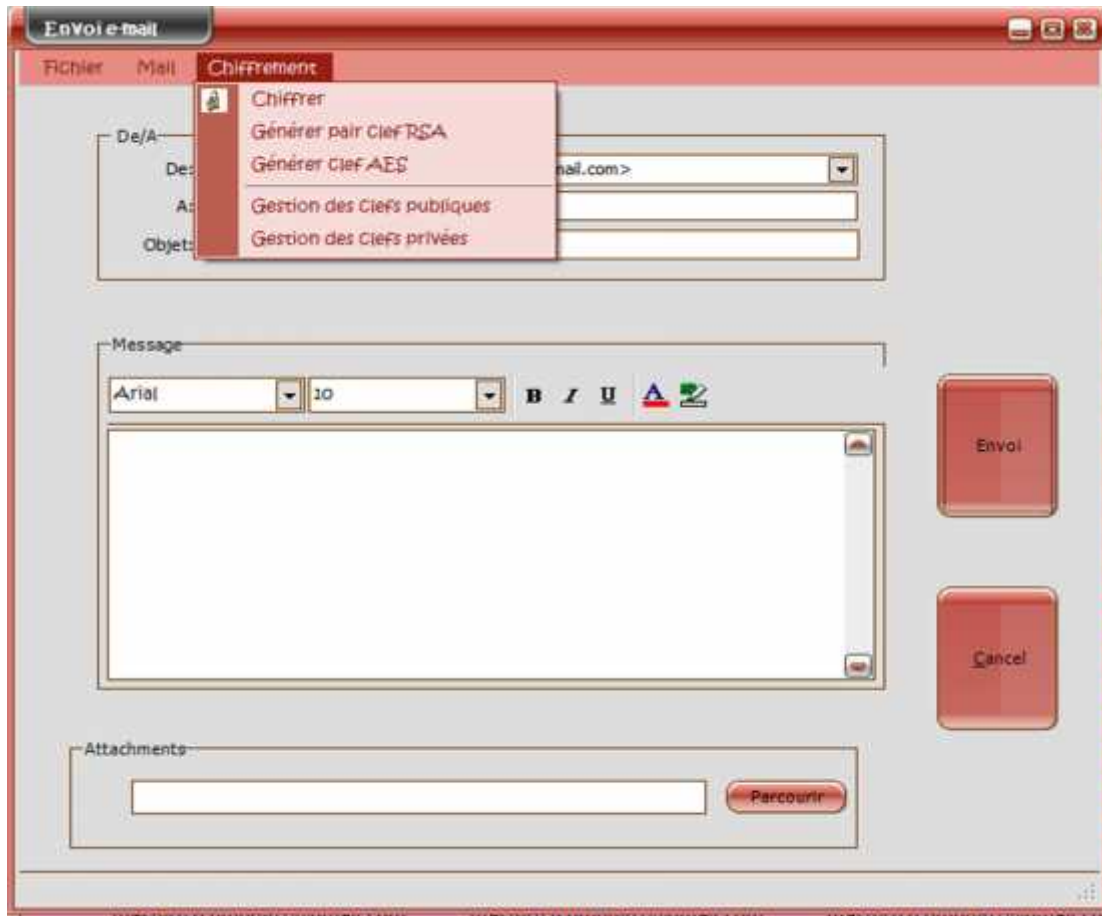





Figure 4.14 : Ecrire un message

Pour chiffrer un mail appuyé sur *Chiffrer* dans le menu *chiffrement*, le système cherche automatiquement la clef publique de destinataire, et sa clef privé (secrète) s'il existe, et faire premièrement de générer une clef de session d'algorithme Rijndael et crypte le corps de message avec cette clef, puis cherchent la clef secrète, s'il existe, puis crypte la clef de session et sa matrice initial (IV) avec cette clef, puis crypte (la clef de session crypté) avec la clef publique de destinataire, sinon crypte la clef et sa matrice initial directement avec la clef publique de destinataire.

Et pour l'envoi du mail appuyer sur le boutons envoi ou bien choisir *envoi* dans le menu *mail*, mais pour joindre un fichier appuyer sur le boutons *parcourir* puis choisir la place du fichier. si la connexion internet est bien existe le mail est envoyer et enregistrer dans la boîte des *mails envoyés* sinon l'email n'envoyé pas et enregistrer dans la boîte des emails en attentes pour l'envoyer ultérieurement, et pour envoyer ces

derniers appuyer sur l'icône  dans la barre de taches, ou bien choisir *Envoyer* dans le menu *Message*.

Maintenant pour recevoir les mails reçus appuyer sur l'icône  ou bien choisir *Recevoir* dans le menu *Message*. Ce dernier peut être crypté par l'émetteur, et on peut savoir que ce mail est crypté par la colonne *Crypté* dans la *Listview* de la forme principale.


Pour décrypter ces mails cliquer sur l'icône  ou bien choisir *Déchiffrer mail* dans le menu *Chiffrement* comme la figure suivant :


Date	Crypté
06/06/2012 09:35:28	
06/06/2012 09:35:35	✖
06/06/2012 09:35:41	
06/06/2012 09:30:39	

Mail né pas crypté

Mail crypté

Figure 4.15 : les mails cryptés

Pour supprimer un mail sélectionner ce dernier et cliquer sur l'icône  ce mail sera stocké dans la boîte des *éléments supprimés*.

Pour Imprimer un mail, ouvrir le mail puis cliqué sur l'icône  dans la barre de taches ou bien choisir *Imprimer* dans le menu *Fichier*.

4. Test du programme

Jeu d'essai 01

On va crypter le message suivant :

« Salut tout le monde, c'est le premier jeu d'essai 01 »

Nous avons créé deux comptes, le premier pour l'envoi

(master02compte01@gmail.com) et le deuxième pour la réception

(master02compte02@gmail.com), et entrer les informations des deux comptes

précédents sur le logiciel, puis générer une clef publique RSA et une autre privée,

finallement générer une clef secrète AES et entrer ces clefs au logiciel, la figure suivant monter comment crypté et envoi ce mail :

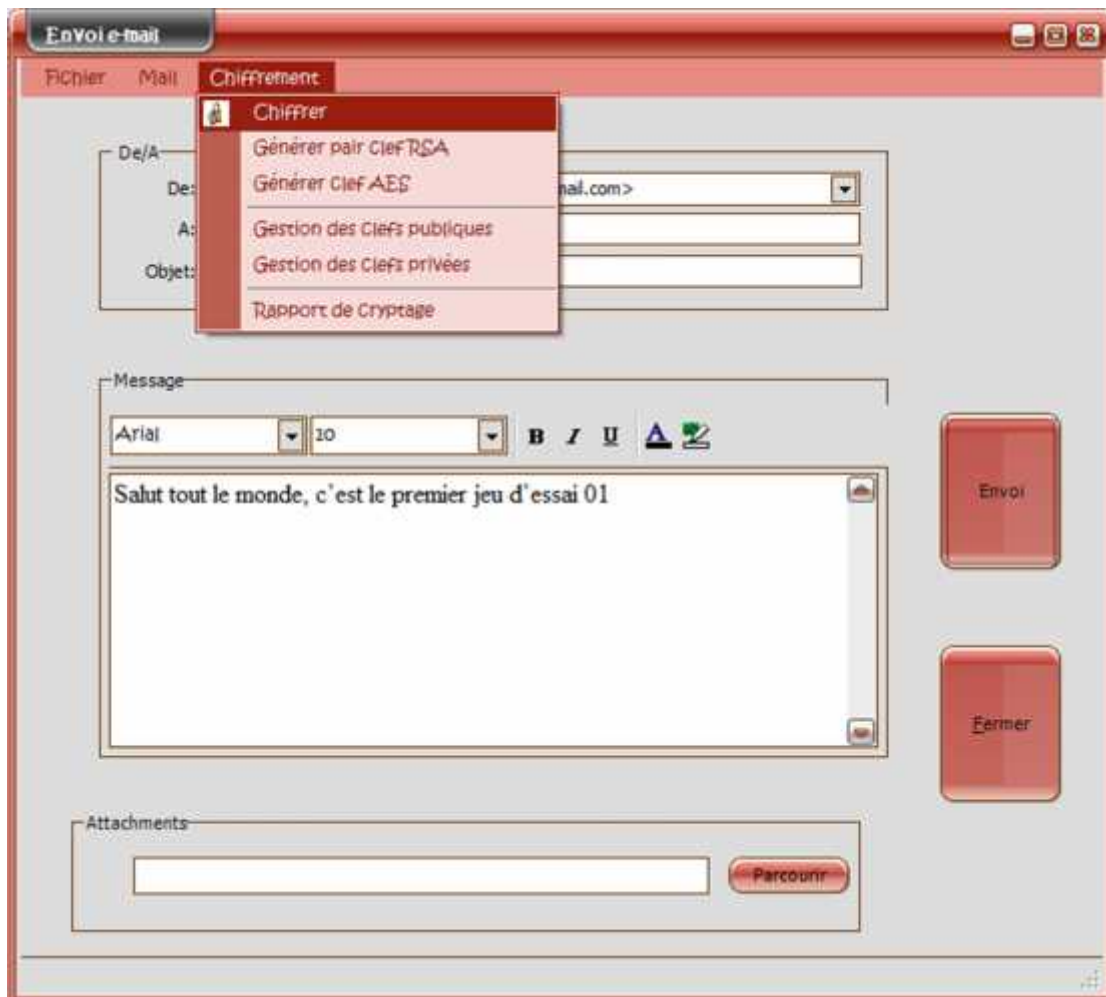


Figure 4.16 : les mails cryptés

Puis chiffrer et envoi ce mail comme suit :

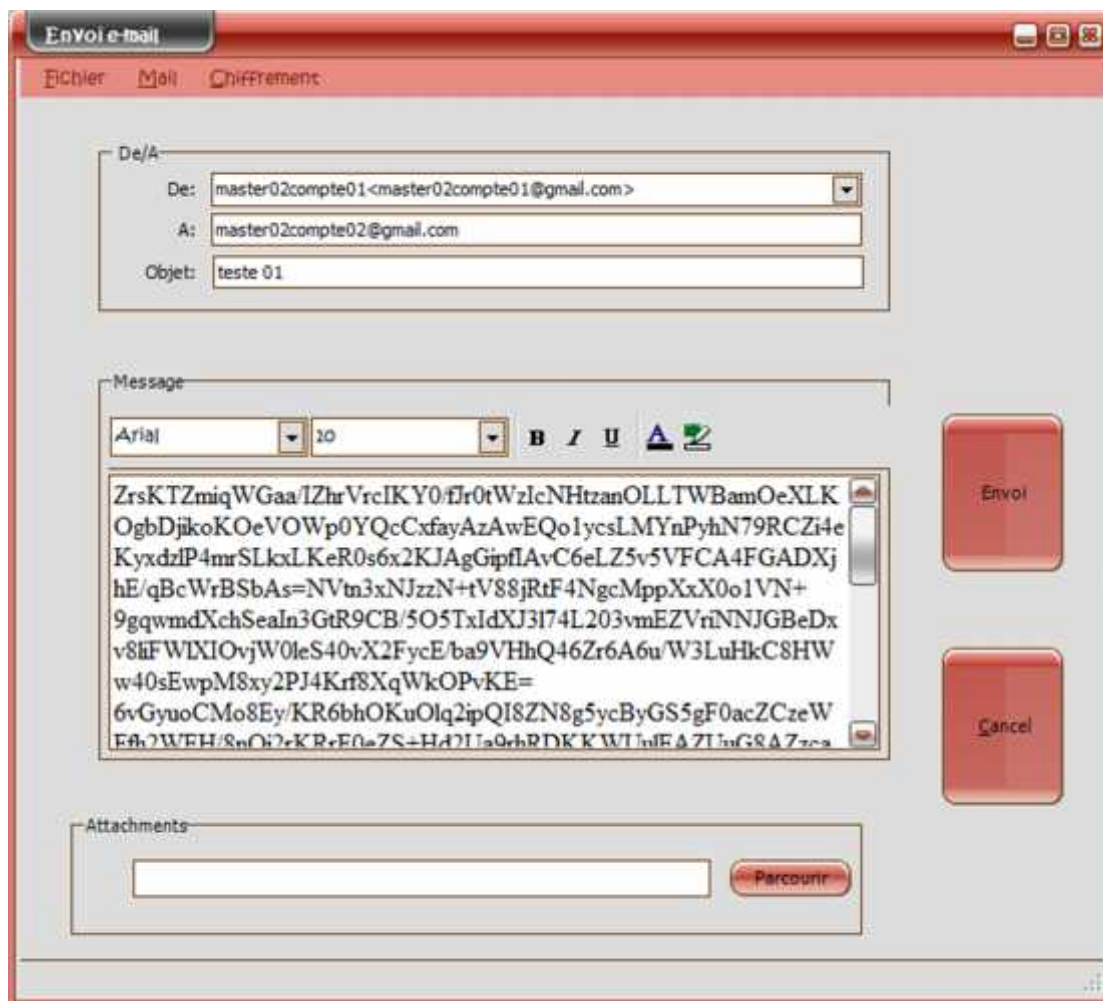


Figure 4.17 : envoi un mail crypté

Maintenant check le boîte de réception d'autre compte et après le recevoir, ouvrir cette mail comme la figure suivante :



Figure 4.18 : envoi un mail crypté

Et après le déchiffrement, on obtient le texte original comme la figure suivante:

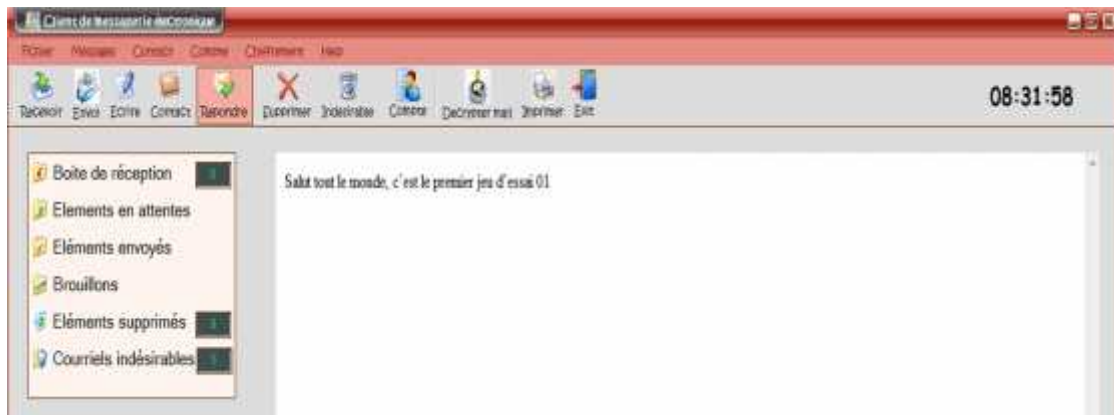


Figure 4.19 : Déchiffrer un mail reçu

Et aussi on peut envoyer un mail clair, comme suit :

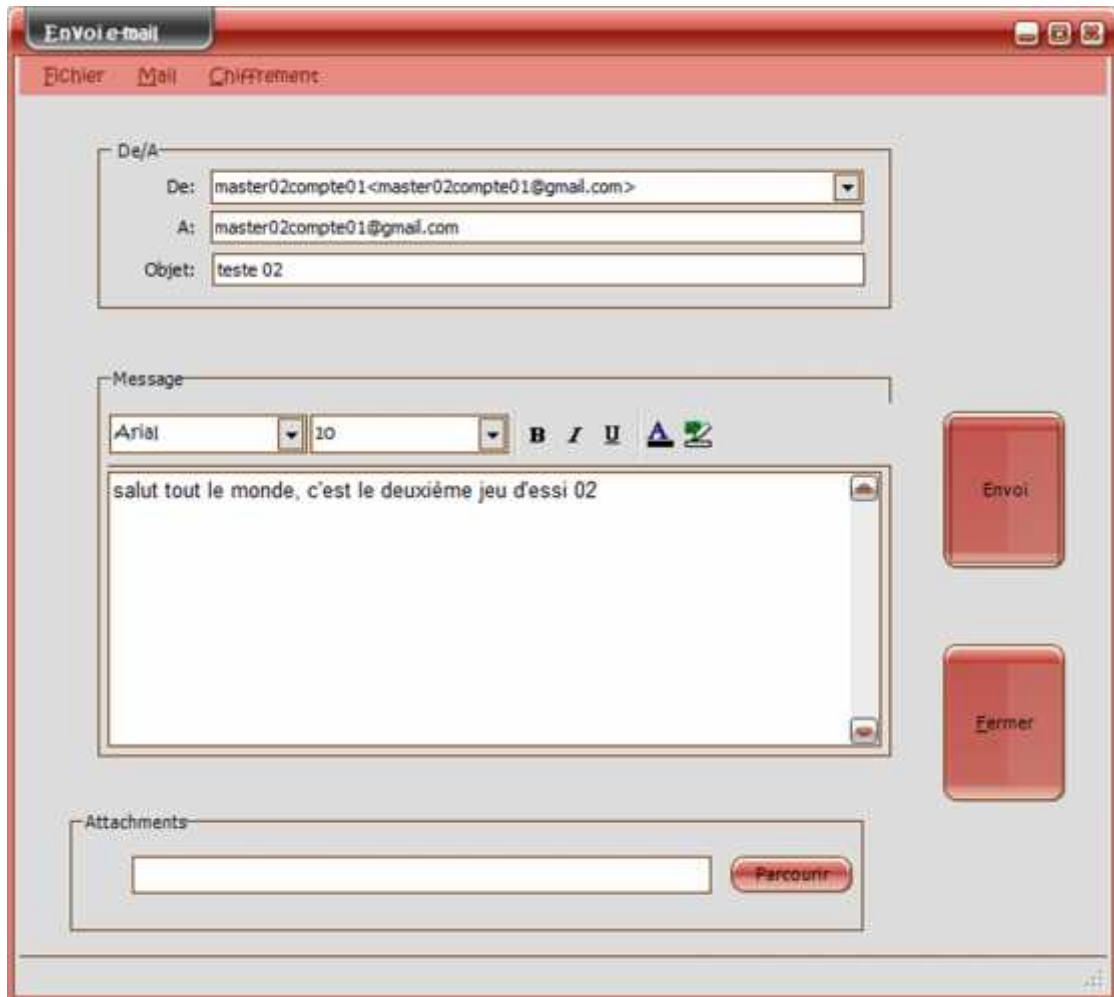


Figure 4.20: Envoi un mail clair

Le mail reçu apparaît dans la figure suivante :

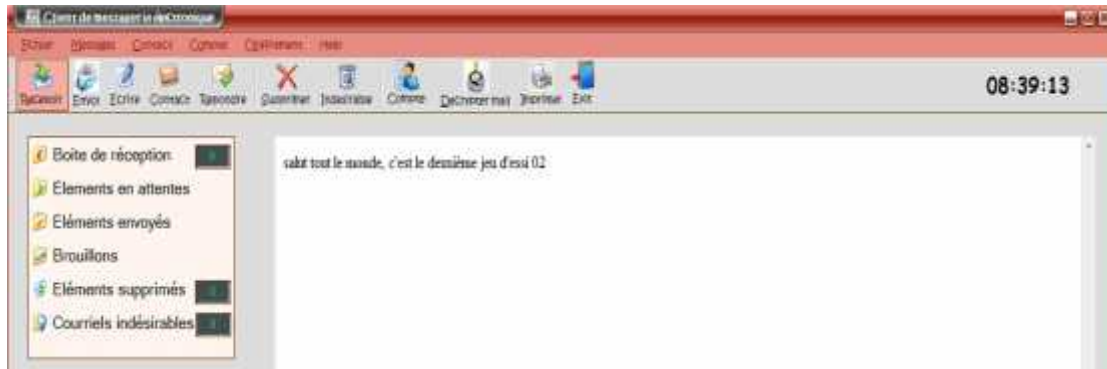


Figure 4.21: recevoir un mail clair

5. Les étapes de réalisation

Tout au long de la phase de développement, nous avons commencé par la conception de notre logiciel avec le langage UML, après cette étape nous avons installé le Visual Studio 2010 pour utiliser le langage C# dans cet environnement, et concevoir les interfaces principales, puis chercher les composants nécessaires. En fin nous avons fait la réalisation et les tests.

6. L'installation

Pour faire fonctionner le logiciel sur une machine, il faut l'installer, cette installation consiste à copier le répertoire contenant l'exécutable et tous les fichiers nécessaires vers la machine en question. Pour lancer le programme, il suffit alors de lancer l'exécutable en question.

7. Conclusion

Dans ce chapitre nous avons présenté l'environnement matériel et logiciel, pour notre travail, Par la suite, nous avons exposé quelques interfaces homme machine qui présentent avec les fonctionnalités du système et terminer le chapitre par la présentation des tests.

CONCLUSION

GENERALE

CONCLUSION

L'objectif de ce travail était d'étudier et concevoir un logiciel de messagerie électronique pour la sécurité des courriers électroniques, fonctionne sur le réseau internet, et aussi permettre de la génération des clefs (publiques et privés) des cryptages.

Nous nous sommes intéressées dans ce mémoire au fonctionnement des messageries électroniques et leurs clients (agents), et les protocoles utilisés lors de l'envoi et l'acheminement des courriers, ainsi les techniques utilisées aujourd'hui pour le cryptage des courriers électronique comme PGP. Aussi on étudier dans ce travail la méthode de cryptographie symétrique Rijndael (standard de NIST), et la cryptographie asymétrique RSA, ces méthodes sont largement utilisées dans nos jours.

Pour un simple but de découvrir les secrets et astuces de la programmation en C#, donc d'apprendre plus, on a volontairement intégré des composantes sur l'environnement de visuel studio (Openpop.dll, irisskin.dll) pour utiliser d'autre fonctionnalités.

Nous considérons que l'essentiel de notre but est atteint même s'il reste des choses à ajouter, et comme perspectives nous proposons :

- 1- L'implémentation d'un web mail permettant de gérer le chiffrement des emails sortants et le déchiffrement des emails entrants selon l'amélioration qui nous avons ajouté.
- 2- L'intégration des algorithmes de comprissions des données dans notre client de messagerie pour minimiser la taille des donnés envoyer, et pour le but de difficulté l'attaque.
- 3- Intégration le service de signature numérique et les autres fonctions de PGP pour l'obtention d'un protocole plus sécurisé.
- 4- Intégration de chiffrement des fichiers attachés dans notre logiciel.

En fin, on n'a pas présenté un travail exhaustif, hermétique à tout questionnement futur. En élaborant une solution possible parmi tant d'autres, on a voulu contribuer modestement à la résurgence d'autres idées, à même donner une nouvelle impulsion à la science informatique.

BIBLIOGRAPHIE

BIBLIOGRAPHIE

[1] : Jean-Philippe Gaulier, « *Analyse des algorithmes finalistes concourant pour le futur standard AES* », Mémoire de synthèse soumis dans le cadre d'un probatoire en vue de l'acquisition d'un diplôme en Ingénierie et Intégration Informatique Systèmes d'Information

[2] : Fabien GARGNE, Christian KNOFF, Gaëtan LECOURTOIS, « *Codage Compression et Cryptologie* », 2004–2005, Université de Nice-Sophia Antipolis
<http://www.deptinfo.unice.fr/twiki/pub/Linfo/>

[3] : Bourgeois Morgan, « *Initiation à PGP : GnuPG* », 19/07/2006
<http://www.mbourgeois.developpez.com/articles/>

[4] : Jonathan BLANC, Adrien DE GEORGES, « *TECHNIQUES DE CRYPTOGRAPHIE* »
<http://www.deptinfo.unice.fr/twiki/pub/Linfo>

[5] : Serge Aumont, Roland Dirlwanger, Olivier Porte, « *L'accès sécurisé aux données* », Novembre 1999
<http://www.1999.jres.org/tutoriaux/tutorial4-chiffrement.pdf>

[6] : David Pointcheval, « *Le Chiffrement Asymétrique et la Sécurité Prouvée* », 17 juin 2002, Université Pris7 Habilitation à Diriger des Recherches.
http://www.www.di.ens.fr/~pointche/Documents/Reports/2002_HDRThesis.pdf

[7] : Stephan Robert, *Eléments de cryptographie*, Septembre 2005
http://www.stephan-robert.ch/attachments/File/Networking/crypto_v10-corr1.pdf

[8] : Destree Lucile, Marchal Mickaël, « *Mini-RSA Programme d'initiation au chiffrement RSA* »
http://www.lesitedemika.org/ressources/cryptographie_rsa.pdf

- [9] : Jean-Guillaume Dumas, « *factorisation d'entiers, cryptographie* »
<http://www.ljk.imag.fr/membres/Jean-Guillaume.Dumas/>
- [10] : Emonet Jean-Bruno, « *Algorithmes de chiffrement Mesures de performances réseaux* », 24 juin 2005
http://www.www.rd.cri74.org/repository/securite/algo_chiffrement.pdf
- [11] : « *Introduction à la cryptographie* », 09/02/01, Support de cours du cabinet Hervé Schauer Consultants (HSC)
<http://www.hsc.fr/ressources/cours/crypto/crypto.pdf>
- [12] : Gilles Dubertret, « *INITIATION A LA CRYPTOGRAPHIE* », octobre 1998, Vibert
- [13] : A.Gosselin, « *Le courrier électronique* », Janvier 1998
http://www.etab.ac-caen.fr/montchamp/enseign/documnts/uti_mail.pdf
- [14] : Stéphane Lohier, Dominique Présent, « *internet : services et réseaux* », paris, 2004, Dunod
- [15] : CLUB DE LA SECURITE DES SYSTEMES D'INFORMATION FRANÇAIS, « *SECURITE DE LA MESSAGERIE* », Septembre 2005
http://www.clusif.asso.fr/fr/production/ouvrages/pdf/Securite_Messagerie.pdf
- [16] : « *Le courrier électronique* »
http://www.another-teacher.net/IMG/pdf/05_Le_courrier_electronique__mis-en-forme_custom.pdf
- [17] : Carherine Szaibrum, « *Internet initiation* », paris 2000, DUNOD
- [18] : DOMINIQUE LACHIVER, « *Courrier électronique* », Septembre 2010
http://lachiver.fr/supports/opale/tic/internet/courrier/courrier_papier.publi/paper/courrier_papier.pdf

[19] : GAADI Mohamed, ABDRUBO MOHAMED BAKER Gehad, « *Conception et Réalisation d'un Système de Messagerie Electronique dans un Intranet* », 2009-2010, Mémoire de fin d'études.

[20] : « *Crypter le courrier* »

<http://www.competencemicro.com/supplements/inetsecu/pgp.pdf>

[21] : LAES : Advanced Encryption Standard, May 17, 2012

<http://www.securiteinfo.com/cryptographie/aes.shtml>

[22] : <http://www.commentcamarche.net/contents/base/base64.php3>

[23] : <http://office.microsoft.com/fr-ch/outlook-help/chiffrer-des-messages-electroniques-HP010355559.aspx>

[24] : Fabián Rodríguez, « *Chiffrer son courriel avec Enigmail* », 28 septembre 2004, Guide d'installation et d'utilisation pour Mozilla Thunderbird, Enigmail et WinPT
<http://www.framasoft.net/IMG/tb-enigmail.pdf>

[25] : <http://www.commentcamarche.net/contents/base/ascii.php3>

NOMENCLATURE

Nomenclature

ASCII : American Standard Code for Information Interchange.

AES : Advanced Encryption Standard.

BAL : Boîte Aux Lettres.

BCC : Blind Carbone Copy.

CC : Carbone Copy ou Copie Conforme.

CCI : Copie Conforme Invisible.

DES : Data Encryption Standard.

EBCDIC : Extended Binary-Coded Decimal Interchange Code

ESMTP : Extended Simple Mail Transfer Protocol.

GPG : Gnu Privacy Guard.

HTTP : Hyper Text Transfer Protocol.

IDEA : International Data Encryption Algorithm.

IETF : Internet Engineering Task Force.

IMAP : Interactive Mail Access Protocol.

ISO : International Standards Organization.

MD5 : Message Digest.

MIME : Multipurpose Internet Mail Extensions.

MTA : Message Transport Agent.

MUA : Mail User Agent.

NIST : National Institute of Standards and Technology.

PGP : Pretty Good Privacy.

POP3 : Post Office Protocol version 3.

QP : Quoted Printable.

RAM : Random Access Memory

RC2 : Release Candidate 2.

RIPEMD : Ripe Message Digest.

Rijndael : Contraction des noms des deux inventeurs : Dr. Joan Daemen, Dr. Vincent Rijmen.

RSA : nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman.

SHA-1 : Secure Hash Algorithm 1.

SMTP : Simple Mail Transfer Protocol.

SSL : Secure Socket Layer.

TLS : Transport Layer Security.

UA : User Agent.

URL : Uniform Resource Locator.

US-ASCII : United State American Standard/Society Code for Interchange.

UTF8 : Universal Transformation Format-8.

3DES : Triple Data Encryption Standard.

Résumé

La communication est un aspect essentiel de la vie humaine, de nouvelles technologies de l'information et de la communication se développent continuellement en offrant de nouvelles potentialités. Notre travail, se situe dans le cadre du développement d'un système de cryptage pour un client de messagerie électronique dans internet permettant ainsi une bonne communication sécurisée entre les sociétés et les personnes. Pour ce faire, on a conçu un logiciel de messagerie électronique permettant de gérer le cryptage des emails sortants et décryptage des emails entrants, utilisant le mécanisme de PGP avec quelque amélioration. La conception est a été réalisée par le langage UML. L'implémentation a été développée par le langage orienté-objet C# dans l'environnement de Visual studio 2010.

Mots clés : Cryptographie, Messagerie électronique, Cryptage Messagerie électronique, Algorithme de Cryptage, Protocoles de messagerie électronique, UML.

Abstract

The Communication is an essential aspect of human life, new technologies of information and communication develop continually offering new potentialities. Our work is situated in the development of a system of encryption for e-mail client in Internet allowing secure good communication between companies and individuals. To do this, we designed an e-mail software used to manage the encryption of outgoing emails and decryption of incoming emails, using the PGP mechanism with some improvements. The design was carried out by the UML. The implementation was developed by the object-oriented language C # in Visual Studio 2010 environment. **Keywords:** Cryptography, Email, E-mail Encryption, Encryption Algorithm, email protocols, UML.

ملخص

الاتصال هو أحد الجوانب الأساسية للحياة البشرية، والتكنولوجيات الجديدة للمعلومات والاتصالات تتطور باستمرار لتقديم إمكانيات جديدة. ويتمثل عملنا في تطوير نظام التشفير لعمل البريد الإلكتروني في الإنترنت، كما يسمح بالتواصل الجيد و الأمن بين الشركات والأفراد. للقيام بذلك، قمنا بتصميم عميل بريد إلكتروني يسمح بتشفير الرسائل الصادرة وفك تشفير الرسائل الواردة ، وذلك باستخدام تقنية PGP مع بعض التحسينات. وقد تم التصميم بواسطة لغة التمثيل UML. كما تمت البرمجة بواسطة لغة السي شارب في محيط العمل فيجول ستوديو2010.

الكلمات المفتاحية: التشفير، البريد الإلكتروني، تشفير البريد الإلكتروني ، خوارزميات التشفير، بروتوكولات البريد الإلكتروني، UML.