الجمهوريـة الجزائـريـة الديمقراطيـة الشعبيـة
République Algérienne Démocratique et Populaire
وزارة التعليـم العالـي والبحـث العلمـي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

N° Réf :……

Centre Universitaire
Abd elhafid Boussouf Mila

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

Mémoire préparé en vue de l'obtention du diplôme de Master
En : Informatique

Spécialité: Sciences et Technologies de l'Information et de la Communication (STIC)

Gestion de la sécurité du courrier électronique à l'aide de protocoles cryptographiques

Préparé par : - **Sairi Amira**
- **Djader Samira**

Soutenue devant le jury :

Encadré par : Boumassata Meriem…….. M.A.B
Président : Talai Meriem……..... M.A.A
Examinateur : Mezzoudj Saliha…..…….M.A.B

Année universitaire : 2015/2016

Remerciement

Merci Allah de m'avoir donné la capacité d'écrire et de réfléchir, la force d'y croire, la patience d'aller jusqu'au bout du rêve et le bonheur de lever mes mains vers le ciel et de dire " Ya Kayoum ".

Un grand remerciement à notre encadreur Madame « Boumassata Meriem » de nous avoir accordé sa confiance et permis de réaliser ce travail avec elle.

A travers ses qualités professionnelles, en tant que directrice de ce travail, elle nous a transmis de précieuses connaissances.

Merci également pour sa disponibilité, sa patience et sa bonne humeur constantes qui ont rendu ce travail très agréable et enrichissant, ainsi que pour sa rigueur scientifique.

Mes remerciements vont également aux membres de jury d'avoir accepté de juger mon travail.

Je remercie tous les personnes qui m'ont aidé durant la préparation de ce mémoire de près ou de loin.

Nous n'oublions pas nos enseignants qui tout au long du cycle d'étude au centre universitaire de Mila.

Dédicace

Tout d'abord, je rends grâce à dieu tout puissante qui nous a aidés à terminer ce mémoire.

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, À mes chers parents qui m'ont éclairé le chemin de la vie par leur grand soutien et leurs encouragements, par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voire réussir.

À mes frères et mes sœurs et surtout leurs enfants que je les aime énormément

À mon binôme AMIRA

À toute ma famille

À tous mes amies surtout AMEL, ASMA, AMEL MARIEM, IKRAM, AHLEM, MARIA, IMAN, KENZA AMEL, IBTISSAM

À Tous Mes Collègues d'étude

À Tous Mes Enseignants

À Tous qui me connaissent

SAMIRA

Dédicace

Tout d'abord, je rends grâce à dieu tout puissante qui nous a aidés à terminer ce mémoire.

Je dédie ce modeste travail à celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, À mes chers parents ZAHRA ET RABEH qui m'ont éclairé le chemin de la vie par leur grand soutien par leurs dévouements exemplaires et les énormes sacrifices qu'ils m'ont consentis durant mes études et qui ont toujours aimé me voire réussir.

À mes frères FATEHE, SAMAH, SLEMANE, FAIZA MARWA et belle-sœur HANANE et ma grand-mère FATIMA que je les aime énormément.

À les enfants du mon frère SOHAIB ABD ARAHIME et ma cher DHOHA et ARIJ.

À mon binôme SAMIRA

À toute ma famille

À tous mes amies surtout AMEL, ASMA, MARIEM IKRAM, AMEL, IMAN, KENZA, AHLEM, MARIA AMEL, IBTISSAM

À Tous Mes Collègues d'étude

À Tous Mes Enseignants

À Tous qui me connaissent.

AMIRA

RÉSUMÉ

Le service de messagerie électronique en raison de sa simplicité d'utilisation combinée à son efficacité, a constitué l'un des principaux vecteurs de popularisation d'Internet. Il est devenu un service incontournable dont la richesse s'exprime au travers des usages variés et multiples qu'il autorise (privé, professionnel, administratif, officiel, militaire, etc.).

La sécurité des données qui peuvent contenir des informations sensibles, transitant entre deux interlocuteurs ou plusieurs, via un vecteur d'information comme les réseaux de télécommunications actuels est une préoccupation très importante.

Dans ce mémoire nous allons développer une application de messagerie électronique sécurisée que nous avons appelé « Secure Mail ». Il s'agit d'une application comportant un serveur de messagerie et un client de messagerie lourd intégrants des protocoles cryptographiques de chiffrement et de signature numérique. L'application permet aux utilisateurs du système d'échanger du courrier électronique d'une manière sécurisée dont le but est d'assurer la confidentialité et l'intégrité des messages échangés entre les utilisateurs.

Mots clés: messagerie, courrier électronique, sécurité, cryptographie, chiffrement signature électronique.

TABLE DES MATIERES

Chapitre 02: Sécurité du courrier électronique

Chapitre 03: Conception de l'application de la messagerie électronique sécurisée

Chapitre 04: Implémentation

LISTE DES FIGURES

LISTE DES TABLES

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

En raison de sa simplicité d'usage combinée à son efficacité, le service de messagerie électronique, a constitué l'un des principaux vecteurs de popularisation d'Internet. Ce service a été progressivement introduit dans les environnements professionnels en se substituant aux services du fax et du courrier postal. Les différentes fonctionnalités offertes par le courrier électronique tel que l'envoi de documents électroniques, la rapidité d'acheminement sans la contrainte géographique, la possibilité d'envoi d'un message à des destinataires multiples représentent quelques atouts importants par rapport aux limitations du courrier postal.

Le succès croissant de la messagerie électronique a également été accompagné d'une forte évolution des menaces. Il est important de noter que ce service a initialement été conçu sans prendre en compte les aspects de sécurité. Ce manque a été partiellement comblé dans le courant des années 2000 avec l'apport de services de sécurité mais ceci ne répond pas encore totalement aux exigences des particuliers et des entreprises. Il est d'ailleurs intéressant de noter que 75% des informations critiques d'une entreprise se retrouvent dans les courriels [1] [2].

L'un des plus importants mécanismes utilisés pour assurer la sécurité des communications réseaux est la cryptographie. Ce mécanisme permet d'assurer la confidentialité des messages transmis dans le réseau par le chiffrement et l'intégrité de ces messages par la signature numérique.

Il existe deux niveaux de protections qui permettent d'assurer la transmission de messages électroniques sécurisés : le niveau protocolaire et le chiffrement du contenu des messages.

Les protocoles réseaux (SMTP, POP et IMAP) permettent tous de chiffrer les connexions. Ce chiffrement assure que le message ne pourra pas être lu pendant l'envoi ou de la réception. Mais le message sera stocké en clair dans la boite du destinataire et sur les zones de stockage temporaire de tous les serveurs de messagerie qui auront participé au transport.

Pour assurer une complète confidentialité, il faut chiffrer le contenu du message lui-même de façon à ce qu'il ne soit lisible que par le destinataire [3].

Notre projet consiste à développer un système de messagerie électronique sécurisée. Il s'agit d'une application comportant un serveur de messagerie et un client de messagerie lourd. Au sein de cette application, nous implémentons et intégrons quelques protocoles cryptographiques, dans le but d'offrir à l'utilisateur de cette application un service qui permet le chiffrement des messages envoyés et le déchiffrement des messages reçus, ainsi que la signature numérique des messages envoyés et la vérification de la signature des messages reçus.

Le reste du mémoire sera organisé en quatre chapitres comme suit :

Dans le premier chapitre, nous présenterons quelques généralités sur les réseaux informatiques et l'internet. Puis, nous présenterons l'architecture Client/Serveur, son principe de fonctionnement et ses types. Nous présenterons, ensuite, les concepts liés à la messagerie électronique, l'architecture de fonctionnement d'un système de messagerie, ainsi que les différents protocoles et outils utilisés dans la messagerie électronique.

Dans le deuxième chapitre, nous aborderons la sécurité du courrier électronique. Nous présenterons les propriétés de la sécurité informatique, les attaques et les mécanismes de sécurité. Nous détaillerons, ensuite, les mécanismes cryptographiques utilisés pour sécuriser les communications réseaux. Puis nous présenterons les attaques existantes sur la messagerie électronique et les mécanismes cryptographiques utilisés contre ces attaques.

Dans le troisième chapitre, nous présenterons les concepts du langage de modélisation unifié UML et ses diagrammes, ainsi que les différentes étapes de la méthode de développement, inspirée des étapes du processus UP, que nous avons choisi pour la conduite de notre projet. Nous expliquerons, ensuite les fonctionnalités de notre application et nous détaillerons les diagrammes de cas d'utilisation, les diagrammes de séquence, le modèle de domaine, quelques digrammes d'interaction et enfin le modèle de déploiement.

Dans le quatrième chapitre, nous présenterons la réalisation de notre projet. Nous commencerons par la présentation de l'environnement logiciel utilisé pour la réalisation du système qui est composée essentiellement du langage de programmation JAVA, l'environnement de développement NetBeans et le système de gestion de base de données Mysql. Et enfin, nous présenterons quelques interfaces de l'application.

CHAPITRE 01 : MESSAGERIE ÉLECTRONIQUE

1. INTRODUCTION

Le service de messagerie électronique dans un intranet même à l'internet représente près de 90% des échanges d'informations entre les humains, les entreprises ou organisations entre les nations [4].

C'est le service le plus populaire et le plus utilisé dans un réseau informatique, dont l'envoi de documents électroniques en attachement, la conservation et l'archivage des messages sont beaucoup plus faciles à effectuer qu'avec les communications écrites et téléphoniques. De plus, au niveau d'utilisateur, le courrier électronique permet d'effectuer un traitement rapide, efficace et automatique sur les messages tel qu'il représente toutes les caractéristiques du courrier classique : envoi par dépôt à la poste, centre de tri, distribution, et boites aux lettres.

Dans ce chapitre, nous allons présenter, d'une façon générale les concepts de réseaux informatiques, d'internet et d'architecture Client/Serveur. Nous présenterons ensuite l'architecture et le fonctionnement d'un système de messagerie électronique, les différents protocoles utilisés par un système de messagerie, les formats standards de messages, et les outils de la messagerie électronique.

2. INTERNET ET ARCHITECTURE CLIENT/SERVEUR

2.1. RÉSEAU INFORMATIQUE

Le terme générique « réseau » définit un ensemble d'entités (objets, personnes, etc.) interconnectées les unes avec les autres. Un réseau permet ainsi de faire circuler des éléments matériels ou immatériels entre chacune de ces entités selon des règles bien définies.

Un réseau informatique est un ensemble d'ordinateurs reliés entre eux grâce à des lignes physiques et échangeant des informations sous forme de données numériques [5].

2.2. INTERNET

C'est une interconnexion de réseaux répartis dans le monde (INTERnational NETwork). Grâce à un langage commun (le protocole TCP/IP - Transfert Control Protocole / Internet Protocol), des ordinateurs très divers, d'un point de vue tant matériel que logiciel (PC, Macintosh, stations de travail, serveurs, gros ordinateurs, fonctionnant sous Unix, Linux, AIX, VMS, MacOS, Windows 95, 98, NT, 2000, etc.) reliés en réseaux, sont interconnectés entre eux. Il s'agit donc d'un réseau de réseaux [6].

2.3. SERVICES ET PROTOCOLES D'INTERNET

Un protocole de communication est un ensemble de règles et de procédures permettant de définir un type de communication particulier [7]. Sur Internet, de nombreux protocoles sont utilisés, ils font partie d'une suite de protocoles qui s'appelle TCP/IP «Transmission Control Protocol/Internet Protocol». TCP/IP repère chaque ordinateur par une adresse appelée adresse IP qui permet d'acheminer les données à la bonne adresse. Puis on associe à ces adresses des noms de domaine (le nom de la machine dans un domaine) pour une utilisation plus facile de la navigation Internet.

Les services utilisés dans l'Internet sont les services développés aux dessus de l'architecture TCP/IP. On distingue:

- **Le service d'émulation de terminal:** Le service Telnet (TErminaL NET-protocol) permet d'utiliser une station en réseau comme un terminal informatique directement raccordé au calculateur. Dans la majorité des cas, il faut disposer d'un compte sur le calculateur en question.
- **Le courrier électronique:** Le courrier électronique est le service le plus utilisé sur Internet, après s'être répandu dans le monde professionnel, il est devenu l'un des outils favoris des particuliers car il permet de communiquer en différé facilement et rapidement. De plus, il offre la possibilité d'envoyer et de recevoir des documents attachés au message, ce qui en fait un outil d'échange performant et intéressant. Ajoutons à cela qu'on peut tout aussi facilement échanger des messages entre des personnes qu'à l'intérieur de tout un groupe [8].
- **Le forum électronique:** C'est une messagerie électronique ouverte, dans laquelle chacun peut écrire et tout le monde peut consulter.
- **Le transfert de fichiers:** Le protocole FTP (File Transfer Protocol) permet outre le transfert, l'accès et la gestion de fichiers à distance. Cette manipulation sur les fichiers doit être autorisée. Des serveurs FTP anonymes permettent d'accéder à une partie de leurs données quel que soit l'utilisateur client.
- **Le Web (World Wide Web) :** permet d'accéder à des pages web, en utilisant le protocole HTTP (HyperText Transfer Protocol). Le web est l'application Internet la plus populaire. Grace à un navigateur web (browser), un utilisateur (internaute) peut lire des pages web stockées sur un ordinateur serveur situé n'importe où dans le monde.

- **Moteur de recherche :** Un moteur de recherche est une application web permettant de retrouver des ressources (pages web, articles de forums, images, vidéos, fichiers, etc.) associées à des mots quelconques. Certains sites web offrent un moteur de recherche comme principale fonctionnalité, on appelle alors moteur de recherche le site lui-même (Google Vidéo par exemple est un moteur de recherche vidéo) [4] [9].

2.4. ARCHITECTURE "CLIENT / SERVEUR"

2.4.1. Définitions

L'architecture Client/Serveur est un modèle de fonctionnement logiciel qui peut se réaliser sur tout type d'architecture matérielle (petites ou grosses machines), à partir du moment où ces architectures peuvent être interconnectées [10].

- **Serveur :** On appelle logiciel serveur un programme qui offre un service sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur. Le terme serveur s'applique à la machine sur laquelle s'exécute le logiciel serveur. Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec accès permanent et s'exécuter en permanence.
- **Client :** On appelle logiciel client un programme qui utilise le service offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut être raccordé par une liaison temporaire [11].
- **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 le résultat de l'opération [12].

2.4.2. Fonctionnement d'un système Client/Serveur

On parle de fonctionnement logiciel dans la mesure où cette architecture est basée sur l'utilisation de deux types de logiciels, à savoir un logiciel serveur et un logiciel client s'exécutant normalement sur 2 machines différentes. Le dialogue entre les applications peut se résumer par :

- Le client demande un service au serveur : le client émet une requête vers le serveur grâce à son adresse IP et le numéro de port, qui désigne un service particulier du serveur.

- Le serveur réalise ce service et renvoie le résultat au client : le serveur reçoit la demande et répond à l'aide de l'adresse IP de la machine cliente et son numéro de port [13].

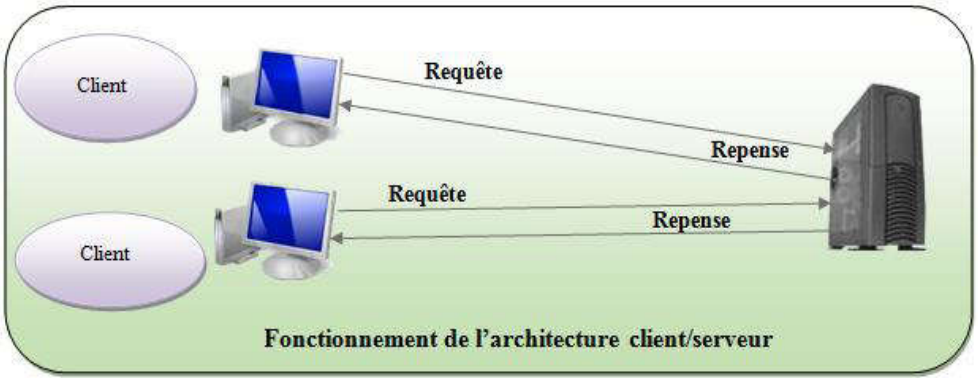

Figure 1: Fonctionnement d'un système Client/Serveur [13]

2.4.3. Types de serveurs

- **Serveur de fichiers :** Le client (généralement un PC) requiert des enregistrements de fichiers en émettant des requêtes sur le réseau en direction d'un serveur de fichiers**.**
- **Serveur de base de données :** Le client émet des requêtes SQL sous forme de message. Le serveur renvoie le résultat de chaque requête [14].
- **Serveur d'application web :** Le World Wilde Web est la première véritable application Client/Serveur intergalactique. Ce modèle consiste en des clients qui communiquent avec des très gros serveurs. Dans sa concrétisation la plus simple, un serveur web renvoie des documents lorsque le client les demandes par leur nom [15].

2.4.4. Types d'Architectures Client/serveur

❖ **Architecture à 2 niveaux**

L'architecture à deux niveaux (aussi appelée architecture 2-tiers) caractérise les systèmes clients-serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service [16].

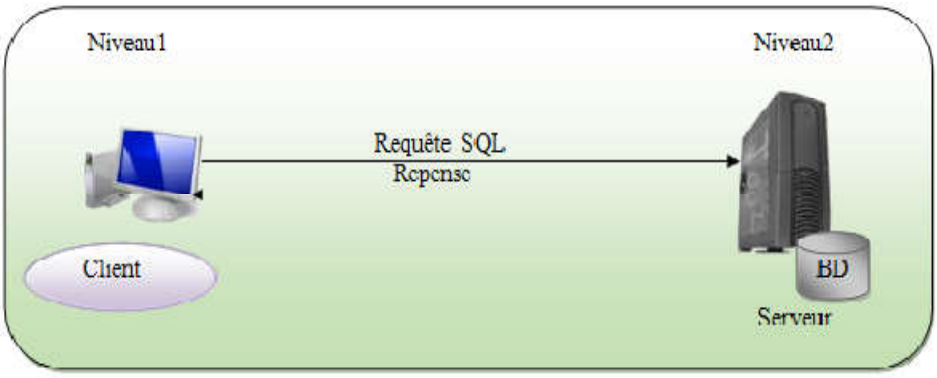

Figure 2: Architecture à 2 niveaux [13].

❖ **Architecture à 3 niveaux**

Dans l'architecture à 3 niveaux (appelée architecture 3-tiers), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur chargée de la présentation ;

Le serveur d'application (appelé également middleware), chargé de fournir la ressource mais faisant appel à un autre serveur ;

Le serveur de bases données, fournissant au serveur d'application les données dont il a besoin [16].

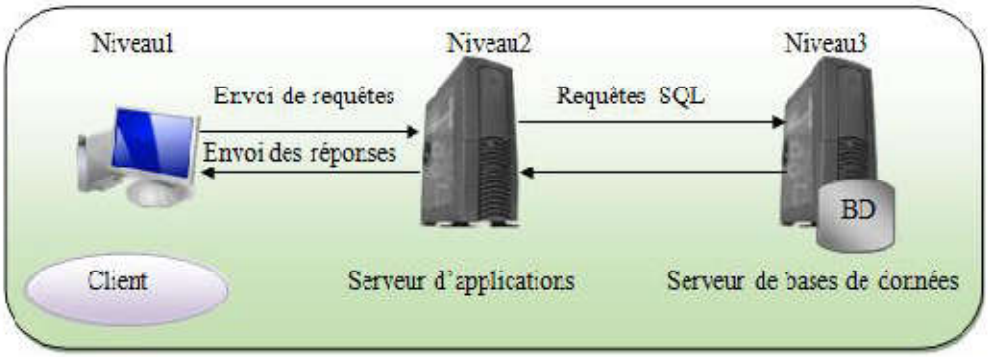

Figure 3: Architecture à 3 niveaux [13].

❖ **Architecture à n niveaux**

L'architecture à N niveaux est conçue pour pallier aux limites des architectures à trois niveaux et concevoir des applications puissantes et simples à maintenir.

Ce type d'architecture permet de distribuer plus librement la logique applicative, ce qui facilite la répartition de la charge entre tous les niveaux. Ces composants rendent un service si possible générique et clairement identifié. Ils sont capables de communiquer entre eux et peuvent donc coopérer en étant implantés sur des machines distinctes [12].

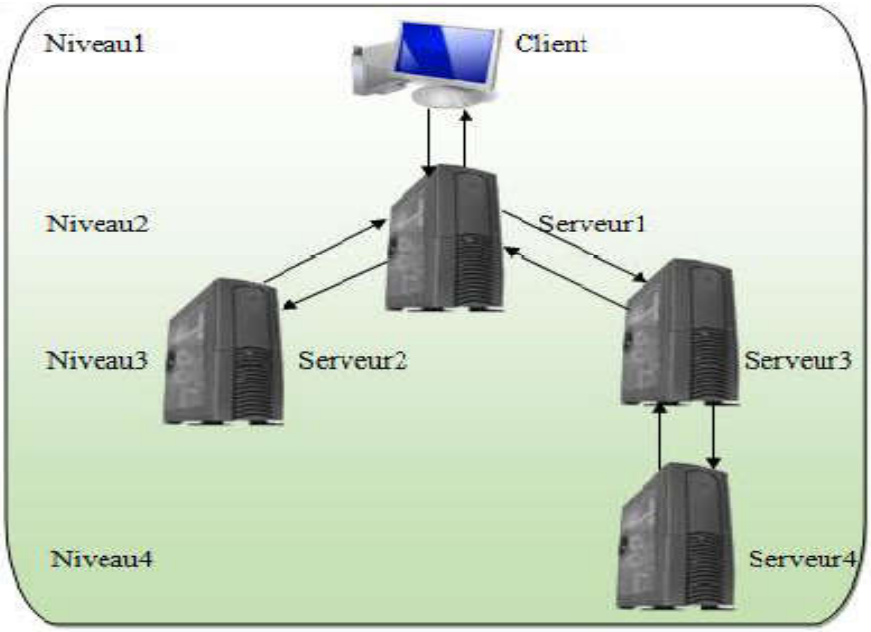

Figure 4: Architecture à n niveaux [13].

2.4.5. Les avantages et les inconvénients de l'architecture Client/Serveur

❖ **Avantages**

- Toutes les données sont centralisées sur un seul serveur, ce qui simplifie les contrôles de sécurité, l'administration, la mise à jour des données et des logiciels.
- La complexité du traitement et la puissance de calculs sont à la charge du ou des serveurs, les utilisateurs utilisant simplement un client léger sur un ordinateur terminal qui peut être simplifié au maximum.
- Recherche d'information : les serveurs étant centralisés, cette architecture est particulièrement adaptée et véloce pour retrouver et comparer de vastes quantités d'informations (moteur de recherche sur le Web).

❖ **Inconvénients**

- Si trop de clients veulent communiquer avec le serveur au même moment, ce dernier risque de ne pas supporter la charge.
- Si le serveur n'est plus disponible, plus aucun des clients ne fonctionne.
- Les coûts de mise en place et de maintenance peuvent être élevés.
- En aucun cas, les clients ne peuvent communiquer entre eux, entrainant une asymétrie de l'information au profit des serveurs [17].

3. MESSAGERIE ÉLECTRONIQUE

3.1. HISTORIQUE

Avant de voir en détail les concepts de la messagerie électronique, il est intéressant de faire un court historique de l'évolution de ce service. La messagerie électronique est un service qui est étroitement lié à l'histoire de l'Internet. Ce service a contribué, avec l'émergence de la technologie Web, au formidable développement et au succès du réseau Internet.

Dès l'apparition des premiers programmes informatiques, les ingénieurs ont voulu développer des solutions permettant l'échange d'informations entre ordinateurs. Dans les années 60, les premiers systèmes de messagerie ont vu le jour. Le Time-Shared Operating System intégrait un système de remise de message. En 1971, D. Watson publie un premier document de description de la messagerie électronique, appelée "Mail Box Protocol". L'idée était de fournir un mécanisme permettant au NIC (Network Information Center) de distribuer des documents aux différents sites du réseau ARPANET (Advanced Research Projects Agency Network). En 1972, le premier protocole de transfert de message est publié et les premiers outils de création et de gestion de messages sont développés. Ces premiers travaux sont les prémices d'une évolution qui ne s'est pas arrêtée lors des quarante années qui ont suivies pour arriver aux systèmes de messagerie que nous connaissons aujourd'hui [18] [1].

3.2. SERVICE DE MESSAGERIE

3.2.1. Définition

Un service de messagerie, dans sa forme la plus basique, est un service permettant essentiellement l'échange de messages textuels entre les différents utilisateurs enregistrés (ayant une adresse électronique valide) et connectés à un réseau informatique, que ce soit en local ou sur internet.

Mais, actuellement, les services de messagerie sont beaucoup plus riches et présentent beaucoup plus de fonctionnalités; à savoir l'intégration des pièces jointes (joindre un fichier quelconque au message envoyé), la gestion du courrier indésirable (les spams) et la manipulation des listes de diffusion (permettre l'envoi multiple).

Cet échange de messages peut s'effectuer en différé, c'est-à-dire il n'est pas nécessaire que le destinataire soit connecté au moment de l'envoi, son message sera enregistré sur un serveur et il pourra le consulter ultérieurement. On parle à ce moment de la messagerie électronique simple (courrier électronique (en anglais email ou e-mail)) [19].

La messagerie électronique et la messagerie instantanée sont deux moyens différents d'envoi de messages sur Internet. La messagerie instantanée permet de dialoguer instantanément par ordinateur avec un interlocuteur distant connecté au même réseau informatique, notamment Internet. La messagerie instantanée constitue un complément idéal au courrier électronique [20].

Nous nous intéressons, dans ce mémoire, qu'à la messagerie électronique simple (le courrier électronique).

3.2.2. Fonctions d'un service de messagerie

Les fonctions principales d'une messagerie sont bien sûr d'émettre et de recevoir des courriers et pour cela on doit définir au départ un système de désignation des entités qui s'échangent du courrier: permettant de définir le destinataire ou de savoir qui est l'émetteur. En général, les systèmes de messageries prévoient aussi les listes de diffusion c'est à dire que le système de désignation doit prévoir les groupes de destinataires.

Pour faciliter le travail des utilisateurs à l'émission, les logiciels de messagerie proposent des outils de composition des textes à transmettre.

On doit bien sur fournir une interface pour l'émission des courriers et pour la transmission de fichiers joints si la messagerie permet l'attachement de fichiers.

Symétriquement il faut définir des fonctions de récupération du courrier et d'affichage ce qui peut être assez complexe quand les courriers comportent des attachements multimédia.

Un logiciel de messagerie offre aussi généralement des fonctions de gestion des archives de courrier c'est à dire de classement des courriers selon différents critères dans des boites à lettre différentes [19].

3.3. ARCHITECTURE ET FONCTIONNEMENT D'UN SYSTEME DE MESSAGERIE ÉLECTRONIQUE

Un système de messagerie électronique est l'ensemble des éléments contribuant à transmettre un courriel (courrier électronique) de l'émetteur au récepteur.

Les différents éléments du système de messagerie sont agencés selon une architecture logique, pour en assurer le fonctionnement. La figure 5 présente le schéma de transfert d'un courriel d'un expéditeur à un destinataire [21].

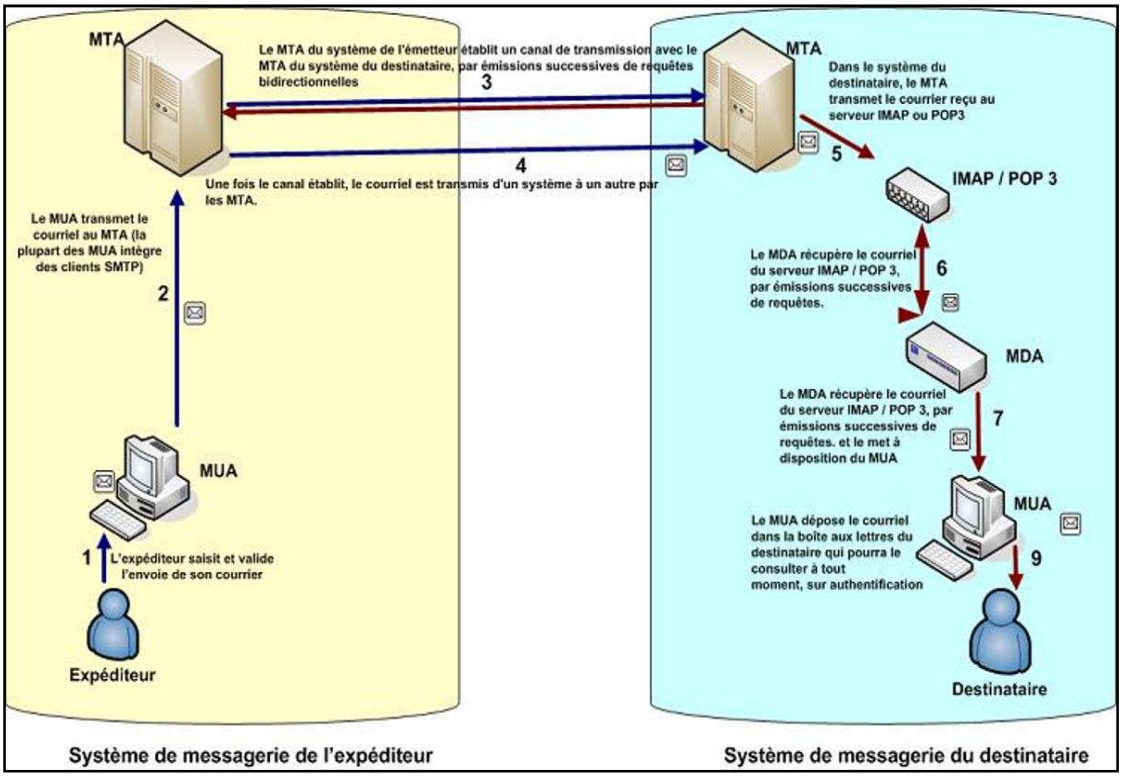

Figure 5: Acheminement d'un message électronique [21].

3.3.1. Les différents agents d'un serveur de messagerie

3.3.1.1. Le Mail User Agent (MUA)

Le MUA est un programme qui sert à lire, écrire, répondre et recevoir des messages. Il présente généralement une interface graphique riche à la disposition de l'utilisateur. Le MUA ne permet pas de transférer le message au destinataire, pour cela il fait appel à un MTA (Mail Transfert Agent) [19].

3.3.1.2. Le Mail Transfert Agent (MTA)

Le MTA est un programme dédié à la transmission des messages entre utilisateurs que ce soit en local ou sur des machines distantes. En général on a un seul MTA par machine. Un MTA reçoit, habituellement par le protocole SMTP (Simple Mail Transfer Protocol), les emails envoyés soit par des clients de messagerie électronique (MUA), soit par d'autres MTA. Son rôle est de redistribuer ces courriers à des Mail Delivery Agent (MDA) et/ou d'autres MTA.

Parmi les principaux MTA on trouve : Qmail, Postfix, Sun ONE Messaging Server, Send-mail et Sendmail Switch (version commerciale de Sendmail) [19].

3.3.1.3. Le Mail Delivery Agent (MDA)

Un MDA marque la fin de l'acheminement du mail vers la destination. C'est le MDA qui reçoit le message du MTA et se charge de le placer dans la boite aux lettres de l'utilisateur. Il doit donc gérer les éventuels problèmes tels qu'un disque plein ou une boite aux lettres corrompue et doit impérativement signaler au MTA toute erreur de délivrance.

Parmi les MDA connus nous citons : Procmail, maildrop, etc… [19].

3.3.2. Les protocoles de messagerie

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

L'acheminement de l'e-mail se fait en plusieurs étapes. Tous 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 boite aux lettres personnelles. Différents protocoles applicatifs sont utilisés au-dessus des couches réseaux et transport (TCP/IP) d'Internet : SMTP (Simple Mail Transfer Protocol), POP3 (Post Office Protocol 3), IMAP (Internet Message Access Protocol) [2].

3.3.2.1. Le protocole SMTP (Simple Mail Transfer Protocol)

Le protocole SMTP (Simple Mail Transfer Protocol) est le protocole standard permettant de transférer le courrier soit d'un client à un serveur, soit d'un serveur à autre en connexion point à point. Ce protocole fonctionne en mode commuté, encapsulé dans des trames TCP/IP, grâce à des commandes textuelles (chaine de caractères ASCII terminée par les caractères CR/LF (Carriage Return/Line Feed).

Dans un dialogue, par exemple entre un client et un serveur de messagerie, chaque commande envoyée du client est suivie d'une réponse du serveur SMTP. Une réponse est composée d'un numéro et d'un message afin de signaler si la commande est prise ou non en compte par le serveur [2].

❖ **Récapitulatif des principales commandes SMTP**

Commande	**Description**
HELO	Identification à l'aide de l'adresse IP ou du nom de domaine de l'ordinateur expéditeur.
MAIL FROM	Adresse de l'expéditeur.
RCPT TO	Adresse du destinataire.

DATA	Partie de l'en-tête et corps du mail.
HELP	Permet de récupérer la liste des commandes supportées par le Serveur SMTP.
EXPN	La fonction d'expansion renvoi les adresses de la liste de diffusion passée en paramètre.
QUIT	Permet de quitter la connexion sur le serveur SMTP.

Tableau 1: les commandes SMTP.

3.3.2.2. 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 [2].

❖ **Récapitulatif des principales commandes ESMTP**

Commande	**Description**
8BITMIMEDSN	Permet au client d'envoyer des messages comportant des caractères 8 bits.
DSN	Deliverystatus notification : permet d'envoyer à l'expéditeur un accusé de délivrance au destinataire.
SIZE	Indique, avant l'envoi par le client, la taille maximale de message admissible par le serveur.

Tableau 2: les commandes ESMTP.

3.3.2.3. Le protocole POP3 (Post Office Protocol 3)

Ce protocole permet de récupérer son courrier sur un serveur de mail, et ne fonctionne qu'en mode connecté.

Tout comme dans le cas du protocole SMTP, le protocole POP3 fonctionne grâce à des commandes textuelles envoyées au serveur POP3. Chacune des commandes envoyées par le client est composée d'un mot-clé, éventuellement accompagné d'un ou plusieurs arguments. A chaque commande, le serveur POP3 envoie une réponse (soit +OK ou –ERR) [2].

❖ **Récapitulatif des principales commandes POP3**

Commande	Description
USER<identifiant>	Cette commande permet de s'identifier. Elle doit être suivie du nom de l'utilisateur, c'est-à-dire une chaine de caractères identifiant l'utilisateur sur le serveur. La commande USER doit précéder la commande PASS.
PASS<mot_de_passe>	La commande PASS, permet d'indiquer le mot de passe de l'utilisateur dont le nom à été spécifié lords d'une commande USER préalable. Combiné à l'identifiant, il permet d'authentifier l'utilisateur.
STAT	Information sur les messages contenus sur le serveur.
RET<n°msg>	Numéro du message à récupérer.
DELE <n°msg>	Numéro du message à supprimer.
LIST [<n°msg>]	Numéro du message à afficher.
NOOP	Permet de garder les connexions ouvertes en cas d'inactivité
TOP<n°msg><n>	Commande affichant n lignes du message, dont le numéro est donné en argument. En cas de réponse positive du serveur, celui-ci renvoie les en-têtes du message, puis une ligne vierge et enfin les n lignes du message.
UIDL [<n°msg>]	Demande au serveur de renvoyer une ligne contenant des informations sur le message éventuellement donné en argument. Cette ligne contient une chaine de caractère, appelée listing d'identificateur unique, permet d'identifier de façon unique le message sur le serveur, indépendamment de la session. L'argument optionnel est un numéro correspondant à un message existant sur le serveur POP, c'est-à-dire un message non effacé.
QUIT	La commande QUIT demande la sortie du serveur POP3.Elle entraine la suppression de tous les messages marqués comme effacés et renvoie l'état de cette action.

Tableau 3: les commandes POP3

3.3.2.4. Le Protocole IMAP (Internet Message Access Protocol)

Ce protocole présente des avantages par rapport à POP3 :

- Possibilité de stocker des messages sur le serveur de manière structurée.
- Gestion de plusieurs boites aux lettres.
- Permet l'accès direct à des parties de messages.
- Supporte des accès concurrents et des boites aux lettres partagées.
- Les utilisateurs peuvent accéder à leur courrier à partir de plusieurs machines distantes.

Un service IMAP est un système de fichiers, dont les répertoires sont des classeurs. Chaque classeur contient des messages éventuellement d'autres classeurs. A chaque message sont associées des informations (en plus du corps et de l'en-tête des messages) telles que :

- Un numéro unique
- Un numéro de séquence dans son classeur.
- Un série de drapeaux (message lu, réponse envoyée, message à effacer, brouillon, etc.)
- Une date de réception du message.

Pour les serveurs qui supportent la RFC 2086 (Requests for Comments) sont des documents techniques ou des notes organisationnelles dont le sujet est Internet. Ils couvrent de nombreux aspects (concepts, protocoles, procédures, opinions, ...). Une ACL (Access Control List) est associé au classeur. Cette ACL permet d'attribuer des droits à un utilisateur ou à un groupe d'utilisateurs [2].

❖ **Récapitulatif des principales commandes IMAP**

Commande	Description
LOGIN	Connexion au serveur IMAP.
LIST	Liste des dossiers.
SELECT	Sélection d'un dossier.
SEARCH	Recherche de messages dans un dossier en fonction de critères.
FETCH	Récupération d'un message.
STORE	Association de drapeau à un message.
LOGOUT	Déconnexion du serveur IMAP.

Tableau 4: les commandes IMAP.

Le protocole IMAP permet de gérer plusieurs accès simultanés. Les commandes envoyées par le client sont précédées d'un identificateur (une chaine de caractères quelconque mais unique pour une session). Les réponses du serveur sont également précédées par l'identificateur correspondant à la commande qui les a déclenchées.

3.4. STRUCTURE GÉNÉRALE ET FORMAT D'UN MESSAGE

Un e-mail est composé d'une enveloppe (aussi appelée « en-tête »), comportant les données relatives aux adresses des émetteurs et récepteurs, ainsi que le sujet du message, la date, etc. A la suite de l'en-tête s'ajoute évidement le contenu de l'e-mail (appelé « corps du message »), comprenant éventuellement des pièces jointes [2].

3.4.1. Composition de l'adresse e-mail

Quand un correspondant veut vous envoyer un courrier postal, il doit connaître votre adresse postale dont le contenu est régi par un certain formalisme. Il en va de même pour le courrier électronique : on ne peut vous envoyer un courriel que si l'on possède votre adresse électronique.

On y trouve dans l'ordre :

- Le nom de l'utilisateur, choisi librement par chaque internaute.
- Le séparateur @ (a commercial ou arobas).
- Le serveur de messagerie : c'est l'endroit où est hébergé le courrier.
- Le nom de domaine (par exemple : dz pour l'Algérie) [10].

Voici un exemple d'une adresse e-mail :

moi@monserveur.mondomaine

3.4.2. L'en-tête d'un e-mail

L'en-tête contient les informations suivantes :

- Une adresse e-mail d'auteur du message.
- L'adresse e-mail du (ou des) destinataires.
- Une adresse à utiliser en cas de réponse.
- Le chemin suivi par le message.
- Le type d'encodage du message.
- Des informations « subsidiaires » [2].

Lors de la lecture d'un e-mail, le« MUA » indique :

- L'expéditeur.
- L'objet.
- Le texte (ou corps) du message.

Mais cet e-mail contient toute une partie « cachée » qui permet de connaitre le chemin suivi par cet e-mail avant d'arriver dans la boite aux lettres de destination.

3.4.3. Le format standard de messages MIME (Multipurpose Internet Mail Extension)

Pour des raisons de compatibilité, plusieurs normes ont été élaborées afin d'uniformiser la structure des messages et de définir des formats standards. Parmi ces standards nous citons Multipurpose Internet Mail Extension ou MIME. Les principaux apports du standard MIME sont :

- Possibilité d'avoir plusieurs objets (pièces jointes) dans un même message.
- Une longueur de message illimitée.
- L'utilisation de jeux de caractères (alphabets) autres que le code ASCII.
- L'utilisation de texte enrichi (mise en forme des messages, polices de caractères, couleurs, etc.)
- Des pièces jointes binaires (exécutables, images, fichiers audio ou vidéo, etc.) comportant éventuellement plusieurs parties.

MIME utilise des directives d'entête spécifiques pour décrire le format utilisé dans le corps d'un message, afin de permettre au client de messagerie de pouvoir l'interpréter correctement :

- **MIME-Version:** Il s'agit de version du standard MIME utilisée dans le message.
- **Content-type:** Décrit le type et les sous-types des données.
- **Content-Transfer-Encoding:** Définit l'encodage utilisé dans le corps du message.
- **Content-ID:** Représente un identificateur unique de partie de message.
- **Content-Description:** Donne des informations complémentaires sur le contenu du message.
- **Content-Disposition:** Définit les paramètres de la pièce jointe, notamment le nom associé au fichier grâce à l'attribut « filename » [19].

3.4.3.1. Les principaux types de MIME

Le type MIME, utilisé dans l'entête Content-Type, est utilisé pour typer les documents attachés à un courrier. Les principaux types de données, appelés parfois " types de données discrets ", sont les suivants :

- **text:** données textuelles lisibles (text/rfc822 [RFC822] ; text/plain [RF646] ; text/html [RF854]).
- **image:** données binaires représentant des images numériques (image/jpeg, image/gif, image/png).
- **audio:** données numériques sonores (audio/basic; audio/wav).
- **vidéo:** données vidéo (video/mpeg).
- **application:** données binaires autres (application/octet-stream; application/pdf) [19].

3.4.3.2. Les types de codage

Pour transférer des données binaires, MIME propose cinq formats de codage pouvant être utilisé dans l'entête Transfer-encoding:

- **7bit:** format texte codé sur 7 bits (pour les messages non accentués).
- **8bit:** format texte 8 bits.
- **quoted-printable:** format Quoted-Printable, recommandé pour les messages utilisant un alphabet codé sur plus de 7 bits (présence d'accents par exemple).
- **base64:** format Base 64, recommandé pour l'envoi de fichiers binaires en pièce jointe.
- **binary:** format binaire [19].

3.5. LES OUTILS DE LA MESSAGERIE ÉLECTRONIQUE

3.5.1. Les serveurs de messagerie

Un serveur de messagerie électronique est un logiciel qui, connecté à Internet, permet à ses utilisateurs d'envoyer et de recevoir des courriers électroniques. Pour se connecter au serveur de messagerie, l'utilisateur a recours à un logiciel client, tel que Microsoft Outlook ou Mozilla Thunderbird, capable de gérer l'adressage (ou envoi) du courriel mais aussi sa réception.

Le logiciel client peut être également émulé en HTTP, ce qui permet l'accès au courriel depuis un simple navigateur Internet [22].

❖ **Exemples de serveurs de messagerie**

Serveur	**Sendmail**	**Postfix**	**MS (Microsoft Exchange)**
Caractéristiques	- Très puissant et résiste - beaucoup à la grande charge. - Une très bonne sécurité. - Code source libre. - Multi plate-forme de type UNIX (MAC OS, GNU/LINUX). Difficile à configurer car son architecture est vieille. - Lent. - Très complexe avec une maintenance difficile.	- Il est adapté pour des gros besoins. - Facile à installer et à configurer. - Maintenance aisée. Sécurisé avec anti Spam. - Codes sources libres. - Multi plate-forme de type UNIX (MAC OS, GNU/LINUX). - Gratuit. -Accessible en mobilité. - Pas d'inconvénients majeurs.	- Accès en mobilité. - Une bonne sécurité antispam qui sauvegarde et archive vos données critiques. - Une bonne efficacité permettant de partager votre agenda et contact professionnel avec vos collègues ou collaborateurs. - Les codes sources ne sont pas libres. Il n'est pas gratuit.

Tableau 5: Exemples de serveurs de messagerie.

3.5.2. Les clients de messagerie

Un client de messagerie est un logiciel qui sert à lire et envoyer des courriers électroniques, il existe deux types de client de messagerie :

❖ **Les clients lourds**

Un client de messagerie de type lourd, est un logiciel qui permet de lire, d'écrire et d'expédier des courriers électroniques. Il s'installe sur un poste client qui se connecte au serveur de messagerie. Le client lourd a l'avantage de récupérer nos messages et de les copier sur notre poste local, en mode connecté au serveur. Ainsi en mode hors connexion, nous avons accès à nos messages [19].

❖ **Exemples de clients de messagerie lourds**

Client	**Thurderbird de Mozilla**	**Zimbra Desktop**
Caractéristiques	- Très léger. - Multi plate-forme (Windows, Mac OS, Linux). - Rapide. - Extensible (peut recevoir de nouvelles fonctionnalités). - Les codes sources sont libres d'accès. - Installation et configuration simples. - Transfert de messages avec pièces jointes. - Il est gratuit.	- Multi plate-forme (Windows, Mac OS, Linux). - Il est gratuit. - Codes sources sont libres. - Regroupe tous les comptes dans un seul répertoire. - Installation et configuration rapide, facile. - Transfert les messages avec pièces jointes. - Extensible.

Tableau 6: Exemples de clients de messagerie lourds.

❖ **Les clients légers ou Webmails**

Un client de messagerie de type léger est une interface Web rendant possible l'émission, la consultation et la manipulation de courriers électroniques directement sur le Web depuis un navigateur web (Internet explorer, Firefox, …). Il fonctionne uniquement en mode connecté et ne copie pas en local les messages stockés sur le serveur. Ainsi, en mode hors connexion nous n'avons plus accès à nos courriers [19].

❖ **Exemples de clients de messagerie légers**

Client	**MS Outlook Web Access**	**Web mail Ajax de Zimbra**
Caractéristiques	- Installation rapide et facile. - Très léger. - Codes sources ne sont pas libres. - Il n'est pas gratuit. - Uni plate-forme (MS Windows).	- Installation automatique à la première connexion au serveur via un navigateur web. - Multi plate-forme (MS Windows, Mac OS, Linux, etc.). - Gratuit. - Codes sources libres - Installation et configuration faciles.

Tableau 7: Exemples de clients de messagerie légers.

4. CONCLUSION

Dans ce chapitre, nous avons présenté les différents concepts liées à la messagerie électronique, tout en présentant ses services et les différents outils nécessaires pour sa mise en œuvre.

Dans le chapitre suivant, nous aborderons la sécurité informatique, en particulier la cryptographie, ainsi que son utilisation pour sécuriser la messagerie électronique.

CHAPITRE 02 :
SÉCURITÉ DU COURRIER ÉLECTRONIQUE

1. INTRODUCTION

En entreprise, la messagerie électronique est de plus en plus utilisée et est devenue une application Internet indispensable. Parce que la messagerie électronique est un vecteur de communication, de productivité et de production, sa sécurité et sa disponibilité sont des préoccupations légitimes.

Il faut savoir que lorsqu'un message électronique est envoyé, celui-ci transite par plusieurs ordinateurs (le message est enregistré sur le disque dur de ce qu'on appelle des serveurs relais) avant d'arriver sur l'ordinateur du destinataire. Ce voyage n'est pas sans péril. En effet, ces ordinateurs appartiennent à des entreprises, des administrations ou des personnes qui peuvent ne pas s'embarrasser de scrupules et lire ce courriel. Celui-ci, de par sa conception même et les protocoles qu'il utilise, circule en clair, c'est-à-dire non sécurisé, « comme une carte postale sans enveloppe ». Il est donc important de cacher son contenu, pour protéger d'éventuels secrets professionnels ou tout simplement son intimité.

Dans ce chapitre nous allons tout d'abord définir les concepts de base de la sécurité informatique. Nous présenterons, ensuite les mécanismes cryptographiques utilisés pour sécuriser les communications réseaux. Enfin, nous présenterons la sécurité du courrier électronique, les attaques sur la messagerie électronique, ainsi que les applications existantes de la cryptographie utilisées pour la protection contre ces attaques.

2. GÉNÉRALITÉS SUR LA SÉCURITÉ INFORMATIQUE

2.1. QUELQUES DÉFINITIONS

2.1.1. Sécurité informatique

La sécurité informatique est l'ensemble des moyens mis en œuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. Un ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires sont mis en place pour conserver, rétablir, et garantir la sécurité de l'information et du système d'information [23].

2.1.2. Menace

Violation potentielle d'une propriété de sécurité [24]. On peut classer les menaces en deux catégories selon qu'elles ne changent rien (menaces passives) ou qu'elles perturbent effectivement le réseau (menaces actives).

- **Les menaces passives :** consistent essentiellement à copier ou à écouter l'information sur le réseau, elles nuisent à la confidentialité des données. Dans ce cas, celui qui prélève une copie n'altère pas l'information elle-même.
- **Les menaces actives :** nuisent à l'intégrité des données. Elles se traduisent par différents types d'attaques. On distingue le brouillage, le déguisement (modification des données au cours de leur transmission, modification de l'identité de l'émetteur ou du destinataire), l'interposition (création malveillante de messages en émission ou en réception) [25].

2.1.3. Attaque

Lorsqu'une menace délibérée est mise à exécution, on parle alors d'attaque. Les attaques peuvent être réalisées grâce à des moyens techniques ou par ingénierie sociale (employer des méthodes non techniques pour obtenir un accès non autorisé) [26].

2.1.4. Vulnabilité

La vulnérabilité (en anglais « vulnerability », appelée parfois faille ou brèche) est une faute de conception ou de configuration du système informatique, intentionnelle ou accidentelle, qui favorise la réalisation d'une menace ou la réussite d'une attaque [27].

2.1.5. Contre-mesure

Les contre-mesures sont les procédures ou techniques permettant de résoudre une vulnérabilité ou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaques sur la même vulnérabilité) [28].

2.1.6. Risque

Exposition accidentelle et imprévisible de l'information ou violation de l'intégrité des opérations due au mal fonctionnement du hardware ou à un software incorrect ou incomplet [29].

Le risque en termes de sécurité est généralement caractérisé par l'équation suivante:

Risque= Menace* Vulnabilité /Contre mesure [27].

2.2. PROPRIÉTÉS DE LA SÉCURITÉ INFORMATIQUE

Les solutions de sécurité doivent contribuer à satisfaire au moins les critères suivants :

- **La confidentialité**: ce critère de sécurité garantit que seules les parties autorisées peuvent accéder aux données transmises à travers le réseau. Ces données peuvent concerner la couche applicative ou les couches inférieures [30].
- **L'authentification** : consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être [27].
- **L'intégrité** : permet de s'assurer que les données échangées ne sont pas soumises à une altération volontaire ou accidentelle. Donc, elle permet aux destinataires de détecter les manipulations de données effectuées par les entités non autorisées et rejeter les paquets correspondants.
- **La disponibilité :** ce critère de sécurité vise à garantir aux entités autorisées d'accéder aux ressources du réseau avec une qualité de service adéquate.
- **La non-répudiation :** permet de s'assurer qu'aucun émetteur ne peut nier d'être à l'origine d'un message. Cet objectif est indispensable dans les transactions électroniques et dans toutes les communications sensibles [31].

2.3. MISE EN PLACE D'UNE POLITIQUE DE SÉCURITÉ

La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leur ont été octroyés.

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

C'est la raison pour laquelle il est nécessaire de définir dans un premier temps une politique de sécurité, dont la mise en œuvre se fait selon les quatre étapes suivantes :

- Identifier les besoins en termes de sécurité, les risques informatiques et leurs éventuelles conséquences.
- Elaborer des règles et des procédures à mettre en œuvre pour les risques identifiés.
- Surveiller et détecter les vulnérabilités du système d'information et se tenir informé des failles sur les applications et matériels utilisés.
- Définir les actions à entreprendre en cas de détection d'une menace.

Une politique de sécurité informatique est donc un plan d'actions définies pour maintenir un certain niveau de sécurité. Elle reflète la vision stratégique de la direction de l'organisme (industrie, administration, état, unions d'états, etc.) en matière de sécurité informatique [23] [32].

2.4. DOMAINES D'APPLICATION DE LA SÉCURITÉ INFORMATIQUE

En fonction de son domaine d'application la sécurité informatique se décline en :

2.4.1. Sécurité physique

Elle Concerne tous les aspects liés à la maitrise des systèmes et de l'environnement dans lequel ils se situent. Elle repose essentiellement sur :

- Les normes de sécurité.
- La protection de l'environnement (incendie, température, humidité, etc.).
- La protection des sources énergétiques (alimentation).
- La protection des accès (protection physique des équipements, infrastructure câblée, redondance des alimentations énergétiques, etc.).
- La sureté de fonctionnement et la fiabilité des matériels (composants, câbles, etc.).
- La redondance physique.
- Plan de maintenance préventive (test) et corrective (pièce de rechange).

2.4.2. Sécurité de l'exploitation

On entend par sécurité de l'exploitation tout ce qui touche au bon fonctionnement des systèmes. Cela comprend la mise en place d'outils et de procédures relatives aux méthodologies d'exploitation, de maintenance, de test, de diagnostic et de mise à jour.

Quelques points clés de cette sécurité sont les suivants :

- Plan de sauvegarde, de secours, de continuité, et de tests.
- Inventaires réguliers et si possible dynamiques.

- Automatisation, contrôle et suive de l'exploitation.
- Gestion des contrats de maintenance.

2.4.3. Sécurité logique

La sécurité logique fait référence à la réalisation de mécanismes de sécurité par logiciel contribuant au bon fonctionnement des applications et services. Bien qu'elle s'appuie principalement sur une mise en œuvre adéquate d'un processus de contrôle d'accès logique. La sécurité logique repose également sur :

- Les dispositifs mis en place pour garantir la confidentialité dont la cryptographie.
- Une gestion efficace des mots de passe et des procédures d'authentification.
- Des mesures antivirus et de sauvegarde des informations sensibles.

2.4.4. Sécurité applicative

L'objectif est d'éviter les «bugs» dans les applications. Cette sécurité repose essentiellement sur :

- Une méthodologie de développement.
- La robustesse des applications.
- Des contrôles programmés.
- Des jeux de tests.
- Un plan de migration des applications critiques.
- La validation et l'audit des programmes.
- Un plan d'assurance sécurité.

2.4.5. Sécurité des télécommunications

La sécurité des télécommunications consiste à offrir à l'utilisateur final, c'est-à-dire aux applications communicantes, une connectivité fiable et de qualité de « bout en bout ». Pour cela, un « canal de communication » sûr entre les correspondants, quels que soient le nombre et la nature des éléments intermédiaires (système ou réseaux) nécessaires au transport des données, doit pouvoir être offert.

Ceci implique la réalisation d'une infrastructure réseau sécurisée au niveau des accès, de l'acheminement, des protocoles de communication, des systèmes d'exploitation et des équipements de télécommunication et des supports de transmission. La sécurité des réseaux et des ressources est également atteinte par un cloisonnement judicieux d'environnements informatiques, par des mesures architecturales adaptées [30] [33].

2.5. ATTAQUES

2.5.1. Attaques exploitant les protocoles de réseaux

Il existe différents types d'attaque exploitant les protocoles de réseaux, par exemple:

- **Attaque par fragmentation** : Cette attaque outrepasse la protection des équipements de filtrage IP.
- **IP Spoofing** : Le but de cette attaque est l'usurpation de l'adresse IP d'une machine. Ceci permet au pirate de cacher la source de son attaque.
- **TCP Session Hijacking**: permet de rediriger un flux TCP. Un pirate peut alors outrepasser une protection par un mot de passe (comme Telnet ou FTP).
- **Déni de service** : Cette attaque porte bien son nom puisqu'elle aboutira à l'indisponibilité du service (application spécifique) ou de la machine visée. Il existe différents types de dénis de service comme SYN Flooding, UDP Flooding, Packet Fragment, Smurfing et Déni de service distribué.

2.5.2. Attaques exploitant les programmes

Il existe différents types d'attaque exploitant les programmes, par exemple :

- **Injection SQL** : est un type d'exploitation d'une faille de sécurité d'une application interagissant avec une base de données, en injectant une requête SQL non prévue par le système et pouvant compromettre sa sécurité.

2.5.3. Attaques exploitant l'e-mail

Il existe différents types d'attaque exploitant l'e-mail : Spam, Phishing, usurpation d'identité (email spoofing), etc. Nous décrirons ces attaques dans la partie « Sécurité du courrier électronique ».

2.5.4. Attaques par codes malicieux

Ce type d'attaque est dirigé contre l'intégrité de l'information, nous citons:

- **Virus :** est un petit programme informatique situé dans le corps d'un autre, qui, lorsqu'on l'exécute, se charge en mémoire et exécute les instructions que son auteur a programmé.
- **Vers :** sont des virus capables de se propager à travers un réseau.
- **Chevaux de Troie :** sont des virus permettant de créer une faille dans un système (généralement pour permettre à son concepteur de s'introduire dans le système infecté afin d'en prendre le contrôle).
- **Bombes logiques :** sont des virus capables de se déclencher suite à un événement particulier (date système, activation distante, ...) [25] [34].

2.6. MÉCANISMES DE SÉCURITÉ

Un mécanisme de sécurité est conçu pour détecter, prévenir et lutter contre une attaque de sécurité pour assurer les propriétés de sécurité.

2.6.1. Antivirus

Un antivirus est un programme capable de détecter la présence de malware sur un ordinateur et, dans la mesure du possible, de désinfecter ce dernier. On parle ainsi d'éradication de virus pour désigner la procédure de nettoyage de l'ordinateur.

Il existe plusieurs méthodes d'éradication :

- ✓ La suppression du code correspondant au virus dans le fichier infecté ;
- ✓ La suppression du fichier infecté ;
- ✓ La mise en quarantaine du fichier infecté, consistant à le déplacer dans un emplacement où il ne pourra pas être exécuté.

2.6.2. Pare-feu

Un pare-feu (appelé aussi coupe-feu, garde-barrière ou firewall), est un système permettant de protéger un ordinateur, ou un réseau d'ordinateurs, des intrusions provenant d'un réseau tiers (notamment Internet). Le pare-feu est un système permettant de filtrer les paquets de données échangés avec le réseau, il s'agit ainsi d'une passerelle filtrante comportant au minimum les interfaces réseau suivantes :

- Une interface pour le réseau à protéger (réseau interne).
- Une interface pour le réseau externe.

Le système pare-feu est un système logiciel, reposant parfois sur un matériel réseau dédié, constituant un intermédiaire entre le réseau local (ou la machine locale) et un ou plusieurs réseaux externes.

2.6.3. Systèmes de détection d'intrusion

On appelle IDS (Intrusion Detection System) un mécanisme écoutant le trafic réseau de manière furtive afin de repérer des activités anormales ou suspectes et permettant ainsi d'avoir une action de prévention sur les risques d'intrusion.

Il existe deux grandes familles distinctes d'IDS :

- ❖ Les N-IDS (Network Based Intrusion Detection System) : ils assurent la sécurité au niveau du réseau.
- ❖ Les H-IDS (Host Based Intrusion Detection System) : ils assurent la sécurité au niveau des hôtes [35].

2.6.4. Cryptographie

La cryptographie est à la base de la plupart des mécanismes de sécurité. Elle a pour but de garantir la protection des communications transmises sur un canal public contre différents types d'adversaires. La protection des informations se définit en termes de confidentialité, d'intégrité et d'authentification. Nous décrirons ce mécanisme dans la section suivante [36].

3. CRYPTOGRAPHIE

3.1. TERMINOLOGIE

Comme toute science, la cryptographie possède son propre langage. Dans ce qui suit, on présente les mots clés du domaine cryptographique.

- **Texte en clair :** c'est le message à protéger.
- **Texte chiffré (Cryptogramme) :** c'est le résultat du chiffrement du texte en clair.
- **Chiffrement :** c'est la méthode ou l'algorithme utilisé pour transformer un texte en clair en texte chiffré.
- **Déchiffrement :** c'est la méthode ou l'algorithme utilisé pour transformer un texte chiffré en texte en clair.
- **Clé :** c'est le secret partagé utilisé pour chiffrer le texte en clair en texte chiffré et pour déchiffrer le texte chiffré en texte en clair. On peut parfaitement concevoir un algorithme qui n'utilise pas de clé, dans ce cas c'est l'algorithme lui-même qui constitue la clé, et son principe ne doit donc en aucun cas être dévoilé.
- **Cryptologie :** Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse [37].
- **Cryptographie :** Le mot cryptographie est composé de deux mots grecs : « crypto » qui signifie caché et « graphie » qui signifie écrire, d'où la signification complète de la cryptographie est « l'écriture secrète ». La cryptographie est définie comme la science qui convertit les informations en clair en informations cryptées c'est-à-dire codées. Le principe est présenté dans la figure 3. La plupart des mécanismes de sécurité des communications sont basés sur des outils cryptographiques utilisant des informations secrètes représentées par des nombres premiers, dites clés, combinées, en entrée d'une opération cryptographique, au message à coder pour produire le message crypté.

Pour assurer un système cryptographique fiable, deux communicants doivent choisir soigneusement leur clé de chiffrement/déchiffrement et doivent appliquer les principes suivants, car un attaquant peut essayer un certain nombre de possibilités et par chance tomber rapidement sur la solution :

- ✓ La sécurité doit se reposer sur le secret de la clé et non pas sur la sécurité de l'algorithme.
- ✓ Le déchiffrement sans connaissance préalable de la clé doit être impossible en un temps raisonnable.
- ✓ Calculer la clé à partir du texte en clair et du texte chiffré doit être impossible en un temps raisonnable [38].

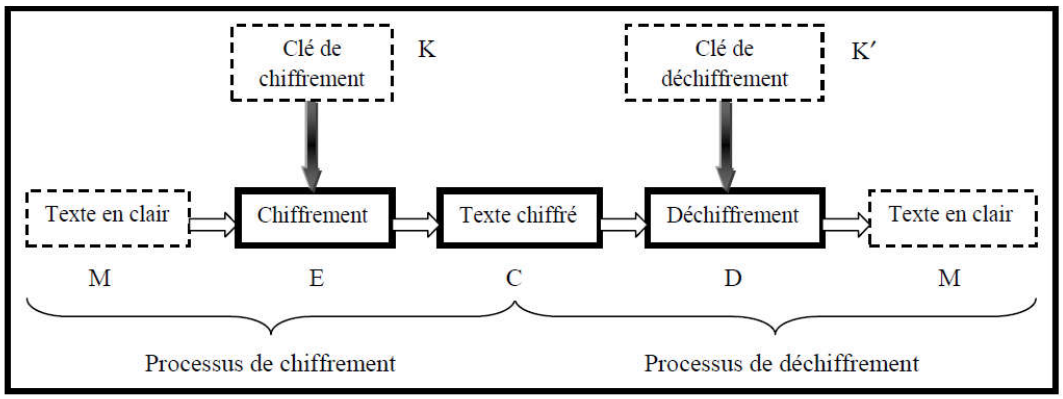

Figure 6: Principe de la cryptographie [37].

- **Cryptosystème :** un terme utilisé en cryptographie pour désigner un ensemble composé d'algorithmes cryptographiques et de tous les textes en clairs, textes chiffrés et clés possibles nécessaires à son fonctionnement [39].
- **Cryptanalyse :** C'est l'art d'étude des cryptosystèmes en cherchant leurs failles et leurs vulnérabilités afin de retrouver des messages clairs correspondant à des messages chiffrés sans avoir à connaître les clés utilisées dans le chiffrement. Lorsque tous les éléments de la méthode utilisée pour coder des messages sont repérés, on dit qu'on a cassé ou brisé le système cryptographique utilisé. Plus un système est difficile à briser, plus il est sûr. La personne qui pratique la cryptanalyse est appelée : cryptanalyste. Il tente à décrypter le message chiffré pour découvrir son secret [37].

3.2. CRYPTOGRAPHIE CLASSIQUE ET CRYPTOGRAPHIE MODERNE

Le but de la cryptographie classique est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle par chiffrement. La plupart des méthodes de chiffrement classique reposent sur deux principes essentiels : la substitution (remplacer certaines lettres par d'autres ou par des symboles) et la transposition (permuter les lettres du message afin de le rendre inintelligible).

Le but de la cryptographie moderne est de traiter plus généralement des problèmes de sécurité des communications et de fournir un certain nombre de services de sécurité : Confidentialité, Authentification de l'origine des données, Intégrité et la Non-répudiation pour cela on utilise un certain nombre de mécanismes basé sur des algorithmes cryptographiques. La sécurité des données chiffrées dépond des éléments suivants :

- ✓ La robustesse de l'algorithme cryptographique (difficile à casser).
- ✓ La confidentialité de la clé [39] [40].

3.3. LES OUTILS DE LA CRYPTOGRAPHIE MODERNE

Les outils cryptographiques utilisant le principe de clé pour sécuriser les liens de communication sont nombreux, on cite :

3.3.1. Le chiffrement

Le chiffrement est un système basé sur des clés cryptographiques pour assurer la confidentialité des données. Les clés peuvent être privées (symétriques) ou publiques (asymétriques).

3.3.1.1. Le chiffrement symétrique (à clé secrète)

Une même clé est utilisée entre l'émetteur et destinataire pour chiffrer et déchiffrer les messages échangés entre eux en utilisant un algorithme de chiffrement symétrique. Le chiffrement symétrique se fait soit en fractionnant les données en bit à bit (chiffrement par flots) ou bien en blocs de taille fixe (chiffrement par blocs).

- ✓ **Les algorithmes de chiffrement par blocs :** Ce sont les algorithmes les plus utilisés dans les systèmes cryptographiques. Les algorithmes de chiffrement par blocs (Block Cipher en anglais) décomposent les données à transmettre en blocs de tailles généralement égales (comprises entre 64 et 512 bits) et sont ensuite chiffrés les uns après les autres selon différents modes (ECB (Electronique code book), CBC (Cipher Block Chaining), OFB (Output Feedback), etc.). Outre la confidentialité, les algorithmes de chiffrement par blocs peuvent être utilisés dans d'autres outils de cryptographie comme les fonctions de hachage et les codes MAC (Message Authentication Code).
- ✓ **Les algorithmes de chiffrement par flots** : ils convertissent les données en clair en texte chiffré bit par bit en combinant les flux de bits en clair avec un flux de bits aléatoires. L'opération de combinaison est souvent bijective (un ou exclusif (XOR)). Le déchiffrement est réalisé par l'inversement de l'opération bijective en combinant les données chiffrées avec le même flux aléatoire de bits pour retrouver les données en clair. Le flux de bits aléatoire est généré avec un générateur pseudo-aléatoire initialisé avec la clé de chiffrement et un vecteur public d'initialisation différent pour chacun des messages à chiffrer [41].

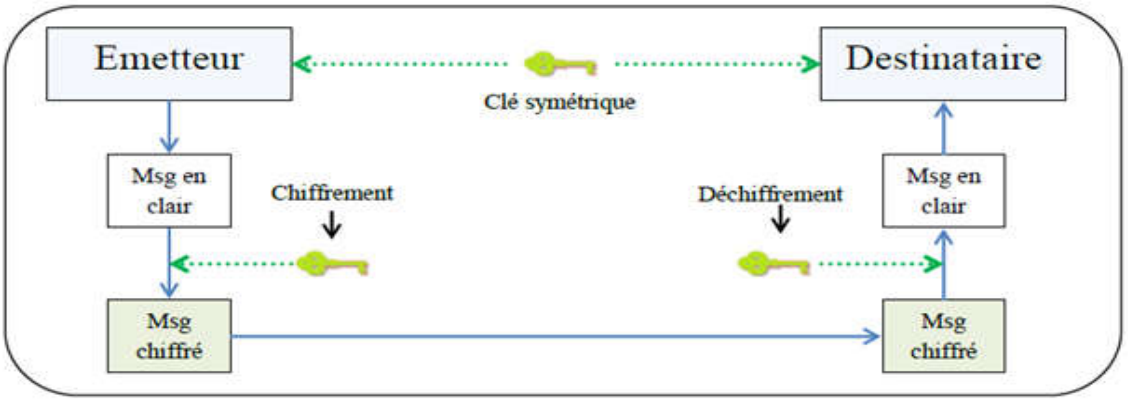

Figure 7: Chiffrement symétrique [38].

❖ **Exemples d'algorithmes cryptographiques à clef secrète**

Plusieurs systèmes à clé secrète ont été proposés. Les plus connus sont les suivants:

- **RC4 (Rivest Cipher 4)**

un algorithme de chiffrement par flux à clé de longueur variable développé en 1987 par Ron Rivest. Il est resté secret longtemps avant d'être publié, et est maintenant beaucoup utilisé, en particulier dans le protocole SSL.

- **DES (Data Encryption Standard)**

Créé dans les années 70 (public en 1981), l'algorithme DES a été l'algorithme de cryptographie le plus usité jusqu'à ces dernières années. le DES est un algorithme de chiffrement par blocs qui agit sur des blocs de 64 bits. La longueur de la clef est de 56 bits. Généralement, celle-ci est représentée sous la forme d'un nombre de 64 bits, mais un bit par octet est utilisé pour le contrôle de parité. Les sous-clefs utilisées par chaque ronde ont une longueur de 48 bits.

- **AES (Advanced Encryption Standard)**

En octobre 2000, un nouveau standard de chiffrement à clef secrète fut élu parmi 15 candidats par le NIST (National Institute of Standards and Technology) afin de remplacer le vieillissant DES dont la taille des clefs devenait trop petite. L'algorithme choisi pour devenir l'AES est le Rijndael, du nom condensé de ses concepteurs, Rijmen et Daemen. Celui-ci est un système de chiffrement par blocs. Les messages sont chiffrés par blocs de 128 bits [41] [42].

❖ **Avantages de la cryptographie symétrique**

- ✓ Calcul non couteux et rapide des clés.
- ✓ Les clés sont relativement courtes (en moyenne 128 bits).
- ✓ Adaptée au cryptage de flux de données.

❖ **Inconvénients de la cryptographie symétrique**

- ✓ La distribution de clés entre les entités communicantes est très problématique.
- ✓ Le nombre de clés à gérer accroit sensiblement avec le nombre de nœuds déployés.
- ✓ Certaines propriétés sont difficiles à réaliser (exemple : la signature) [37] [41].

3.3.1.2. Le chiffrement asymétrique

Deux clés sont générées par le récepteur du message : une clé privée maintenue chez le récepteur et une clé publique diffusée à tous les nœuds émetteurs. La clé publique sert au chiffrement d'un message et la clé privée sert au déchiffrement du même message [37].

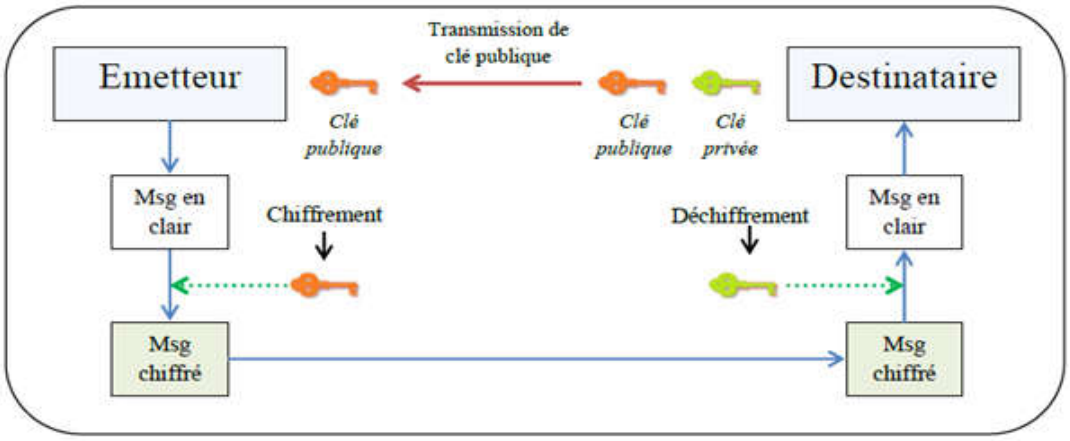

Figure 8: Chiffrement asymétrique [38].

❖ **Exemples d'algorithmes cryptographiques à clef publique**

Plusieurs systèmes à clé publique ont été proposés. Les plus connus sont les suivants :

- **RSA (Rivest-Shamir-Adleman)**

La technique de chiffrement asymétrique RSA est inventée par Rivest, Shamir et Adleman en 1978 et il repose sur la difficulté de factoriser des grands nombres. Voici comment se fait la génération des paires de clefs :

- On commence par choisir deux grands nombres premiers, p et q, et on calcule $n = pq$. n est rendu public, p et q doivent rester secrets et sont donc détruits une fois les clefs générées.
- On choisit ensuite aléatoirement une clef publique e telle que e et $(p-1)(q-1)$ soient premiers entre eux.
- La clef privée d est obtenue grâce à: $ed = 1 \bmod (p-1)(q-1)$.

Soit m le message en clair et c le cryptogramme. La fonction de chiffrement est, de façon simplifiée, $c = m^e \bmod n$ .Du fait de la relation entre e et d, la fonction de déchiffrement correspondante est $m = c^d \bmod n$. La signature se fait de manière similaire, en inversant e et d, c'est-à-dire en chiffrant avec une clef privée et en déchiffrant avec la clef publique correspondante : $s = m^d \bmod n$ et $m = s^e \bmod n$ [41].

- **ElGamal**

En 1985, ElGamal invente une technique de chiffrement asymétrique probabiliste c.à.d. un même message M peut avoir plusieurs chiffres différents. Il est basé sur la difficulté du problème des logarithmes discrets.

D'une manière formelle, l'algorithme ElGamal peut être résumé comme suit : Soit p un nombre premier tel que le problème du logarithme discret dans Zp soit difficile, et soit $\alpha \in Zp^*$ un élément primitif. Soit P = Zp^*, C = $Zp^* \times Zp^*$et K = {(p, α, a, ß) : ß ≡ α^a(mod p)}. Les valeurs p, α et ß sont publiques, et a est secret.

Pour k = (p, α, a, ß), et pour un $k \in z_{p-1}$ aléatoire (secret), la fonction de chiffrement est définit par :

$$e_k(x,k) = (y1, y2)$$

Où

$$y1 = \alpha^k \bmod p$$

Et

$$y2 = x\ \beta^k \bmod p$$

Pour y1, y2 ∈ Zp^*, la fonction de déchiffrement est définit comme suit :

$$d_k\ (y1, y2) = y2(y1^a)^{-1} \bmod p$$

Informellement, le fonctionnement du chiffrement ElGamal peut être décrit par la suite des points suivants :

- ✓ Le texte clair est masqué par la multiplication par β^k, en produisant y2.
- ✓ La valeur α^k est également transmise en tant que partie du texte chiffré.
- ✓ Bob, qui connaît l'exposant secret a, peut calculer β^k à partir de α^k . Il peut alors « enlever le masque » en divisant y2 par β^k et obtenir le texte clair x [37].

❖ **Avantages de la cryptographie asymétrique**

- ✓ Très sécurisée à cause de l'utilisation de deux clés distinctes, l'une ne permettant pas de retrouver l'autre.
- ✓ Facilité de la distribution des clés.
- ✓ Permet la signature électronique.
- ✓ Nombre réduit de clés à distribuer.

❖ **Inconvénients de la cryptographie asymétrique**

- ✓ Taille des clés, plus grandes que celles des systèmes symétriques.
- ✓ Le temps d'exécution : plus lent que le cryptage symétrique [37] [41].

3.3.1.3. Le chiffrement hypride

La cryptographie asymétrique est beaucoup plus lente que la cryptographie symétrique qui brille par sa rapidité. En revanche, cette dernière souffre d'une grave lacune , assurer une transmission secrète de la clé. Pour palier ce défaut et cumuler les avantages des deux méthodes, la combinaison des deux techniques s'avère intéressante et elle a tendance à devenir un système cryptographique moderne.

On code tout d'abord les données avec une clé privée dite clé de session, ensuite cette clé est cryptée à l'aide d'une clé publique classique. Comme la clé est courte, on utilise l'algorithme asymétrique puisqu'il prend peu de temps. En revanche, chiffrer l'ensemble du message avec un algorithme asymétrique serait plus lourd. Il suffit ensuite d'envoyer le message chiffré avec une clé privée et accompagné de cette dernière chiffrée avec une clé publique.

Le destinataire procède inversement, il commence à déchiffrer la clé symétrique avec sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, via un déchiffrement symétrique pour retrouver le message original. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques [37] [43].

3.3.2. Fonction de hachage

Une fonction de hachage est une fonction mathématique qui assure l'intégrité des informations qui circulent sur le réseau. La fonction de hachage sert à calculer une courte empreinte de taille fixe à partir d'une information de taille arbitraire pour remplir les trois conditions suivantes :

- Calculer l'empreinte facilement à partir du contenu du message.
- La difficulté de trouver le contenu du message à partir de l'empreinte (la fonction est à sens unique, c'est-à-dire à partir de H(M), il est impossible de trouver M).
- La difficulté de trouver une même empreinte pour deux messages aléatoires différents (c'est-à-dire à partir de H(M) et M, il est impossible de trouver $M' \neq M$ tel que $H(M') = H(M)$) et cela mène à la résistance aux collisions.

Exemples de fonctions de hachage : MD5 (Message Digest5), SHA-1 (Secure Hash Algorithm) [38].

3.3.3. Signature numérique

Le principe de la signature numérique consiste à appliquer une fonction de hachage sur une portion du message et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message. Il faut noter que la fonction est choisie de telle manière qu'il soit impossible de changer le contenu du message sans altérer le code de hachage. Ce dernier est ensuite crypté avec la clé privée de l'émetteur et rajouté au message.

Lorsque le destinataire reçoit le message, il décrypte ce code grâce au message reçu. Si les deux correspondent, le destinataire sait que le message n'a pas été altéré et que son intégrité n'a pas été compromise. Le destinataire sait aussi que le message provient de l'émetteur puisque seul ce dernier possède la clé privée qui a crypté le code. Ce principe de signature fut amélioré avec la mise en place de certificats permettant de garantir la validité de clé publique fournie par l'émetteur [37].

3.3.4. Le code d'authentification de messages MAC

C'est un système cryptographique qui permet de vérifier l'authenticité de l'origine et l'intégrité des informations reçues. Le code d'authentification de messages MAC (Message Authentication Code) est produit par des fonctions de hachage à clé symétrique puis envoyé avec les données. Le récepteur recalcule le code MAC avec la même clé (puisque on utilise une clé symétrique) et le compare au code reçu. Si les deux codes sont identiques, la source est authentifiée et les données sont enregistrées, sinon la source est suspectée et les données sont considérées comme altérées.

Exemple de code MAC : HMAC (keyed-Hash Message Authentication Code) [38].

3.4. INFRASTRUCTURE DE GESTION DE CLÉS (PKI)

Une PKI (Public Key Infrastructure), aussi communément appelée IGC (Infrastructure de Gestion de Clefs) ou ICP (Infrastructure à Clefs Publiques) est un système de gestion des clefs publiques qui permet de gérer des listes importantes de clefs publiques et d'en assurer la fiabilité, pour des entités généralement dans un réseau. Une infrastructure PKI fournit donc quatre services principaux:

- Fabrication de bi-clés.
- Certification de clé publique et publication de certificats.
- Révocation de certificats.
- Gestion la fonction de certification [44].

3.4.1. Organisation d'une PKI

La PKI est une association de plusieurs composants qui interviennent à différentes étapes mises en œuvre depuis la création du certificat jusqu'à la l'utilisation de celui-ci. Qui sont illustrés dans la figure suivante :

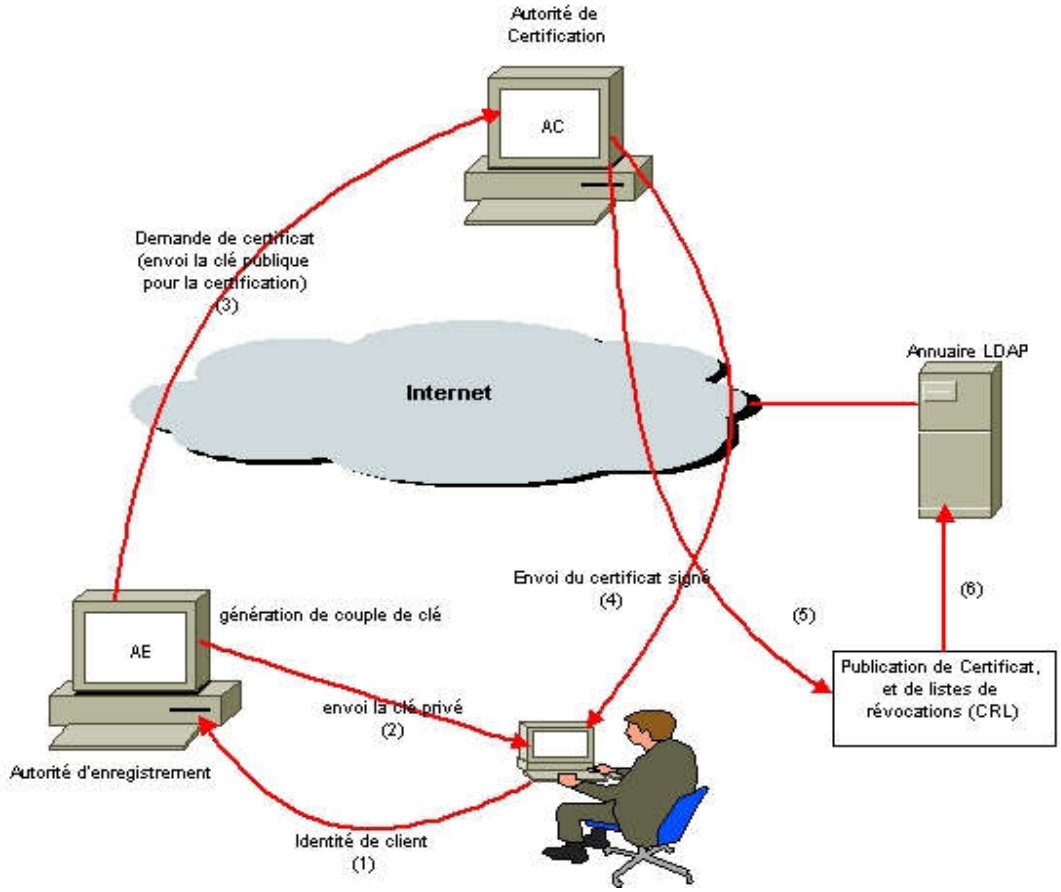

Figure 9: Organisation d'une PKI [44].

Dans une infrastructure à clé publique pour obtenir un certificat numérique, l'utilisateur fait une demande auprès de l'autorité d'enregistrement. Celle-ci génère un couple de clé (clé publique, clé privée), envoie la clé privée au client, applique une procédure et des critères définis par l'autorité de certification qui certifie la clé publique et appose sa signature sur le certificat, parfois fabriqué par un opérateur de certification [45] [44].

3.4.2. Gestion des clés

Les organismes d'infrastructure à clé publique ont besoin d'une discipline rigoureuse dans la gestion des clés. La gestion des clés proprement dite se compose des opérations suivantes :

- **Génération :** Il s'agit du moment où les clés sont initialement crées. Les clés sont générées de façon aléatoire, de manière à ce qu'elles soient non prédictibles.

- **Distribution :** La distribution est l'action de déplacer une clé de cryptage. Il existe deux étapes distinctes pour la distribution de clés, créer la clé initiale et créer les clés ultérieures. La clé initiale est établie et utilisée pour distribuer les autres clés. La génération de la clé initiale ou maîtresse est une opération critique dans tout le processus de distribution des clés au sein de la PKI.
- **Stockage :** L'étape qui suit la distribution des clés, est le stockage de la clé de façon sûre. La clé doit être protégée et doit garder à tout prix son intégrité et sa confidentialité.
- **Suppression :** La suppression signifie la destruction de toutes les copies de la clé symétrique pour un système symétrique et de la clé publique pour un système asymétrique. Une exception à cette règle intervient si le système dispose d'un processus d'archivage, dans ce cas les clés archivées ne sont jamais détruites.
- **Archivage :** L'archivage des clés permet de conserver une copie des clés même si elles ne sont plus utilisées, le but est de pouvoir valider des données qui ont été précédemment protégées par la clé.
- **Recouvrement :** Le recouvrement des clés est une procédure délicate qui permet de retrouver la clé privée d'un client. Elle peut être envisagée dans le cas où le client a perdu sa clé privée [46].

3.4.3. Le certificat numérique

La technologie des certificats électroniques est utilisée dans le cadre d'une infrastructure à clé publique. Un certificat est l'équivalent d'une carte d'identité ou d'un passeport. Il repose sur les mêmes principes. Il permet de justifier de l'identité d'un individu (ou d'une entité) sur présentation du certificat. Tout comme le passeport, il contient des informations concernant son propriétaire (nom, prénom, adresse, une signature, une date de validité). Il doit être également possible de s'assurer que ce certificat n'est pas un faux et qu'il a été délivré par une autorité reconnue. Le passeport est certifié par la préfecture tandis que le certificat est validé par une autorité de certification (AC).

Les certificats numériques associent l'identité d'une personne physique ou morale à une paire de clés qui vont être utilisées pour chiffrer et signer des informations numériques [43].

4. SÉCURITÉ DU COURRIER ÉLECTRONIQUE

4.1. LES MENACES ET RISQUES TOUCHANT AU SYSTEME DE MESSAGERIE

Compte tenu du succès de la messagerie électronique et de son aspect pervasif, les menaces sont nombreuses et peuvent être d'origines diverses. Elles peuvent être le fait d'employés mécontents, de pirates, d'espions, de terroristes, d'agences gouvernementales ou encore de professionnels du crime. En fonction de leurs origines, ces menaces peuvent prendre différentes formes d'attaques :

- **Spam**

Le spam est l'opération qui consiste à inonder les boîtes aux lettres de courriers indésirables et non sollicités. Ils exploitent des listes d'e-mail souvent obtenues à l'insu de leurs propriétaires. Leur but est soit la publicité pour des sites marchands plus ou moins recommandables, soit simplement de nuire aux systèmes de messagerie par saturation des réseaux et des boîtes aux lettres (mail bombing). Le spam est également utilisé pour diffuser en masse de faux virus (hoax).

- **Phishing**

Le phishing a pour but le vol de mots de passe, de carte bancaire notamment. La messagerie est utilisée pour transmettre un e-mail avec par exemple le logo de la banque de la victime. Dans le message, un prétexte quelconque incite le destinataire à « cliquer » sur un lien pour se rendre sur le site de sa banque. La page web reçue est une réplique de la page d'accueil du site bancaire, sur laquelle on demande la saisie par exemple des références du compte et du mot de passe associé. En réalité, ces informations sont transmises à un site web pirate.

- **Usurpation d'identité (email spoofing)**

Est une attaque classique dans les systèmes de messagerie électronique. Le principe est de modifier l'adresse d'émission du message. Cela est possible grâce à l'utilisation d'une adresse d'émission présente dans l'enveloppe SMTP différente de celle présente dans le champ d'entête du message. Cependant, les serveurs de messagerie moderne fournissent des fonctionnalités de sécurité permettant de vérifier la cohérence de ces adresses.

- **Écoute clandestine**

Permet de capturer les informations transmises entre différents protagonistes d'un échange de message. Cela peut être réalisé plus ou moins aisément. Un administrateur pourra facilement copier les messages transitant par le serveur qu'il administre. De la même manière, des transmissions sans fils non sécurisées pourront être écoutées. Un autre risque lié à l'écoute clandestine, est la perte des informations d'authentification auprès d'un service de message, typiquement le login et le mot de passe. Dans ce cas, l'individu malintentionné peut lire les messages reçus et présents dans la boite aux lettres de l'utilisateur mais il peut également envoyer des messages sous l'identité de l'utilisateur abusé.

- **Modification de messages**

Peut facilement être réalisée par un administrateur qui possède des permissions sur le serveur de messagerie. De la même manière, il peut supprimer certains messages sans que l'émetteur ni le destinataire en soit informés.

- **Génération de faux messages**

On parle de forger un message, son fait parti des attaques courantes dans les systèmes de messagerie. Dans les systèmes non sécurisés, il est relativement simple de générer un faux message émanant d'un utilisateur abusé. Il est difficile pour ce dernier de prouver qu'il n'est pas l'émetteur du message, il peut dans ces cas, nier être l'émetteur.

- **Rejeu de message**

Est un type d'attaque dont l'objectif est de sauvegarder un message puis de le réémettre plus tard en faisant en sorte qu'il apparaisse comme valide.

- **Logiciels malveillants**

La messagerie est un vecteur important de transmission de logiciels malveillants (malware) comme les vers, virus, etc. Ils peuvent être directement intégrés dans le message sous la forme de pièces jointes ou bien être transportés sous la forme de lien vers des sites hébergeant des logiciels malveillants [1] [2].

4.2. SOLUTIONS DE SÉCURITÉ POUR LA MESSAGERIE ÉLECTRONIQUE

Aujourd'hui, les solutions de sécurisation des systèmes de messagerie sont basées en grande partie sur des mécanismes cryptographiques. Ces mécanismes permettent d'assurer et de vérifier les propriétés décrites précédemment. Les solutions de sécurité s'appliqueront sur les deux catégories : l'acheminement du message et le message.

4.2.1. Sécurisation de l'acheminement du message

- **SSL (Secure Socket Layer)**

Le protocole SSL (Secure Socket Layer) est un protocole cryptographique, défini en 1994 par Netscape, appliqué à la communication sécurisée point à point entre deux entités d'un réseau. C'est un protocole de sécurisation des échanges électroniques, qui est intégré dans tous les butineurs (les navigateurs) des terminaux connectables à Internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification et d'assurer la confidentialité des données, l'authentification de l'utilisateur et du serveur, ainsi que l'intégrité des données par signature électronique [23] [47].

- **TLS (transport Layer Security)**

SSL (version 3) a été repris et transformé par l'IETF en TLS (transport Layer Security). TLS offre des services de sécurité qui permet à une application cliente et une application serveur de communiquer de manière sécurisée, répondant notamment aux menaces d'écoute passive, de génération de faux messages et d'altération de messages. TLS est composé de deux protocoles : TLS Record Protocol et TLS Handshake Protocol. Le protocole TLS Handshake est chargé d'authentifier le client et le serveur, de négocier un algorithme de chiffrement et de générer les clés cryptographiques permettant une communication sécurisée. Le protocole TLS Record est chargé de protéger les données échangées par l'application. Un des avantages de TLS est d'être indépendant des couches applicatives. Cela permet notamment à la messagerie de mettre en œuvre ce protocole lors de l'acheminement d'un message [1].

❖ **PPTP (Point-to-Point Tunneling Protocol)**

PPTP (Point-to-Point Tunneling Protocol) est une extension du protocole PPP (Point-to-Point Protocol), soumise à l'IETF en 1996. PPTP est un protocole de tunnel qui permet aux données passant d'une extrémité à l'autre du tunnel d'être sécurisées par des algorithmes de cryptographie. PPTP encapsule des trames PPP dans des datagrammes IP pour une transmission sur un réseau IP, tel qu'Internet. C'est un protocole utilisé dans la mise en place des VPN.

❖ **Les réseaux virtuels privés (VPN)**

Un réseau virtuel privé (Virtual Private Network, VPN) consiste en la fabrication d'un tunnel logique qui sera contracté par les communications de l'entreprise, lesquelles seront véhiculées dans cette tranchée numérique construite sur un réseau fréquenté par d'autres usagers. Le caractère privé est créé par un protocole cryptographique (PPTP). Un VPN est donc une communication sécurisée entre deux points d'un réseau public, d'où l'expression de tunnel [47].

4.2.2. Sécurisation du message

❖ **PGP (Pretty Good Privacy)**

Le système PGP (Pretty Good Privacy) sert pour la protection du courrier électronique. Conçu par Philip. Zimmerman en 1991, il avait pour objectif de proposer une solution simple pour assurer la sécurité de la messagerie électronique de personne à personne. Après bien des déboires (liés à la législation américaine sur l'exportation et à la protection industrielle de certains algorithmes qu'il utilisait), il est maintenant largement répandu grâce à sa version libre téléchargeable. Chacun des utilisateurs génère sa propre paire de clés (publique/ privée) et distribue sa clé publique à ses interlocuteurs de confiance. Il est possible de crypter les messages (avec la clé publique du récepteur) et de les signer (avec la clé secrète de l'émetteur).

Il possède les caractéristiques suivantes :

- ✓ Il garantit l'origine des messages et leur confidentialité.
- ✓ Il s'agit d'une solution qui s'exécute sur un grand nombre de plates formes dont DOS, Windows, Unix, Linux, Mac OS.
- ✓ Elle est disponible gratuitement [48].

- **GPG (GNU Privacy Guard)**

GPG est la version GNU (**G**NU's **N**ot **U**NIX) de PGP. En raison qu'il est sous licence GPL (General Public License) il permet la fourniture du code-source, la libre modification et la libre redistribution de ce code-source. Ainsi, il est remis à jour continuellement, aussi bien au niveau des fonctionnalités, qu'au niveau des éventuels problèmes d'implémentation. [37]

- **S/MIME (Secure/Multipurpose Internet Mail Extension)**

Le standard S/MIME développé initialement par la société RSA Data Security, est largement implémenté dans les solutions de messagerie actuelles. Il propose des mécanismes pour sécuriser le contenu des messages, et plus précisément le contenu MIME qui, pour sa part, ne fournit aucun service de sécurité.

S/MIME permet de garantir l'authentification de l'origine, l'intégrité du message et la non-répudiation de l'origine en utilisant les mécanismes de la signature électronique et la confidentialité du message en utilisant les mécanismes de chiffrement. La force de S/MIME représente aussi son défaut. Quand un message est chiffré, il ne peut être déchiffré que par son destinataire. Il ne peut donc pas subir d'analyse antivirale, ni de vérification de contenu pour prévenir la divulgation d'information sensible [1].

5. CONCLUSION

Nous avons essayé à travers ce chapitre de mettre le point sur la sécurité informatique, ses propriétés, ainsi que les principales attaques réseau et les mécanismes de protection contre ces attaques. Nous avons, ensuite détaillé les mécanismes cryptographiques. Enfin, nous avons présenté les menaces existantes sur les systèmes de messagerie électronique et les solutions utilisés pour assurer la sécurité de la transmission de messages.

Dans le prochain chapitre nous allons présenter une conception détaillée de notre application de messagerie électronique sécurisée.

CHAPITRE 03 :

CONCEPTION DE L'APPLICATION DE MESSAGERIE ÉLECTRONIQUE SÉCURISÉE

1. INTRODUCTION

Pour développer une application, il convient d'organiser ses idées et de les documenter avant l'écriture même du code. La suite des étapes antérieure à l'écriture est dite modélisation. Elle doit nécessairement respecter l'une des méthodes de développement d'applications informatiques. Le développeur doit également prendre en considération le modèle de fonctionnement logiciel.

C'est pourquoi, pour étudier et modéliser notre développement, des méthodes de conception et de développement orientées objet ont été mises au point. Il existe de nombreuses méthodes disponibles. Notre choix c'est porté vers une méthode basée sur UP qui s'appuie sur le langage de modélisation UML tout au long du cycle de développement.

Dans ce chapitre, nous allons présenter les concepts du langage de modélisation unifié UML et ses diagrammes, ainsi que les principes et les phases du processus UP. Nous compléterons ensuite cette présentation générale en revue les différentes étapes de la méthode de développement, inspirée des étapes du processus UP, que nous avons choisi pour la conduite de notre projet. Pour finir nous détaillerons le diagramme de cas d'utilisation, les diagrammes de séquence, le modèle de domaine, quelques digrammes d'interaction et enfin le modèle de déploiement.

.

2. LANGAGE DE MODÉLISATION ET MÉTHODE DE CONCEPTION

2.1. LE LANGAGE DE MODÉLISATION UML

2.1.1. Définition

UML (Unified Modeling Language), que l'on peut traduire par (Langage de Modélisation Unifié), est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et est devenu désormais la référence en termes de modélisation objet.

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier, concevoir des solutions et communiquer des points de vue. Il unifie les notations nécessaires aux différentes activités d'un processus de développement et offre, par ce biais, le moyen d'établir le suivi des décisions prises, depuis la définition des besoins jusqu'au codage.

UML permet de représenter un système selon différentes vues complémentaires : les diagrammes [49].

2.1.2. Historique

Au début des années 90, de cinquantaine de méthodes d'analyse et de conception objet qui existaient, seulement trois d'entre elles se sont détachées nettement au bout de quelques années. En effet, la volonté de converger vers une méthode unifiée était déjà bien réelle et c'est pour cette raison que les méthodes OMT, BOOCH et OOSE se sont démarquées des autres.

OMT (Object Modeling Technique) de James Rumbaugh et BOOCH de Grady Booch ont été les deux méthodes les plus diffusées en France durant les années 90. Par ailleurs, OOSE (Object Oriented Software Engineering) d'Ivar Jacobson s'est aussi imposée dans le monde objet pour la partie formalisation des besoins.

Pour aller plus loin dans le rapprochement, James Rumbaugh et Grady Booch se sont retrouvés au sein de la société Rational Software et ont été ensuite rejoints par Ivar Jacobson en se donnant comme objectif de fusionner leur méthodes et créer UML (Unified Methode Language).

UML est donc une norme du langage de modélisation objet qui a été publiée, dans sa première version, en novembre 1997 par l'OMG (Object Management Group), instance de normalisation internationale du domaine de l'objet.

En quelques années, UML s'est imposée comme standard à utiliser en tant que langage de modélisation objet.

2.1.3. Les différents diagrammes UML

Les diagrammes sont des éléments graphiques décrivant le contenu des vues, qui sont des notions abstraites. Les diagrammes peuvent faire partie de plusieurs vues.

UML dans sa version 2 propose treize diagrammes qui peuvent être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux vues différentes du système. Nous présentons, ici, les diagrammes que nous allons utiliser pour la conception de notre application.

2.1.3.1. La vue statique (structurelle)

Représente la structure statique du système en utilisant des objets, des attributs, des opérations et des relations. Parmi les diagrammes de la vue statique :

- **Diagramme de classes**

Ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements. C'est le diagramme pivot de l'ensemble de la modélisation d'un système.

- ✓ **Objet :** est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.
- ✓ **Classe :** décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).
- ✓ **Attribut** : est une propriété élémentaire d'une classe. Pour chaque objet d'une classe, l'attribut prend une valeur.
- ✓ **Opération :** est une fonction applicable aux objets d'une classe. Une opération permet de décrire le comportement d'un objet. Une méthode est l'implémentation d'une opération.
- ✓ **Lien et association :** un lien est une connexion physique ou conceptuelle entre instances de classes donc entre objets. Une association entre classes représente les liens qui existent entre les instances de ces classes.

- **Diagramme de déploiement**

Ce diagramme décrit l'architecture technique d'un système avec une vue centrée sur la répartition des composants dans la configuration d'exploitation.

2.1.3.2. La vue dynamique (comportementale)

Met l'accent sur le comportement dynamique du système en montrant la collaboration entre les objets et les modifications apportées à l'état interne des objets. Parmi les diagrammes de la vue dynamique :

- **Diagramme des cas d'utilisation**

Ce diagramme est destiné à représenter les besoins des utilisateurs par rapport au système. Il constitue un des diagrammes les plus structurants dans l'analyse d'un système. Les éléments de modélisation des cas d'utilisation sont :

- ✓ **Acteur :** entité externe qui agit sur le système. Le terme acteur ne désigne pas seulement les utilisateurs humains mais également les autres systèmes.
- ✓ **Cas d'utilisation :** ensemble d'actions réalisées par le système en réponse à une action d'un acteur.

- ✓ **Relations entre les cas d'utilisation :** UML définit trois types de relations standardisées entre cas d'utilisation, détaillées ci-après.

  - **Relation d'inclusion :** indique que le cas d'utilisation source contient obligatoirement le cas d'utilisation destination. Formalisée par « include ».
  - **Relation d'extension:** signifie que le cas d'utilisation source étend le comportement du cas d'utilisation destination, l'extension peut être soumise à condition. Formalisée par le mot clé « extend ».
  - **Relation de généralisation/spécialisation :** utilisée pour montrer l'hiérarchie entre les cas d'utilisations.

- ✓ **Scénario :** un cas d'utilisation est une abstraction de plusieurs chemins d'exécution. Une instance de cas d'utilisation est appelée : « scénario ». Il y'a trois types de scénario : scénario nominal, scénario alternatif et scénario d'erreur.

- **Diagramme de séquence**

Ce diagramme permet de décrire les scénarios de chaque cas d'utilisation en mettant l'accent sur la chronologie des opérations en interaction avec les objets [50].

- ✓ **Ligne de vie** : représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.
- ✓ **Message synchrone et asynchrone :** Dans un diagramme de séquence, deux types de messages peuvent être distingués : message synchrone dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. Message asynchrone Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations [50].

- **Diagramme global d'interaction**

Permet de représenter une vue générale des interactions décrites dans le diagramme de séquence et des flots de contrôle décrits dans le diagramme d'activité [51].

2.2. LE PROCESSUS UNIFIE UP

2.2.1. Définition

Le UP ou Unified Process est un processus de conception/développement de logiciel construit sur UML. Il apporte une approche disciplinée pour assigner des tâches et des responsabilités dans une organisation de développement. Son but est d'assurer la production de logiciels de qualité qui rencontrent les besoins de ses utilisateurs finaux dans un horaire et un budget prédictibles [52].

Donc le processus UP est un patron de processus pouvant être adaptée à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences et à différentes tailles de l'entreprise [53].

2.2.2. Les principes d'UP

- **Itératif et incrémental :** la méthode est itérative dans le sens où elle propose de faire des itérations lors de ses différentes phases, ceci garantit que le modèle construit à chaque phase ou étape soit affiné et amélioré. Chaque itération peut servir à ajouter de nouveaux incréments. Un incrément est une augmentation ou amélioration apportée à la construction en cours d'un système.
- **Centré sur l'architecture :** les modèles définis tout au long du processus de développement vont contribuer à établir une architecture cohérente et solide.
- **Conduit par les cas d'utilisation** : le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc axé sur l'utilisateur. Les cas d'utilisation permettent d'illustrer ces besoins. Ils détectent puis décrivent les besoins fonctionnels (du point de vue de l'utilisateur), et leur ensemble constitue le modèle de cas d'utilisation qui dicte les fonctionnalités complètes du système [51].
- **Orienté par la réduction des risques :** l'analyse des risques doit être présente à tous les stades de développement d'un système. Il est important de bien évaluer les risques des développements afin d'aider à la bonne prise de décision. Du fait de l'application du processus itératif, UP contribue à la diminution des risques au fur et à mesure du déroulement des itérations successives [50].

2.2.3. Les phases d'UP

La gestion d'un tel processus est organisée d'après les 4 phases suivantes: pré-étude (analyse des besoins), élaboration, construction et transition.

❖ **Pré-étude (analyse des besoins)**

L'analyse des besoins donne une vue du projet sous forme de produit fini. Elle porte essentiellement sur les besoins principaux (du point de vue de l'utilisateur), l'architecture générale du système, les risques majeurs. Elle répond aux questions suivantes :

- ✓ Que va faire le système ? Par rapport aux utilisateurs principaux, quels services va-t-il rendre?
- ✓ Quelle va être l'architecture générale (cible) de ce système?
- ✓ Quels vont être : les délais, les coûts, les ressources, les moyens à déployer?

❖ **Elaboration**

L'élaboration reprend les éléments de la phase d'analyse des besoins et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. Elle permet de préciser la plupart des cas d'utilisation et de concevoir l'architecture du système. L'architecture doit être exprimée sous forme de vue de chacun des modèles. Au terme de cette phase, le chef de projet doit être en mesure de prévoir les activités et d'estimer les ressources nécessaires à l'achèvement du projet [54].

❖ **Construction**

Cette phase correspond à la production d'une première version du produit. Elle est donc fortement centrée sur les activités de conception, d'implémentation et de test. En effet, les composants et fonctionnalités non implémentés dans la phase précédente le sont ici. Au cours de cette phase, la gestion et le contrôle des ressources ainsi que l'optimisation des coûts représentent les activités essentielles pour aboutir à la réalisation du produit. En parallèle est rédigé le manuel utilisateur de l'application.

❖ **Transition :**

Permet de faire passer le système informatique de mains des développeurs à celles des utilisateurs finaux. Après les opérations de test menées dans la phase précédente, il s'agit dans cette phase de livrer le produit pour une exploitation réelle. C'est ainsi que toutes les actions liées au déploiement sont traitées dans cette phase. De plus, des « bêta tests » sont effectués pour valider le nouveau système auprès des utilisateurs [50].

2.3. MÉTHODE DE DÉVELOPPEMENT

Le processus UP (Unified Process) constitue un cadre général très complet de processus de développement. Pour réaliser notre projet, nous avons choisi de suivre une méthode simple et générique basée sur le processus UP.

Cette méthode consiste à une démarche grammatique et simplifiée pour conduire l'activité d'analyse et de conception d'un système informatique. Et qui comporte les étapes suivantes :

- **Identification des besoins**
  - ✓ Présentation du projet.
  - ✓ Diagramme de cas d'utilisation (avec identification des acteurs).
  - ✓ Description des cas d'utilisation.
  - ✓ Les diagrammes de séquence système.
- **Analyse du domaine**
  - ✓ Modèle de domaine.
- **Conception**
  - ✓ Diagramme d'interaction.
  - ✓ Développement du modèle de déploiement (architecture adoptée, déploiement du modèle d'exploitation).

3. ANALYSE ET CONCEPTION DE L'APPLICATION DE MESSAGERIE ÉLECTRONIQUE SÉCURISÉE

3.1. IDENTIFICATION DES BESOINS

3.1.1. PRESENTATION DU PROJET

Dans le cadre de ce projet de fin d'études, nous allons développer une application de messagerie électronique sécurisée que nous avons appelé « Secure Mail ». Il s'agit d'une application comportant un serveur de messagerie et un client de messagerie lourd, permettant aux utilisateurs d'échanger du courrier électronique d'une manière sécurisée.

Dans le but d'assurer la confidentialité et l'intégrité des messages échangés entre les utilisateurs du système, nous allons implémenter et intégrer des protocoles cryptographiques au niveau de notre application. Ces protocoles permettent le chiffrement des messages envoyés et le déchiffrement des messages reçus, ainsi que la signature des messages envoyés et la vérification de la signature des messages reçus.

L'application « Secure Mail » représente un système de messagerie électronique complet qui pourra être déployé dans le réseau Internet ou bien installé dans un réseau local LAN à l'intérieur d'un établissement ou une entreprise et sera exclusivement réservée aux utilisateurs de l'entreprise. Les utilisateurs du système « Secure Mail » ne pourrons échanger de messages qu'avec les membres de ce système.

L'application « Secure Mail » fournit à ses utilisateurs les services suivants :

Dans un premier temps, il est nécessaire pour l'utilisateur de l'application de s'inscrire, c.à.d. Créer un compte pour devenir un membre du système. Par la création d'un compte, l'utilisateur sera enregistré dans le système et identifié par une adresse électronique unique qui a le format :

nomd'utilisateur@smail.dz

L'adresse est composée d'un nom d'utilisateur unique, suivi de @, après du nom de notre serveur de messagerie « smail », et enfin du nom de domaine « dz ».

Pour pouvoir accéder aux services offerts par le système, un membre doit s'authentifier auprès du système par son adresse électronique et son mot de passe.

Après avoir accédé à sa session, un membre peut :

- Mettre à jour les informations de son compte (modifier ces informations : profil, mot de passe, etc.).
- Gérer ses contacts (ajouter, modifier, supprimer des contacts) qui doivent être des membres du système « Secure Mail ».
- Gérer ses paires de clés publiques/privés (créer ou supprimer des paires de clés) avec choix de l'algorithme de chiffrement et des tailles de clefs (l'algorithme de chiffrement asymétrique RSA avec les tailles de clefs : 1024, 2048 ou 4096, ou bien l'algorithme de chiffrement mixte ElGamal&AES avec les tailles de clefs : 1024, 2048, ou 4096).
- Composer un message qui peut contenir : la liste des destinataires, un texte, une pièce jointe. L'utilisateur peut chiffrer le texte contenu dans le message avec choix des clefs publiques de chiffrement. Il peut aussi signer le texte du message (en utilisant l'algorithme RSA et la fonction de hachage MD5) avec choix de clef privée. Ensuite, l'utilisateur peut envoyer ce message ou l'enregistrer dans le brouillon.

- Consulter ses messages envoyés, reçus, et enregistrés dans le brouillon. L'utilisateur peut déchiffrer le texte d'un message reçus et vérifier sa signature. Il peut aussi faire d'autres opérations comme télécharger une pièce jointe, transférer et supprimer un message.
- Se déconnecter du système.

3.1.2. DIAGRAMME DE CAS D'UTILISATION

❖ **Identification des acteurs**

Nous avons identifié les acteurs suivants qui interagissent avec notre système :

- **Utilisateur :** c'est la personne qui consulte le système pour créer un compte et devenir un membre.
- **Membre :** c'est une personne déjà inscrite dans le système, et qui peut accéder au système pour envoyer des messages, consulter ses messages reçus, envoyés et enregistrés dans le brouillon, gérer ses contacts, mettre à jour son compte, et gérer ses clefs.

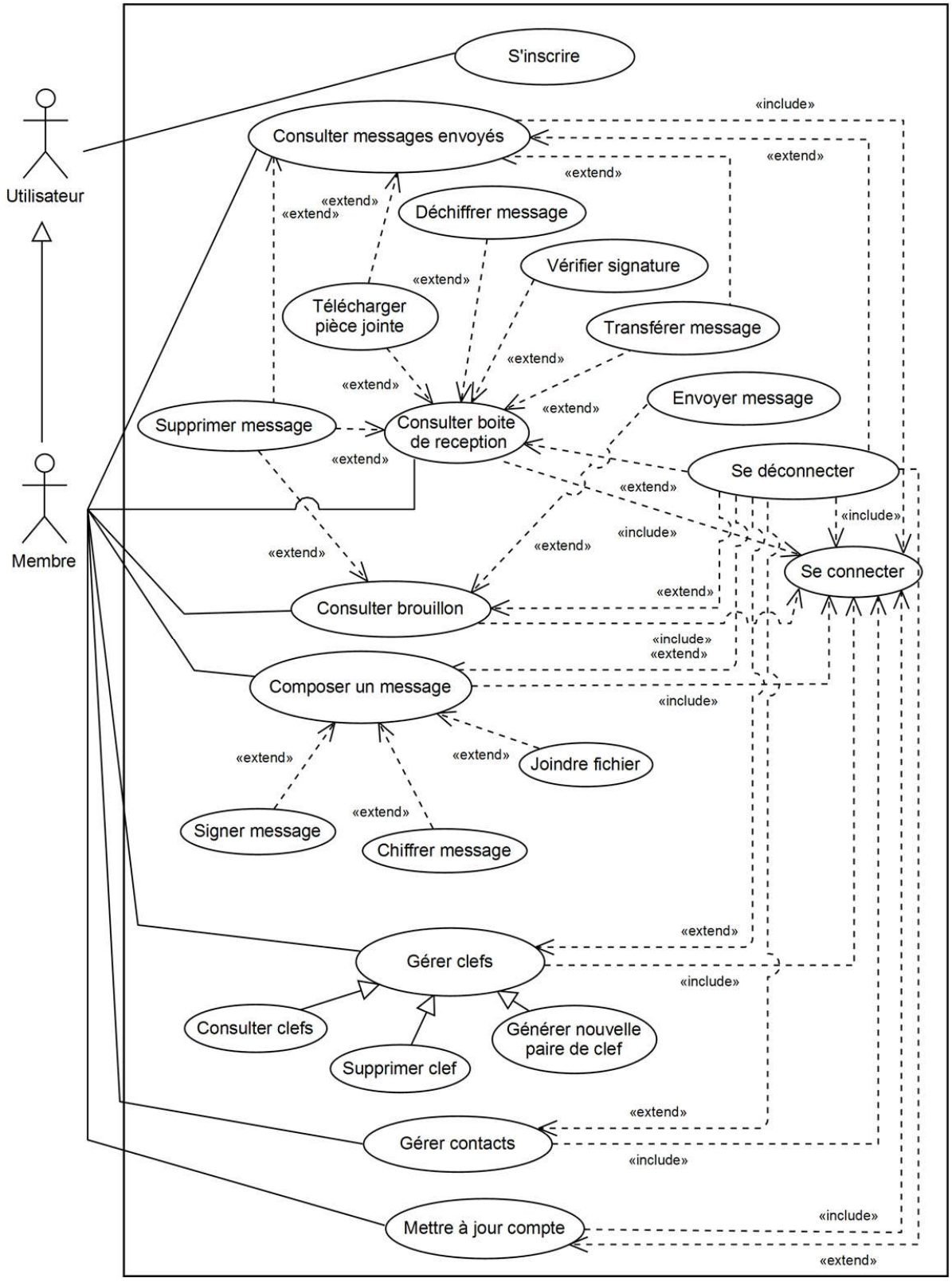

Figure 10: Diagramme de cas d'utilisation.

3.1.3. DESCRIPTION DES CAS D'UTILISATION

❖ **Cas d'utilisation « S'inscrire »**

Cas d'utilisation	S'inscrire
Acteur	Utilisateur
But	Inscrire un nouvel utilisateur dans le système de messagerie électronique.
Pré-condition	/
Post-condition	L'utilisateur est enregistré dans la base de données.
Scénario nominal	1. L'utilisateur demande de s'inscrire dans le système de messagerie électronique. 2. Le système affiche le formulaire d'inscription. 3. L'utilisateur saisit les informations nécessaires et envoie le formulaire. 4. Le système vérifie les informations entrées. 5. Le système enregistre les informations entrées dans la base de données. 6. Le système affiche un message de confirmation.
Scénario alternatif	A1. L'utilisateur n'a pas remplit certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. A2. L'utilisateur existe déjà dans la base de données. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

Tableau 8: Description textuelle du cas d'utilisation « S'inscrire ».

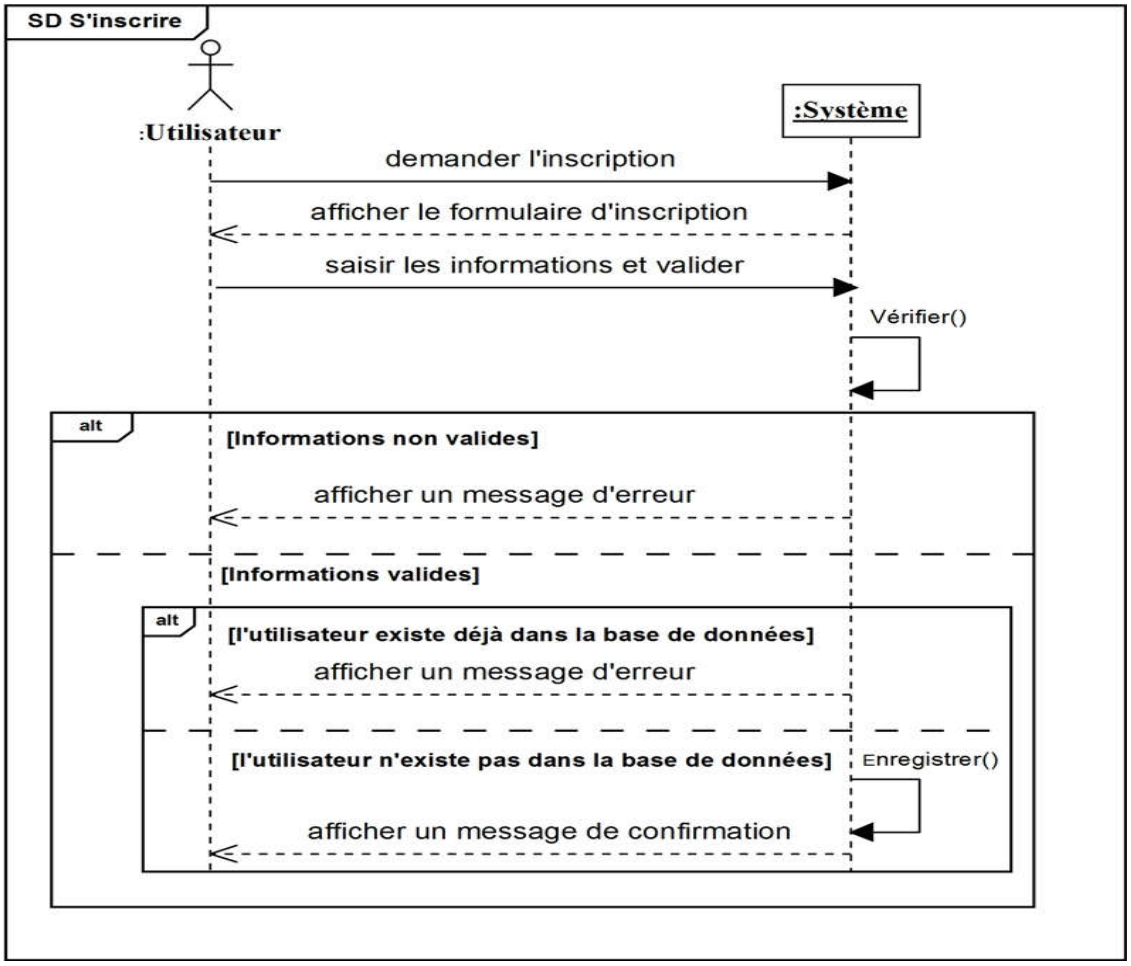

Figure 11: Diagramme de séquence « S'inscrire ».

❖ **Cas d'utilisation « Se connecter »**

Cas d'utilisation	Se connecter
Acteur	Membre
But	Permet au membre d'accéder à sa session.
Pré-condition	Le membre doit être déjà inscrit.
Post-condition	Le membre est authentifié par le système.
Scénario nominal	1. Le membre demande de se connecter au système de messagerie électronique. 2. Le système affiche le formulaire d'authentification. 3. Le système demande au membre de saisir son adresse e-mail et son mot de passe. 4. Le membre saisit l'adresse e-mail et le mot de passe puis valide.

	5. Le système vérifie les informations entrées. 6. Le système affiche la session du membre.
Scénario alternatif	A1. Le membre n'a pas remplit certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

Tableau 9: Description textuelle du cas d'utilisation « Se connecter ».

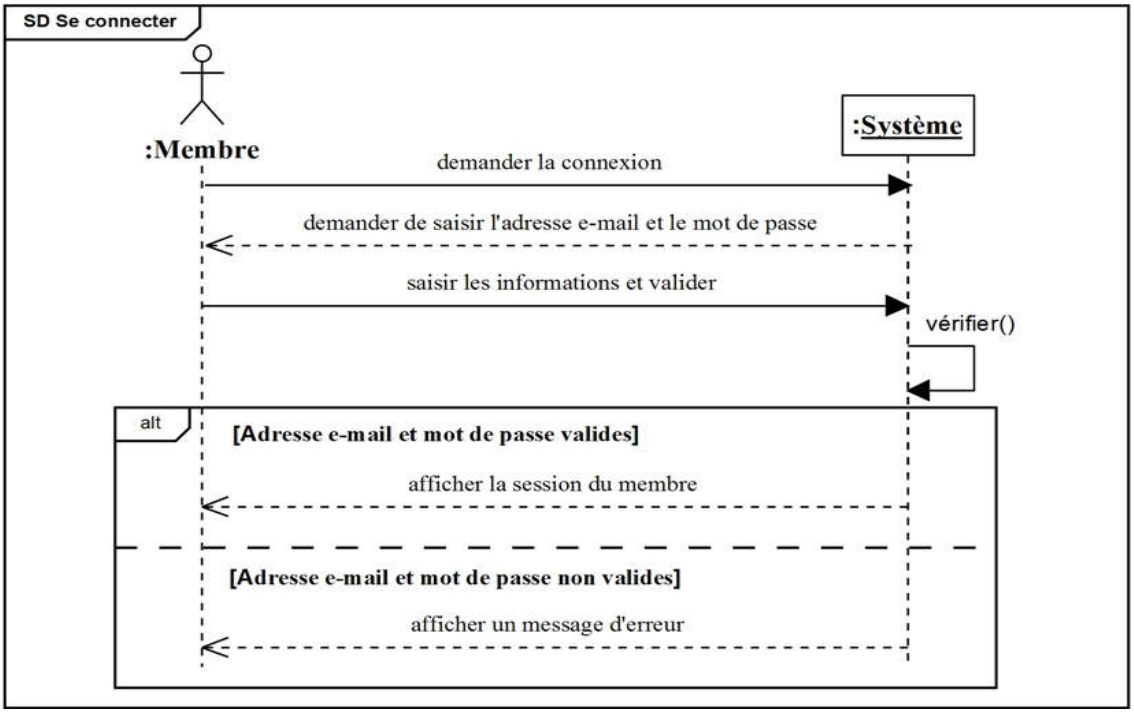

Figure 12 : Diagramme de séquence « Se connecter ».

❖ **Cas d'utilisation « Mettre à jour compte »**

Cas d'utilisation	Mettre à jour compte
Acteur	Membre
But	Permet au membre de mettre à jour son compte.
Pré-condition	Le membre doit être connecté.
Post-condition	Mise à jour de la base de données.

Scénario nominal	1. Le membre demande de mettre à jour son compte. 2. Le système affiche les informations du membre. 3. Le membre modifie les informations désirées et valide. 4. Le système enregistre les nouvelles informations dans la base de données. 5. Le système affiche un message de confirmation.

Tableau 10: Description textuelle du cas d'utilisation « Mettre à jour compte ».

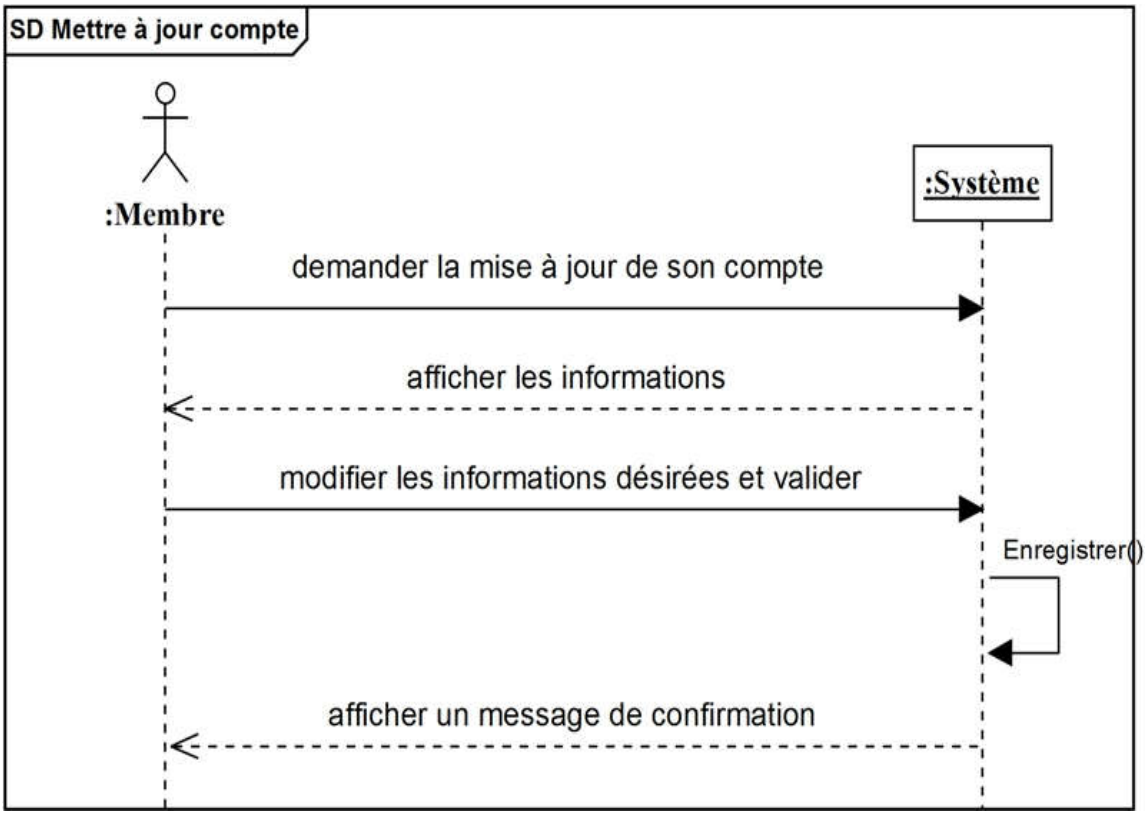

Figure 13: Diagramme de séquence « Mettre à jour compte ».

❖ **Cas d'utilisation « Gérer contacts »**

Cas d'utilisation	Gérer contacts
Acteur	Membre
But	Permet au membre de gérer ses contacts (ajouter un contact, modifier un contact, supprimer un contact).
Pré-condition	Le membre doit être connecté.
Post-condition	Mise à jour de la base de données.

Scénario nominal	1. Le membre demande de gérer ses contacts. 2. Le système affiche un choix (Ajouter contact, Modifier contact, Supprimer contact). 3. Le membre choisit « Ajouter contact ». 4. Le système affiche le formulaire correspondant. 5. Le membre saisit les informations puis valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre le contact dans la base de données. 8. Le système affiche un message de confirmation.
Scénario alternatif	A1. Le membre n'a pas rempli certains champs ou a saisit des informations non valides. 1. Le système indique à membre les champs non accepté. 2. Le système retourne à l'étape numéro 2 du scénario nominal. A2. 1. Le membre choisit « Modifier contact ». 2. Le système affiche la liste des contacts. 3. Le membre sélectionne un contact. 4. Le système affiche les informations du contact. 5. Le membre modifie les informations désirées et valide. 6. Le système enregistre les nouvelles informations dans la base de données. 7. Le système affiche un message de confirmation. A3. 1. Le membre choisit « Supprimer contact ». 2. Le système affiche la liste des contacts. 3. Le membre sélectionne un contact. 4. Le système met à jour la base de données. 5. Le système affiche un message de confirmation.

Tableau 11: Description textuelle du cas d'utilisation « Gérer contacts ».

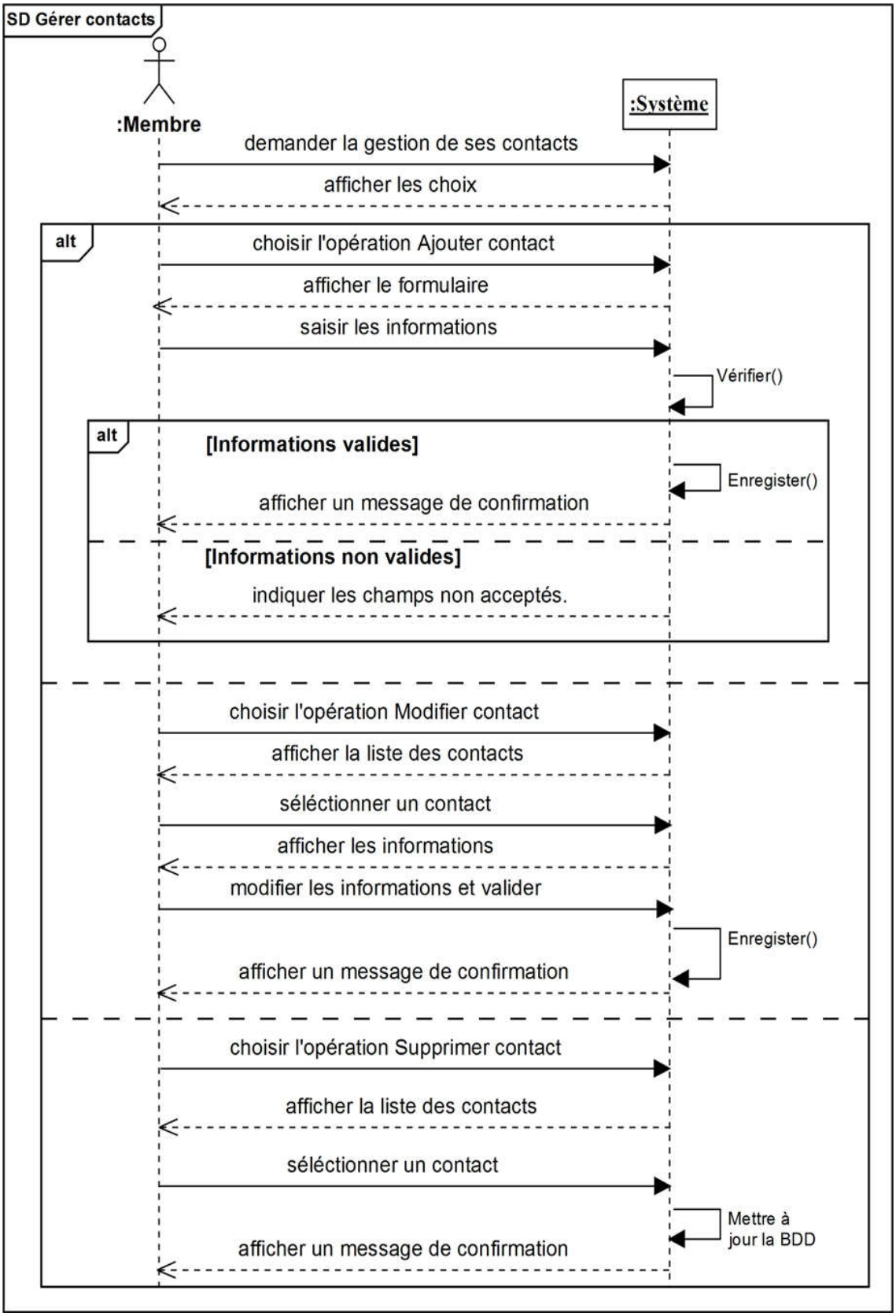

Figure 14: Diagramme de séquence « Gérer contacts ».

❖ **Cas d'utilisation « Gérer clefs »**

Cas d'utilisation	Gérer clefs
Acteur	Membre
But	Permet au membre de gérer ses clefs (Générer nouvelle paire de clefs, Consulter, Supprimer clef).
Pré-condition	Le membre doit être connecté.
Post-condition	Mise à jour de la base de données.
Scénario nominal	1. Le membre demande de gérer ses clefs. 2. Le système affiche les opérations (Générer nouvelle paire de clefs, Consulter clefs, Supprimer clef). 3. Le membre choisit Générer nouvelle paire de clefs. 4. Le système affiche le formulaire correspondant. 5. Le membre saisit les informations puis valide. 6. Le système vérifie les informations entrées. 7. Le système enregistre la paire de clefs dans la base de données. 8. Le système affiche un message de confirmation.
Scénario alternatif	A1. Le membre n'a pas rempli certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 4 de scénario nominal. A2. 1. Le membre choisit l'opération Consulter clefs. 2. Le système affiche la liste des clefs. A3. 3. Le membre choisit l'opération Supprimer clef. 4. Le système met à jour la base de données. 5. Le système affiche un message de confirmation.

Tableau 12: Description textuelle du cas d'utilisation « Gérer clefs ».

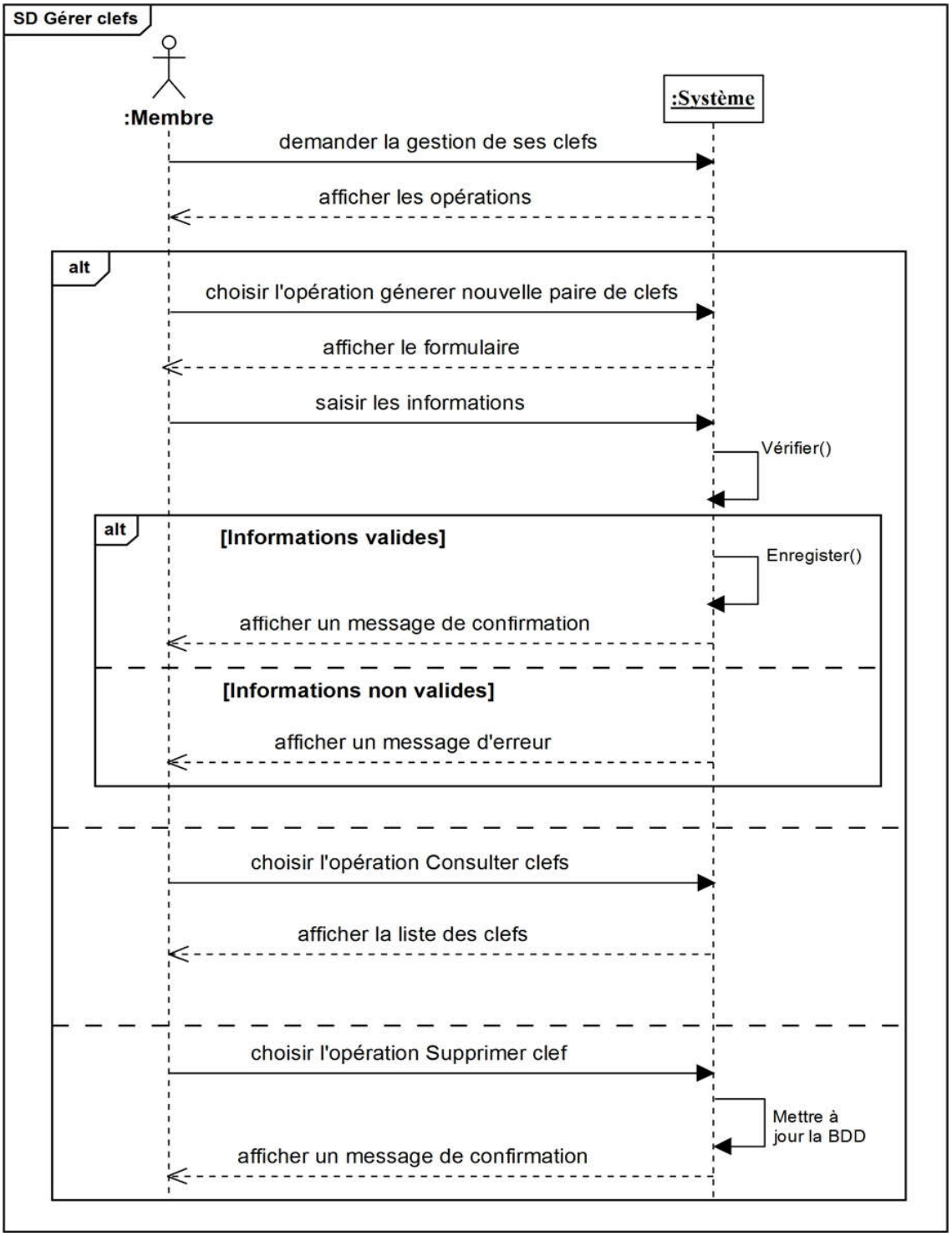

Figure 15: Diagramme de séquence « Gérer clefs ».

❖ **Cas d'utilisation « Composer un message »**

Cas d'utilisation	Composer un message
Acteur	Membre
But	Permet au membre de composer, et d'envoyer ou d'enregistrer un message.
Pré-condition	Le membre doit être connecté.
Post-condition	Le message est enregistré dans la base de données.
Scénario nominal	1. Le membre demande de composer un nouveau message. 2. Le système affiche le formulaire de composition d'un message. 3. Le membre remplit les champs du formulaire. 4. Le système affiche les opérations (Joindre fichiers, Chiffrer message, Signer message, Enregistrer dans le brouillon, Envoyer). 5. Le membre choisit l'opération Envoyer. 6. Le système vérifie les informations entrées. 7. Le système enregistre le message dans la base de données. 8. Le système affiche un message de confirmation.
Scénario alternatif	A1. Le membre n'a pas remplit certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. A2. 1. Le membre choisit l'opération Enregistrer dans le brouillon. 2. Le système enregistre le message dans la base de données. 3. Le système affiche un message de confirmation. A3. 1. Le Membre choisit l'opération Joindre fichier. 2. Le système appelle le cas d'utilisation « Joindre fichier ».

	3. Le système retourne à l'étape numéro 4 du scénario nominal. A4. 1. Le Membre choisit l'opération Chiffrer message. 2. Le système appelle le cas d'utilisation « Chiffrer message ». 3. Le système retourne à l'étape numéro 4 du scénario nominal. A5. 1. Le Membre choisit l'opération Signer message. 2. Le système appelle le cas d'utilisation « Signer message ». 3. Le système retourne à l'étape numéro 4 du scénario nominal.

Tableau 13: Description textuelle du cas d'utilisation « Composer un message ».

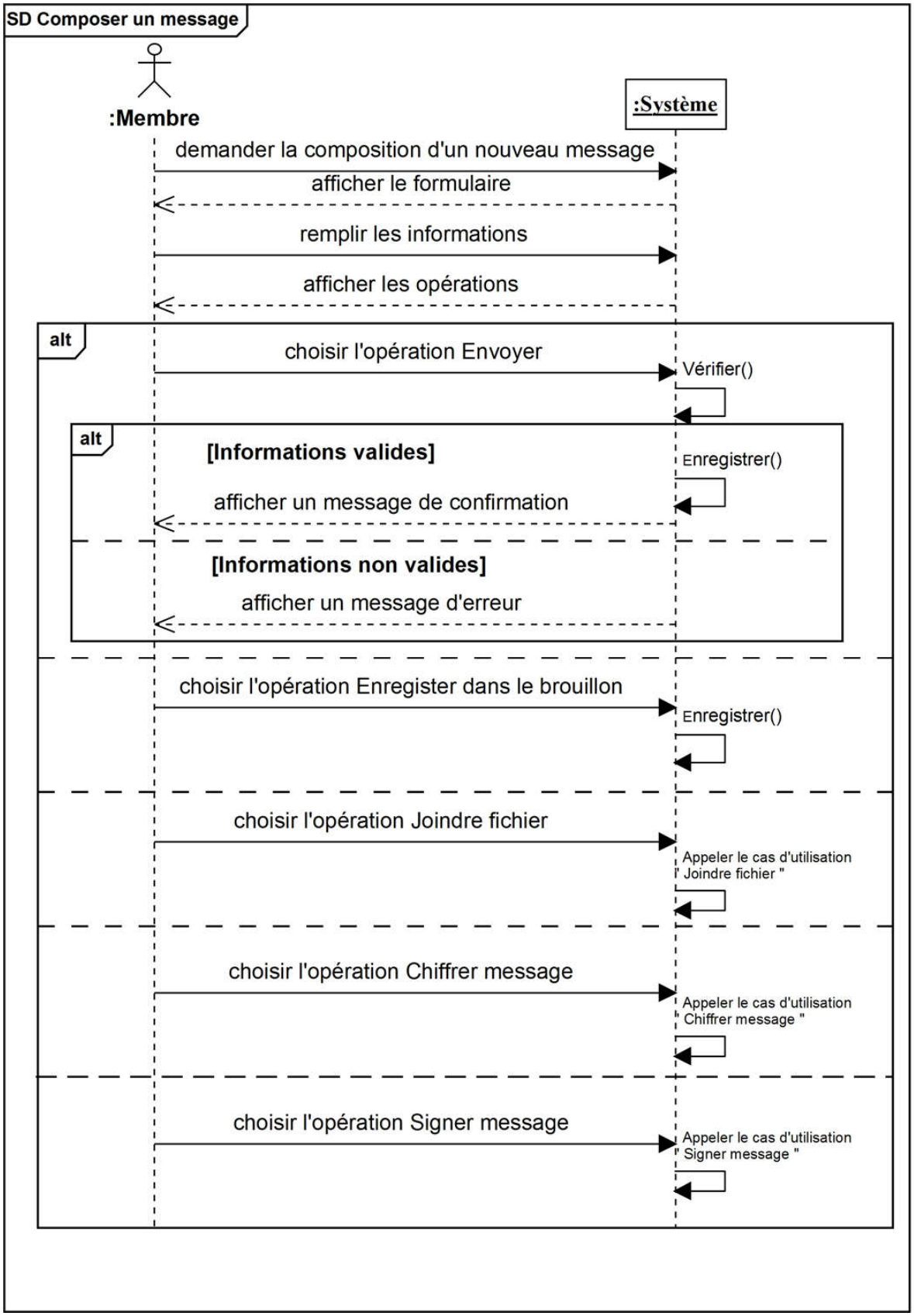

Figure 16: Description textuelle du cas d'utilisation « Composer un message ».

❖ **Cas d'utilisation « Joindre fichier »**

Cas d'utilisation	Joindre fichier
Acteur	Membre
But	Permet au membre de joindre un fichier au message.
Pré-condition	Le membre doit être connecté.
Post-condition	Le fichier est joint au message.
Scénario nominal	1. Le membre demande de joindre un fichier au message. 2. Le système affiche le formulaire de sélection d'un fichier. 3. Le membre sélectionne un fichier. 4. Le système joint le fichier au message.

Tableau 14: Description textuelle du cas d'utilisation « Joindre fichier ».

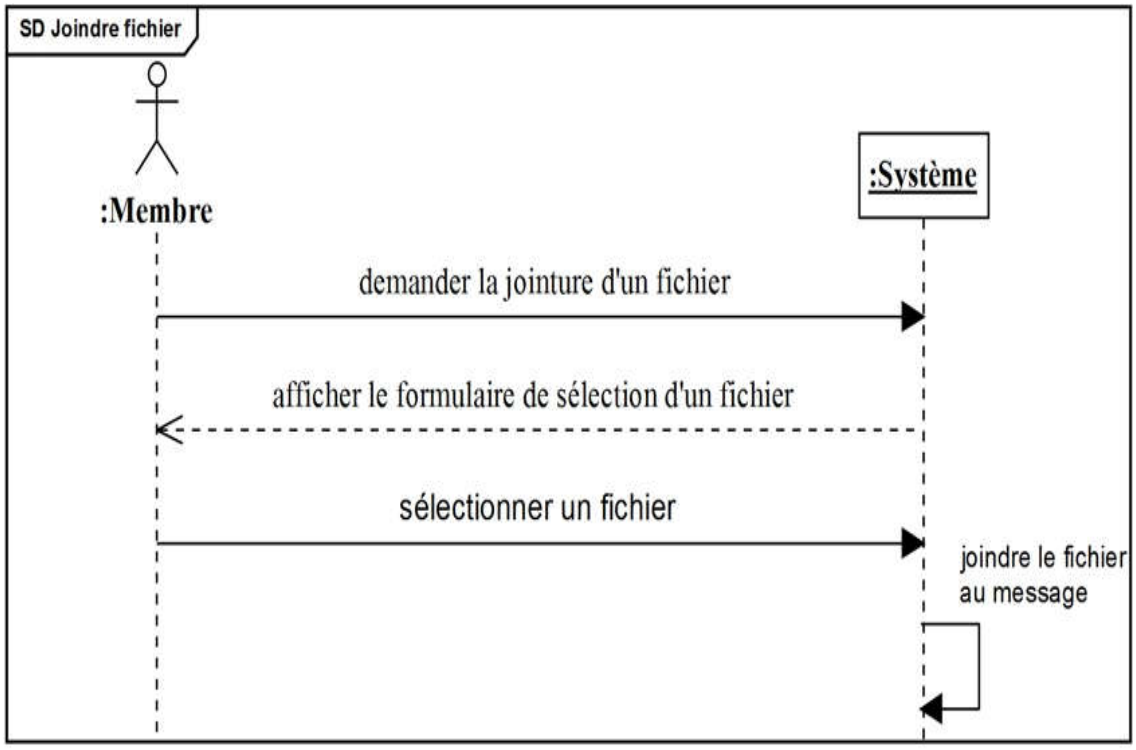

Figure 17: Diagramme de séquence « Joindre fichier ».

❖ **Cas d'utilisation « Chiffrer message »**

Cas d'utilisation	Chiffrer message
Acteur	Membre
But	Permet au membre de chiffrer le texte du message.
Pré-condition	Le membre doit être connecté.
Post-condition	Le texte du message est chiffré.
Scénario nominal	1. Le membre demande de chiffrer un message. 2. Le système vérifie l'existence de clef publique associée au destinataire. 3. Le système affiche la liste des clefs publiques du destinataire. 4. Le membre sélectionne la clef à utiliser et valide. 5. Le système chiffre le texte du message. 6. Le système affiche un message de confirmation.
Scénario alternatif	A1. Il n'existe pas des clefs publiques associées au destinataire dans la base de données. 1. Le système affiche un message d'erreur.

Tableau 15: Description textuelle du cas d'utilisation « Chiffrer message ».

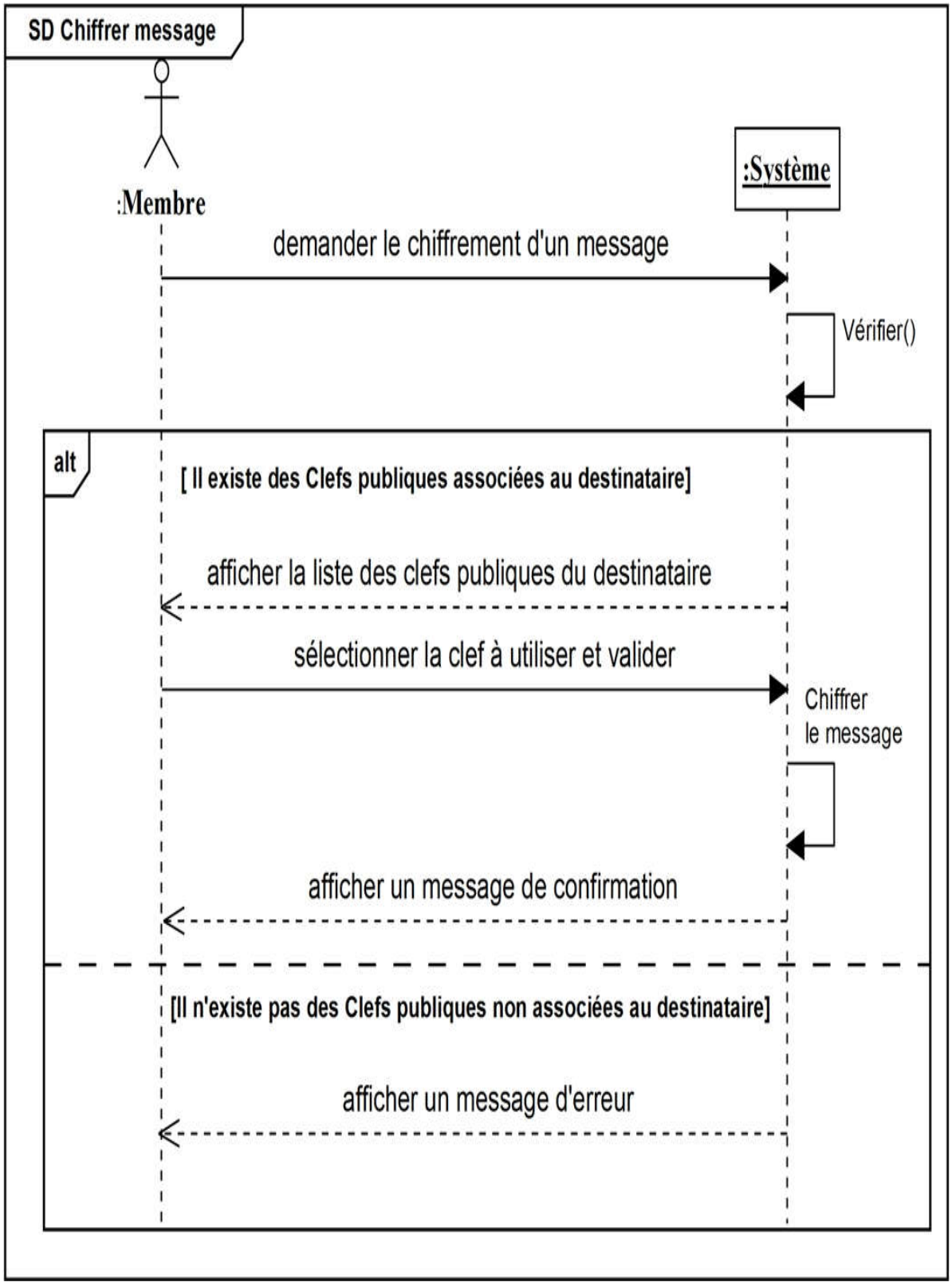

Figure 18: Diagramme de séquence « Chiffrer message ».

❖ **Cas d'utilisation « Signer message »**

Cas d'utilisation	Signer message
Acteur	Membre
But	Permet au membre de signer le texte du message.
Pré-condition	Le membre doit être connecté.
Post-condition	Le texte du message est signé.
Scénario nominal	1. Le membre demande de signer un message. 2. Le système affiche les opérations (Utiliser clefs existantes, Gérer clefs). 3. Le membre choisit l'opération Utiliser clefs existantes. 4. Le système vérifie l'existence des clefs privées associées au membre. 5. Le système affiche la liste de clefs privées du membre. 6. Le membre sélectionne une clef. 7. Le système signe le texte du message. 8. Le système affiche un message de confirmation.
Scénario alternatif	A1. Il n'existe pas de clefs privées associées au membre dans la base de données. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 de scénario nominale. A2. 1. Le membre choisit l'opération Gérer clefs. 2. Le système appelle le cas d'utilisation « Gérer clefs» 3. Le système retourne à l'étape numéro 2 du scénario nominal.

Tableau 16: Description textuelle du cas d'utilisation « Signer message ».

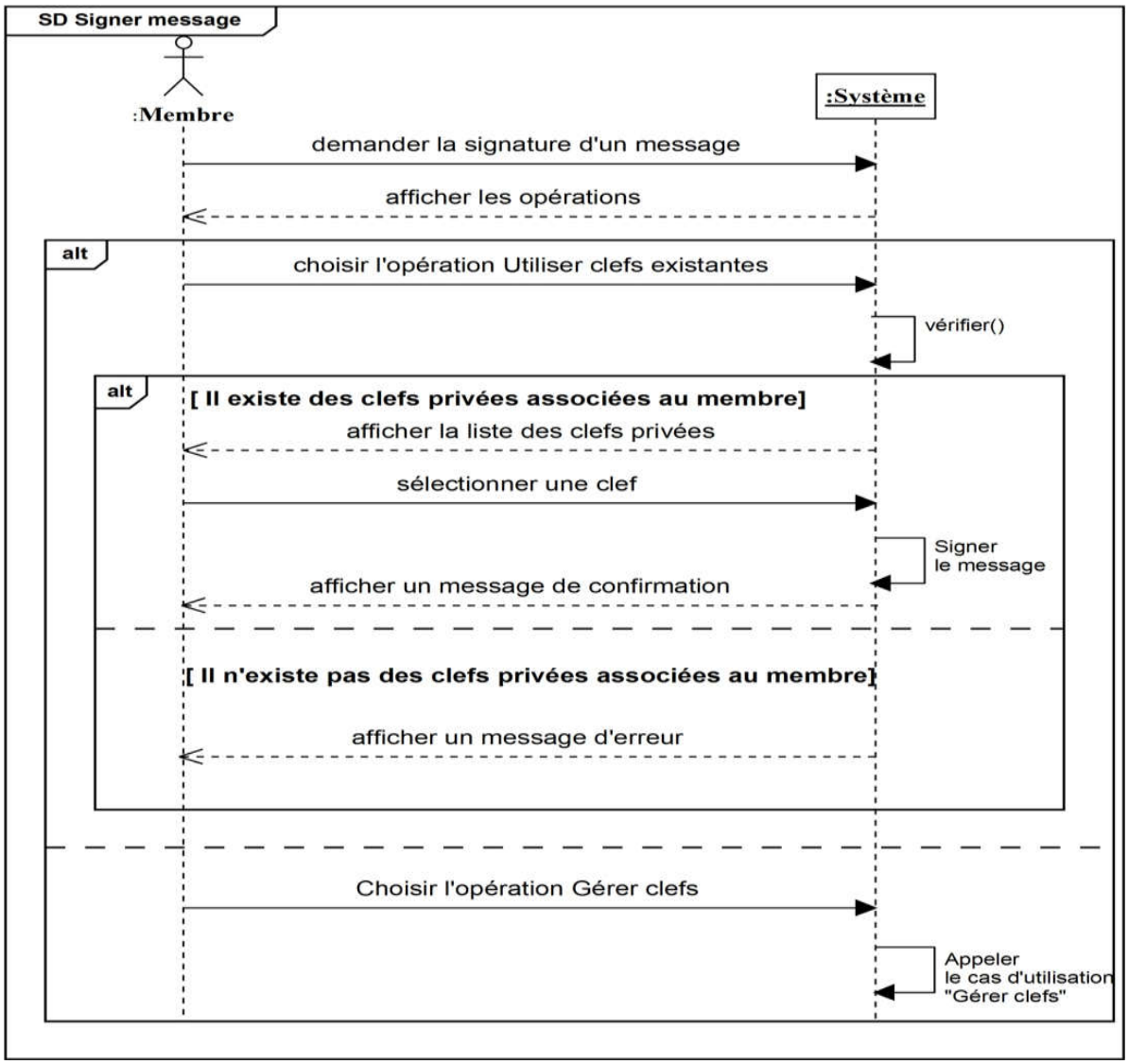

Figure 19: Diagramme de séquence « Signer message ».

❖ **Cas d'utilisation « Consulter boite de réception »**

Cas d'utilisation	Consulter boite de réception
Acteur	Membre
But	Permet au membre de consulter ses messages reçus.
Pré-condition	Le membre doit être connecté.
Post-condition	Afficher le détail des messages reçus.
Scénario nominal	1. Le membre demande de consulter sa boite de réception. 2. Le système affiche la liste des messages reçus. 3. Le membre sélectionne un message. 4. Le système affiche le détail du message. 5. Le système affiche les opérations (Vérifier Signature,

	Déchiffrer message, Télécharger pièce jointe, Transférer message, Supprimer message).
Scénario alternatif	A1. 1. Le Membre choisit l'opération « Déchiffrer message ». 2. Le système appelle le cas d'utilisation « Déchiffrer message». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A2. 1. Le Membre choisit l'opération « Vérifier signature ». 2. Le système appelle le cas d'utilisation «Vérifier signature ». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A3. 1. Le Membre choisit l'opération «Télécharger pièce jointe ». 2. Le système appelle le cas d'utilisation «Télécharger pièce jointe». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A4. 1. Le Membre choisit l'opération «Transférer message». 2. Le système appelle le cas d'utilisation « Transférer message». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A5. 1. Le membre choisit l'opération « Supprimer message ». 2. Le système appelle le cas d'utilisation « Supprimer message ». 3. Le système retourne à l'étape numéro 5 du scénario nominal.

Tableau 17: Description textuelle du cas d'utilisation « Consulter boite de réception ».

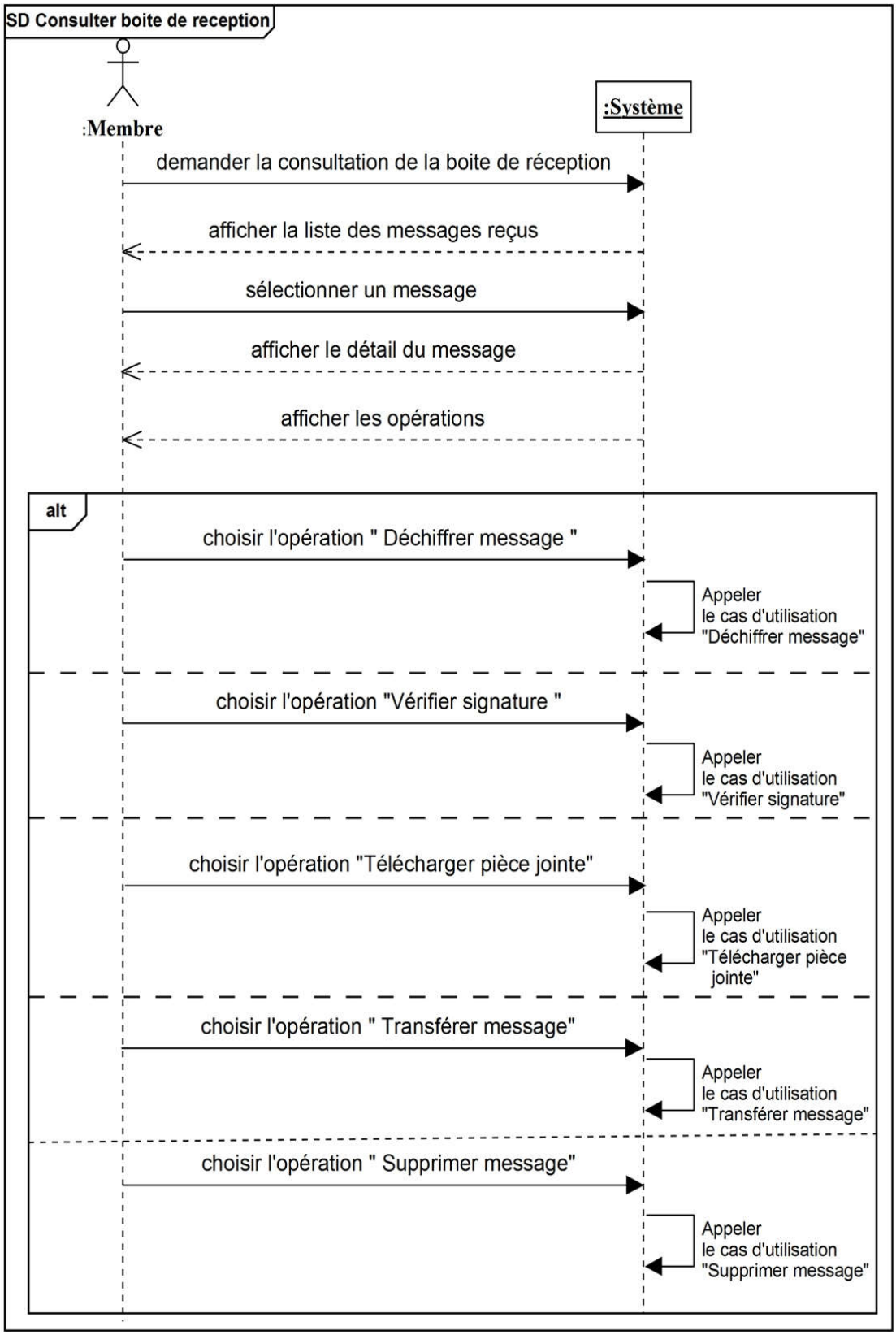

Figure 20: Diagramme de séquence « Consulter boite de réception ».

❖ **Cas d'utilisation « Consulter messages envoyés »**

Cas d'utilisation	Consulter messages envoyés
Acteur	Membre
But	Permet au membre de consulter ses messages envoyés.
Pré-condition	Le membre doit être connecté.
Post-condition	Afficher le détail des messages envoyés.
Scénario nominal	1. Le membre demande de consulter ses messages envoyés. 2. Le système affiche la liste des messages envoyés. 3. Le membre sélectionne un message. 4. Le système affiche le détail du message. 5. Le système affiche les opérations (Télécharger pièce jointe, Transférer message, Supprimer message).
Scénario alternatif	A1. 1. Le membre choisit l'opération « Télécharger pièce jointe ». 2. Le système appelle le cas d'utilisation « Télécharger pièce jointe ». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A2. 1. Le membre choisit l'opération « Transférer message». 2. Le système appelle le cas d'utilisation « Transférer message». 3. Le système retourne à l'étape numéro 5 du scénario nominal. A3. 1. Le membre choisit l'opération « Supprimer message». 2. Le système appelle le cas d'utilisation « Supprimer message» 3. Le système retourne à l'étape numéro 5 du scénario nominal.

Tableau 18: Description textuelle du cas d'utilisation « Consulter messages envoyés ».

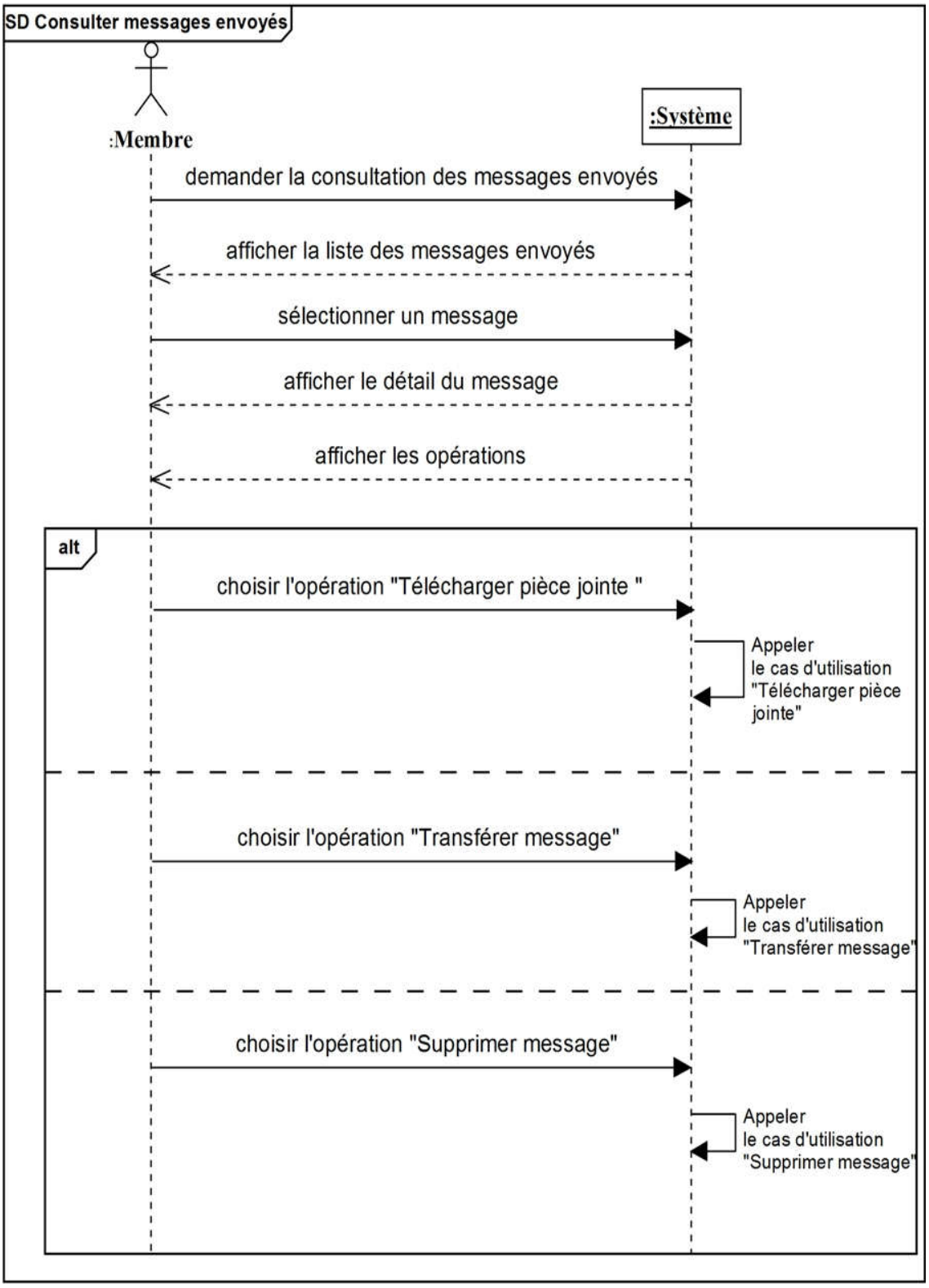

Figure 21: Diagramme de séquence « Consulter messages envoyés ».

❖ **Cas d'utilisation « Consulter brouillon »**

Cas d'utilisation	Consulter brouillon
Acteur	Membre
But	Permet au membre de consulter ses messages enregistrés dans le brouillon.
Pré-condition	Le membre doit être connecté.
Post-condition	Afficher le détail des messages enregistrés dans le brouillon.
Scénario nominal	1. Le membre demande de consulter son brouillon. 2. Le système affiche la liste des messages enregistrés dans le brouillon. 3. Le membre sélectionne un message. 4. Le système affiche le détail du message. 5. Le système affiche les opérations (Envoyer message, Supprimer message).
Scénario alternatif	A1. 1. Le membre choisit l'opération Envoyer message. 2. Le système vérifie les informations entrées. 3. Le système enregistre le message dans la base de données. 4. Le système affiche un message de confirmation. A2. Le membre n'a pas remplit certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal. A3. 1. Le membre choisit l'opération « Supprimer message». 2. Le système appelle le cas d'utilisation « Supprimer message». 3. Le système retourne à l'étape numéro 5 du scénario nominal.

Tableau 19: Description textuelle du cas d'utilisation « Consulter brouillon ».

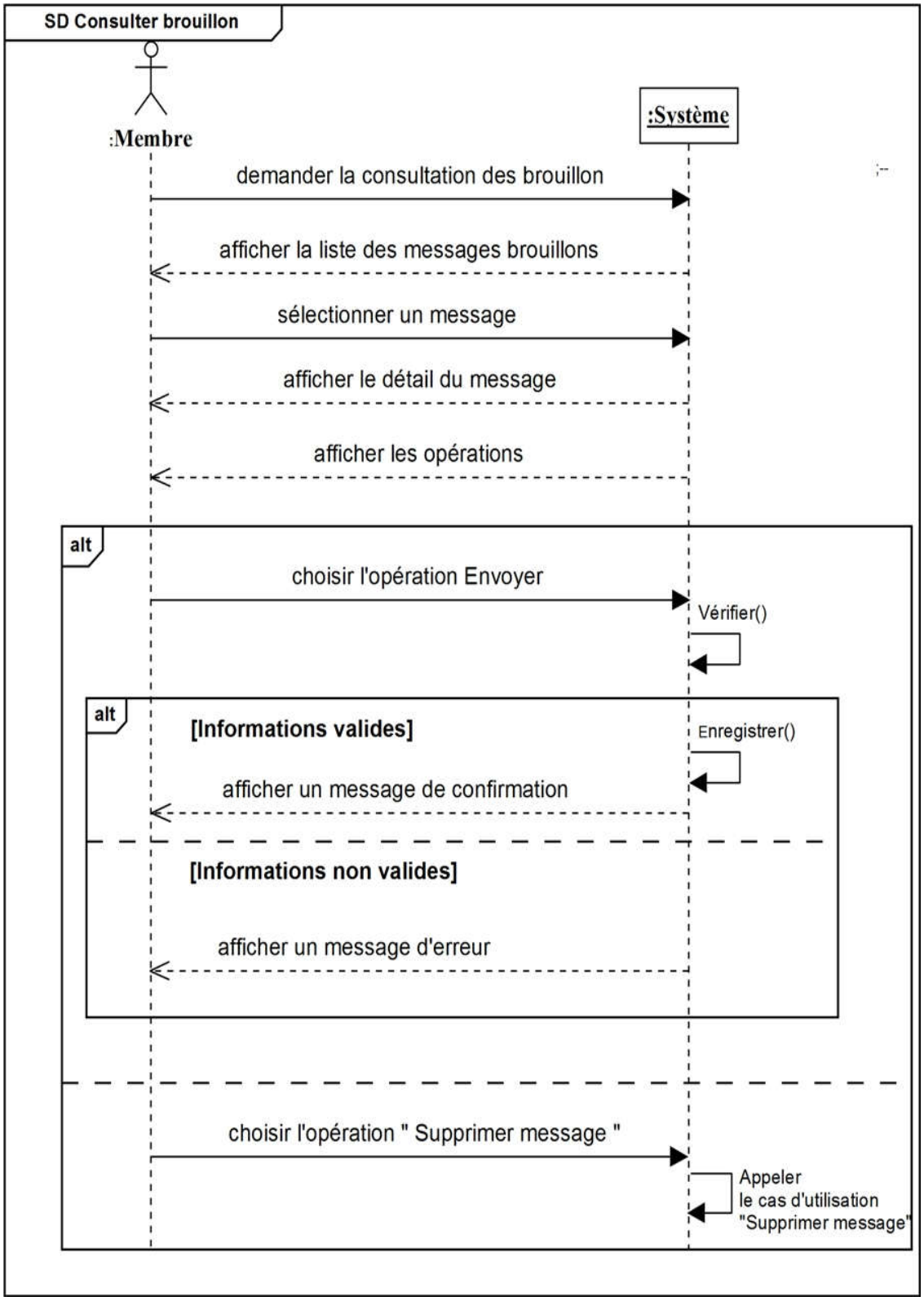

Figure 22: Diagramme de séquence « Consulter brouillon ».

❖ **Cas d'utilisation « Déchiffrer message »**

Cas d'utilisation	Déchiffrer message
Acteur	Membre
But	Permet au membre de déchiffrer le texte d'un message reçu.
Pré-condition	Le membre doit être connecté.
Post-condition	Le texte du message est affiché en clair.
Scénario nominal	1. Le membre demande de déchiffrer un message. 2. Le système vérifie l'existence d'une clef privée correspondante associée au membre. 3. Le système demande d'entrer la phrase secrète. 4. Le membre saisit la phrase secrète. 5. Le système vérifie l'information entrée. 6. Le système affiche le texte du message en clair.
Scénario alternatif	A1. Il n'existe pas une clef privée correspondante associée au membre dans la base de données. 1. Le système affiche un message d'erreur. 2. Le système affiche le message chiffré. A2. Le membre saisit une phrase secrète non valide. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 3 du scénario nominal.

Tableau 20: Description textuelle du cas d'utilisation « Déchiffrer message ».

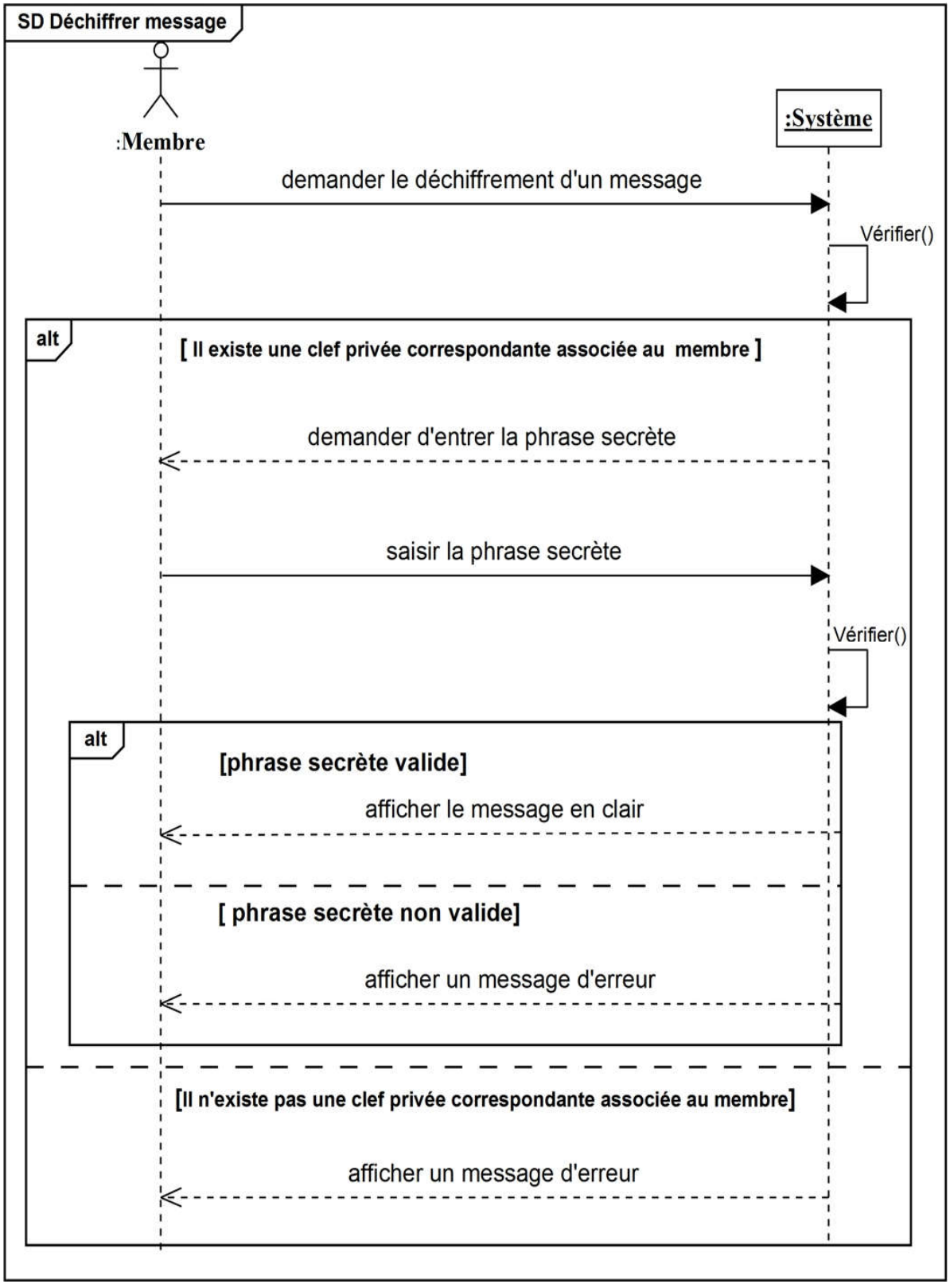

Figure 23: Diagramme de séquence « Déchiffrer message ».

❖ **Cas d'utilisation « Vérifier Signature »**

Cas d'utilisation	Vérifier Signature
Acteur	Membre
But	Permet au membre de vérifier la signature de l'émetteur du message reçu.
Pré-condition	Le membre doit être connecté.
Post-condition	La signature est vérifiée.
Scénario nominal	1. Le membre demande de vérifier la signature du message. 2. Le système vérifie l'existence d'une clef publique correspondante associée à l'émetteur. 3. Le système vérifie la signature. 4. Le système affiche un message de confirmation.
Scénario alternatif	A1. Il n'existe pas une clef publique correspondante associée à l'émetteur dans la base de données. 1. Le système affiche un message d'erreur. 2. Le système affiche le message signé. A2. Le système détecte une erreur de la signature. 1. Le système affiche un message d'erreur. 2. Le système affiche le message signé.

Tableau 21: Description textuelle du cas d'utilisation « Vérifier signature ».

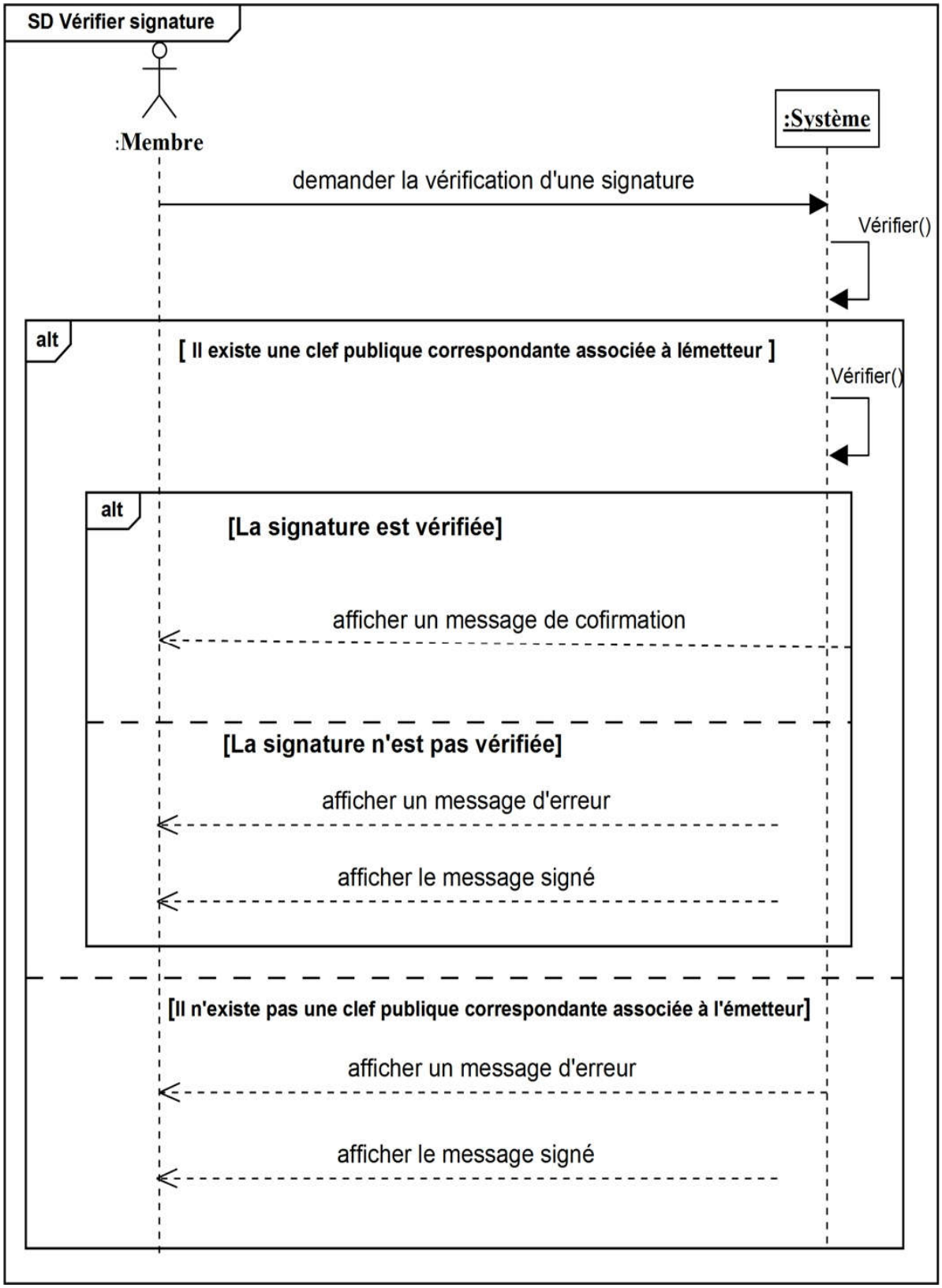

Figure 24: Diagramme de séquence « Vérifier signature ».

❖ **Cas d'utilisation « Télécharger pièce jointe »**

Cas d'utilisation	Télécharger pièce jointe
Acteur	Membre
But	Permet au membre de télécharger une pièce jointe au message.
Pré-condition	Le membre doit être connecté.
Post-condition	La pièce jointe au message est téléchargée.
Scénario nominal	1. Le membre demande de télécharger une pièce jointe. 2. Le système affiche la liste des pièces jointes. 3. Le membre sélectionne une pièce jointe. 4. Le système demande de sélectionner un dossier de sauvegarde. 5. Le membre sélectionne un dossier. 6. Le système enregistre la pièce jointe dans le dossier sélectionné. 7. Le système affiche un message de confirmation.

Tableau 22: Description textuelle du cas d'utilisation « Télécharger pièce jointe ».

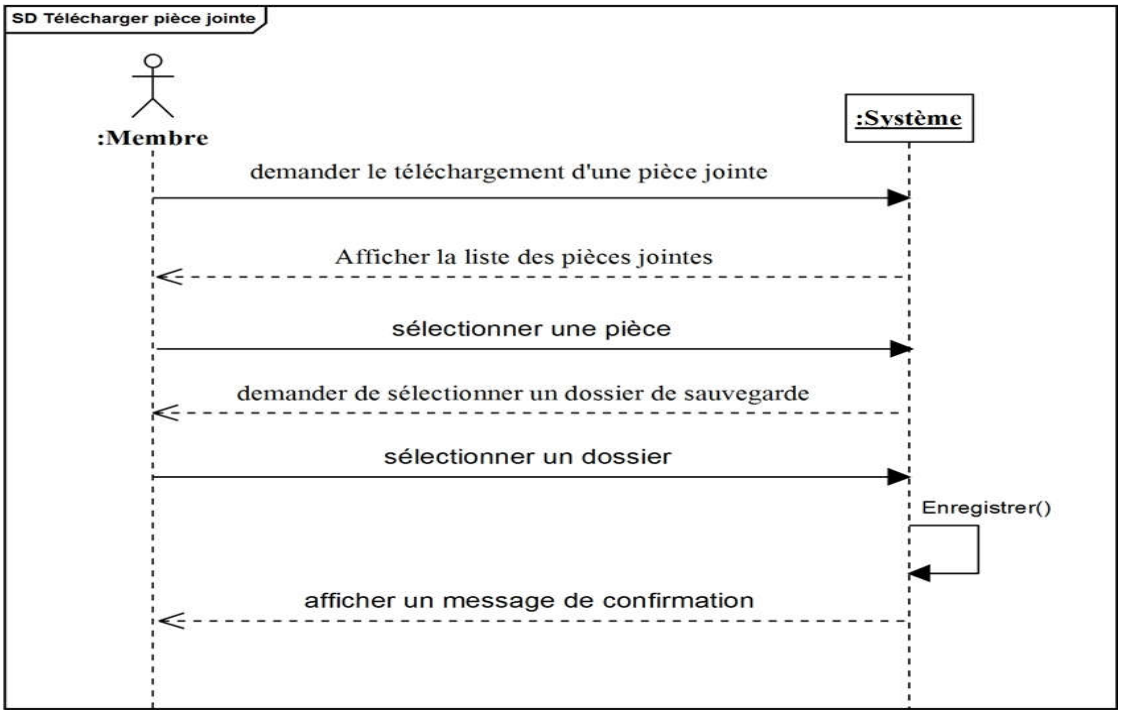

Figure 25: Diagramme de séquence « Télécharger pièce jointe ».

❖ **Cas d'utilisation « Transférer message »**

Cas d'utilisation	Transférer message
Acteur	Membre
But	Permet au membre de transférer un message.
Pré-condition	Le membre doit être connecté.
Post-condition	Mise à jour de la base de données.
Scénario nominal	1. Le membre demande de transférer un message. 2. Le système demande de saisir les adresses des destinataires et valider l'envoie du message. 3. Le membre entre les adresses et valide. 4. Le système vérifie les informations entrées. 5. Le système enregistre le message dans la base de données. 6. Le système affiche un message de confirmation.
Scénario alternatif	A1. Le membre n'a pas remplit certains champs ou a saisit des informations non valides. 1. Le système affiche un message d'erreur. 2. Le système retourne à l'étape numéro 2 du scénario nominal.

Tableau 23: Description textuelle du cas d'utilisation « Transférer message ».

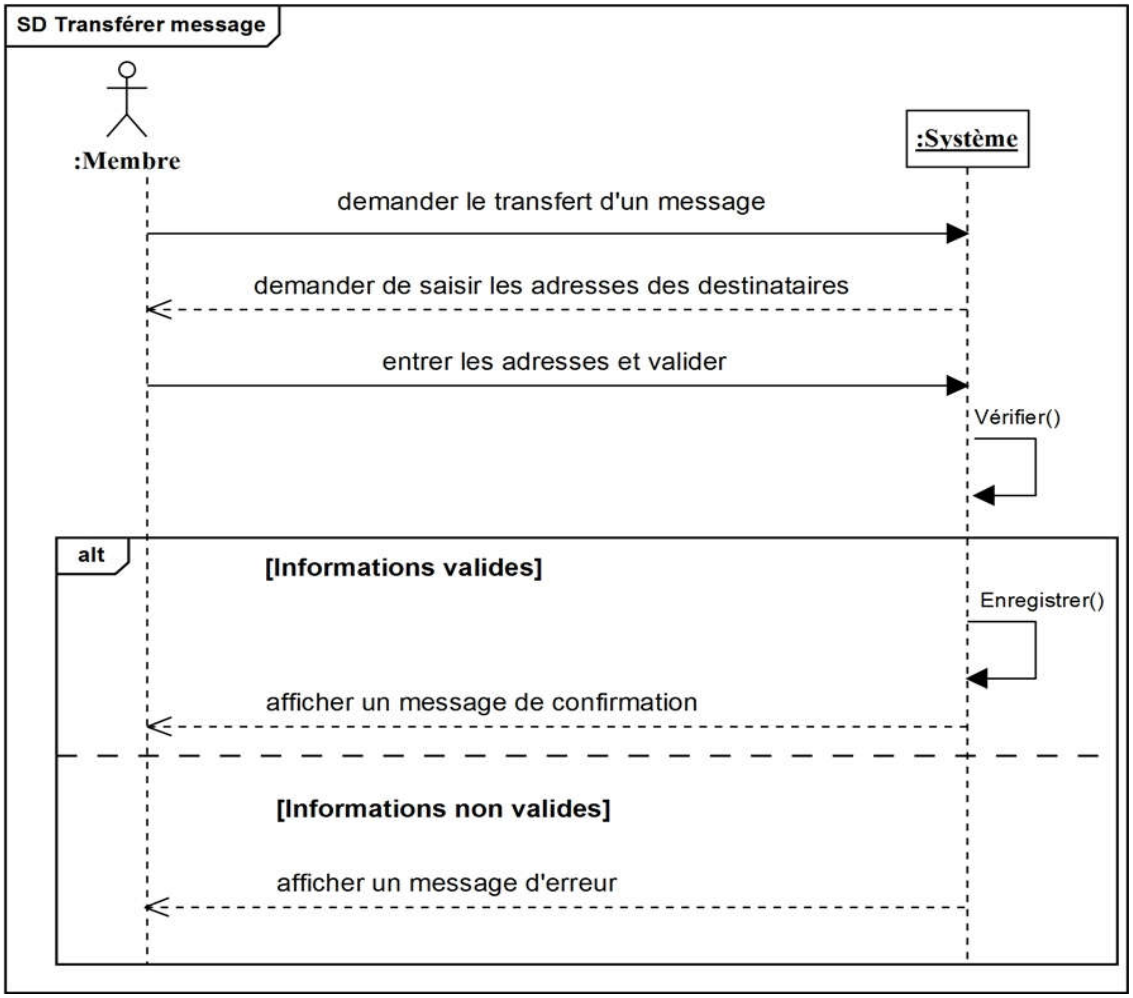

Figure 26: Diagramme de séquence « Transférer message ».

❖ **Cas d'utilisation « Supprimer message »**

Cas d'utilisation	Supprimer message
Acteur	Membre
But	Permet au membre de supprimer un message envoyé, reçu, ou bien enregistré comme brouillon.
Pré-condition	Le membre doit être connecté.
Post-condition	Mise à jour de la base de données.
Scénario nominal	1. Le membre demande de supprimer un message. 2. Le système met à jour la base de données. 3. Le système affiche un message de confirmation.

Tableau 24: Description textuelle du cas d'utilisation « Supprimer message ».

Figure 27: Diagramme de séquence « Supprimer message ».

❖ **Cas d'utilisation « Se déconnecter »**

Cas d'utilisation	Se déconnecter
Acteur	Membre
But	Permet au membre de sortir de sa session.
Pré-condition	Le membre doit être déjà connecté.
Post-condition	La session du membre est fermée.
Scénario nominal	1. Le membre demande de se déconnecter du système de messagerie électronique. 2. Le système ferme la session du Membre.

Tableau 25: Description textuelle du cas d'utilisation « Se déconnecter ».

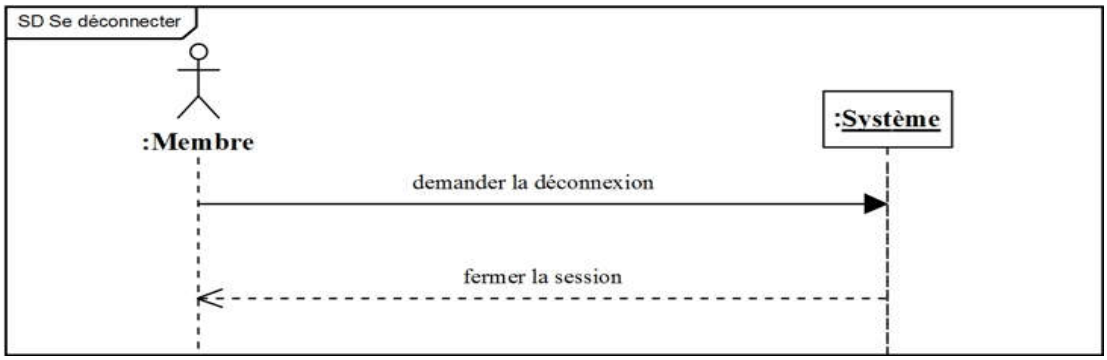

Figure 28: Diagramme de séquence « Se déconnecter ».

3.2. ANALYSE DU DOMAINE

3.2.1. Identification des concepts du domaine

Nous allons prendre les cas d'utilisations un par un et nous poser pour chacun la question suivante : quel sont les concepts métier qui participent à ce cas d'utilisation ?

- ✓ **S'inscrire** : utilisateur.
- ✓ **Se connecter** : membre.
- ✓ **Consulter boite de réception** : membre.
- ✓ **Consulter messages envoyés** : membre.
- ✓ **Consulter brouillon** : membre.
- ✓ **Composer un message** : membre.
- ✓ **Gérer clefs** : membre.
- ✓ **Gérer contactes** : membre.
- ✓ **Mettre à jour compte** : membre.
- ✓ **Se déconnecter**: membre.

3.2.2. Modèle de domaine

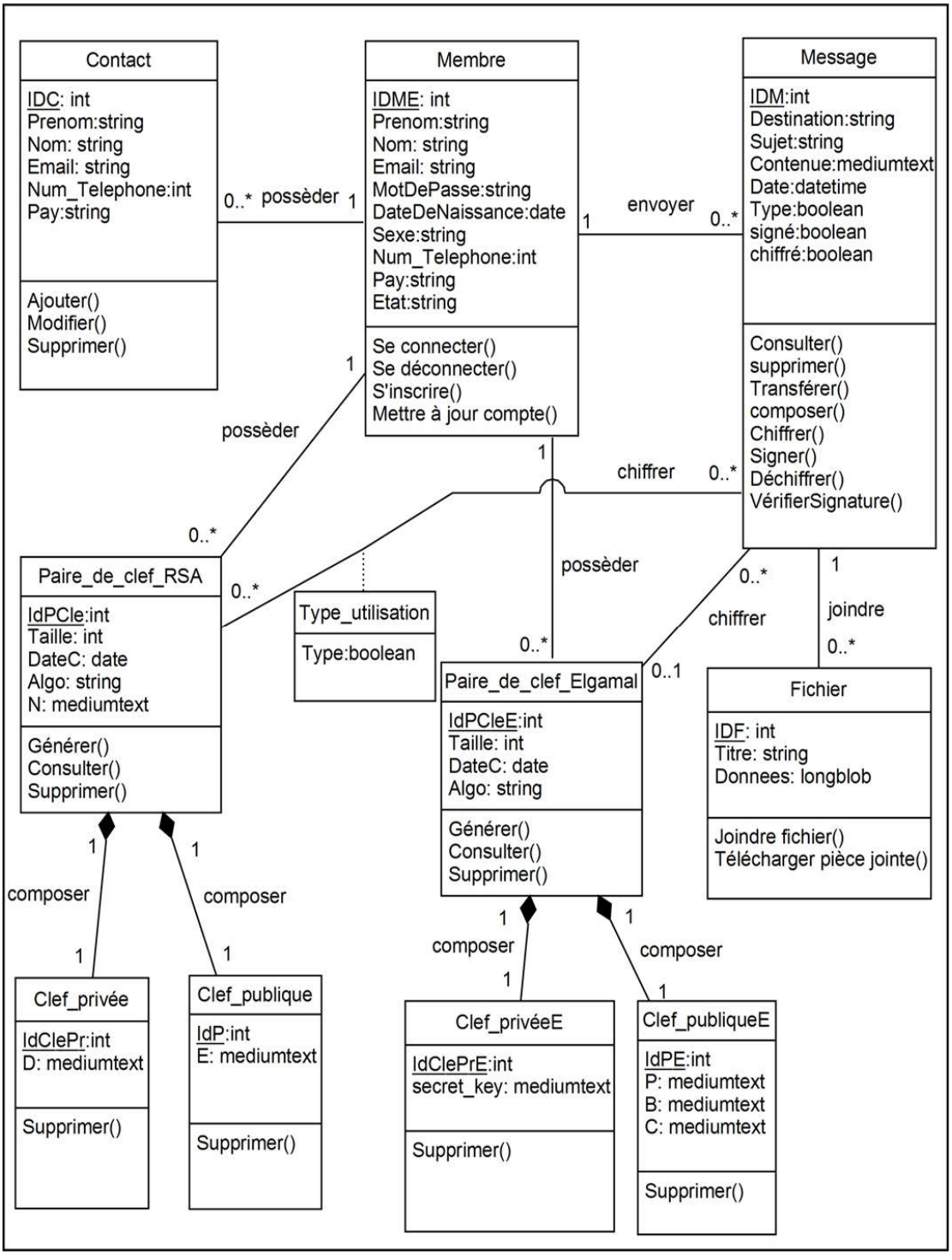

Figure 29: Modèle de domaine.

3.3. PHASE DE CONCEPTION

3.3.1. Diagrammes d'interactions

- **Cas d'utilisation « Se connecter »**

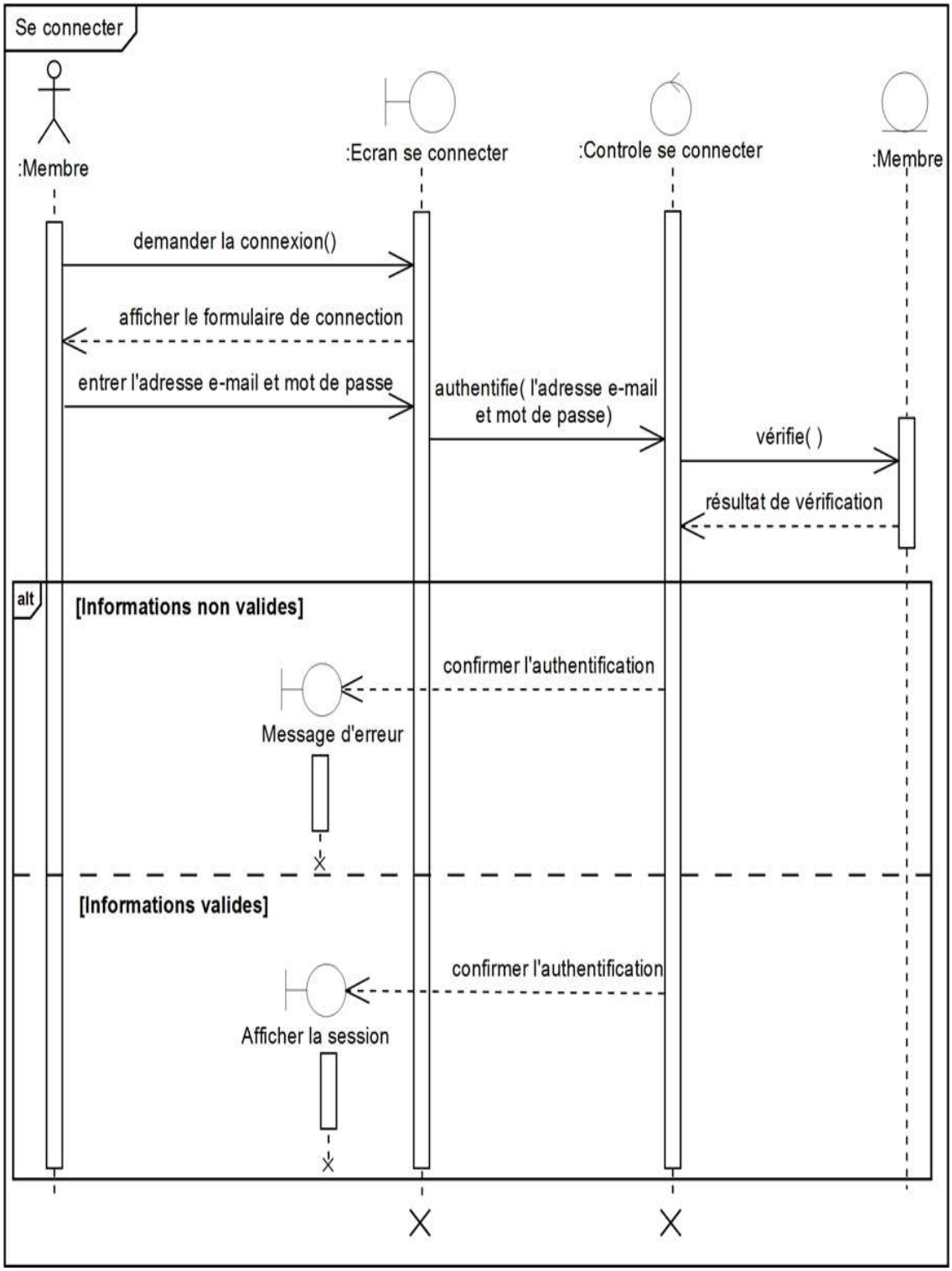

Figure 30: Diagramme d'interaction « Se connecter ».

❖ **Cas d'utilisation « Gérer clefs »**

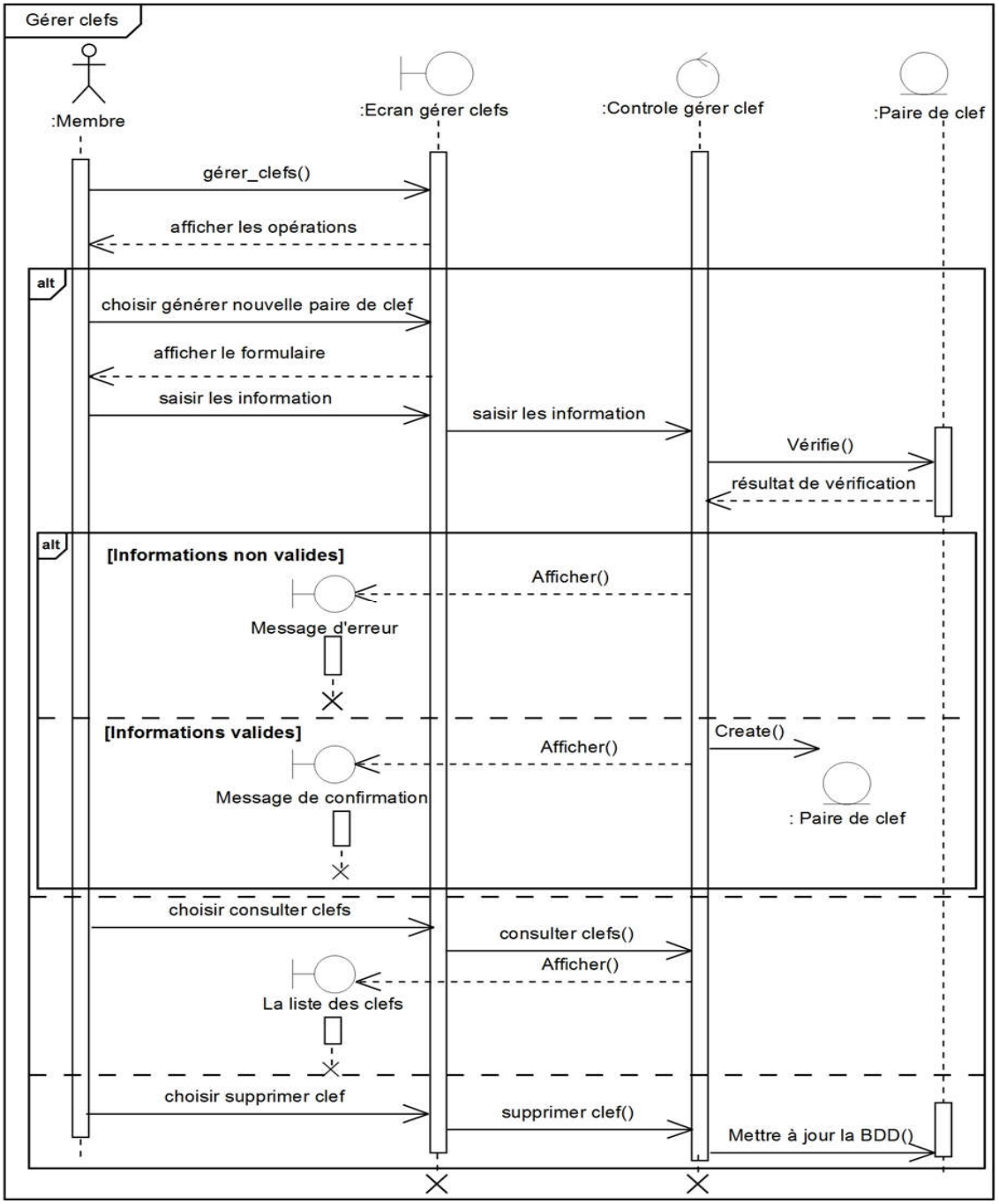

Figure 31: Diagramme d'interaction « Gérer clefs ».

❖ **Cas d'utilisation « Chiffrer message »**

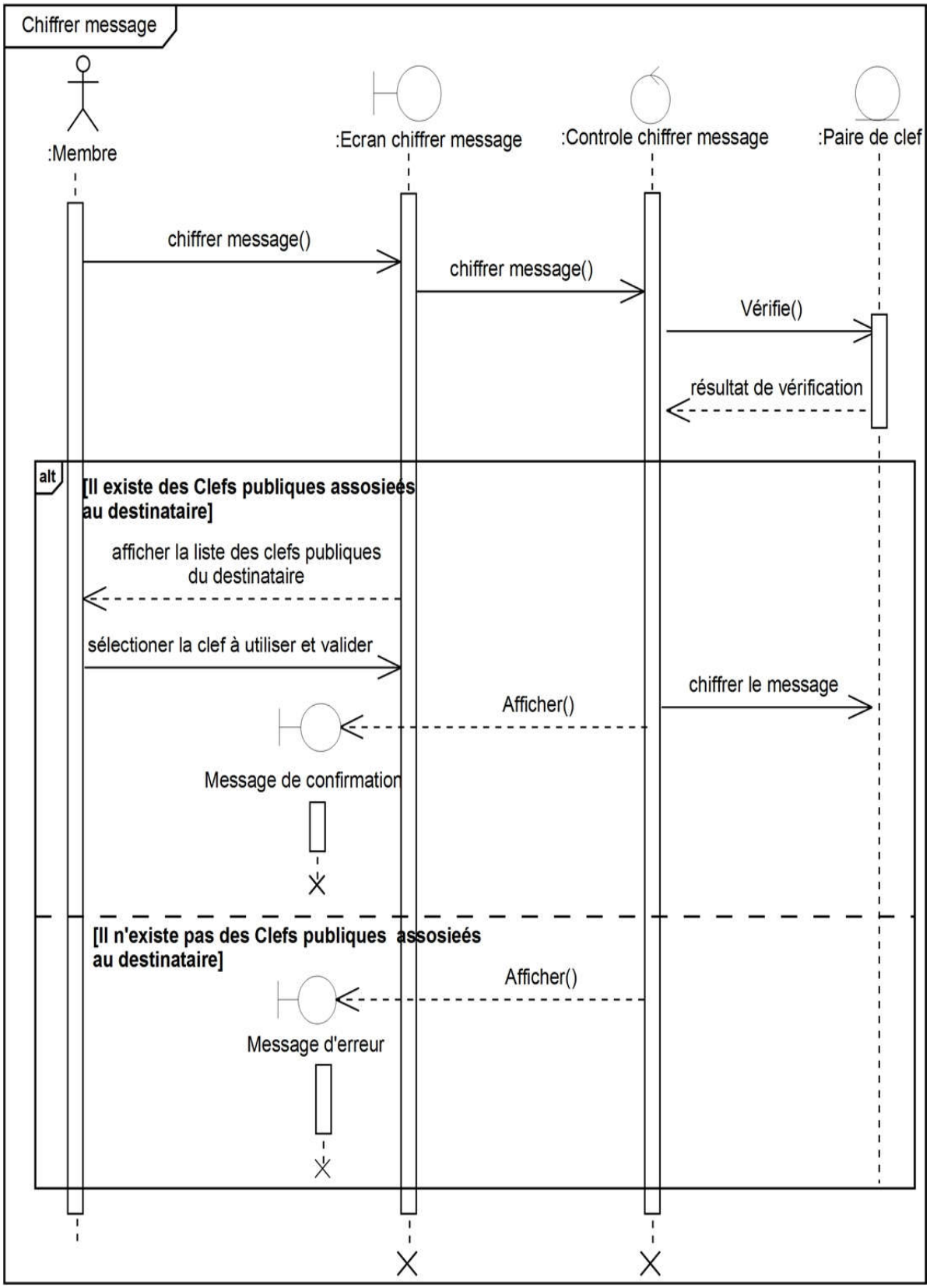

Figure 32: Diagramme d'interaction « Chiffrer message ».

❖ **Cas d'utilisation « Déchiffrer message »**

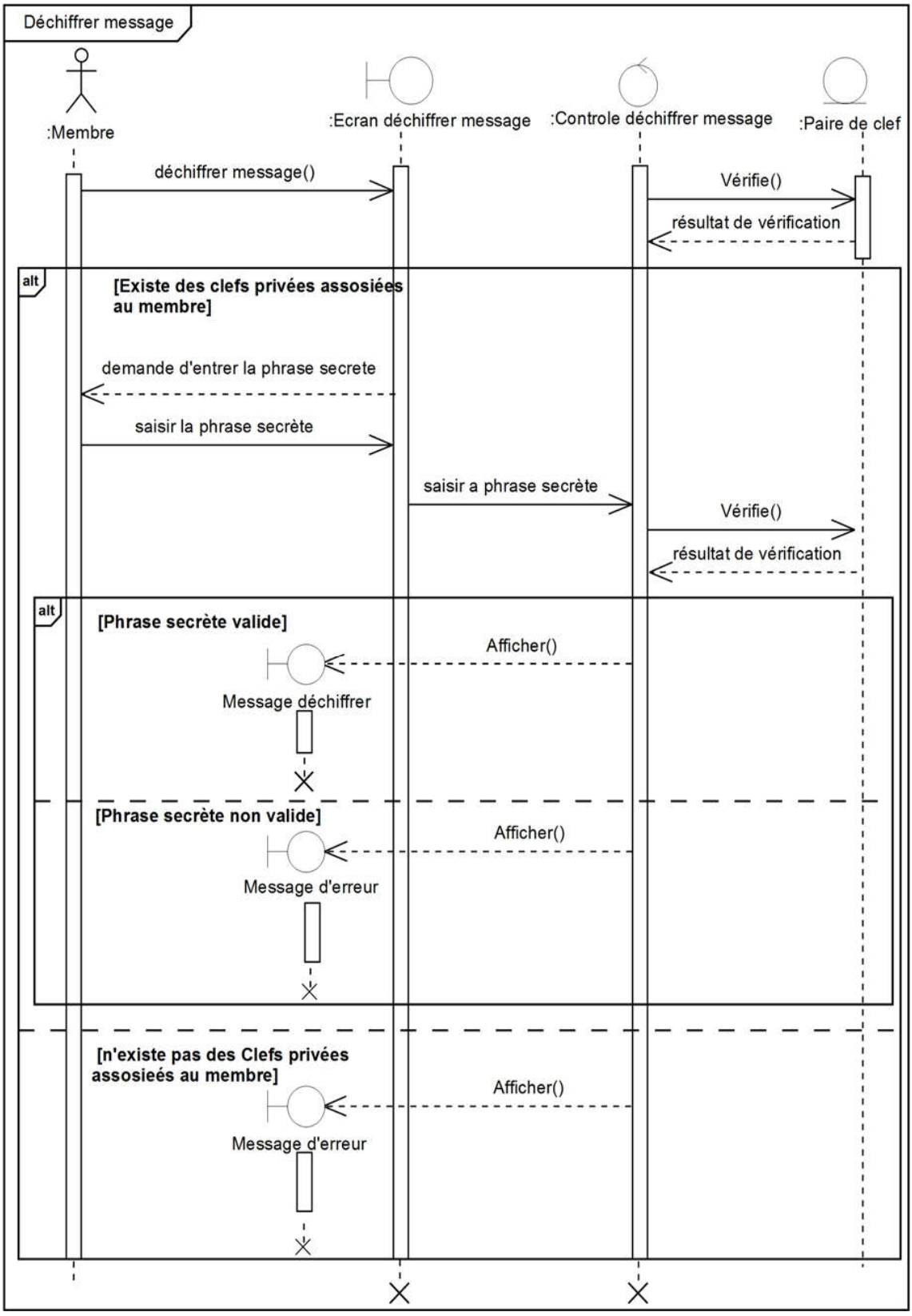

Figure 33: Diagramme d'interaction « Déchiffrer message ».

3.3.2. Développement du modèle de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système, le déploiement d'une solution client/serveur se construit sur la définition des postes de travail.

3.3.2.1. Architecture adoptée

Le choix de notre solution s'est porté sur une architecture Client/serveur 3-tiers dans laquelle plusieurs membres utilisant chacun une application client lourd, sont connectés à une application serveur qui est connectée à son tour à un serveur de base de données. Chaque client, demandeur de ressources, est équipé d'une interface utilisateur chargée de la présentation. Le serveur d'application est chargé de fournir la ressource au client mais pour cela il fait appel au serveur de base de données qui lui fournit les données dont il a besoin. En fin le serveur d'application fournit au client le service demandé.

La communication entre le serveur d'application et les clients est organisée par des threads, des sockets, et des numéros de ports.

- **Thread**

Un thread est une unité d'exécution faisant partie d'un programme. Cette unité fonctionne de façon autonome et parallèlement à d'autres threads. En fait, sur une machine mono processeur, chaque unité se voit attribuer des intervalles de temps au cours desquels elles ont le droit d'utiliser le processeur pour accomplir leurs traitements. Le principal avantage des threads est de pouvoir répartir différents traitements d'un même programme en plusieurs unités distinctes pour permettre leur exécution "simultanée" [55].

- **Les numéros de port**

Un client a besoin de deux informations pour trouver et se connecter à un serveur: un nom de machine et un numéro de port. Le numéro de port est un identificateur qui permet de distinguer les différents clients et les serveurs qui fonctionnent sur une même machine [56].

- **Socket**

Un socket représente une prise par laquelle une application peut envoyer et recevoir des données. Les sockets servent à communiquer entre deux hôtes appelés Client / Serveur à l'aide d'une adresse IP et d'un port, ces sockets permettront de gérer des flux entrant et sortant afin d'assurer une communication entre les deux (le client et le serveur), soit de manière fiable à l'aide du protocole TCP/IP, soit non fiable mais plus rapide avec le protocole UDP [57].

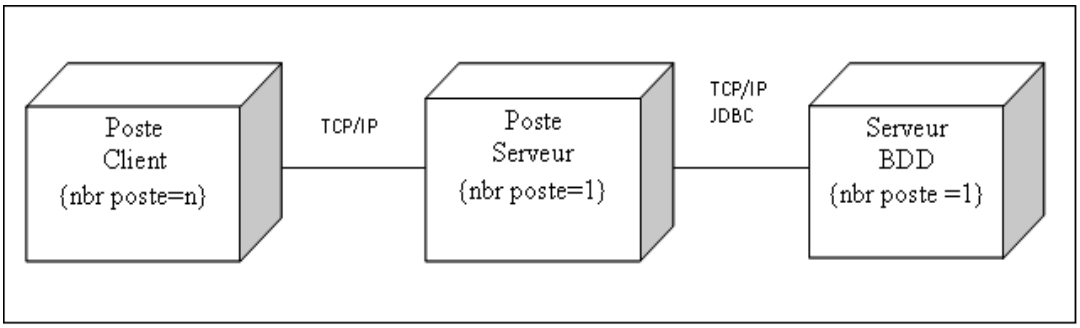

Figure 34: Schéma du modèle de déploiement de notre système.

3.3.2.2. Déploiement du modèle d'exploitation

Par le biais du modèle d'exploitation, nous définissons les applications installées sur les postes de travail, les composants métier déployés sur les serveurs et les instances de base de données implantées sur les serveurs également.

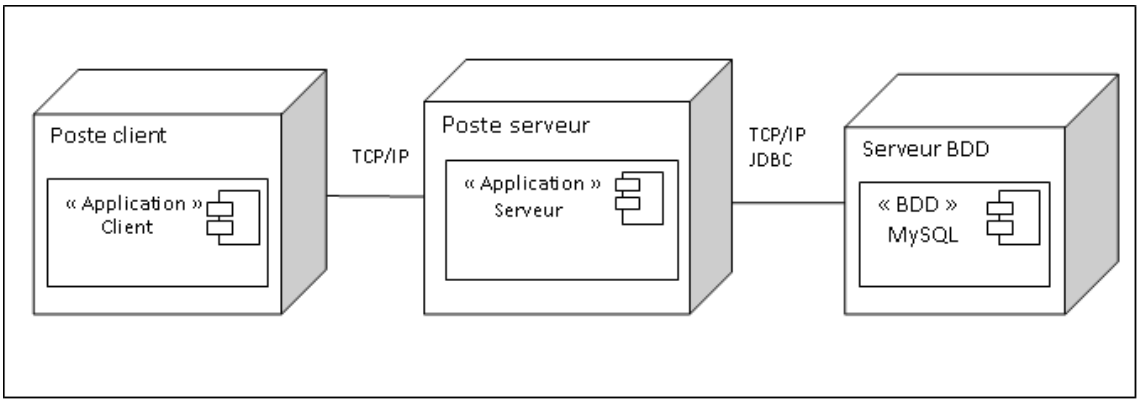

Figure 35: Définition des applications dans le modèle d'exploitation.

4. CONCLUSION

Dans ce chapitre nous avons détaillé l'analyse et la conception de notre application. Nous avons choisie une démarche simplifiée basée sur le processus UP et le langage de modélisation UML comme une méthode de conception d'application. Nous avons représenté les diagrammes de cas d'utilisation, les diagrammes de séquence, les diagrammes d'interaction, le modèle de domaine et enfin le modèle de déploiement.

Dans le prochain chapitre nous allons présenter l'implémentation de notre application.

CHAPITRE 04 :
IMPLÉMENTATION

1. INTRODUCTION

Après la phase de conception de l'application, nous allons passer à l'étape finale dans ce projet. Nous présentons, dans ce chapitre, la partie réalisation et mise en œuvre de notre travail. Pour cela, nous présentons, en premier lieu, l'environnement de travail et les outils de développement utilisés. En second lieu, nous élaborons une présentation de quelques interfaces de notre système.

2. ENVIRONNEMENT DE TRAVAIL

Pour implémenter notre système nous avons utilisée le langage de programmation Java avec l'IDE NetBeans. Pour implémenter notre base de données nous avons utilisée le SGBD MySQL.

2.1. LANGAGE DE PROGRAMMATION JAVA

Java est le nom d'une technologie mise au point par Sun Microsystems qui permet de produire des logiciels indépendants de toute architecture matérielle. Java est à la fois un langage de programmation et une plateforme d'exécution.

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du langage C. Il est notamment largement utilisé pour le développement d'applications d'entreprises et mobiles. Le langage Java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation tels que Windows et Linux. C'est la plateforme qui garantit la portabilité des applications développées en Java. Le choix de JAVA pour le développement de notre application est dû à ses caractéristiques suivantes :

- ✓ Utilise le concept orienté objet.
- ✓ Permet la création des interfaces graphiques (menues, boutons, cases à cocher) essentiels pour la conception de l'interface graphique de notre application.
- ✓ Est facile à utiliser et possède les points forts des langages de programmation orientés objet comme C++.
- ✓ Il est multitâche.
- ✓ Il est simple [58].

2.2. IDE NETBEANS

NetBeans est un environnement de développement intégré (IDE) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL (Common Developpment and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages web).

NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X et Open VMS. NetBeans est lui-même développé en Java, ce qui peut le rendre assez lent et gourmand en ressources mémoires [59].

2.3. SGBDR(MYSQL)

MySQL est l'un des systèmes de gestion de base de données (SGBD) les plus utilisés au monde, autant par le grand public (application Web principalement) que par les professionnels MySQL est un SGBD multi-utilisateurs, qu'il fonctionne sur de nombreux systèmes d'exploitation différents, incluant Linux, Mac OS X, Solaris, Windows 9x, NT, XP et Vista. MySQL est très facile à administrer. Il prend en charge les API (Application Programming Interface) clients de nombreux langages de programmation (par exemple java) ce qui permet d'écrire facilement les programmes clients qui doivent accéder aux données d'une base MySQL.

2.4. PHPMYADMIN

PHPMyAdmin est une application graphique qui peut gérer un serveur MySQL. Il permet d'administrer les éléments suivants :

- Les bases de données.
- Les tables et leurs champs (ajout, suppression, définition du type).
- Les index, les clés primaires et étrangères.
- Les utilisateurs de la base et leurs permissions.
- Exporter les données dans divers formats (CSV, XML, PDF, OpenDocument, Word, Excel et LaTeX).

3. QUELQUES INTERFACES DE L'APPLICATION

3.1. FENETRE D'ACCUEIL

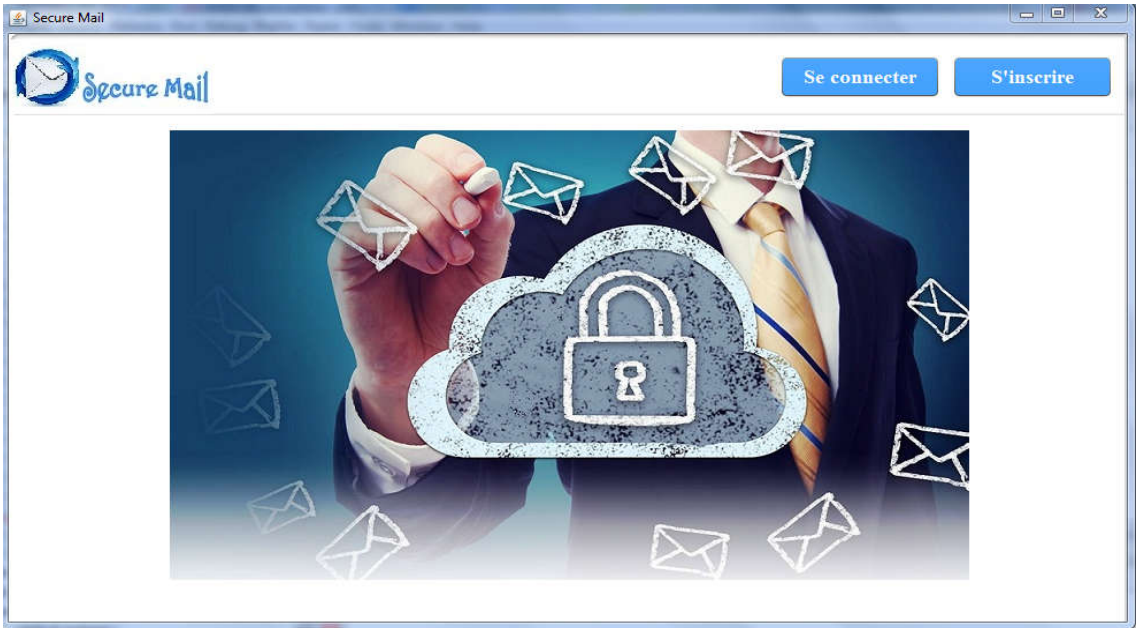

Figure 36: Fenêtre d'accueil

3.2. FENETRE D'INSCRIPTION

Figure 37: Fenêtre d'inscription

3.3. FENETRE DE CONNEXION

Figure 38: Fenêtre de connexion.

3.4. FENETRE D'AJOUT D'UN NOUVEAU CONTACT

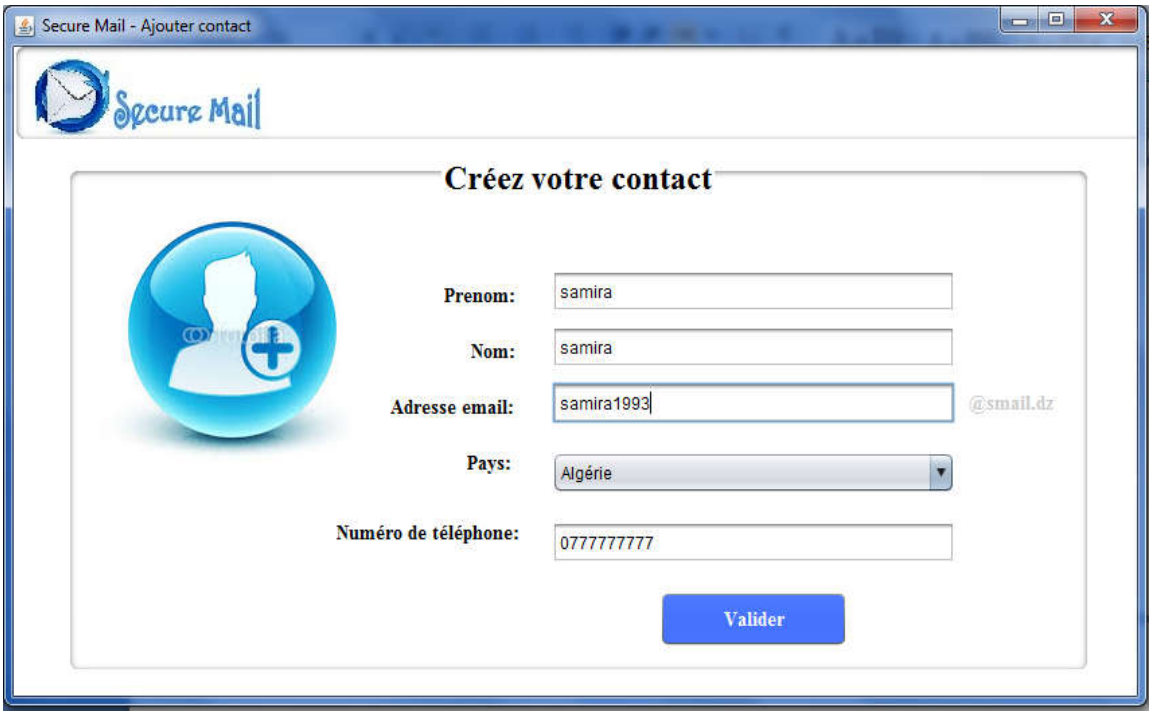

Figure 39: Fenêtre d'ajout d'un nouveau contact.

3.5. FENETRE DE CREATION D'UNE NOUVELLE PAIRE DE CLEF RSA

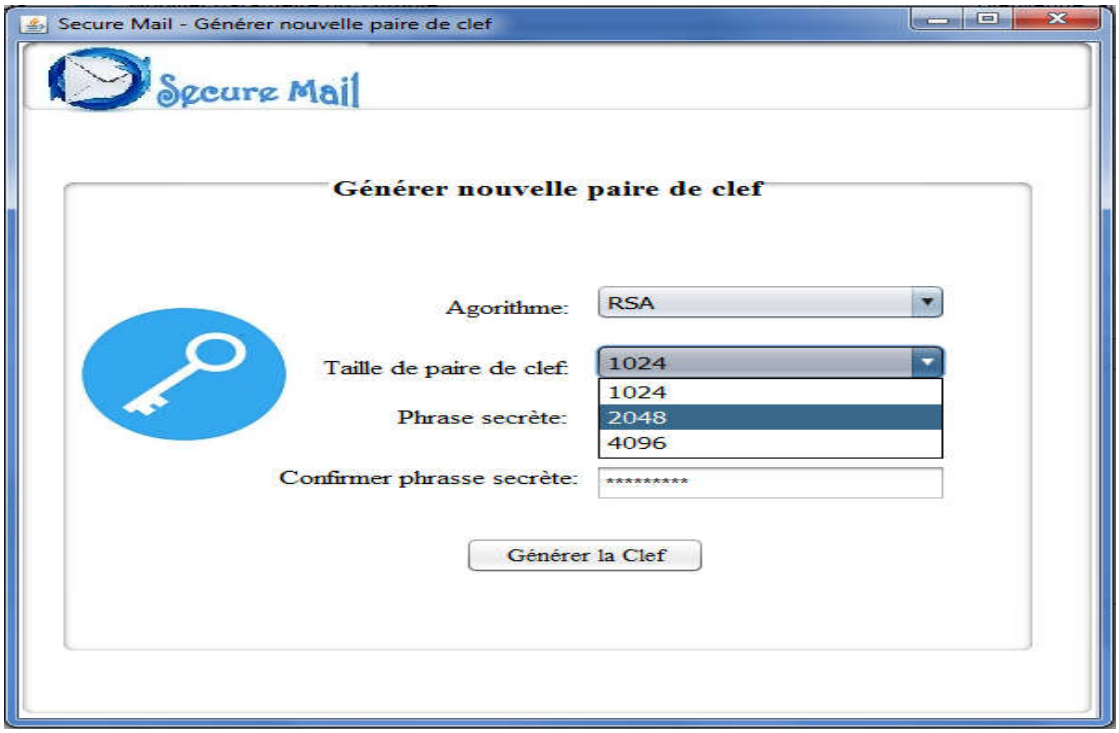

Figure 40: Fenêtre de création d'une nouvelle paire de clef RSA.

3.6. FENETRE DE COMPOSITION D'UN MESSAGE

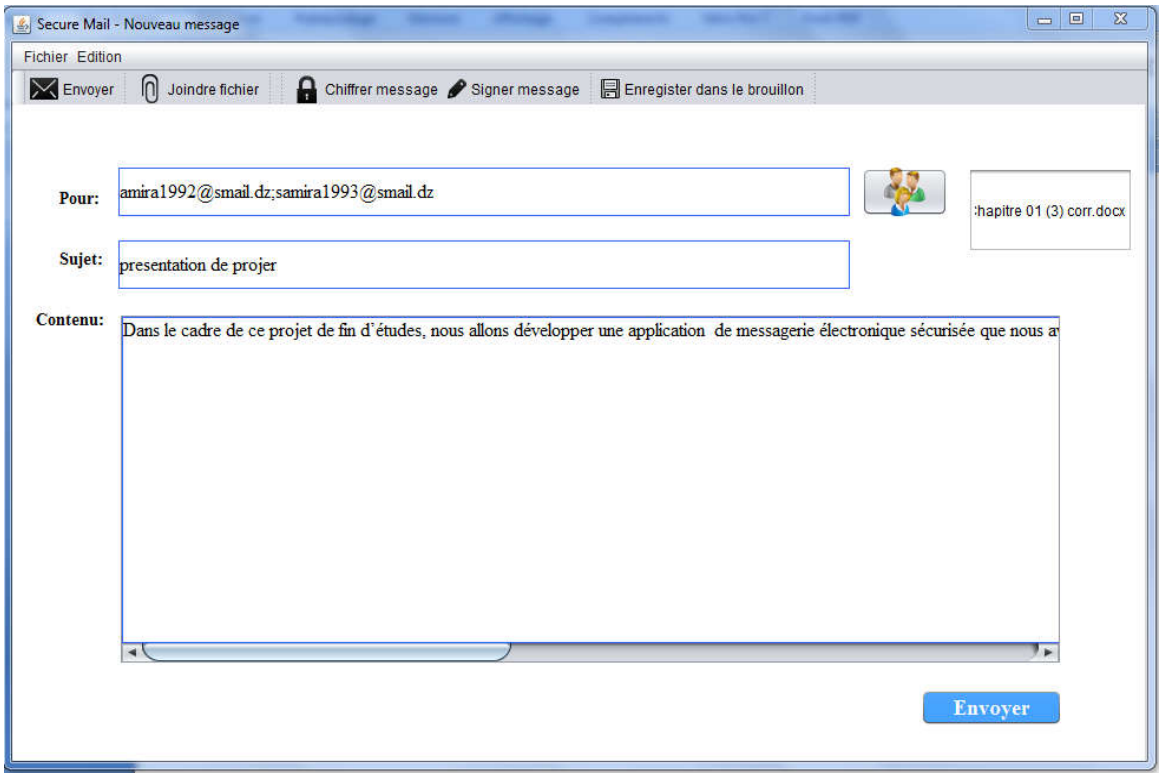

Figure 41: Fenêtre de composition d'un message.

3.7. FENETRE DE CHIFFREMENT D'UN MESSAGE A ENVOYER

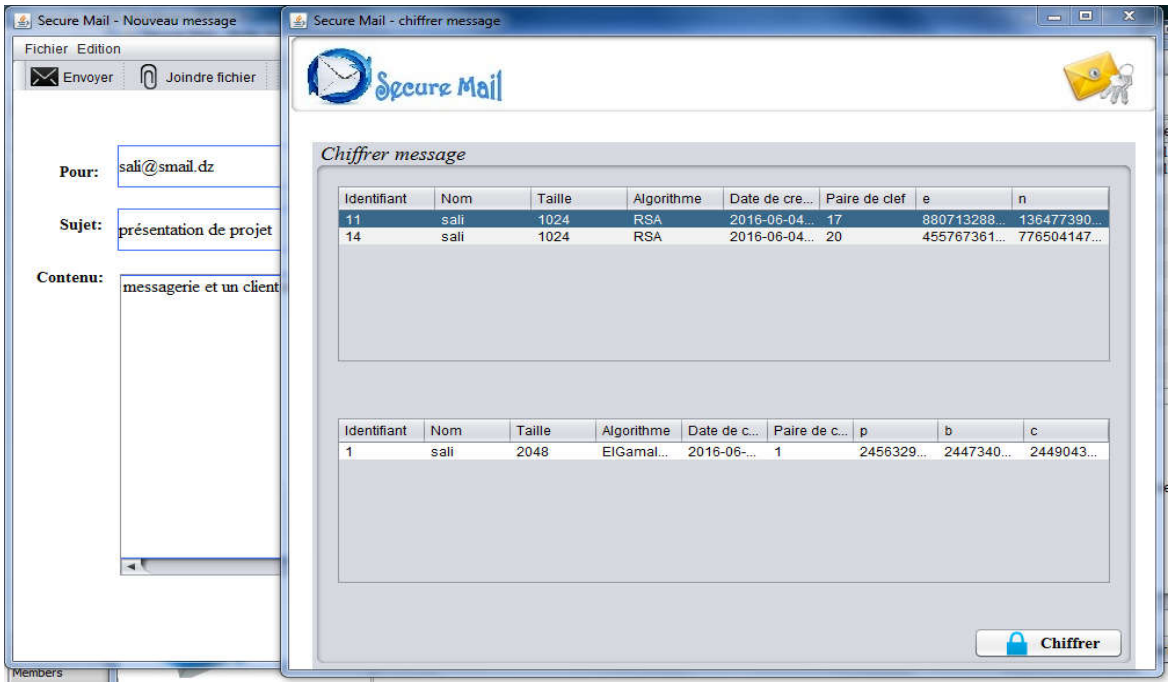

Figure 42: Fenêtre de chiffrement d'un message à envoyer.

3.8. FENETRE DE CONSULTATION DES MESSAGES REÇUS

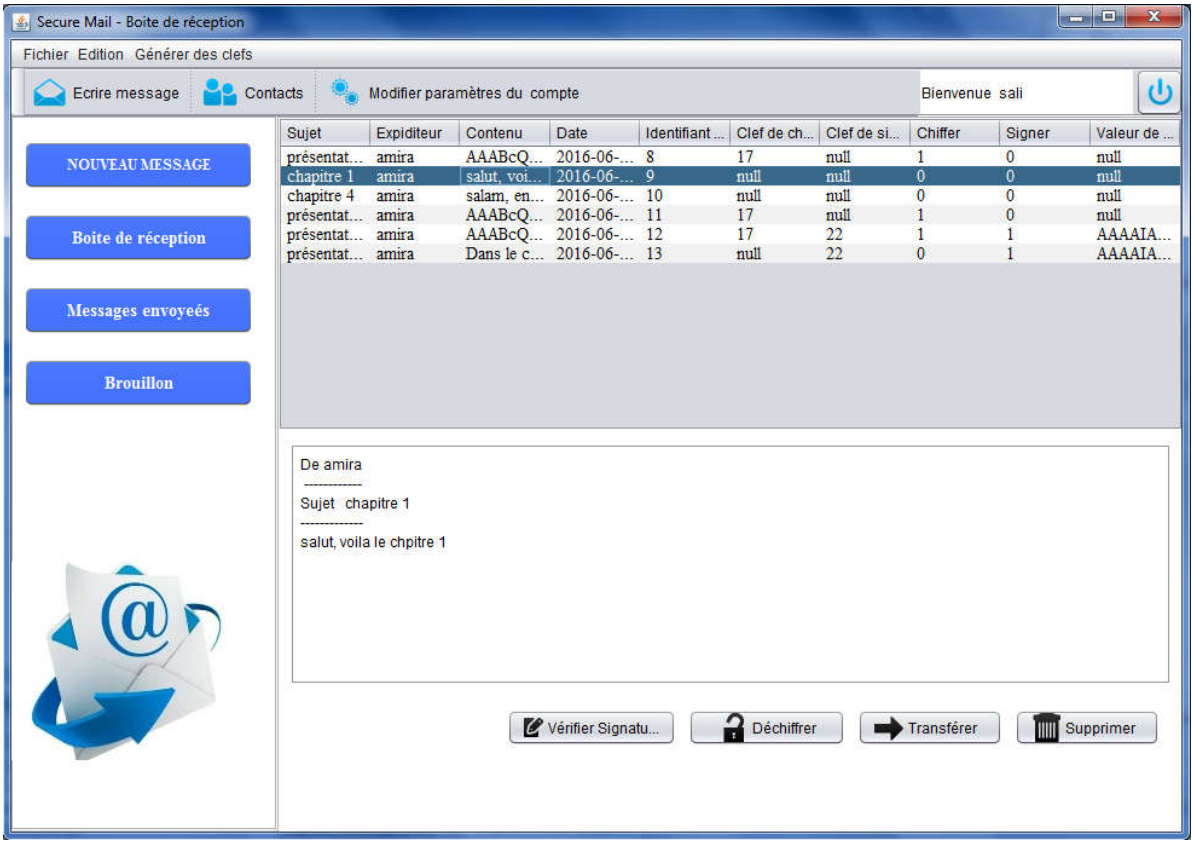

Figure 43: Fenêtre de consultation des messages reçus.

3.9. FENETRE DE CONSULTATION D'UN MESSAGE REÇU CHIFFRÉ

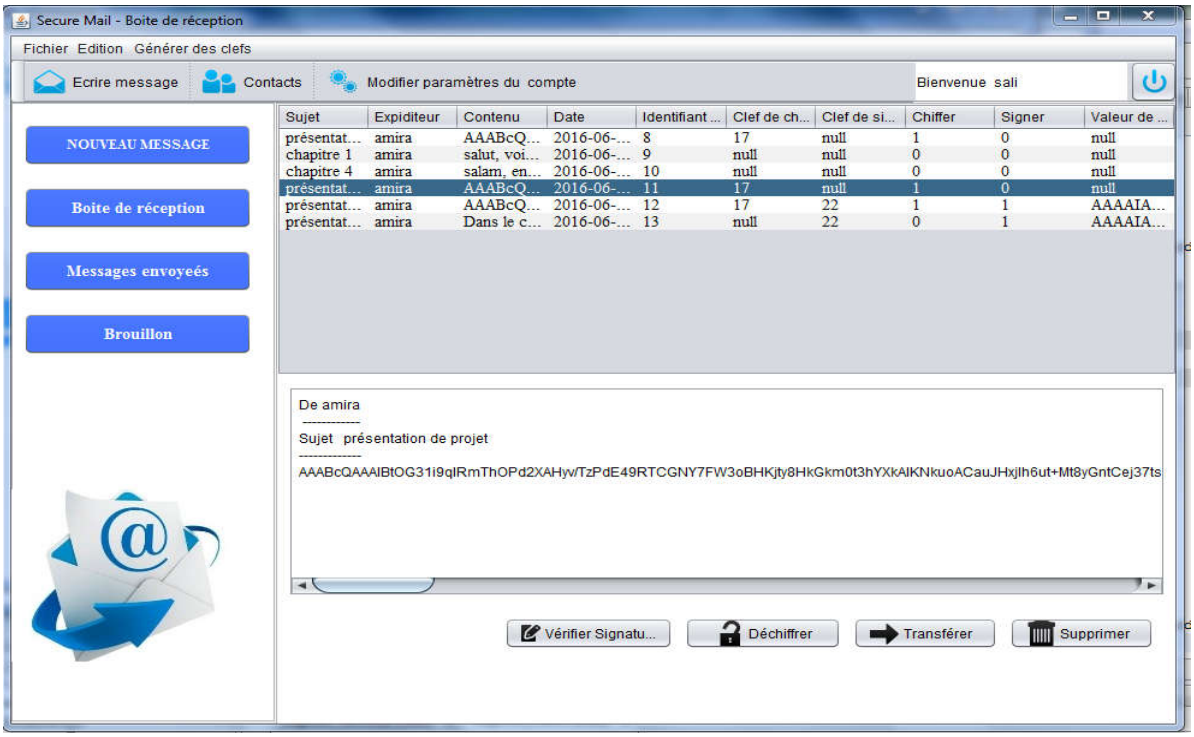

Figure 44: Fenêtre de consultation d'un message reçu chiffré.

3.10. FENETRE DE DÉCHIFFREMENT D'UN MESSAGE REÇU

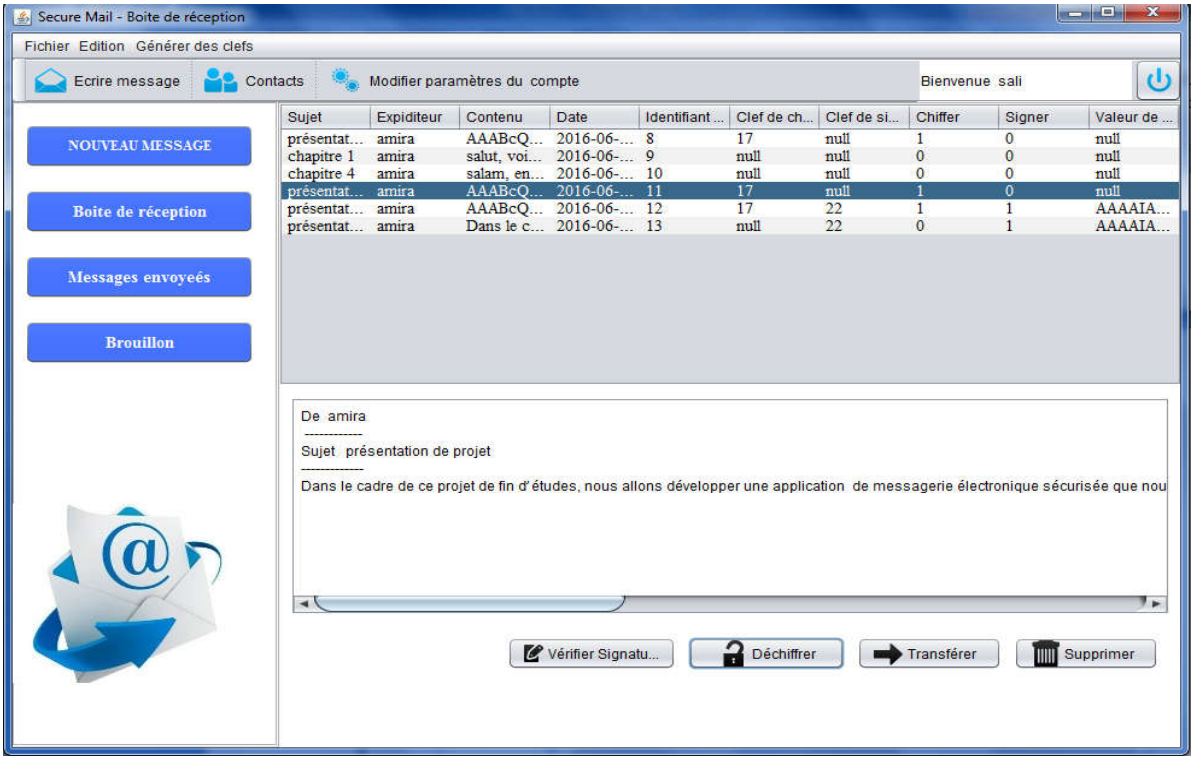

Figure 45: Fenêtre de déchiffrement d'un message.

3.11. FENETRE DE SIGNATURE D'UN MESSAGE A ENVOYER

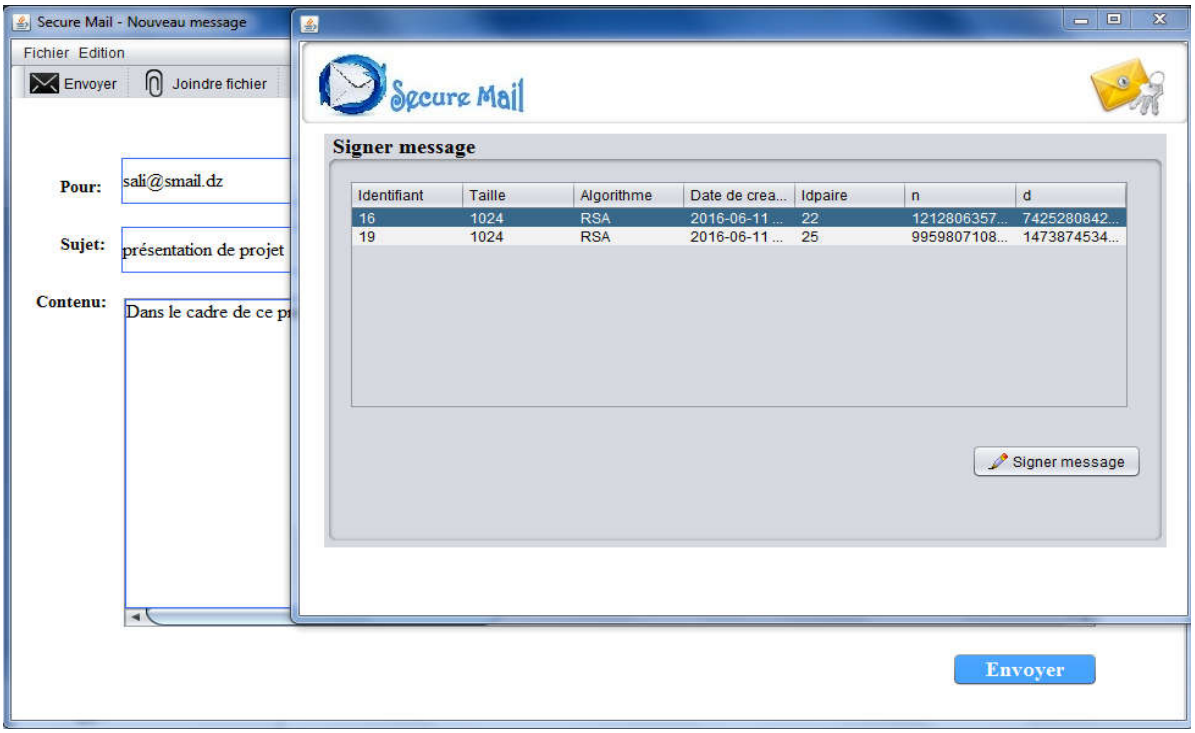

Figure 46: Fenêtre de signature d'un message à envoyer.

3.12. FENETRE DE VERIFICATION DE LA SIGNATURE D'UN MESSAGE REÇU

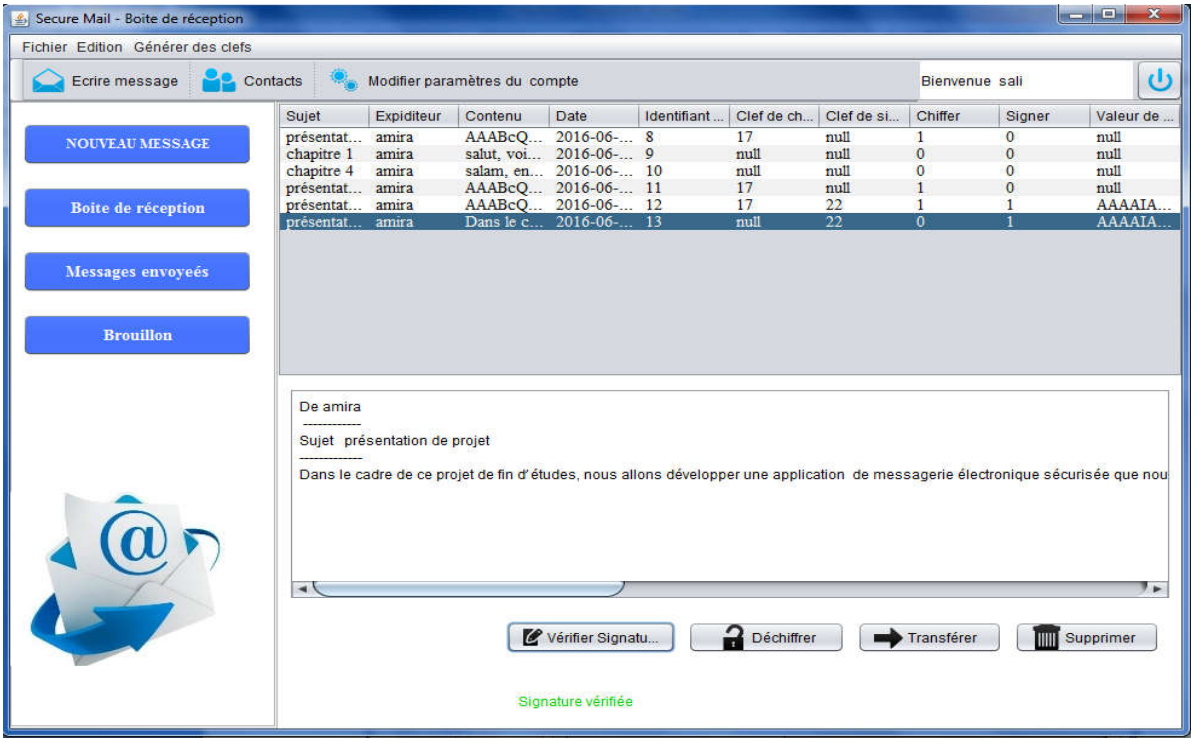

Figure 47: Fenêtre de vérification de la signature d'un message reçu.

4. CONCLUSION

A travers ce chapitre, nous avons présenté l'environnement de développement de notre application, et nous avons présenté quelques interfaces graphiques de notre système que nous avons jugé les plus importantes.

CONCLUSION GÉNÉRALE

Assurer la sécurité du courrier électronique, est une tâche très importante. Ceci est du à la sensibilité des données qui peuvent être contenues dans ce courrier. Il est donc important de cacher son contenu, pour protéger d'éventuels secrets professionnels ou tout simplement son intimité.

Notre travail se résume en la conception et la réalisation d'une application Client /Serveur permettant la gestion de la sécurité du courrier électronique à l'aide de protocoles cryptographiques. Dans ce contexte, nous avons cherché à développer une application flexible et évolutive permettant son amélioration par la suite afin d'anticiper les changements continus des besoins des utilisateurs.

Le projet s'est déroulé selon trois axes principaux afin de passer par les étapes essentielles de tout projet : l'analyse, la conception et la réalisation.

Pour la conception de notre application, nous avons choisi de suivre une méthode simple et générique basée sur le processus UP. Cette approche nous a permis de bien comprendre la problématique et de bien modéliser les objectifs à atteindre. Ce qui nous a donné la possibilité de réaliser un système stable et évolutif.

Pour la réalisation, nous avons utilisé JAVA comme langage de programmation et Mysql comme système de gestion de base de données.

La réalisation de ce projet nous a permis d'exploiter nos connaissances et de les améliorer dans les domaines suivants : le langage de modélisation UML, le processus de développement UP, le langage de programmation JAVA, l'utilisation d'un système de gestion de base de données Mysql, la manipulation des threads, et l'implémentation de différents protocoles cryptographiques.

BIBLIOGRAPHIE

[1] Laurent Cailleux. « Un nouveau modèle de correspondance pour un service de messagerie électronique avancée ». Cryptography and Security. Telecom Bretagne. Université de Rennes 1, 2015. French.

[2] Club de la sécurité des systèmes d'information Français, « La sécurité de messagerie ». CLUSIF, Septembre 2005.

[3] http://deptinfo.unice.fr/~julia/Crypto/10crypto.pdf

[4] Laurent Fabrice Mumposa Ungudi. « La mise en place et la sécurisation d'un système de messagerie électronique ». Institut supérieur de statistique de Kinshasa - Licence en ingénierie réseau informatique 2012.

[5] Jean-François PILLOU. « Le concept de réseau ». mai 2016.

[6] Le club informatique du Lycée du Pays de Soule . « Initiation à l'internet » support de cours (http://www.lyceedupaysdesoule.fr/informatique/cours/internet/cestquoi.htm). 2008

[7] « les protocoles reseaux » . http://www.zeitoun.net/articles/les_protocoles_reseaux/start.

[8] Guy Vastersavendts. « Utiliser internet et ses services ». Octobre 1998.

[9] Franck le Billon. « Réaliser des recherches a l'aide de techniques avancées. » Master entreprises et échangés internationaux. Paris Sorbonne.

[10] perso http://perso.modulonet.fr/placurie/Ressources/BTS1-ALSI/Chap-12-%20Le%20client-serveur.pdf.

[11] André Aoun, Jacques Chabert, Michel Jacob.« Architecture Client/Serveur » Université Paul Sabatier (Toulouse III). 2001.

[12] olivier aubert. « Le modèle client serveur » support de cours. (https://www.olivieraubert.net/cours/reseaux-iup/archi-client-serveur).

[13] B. Meriem et B. Amal. Réalisation d'une application client -serveur pour la gestion du patrimoine communal de la wilaya de Mila. Centre Universitaire De Mila. Année universitaire : 2012/2013

[14] http://fr.scribd.com/doc/19215590/Architecture-Client-Serveur.

[15] H.Amel, M. Seloua . « Modélisation et réalisation d'un site web dynamique d'un agence commerciale avec le langage de modélisation UML et le processus de développement 2TUP ». Université mentouri Constantine. 2007.

[16] Jean-François PILLOU. « environnement-client-serveur ». Septembre 2015.

[17] Jean-François Pillou. « Tout sur les systèmes d'information ». 2006.

[18] Craig Partridge. « The technical development of internet email ». 2008.

[19] Abdelkerim Douiri. « Etude et développement d'une application de messagerie électronique ». Ecole Nationale des Sciences de L'Informatique - Ingénieur informatique 2010.

[20] Jean-Etienne Poirrier. « Namur Linux Days2006 » .18 mars 2006.

[21] Guillaume Sigui. Article de Mise en place d'un système de messagerie électronique sous Linux. Date de publication : 30 décembre 2008. developpez.com.

[22] MOSAIQUE Informatique. Formations, créations web, rédaction d'ouvrages informatiques - Nancy. Glossaire : web, référencement, infographie, multimédia.

[23] Riad LEKHCHINE. « Construction d'une ontologie pour le domaine de la sécurité ». Application aux agents mobiles. 2008-2009.

[24] Anas Abou El Kalam. Sécurité des RO. Partie 2: Etude des attaques. 06/12/2007.

[25] Stéphane Gill. « Type d'attaques ». Copyright 2003.

[26] Sécurité dans les télécommunications et les technologies de l'information. UIT-T Bureau de la normalisation des télécommunications. Juin 2006.

[27] Jean-François PILLOU. « Introduction à la sécurité informatique ». Septembre 2015.

[28] Introduction et initiation à la Sécurité Informatique. www.securiteinfo.com.

[29] Abdesselem Beghriche. « De la Sécurité à la E-Confiance basée sur la Cryptographie à Seuil dans les Réseaux sans fil Ad hoc». Promotion 2008/2009.

[30] Solange Gheenaouti-Hélie. « Sécurité informatique et réseaux ». 2édition, 2006-2008,

ISBN 978-2-10-0521562

[31] C. TCHEPNDA. « Authentification dans les Réseaux Véhiculaires Opérés ». Ecole Nationale Supérieure des Télécommunications Thèse de doctorat, 2008.

[32] Introduction à la sécurité informatique.
https://www.securiteinfo.com/conseils/introsecu.shtml

[33] Gérard - Michel Cochard. Sécurité des systèmes d'information. 27 septembre 2006.

[34] Eric Detoisien. Les attaques externes. LinuxFocus article number 282. 2005-01-14. generated by lfparser_pdf version 2.51.

[35] Jean-François Pillou, Jean-Philippe Bay. « Tout sur la sécurité informatique ». 2005, 2009, 2013, 2016. 5 rue Laromiguière, 75005 Paris. ISBN 978-2-10-074608-8.

[36] Pierre-Alain Fouque. « Cryptographie appliquée » rapport technique. Laboratoire de cryptographie de la direction centrale de la Sécurité des systèmes d'information (DCSSI). 2004.

[37] SOUICI Ismahane. « Sécurisation évolutionnaire du transfert d'images ». Présentée en vue d'obtention du diplôme de DOCTORAT en Informatique. 2012/2013.

[38] Ramdani Mohamed. « Problèmes de sécurité dans les réseaux de capteurs avec prise en charge de l'énergie ». Blida, Novembre 2013.

[39] ram-0000. « Introduction à la Cryptographie ». 8/01/2009.

[40] Ahmed Mehaoua. « crypto_synthese». http://www.mi.parisdescartes.fr.

[41] La sécurité des réseaux. Chapitre 10. Mercredi, 8. novembre 2006.

[42] Rolland Balzon Philippe. « Principaux algorithmes de cryptage ». Department of Computer Science. SEPRO Robotique. France. 11juillet 2002.

[43] Carine BERNARD. « L'utilisation de la signature électronique au CNRS ». Stage d'application Institut régional d'administration de Lille Promotion « Jules Verne » 2003-2004.

[44] Mustapha Benjada. « La sécurité informatique - La sécurité des informations ». https://www.securiteinfo.com/cryptographie/pki.shtml.

[45] Dr. Y. Challal. « Infrastructures à Clés Publiques. Aspects Techniques et organisationnels ». Maître de conférences. Université de Technologie de Compiègne. France.

[46] Pascal Gachet. « Sécurité et PKI ». 2002.

[47] Michel Riguidel. « La sécurité des réseaux et des systèmes ». Paris année scolaire 2006-2007.

[48] Omar Cheikhrouhou. « Sécurité des réseaux ad hoc ». Présenté à l'Ecole Nationale d'Ingénieurs de Sfax. 4 juillet 2005.

[49] Laurent AUDIBERT. « UML 2 ». Editions Ayrolles. 2007/2008.

[50] Joseph Gabay, David Gabay. Dunod . "UML 2 ANALYSE ET CONCEPTION ». Mise en œuvre guidée avec études de cas. Paris, 2008. ISBN 978-2-10-053567-5

[51] P. Roques et F.Vallée. « UML 2 en action De l'analyse des besoins à la conception » 4ème édition 2007, EYROLLES.

[52] Yves Wautelet, Laurent Louvigny, Manuel Kolp. L'unified process comme methodologie de gestion de projet informatique. Université Catholique de Louvain, 1 Place des Doyens, 1438 Louvain-la-Neuve, Belgique

[53] http://sabricole.developpez.com/uml/tutoriel/unifiedProcess/

[54] Di Gallo Frédéric. « Méthodologie des systèmes d'information – UML » .Cours du Cycle Probatoire.

[55] http://www.techno-science.net/?onglet=glossaire&definition=5346

[56] Article sur « Java et la programmation Client/Serveur ». RIST Vol. 7 N°02. 1997.

[57] Michael J. Donahoo - Kenneth L. Calvert. « TCP/IP Sockets in C: Practical Guide for Programmers». 2009.

[58] Jean-Michel Doudoux. « Développons en Java ». 1999-2016.