REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULARE

Ministère de l'enseignement supérieur et de la recherche scientifique

Université Mouloud Mammeri de Tizi ouzou

Faculté du Génie Electrique et Informatique

Département Informatique

Mémoire

De fin d'étude

En vue de l'obtention du diplôme de Master 2
En Informatique
Cycle LMD

Conception et réalisation d'un système de hachage basé sur la fonction SHA-1

Proposé et dirigé par :
Me. HADAOUI.

Présenté par :
- *Mr MEGHNINE KHALED*
-*Mr KHELIL RAFIK*

Année 2013/2014

Remerciement

« Il y a pire dans la vie que d'avoir échoué, c'est de ne pas avoir essayé »

Roosevelt

En premier lieu, nous remercions Dieu le tout puissant de nous a voir aidé pour arriver à terme de ce travail qui représente le fruit de plusieurs années d'études.

A travers ce modeste travail, nous tenons à remercier vivement notre encadreur « Me. HADAOUI. », Et pour toutes les commodités et aisances qu'il nous a apportées durant notre étude et réalisation de ce projet.

Nos remerciements les plus vifs s'adressent aussi à messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer notre travail.

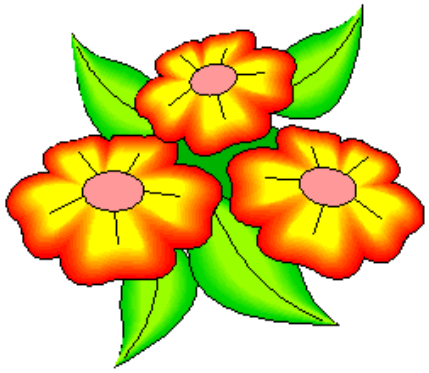
Nous exprimons également notre gratitude à tous les professeurs qui ont collaboré à notre formation depuis notre premier cycle d'étude jusqu'à la fin de Notre cycle universitaire.

Sans omettre bien sur de remercier profondément tous ceux qui ont contribué de prés ou de loin à la réalisation du présent travail.

PAR:

Mr. MEGHNINE KHALED

Mr. KHELIL RAFIK



Dédicace

Je dédie ce modeste travail :

✍ *Je dédie ce travail à mes très chers parents, pour leur soutien et tous les efforts qu'on m'a donnée le long de mon parcours et je leurs souhaite bonne santé et longue vie.*

✍ *A Mon frère et mes sœurs et toute la famille sans exception.*

✍ *A Tous mes amis et collègues un peu par tous.*

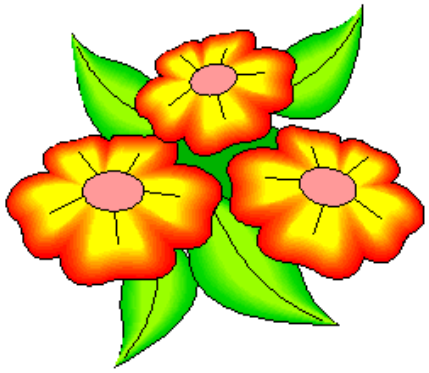
✍ *A tous ceux qui m'ont aidé durant ma vie universitaire.*

✍ *A toute ma promotion.*

✍ *A ceux qui m'aiment*

Je dédie ce travail

RAFIK



Dédicace

Je dédie ce modeste travail :

✍ *A Mes très chères parents, pour leur sacrifie et leur dévouement pour mon bonheur.*

✍ *A Mes frères et sœurs sans et toute la famille sans exception.*

✍ *A Tous mes amis et collègues un peu par tous.*

✍ *A tous ceux qui m'ont aidé durant ma vie universitaire.*

✍ *A toute ma promotion.*

✍ *A ceux qui m'aiment*

Je dédie ce travail

KHALED

Sommaire :

Chapitre 3 : Analyse et Conception

Introduction générale

Liste de Figure :

Introduction générale :

Depuis l'apparition de l'informatique, les entreprises n'ont pas cessé de l'utiliser et de la développer selon leurs besoins pour faciliter le travail et pour mieux s'approcher de clients, dans le but d'augmenter leurs profits. En revanche, les entreprises en mal à contrôler les systèmes informatiques, entre autre, leur sécurité.

En effet, les entreprises doit faire face à la curiosité de savoir, le vol des idées ou des informations stratégiques des malfaiteurs informatiques qui cherchent toujours à développer la criminalité informatique. Tous cela, à pousser la communauté informatique à apprendre des mesures de sécurité dans la complexité de ses solutions.

Pour cela, notre travail consiste à étudier une solution existante dans le domaine de cryptographie qu'est l'un des mécanismes de sécurité qui permettent de résoudre les nombreux problèmes menaçants la vie privé et la sécurité sur internet, elle consiste a transformer un message clair en un message crypté .

Pour assurer les objectifs de la cryptographie ;la solution proposé une solution complémentaire qui consiste à appliquer une fonction mathématique sur une portion du message. Cette fonction mathématique s'appelle fonction de hachage et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message.

Des différentes fonctions de hachage ont été proposées, nous avons basé principalement sur la fonction SHA-1 afin d'éviter les conséquences désagréables sur la sécurité d'un vol de base de données.

Pour mener à bien notre travail, nous avons structuré le présent mémoire comme suit :

- ✓ **Chapitre I** : Réseaux informatiques et leur sécurité : divisé en deux parties.
  - **Partie 1** : Généralité sur les réseaux informatiques : comprend des généralités sur les réseaux, ainsi les modèles OSI, TCP/IP et client/serveur.
  - **Partie 2** : Sécurité informatique : comprend des généralités sur la sécurité informatique, notamment la malveillance informatique et les mécanismes de sécurité.

- ✓ **Chapitre II** : Les concepts fondamentaux de la cryptographie moderne divisé en trois parties.
  - **Partie 1 :** La cryptographie : comprend des généralités sur la cryptographie classique, moderne, objectifs et techniques de cryptographie.
  - **Partie 2 :** Les fonctions de hachage : est consacrée à la présentation des fonctions de hachage, leur domaines d'utilisations, les principaux algorithmes de hachage et comment construire une fonction de hachage.
  - **Partie 3 :** Le SHA-1 : consacrée à la présentation, description de l'algorithme SHA-1 et les attaques possible sur le SHA-1.
- ✓ ***Chapitre III :*** Analyse et conception : ce chapitre est consacré à la conception de notre application où nous avons opté pour le langage UML.
- ✓ ***Chapitre IV :*** Implémentation et réalisation : comporte quand a lui la représentation de l'environnement de développement dont lequel notre application à été réalisée, les outils utilisées et quelques interfaces de notre application.

Nous terminerons notre travail par une conclusion générale.

Chapitre I

Les réseaux informatiques

et leur sécurité

Partie I: Réseaux informatiques

I.1. Introduction :

Les réseaux informatiques sont nés du besoin de relier des terminaux distants à un site central, puis des ordinateurs entre eux, et en fin des machines terminales, telles que des stations de travail, à leur serveur. Dans un premier temps, ces communications étaient destinées au transport de données informatiques. Aujourd'hui, les réseaux tendent à intégrer également la parole et la vidéo. **[01]**

Les réseaux informatiques qui permettaient à leur origine de relier des terminaux passifs à de gros ordinateurs centraux autorisent à l'heure actuelle l'interconnexion de tous types, d'ordinateurs que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Les services qu'ils offrent font partie de la vie courante des entreprises et administrations (banques, gestion, commerce, bases de données, recherche,...) et des particuliers (messagerie, loisirs, services d'informations par minitel et Internet ...).**[02]**

I.2. Définition :

Un réseau informatique est un ensemble de machines (ou de périphériques) interconnectés entre eux grâce à des lignes physiques (câble, fibre optique, satellites…), permettant à plusieurs personnes distantes de communiquer et de partager des ressources de types variés, données, paroles, photos, vidéos etc. **[03]**

Autrement dit,un réseau informatique est un ensemble des moyens matériels et logiciels qui permet de communiquer entre eux afin d'assurer les échanges d'informations, le transfert de fichiers et le partage de ressources de la messagerie ou de l'exécution des programmes à distance.

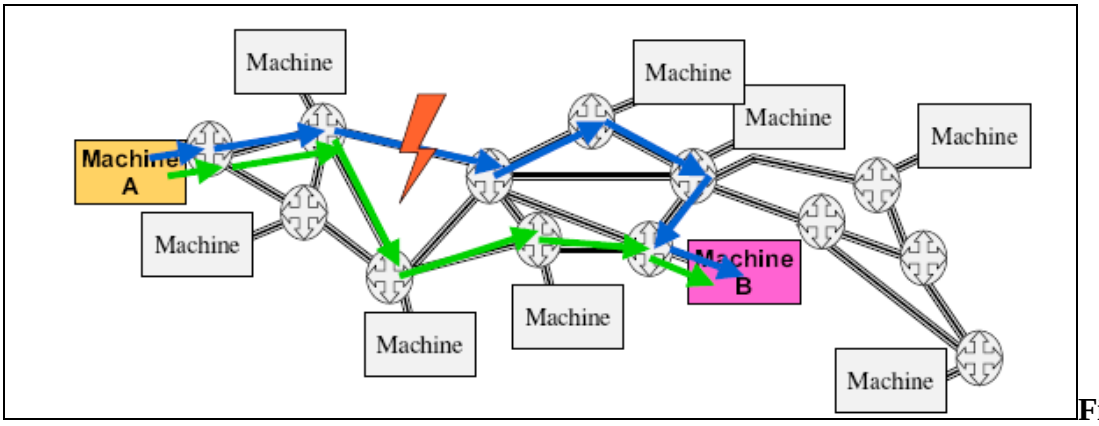


Fig. I.1. Structure générale d'un réseau

I.3. Objectifs des réseaux:

Les réseaux informatiques permettent :

- Le partage de fichier, d'application et de périphériques informatiques ;
- La communication entre personnes (grâce au courrier électronique, la discussion en direct,…) ;
- La communication entre processus (entre des machines industrielles) ;
- La garantie de l'unicité de l'information (bases de données) ;
- Diminution des coûts grâce aux partages des données et des périphériques ;
- Standardisation des applications ;
- Accès aux données en temps utile ;
- Communication et organisation en temps efficace.

I.4. Classification des réseaux:

On a deux types de classifications :

I.4.1. Classification selon la distance :[04]

Il ya généralement 3 catégories de réseaux :

- **Réseaux locaux LAN :**

  Les réseaux locaux, qu'on appelle également **LAN** (Local Area Networks) sont des réseaux privés dont la taille ne dépasse pas quelques kilomètres. On les utilise principalement pour relier les ordinateur personnels ou les stations de travail que l'on trouve dans les entreprise à des ressources partagées (par exemple des imprimantes) avec lesquels ils échangent des informations

- **Réseaux métropolitain MAN :**

  Les **MAN** (Métropolitain Area Network) interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un réseau MAN permet à deux nœuds distants de communiquer comme s'ils faisaient partie d'un même réseau local. Un réseau MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

- **Réseaux étendue WAN :**

Un réseau **WAN** (Wilde Area Network ou réseau étendu) interconnecte plusieurs LAN à travers de grandes distances géographiques, les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles, les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau, le plus connu des WAN est Internet.

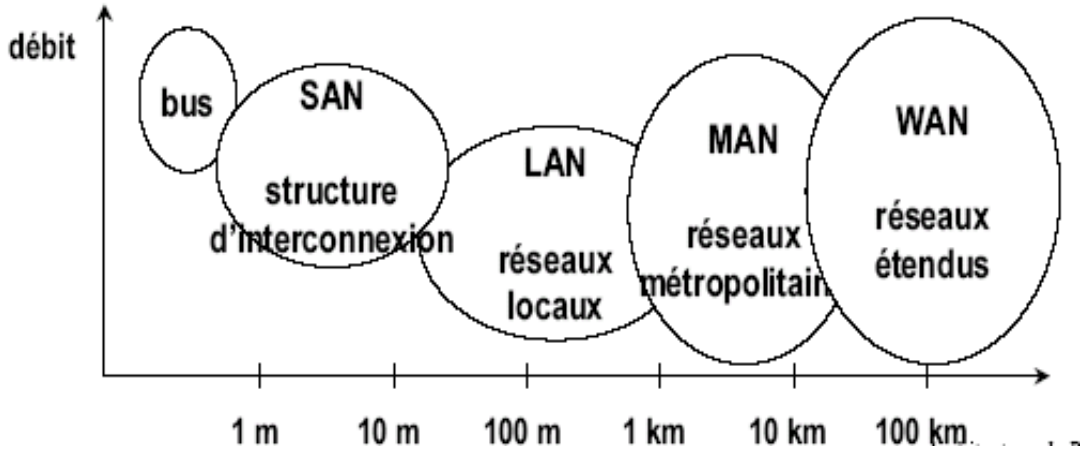


Fig. I.2. Classification des réseaux informatiques selon leur taille

I.4.2 Classification selon la topologie:[05]

La topologie est la manière de relier entre eux les équipements informatiques, il s'agit de la structure du réseau. Il y a plusieurs topologies, pour des performances différentes. A savoir, les débits, le nombre d'utilisateurs maximum, le temps d'accès, la tolérance aux pannes, la longueur de câblage et les types d'applications différentes.

- **La topologie en anneau :**

C'est une configuration de liaisons point à point entre deux stations voisines, l'ensemble des liaisons formant un anneau. Un jeton circule de station en station, donnant à la station qui a le jeton le droit d'émettre un message. Toute station doit attendre de recevoir le jeton pour émettre, les stations successives reçoivent tous les messages, s'il leur est destiné, elles le gardent, sinon elles le retransmettent à la station suivante de l'anneau. Les stations sont toutes actives. Le protocole du jeton permet une résolution très efficace des conflits de transmission. Exemple d'anneau est :**Token Ring.**

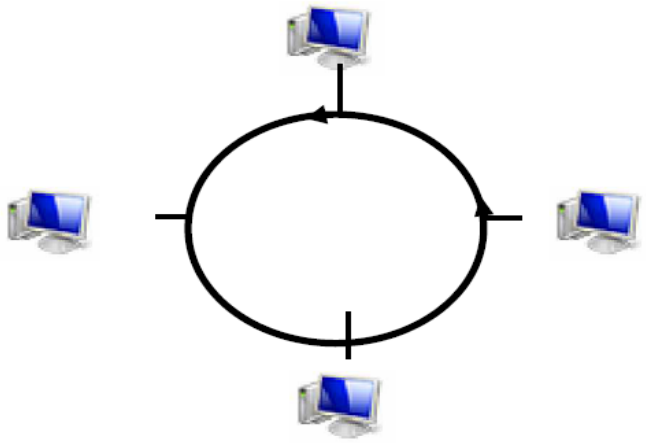


Fig.I.3 : Topologie en anneau.

- **La topologie en bus :**

Elle met en œuvre une liaison multipoints, les stations sont en permanence à l'écoute sur le réseau. Chaque fois qu'elles détectent qu'un message leur est destiné, elles le prennent. Chaque fois qu'une station veut envoyer un message, elle le transmet sur le bus. Si le hasard fait que deux stations émettent en même temps, il y a collision des transmissions, et un protocole particulier gère la réémission des messages perdus. Les principaux problèmes dus à cette topologie en bus sont la détection et la gestion des collisions.

Les stations sont toutes passives, il n'y a pas de hiérarchie entre elles, l'exemple le plus connu est **Ethernet.**

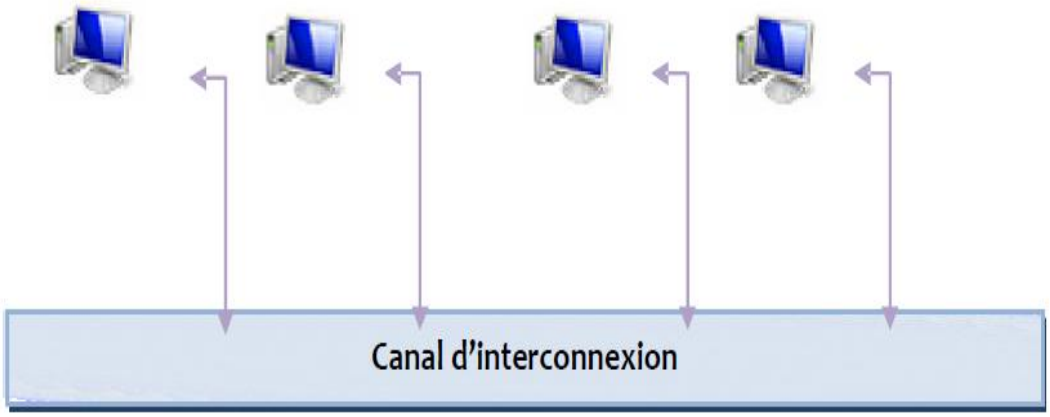


Fig.I.4 : Topologie en bus.

- **La topologie en étoile :**

Dans une **topologie en étoile**, les ordinateurs du réseau sont reliés à un système matériel central appelé **concentrateur** (en anglais hub, littéralement moyen de roue), il s'agit d'une boîte comprenant un certain nombre de jonctions auxquelles il est possible de raccorder les câbles réseau en provenance des ordinateurs, celui-ci a pour rôle d'assurer la communication entre les différentes jonctions.

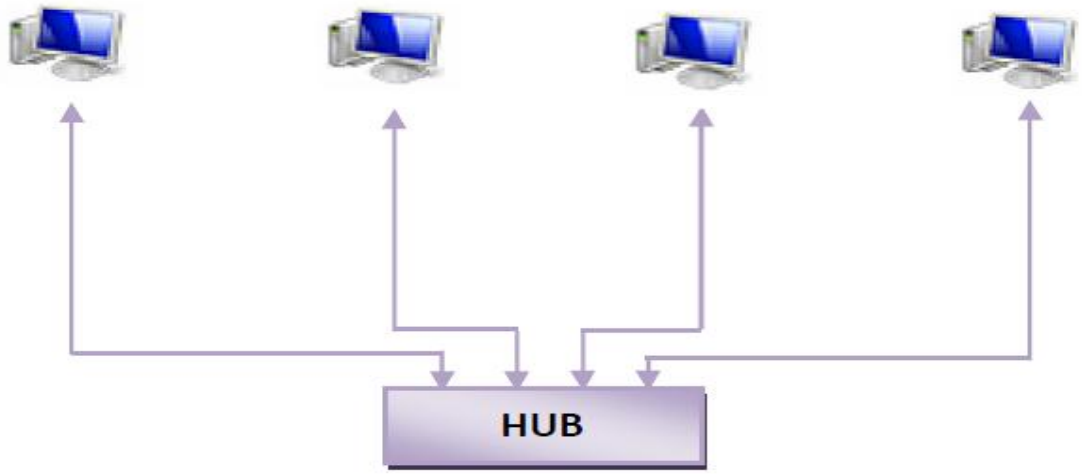


Fig.I.5 : Topologie en étoile

Contrairement aux réseaux construits sur une topologie en bus, les réseaux suivant une topologie en étoile sont beaucoup moins vulnérables car une des connexions peut être débranchée sans paralyser le reste du réseau. Le point névralgique de ce réseau est le concentrateur, car sans lui plus aucune communication entre les ordinateurs du réseau n'est possible.

En revanche, un réseau à topologie en étoile est plus onéreux qu'un réseau à topologie en bus car un matériel supplémentaire est nécessaire (le hub).

I.5. Fonctionnement des réseaux

I.5.1 Le modèle OSI : [06]

Les réseaux informatiques fondent leur conception sur le modèle de référence à sept couches, OSI (Open System Interconnections) défini par l'ISO (International Standard Organisation). Dans ce modèle, chaque couche a une fonction particulière et se base sur le service de la couche immédiatement inférieure. Rappelons succinctement les fonctionnalités de ces différentes couches :

- **Couche 7 : La couche application**

La couche application joue le rôle d'une interface d'accès des applications au réseau. C'est la couche OSI la plus proche de l'utilisateur. Elle fournit des services réseaux aux applications de l'utilisateur. Principalement des services de transfert de fichiers (FTP), de messagerie (SMTP), de documentation hypertexte (HTTP), etc...

- **Couche 6 : La couche présentation :**

La couche Présentation s'intéresse à la syntaxe et à la sémantique des informations (elle résout les problèmes des différences syntaxiques des données échangées entre deux applications) →détermine le format utilisé pour l'échange des données entre ordinateurs du réseau.

Cette couche assure essentiellement les fonctions suivantes:

- La communication entre des applications qui utilisent des types de messages (données, voix, images) ou des langages différents et ce sans qu'elles se rendre compte des conversions syntaxiques.
- La représentation des données transférées entre entités applications et de la représentation de la structure de données et représentation de l'ensemble des actions effectuées sur cette structure de données.
- Les conversions correspondantes aux caractéristiques des écrans et des imprimantes (jeux des caractères, définition, mode de fonctionnement), ainsi que les conversions nécessaires pour les structures des fichiers sur les disques.
- L'encodage dans une norme agrée permettant à des équipements ASCII et EBCDIC par exemple de communiquer.
- La compression / décompression des données et leur chiffrement si nécessaire.
- Cryptage / décryptage.

- **Couche 5 : La couche session :**

La couche Session gère la connexion entre deux ordinateurs du réseau.

Cette couche assure essentiellement les fonctions suivantes :

- L'ouverture et la fermeture d'une connexion (d'une session) avec le site distant.
- La synchronisation des tâches utilisateur à l'aide des points de resynchronisation (pour redémarrer en cas de problème sur un point précis).
- Le contrôle du dialogue entre les processus communicants (qui transmet, à qui, à quel moment, pour combien de temps, …).

- **Couche 4 : La couche transport :**

La couche Transport est responsable de la bonne transmission des messages complets au destinataire. Le rôle principal de cette couche est d'accepter des données de la couche supérieur (couche session), de les découper en paquets si nécessaire et de les transmettre à la couche réseau. En réception, cette couche effectue donc aussi le réassemblage des paquets reçus de la couche inférieure (couche réseau) pour reconstituer les messages.

- **Couche 3 : La couche réseau :**

Le rôle essentiel de la couche Réseau est de déterminer la manière dont les paquets sont routés de la source à la destination → 'le routage'.

En plus du routage, cette couche assure la gestion des congestions pour éviter freinage et éventuellement blocage de la circulation des paquets. Une des solutions par exemple consiste à contourner les points de congestion en empruntant d'autres routes.

- **Couche 2 : La couche liaison de données :**

Le rôle principal de la couche liaison de données est de fournir à la couche supérieure (couche réseau) un moyen de communication fiable essentiellement en assurant un contrôle d'erreurs de transmission.

Pour cela, les données sont décomposées en trames de données puis envoyées en séquence. Le récepteur doit confirmer ou infirmer la bonne réception de chaque trame par le biais d'acquittements. En général, des fonctions de gestion des erreurs (survenues sur la couche physique) y sont intégrées. Cette couche doit aussi assurer qu'un récepteur lent ne soit pas submergé par des données émises par un émetteur rapide (régulation de flux).

- **Couche 1 : La couche physique :**

Cette couche se charge essentiellement de la transmission fiable des bits de façon brute (un bit 1 envoyé doit bien être reçu comme bit valant 1) sur un canal de communication d'une extrémité à une autre. Concrètement, cette couche doit normaliser les caractéristiques électriques (un bit 1 doit être représenté par une tension de 5 V, par exemple).

Cette couche est principalement constituée de :

- Support physique.

- Interfaces de connexion (carte réseau : ETHERNET, …) auxquels peuvent s'ajouter des :

- Répéteurs.

- Multiplexeur / démultiplexeur.

- Modems.

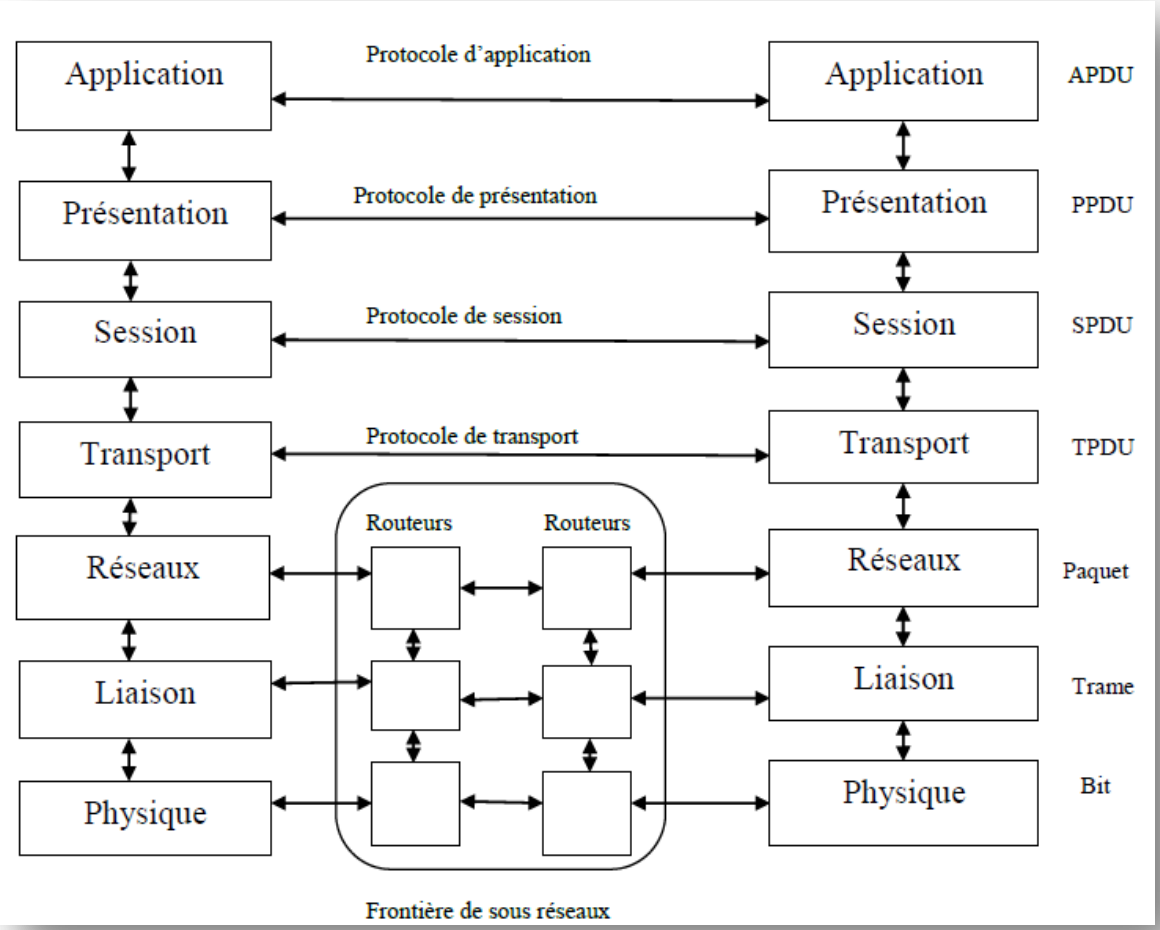


Fig.I.6: Le modèle OSI des réseaux.

I.5.2. Le modèle TCP/IP (le modèle internet) [07]

TCP/IP est la source du réseau Internet, il est aussi adopté par de nombreux réseaux privés, les deux principaux protocoles définis sont les suivants :

- Internet Protocole (IP) : qui est un Protocole de niveau réseau assurant un service sans connexion.
- Transmission Control Protocole (TCP) : qui est un Protocole de niveau transport qui fournit un service fiable avec connexion.

TCP/IP représente d'une certaine façon l'ensemble des règles de communication sur Internet et se base sur la notion adressage IP, c'est-à-dire le fait de fournir une adresse IP

à chaque machine du réseau afin de pouvoir acheminer des paquets de données. Étant donné que la suite de protocoles TCP/IP a été créé par la défense américaine, elle est conçue pour répondre à un certains nombres de critères parmi lesquels :

- Le fractionnement des messages en paquets.
- L'utilisation d'un système d'adresses.
- L'acheminement des données sur le réseau (routage).
- Le contrôle des erreurs de transmission de données.

Le modèle TCP/IP s'appuie sur le modèle OSI (7 couches) mais ne comporte que 4 couches ;

Fig.I.7 : Les quatre couches de TCP/IP.

Comme on peut le remarquer, les couches du modèle TCP/IP ont des tâches beaucoup plus diverses que les couches du modèle OSI, étant donné que certaines couches du modèle TCP/IP correspondent à plusieurs couches du modèle OSI.

Les rôles des différentes couches sont les suivants :

- **Couche application :** C'est à ce niveau que de nombreuses applications accèdent au réseau, elles sont basées sur des protocoles de haut niveau tels que : FTP, SMTP et http.
- **Couche transport :** Assure l'acheminement des données et les mécanismes permettant de connaître l'état de la transmission .les protocoles **TCP** (Transport Control Protocol) et **UDP** (User Datagram Protocol) permettent de mettre en place un

transfert de données en mode connecté ou sans connexion pour chacun des messages fournis par les applications disponibles.

- **Couche Internet :** Est chargée de fournir les paquets des données. Elle définit les datagrammes et gère la décomposition/recomposition des segments. Les rôles de cette couche sont réalisés par le protocole universel **IP** (Internet Protocol) qui fournit l'acheminement qui assure que les messages seront correctement fournis à leur destination (routage).Il existe d'autres protocoles : ICMP, ARP, RARP.
- **Couche accès réseau :** Appelée aussi «couche de liaison de données» spécifie la forme sous laquelle les données doivent être acheminéesquel que soit le type de réseau utilisé.

I.6. Comparaison des modèles OSI et TCP/IP

Les protocoles qui constituent la suite de protocole TCP/IP peuvent être décrits selon les termes du modèle de référence OSI. Dans le modèle OSI, la couche d'accès réseau et la couche application du modèle TCP/IP sont encore divisées pour décrire des fonctions discrètes qui doivent intervenir au niveau de ces couches.

Au niveau de la couche d'accès au réseau, la suite de protocole TCP/IP ne spécifie pas quels protocoles utiliser lors de la transmission à travers un support physique ; elle décrit uniquement la remise depuis la couche Internet aux protocoles réseau physiques. Les couches 1 et 2 du modèle OSI traitent des procédures nécessaires à l'accès aux supports et des moyens physiques pour envoyer des données à travers un réseau.

Les principaux parallèles entre les deux modèles de réseau se situent aux couches 3 et 4 du modèle OSI. La couche 3 du modèle OSI est utilisée presque partout dans le monde afin de traiter et de documenter l'éventail des processus qui interviennent dans tous les réseaux de données pour adresser et acheminer des messages à travers un inter réseau. Le protocole IP est le protocole de la suite TCP/IP qui contient la fonctionnalité décrite à la couche 3.

La couche 4, la couche transport du modèle OSI sert souvent à décrire des services ou des fonctions générales qui gèrent des conversations individuelles entre des hôtes source et de destination. Ces fonctions incluent le reçu, la reprise sur erreur et le séquençage. Au niveau de cette couche les protocoles TCP/IP Transmission Control Protocol (TCP) et User Datagramme Protocol (UDP) fournissent les fonctionnalités nécessaires.

La couche application TCP/IP inclut plusieurs protocoles qui fournissent des fonctionnalités spécifiques à plusieurs applications d'utilisateur final. Les couches 5, 6 et 7 du modèle OSI sont utilisées en tant que références pour les développeurs et les éditeurs des

logiciels d'application, afin de créer des produits qui doivent accéder aux réseaux pour des communications.

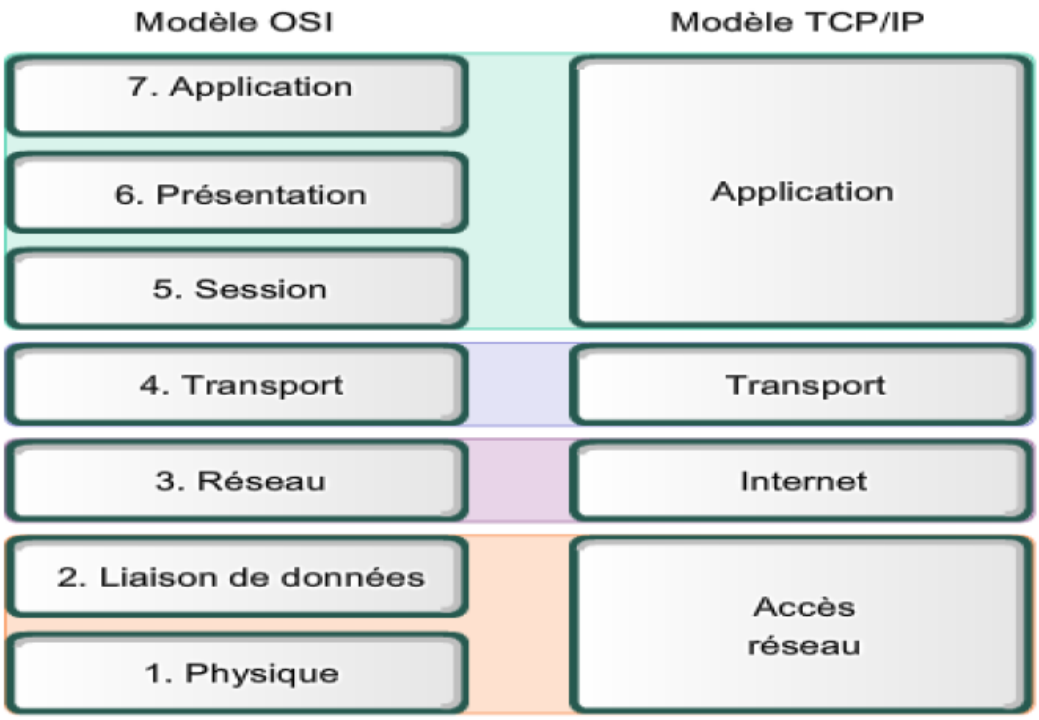


Fig. I.8. Comparaison des modèles OSI et TCP/IP

I.7. Le modèle Client-serveur

I.7.1. Qu'est-ce que le Client-serveur ?

C'est un mode de dialogue entre deux processus, le premier appelé client demande l'exécution de services au second appelé serveur. Ce dernier accomplit les services et envoie en retour des réponses. En général, un serveur est capable de traiter les requêtes de plusieurs clients et aussi permet de partager des ressources entre plusieurs clients. Dans un environnement purement client/serveur, les ordinateurs du réseau (les clients) ne peuvent voir que le serveur, c'est l'un des principaux atouts de ce modèle.

Type d'architecture applicative ou les programmes sont repartis entre processus client et serveurs communiquant par des requêtes avec des réponses.

I.7.2. Notions de bases

- **Client :**C'est le processus demandant l'exécution d'une opération à un autre processus par envoi d'un message contenant le descriptif de l'opération à exécuter et attendant la réponse à cette opération par un message en retour.
- **Serveur :**C'est un processus accomplissant une opération sur demande d'un client.
- **Requête :** C'est un message transmis par un client à un serveur décrivant l'opération à exécuter pour le compte d'un client.
- **Réponse :** C'est un message transmis par un serveur à un client suite à l'exécution d'une opération contenant les paramètres de retour de l'opération.
- **Middleware :**C'est le logiciel qui assure les dialogues entre les clients et les serveurs souvent hétérogènes.

I.7.3. Utilisation du modèle client /serveur

L'utilisation du modèle client/serveur s'étend de plus en plus vers tous les domaines d'activités :

- Word wide web.
- Gestion de bases de données.
- Systèmes transactionnels.
- Système de messagerie.
- Système de partage de données.
- Calcul scientifique etc.…

I.7.4. Exemple de modèle client/serveur

- Serveur de fichier (auffs, nfsd).
- Serveur d'impression (ipd).
- Serveur de calcul.
- Serveur base de données.
- Serveur de noms (annuaire des services).

I.7.5. Fonctionnement d'un système Client/serveur

Un système client/serveur fonctionne selon le schéma suivant :

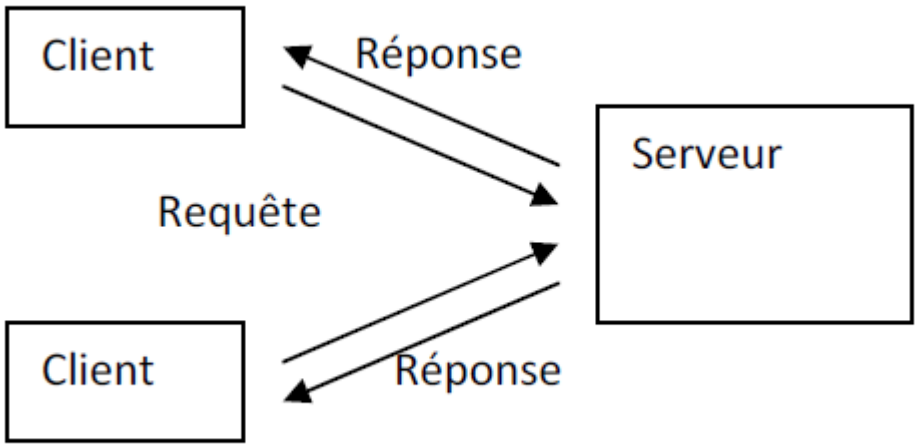


Fig. I.9. Architecture client/serveur

I.7.6. Avantages et inconvénients

I.7.6.1. Avantages de l'architecture client/serveur

L'intérêt des réseaux Client/serveur est de réunir deux avantagescomplémentaires, l'indépendance et la centralisation :

- **Flexibilité :** il est aisé de supprimé, modifier ou ajouté des entités(Clients) au réseau sans perturber son fonctionnement.
- **Exploitation:** il est facile d'utilisation (réparation et administration)
- **Centralisation:** elle permet de centraliser les informations de la base dedonnées et de répondre à un grand nombre de requêtes simultanées de la part des clients.

I.7.6.2. Inconvénients de l'architecture client/serveur

Tout fois, L'architecture client/serveur présente quelques défauts dont :

- Un coût élevé dû à la technicité du serveur.
- Un maillon faible : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour delui.

Conclusion

Tout au long de ce chapitre nous avons pu présenter certains concepts des réseaux informatiques tel que : sa définitions, ses mode de connexion et ses modèles de références, ainsi que l'architecture client/serveur…

Les différents concepts traités dans ce chapitre nous aiderons à mieux comprendre notre mode d'opération et les notions fondamentales pour mener à bien notre projet. Le chapitre suivant, est consacré au cluster.

Partie II: Sécurité informatique

II.1. Introduction :

Les réseaux informatiques sont composés de machines hétérogènes et ils deviennent de plus en plus complexes car les technologies évoluent très rapidement, De plus, ces réseaux sont "ouverts" du fait qu'ils sont pour la plupart raccordés à Internet.

Le nombre d'ordinateurs de particuliers connectés sur Internet est constamment en croissance. Ceci s'explique par la baisse des prix des ordinateurs, de l'accès à Internet (accès ADSL par exemple) et aussi par l'augmentation des performances de ceux-ci (puissance CPU, mémoire, capacité de stockage, débit Internet important avec une connexion permanente).
Cette merveilleuse avancée technologique, qui permet de faciliter la communication, a malheureusement engendré des risques importants dans le domaine de la sécurité informatique.

Toutes ces machines hétérogènes sont potentiellement vulnérables à des attaques réalisées par des pirates, les conséquences de ces attaques peuvent être lourdes pour un particulier (pertes d'information ou pire encore vol d'information, atteinte à la vie privée ...) et pour une entreprise (perte du savoir-faire, atteinte à l'image de marque, perte financière ...).
Par conséquent, il est important d'être conscient de ces menaces et de prendre des mesures adéquates afin de s'en protéger.

II.2. Définition :

La sécurité d'un réseau est un niveau de garantie que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs dites machines possèdent uniquement les droits qui leur ont été octroyé.
Il peut s'agir :

- D'empêcher des personnes non autorisées d'agir sur le système d'une façon malveillante.
- D'empêcher les utilisateurs d'effectuer des opérations involontaires capables denuire au système.
- De sécurisé les données en prévoyant les pannes.
- De garantir le non interruption d'un service.

II.3. La sécurité informatique :[08]

Aujourd'hui, la communication par voie informatique est omniprésente, même dans les secteurs sensibles comme le secteur bancaire, judiciaire, ou militaire. Lors de la création des premiers réseaux informatique, la sécurité avait une importance secondaire.

Un des plus anciens protocoles dans l'histoire de l'internet est un protocole simple de transfert de courrier SMTP. Il ne permettait pas l'authentification de l'expéditeur, et ne s'occupait pas du chiffrement de message. Ce problème n'a pas été résolu car la plupart des agents de transfert de courrier acceptent encore du courrier non authentifié. Aujourd'hui, il est encore extrêmement facile d'envoyer un mail sous un autre nom et une autre adresse de messagerie, même avec de simples logiciels de messagerie.

Pour transmettre des informations de manière sécurisée vie un canal non sur comme le réseau Internet ; il est donc nécessaire de prévoir des mécanismes de sécurité de bout en bout.

Lors de l'échange d'un message entre un expéditeur et un destinataire, on plusieurs notions de sécurité réseau.

- ✓ **La confidentialité :** Le fait de garder secret le contenu de message.
- ✓ **L'intégrité :** s'assurer que le message n'a pas été modifié. En générale on ne peut pas garantir l'intégrité d'un message mais bien contrôler son intégrité.
- ✓ **La non-répudiation :** s'assurer de l'identité de l'auteur du message.
- ✓ **L'authentification :** garder une preuve de l'authenticité du message, que l'émetteur ne peut nier avoir envoyé.

II.4 Terminologie de la sécurité informatique

La sécurité informatique utilise un vocabulaire bien défini que nous utilisons dans nosarticles.de manière à bien comprendre ces articles, il est nécessaire de définir certainstermes:

- ➢ **Les vulnérabilités :**

Ce sont les failles de sécurité dans un ou plusieurs systèmes. Tout système vudans sa globalité présente des vulnérabilités, qui peuvent être exploitables ounon.

- ➢ **Les attaques (exploits):**

Elles représentent les moyens d'exploiter une vulnérabilité. Il peut y avoirplusieurs attaques pour une même vulnérabilité mais toutes les vulnérabilités nesont pas exploitables.

- ➢ **Les contre-mesures :**

Ce sont les procédures ou techniques permettant de résoudre une vulnérabilitéou de contrer une attaque spécifique (auquel cas il peut exister d'autres attaquessur la même vulnérabilité).

- **Les menaces :**

Ce sont des adversaires déterminés capables de monter une attaque exploitantune vulnérabilité.

II.5 Malveillance informatique :

Parmi les multiples procédés d'attaque contre le système d'information, il convientde réserver une place spéciale à une famille de logiciels malveillants(les anglophones ont créé à leur intention le néologisme *malware*) quise répandent en général par le réseau, soit par accès direct à l'ordinateur attaqué,soit cachés dans un courriel ou sur un site Web attrayant, mais aussi éventuellementpar l'intermédiaire d'une disquette, d'une clé USB ou d'un CD-Rom. Ladestination de ces logiciels est de s'installer sur l'ordinateur dont ils auront réussi àvioler les protections pour y commettre des méfaits, et aussi pour se propager versd'autres victimes.

II.5.1 Les ennemis

II.5.1.1 Pirates informatiques (hackers)

Ce terme générique et trop souvent source de fantasmes s'applique aux passionnés d'informatique s'amusant à accéder aux ordinateurs et aux réseaux d'autres personnes.De nombreux pirates se contentent simplement d'accéder aux réseaux et d'y laisser leur "empreinte" sous la forme d'applications divertissantes ou de messages sur le bureau de l'ordinateur.D'autres, souvent appelés "braqueurs", sont plus dangereux et mettent en panne des systèmes informatiques entiers, volent ou endommagent des données confidentielles, détériorent des pages Web et vont même jusqu'à interrompre l'activité.

Certains pirates amateurs se procurent simplement des outils de piratage en ligne et les déploient sans vraiment comprendre leur fonctionnement ou leurs effets.

II.5.1.2 Personnel non avisé :

Il arrive souvent que des employés, concentrés sur leurs activitésprofessionnelles spécifiques outrepassent les règles de base de sécurité du réseau. Ilspeuvent, par exemple, choisir des mots de passe faciles à mémoriser afin de seconnecter aisément au réseau. Ces mots de passe sont alors faciles à deviner ou àforcer par les pirates, à l'aide d'un utilitaire logiciel de " cracking " largement répandu.

Les employés peuvent involontairement être la source de failles dans la sécurité,y compris la contamination accidentelle par des virus informatiques et leur propagation. L'un

des modes les plus classiques d'infection par un virus est la contamination à partir d'une disquette ou d'un téléchargement de fichiers depuis
Internet. Les employés qui échangent des données au moyen de disquettes peuvent involontairement infecter les réseaux de leur entreprise avec des virus contractés sur d'autres ordinateurs.

Les entreprises risquent également d'être infectées lorsque leurs employéstéléchargent des fichiers tels que des présentations PowerPoint,

Enfin, les employés, experts ou novices en informatique, peuvent faire des erreurs et mal installer un logiciel de protection Contre les virus ou ignorer accidentellement des avertissements relatifs à la sécurité.

II.5.2. Les attaque :

II.5.2.1 Les Buts des attaques :

- Interruption : vise la disponibilité des informations.
- Interception : vise la confidentialité des informations.
- Modification : vise l'intégrité des informations.
- Fabrication : vise l'authenticité des informations.

II.5.2.2 Les types d'attaques :

II.5.2.2.1 L'usurpation d'adresse IP :

L'usurpation d'adresse IP (spoofing IP) est une technique consistant à remplacer l'adresse IP de l'expéditeur d'un paquet IP par l'adresse IP d'une autre machine.
Cette technique permet ainsi à un pirate d'envoyer des paquets anonymement, il ne s'agit pas pour autant d'un changement d'adresse IP, mais d'une **mascarade** de l'adresse IP au niveau des paquets émis.

Ainsi, certains tendent à assimiler l'utilisation d'un proxy (permettant de masquer d'une certaine façon l'adresse IP) avec du spoofing IP. Toutefois, le proxy ne fait que relayer les paquets. Ainsi même si l'adresse est apparemment masquée, un pirate peut facilement être retrouvé grâce au fichier journal (logs) du proxy

II.5.2.2.2 Scanner d'adresse IP :

Différents logiciels existent pour scanner un réseau, le plus simple se contentent se scanner toutes les adresse IP et tous les ports d'un réseau.

Un scanner de vulnérabilité est capable de déterminer les ports ouverts sur un système en envoyant des requêtes successives sur les différents ports et analyse les réponses afin de déterminer lesquels sont actifs.

En analysant très finement la structure des paquets TCP/IP reçus, les scanners de sécurité évolués sont parfois capables de déterminer le système d'exploitation de la machine distante ainsi que les versions des applications associées aux ports et, le cas échéant, de conseiller les mises à jour nécessaires.

II.5.2.2.3 Les virus :[09]

Les virus sont des programmes autonomes conçus pour se reproduire et se diffuser de manière autonome.

Deux éléments caractérisent un virus :

-La façon dont il se reproduit et infecte le système informatique

-Les différente actions qu'il va réaliser.

Il existedifférents types de virus **:**

- **Les vers** sont des virus capables de se propager à travers un réseau.
- **Les chevaux de Troie** (troyens) 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).
- Les **bombes logiques** sont des virus capables de se déclencher suite à un événement particulier (date système, activation distante, ...).
- Un **keylogger** (enregistreur de touches) est un dispositif chargé d'enregistrer les frappes de touches du clavier et de les enregistrer, à l'insu de l'utilisateur.

Depuis quelques années un autre phénomène est apparu, il s'agit des canulars (en anglais hoax), c'est-à-dire des annonces reçues par mail (par exemple l'annonce de l'apparition d'un nouveau virus destructeur ou bien la possibilité de gagner un téléphone portable gratuitement) accompagnées d'une note précisant de faire suivre la nouvelle à tous ses proches. Ce procédé a pour but l'engorgement des réseaux ainsi que la désinformation

II.5.2.2.4 L'interception des informations confidentielles :

Il faut savoir que dans une entreprise ; sur une connexion réseau et/ou Internet, il ya un très grands pourcentage que les informations qui circulent ne sont pas cryptées et peuvent donc être interceptés par n'importe qui.

Il existe de nombreux logiciels permettant d'enregistrer, puis de consulter tout ce qui se passe sur un réseau informatique (ces logiciels sont souvent appelés « packet sniffer » ;

On peut donc y trouver : les mots de passes saisis, les pages web consultés, les documents transmis, les emails envoyés, etc.…

A chaque paquet de données émis sur le réseau par un ordinateur, tous les ordinateurs situés sur le même réseau le reçoivent systématiquement.

Chaque ordinateur regarde l'adresse du destinataire de ce paquet d'informations et le compare à sa propre adresse.

Si les deux adresses ne correspondent pas, c'est-à-dire si l'ordinateur s'aperçoit que le paquet d'informations ne lui est pas destiné, il va l'ignorer. Seule la machine à qui ce paquet d'informations est destiné va en tenir compte et y répondre.

On constate cependant que de nombreuse machine reçoivent les données qu'un ordinateur émet, même si ces données ne leurs sont pas destinés.

Dès lors, il est facile d'utiliser certains logiciels qui vont lire et conservé tous les paquets de données, même ceux qui ne sont pas destinés à la machine.

II.5.2.2.5 Le déni de Service :[10]

L'attaquant n'obtient pas un accès au système informatique de l'entreprise, mais il parvient à mettre en panne certains composants stratégiques (le serveur de messagerie, le site web etc...).

C'est une méthode n'ayant qu'un seul but : mettre hors service tout ou partie de votre système.

Il existe différents types de déni de services :

II.5.2.2.5.1. Générer un dysfonctionnement de l'application en utilisant une faille du logiciel :

Le pirate envoie à votre serveur des informations dans le format est volontairement anormale.

II.5.2.2.5.2 Saturer le service :

La technique consiste à envoyer à l'application cible un grand nombre de requêtes, cela ne génère pas directement une panne, mais l'application est tellement saturés par le travail de réponses aux requêtes factices, que le temps de réponse aux requête légitimes se dégradent au point que l'application devient inutilisable.

II.5.2.2.5.3 Le déni de services distribués :

Le déni de service distribués consiste à réaliser la même opération que précédemment mais simultanément sur un grand nombre de machine différents, l'auteur du déni de service charge le même programme d'attaque sur plusieurs machine, les programmes sont conçue pour se déclencher à une date et heure prédéterminé .Ainsi à l'heure dite le serveur cible reçoit des millions de requêtes venant de nombreuse sources différentes.

II.5.2.2.5.4 Le déni de service par Virus :

Cette fois ci l'auteur de déni de service ne procède pas sur des machines extérieures pouvant lui servir de relais, mais à la place, il crée un virus qu'il lance sur internet, le virus se reproduit de manière autonome et infecte un très grand nombre de machine dans le monde.

II.6 Mécanismes de sécurisation :

II.6.1 Cryptographie:[11]

Les récents développement de la cryptographie permettent de résoudre les nombreux problèmes menaçants la vie privé ou la sécurité sur internet, la cryptographie est l'étude des principes, méthodes et techniques mathématiques reliées aux aspects de sécurité de l'information telle la confidentialité, l'intégralité des données, authentification d'entités, et l'authentification de l'originalité des données. C'est un ensemble de techniques qui fournit la sécurité de l'information.

La cryptographie vous permet de stocker des informations sensibles ou de les transmettre à travers des réseaux non sûrs (comme Internet) de telle sorte qu'elles ne peuvent être lues par personne à l'exception du destinataire convenu.

La cryptographie repose sur quatre concepts généraux : le texte en claire, l'algorithme cryptographique, le texte codé et la clé.

- **Le texte en claire** : est le message sous sa forme originale, que ce soit avant le chiffrement ou après le déchiffrement.
- **L'algorithme Cryptographique** est l'opération mathématique utilisé pour chiffrer ou déchiffrer le message.
- **Le texte codé** : est le message après qu'il a été chiffré. Le message du texte codé apparait confus et intangible aux personnes auxquels il n'est pas destiné.
- **La clé** : est un ensemble unique de caractère (tel qu'un nombre ou un mot de passe) que le chiffre utilise pour chiffrer et déchiffrer le message.

II.6.1.1Signature électronique :

Le principe de la signature numérique consiste à appliquer une fonction mathématique sur une portion du message. Cette fonction mathématique s'appelle fonction de hachage 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 code de hachage 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 à la clé publique de la source puis il compare ce code à un autre code qu'il calcule grâce au message reçu. Si les deux correspondent, 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 fût amélioré avec la mise en place de certificats permettant de garantir la validité de la clé publique fournie par l'émetteur.

Exemple d'algorithme de signature :
Exemple SHA, MD4, MD5

II.6.1.2Les certificats : [12]

Pour assurer l'intégrité des clés publiques, les clés publiques sont publiées avec un certificat. Un certificat (ou certificat de clé publique) est une structure de données qui est numériquement signée par une autorité certifiée (CA : Certification Authority) une autorité en qui les utilisateurs peuvent faire confiance.

Il contient une série de valeurs, comme le nom du certificat et son utilisation, des informations identifiant le propriétaire et la clé publique, la clé publique elle-même, la date

d'expiration et le nom de l'organisme de certificats. Le CA utilise sa clé privée pour signer le certificat et assure ainsi une sécurité supplémentaire.

Si le récepteur connaît la clé publique du CA, il peut vérifier que le certificat provient vraiment de l'autorité concernée et assuré que le certificat contient donc des informations viables et une clé publique valide.

Les certificats permettent aux utilisateurs et aux serveurs web de s'authentifier mutuellement avant d'établir une connexion. Ils contiennent des valeurs de cryptage (clés) qui permettent d'établir une connexion SSL (Secure Sockets Layer) entre le client et le serveur.

II.6.2.Logiciels antivirus :

La plupart des ordinateurs sont dotés d'un logiciel antivirus pré-intégré capablede détecter et effacer les principales menaces virales s'il est régulièrement mis à jour etcorrectement entretenu.

Avec des milliers de nouveaux virus générés chaque mois, il est crucial que labase de données des virus soit tenue à jour. La base de données des virus estl'enregistrement du logiciel d'antivirus qui permet d'identifier les virus connuslorsqu'ils surviennent.

La politique de sécurité du réseau doit mentionner que tous les ordinateurs duréseau doivent être tenus à jour et théoriquement qu'ils doivent tous être protégés par lemême système d'antivirus (entre autres, afin de réduire au maximum les frais demaintenance et de mise à jour).

II.6.3. Parfeu : [13]

Les par feu ont pour mission de contrôler les données qui sortent et qui entrent dans le réseau. C'est la première mesure à mettre pour sécuriser un réseau.

L'emplacement du par feu dans le réseau est fondamentale, il doit être situé à l'endroit où en est sûr qu'il va pouvoir intercepter toutes les données, généralement entre l'arrivé de la liaison internet et le réseau.

Par ailleurs il faut veiller à protéger tous les points du réseau local qui peuvent être émis en contact avec le réseau extérieur .Il est donc parfois nécessaire d'installer plusieurs par feu dans l'entreprise. Et aussi faut faire attention, si des utilisateurs disposent d'un modem sur leur ordinateur ils présentent un risque potentiel, s'ils l'utilisent pour se connecter à internet, ils ouvrent une porte non sécurisé à toutes les personnes malveillante.

Une connexion Modem peut être utilisée de manière bidirectionnelle, par l'utilisateur pour consulter internet, mais également pour les pirates informatique pour entrer sur votre

réseau. Par contre s'il est indispensable que l'utilisateur puisse employer son modem, il faudra alors installer un logiciel par feu sur chaque ordinateur présente un risque.

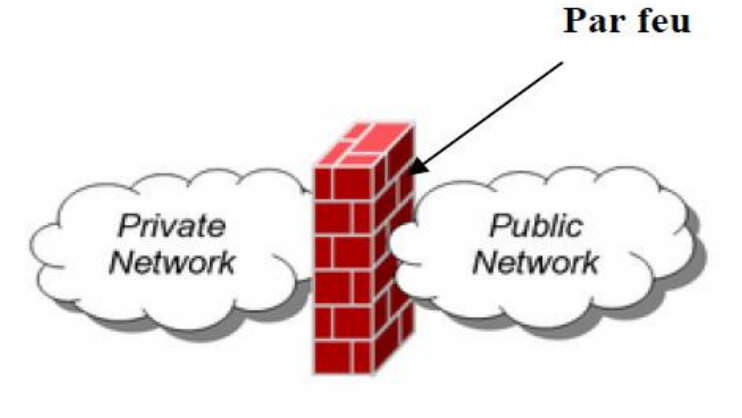


Fig.I.10 : Architecture générale d'un par feu

II.6.4. Parfeu Mots de passe :

Le moyen le plus simple et le plus classique de s'assurer que seules les personnes autorisées peuvent accéder à une certaine partie du réseau est de protéger certaines zones du réseau par un mot de passe.

De nombreux utilisateurs choisissent des chiffres ou des mots faciles à retenir pour leurs mots de passe, comme des dates d'anniversaires, des numéros de téléphone ou des noms d'animaux de compagnie, d'autres ne changent jamais leurs mots de passe et ne se soucient pas de leur confidentialité.

Il est aisément compréhensible que plus un mot de passe soit long, plus il est difficile à décrypter. D'autre part, un mot de passe avec uniquement des chiffres sera beaucoup plus simple à décrypter qu'un mot de passe contenant des lettres.

II.6.5.L'analyse des journaux : [14]

Pour permettre la sauvegarde des flux réseau entrant et sortant, il faut effectuer des opérations d'enregistrement sur fichier, des fichiers logs.

Un fichier log est un fichier sous forme ASCII (américain standard code for information and inter change). Le fichier log est porteur d'informations essentielles qui concernent toutes les activités du serveur, car ce fichier est créé en général au niveau du serveur par un logiciel. Le contenu d'un fichier log et fonction du type d'activité du serveur. Un fichier log d'un serveur du réseau par exemple doit contenir la trace du trafic des réseaux entrant et sortant durant tout le temps de son activité.

Plusieurs formats existent pour les fichiers logs, le contenu des fichiers log dépend de l'activité du serveur et du niveau d'enregistrement.

Par ailleurs, la de notation du fichier log veux dire <<le journal de bord des connexions>>.

C'est en quelque sorte un fichier qui sert d'historique pour administrateur réseau, et qui renferme la trace de toute les requêtes et réponse concernant le serveur sur lequel la journalisation est effectué.

II.6.6. Proxy :

Le but d'un serveur Proxy est d'isoler une ou plusieurs machines pour les protéger, comme indiqué sur le schéma :

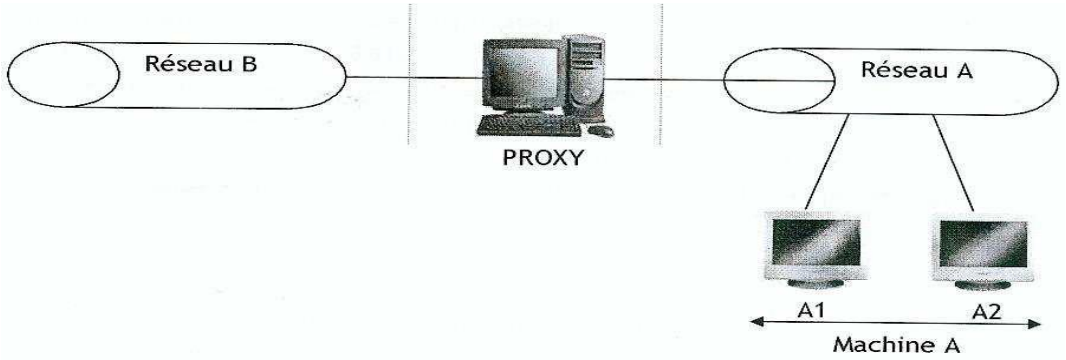


Fig.I.11 : Fonctionnement d'un Proxy.

Les machines A doivent se connecter au réseau par l'intermédiaire du serveur Proxy. Ce dernier sert de relais entre le réseau et les machines à cacher. Ainsi, les machines du réseau B auront l'impression de communiquer avec le Proxy, et non les machines A.

Pour les applications du réseau B, l'adresse IP du client sera celle du serveur Proxy. Par exemple, lors d'une connexion à un serveur http, le browser se connecte au serveur

Proxy et demande l'affichage d'une URL. C'est le serveur Proxy qui gère la requête et qui renvoie le résultat à votre browser.

Ainsi, en utilisant un numéro de port différent, le Proxy oblige toutes les requêtes à passer par lui en supprimant les trames dont le numéro de port ne lui correspond pas. De plus, le Proxy possède un avantage supplémentaire en termes de performance. Si deux utilisateurs demandent à peu de temps d'intervalle la même page, celle-ci sera mémorisée dans le Proxy, et apparaîtra donc beaucoup plus rapidement par la suite. Ce procédé est très intéressant en termes de sécurité sur Internet, les machines sont protégées. Le serveur Proxy peut filtrer les requêtes, en fonction de certaines règles.

II.6.7. Translation d'adresse : [15]

La translation d'adresse consiste à utiliser une seul adresse IP publique officiel qui sera attribué à votre par feu ou bien votre proxy, vous attribuez ensuite des adresse IP privées à chaque machine de votre réseau local.

Les adresse privées ne sont routées par le réseau internet, Si une machine de votre réseau local possédant une adresse privée telle que 192.168.0.0 par exemple, souhaite communiquer avec un serveur se trouvant sur internet, elle pourra le joindre, mais celui-ci ne pourra pas répondre car le paquet de retour sera détruit par les routeurs du réseau Internet.

Avec la translation d'adresse le proxy ou le par feu transforme en temps réel les adresse privés en adresse publique, puis au retour de la réponse, l'adresse publique en adresse privé.

Ainsi même si vous avez plusieurs poste utilisateurs disposant chacune d'une adresse privé différente, vu de l'extérieur, c'est comme s'il n'y avait qu'un seul utilisateur ans l'entreprise.

- **Avantages en termes de sécurité de la translation d'adresse :**

  L'avantage d'utiliser une translation d'adresse pour votre réseau local réside dans le fait que ces adresses privées ne seront pas utilisables sur internet.

  Même si votre par feu est mal réglé, un pirate qui essaierait de communiquer avec une machine utilisateur de votre réseau local ayant une adresse IP privé n'obtiendrait aucune réponse car tous les paquets qu'il enverrait serait détruit par le réseau, les paquets n'arriverai même pas jusqu'à votre par feu.

  En revanche, les utilisateurs du réseau local n'auront aucun problème pour communiquer vers le réseau Internet car tous les paquets sortant seront traités par la translation d'adresse.

  Par ailleurs si votre par feu ou bien votre proxy cesse de fonctionner ou bien s'arrête complètement pour une raison ou une autre, la fonction de translation d'adresse s'arrêtera en même temps, de ce fait aucune machine n'arriverait à communiquer avec l'internet.

II.6.8. VPN(Virtual Private Network):

Un réseau VPN repose sur un protocole appelé "protocole de tunneling", ce protocole permet de faire circuler les informations de l'entreprise de façon cryptée d'un bout à l'autre du tunnel. Ainsi, les utilisateurs ont l'impression de se connecter directement sur le réseau de leur entreprise.

Le principe de tunneling consiste à construire un chemin virtuel après avoir identifié l'émetteur et le destinataire. Par la suite, la source chiffre les données et les achemine en empruntant Ce chemin virtuel. Afin d'assurer un accès aisé et peu coûteux aux intranets ou aux extranets d'entreprise, les réseaux privés virtuels d'accès simulent un réseau privé, alors qu'ils utilisent en réalité une infrastructure d'accès partagée, comme Internet. Les données à transmettre peuvent être prises en charge par un protocole différent d'IP. Dans Ce cas, le protocole de tunneling encapsule les données en ajoutant un en-tête. Le tunneling est l'ensemble des processus d'encapsulation, de transmission et de dé encapsulation.

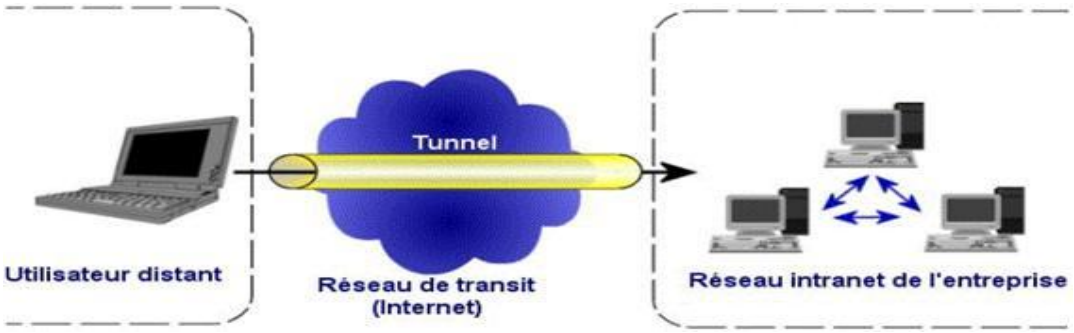


Fig.I.12:Les VPN (Virtual Privat Network)

Le VPN est utilisé afin de permettre à un hôte distant d'accéder à l'intranet de son entreprise ou à celui d'un client grâce à l'internet, et en plus les VPN peuvent également être utilisé à l'intérieur même de l'entreprise, sur l'intranet, pour l'échange des données confidentielles.

- **Caractéristiques fondamentales d'un VPN :**

Un système de VPN doit pouvoir mettre en œuvre les fonctionnalités suivantes :

- **Authentification d'utilisateur** : Seuls les utilisateurs autorisés doivent pouvoir s'identifier sur le réseau virtuel. De plus, un historique des connexions et des actions effectuées sur le réseau doit être conservé.

- **Gestion d'adresses :** Chaque client sur le réseau doit avoir une adresse privée. Cette adresse privée doit rester confidentielle. Un nouveau client doit pouvoir se connecter facilement au réseau et recevoir une adresse.

- **Cryptage des données :** Lors de leurs transports sur le réseau public les données doivent être protégées par un cryptage efficace.

- **Gestion de clés :** Les clés de cryptage pour le client et le serveur doivent pouvoir être générées et régénérées.

- **Prise en charge multi protocole** : La solution VPN doit supporter les protocoles les plus utilisés sur les réseaux publics en particulier IP.

II.6.9Systèmes de détection d'intrusions :

La détection d'intrusion est 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.

Les IDS sont des systèmes qui collectent des informations de différente manière, soit à partir du système ou du réseau, sur les différentes activités se rapportant à ce système pour les analyser à la recherche des signes d'une intrusion (attaques suspectes) et alerter dans le cas positif.

Il existe deux grandes familles distinctes d'IDS :

- Les **N-IDS** (Network Based Intrusion Détection System), ils assurent la sécurité au niveau du réseau.
- Les **H-IDS** (Host Based Intrusion Détection System), ils assurent la sécurité au niveau des hôtes.

II.7 Conclusion :

Nous avons consacrés ce chapitreà décrire d'une manière générale les concepts de base de réseaux informatiques et leur sécurité, pour cela on a divisé ce chapitre en deux parties généralitéssur les réseaux informatiques et sécurité réseaux où on a constaté que ce dernier joue un rôle primordiale dans les réseaux informatiques à savoir son efficacité et ses différentes techniques de protection contre les menaces informatiques.

Chapitre II

Les concepts faudamenteaux

de la cryptographie

Première partie : La cryptographie

I.1. Introduction

Dans la société de l'information d'aujourd'hui, l'usage de la cryptologie s'est banalisé. On le retrouve quotidiennement avec les cartes bleues, téléphones portables, internet ou encore les titres de transport. La cryptologie moderne a pour l'objet l'étude des méthodes qui permettent d'assurer les services d'intégrité, d'authenticité et de confidentialité dans les systèmes d'information et de communication. La cryptographie moderne est basée sur les mathématiques pour sécuriser l'information.

I.2. La cryptologie

La cryptologie, ou étude du secret, est divisée en deux branches : la cryptographie et la cryptanalyse. Alors que la cryptographie définit les systèmes, la cryptanalyse tente de trouver les failles dans ces systèmes.

Le début de la cryptographie remonte à l'Antiquité. Le chiffrement de César est l'un des plus anciens. Si dans le passé, on utilisait la cryptographie principalement dans le but d'échanger des informations confidentielles, aujourd'hui, on veut également s'assurer de l'intégrité et de l'authenticité des données, sans que celles-ci ne soient nécessairement confidentielles.

I.2.1. La cryptographie

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

I.2.2. Cryptanalyse

La cryptanalyse est la science de reconstitution du texte en claire sans connaitre la clé. Une cryptanalyse réussie peut nous fournir le texte en claire ou la clef, comme elle peut mettre en évidence les faiblesses d'un crypto système qui peuvent éventuellement faciliter les attaques contre celui-ci. On peut dit alors que l'algorithme de chiffrement a été cassé.

On distingue quatre méthodes de cryptanalyse :

- ✓ Une attaque sur un texte chiffré consiste à retrouver la clé de déchiffrement à partir d'un texte chiffré ou plusieurs ;
- ✓ Une attaque sur un texte connu consiste à retrouver la clé de déchiffrement à partir d'un texte chiffré ou plusieurs, connaissant le texte en clair correspondant ;
- ✓ Une attaque sur un texte connu consiste à retrouver la clé de déchiffrement à partir d'un texte chiffré ou plusieurs, l'attaquant ayant la possibilité de les générer à partir d'un texte en clair ;
- ✓ Une attaque sur un texte chiffré consiste à retrouver la clé de déchiffrement à partir d'un texte chiffré ou plusieurs, l'attaquant ayant la possibilité de les générer à partir d'un texte en clair

I.2. 3. Stéganographie

Sert à cacher des messages secrets dans d'autre messages, généralement l'expéditeur écrit un message inoffensif et dissimile un massage secret dedans.

I.2.3.1. Le texte caché

J'aime	garder	le	secret
Envoyer	un	message	d'amour
Voici	papa	arrive	enfin
Chant	un	morceau	rare

J'aime	garder	le	**secret**	
Envoyer	un	**message**	d'amour	Clé : CDBA
Voici	papa	arrive	enfin	Le texte caché est :
Chant	**un**	morceau	rare	**Voici un message secret**

I.2.3.2. Les fichiers images cachées

Il possible de cacher de l'information dans des fichiers images de la même manière que le texte. En modifiant quelques bits du fichier, il est aussi possible de cacher un message de son choix sans que cela se voit. En effet cette altération sera peu ou pas visible car l'œil

humain n'est pas capable de discerner des petites aberrations sur une grande image à condition que le ratio du message ou la taille de l'image ne soit pas trop grand.

On peut conclure que la stéganographie est une méthode de cryptographie très fiable : elle repose uniquement sur le fait que personne ne remarquera le canal caché. Dès que ce canal est connu, il n'y a plus aucune protection.

I.3. Cryptographie classique

La cryptographie classique elle repose sur plusieurs algorithmes ou méthodes de cryptographie à l'époque où les mathématiques ne régnaient pas encore sur ce domaine.

I.3.1. Cryptographie par transposition

Cette méthode de chiffrement datée entre le Xème et VIIème siècle avant jésus Christ repose sur l'utilisation d'un bâton d'un diamètre fixé. Une lanière en cuire était enroulée en hélice autour de ce bâton et le texte en claire était alors écrit sur la lanière. Ensuite, la lanière était déroulée et pouvait être envoyée (sans le bâton) au destinataire.

Pour déchiffrer le texte chiffré, il suffisait d'utiliser un bâton de même diamètre que le précédent, dès que la lanière est enroulée le texte en clair pouvait être relu.

I.3.2. Cryptographie par substitution

Les systèmes de cryptographie par substitution sont considérés comme des applications bijectives des lettres de l'alphabet des messages clairs sur des lettres de l'alphabet des cryptogrammes : on remplace des caractères par d'autres. On distingue :

I.3.2.1. Chiffrement monoalphabétique

On parle de chiffrement monoalphabétique lorsque chaque lettre du message claire est toujours remplacée par le même symbole. On aura une bijection entre les lettres clairs et les symboles de l'alphabet de chiffrement.

Ce chiffrement était utiliser par les Romains, sous le nom « chiffrement de César », il suffit de décaler les lettres d'un certain nombre connu aussi bien par celui qui écrit le message que par celui qui le reçoit.

Par exemple si n=5, cela donne :

a=f, b=g, c=h, d=i, …, y=d, z=e.

Si le texte en clair est « chiffrement monoalphabétique », avec un décalage de cinque lettres, le texte chiffré est « HMNKKWJRJSY RTSTFQUMFGJYNVZJ».

I.4. Cryptographie moderne

Avec le temps, les cryptographes ont pris conscience qu'il n'était pas réaliste de faire reposer un système de chiffrement sur l'hypothèse que un attaquant ne connais pas la méthode utilisé. En conséquence ils ont introduit un nouveau type de cryptographie dans lequel la sécurité des algorithmes de chiffrements repose uniquement sur le secret de la clé de déchiffrement, il s'agit de la cryptographie moderne.

La cryptographie moderne repose aujourd'hui uniquement sur les mathématiques. Ou les règle de bases sont :

- ✓ L'algorithme utiliser n'est pas secret, il peut être diffusé librement cela ne peut pas avoir impact sur la fiabilité ou non à déchiffrer le message.
- ✓ La clé de chiffrage utilisé est secrète

Si un protocole cryptographique basé sur la non-divulgation de l'algorithme mathématique utiliser n'est pas fiable. Un jour l'algorithme utilisé sera connu et le protocole deviendra faible. Au contraire, diffuser l'algorithme mathématique utilisé permet à la communauté de valider et de tester la robustesse de cet algorithme.

I.4.1. Objectif de la cryptographie

Les principaux objectifs à garantir par l'application de la cryptographie sont:

- **Confidentialité:** Un mécanisme pour transmettre des données de telle sorte que seul le destinataire autorisé puisse les lire.

- **Intégrité des données:** Un mécanisme pour s'assurer que les données reçues n'ont pas été modifiées durant la transmission, frauduleusement ou accidentellement.

- **Authentification:** Un mécanisme pour permettre d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources aux seules personnes autorisées (un mot de passe par un nom de login ou un certificat numérique).

- **Non-répudiation:** Un mécanisme pour enregistrer un acte ou un engagement d'une personne ou d'une entité de telle sorte que celle-ci ne puisse pas nier avoir accompli cet acte ou pris cet engagement. Ce mécanisme se décompose:
  - non-répudiation d'origine l'émetteur ne peut nier avoir écrit le message.
  - non-répudiation de réception le receveur ne peut nier avoir reçu le message.
  - non-répudiation de transmission l'émetteur du message ne peut nier avoir envoyé le message.

I.4.2. Techniques de cryptographie

Pour assurer les objectifs de la cryptographie on propose deux techniques complémentaires :

- Le chiffrement.
- La signature.

I.4.2.1. Le chiffrement[16]

Le chiffrement est de transformer un message, appelé texte clair, en texte chiffré qui ne sera pas lisible que par son destinataire légitime. Cette transformation est effectuée par une fonction de chiffrement paramétrée par une clé de chiffrement. Un interlocuteur peut alors déchiffrer le message en utilisant la fonction de déchiffrement s'il connait la clef de déchiffrement correspondant. Un tel système n'est sûr que s'il est impossible à un intrus de déduire le texte clair du message chiffré.

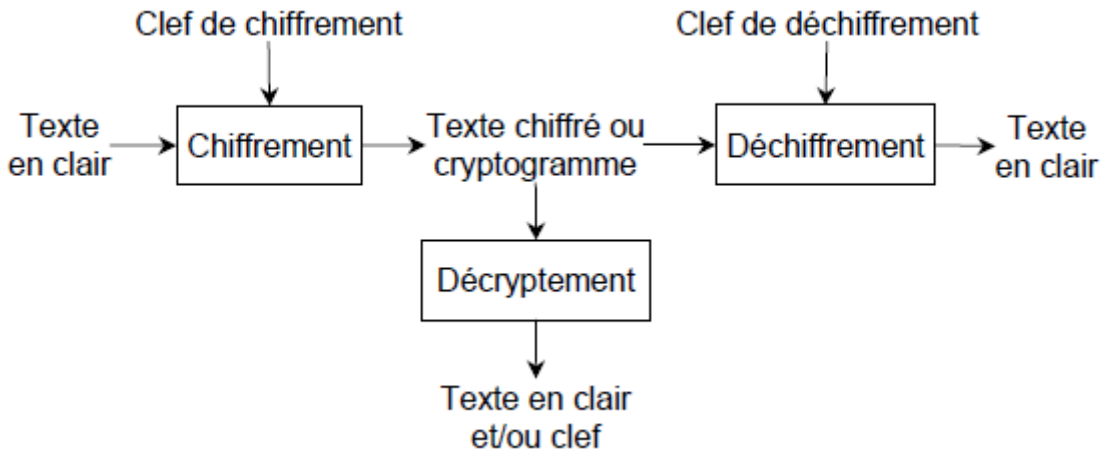


Fig.II.1 : Terminologie de chiffrement, déchiffrement et décryptement.

Cryptographie asymétrique : [16]

Dans les systèmes à clefs asymétriques, on utilise deux clefs différentes : une clef privée qui doit être gardée secrète par son propriétaire et que lui seul possède, et une clef publique qui peut être connue de tous. Le premier algorithme à clef asymétrique a été inventé par Diffie et Hellman.

Lorsqu'un expéditeur A envoie un message confidentiel, il le chiffre avec la clef publique du destinataire B. Ce dernier peut ensuite déchiffrer le message avec sa propre clef privée.

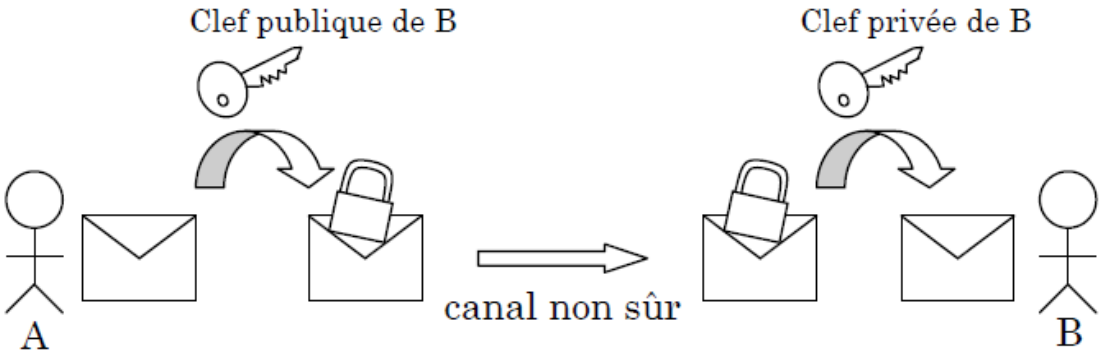


Fig. II. 2 : Chiffrement d'un message avec clef asymétrique.

Algorithme RSA :

RSA, du nom de ces inventeurs, est un algorithme de chiffrement appartenant à la famille ‘Cryptographie asymétrique’.

RSA peut être utilisé pour assurer :

- La confidentialité : seul le propriétaire de la clé privée pourra lire le message chiffré avec la clé publique correspondante.
- La non-altération (avec la clé privée). Une signature déchiffrée avec la clé publique prouvera donc l’authenticité du message.

Principe de fonctionnement de RSA

Si bob souhaite recevoir des messages en utilisant le RSA, il procède de la façon suivante :

- **Création des clés :** Bob crée 4 nombres p, q, e et d
  - ✓ p et q sont deux grands nombres premiers distincts. Leur génération se fait au hasard, en utilisant un algorithme de test de primalité probabiliste.
  - ✓ e est un entier premier avec le produit (p-1) (q-1).
  - ✓ d est tel que e d=1 modulo (p-1) (q-1). Autrement dit, ed-1 est un multiple de (p-1) (q-1). On peut fabriquer d à partir de e, p et q, en utilisant l’algorithme d’Euclide.
- **Distribution des clés :** Le couple (n, e) constitue la clé publique de Bob. Il la rend disponible par exemple en la mettant dans un annuaire. Le couple (n, d) constitue sa clé privée. Il la grande secrète.
- **Envoi du message codé :** Alice veut envoyer un message codé à Bob. Elle le représente sous la forme d’un ou plusieurs entiers M compris entre 0 et n-1. Alice possède la clé publique (n, e) de Bob. Elle calcule $C=M^e$ mode n. C’est ce dernier nombre qu’elle envoie à Bob.
- **Réception du message codé :** Bob reçoit C, et il calcule grâce à sa clé privé $D = C^e$ (mode n). D’après un théorème du mathématicien Euler, $D = M^{de} = M$ e (mode n). Il a donc reconstitué le message initial.

I.4.2.2. La signature numérique [16][17]

La signature électronique est un élément fondamental sans lequel la notion d'acte officiel sur Internet n'existerait pas. Au même titre que la signature manuscrite, la signature électronique fournit la preuve de l'authentification de l'origine d'un document, de l'intégrité d'un document, de la validité d'informations importantes comme l'horodatage, et rend impossible la répudiation d'un acte.

Lorsqu'un expéditeur A veut signer un message, il calcule l'empreinte numérique du message avec une fonction de hachage, puis chiffre cette empreinte avec sa propre clef privée. Le destinataire B peut vérifier l'intégrité et l'authenticité du message en recalculant l'empreinte, comme le montre la **Figure (3),** et en le comparant avec l'empreinte chiffrée par A qu'il déchiffre avec la clef publique de A.

Sur le plan technique, une signature électronique repose sur deux opérations mathématiques: l'élaboration d'un condensé cryptologique (aussi appelé empreinte numérique ou code de hachage), et une opération de chiffrement à l'aide d'un algorithme à clés publiques, tel que RSA.

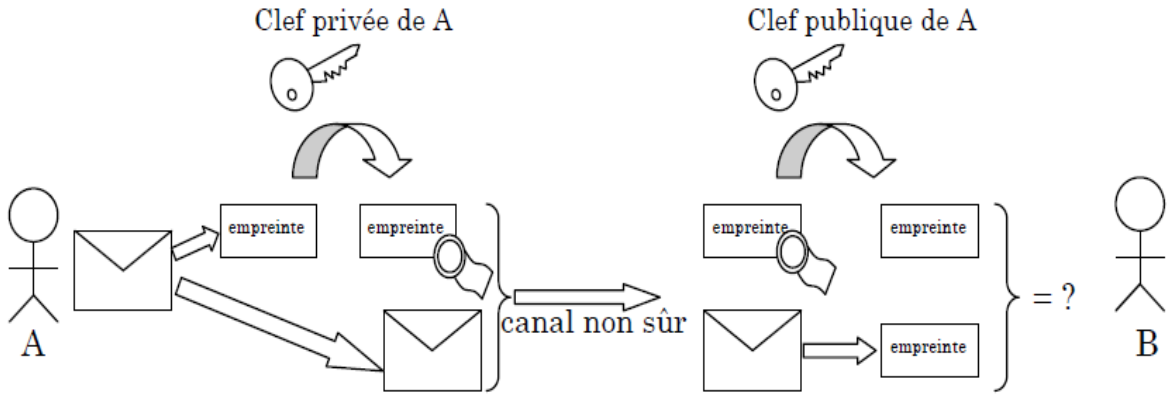


Fig.II.3 : Signature d'un message avec clef asymétrique.

I.4.2.2.1. Principes de signature numérique [17]

La signature doit donc :

- permettre d'identifier la personne expéditrice,
- garantir que le document n'a pas été modifié entre l'expédition et la réception.

Il faut donc que :

- la signature ne soit pas modifiable (identification),
- la signature ne soit pas réutilisable (non répudiation et identification),
- le document signé soit inaltérable (intégrité du document).

I.4.2.2.2. Format de signature de document

a. S/MIME

Le standard de courrier électronique sécurisé S/MIME version 2, S/MIME permet la signature ou le chiffrement de données MIME14. La signature peut se faire de deux manières :

- soit le contenu MIME est encapsulé dans la signature CMS (dans content Info).
  Dans ce cas, si le logiciel de messagerie du récepteur n'implémente pas S/MIME, le récepteur ne saura pas lire le message.
- soit le contenu MIME est détaché du message (signature externe). Cette deuxième solution est la plus utilisée car elle permet aux anciens logiciels de messagerie qui n'implémentent pas S/MIME de pouvoir lire le message. La signature se présente comme un fichier attaché.

Un problème que le standard S/MIME a dû surmonter est le fait que l'encodage du contenu MIME peut changer lors de son transfert entre serveur SMTP. Par exemple, le retour chariot n'est pas encodé de la même manière sur Windows ou Linux. Lors de la vérification de la signature, il faut procéder à un recodage selon les règles définies par S/MIME.

b. XML Signature référence

« *XML-Signature Syntax and Processing* » est un standard produit par le groupe de travail XML Signature de l'IETF/W3C. Ce standard définit une signature pour du contenu XML ou tout autre type de document (fichier binaire, etc.) et est publié sur le site officiel du W3C 150]. Il existe trois types de signature :

- une signature enveloppée : la signature XML est incluse dans le document XML signé.
- une signature enveloppante : la signature XML contient le document signé XML.
- une signature détachée : la signature fait référence à un ou plusieurs documents signés qui peuvent être de n'importe quel type.

Une signature XML a la structure suivante (où « ? » signifie zéro ou une occurrence, « + » signifie une ou plusieurs occurrences, et « * » signifie zéro, une ou plusieurs occurrences)

c. PGP [100]

PGP (Pretty Good Privacy) est un logiciel commercial qui offre des services de signature électronique et de confidentialité pour les courriers électroniques et les fichiers de données. Son fonctionnement repose également sur des opérations cryptographiques à clefs asymétriques. PGP a été créé par Philip Zimermann, et sa version 1.0 est sortie en 1991. L'IETF a publié en 1998 le standard OpenPGP (sur base de PGP 5.x), laissant la porte ouverte à d'autres implémentations libres comme *Gnu Privacy Guard* (GPG).

Les messages PGP signés et/ou chiffrés sont convertis en base 64 avant d'être transférés.

Afin de reconnaître le message PGP, un entête contenant BEGIN PGP MESSAGE est ajouté au début du message. PGP n'utilise pas ASN.1 ni les certificats X.509. Le système

PGP possède son propre format de données16. Il possède également sa propre définition de certificat.

I.5. Présentation de PGP

Le programme de cryptographie à clé publique le plus connu est PGP (pour « *Pretty Good Privacy* », en anglais : « *Assez Bonne Confidentialité*»).

Le format **OpenPGP** est le standard ouvert de cryptographie issu de PGP. Il est considéré par les cryptographes comme l'un des plus sur procédés de cryptage.

PGP permet la signature de message électronique. Les signatures sont détachées du document signé. Une application de messagerie ne supportant pas PGP pourra tout de même visualiser le message.

PGP possède une structure de certification particulière : n'importe qui peut facilement signer le certificat d'une autre personne et jouer ainsi le rôle d'autorité de certification.

I .6. Conclusion

Dans cette partie, nous avons étudié les fondements de la cryptographie et ses objectifs, et son développement moderne pour assurer la protection des informations personnelles ou privés.

Nous détaillerons dans la partie qui va suivre le principe et le fonctionnement de la fonction de hachage.

Deuxième partie : Les fonctions de hachage cryptographique

II.1. Introduction

Après avoir veut les fondements de la cryptographie, dans cette partie nous nous intéressons aux fonctions de hachage.

Nous soulignons la description, les domaines d'utilisation d'une fonction de hachage cryptographie et la construction d'une fonction de hachage et en fin en va spécifier quelque algorithmes de hachage et les propriétés de sa sécurité.

II.1. Principe des fonctions de hachage : [18]

Une fonction de hachage est une fonction prenant comme argument un élément de taille arbitraire finie et renvoyant une langueur fixée. Une illustration du principe du hachage se trouve figure 1.

Ces fonctions sont très employées notamment dans le domaine des bases de données, on parle alors parfois de tables de hachage. Elles sont particulièrement utiles dans les mécanismes d'indexation qui permettent d'améliorer considérablement les performances lors de la recherche d'éléments. Cependant, les tables de hachage diffèrent fondamentalement des fonctions de hachage cryptographiques de par les propriétés que l'on attend de ces fonctions. Les fonctions de hachage cryptographiques doivent vérifier des propriétés particulières liées à leur utilisation dans le domaine de la sécurité de l'information.

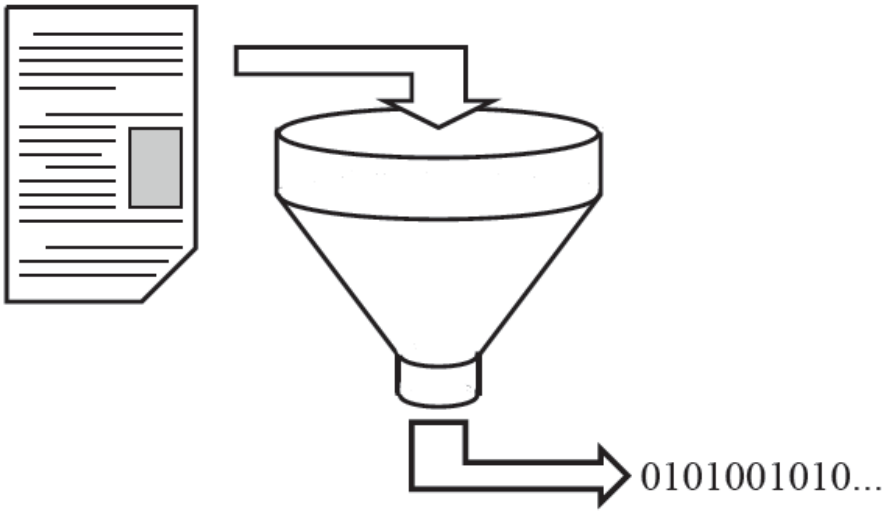


Fig.II.4 : Principe du hachage

II.2. Les fonction de hachage cryptographique :

Les fonctions de hachage d'un point de vue cryptographie, la fonction ne doit pas seulement avoir un bon comportement quand les entrées sont aléatoires, mais en va considérer des notions mettant en jeu un adversaire. Par exemple si on utilise une fonction de hachage comme somme de contrôle pour vérifier l'intégrité d'un document après un transfert, donc on envisage le cas d'un adversaire qui va introduire des erreurs volontairement pendant la transmission.

Dans ce scénario, une fonction de hachage simple, telle que la somme de tous les octets du fichier, ne protège pas le document. Charlie peut facilement créer un document quelconque dont la somme de contrôle soit la même que celle du document original, simplement en ajustant la valeur du dernier octet. La notion de sécurité dont on a besoin est ce qu'on appelle la sécurité contre les deuxièmes préimages : Charlie ne doit pas pouvoir construire un deuxième document avec la même empreinte que le document original.

Une fonction de hachage qui résiste à des attaques de ce type est appelée une fonction de hachage cryptographique. Elle est diffèrent des autres types de fonction de hachage par les propriétés de sécurité qui leur sont imposées. En effet, elles sont employées dans des domaines ou la sécurité des données traitées est critique. Une fonction de hachage cryptographique doit elle aussi être rapide à calculer. Cependant, c'est la résistance aux tentatives de manipulations malveillantes des données qui constitue l'élément principal pris en considération lors de la conception d'une telle fonction.

Une fonction de hachage *cryptographique* doit aussi satisfaire d'autres contraintes. La première stipule qu'il doit être très difficile de retrouver ou générer un texte à partir de l'empreinte (on parle alors de fonction à sens unique). Par très difficile, on entend que même avec une armée de machines dédiées à cette tâche, il sera impossible d'effectuer une telle extraction en un temps raisonnable.

II.3. Domaines d'utilisation des fonctions de hachage cryptographiques :

Les fonctions de hachage cryptographique possèdent de nombreux domaines d'utilisation, Chacun de ces domaines requiert des propriétés particulières de sécurité, nous dressons de cette section une liste non exhaustive de ces domaines.

II.3.1.Intégrité des données :

Il s'agit de la fonctionnalité principale demandée à une fonction de hachage cryptographique. Elle permet de vérifier que les données n'ont pas été altérées depuis leur création ou lors de leur transmission. Le moindre changement dans les données doit, avec une très grande probabilité, aboutir à l'obtention d'empreintes différentes. Historiquement, les premières fonctions proposé pour assurer cette fonctionnalité étaient fondées sur la théorie des codes et étaient nommées codes de détection de manipulations (MDC pour Manipulation Detection Codes).

II.3.2. Authentification de messages :

La propriété d'intégrité ne permet pas de se prémunir contre un adversaire actif qui essaierait d'altérer malicieusement les données. Un moyen de pallier ce problème consiste à procéder à l'authentification de la source des données en utilisant des codes d'authentification de messages (MAC pour Message Authentication Codes). L'objectif d'un code d'authentification de message est double : il doit permettre d'authentifier la source d'un message et de vérifier l'intégrité des données sans l'aide d'un mécanisme additionnel. L'authentification de message relève du domaine de la cryptographie symétrique, l'utilisation d'une clé secrète étant nécessaire.

II.3.3. Signature électronique :

Les schémas de signature électronique sont aucun doute l'application la plus importante des fonctions de hachage cryptographiques. Une signature électronique est un équivalent électronique d'une signature écrite. Elle permet de plus, de détecter si l'information signée a été altérée après sa signature. Les algorithmes utilisés pour signer des données nécessitent des calculs importants et sont donc relativement lents comparativement aux vitesses d'exécution des fonctions de hachage. Aussi, afin d'accélérer les procédures de signature et de vérification de signature, on utilise une fonction de hachage cryptographique pour calculer l'empreinte des données à signer et appliquer l'algorithme de signature à cette empreinte.

II.3.4. Protection de mots de passe :

Une autre application courante des fonctions de hachage cryptographiques est la protection de mots de passe. Un mot de passe est une chaine de caractère utilisé pour authentifier l'identité d'un utilisateur ou autoriser l'accès aux ressources d'un système informatique. Il est nécessaire de protéger les mots de passe afin de les stocker. Une solution courante consiste à ne stoker que leur empreinte calculée en appliquant une fonction de hachage cryptographique à une combinaison de mots de passe d'un sel(Salt).

II.3.5.Dérivation de clé :

Dans le cadre de la cryptographie symétrique, les paries partagent une clé secrète commune. Il est alors fréquent que différents clés supplémentaire soient nécessaires pour différentes application. La dérivation de clé (ou diversification de clé) consiste à générer une ou plusieurs clés à partir d'une même valeur secrète. Cette utilisation des fonctions de hachage cryptographiques a pour but d'empêcher un adversaire ayant obtenu une clé dérivée d'obtenir des informations sur la valeur secrète ou les autres clés dérivées.

II.3.6. Protocoles d'engagement :

Les fonctions de hachage cryptographiques sont aussi employées dans les protocoles d'engagement. Un protocole d'engagement consiste à permettre à une partie de s'engager sur une valeur sans divulguer aucune information sur celle-ci, la valeur engagée étant révélée ultérieurement. Ces protocoles sont utiliser pour lier des parties à des valeurs d'engagement de façon à ce que qu'aucune des parties ne puisse tirer à posteriori inapproprié sur les autres.

II.4. Les fonctions de hachage :

Une fonction de hachage **H** est une fonction qui prend en entrée une donnée de taille aléatoire **m**, et donne en sortie un condensé de taille fixe, **n**.

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$

$$m \rightarrow H(m)$$

Une fonction de hachage a pour objet de convertir une chaîne d'une longueur arbitraire (un fichier texte, un fichier exécutable de plusieurs mégaoctets, un fichier musical....) en une chaîne de caractères de taille inférieure et généralement fixe, qui s'étend le plus souvent sur 128, 160, 256, voire 512 bits. La chaine résultante est appelée empreinte (digest en anglais) ou condensé de la chaine initiale.

II.4.1. Propriétés de la fonction de hachage H

- H utilise un bloc de données d'une taille quelconque
- H produit une signature de longueur fixe
- H(x) est facile à calculer
- Il est carrément impossible de trouver x tel que H(x)=h. (non réversibilité)
- Il est carrément impossible de trouver $H(x) = H(y)$ tel que $x \neq y$
- Une modification à l'intérieure d'un document, même infinitésimale, provoque un changement radicale de son empreinte : toute modification devient ainsi immédiatement détectable.

II.4.2. Une fonction à sens unique

Est une fonction facile à calculer mais difficile à inverser.

II.4.3. Une fonction de hachage à sens unique

Est une fonction de hachage qui est en plus une fonction à sens unique, il est facile de calculer l'empreinte (digest en anglais) d'une chaine de donnée mais il est difficile d'engendrer des chaines qui ont une empreinte donnée,

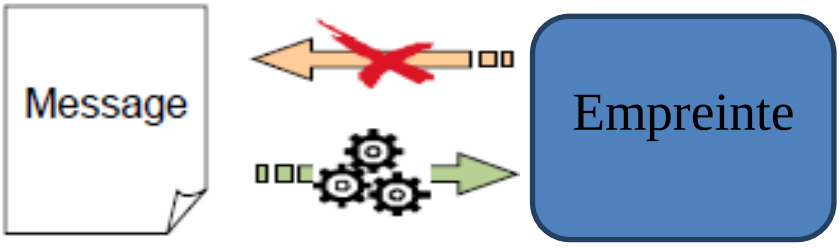


Fig.II.5 : Fonction de hachage à sens unique.

II.5. Les principaux algorithmes de hachage utilisés actuellement sont

- **MD5** (MD pour **Message Digest**): MD5 crée une empreinte de 128 bits. MD5 était un très bon algorithme, mais l'augmentation de la puissance de calcul des ordinateurs et la progression des techniques de cryptanalyse le rendent aujourd'hui moins sûr.
- **SHA-1** (**Secure Hash Algorithm**): SHA-1 crée des empreintes de 160 bits. Il est considéré plus fiable que MD5.

II.6. Comment construire une fonction de hachage ? [20]

Pour des raisons historiques, quasiment toutes les fonctions de hachage sont composées de deux éléments:

- **Une fonction de compression** h**:** une fonction dont la taille d'entrée et de sortie est fixe.
- **Un algorithme d'extension de domaine:** un processus (généralement itératif) utilisant la fonction de compression h pour que la fonction de hachage H puisse hacher des messages de taille arbitraire.

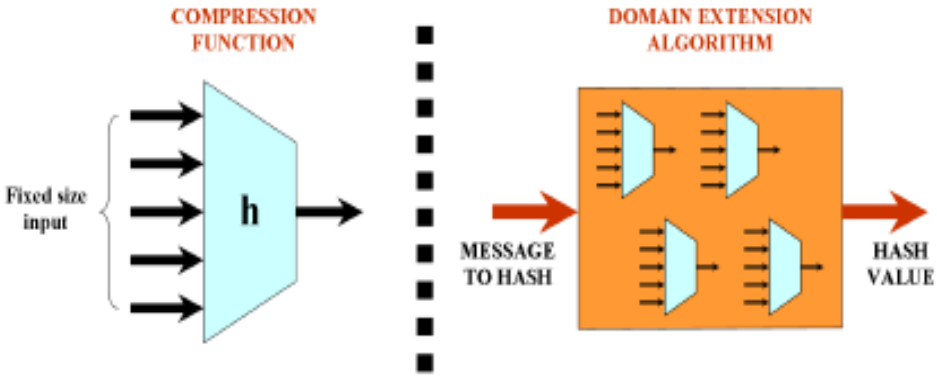


Fig.II.6 : Squelette général pour construire une fonction de hachage.

II.6.1. Fonction de compression : [1]

La construction d'une fonction de compression est en elle-même un chalenge. C'est sur cette fonction que résident les performances d'une fonction de hachage itérative et une part importante de sa sécurité (avec l'algorithme d'extension de domaine). Malheureusement, il s'avère particulièrement difficile de concilier ces deux impératifs. On peut différencier les différents types de conception de fonction de compression en deux familles. La première de ces familles regroupe les constructions fondées sur des algorithmes de chiffrement par blocs. Au sien de cette famille on peut encore distinguer différentes catégories selon l'algorithme de chiffrement utilisé est normalisé ou s'il s'agit d'un algorithme ad hoc. La seconde famille regroupe les fonctions mettant en œuvre des algorithmes dont la sécurité est réductible à la difficulté de résoudre des problèmes algorithmiques réputés durs.

II.6.1.1. Algorithme de chiffrement par blocs :

Le principe des constructions fondées sur un algorithme de chiffrement par bloc consiste à s'appuyer sur une primitive cryptographique existante pour laquelle on possède de bons arguments de sécurité. C'est le cas, par exemple, des schémas de chiffrement par blocs. Ces fonction de compression présentent l'avantage de permettre de faire reposer la sécurité relative à une ou plusieurs des propriétés exigées des fonctions de hachage cryptographiques sur la sécurité de l'algorithme de chiffrement mis en œuvre.

On distingue les constructions fondées sur des algorithmes de chiffrement par bloc selon leur taux de hachage. Ont défini le taux de hachage comme le ratio du nombre de blocs de message traités par le nombre d'appels à l'algorithme de chiffrement employé. Cette définition constitue la mesure standard utilisée pour évaluer l'efficacité des fonctions de hachage fondées sur un schéma de chiffrement par bloc.

- **Algorithmes de chiffrement ad hoc :**

  La première stratégie consiste à construire des algorithmes de chiffrement par bloc ad hoc utilisant des tailles de bloc plus grandes. Les fonctions de compression dénommées ad hoc par extension, sont employées dans la plupart des fonctions cryptographiques utilisées en pratique. Le choix de ce type de construction réside essentiellement dans le gain de performance qu'il apporte. Cependant, la sécurité de ces fonctions repose uniquement sur l'incapacité de la communauté cryptographique à produire des cryptanalyses et repose en général sur un consensus plutôt que sur une

preuve. On peut citer les fonctions de la famille MD-SHA, qui regroupe les fonctions MD [MD4, MD5] et les fonctions SHA [SHA0, SHA1].

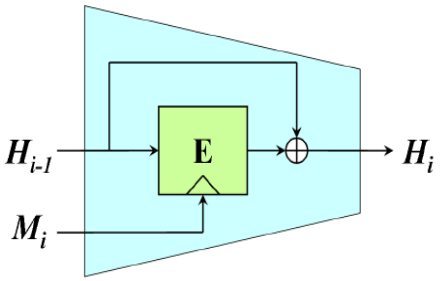


Fig.II.7 : Fonction de compression ad hoc, la famille MD-SHA.

II.6.1.2. Fonction de compression fondée sur un problème réputé difficile

L'idée des fonctions de compression fondées sur des problèmes réputés difficiles est de faire reposer la sécurité de la fonction sur une conjecture éprouvée. En effet, nous connaissons un certain nombre problèmes pour lesquels aucun algorithme de résolution efficace n'est connu. Par exemple le problème de formalisation ou le décodage de syndrome. L'intérêt d'utiliser ces problèmes comme base pour des fonctions de compression réside dans le fait qu'il est peu vraisemblable que des algorithmes efficaces de résoudre ces problèmes puissent voir le jour dans un avenir proche. En effet, les fonctions fondées sur ces problèmes sont en général peu performantes par rapport aux fonctions ad hoc, ce qui a pu conduire certains concepteurs à proposer des paramètres sous-dimensionnées. On parle aussi parfois de difficulté en moyenne, ce qui désigne le fait que la plupart des instances d'un problème sont difficiles à résoudre. Cependant il peut aussi exister des instances faciles même pour un problème difficile. Enfin, la construction choisie pour mettre en œuvre le problème algorithmique peut avoir des conséquences sur la qualité de la réduction de sécurité. Cependant, les quelques échecs de fonctions de hachage fondées sur des problèmes réputés difficiles ne remettent pas en cause la validité du principe.

II.6.2. Un algorithme d'extension de domaine: [4]

Une fonction de compression **h** permet d'obtenir une empreinte pour des messages de talle fixée, et une fonction de hachage **H** permet d'obtenir une empreinte pour des messages de taille arbitraire finie. Les constructions itératives utilisent un algorithme d'extension du domaine de la fonction de compression pour atteindre cet objectif. L'algorithme utilisé ne doit pas introduire de nouvelle vulnérabilité et devrait idéalement préserver les propriétés cryptographiques de la fonction de compression.

L'algorithme d'extension de domaine le plus utilisé en pratique est l'algorithme de Merkle-Damgard.

II.6.2.1. Algorithme de Merkle-Damgard :

En 1989, Merkle et Damgard développèrent de façon indépendante un d'extension de domaine préservant certaines des propriétés cryptographiques de la fonction de compression utilisée. Son principe est décrit dans cet algorithme.

- Algorithme de Merkle-Damgard

ENTREE : une fonction de compression $f: \{0,1\}^n \times \{0,1\}^r \rightarrow \{0,1\}^n$ résistante aux collisions.

SORTIE : une fonction de hachage $h: \{0,1\}^* \rightarrow \{0,1\}^n$ résistante aux collisions.

Découper un message x de taille b bits en blocs $x_1, x_2, \ldots, x_t$ de taille r bits en ajoutant des bits 0 au bloc x_t si nécessaire, et définir un bloc supplémentaire x_{t+1}, destiné à contenir le codage de la longueur du message.

$H_0 = 0^n$ $\{n\ bits$ à $0\}$

$i = 1$

while $i \leq t + 1$ **do**

$H_i = f(H_{i-1} \parallel x_i)$

end while

L'empreinte du message x est alors $h(x) = H_{t+1} = f(H_t \parallel x_{t+1})$.

Les variables H_i sont appelées variables de chainage, et la variable H_0 est qualifiée de valeur initiale *(Initial Value)*.

La première étape de l'algorithme se nomme rembourrage *(padding)*. Ce rembourrage remplit deux objectifs :

- Il permet de ramener la taille du message à hacher à un multiple de r,
- Le codage de la longueur exacte du message dans un bloc supplémentaire assure, de plus, qu'aucun message ne peut constituer le préfixe d'un autre message.

En pratique, l'algorithme de Merkle-Damgard se présente sous une forme légèrement différente. Le rembourrage est réalisé en ajoutant un bit à 1 suivi d'autant de bits à 0 que nécessaires afin de permettre l'insertion du codage de la longueur du message sur les dernier bits d'un bloc complet. Le nombre $\boldsymbol{b}$ réservé au codage de la longueur détermine la taille maximale des messages ($\mathbf{2}^{\boldsymbol{b}}$ $bits$). De plus, la valeur initiale est fixée lors de la spécification de la fonction de hachage.

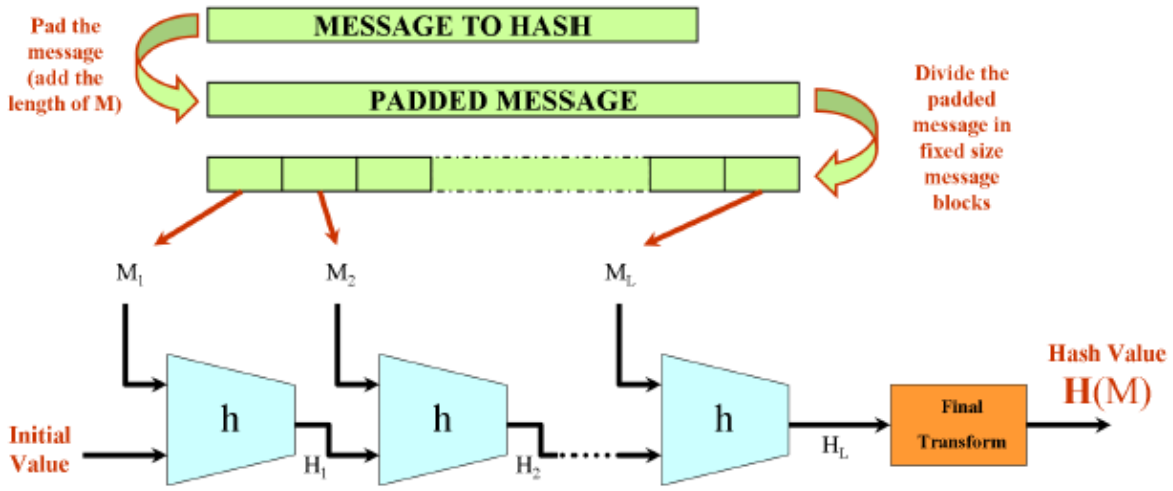


Fig.II.8 : L'algorithme d'extension de domaine Merkle-Damgard.

II.6.2.2. Avantage et inconvénient de l'algorithme de Merkle-Damgard :

➢ **Avantage :**

L'algorithme de Merkle-Damgard permet de réduire le problème de la construction d'une fonction de hachage résistante à la recherche de collision/preimage à celui de la construction d'une fonction de compression résistante à la recherche de collision/preimage : si l'on ne trouve pas de collision/preimage pour h, alors on ne trouve pas de collision/preimage pour H.

- **Inconvénient :**
  Ce processus n'est pas un candidat d'extension de domaine idéal : multi-collision, seconde preimage longues, …

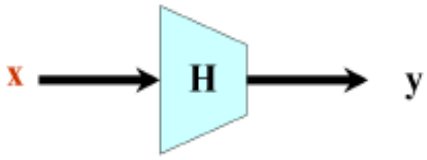
II.7. Sécurité [2]

II.7.1. Propriétés recherchées dans une fonction de hachage :

Dans les lignes qui suivent, on entend par très difficile un problème impossible à résoudre avec la technologie d'aujourd'hui en un temps raisonnable ; on dit que la complexité de la tâche est trop élevée.

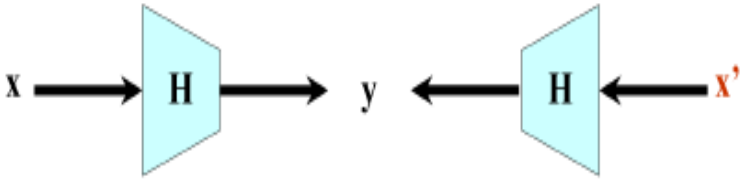
II.7.1.1. Résistance à la recherche de preimage : [19]

La propriété de résistance à la préimage correspond au caractère unilatéral de la fonction de hachage: lorsque l'on a une sortie y (le hach), il doit être très difficile de trouver une entrée x telle que $h(x) = y$; h étant la fonction de hachage. Cela interdit le retour en arrière et garantit par exemple la non-lecture d'un mot de passe stocké sous la forme d'un hach.



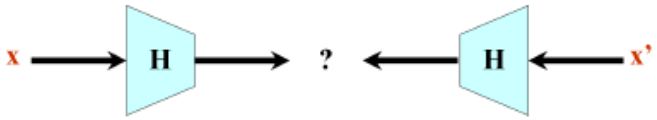
II.7.1.2. Résistance à la recherche de seconde preimage :

La propriété de résistance à la seconde préimage diffère de la précédente par les informations dont l'attaquant dispose : lorsque x est connu, il doit être très difficile de trouver $x' \neq x$ tel que $h(x') = h(x)$. Cette propriété a son importance dans les systèmes d'authentification.



II.7.1.3. Résistance à la recherche de collision:

Cette fois-ci, l'attaquant ne dispose d'aucune condition de départ, si ce n'est l'algorithme utilisé. Il doit alors être très difficile, pour un attaquant, de trouver x et x' (avec $x \neq x'$) tels que $h(x) = h(x')$. En moins de $\theta\left(2^{\frac{n}{2}}\right)$ opérations (paradoxe des anniversaires).



II.7.2. Attaques :

Plusieurs types d'attaques peuvent être adressés à une fonction de hachage ou à son utilisation, parmi lesquels :

- attaque par dictionnaire : afin de deviner le mot de passe à partir d'un hach, l'attaquant peut essayer de hacher chaque mot d'un dictionnaire de mots de passes, afin de comparer les résultats
- attaque par table de hachage : plutôt que de passer du temps à calculer tous les haches, un attaquant peut se procurer une table préconçue de correspondances texte/hach, afin d'y chercher le hach à casser. Inconvénients : de telles tables sont très volumineuses, et l'utilisation de salages rend ce type d'attaques inutile
- attaque par tables arc-en-ciel : également inefficaces contre un salage
- attaque par force brute : toutes les possibilités sont essayées jusqu'à trouver un texte qui donne le même hach. Inconvénient majeur : cela n'est pas raisonnable en temps de calcul
- attaques par collisions, 1st preimage, 2nd preimage

II.8. Conclusion

Après avoir présenté les notions générales de la fonction de hachage, dans cette derrière partie de ce chapitre nous nous intéressons à l'algorithme de hachage SHA-1.

Nous définirons la terminologie et les fonctions utilisées comme éléments de la construction de SHA-1.

Troisième partie : La fonction de hachage SHA-1

III.1. Introduction :

Après avoir présenté les notions générales d'une fonction de hachage cryptographique, nous nous spécifient un algorithme de hachage sécurise (SHA-1, *Secure Hash Algorithm*) pour calculer une représentation condensée d'un message ou d'un fichier de données. Lorsque un message d'une longueur quelconque.

III.2. Bref histoire sur la fonction de hachage SHA-1:

Parmi les fonctions de hachage les plus connues, on retrouve le Secure Hash Algorithm (SHA-1) et la famille de Message Digest (MD). Le premier algorithme MD à avoir été publié fut le MD2, conçu en 1989 par Ronald Rivest. Il a été suivi des versions améliorées MD4 et MD5. Le MD5, introduit en 1991 pour remplacer le MD4, produit un message condensé de 128 bits. Nous savons maintenant que cela n'est pas suffisant pour assurer une bonne sécurité. En 1996, une faille a été découverte dans le MD5, démontrant qu'il y avait une possibilité de trouver des collisions à la demande. Ainsi le MD5 a graduellement été retiré pour être remplacé par le SHA-1. En 2004, des collisions complètes ont été trouvées, démontrant ainsi que le MD5 n'est plus sûr au sens cryptographique. Le SHA-1 a été développé pas la National Security Agency (NSA) et soumis à la National Institute of Standards and Technology (NIST) une agence du département du commerce aux États-Unis, dont le but est de promouvoir l'économie. La version originale, souvent appelée SHA-0, a été publiée en 1993. Toutefois, elle présentait des faiblesses qui ont été corrigées pour donner naissance au SHA-1 en 1995 qui, jusqu'à tout récemment, a été la version recommandée par la NIST.

III.3. Présentation du SHA-1 :

SHA-1 est une fonction de hachage cryptographique conçue par la NSA et publié comme un standard de traitement de l'information Elle prend des messages de taille quelconque inférieure à 2^{64} bits et produit une empreinte de 160 bits. La fonction suit la construction de Merkle-Damgãrd :

III.4. Description de SHA-1 :

Après avoir constaté que l'algorithme de hachage SHA-1 produise des empreintes de 160 bits pour des messages et des fichiers d'une taille arbitraire, en vas définir la terminologie et les fonctions utilisées comme éléments de la construction de SHA-1.

1. **Opération sur les mots :**

   a. Opération de mots logique

   $X \wedge Y$ = ET logique.

   $X \vee Y$ = OU inclusif logique

   $X \oplus Y$ = OU exclusif logique

   $\neg X$ = Complément logique

   $X \hookleftarrow r$ = décale les bits de X de r positions vers la gauche (les r bits du début deviennent les bits de la fin).

2. **Fonction et constantes utilisées :**

- Une séquence de la fonction logique f (0), f (1),…. f(79) est utilisée dans SHA-1. Chaque f(t), $0 \leq t \leq 79$, opère sur trois mots B, C, D de 32 bits et produit un mot de 32 bits en résultat. f(t ; B, C, D) est défini comme suite :

$$\boldsymbol{f_t(B,C,D)} = \begin{cases} (B \wedge C) \vee (\neg B \wedge D) & \text{si } 0 \leq t \leq 19 \\ B \oplus C \oplus D & \text{si } 20 \leq t \leq 39 \\ (B \wedge C) \vee (B \wedge D) \vee (C \wedge D) & \text{si } 40 \leq t \leq 59 \\ B \oplus C \oplus D & \text{si } 60 \leq t \leq 79 \end{cases}$$

- Une séquence de mots constants K(0), K(1),…. K(79) est dans SHA-1. En hexadécimal, ils donnent :

$$\boldsymbol{K(t)} = \begin{cases} 5A827999 & \text{si } 0 \leq t \leq 19 \\ 6ED9EBA1 & \text{si } 20 \leq t \leq 39 \\ 8F1BBCDC & \text{si } 40 \leq t \leq 59 \\ CA62C1D6 & \text{si } 60 \leq t \leq 79 \end{cases}$$

Notons que les constantes ci-dessus représentent des séquences de 32 bits.

Nous sommes maintenant en mesure de définir formellement chacune des étapes du SHA-1

III.4.1. L'algorithme de SHA-1 :

a. Bourrage du message :

SHA-1 est utilisé pour calculer un résume de message pour un message ou fichier de données qui est fourni en entrée. Le message ou fichier de données devrait être considéré comme une chaine binaire. La longueur du message est le nombre de bits dans le message (le message vide a une longueur 0). Si le nombre de bits dans un message est un multiple de 8, pour le rendre compact, on peut représenter le message en hexadécimal. L'objet du bourrage de message est de rendre la longueur totale d'un message bourre multiple de 512. SHA-1 traite en séquence les blocs de 512 bits lors du calcul du résume du message. Ce qui suit spécifie comment devra être effectue ce bourrage. En résume, un "1" suivi par des "0" suivis par un entier de 64 bits sont ajoutés à la fin du message pour produire un message bourre de longueur 512 * n. L'entier de 64 bits est la longueur du message d'origine. Le message bourre est alors traite par SHA-1 comme des blocs de 512 bits.

Supposons qu'un message a une longueur $l < 2^{64}$. Avant qu'il soit entre dans SHA-1, le message est bourre à droite comme suit :

1. "1" est ajoute.

Exemple 01 :
Si le message d'origine est "01010000", il devient avec le bourrage "010100001".

2. Les "0" sont ajoutes. Le nombre de "0" va dépendre de la longueur d'origine du message. Les derniers 64 bits du dernier bloc de 512 bits sont réserves pour la longueur *l* du message d'origine.

Exemple 02 :
Supposons que le message d'origine soit la chaine binaire 01100001011000100110011. Après l'étape (**1**), cela donne 01100001011000100110011 1. Comme *l* = 24, le nombre de bits ci-dessus est 25 et 423 "0" sont ajoutes, faisant un total de 448. Cela donne (en hexa) 61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000.

3. Obtenir la représentation de *l* en deux mots, le nombre de bits dans le message d'origine. Si *l* < 2^32 le premier mot est alors tout de zéros. Ajouter ces deux mots au message bourre.

Exemple 03 ; (suit de Exemple 02):

Supposons que le message d'origine est comme dans (Exemple **02**). Alors *l* = 24 (noter que l est calculé avant tout bourrage). La représentation en deux mots de 24 est l'hexadécimal 00000000 00000018. Donc, le message bourre final est en hexadécimal :
61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018.
Le message bourre va contenir 16 * n mots pour toute n > 0. Le message bourre est considère comme une séquence des n blocs M(1), M(2), des premiers caractères (ou bits) du message.

b. Calcul du résumé de message

Appris avoirs les n blocs de 512 bits chacun : m= $(m_1, m_2, \ldots., m_n)$. En entames les étapes suivantes :

1. On pose :
   $H_0 = 67452301,$
   $H_1 = EFCDAB89,$
   $H_2 = 98BADCFE$
   $H_3 = 10325476$
   $H_4 = C3D2E1F0,$
   Ou H_i est en hexadécimal pour $i = 1, \ldots.,4.$
2. Pour $i = 1, \ldots, l,$
   a) On écrit $m_i = (W_0, W_1, W_2, \ldots, W_{15})$, ou chaque W_j est composé de 32 bits.
   b) Pour $t = 16, \ldots,79$, on pose $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \curvearrowleft 1.$
   c) Soit $a = H_0,\ b = H_1, c = H_2, d = H_3, e = H_4.$
   d) Pour $t = 0, \ldots,79 :\ T = (a \curvearrowleft 5) + f_t(b, c, d) + e + W_t + K_t,\ e = d,\ d = c,$
   $c = (b \curvearrowleft 30),\ b = a,\ a = T.$
   e) Pour $j = 0, \ldots,4$, on attribue à H_j une nouvelle valeur de la façon suivante :
   $$H_0 = H_0 + a, H_1 = H_1 + b, H_2 = H_2 + c, H_3 = H_3 + d, H_4 = H_4 + e$$
3. Le message condensé est $H = (H_0,\ H_1,\ H_2,\ H_3,\ H_4).$

III.5. Sécurité :

III.5.1. Attaques sur SHA-1 :

III.5.1.1. Attaque directe :

Nous avons tout d'abord essayé d'attaquer directement le problème en essayant de trouver la variété de l'idéal engendré par le système dans lequel nous avons fixé les variables qui correspondent au hash pour obtenir une attaque en préimage. Le nombre de variable est beaucoup trop important pour avoir une attaque en pratique, même si les polynômes ne dépassent pas le degré 3. Nous avons donc abandonne l'idée et nous avons étudié les attaques en collisions basées sur la cryptanalyse différentielle.

III.5.1.2. Attaque différentielle :

Les attaques différentielles sur les fonctions de hachage cryptographiques sont sans aucun doute celles qui ont donné les meilleurs résultats, avec en tête les travaux de Wang et al qui ont permis de trouver des collisions sur des fonctions basées sur le même principe. Ils ont aussi une attaque en collision sur SHA-1 présentée à la conférence CRYPTO 2005 qui a une complexité estimée à 263 calculs d'empreintes.

Le principe de l'attaque différentielle a été introduit par Biham et Shamir pour analyser la sécurité du DES. Ils définissent la cryptanalyse différentielle comme étant l'analyse des effets que peuvent produire une certaine différence dans un couple de messages clairs sur les différences entres les chiffrés. On voit très bien qu'on peut adapter cette analyse sur une fonction de hachage telle que SHA-1. L'attaque que propose Wang peut se décomposer de la manière suivante.

1. choisir un bon" vecteur Γ de 80 ×32 bit qui correspondront à la différence entre les 2 messages étendus de la collision.
2. choisir un chemin différentiel qui correspondra aux différences qu'auront les variables de chaînage au fil des 80 tours.
3. trouver un ensemble de conditions suffisantes sur m qui assurent avec une bonne probabilité que le couple de message $(m, m + \Gamma)$ suive le chemin différentiel (donc qui mène à une collision)
4. choisir aléatoirement un message m et le " modifier" jusqu'à ce qu'il satisfasse les conditions suffisantes.

III.5.1.3. Attaque algébrique

Dans l'article de Sugita [20] où les auteurs proposent une méthode pour trouver des collisions sur SHA-1. Le principe est basé sur l'attaque différentielle telle que proposée par Wang légèrement modifiée appliquée à une version de SHA-1 réduite à 58 tours. La principale différence est que la modification de message ne s'effectue pas sur les bits du message mais celles des variables de chainage. (On peut facilement voir que dans SHA-1 les 512 bits du message d'origine sont en bijection avec les 512 bits des 16 premiers a_i). Le principe est de voir les "conditions suffisantes" de l'attaque de Wang d'un point de vue algébrique, et de transformer ces conditions en polynômes dans $\mathbb{F}_2[x]$. Malheureusement, les conditions suffisantes sont exprimées comme fonctions booléennes sur les variables du message (dans notre modélisation les x_i), et pour la modification de message, on a besoin de connaitre la fonction f telle que $x_i = f(a_0, \dots a_{15})$ qui n'est pas triviale, et correspond à des polynômes de très haut degré.

Nous avons donc ajouté à ces conditions suffisantes les équations de notre modélisation de SHA-1 qui modélisent à fortiori les relations qui existent entre les x_i et les a_i. Le problème reste le même que pour l'attaque en préimage et si on arrive à appliquer cette méthode sur des systèmes jouets (messages de 4×4 bits sur 6 tours), avec les paramètres de SHA-1, même pour un nombre de tour restreint, le calcul de base de Gröbner nécessite un trop grand espace en mémoire pour pouvoir être mené à bout.

III .6. Conclusion

Dans cette partie on a présenté le fonctionnement détaillé de l'algorithme de hachage SHA-1.

Nous implémenterons dans le chapitre qui va suivre l'analyse et la conception de notre application qui nous permettre de condenser des messages, des documents et des fichiers (Html, Texte, Image,…), cette dernière elle est basé sur l'algorithme de hachage SHA-1.

Chapitre III

Analyse et conception

III.1. Introduction

Avant d'entamer la création de toutes applications informatiques, il convient de suivre une démarche méthodologie et rigoureuse pour planifier et concevoir l'application avec précision et en détail, pour cela en va mettre en évidence tous les objectifs tracés pour la bonne élaboration du projet souhaité.

Dans notre cas nous avons opté pour la modélisation orienté objet avec le langage UML car il est le reflet du futur système avant même sa réalisation, dans le but d'avoir une meilleure analyse et de rendre la conception de notre application plus complète et tous cela grâce au diagramme qu'il offre.

III.2. Présentation d'UML

III.2.1 Origine et définition d'UML

UML (**U**nified **M**odeling **L**anguage) que l'on peut traduire par « Langage de Modélisation Unifié », est né de la fusion des trois méthodes qui s'imposaient dans le domaine de la modélisation objet au milieu des années 90 : **OOD** (**O**bject **O**riented **D**evelopment) de Grady Booch, **OMT** (**O**bject **M**odeling **T**echnique) de James Rumbaugh, **OOSE** (**O**bject **O**riented **S**oftware **E**ngineering) de Ivar Jacobson. Il a été normalisé par l'**OMG** (**O**bject **M**anagement **G**roup) en 1997, et est rapidement devenu la référence en termes de modélisation objet.

UML comble une lacune importante des technologies objet. Il permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation. Mais UML offre bien plus encore c'est un langage formel, qui décrit de manière très précise tous les éléments de modélisation et la sémantique de ses éléments. En d'autre terme : UML normalise les concepts objet.

UML est essentiellement un support de communication qui facilite la représentation et la compréhension des solutions objets :

- un langage couvrant toutes les étapes nécessaires au développement d'un système.
- sa notation formelle limite les ambigüités
- un langage universel pouvant servir de support pour tout langage orienté objet
- un moyen de définir la structure d'un programme
- une représentation visuelle permettant la communication entre les acteurs d'un même projet

- une notation graphique simple, compréhensible même par des non informaticiens

III.3. But et contexte de la plate-forme

Le système doit offrir une IHM (L'interface homme –machine) simple et facile à utiliser.

La fenêtre principale permet à l'utilisateur d'avoir l'empreintes de tous type de fichier (par exemple Word, PDF, image, …etc.) comme il peut aussi avoir l'empreinte de ce dernier à partir de sans adresse physique

III.4. La modélisation UML

UML offre un moyen astucieux permettant de représenter diverses projections d'un même système informatique. Plusieurs diagrammes UML existent, et pour modéliser notre site Web nous allons présenter les diagrammes suivant :

- Diagramme de cas d'utilisation
- Diagramme de séquence
- Diagramme de classes

III.4.1. Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation est un graphe d'acteur, un ensemble de cas d'utilisation réunis par la limite de système, des associations de communication entre les acteurs et les cas d'utilisation, et des généralisations entre cas d'utilisation.

Dans notre cas on a un seul acteur qu'est l'utilisateur, la figure suivante représente son diagramme d'utilisation.

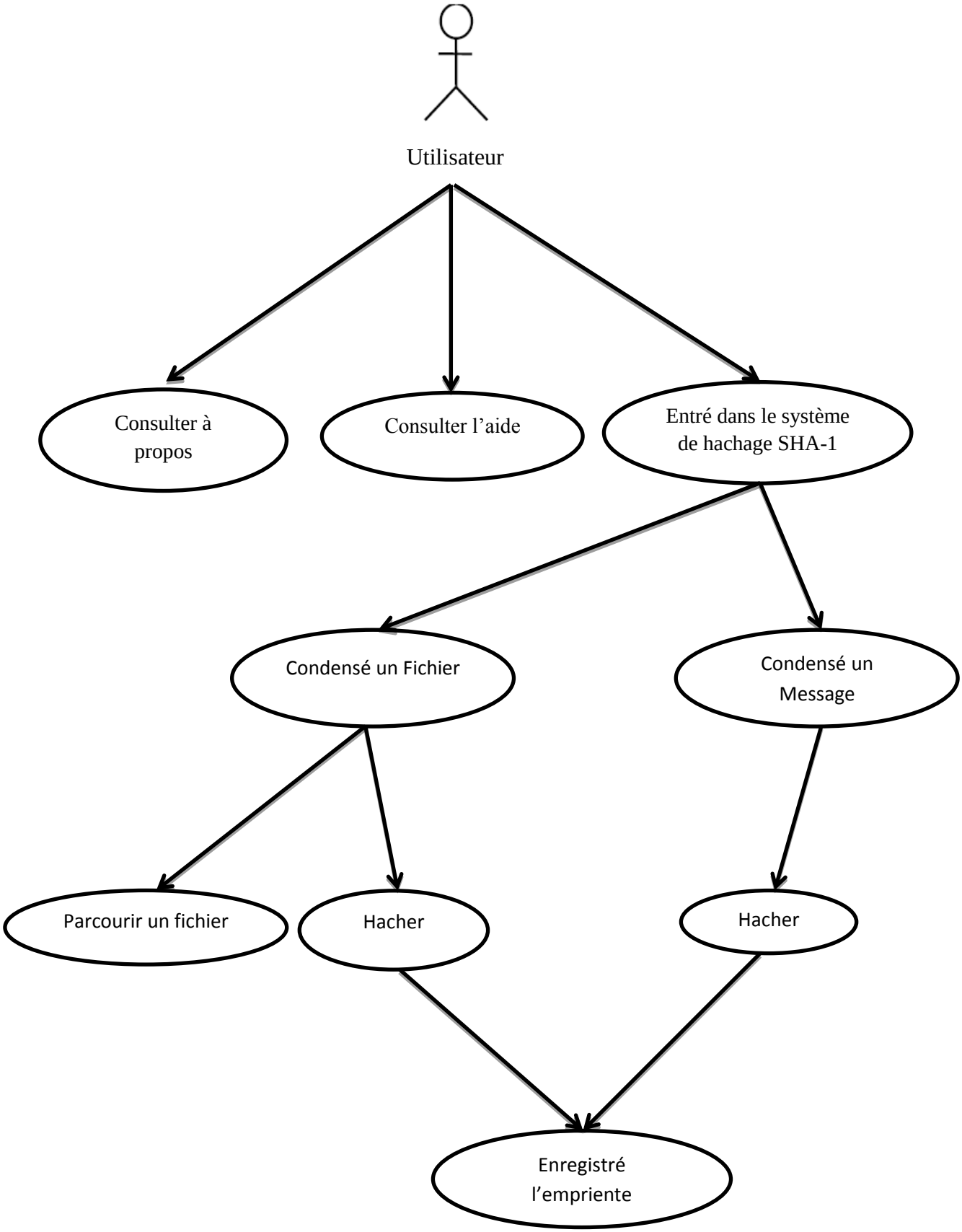


Fig.III.1 : Diagramme de cas d'utilisation

III.4.2. Diagramme de séquence détaillé

Le diagramme de séquence détaillé présente la vue dynamique du système, sont objectif est de présenté séquentiellement le déroulement des traitements et des interactions entre les objets en indiquant la chronologie des échanges. Cette représentation se réalise par cas d'utilisation.

III.4.2.1. Diagramme de séquence détaillé du cas d'utilisation « Condensé un message »

Ce diagramme déroule selon les étapes suivantes :

1. L'utilisateur attient l'application, le système lui affiche la fenêtre principale.
2. Après avoir choisi entre condensé un message l'utilisateur clique sur le bouton « Entré » pour entrer dans le système de hachage, une autre fenêtre s'affiche.
3. L'utilisateur saisit le message a condensé et clique sur le bouton « hacher » le système lui retourne l'empreinte, et une page de confirmation que l'empreinte a été générer, elle demande aussi à l'utilisateur d'enregistré l'empreinte ou non.
4. L'utilisateur clique sur le bouton « enregistré » si il tient à enregistré l'empreinte.

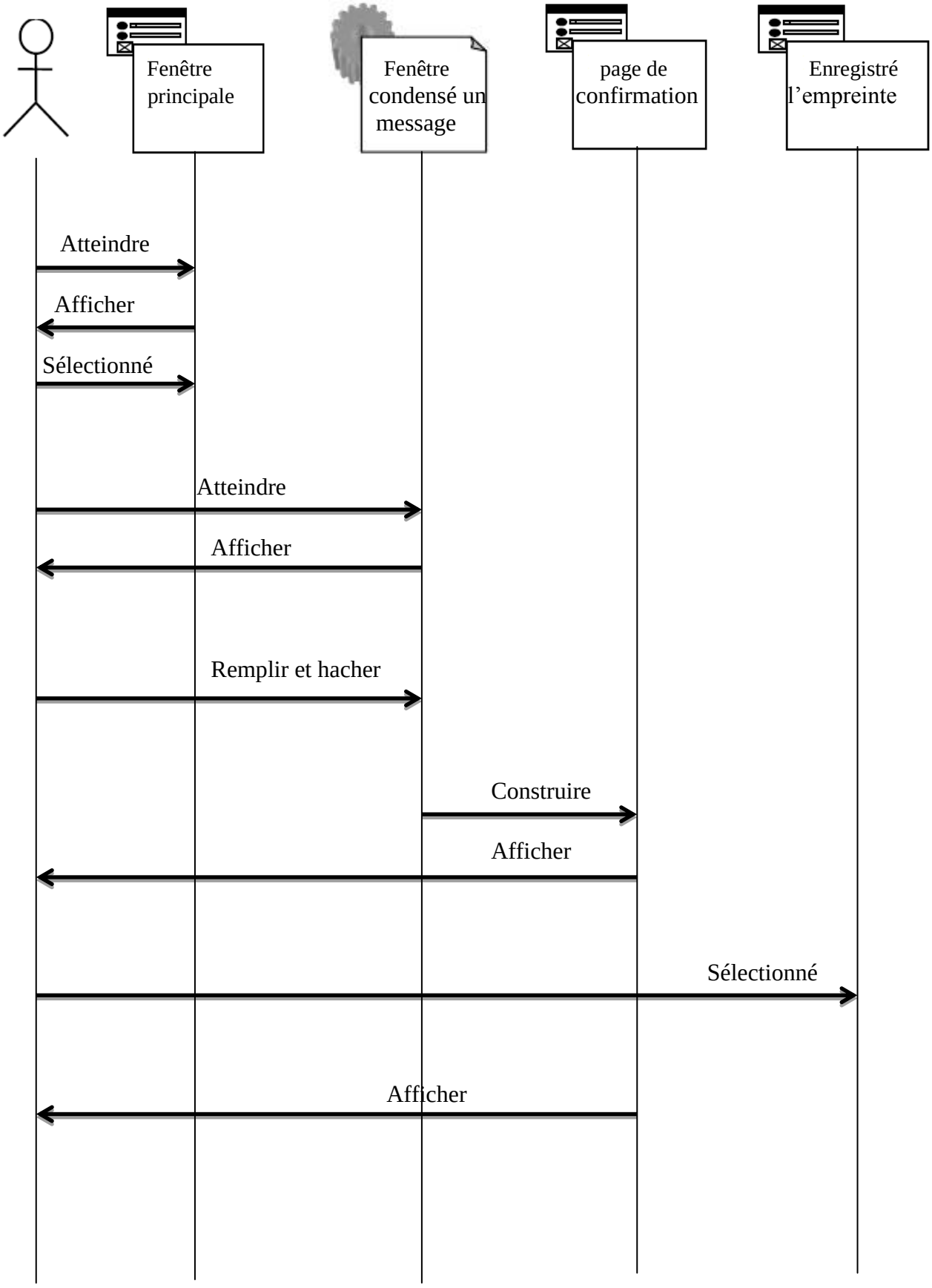


Fig.III.2 : Diagramme de séquence détaillé du cas d'utilisation « Condensé un message »

III.4.2.2. Diagramme de séquence détaillé du cas d'utilisation « Condensé un fichier »

Ce diagramme déroule selon les étapes suivantes :

1. L'utilisateur attient l'application, le système lui affiche la fenêtre principale de cette dernière.
2. Cliquer sur le bouton « Entré » pour entrer dans le système de hachage de fichier après avoir choisi le fichier, une autre fenêtre s'affiche.
3. L'utilisateur sélectionne un fichier à condensé en cliquant sur le bouton « parcourir » puis « ouvrir ».
4. Appris avoir sélectionné le fichier, l'utilisateur clique sur le bouton « hacher » le système lui retourne l'empreinte de ce dernier avec une page de confirmation qui demande à l'utilisateur d'enregistré l'empreinte ou non.
5. L'utilisateur clique sur le bouton « enregistré » si il tient à enregistré l'empreinte.

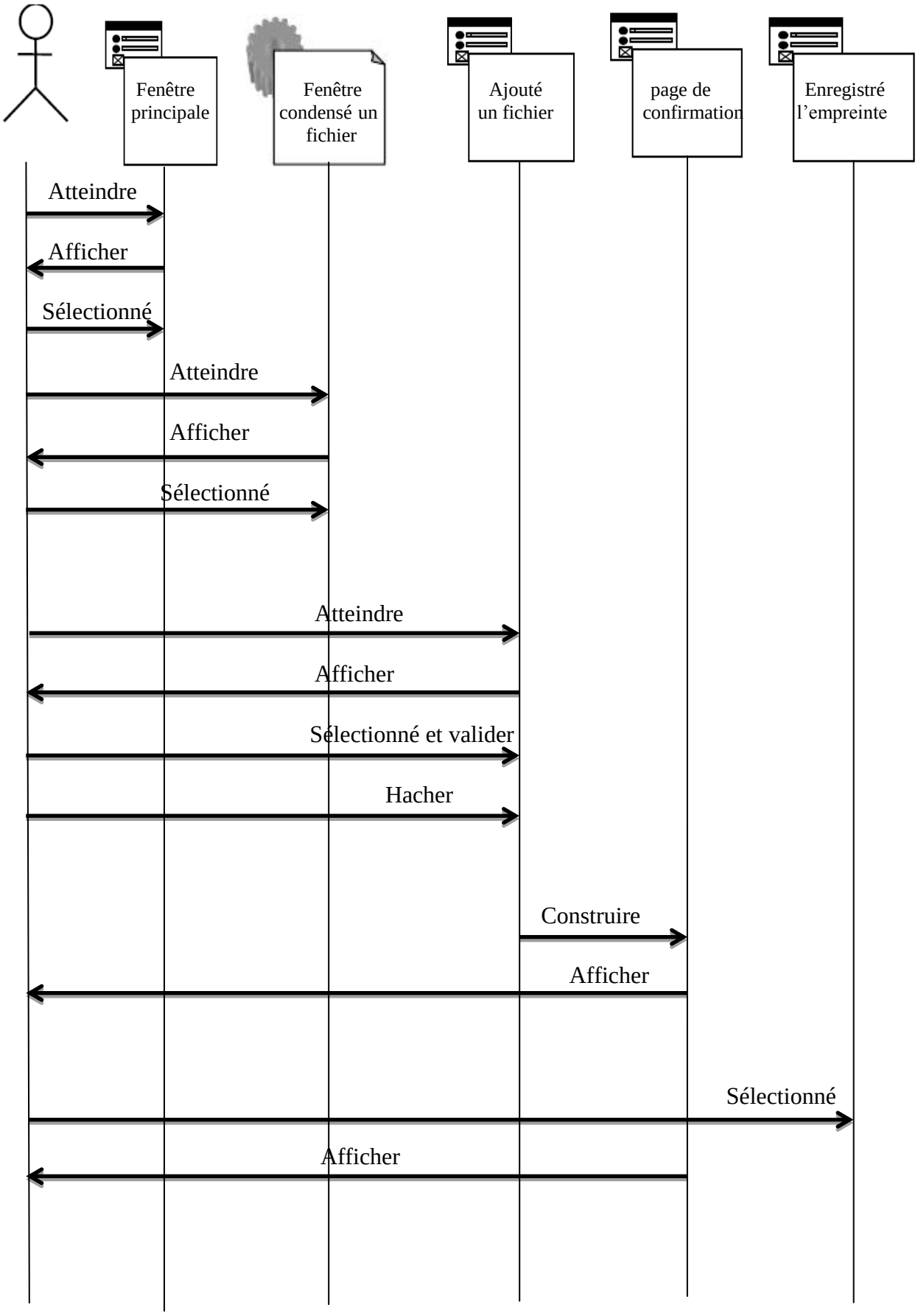


Fig.III.3 : Diagramme de séquence détaillé du cas d'utilisation « Condensé un fichier »

III.4.2.3. Diagramme de séquence détaillé du cas d'utilisation « Consulter l'aide »

Ce diagramme déroule selon les étapes suivantes :

6. L'utilisateur attient l'application, le système lui affiche la fenêtre principale de cette dernière.
7. Sur la barre de menu de la fenêtre principale il clique sur le lien aide, une page qui fournit l'aide de l'utilisation de l'application s'affiche.

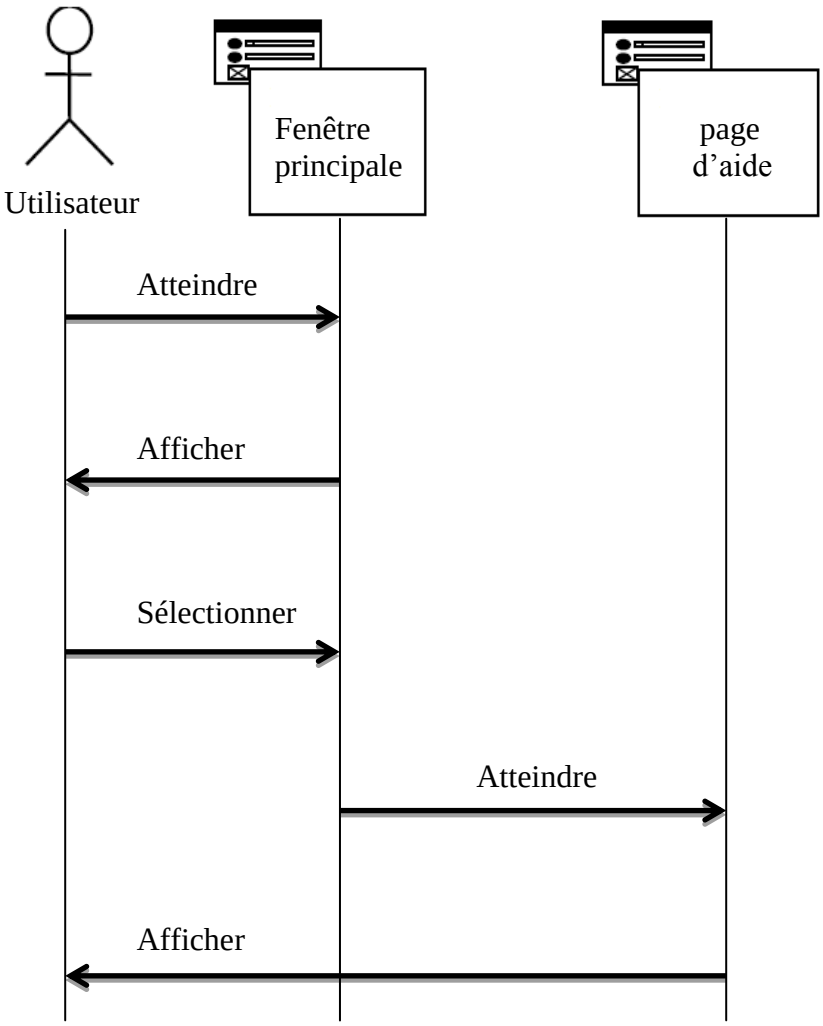


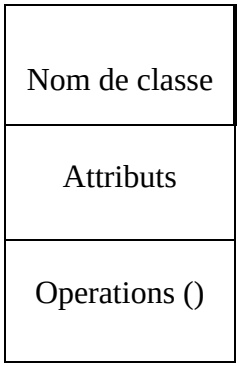
Fig.III.4 : Diagramme de séquence détaillé du cas d'utilisation « Consulter l'aide »

III.4.3. Les Diagrammes de classes

Représentent la vue statique des objets pages, il contient des classes et leur association et éventuellement des objets. L'intérêt majeur de diagramme de classe est de modéliser les entités d'un système.

La classe est un ensemble d'objets ayant les mêmes caractéristiques

- **Représentation graphique de classe :**



L'ensemble des classes est partitionné en trois composants :

- **Les classes entités :** Elles permettent de modéliser toutes les informations que l'on veut gérer.
- **Les classes acteurs :** Elles représentent les rôles attribués.
- **Les classe processus :** Elles permettent de répertorier les activités organisées pour accomplir les missions.

Les associations utilisées dans ces diagrammes :

❖ **Soumet « Submit » :**

Une association de soumission se trouve toujours entre un formulaire et une page serveur. Les valeurs des champs du formulaire sont soumises au serveur qu'il les traite, par l'intermédiaire de pages serveur**.**

Icone : « Submit » ⟶

- **« Build » :**

  Est une association orientée entre les pages client et les pages serveur. Elle indique quelle page serveur est responsable de la création de la page client. Une page serveur peut construire plusieurs pages client, mais une page client n'est construite que par une et une seule page serveur.

Icone : « Build » ⟶

- « **Redirect » :**

  Est une association unidirectionnelle qui relie deux pages client ou serveur.

Icone : « Redirect » ⟶

- **« Link »**

  C'est une association entre une page client et une autre page client ou serveur.

Icone : **«** Link **»** ⟶

Apres les diagrammes de séquence des différents cas d'utilisations représentés, nous allons passer aux diagrammes de classes qui représentent une vue logique des objets.

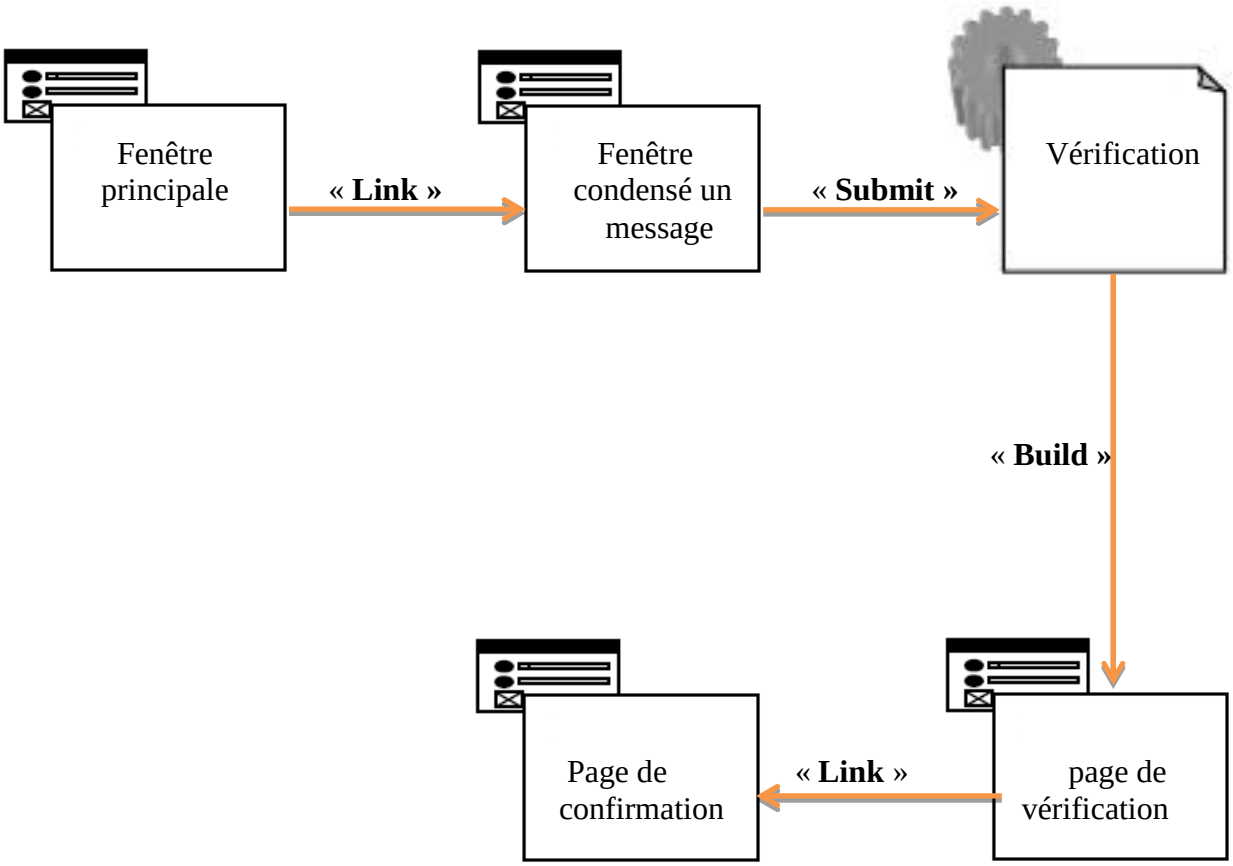


Fig.III.5 : Diagramme de classes « Condensé un message »

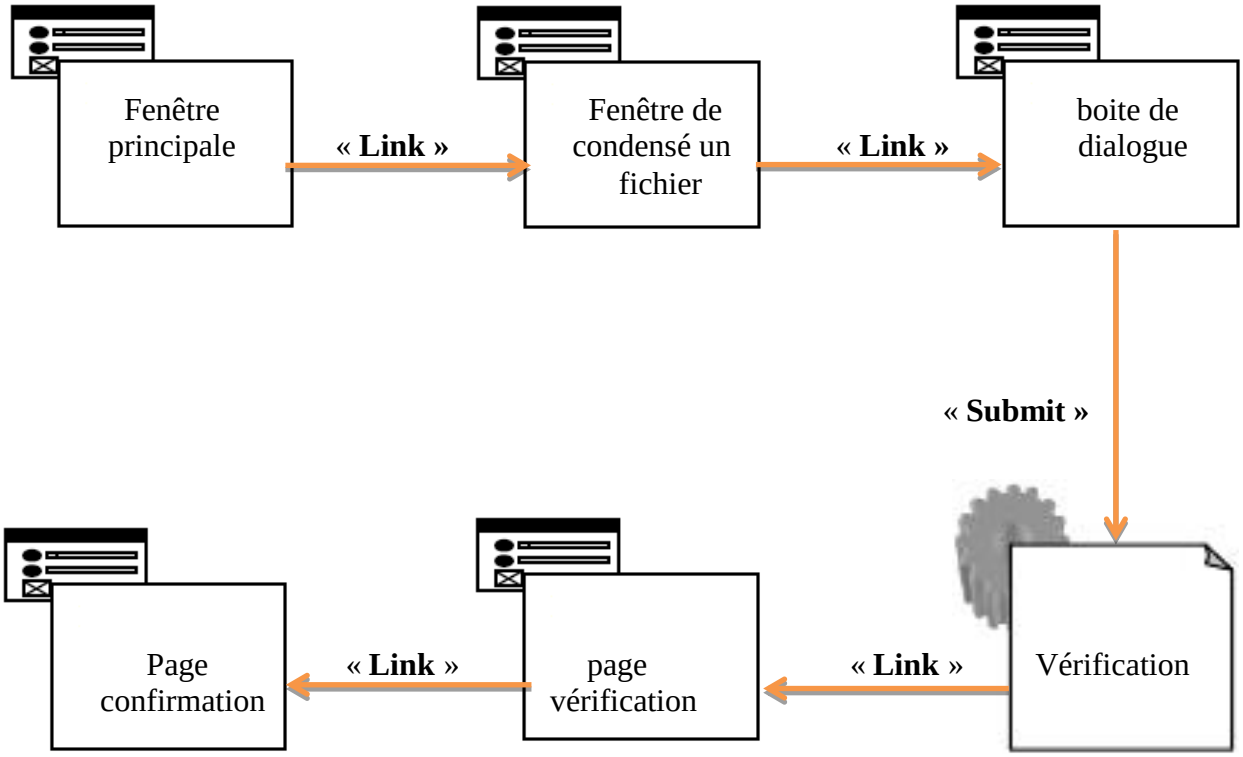


Fig.III.6 : Diagramme de classes « Condensé un fichier »

III.5. Conclusion

Dans ce chapitre nous avons pu cerner les axes de la modélisation en basant sur la méthode UML qui nous a permis d'avoir la modélisation de l'aspect fonctionnel d'un système, c'est-à-dire que nous modélisons ce que le système doit faire sans dire comment il va le faire, en utilisant le diagramme de cas d'utilisation. Par la suite, la modélisation de l'aspect statique d'un système qui est la modélisation de l'intérieur du système par le diagramme de classe. Il nous reste à définir l'environnement de développement et la réalisation de notre application, ce qui sera l'objet de chapitre suivant.

Chapitre IV

Réalisation

IV.1. Introduction :

Après avoir présenté dans le chapitre précédent l'Analyse et la Conception de notre système, nous allons présenter dans ce chapitre en premier lieu, le l'éternuementqui nous a servi d'appui pour le développement de notre application et nous finirons par la présentation des interfaces de notre application.

IV.2. Outils de développement :

Dans notre réalisationnous avons opté pour l'utilisation de la plateforme Windows avec le système d'exploitation Windows 7 et le logiciel NetBeans.

IV.3. Le langage Java :

Javaest un langage de programmation orienté objetsdéveloppé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitation (Windows, Linux, Macintosh, Solaris). Un objet est une représentationsimplifiée d'une entité du monde réel : entité concrète (ex : ma voiture) ou non (ex : la date d'aujourd'hui). Un objet se caractérise par son état et son comportement. Un objet stocke son état dans des variables appelées champs (ou attributs) et présente son comportement a travers de fonctionnalités appelées méthodes.

Afin de réaliser l'interface permettant aux utilisateurs de manipuler notre système, nous avons choisi le langage Java.

Notre choix s'est porte sur ce langage pour les raisons suivantes:

- Java est un langage multiplateforme qui permet aux concepteurs, selon la célèbre maxime « Write once, run everywhe », d'écrire un code capable de fonctionner dans tous les environnements. Pour cela, il suffit que l'environnement possède une JVM (Java Virtual Machine);
- Java est un langage oriente objet, simple, qui réduit le risque d'erreurs d'incohérence;
- Java est doté d'une riche bibliothèque de classes, comprenant la gestion des interfaces graphiques (fenêtres, boites de dialogue, contrôles, menus, graphisme) et la gestion des exceptions;
- Le JDK (Java Development Kit), fournit gratuitement par Sun, regroupe l'ensemble des éléments permettant le développement, la mise au point et l'exécution des programmes Java.

IV.4. Présentation l'environnement de développement :

IV.4.1. NetBeans :

NetBeans est à l'origine un environnement de développement intégré (EDI) pour Java, placé en open source par Sun en juin 2000 sous licence CDDL et GPLv2 (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, XML, Ruby, PHP 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).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 etSPARC), Mac OS X et Open VMS.

NetBeans permet de programmer et concevoir les interfaces utilisateur de manière visuelle. Pour ce faire, il offre de nombreux outils de conception visuelle qui permettent de concevoir les interfaces utilisateur avec rapidité et efficacité en attachant des événements et en modifiant les dispositions.

Pour notre réalisation nous avons utilisés la version 7.0.1 (NetBeans 7.0.1).

IV.4.2. Outils de conception visuelle de NetBeans :

- **Volet Project :** contient tous les packages de projet, et pour chaque package on trouve les classes crées avec les quelles l'application est réalisé.
- **Palette des composants** : la palette des composants contient des composants visuels et non visuels qu'on peut faire glisser dans l'inspecteur ou dans l'arborescence des composants.
- **Volet inspecteur** : L'arborescence des composants apparait dans le volet inspecteur au dessous du volet projet. Elle affiche une vue structurée de tous les composants du fichier source et de leurs relations.
- **Le volet propriétés des composants** : il est utilisé pour définir les valeurs des propriétés des composants et pour attacher des méthodes aux événements. Les modifications effectuées dans ce volet sont répercutées visuellement dans la conception et immédiatement dans le code source.

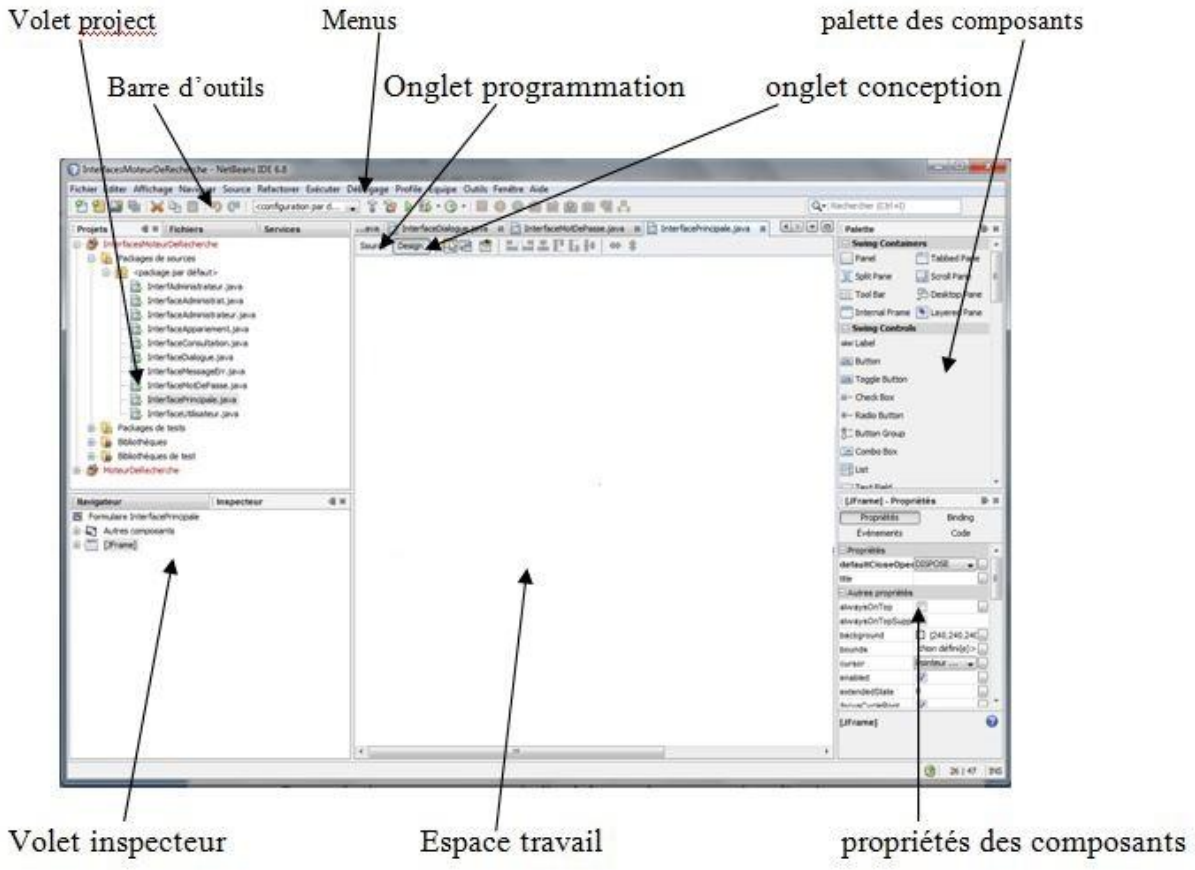


Fig.IV.1 : Environnement de développement NetBeans.

IV.5. Description de notre Application :

IV.5.1. Architecture générale de l'application :

Notre application est subdivisée en trois parties :

- La partie applicative : elle contientle processus de hachagedes documents à partir d'une base documentaire ou des messages donnés par l'utilisateur.
- La partie stockage : c'est un fichier texte sert à stocker le résultat de hachage.
- La partie interfaces : contient les interfaces de communication entre le système et l'utilisateur.

La figure suivante montre la structure générale de système :

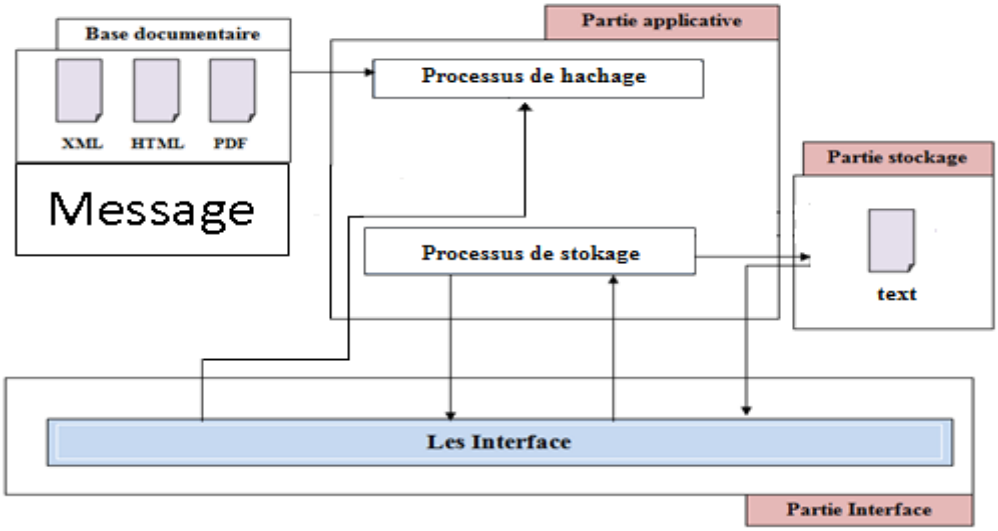


Fig.IV.2 : la structure générale de notre système.

IV.5.2. Description de quelque interface:

Via cette interface, les utilisateurs de notre application sont appelés à choisir entre condenser un fichier ou un message.

Fig.IV.3 : Interface Principale.

Mode d'accès :

1-Fichier :

Si le choix se porte sur le bouton « Fichier », la fenêtre de « condensé un fichier » sera affichée comme le montre la figure suivante :

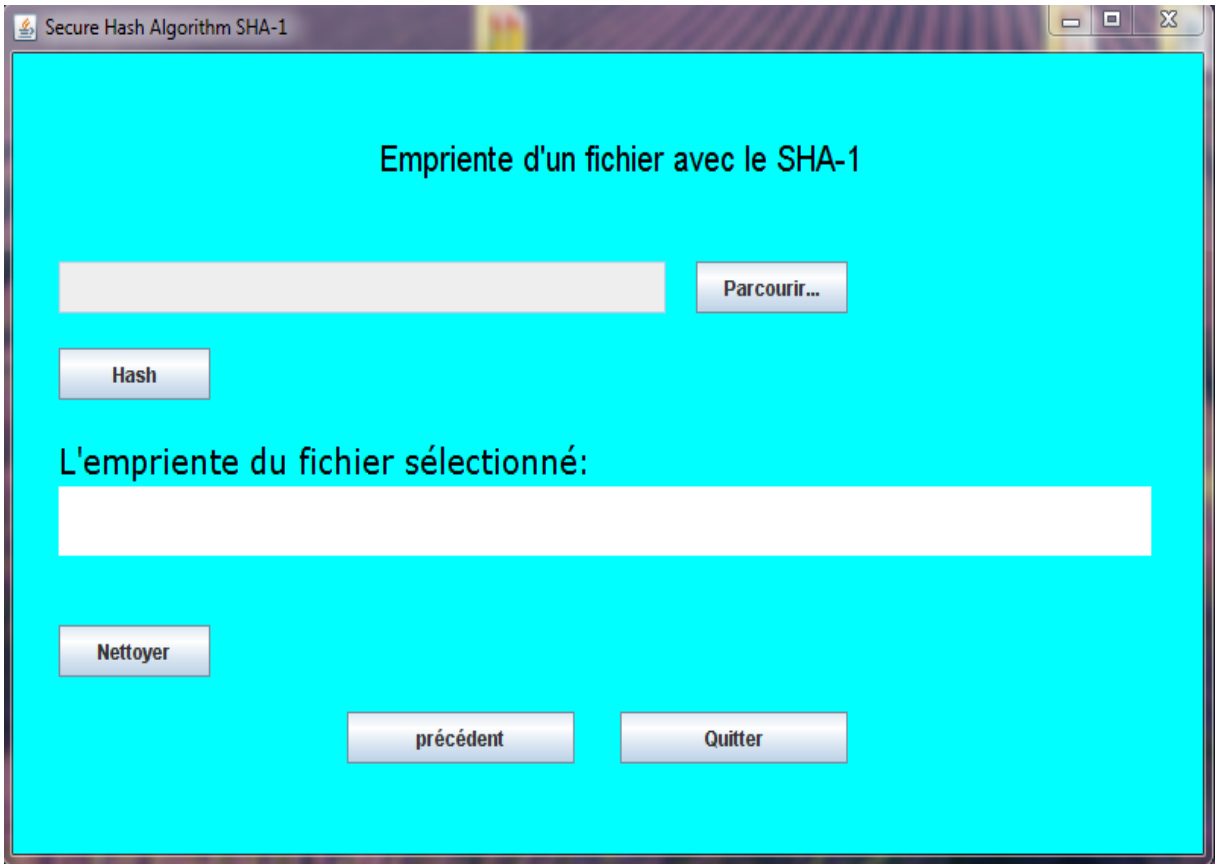


Fig.IV.4 : condenser un fichier.

- Pour condenser un fichier l'utilisateur doitcliquer sur le bouton « Parcourir », une fenêtre de dialogue s'affiche pour choisir le fichier à condenser, comme nous le montre la figure suivante :

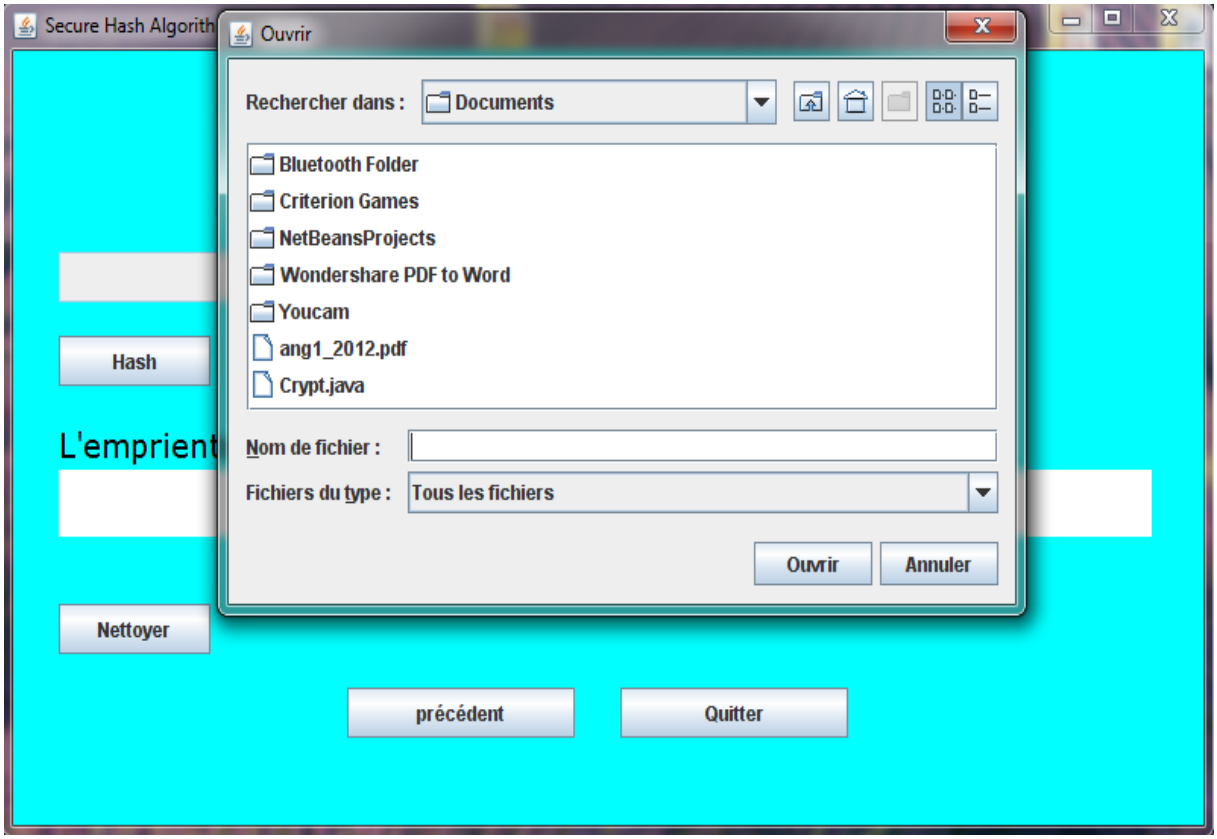


Fig.IV.5 : Fenêtre de dialogue.

Une fois l'utilisateur à cliquer sur le bouton « ouvrir »après choisi un fichier le système va importer l'adresse physique de fichier pour qu'il puisse le récupérer.

- On cliquant sur le bouton « hash » l'empreinte sera générée et une fenêtre de confirmation qui demande à l'utilisateur s'il souhaite enregistrer l'empreinte dans un fichier texte comme le montre la figure suivante :

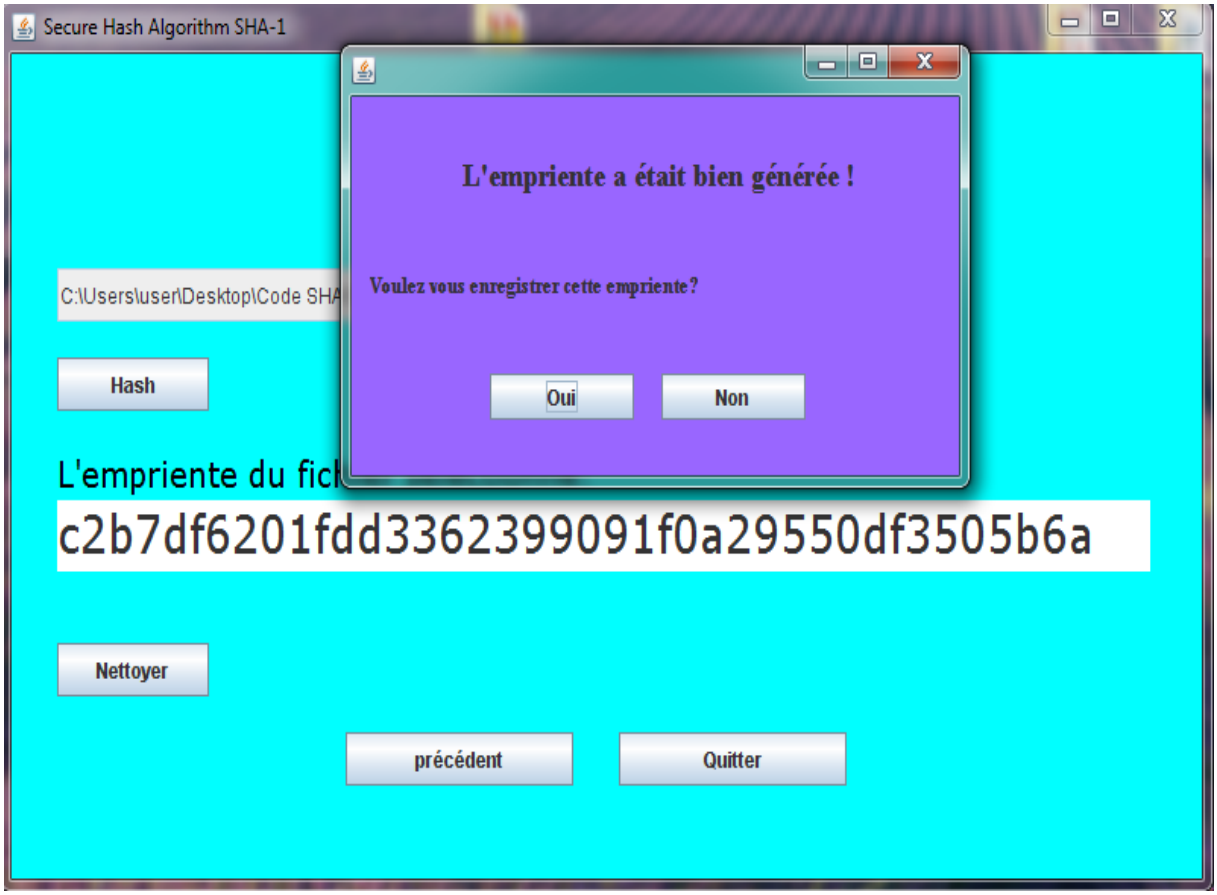


Fig.IV.6 : Interface de confirmation.

2-Message :

Si le choix se porte sur le bouton « Message », la fenêtre de « condensé un message » sera affichée comme le montre la figure suivante :

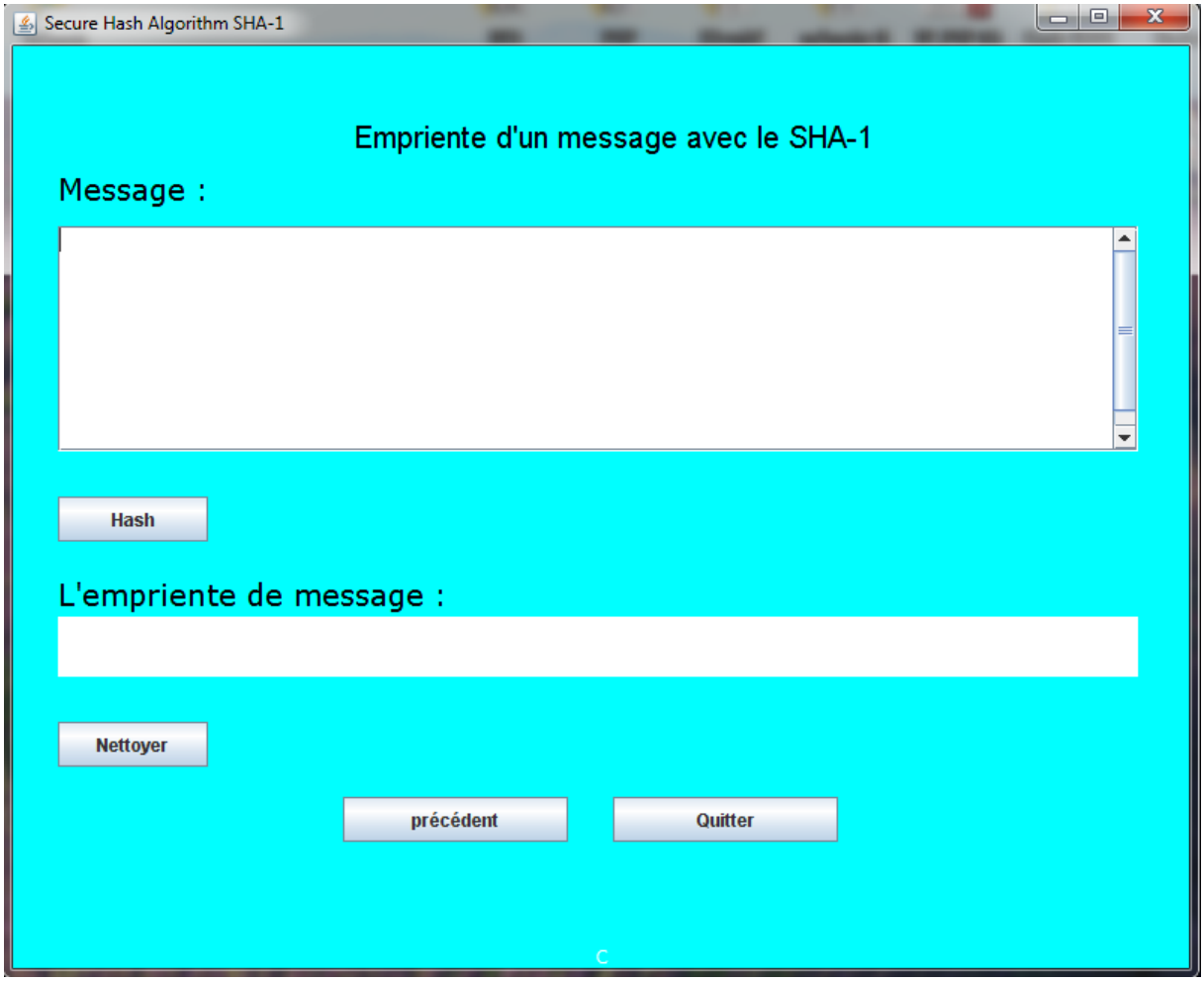


Fig.IV.7 : Condenser un message.

- L'utilisateur saisi le message à condenser et avoir son empreinte en cliquant sur le bouton Hashet une fenêtre de confirmation qui demande à l'utilisateur s'il souhaite enregistrer l'empreinte dans un fichier texte.

IV.6. Conclusion :

Dans ce dernier chapitre de notre projet, nous avons présenté le langage de programmation et l'environnement de développement quenous avons utilisé pour implémenter et décrire notre application, par la suite, nous avons présentéquelques interfaces de notre système de hachage.

Conclusion générale :

Depuis toujours la sécurité réseaux estl'un des domainesles plus sensible que connaissent les entreprises dotées d'un réseau informatique, et malgré les efforts qui sont fourni par la communauté informatique pour mieux sécurisé la transmission des informations sur le réseau, il reste toujours des failles provoqués par des hackers qui cherchent la moindre erreur pour l'exploiter et pour l'utiliser à des fins négatives contre les entreprises. Mais il y a toujours des propositions ou des solutions qui peuvent empêcher tout cela et maitre un terme à ses failles.

Dans le cadre de ce mémoire, nous avons pu appliquer les principes de la cryptographie, pour assurer l'authentification ou l'intégrité des données. Précisément la fonction de hachage SHA-1 qui nous permuter de condenser des données.

Bibliographie

[01] : cours de réseaux maitrise d'informatique, université d'Angers

[02] : Guy Rujolle , les réseaux

[03] : www.commentça arche.com

[04] :« RESEAUX », 3emme Edition, ANDREW TANENBAUM

[05] « Les réseaux », Edition 2003, GUY PUJOLLE

[06] [CLA 00] Clavel Gilles / Fagart Nicolas / Grenet David, "le standard ISO" Edition Dunod, 2000.

[07] "LINUX DEBIAN TCP/IP Les Services réseaux "; l'auteur Mikael PIRIO

[08]Signatures électroniques dans les applications INTERNET ECOLE ROYALE MILITAIRE 156e Promotion PolytechniqueLieutenant-Général Baron de GREEF

[09]« Sécuriser l'informatique de l'entreprise enjeux, menaces, prévention et parades » Jean Marc Royer

[10] « Sécuriser l'informatique de l'entreprise enjeux, menaces, prévention et parades »

Jean Marc Royer

[11] « La Sécurité sur le Web », MATTHEW DANDA

[12] [GUIL, 00] : La sécurité des réseaux Guillaume Desgeorge 2000

[13] « Sécuriser l'informatique de l'entreprise enjeux, menaces, prévention et parades » Jean Marc Royer

[14] HASSINA BENSAFIA << Conception d'une base de connaissance pour l'aide À l'investigation dans un firewall forensics>>,2002

[15] « Sécuriser l'informatique de l'entreprise enjeux, menaces, prévention et parades » Jean Marc Royer

[16] Introduction à la cryptographie Hervé Schauer Consultants.

[17] Le **C**entre de **R**essources et d'**I**nnovation **P**édagogiques (CRIP) de l'Université de Limoges Mai 2006

[18] Fouque Pierre-AlainEquipe de CryptographieEcole normale supérieure

[19]Attaques par Collision contre SHA-1 Stéphane Manuel
CRI Paris-Rocquencourt, Équipe SECRET Journées Codage et Cryptographie 4 octobre 2009 Fréjus

[20] Analyse et conception de fonctions de hachage cryptographiques THEESE préesentée et soutenue publiquement le 23 novembre 2010

Sites

WWW.hsc.fr

https://www.google.fr/search?newwindow=1&biw=1366&bih=624&q=th%C3%A8se+sur+la+cryptographie+avec+le+hachage+sha1&oq=th%C3%A8se+sur+la+cryptographie+avec+le+hachage+sha1.

https://www.google.fr/search?q=th%C3%A8se+sur+empreinte+pour+des+message+envoyer+avec+sah1+java

https://www.google.fr/search?newwindow=1&biw=1366&bih=624&q=th%C3%A8se+sur+les+fonction+hachage+sha1+empreinte

Résumé

Depuis l'apparition de l'informatique, les entreprises n'ont pas cessé de l'utiliser et de la développer selon leurs besoins pour faciliter le travail et pour mieux s'approcher de clients, dans le but d'augmenter leurs profits. En revanche, les entreprises en mal à contrôler les systèmes informatiques, entre autre, leur sécurité.

En effet, les entreprises doit faire face à la curiosité de savoir, le vol des idées ou des informations stratégiques des malfaiteurs informatiques qui cherchent toujours à développer la criminalité informatique. Tous cela, à pousser la communauté informatique à apprendre des mesures de sécurité dans la complexité de ses solutions.

Pour cela, notre travail consiste à étudier une solution existante dans le domaine de cryptographie qu'est l'un des mécanismes de sécurité qui permettent de résoudre les nombreux problèmes menaçants la vie privé et la sécurité sur internet, elle consiste a transformer un message clair en un message crypté .

Pour assurer les objectifs de la cryptographie ;la solution proposé une solution complémentaire qui consiste à appliquer une fonction mathématique sur une portion du message. Cette fonction mathématique s'appelle fonction de hachage et le résultat de cette fonction est appelé code de hachage. Ce code fait usage d'empreinte digitale du message.

Des différentes fonctions de hachage ont été proposées, nous avons basé principalement sur la fonction SHA-1 afin d'éviter les conséquences désagréables sur la sécurité d'un vol de base de données.

Pour mener à bien notre travail, nous avons structuré le présent mémoire comme suit :

- ✓ **Chapitre I** : Réseaux informatiques et leur sécurité : divisé en deux parties.
  - **Partie 1** : Généralité sur les réseaux informatiques : comprend des généralités sur les réseaux, ainsi les modèles OSI, TCP/IP et client/serveur.
  - **Partie 2** : Sécurité informatique : comprend des généralités sur la sécurité informatique, notamment la malveillance informatique et les mécanismes de sécurité.

- ✓ **Chapitre II** : Les concepts fondamentaux de la cryptographie moderne divisé en trois parties.
  - **Partie 1 :** La cryptographie : comprend des généralités sur la cryptographie classique, moderne, objectifs et techniques de cryptographie.
  - **Partie 2 :** Les fonctions de hachage : est consacrée à la présentation des fonctions de hachage, leur domaines d'utilisations, les principaux algorithmes de hachage et comment construire une fonction de hachage.
  - **Partie 3 :** Le SHA-1 : consacrée à la présentation, description de l'algorithme SHA-1 et les attaques possible sur le SHA-1.
- ✓ ***Chapitre III :*** Analyse et conception : ce chapitre est consacré à la conception de notre application où nous avons opté pour le langage UML.
- ✓ ***Chapitre IV :*** Implémentation et réalisation : comporte quand a lui la représentation de l'environnement de développement dont lequel notre application à été réalisée, les outils utilisées et quelques interfaces de notre application.

Nous terminerons notre travail par une conclusion générale.