REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE ABDERAHMANE MIRA DE BEJAIA
FACULTE DES SCIENCES EXACTES

DEPARTEMENT D'INFORMATIQUE
ECOLE DOCTORALE RESEAUX ET SYSTEMES DISTRIBUES

Mémoire de Magistère

En informatique

Option : Réseaux et Systèmes Distribués

Thème

Sécurisation des données sensibles sur téléphone mobile / dispositif d'assistant numérique personnel (PDA)

Présenté par

Wafa BIROUK

Devant le jury composé de :

Président :	Khelfaoui	Youcef	M.C.A	Université de Bejaïa, Algérie.
Rapporteur :	Melit	Ali	M. C.A	Université de Jijel, Algérie.
Examinateur :	Aliouat	Makhlouf	M.C.A	Université de Sétif, Algérie.
Examinateur :	Boukerram	Abdallah	M. C.A	Université de Sétif, Algérie.
Invité :	Bouridane	Ahmed	Professeur	Université de Newcastle, Royaume-Uni.

Promotion 2007-2008

Remerciements

« Dans la vie, les hommes sont tributaires les uns des autres.
Il y a donc toujours quelqu'un à maudire ou à remercier.»

Je tiens à adresser mes plus vifs et mes plus sincères remerciements à Monsieur Ali Melit, maître de Conférences à l'université de Jijel, et à Monsieur Ahmed Bouridane, Professeur à l'université de Newcastle pour leurs conseils avisés et pour m'avoir guidés et prodigués tout l'aide nécessaire pour la réalisation de ce travail.

Mes remerciements les plus sincères sont adressés à Monsieur Youssef Khelfaoui, Maître de Conférences à l'université de Bejaia, de m'avoir fait l'honneur de présider le jury de ce mémoire.

Je remercie vivement Messieurs Boukerram Abdallah et Aliouat Makhlouf, Maîtres de conférences à l'université de Sétif, pour l'intérêt qu'ils ont témoigné à l'égard de ce travail en me faisant l'honneur de participer à ce jury.

Que tous ceux qui ont contribué de prés où de loin à l'élaboration de ce mémoire soient remerciés en particulier : Mr Moustafa Fezani, Salim Grabsi, Houcine Ghebghoub et Aïcha Zibra.

Tous mes remerciements vont également à mes chers parents, ma famille et mes amis pour leur soutient moral qui m'ont apporté de loin ou de prés.

Table des matières

Liste des figures

Liste des tableaux

INTRODUCTION GENERALE

La flexibilité et la mobilité sont des conditions essentielles au succès, l'actualité des informations et la rapidité des décisions plus importantes que jamais dans l'industrie et la vie quotidienne des personnes. Pour communiquer et utiliser au plus vite les informations, on fait souvent appel au dispositif mobile (un PDA, téléphone mobile, Smartphone,…), qu'il permet par exemple à tous les travailleurs nomades d'accéder à leur messagerie et planning, et plus largement au système d'information de leur entreprise; les forces de vente peuvent consulter leurs stocks et leur outil de facturation depuis les locaux de leur client; les médecins peuvent accéder aux dossiers de leurs patients depuis les salles d'hospitalisation; …etc.
En effet, le développement fulgurant des moyens de communication modernes a permis une large distribution des téléphones portables et PDA ces dernières années, et ils sont désormais courantes dans de nombreux endroits du monde, et devrait progressivement remplacer le téléphone classique, compte tenu des avantages qu'ils proposent en termes de souplesse d'utilisation. Actuellement, la plupart des personnes possèdent un téléphone mobile. Bien que ces dispositifs soient connus en tant que *téléphones* dus à leur fonction d'origine où son propriétaire sauvegarde dedans une liste des numéros de ses contactes, les capacités de ces appareils ont été élargies ces dernières années à de nombreux autres domaines, par exemple, de surfer sur Internet, d'échanger du courrier, les fonctions multimédias permettent de prendre des photos, d'enregistrer de la vidéo et du son, et de jouer de la musique.

L'utilisation généralisée des dispositifs mobiles, ainsi que leurs fonctions de plus en plus développées entraînent des risques de sécurité pour les personnes qui les utilisent comme des ordinateurs portables. Le fait d'être un objet de poche que l'on emporte en permanence sur soi, risque son perte ou son vol qui peut compromettre ses données confidentielles que son utilisateur ne pas vouloir que soient disponibles à n'importe qui autre que lui-même. Prenons pour exemple une entreprise qui se fait voler un PDA contenant des données (business plan,

prévisions de vente, grille tarifaire, etc.) ou des informations (par exemple, l'annonce d'une transaction) sensibles d'un point de vue commercial ou stratégique. D'après Financial News, 2.8 millions de téléphones portables sont perdus ou volés chaque année en Corée du Sud [82], en plus, 57% des usagers ne chiffrent pas les données de l'entreprise qu'ils stockent sur leur PDA [84]. Pour cette raison, et pour éviter ce problème, la cryptographie est l'un des mécanismes de base utilisés pour répondre à ses besoins de sécurité. Elle représente l'espoir de la sécurité de l'informatique mobile.

La cryptographie, est l'étude des techniques mathématiques relatives aux aspects de sécurité de l'information, telles celles concernant la confidentialité, l'intégrité ou l'authentification des données ou de leur origine [26]. Les algorithmes cryptographiques *modernes* c'est-à-dire celles étant apparues et utilisées après la Seconde Guerre mondiale se répartissent en deux grandes familles [83] :

Les algorithmes de chiffrement symétrique sont ceux qui utilisent la même clé pour chiffrer et déchiffrer un message. Ils sont souvent basés sur des techniques de substitutions et de transpositions. Cela offre un moyen rapide et efficace pour chiffrer un message. Les algorithmes les plus utilisés sont DES (Data Encryption Standard) et l'AES (Advanced Encryption Standard). Les algorithmes de chiffrement asymétrique sont ceux qui utilisent une clé de chiffrement différente de celle de déchiffrement. Il s'agit d'utiliser une paire de clés, une clé *publique* pour chiffrer et une autre clé *secrète* que seul le propriétaire détient pour déchiffrer. Les chiffrements asymétriques sont la plupart du temps basés sur l'existence de fonctions mathématiques dites à *sens unique* ou *sens unique avec trappe* [83]. L'algorithme le plus utilisés est RSA (Rivest Shamir Adleman) [26].

Ainsi, le travail présenté dans ce mémoire va porter sur le premier type des algorithmes de chiffrement et plus précisément sur l'algorithme AES (Advanced Encryption Standard) [63] qui est le plus récent algorithme à clé symétrique, en plus, c'est le plus fiable, efficace et fort des algorithmes de chiffrement disponibles aujourd'hui.

Cet algorithme est utilisé pour développer une application de chiffrement pour sécuriser tous les données sensibles sur les dispositifs mobiles (PDA / téléphones mobiles), mais le développement des applications sur ces dispositifs est fortement contraint par les limitations de ces derniers (une mémoire limitée, puissance de traitement limitée, consommation d'énergie de processeur,…) [70]. C'est pour cette raison que l'outil de développement Studio Visuel [75] de Microsoft est proposé d'utiliser comme environnement et plateforme pour développer cette application.

Ce mémoire est structuré en quatre chapitres encadrés par une introduction générale et une conclusion et perspectives :

Le premier chapitre, tente, après une description des principes fondateurs de la cryptographie qui aborde sa notion, ses fonctionnalités ainsi que les grandes menaces qui peuvent la subir, de faire une description complète de différents algorithmes cryptographiques, que ce soit la cryptographie classique ou moderne.

Le deuxième chapitre est réservé à la description des dispositifs mobiles sur lesquels on implémente notre application. L'accent est mis particulièrement sur les PDA et les téléphones mobiles où leurs caractéristiques principales, leur structure, ainsi que leurs fonctionnements sont présentés.

Le troisième chapitre est une étude détaillée de l'algorithme cryptographique symétrique AES (Advanced Encryption Standard) [63]. Cette étude constitue une plateforme pour pouvoir implémenter l'application de chiffrement.

Le quatrième chapitre est réservé à la description de l'application de chiffrement que nous proposons comme solution à la problématique de sécurisation des données sensibles sur les dispositifs mobiles (PDA / téléphones mobiles) en utilisant l'algorithme cryptographique AES en mode CBC [24], et l'environnement de développement Studio Visuel [75] de Microsoft comme une plateforme pour développer cette application.

CHAPITRE 1

Cryptographie

CHAPITRE 1
Cryptographie

1.1 Introduction

Depuis toujours, l´être humain a cherché à conserver certaines informations ou données secrètes, à défaut, à en restreindre l´accès à certaines personnes. C'est pourquoi, et dès l'antiquité, les peuples employèrent des codes secrets dans certains de leurs textes : les archéologues en ont découvert dans des hiéroglyphes égyptiens [1]. De même, les Hébreux dissimulaient parfois leurs écrits en inversant l'alphabet.

Jusqu'au début du XX^{ème} siècle, la cryptographie a gardé une importance mineure, et les méthodes utilisées étaient bien souvent rudimentaires [2]. Mais lors de la seconde guerre mondiale, et avec l´apparition de technologies de communication évoluées, telles que la radio, a rendu nécessaire la mise au point de mécanismes de cryptage pour chiffrer les données transmises par les ondes (Enigma) empêchant leurs interceptions par l´ennemi.

Avec l'avènement des réseaux, et tous particulièrement Internet, la cryptographie prend maintenant une nouvelle dimension, économique cette fois [3]. Ainsi la cryptographie s'élargie du domaine confidentiel de la protection des gros serveurs (universités, entreprises, état) à la consommation de masse par les particuliers (commerce électronique, confidentialité des mails,…). À l'inverse, la cryptanalyse (craquage des codes cryptés) change elle aussi d'acteurs et d'objet. Jusqu'alors arme militaire et jeu de quelques génies travaillant pour la célébrité, elle devient une véritable arme de vol à grande échelle (détournement de codes de carte bleue, de fonds,...) et de guerre économique (vol de secrets industriels ou commerciaux).

Dans ce chapitre, nous allons d'abord présenter les principes fondateurs de la cryptographie qui aborde sa notion, ses fonctionnalités ainsi que les grandes menaces qui peuvent la subir. Par la suite, nous nous décrivons les différentes classes d'algorithme cryptographique, que ce soit la cryptographie classique ou moderne avec la présentation des exemples des plus fameux algorithmes dans chaque classe.

1.2 Principes fondateurs de la cryptographie

1.2.1 Notion de cryptologie

La *cryptographie* appartient à la **cryptologie** du grec *kruptos* « secret, caché » et *logos* « discours », qui est la science de l'écriture secrète englobant des pratiques concurrentes à savoir la **cryptographie**, le **déchiffrement** et la **cryptanalyse**. La première pratique qui est la cryptographie (du grec *kruptos* et *graphein* [4]) est : « la discipline incluant les principes, les moyens et les méthodes de transformation des données, dans le but de masquer leur contenu, empêcher leur modification ou leur utilisation illégale, ainsi que les opérations inverses, pour rendre le document à nouveau intelligible » [5]. Le déchiffrement est le processus permettant de transformer le message chiffré en message clair. Quand à la cryptanalyse, qui est un terme créé par le cryptologue américain *William Friedman* en 1920 (du grec *kruptos* et *analusis* « résolution, dissolution » [4]), est l'art de décoder un message chiffré en mêlant une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination, et de chance.
Les deux disciplines de cryptographie et de cryptanalyse s'alimentent l'une l'autre [6]. On ne peut pas évaluer la sécurité d'un mécanisme sans le soumettre à des attaques.

1.2.2 Terminologie

La cryptologie, et par conséquent la cryptographie, est essentiellement basée sur l'arithmétique [7]. Il s'agit dans le cas d'un texte de transformer les lettres qui composent le message en une succession de chiffres, puis ensuite de faire des calculs sur ces chiffres pour :

- ✓ d'une part les modifier de telle façon à les rendre incompréhensibles. Le résultat de cette modification (le message chiffré) est appelé **cryptogramme** (en anglais *ciphertext*) par opposition au message initial, appelé **message en clair** (en anglais *plaintext*) ;
- ✓ faire en sorte que le destinataire saura les déchiffrer.

Le chiffrement se fait généralement à l'aide d'une *clef de chiffrement*, le déchiffrement nécessite quant à lui une *clef de déchiffrement*. On distingue généralement deux types de clés :

- **Les clés symétriques:** il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de *chiffrement symétrique* ou de *chiffrement à clé secrète*.
- **Les clés asymétriques:** dans ce cas, une clé différente est utilisée pour le chiffrement et le déchiffrement. On parle alors de *chiffrement asymétrique* ou de *chiffrement à clé publique.*

On appelle *décryptement* (*décryptage*) le fait d'essayer de *déchiffrer illégitimement* le message (que la clé de déchiffrement soit connue ou non de l'*attaquant*). Lorsque la clé de déchiffrement n'est pas connue de l'attaquant on parle alors de **cryptanalyse** ou **cryptoanalyse** (on entend souvent aussi le terme plus familier de *cassage*).

Le processus cryptographique peut être récapitulé par la figure ci-dessous :

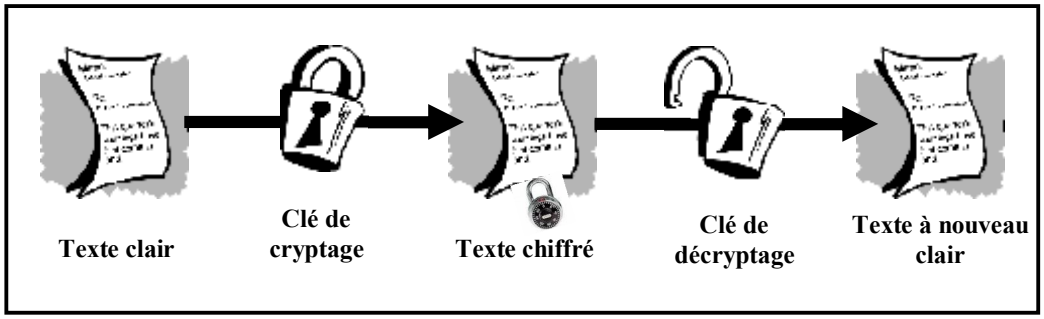


Figure 1.1. *Processus cryptographique*

1.2.3 Les grandes menaces et les fonctionnalités de la cryptographie

1.2.3.1 Les grandes menaces

De façon générale, les grands types de menaces que peut subir un message lors de son échange peuvent être récapitulés à travers les points suivants :

a. Les attaques passives : avec ce type d'attaque et lors d'une communication élaborée entre deux personnes souvent nommées *Alice* et *Bob*, *Oscar* qui est l'attaquant, se contente d'écouter le message tout en essayant de menacer sa confidentialité. Dans ce cas, il se peut qu'une information secrète parvienne également à une personne autre que son destinataire légal.

b. Les attaques actives : ici, *Oscar* peut menacer l'intégrité, qui sera présentée dans la section suivante. Ainsi, ces informations vont parvenir d'une personne autre que leur véritable auteur. Et comme exemple d'attaques actives, on peut citer [8] :

- ✓ l'usurpation d'identité (de l'émetteur ou du récepteur) ;
- ✓ l'altération / modification du contenu des messages ;
- ✓ la destruction de messages/ le retardement de la transmission ;
- ✓ la répétition de messages (jusqu'à engorgement) ;
- ✓ la répudiation de message : l'émetteur nie avoir envoyé le message.

c. La cryptanalyse : elle permet d'étudier la sécurité des procédés de chiffrement utilisés en cryptographie. Ainsi, elle désigne habituellement les techniques qui permettent d'extraire de l'information sur des secrets (le message clair et la clé) en observant uniquement les données publiques d'un cryptosystème. Ce qui compte avant tout dans une cryptanalyse, c'est de gagner de l'information sur le message clair. Ceci dit, il va de soi que gagner de l'information sur la clé de chiffrement privé permettant de déchiffrer tous les messages. Et suivant les données qu'elle nécessite, on distingue habituellement quatre méthodes de cryptanalyse :

- **attaque sur texte chiffré seul (ciphertext-only) :** le cryptanalyste possédant des exemplaires chiffrés des messages, essaye de faire des hypothèses sur les messages originaux qu'il ne possède pas en vue de retrouver la clé de déchiffrement. Dans ce cas, la cryptanalyse sera très difficile à cause du manque d'informations à disposition.
- **attaque à texte clair connu (known-plaintext attack) :** le cryptanalyste essaye de retrouver la clé de déchiffrement à partir de messages ou de parties de messages en clair possédés et de leurs versions chiffrées correspondantes.
- **attaque à texte clair choisi (chosen-plaintext attack) :** consiste à retrouver la clé de déchiffrement à partir de messages en clair, et en ayant la possibilité de générer les versions chiffrées de ces messages avec un algorithme considérer comme une boîte noire.
- **attaque à texte chiffré choisi (chosen-ciphertext attack) :** le cryptanalyste possède des messages chiffrés et demande la version en clair de certains de ces messages pour mener l'attaque (retrouver la clé de déchiffrement).

1.2.3.2 Les fonctions de la cryptographie

La cryptographie est utilisée pour dissimuler des messages clairs aux yeux de certains utilisateurs pour assurer leur *fiabilité* et *confidentialité* surtout s'ils feront l'objet des communications via Internet. Désormais, les fonctions de la cryptographie se sont étendues pour englober de nouvelles fonctions, il s'agit de garantir l'*intégrité* et l'*authenticité* des données échangées. Ceux-ci sont les fonctions principales de la cryptographie. Elle a d'autres fonctions, dites secondaires, qui sont [9] : l'*horodatage*, le *témoignage*, l'*accusé de réception* et la *révocation.*

a. La confidentialité : permet de protéger le contenu des informations sauvegardées ou transmises sur un réseau. Seules les personnes autorisées doivent pouvoir accéder aux informations ainsi protégées. Le *chiffrement de l'information* permet de résoudre le problème de la confidentialité : une personne souhaitant transmettre un message lui applique une

fonction dite de chiffrement, et transmet le résultat au destinataire. Ce dernier retrouve le message original en utilisant une fonction de déchiffrement suivant le modèle de la cryptographie utilisé (à clé secrète ou à clé publique).

b. L'intégrité : c'est la capacité à reconnaître qu'une information a été altérée [10], soit de manière accidentelle ou intentionnelle.

c. L'authentification : consiste à assurer l'identité d'un utilisateur, c'est à dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il doit être.
On distingue deux types d'authentification :

- **Authentification d'un tiers :** c'est l'action qui consiste à prouver son identité. Ce service est généralement rendu par l'utilisateur d'un « échange d'authentification » qui implique un certain dialogue entre les tiers communicants.
- **Authentification de l'origine des données :** elle sert à prouver que les données reçues ont bien été émises par l'émetteur déclaré. Dans ce cas, l'authentification désigne souvent la combinaison de deux services : authentification et intégrité.

d. La non-répudiation : La non-répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction [11].

Ces fonctionnalités représentent des solutions aux problèmes causés par les menaces citées précédemment, ainsi :

- ✓ Pour assurer la confidentialité, on utilise un algorithme de chiffrement.
- ✓ Contre l'usurpation d'identité, on utilise des algorithmes d'authentification.
- ✓ Pour éviter l'altération de données, on utilise des algorithmes de contrôle d'intégrité.
- ✓ Contre la répudiation, des algorithmes de signatures ont été proposés.

Les besoins de sécurisation et de confidentialité s'imposent à divers degrés dans différentes applications. Citons à titre d'exemple :

- ✓ Confidentialité des transactions bancaires,
- ✓ Protection de secrets industriels ou commerciaux,
- ✓ Protection des secrets médicaux,
- ✓ Protection des systèmes informatiques contre les intrusions,
- ✓ Protection de la confidentialité des communications
- ✓ Protection de la vie privée,
- ✓ …etc.

1.3 Algorithmes cryptographiques

Comme nous l'avons déjà mentionné, le but de la cryptographie est de permettre à deux personnes de s'échanger des informations en toute sécurité à travers un canal peu sûr, qui peut être une ligne téléphonique ou tout autre réseau de communication. L'information que l'on souhaite transmettre et que l'on appelle texte clair, sera donc chiffrée par un procédé de chiffrement et en utilisant une clé prédéterminée. Le destinataire est le seul qui peut retrouver l'information originale suite à une opération de déchiffrement de cette information chiffrée en utilisant une clé de déchiffrement sans laquelle ce procédé est impossible.

Le processus de chiffrement ou de déchiffrement utilise une fonction mathématique. C'est l'**algorithme cryptographique** ou encore appelé **chiffre**. La sécurité des données chiffrées est entièrement dépendante de deux choses : la force de l'algorithme cryptographique et le secret de la clé. Un algorithme cryptographique, plus toutes les clés possibles et tous les protocoles qui le font fonctionner constituent un **cryptosystème**.

1.3.1 Le principe de kerckhoffs

Longtemps, la sécurité d'un système cryptographique a reposé sur le secret qui l'entoure (cas du chiffre de César, …). Cette idée s'est ensuite abandonnée du fait qu'un tel secret peut toujours être révélé par un espion. C'est pourquoi un système cryptographique doit dépendre d'un paramètre aisément modifiable : sa clé.

Le premier à avoir formalisé ce principe est le hollandais *Auguste Kerckhoffs* en écrivant en janvier 1883 dans le « Journal des sciences militaires » un article intitulé « La cryptographie militaire » [12]. Il a annoncé que l'attaquant peut posséder tous les détails de l'algorithme sans pouvoir rien faire puisqu'il lui manque la clé pour le chiffrement. Cela mène vers la certitude suivante : si on ne sait pas casser un algorithme même en sachant comment il fonctionne, on ne sait certainement pas le casser sans cette connaissance. Donc, un chiffre basé uniquement sur le secret de l'algorithme n'a aucun intérêt, car un jour ou l'autre ce secret sera découvert ou volé. Par exemple, même si on connaît le mode d'emploi du carré de Vigenère (voir la section 3.3.1.1.b), on ne pourra quand même pas, ou difficilement, déchiffrer un message si on ne connaît pas la clé. Par contre, le chiffre de César repose entièrement sur la manière de chiffrer.

Par fragilité, *Bruce Schneier* accentue sur le fait de garder comme secret une information peu coûteuse à remplacer en cas où le secret sera divulgué. En bref, moins on a de secrets, moins on doit faire de maintenance [13].

1.3.2 Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplet (P, C, K, E, D) satisfaisant les points suivants [14] :

1) P est un ensemble fini de blocs de textes clairs possibles.

2) C est un ensemble fini de blocs de textes chiffrés possibles.

3) K est un ensemble fini de clés possibles.

4) Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondante $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k(e_k(x)) = x$ pour tout texte clair $x \in P$.

La principale propriété est la quatrième. Elle précise que si un texte clair x est chiffré en utilisant e_k, et si le texte chiffré y obtenu est ensuite déchiffré en utilisant d_k, on retrouve le texte clair x original.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$. Supposant qu'ensuite, Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne :

$$x = x_1\, x_2 \ldots x_n \qquad \text{avec : } n \in Z,\ n \geq 1,\ x_i \in P \text{ et } 1 \leq i \leq n.$$

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera: $y = y_1\, y_2 \ldots y_n$

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1\, x_2 \ldots x_n$. Le procédé de communication est illustré sur la figure I.2.

Il est évident que chaque fonction de chiffrement e_k doit être injective [14] (c'est à dire ne pas chiffrer deux blocs différents en deux valeurs égales), sinon, le procédé de déchiffrement ne pourrait être fait sans ambiguïté.

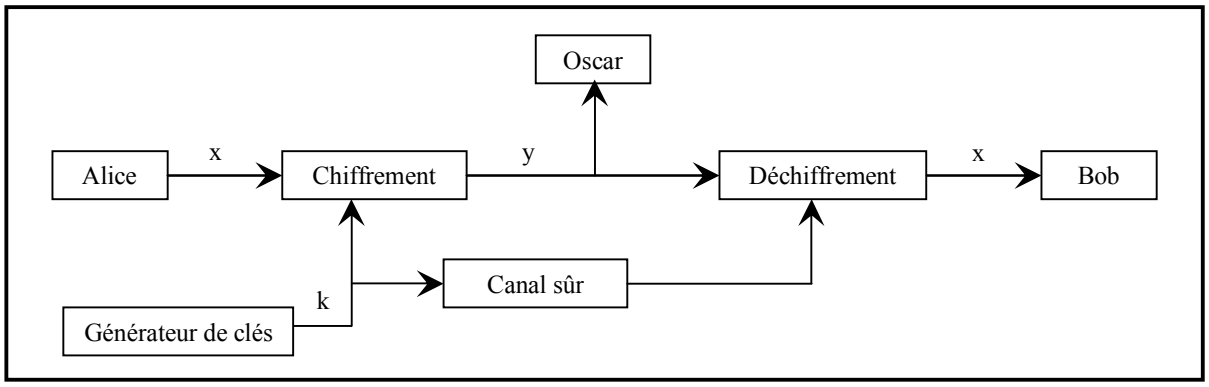


Figure 1.2. *Le procédé de communication*

Les principes fondamentaux d'un algorithme de cryptographie sont basés sur deux notions essentielles, énoncées par *Shannon* en 1949 :

- **Confusion :** sert à cacher la relation entre le clair et le chiffré [15]. Ceci peut se faire par une substitution méthodique de symboles, ou par un algorithme de codage. Comme ça aucune propriété statistique ne peut être déduite du message chiffré [16].
- **Diffusion :** sert à cacher la redondance dans le message et à diffuser sur tout le chiffré l'influence du changement d'un bit de clé ou d'un bit du clair [15]. Donc, chaque symbole chiffré doit dépendre de beaucoup de symboles en clair. Ainsi, toute modification du message en clair se traduit par une modification complète du chiffré [16].

1.3.3 Classes de la cryptographie

Le schéma suivant présente les différentes classes de la cryptographie :

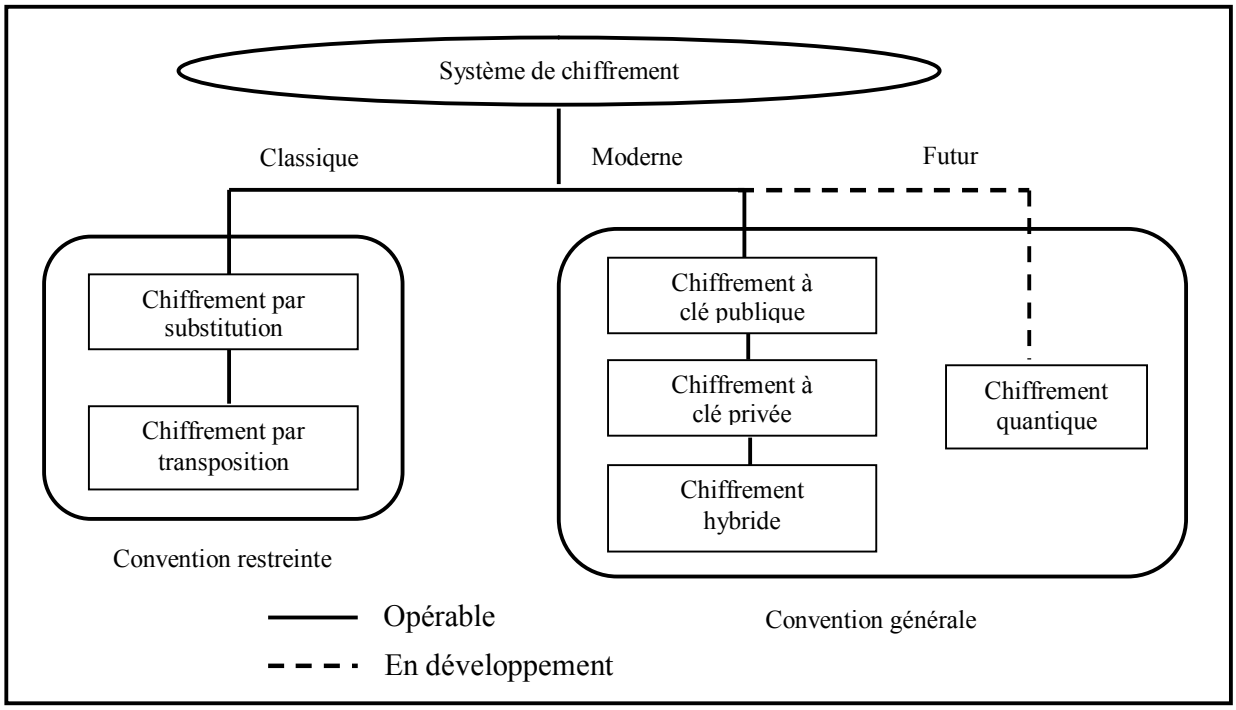


Figure 1.3. *Les classes de la cryptographie*

1.3.3.1 La cryptographie classique

Ce mode décrit la période avant les ordinateurs durant laquelle, les principaux outils utilisés consistent à remplacer des caractères par d'autres et les transposer dans des ordres différents tout en gardant secrètes les procédures de chiffrement ou de déchiffrement. On appelle généralement cette classe de méthodes : le chiffrement **à usage restreint**.

1.3.3.1.1 Cryptographie par substitution

Dans ce mode de cryptage, les lettres du message en clair sont remplacées par d'autres lettres, des chiffres ou d'autres symboles. Selon la façon de substituer, on distingue :

a. Substitution mono-alphabétique : c'est le plus simple des codages à réaliser. Il s'agit de remplacer chaque lettre par une lettre différente, ou même un autre symbole. Plus formellement [17] :

$$f : A_M \longrightarrow A_C$$
$$c_i = f(m_i) = m_i + k \ (\text{mod } |A_M|)$$

Où :

A_M : l'ensemble d'alphabets du message en clair

A_C : l'ensemble d'alphabets du message chiffré.

$M = m_0\ m_1 \ldots m_{n-1}$ tel que : $\forall i, m_i \in A_M$.

$C = c_0\ c_1 \ldots c_{n-1}$ tel que : $\forall i, c_i \in A_C$.

La plus ancienne des méthodes s'inscrivant sous ce mode de chiffrement est le **chiffre de César** utilisé par l'armée romaine (1^{ier} siècle avant JC). Il consiste à décaler les lettres de l'alphabet d'un nombre *n*. Par exemple, pour n=3, A sera remplacé par D, B par E,... Son principe très simple à mettre en œuvre facilite sa cryptanalyse du fait que, le nombre de façon de chiffrer un message reste très faible, puisqu'il est égal au nombre de lettres de l'alphabet (26 façons seulement). Une autre attaque possible contre ce système est la cryptanalyse fréquentielle qui se base sur le fait que les lettres les plus fréquentes dans le texte en clair restent les plus fréquentes dans le texte chiffré. Donc, les algorithmes à base de substitutions mono-alphabétiques sont facilement cassés par les spécialistes.
Il est à noter que le code de César a été utilisé sous le nom de ROT13 (rotation de 13 lettres où A-->N...). Son utilisation est simple: il suffit de re-chiffrer un texte, codé en ROT13, une deuxième fois pour obtenir le texte en clair.

Dans ce même mode de chiffrement, et lorsqu'une même lettre sera substituée par plusieurs lettres qui seront bien déterminées à l'avance; par exemple, 'A' peut correspondre à 5, 13, 25 ou 56 ; 'B' à 7, 19, 31, ou 42,…; cette façon particulière de substitution mono-alphabétique est appelée **Substitution homophonique**. Ce procédé est plus sûr que le précédant, mais aussi craqué par les cryptanalystes ou par des espions expérimentés.

Dans cette catégorie, on peut citer aussi : les **alphabets désordonnés**, le **chiffre affine** [18],...

b. Substitution poly-alphabétique : aussi appelée **à alphabets multiples**. Elle a été inventée par *Trithemius* en 1518 et cryptanalyser par *Kasiski* en 1863 [19]. Avec cette méthode, une même lettre peut être remplacée par plusieurs symboles pris aléatoirement. Cela est garantie grâce à une clé $k = k_0\, k_1 \ldots k_{j-1}$ qui définie j fonctions distinctes définies comme suit [17] :

$$\forall i : 0 \le i < n \qquad f_{k_l} : A_M \longrightarrow A_C \qquad \forall l : 0 \le l < j$$
$$c_i = f_{k_l \bmod j}(m_i) = m_i + k_{i \bmod j} (\mathrm{mod} \left| A_M \right|)$$

Avec A_M, A_C, M et C auront la même signification que dans la substitution précédente.

L'exemple le plus fameux de chiffre poly-alphabétique est le **chiffre de Vigenère**, qui a résisté aux cryptanalystes pendant trois siècles. Ce chiffre a été présenté en 1586 par le diplomate français *Blaise de Vigenère*. Il utilise les 26 alphabets écrits en carré (Figure 1.4).

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

Figure 1.4. *Le carré de Vigenère*

Pour coder un message en utilisant ce chiffre, on choisit une clé de longueur arbitraire et on la répète selon la longueur du message à coder. On l'écrit ensuite au dessus du message à coder lettre par lettre. La première lettre du message chiffré sera la lettre située à l'intersection de la ligne correspondant à la première lettre de la clé avec la colonne correspondant à la première lettre du texte à chiffrer (texte clair), et on continu ainsi jusqu'à terminer le chiffrement de tout le texte. Pour déchiffrer, il suffit de faire la même opération en sens inverse, c'est à dire

que sur la ligne de la lettre de la clé on recherche la lettre du message codé, la véritable lettre se trouve alors au sommet de la colonne correspondante.
Son point fort, c'est qu'il offre des modes de codage et de décodage faciles à appliquer, et son plus grand intérêt est que la même lettre sera codée de différentes manières en pénalisant ainsi toute tentative de cryptanalyse fréquentielle à condition que la longueur du message à chiffrer ne soit pas bien plus longue que celle de la clé.

1.3.3.1.2 Cryptographie par transposition

Ici, c'est l'ordre des éléments d'une information qui est modifié (caractères d'une phrase, pixels d'une image...). Plusieurs types de transposition existent :

a. Transposition simple par colonnes : on écrit le message horizontalement dans une matrice prédéfinie, et pour retrouver le texte chiffré, on lit la grille verticalement. Le procédé inverse représente le procédé de déchiffrement. La figure ci-dessous résume ce principe.

Matrice [6,7]

l	a	c	r	y	p	t
o	g	r	a	p	h	i
e	e	s	t	u	n	d
o	m	a	i	n	e	p
a	s	s	i	o	n	n
a	n	t				

Texte clair : la cryptographie est un domaine passionnant
Texte chiffré : loeoa aagem sncrs astra tiiyp unoph nenti dpn

Figure 1.5. *Transposition simple par colonnes*

b. Transposition complexe par colonnes : un mot clé secret constitué uniquement de caractères différents est utilisé pour construire une séquence de chiffres représentant les ordres d'apparition dans l'alphabet des différentes lettres composant ce mot. Le chiffrement se fait en écrivant d'abord le message par lignes dans un rectangle, comme le montre la figure I.6, puis on lit le texte par colonnes en suivant l'ordre déterminé par la séquence.

Clé : c r y p t o

1	**4**	**6**	**3**	**5**	**2**
l	a	c	r	y	p
t	o	g	r	a	p
h	i	e	e	s	t
u	n	d	o	m	a
i	n	e	p	a	s
s	i	o	n	n	a
n	t				

texte clair : la cryptographie est un domaine passionnant
texte chiffré : lthui snppt asarr eopna oinni tyasm ancgs deo

Figure 1.6. *Transposition complexe par colonnes*

c. Transposition par carré polybique : un mot clé secret est utilisé pour construire un alphabet dans un tableau, permettant d'extraire les coordonnées des lignes et des colonnes correspondant aux lettres du texte à chiffrer. Ainsi, chaque lettre du texte en clair est représentée par deux chiffres écrits verticalement sur deux lignes. L'étape qui suit, consiste à concaténer les deux lignes obtenues précédemment pour avoir une seule ligne de chiffres, puis à recombiner ces chiffres deux par deux pour obtenir les coordonnées de lignes et de colonnes du texte chiffré. Comme la montre la figure suivante (sur le même exemple de texte clair).

Clé : crypto

Coordonnées du texte clair :

22111111212144225133215243212554133231
36123456526426442554161664446226644645

	1	2	3	4	5	6
1	c	r	y	p	t	o
2	d	q	l	e	g	a
3	f	w	z	n	u	v
4	j	h	b	k	x	i
5	m	s	§	£	{	|
6	%	«	&	)	@	#

Texte fractionné groupé en 2 et recombiné en coordonnées :
22 11 11 11 21 21 44 22 51 33 21 52 43 21 25 54 13 32 31 36 12 34 56 52 64 26 44 25 54 16 16 64 44 62 26 64 46 45
q c c c r r k q t z r g n r s x f l y & d b @ g i « k s x % % i k a « i) £

Texte chiffré après division des mots: qcccr rkqtz rgnrs xfly& db@gi «ksx% %ika« i)£

Figure 1.7. *Transposition par carré polybique*

1.3.3.2 La cryptographie moderne

Avec le développement des ordinateurs, les techniques de cryptographie ont clairement évolué. Malgré ça, les procédés de substitution et de transposition restent toujours d'actualité mais en manipulant, cette fois-ci, des séquences de bits du fait que les ordinateurs ne manipulent que des données numériques. D'autre part, il fait que maintenant les algorithmes ne sont plus cachés, mais au contraire sont connus de tous et leurs sécurité est liée seulement aux clés utilisées.

La cryptographie moderne se scinde en deux parties nettement différenciées :

- ✓ La **cryptographie à clé secrète**, ou encore appelée **symétrique**;
- ✓ La **cryptographie à clé publique**, dite également **asymétrique**.

La première, qui est la cryptographie symétrique, est la plus ancienne, et on peut la faire remonter à l'Égypte de l'an 2000 avant. J.C; la seconde, qui est la cryptographie asymétrique, remonte à l'article de *W. Diffie* et *M. Hellman*, « New directions in cryptography » daté de 1976 [20].

1.3.3.2.1 La cryptographie symétrique

a. Principe

Les algorithmes de chiffrement à clé secrète (ou symétriques ou encore dits conventionnels) sont ceux pour lesquels l'émetteur et le destinataire partagent une même clé secrète qui est échangé préalablement à travers un canal sécurisé par exemple, autrement dit, les clés de chiffrement et de déchiffrement sont identiques [21].

Un paramètre essentiel pour la sécurité d'un système à clé secrète est la taille de l'espace des clés. En effet, il est toujours possible de mener sur un algorithme de chiffrement, une attaque dite *exhaustive* pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clés possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si les clés correspondent à des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1}. Donc, pour pénaliser une telle attaque, il faut que l'espace des clés soit suffisamment grand.
Il existe d'autres types d'attaques sur ce type de système dont la plupart consistent à exploiter certaines structures particulières de l'algorithme ou certaines caractéristiques statistiques dans les couples de textes clairs-chiffrés. Les plus connues sont la *cryptanalyse différentielle*, inventée par les Israéliens *Biham* et *Shamir* en 1991 [21], et la *cryptanalyse linéaire* dont le principe a été initialement développé par *Gilbert, Chassé et Tardy-Corfdir* [22].

Les algorithmes symétriques sont de deux types différents :

1. Les algorithmes de **chiffrement en continu** : qui agissent sur le texte en clair un bit à la fois, et n'ont pas besoin de le découper. Ce mode de chiffrement est encore appelé **chiffrement par flot** (*Stream cipher* en anglais) ou **chiffrement de flux** ;
2. Les algorithmes de **chiffrement par blocs** (*Bloc cipher* en anglais) : qui consistent à diviser le texte clair en blocs de taille fixe (généralement 64 ou 128 bits) et chiffrent un bloc à la fois avec la même clé [23]. Ils permettent une meilleure sécurité. On trouve [24] :

   - ✓ **ECB** *Electronic Code Book*. Cas le plus simple : chaque bloc est chiffré ou déchiffré de manière indépendante des autres; ce mode est très vulnérable aux attaques.

   - ✓ **CBC** *Cipher Block Chaining*. C'est le mode le plus courant. Il permet d'introduire une complexité supplémentaire dans le processus de cryptage en créant une dépendance entre les blocs successifs. Pour commencer, un vecteur d'initialisation est combiné

avec le texte clair. Ensuite le résultat de chaque chiffrement de bloc est combiné avec le texte en clair du bloc suivant. Le déchiffrement se fait de façon similaire.

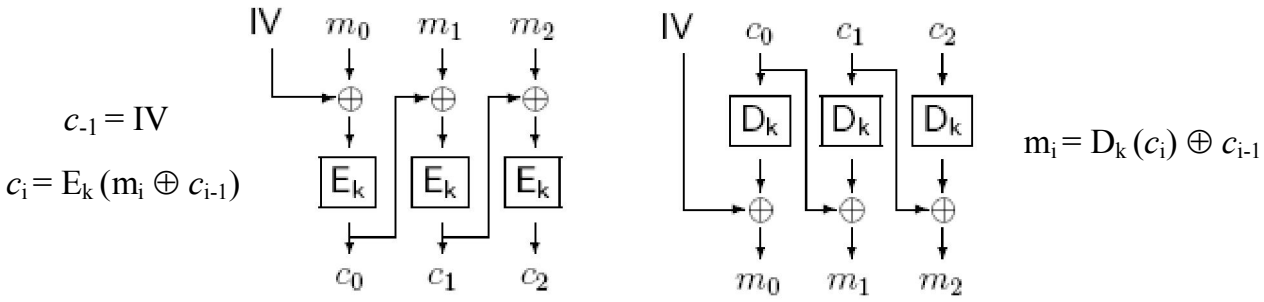


Figure 1.8. *Chiffrement et déchiffrement en mode CBC*

- ✓ **CFB** *Cipher Feedback*. Est un mode qui autorise une utilisation plus souple. En effet, le mode CFB va permettre de chiffrer des blocs dont la longueur pourra varier de n à 1 bits/blocs. Le schéma de CFB se simplifie et ressemble quelque peu à celui de CBC.

- ✓ **OFB** *Output Feedback*. Est une variante de mode CFB. La clé est modifiée à chaque itération et combinée avec la clé suivante.

- ✓ **CTR** *CounTeR*. La valeur IV est convenue à l'avance, et on fabrique le flot $E_k(IV)$, $E_k(IV + 1)$, $E_k(IV + 2)$, …qui sert à chiffrer le message par ou exclusif.

L'explication des modes ECB, CFB, OFB, CTR en schémas se trouve dans Annexe A.

b. Exemples d'algorithmes de chiffrement symétriques

A ce niveau, on va présenter les plus fameux des algorithmes de chiffrement de ce mode.

b.1 Masque jetable (one-time pad)

Cet algorithme, inventé en 1917 et connu aussi sous le nom de **chiffre de Vernam**, est un algorithme de chiffrement prouvé inconditionnellement sûr. Dans ce système, la clé possède la même taille que le texte à chiffrer et est appelée *masque jetable*. « Masque », car cette clé est combinée par *ou exclusif* avec le texte en clair pour obtenir le texte chiffré; « jetable », car une clé ne doit servir qu'une seule fois [25].

La sécurité de ce système repose sur la génération complètement aléatoire de la clé, ce qui représente le grand avantage de ce système. Par conséquence, si le cryptanalyste ne possède aucune information sur laquelle son attaque va appuyer, tous les masques seront

équiprobables. En effet, si M est le message à chiffrer, C le message chiffré correspondant et K le masque jetable, nous avons : $C = M \oplus K$

Malgré que impossible de savoir quel est le bon texte en clair sans connaître la clé, ce système est limité à des applications extrêmes et ne peut être utilisé pour chiffrer des flux importants de données à cause de la taille de la clé nécessitant des générateurs aléatoires pour sa création.

b.2 DES

Le 15 mai 1973, le *National Bureau of Standards* des Etats-Unis lança un appel d'offre de système cryptographique dans le *Federal Register*, qui est un journal officiel américain [14]. Cet appel déboucha sur le **Standard de chiffrement de données DES** qui est devenu le système cryptographique le plus utilisé dans le monde. IBM développa initialement DES comme modification d'un système antérieur appelé **LUCIFER**. DES, ou encore appelé **DEA** (**Data Encryption Algorithm**), fut publié dans le *Federal Register* le 17 mars 1975. En 15 janvier 1977, le DES est adopté comme standard pour des applications non classifiées. Depuis son adoption, DES a été réévalué par le National Bureau of Standards tous les cinq ans, approximativement. La plus récente révision date de janvier 1994 où il a été renouvelé jusqu'en 1998. Il est prévu que le standard s'arrête à cette date.

b.2.1 Description

Une description complète de DES est donnée dans le *Federal Information Processing Standard Publication* (FIPS) № 46 du 15 janvier 1977 [14]. C'est le cryptosystème qui a été le plus utilisé, de plus, il a bien résisté aux efforts des cryptanalystes pendant 25 ans.

Ce cryptosystème est un système de chiffrement *par blocs*. Il découpe le texte clair en blocs de 64 bits. Ces blocs sont chiffrés séparément, puis concaténés. Ainsi, les données en entrée de cet algorithme seront des blocs de 64 bits du texte clair, et les données en sortie seront aussi des blocs de 64 bits. C'est seulement la courte longueur de la clé, utilisée lors du chiffrement qui est de 56 bits, qui ne lui permet pas, aujourd'hui, d'assurer un bon niveau de sécurité, malgré qu'elle a été largement suffisante au moment de sa conception.

L'algorithme est simple puisqu'il combine des permutations et des substitutions. On donne tout d'abord une description générale de ce système qui se déroule en trois étapes :

1) Etant donner un bloc de texte clair x. Une chaîne de bits x_0 est construite en changeant l'ordre des bits de x suivant une *permutation initiale* IP fixée (figure 1.11) [14]. On écrit :

$$x_0 = \mathrm{IP}(x) = \mathrm{L}_0\mathrm{R}_0$$

Où L_0 contient les 32 premiers bits de la chaîne x_0 et R_0 contient les 32 bits restants.

2) 16 itérations (tours) d'une certaine fonction sont effectuées. Chaque tour suit le même schéma qui consiste à prendre en entrée 32 bits : L_{i-1} et R_{i-1} du tour précédent et de produire de nouveau 32 bits L_i et R_i de la manière suivante [26] :

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

D'après la formule qui corresponde au calcul de R_i, on constate que la fonction *f* utilise deux arguments ayant des tailles différentes : R_{i-1} de 32 bits et k_i de 48 bits obtenue par diversification à partir de la clé k de 56 bits. Les opérations de la fonction *f* sont schématisées à travers la figure 1.9 [27].

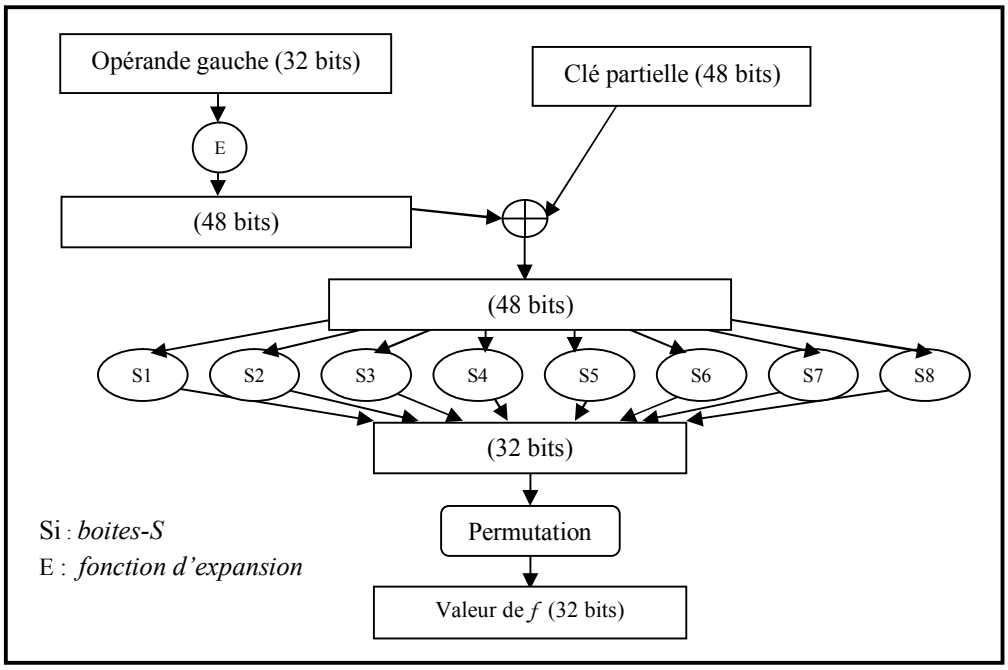


Figure 1.9. *Schéma de la fonction f*

3) Après le dernier tour, les moitiés gauche et droite (L_{16} et R_{16}) sont échangées puis le texte sera permuté bit à bit par IP^{-1} pour obtenir le bloc de texte chiffré y. Plus formellement, s'y obtient comme suit [14] :

$$y = IP^{-1}(R_{16}L_{16}).$$

Cette description peut être résumée par le schéma général illustré par la figure suivante, où on a seulement représenté quelques-unes des 16 étapes.

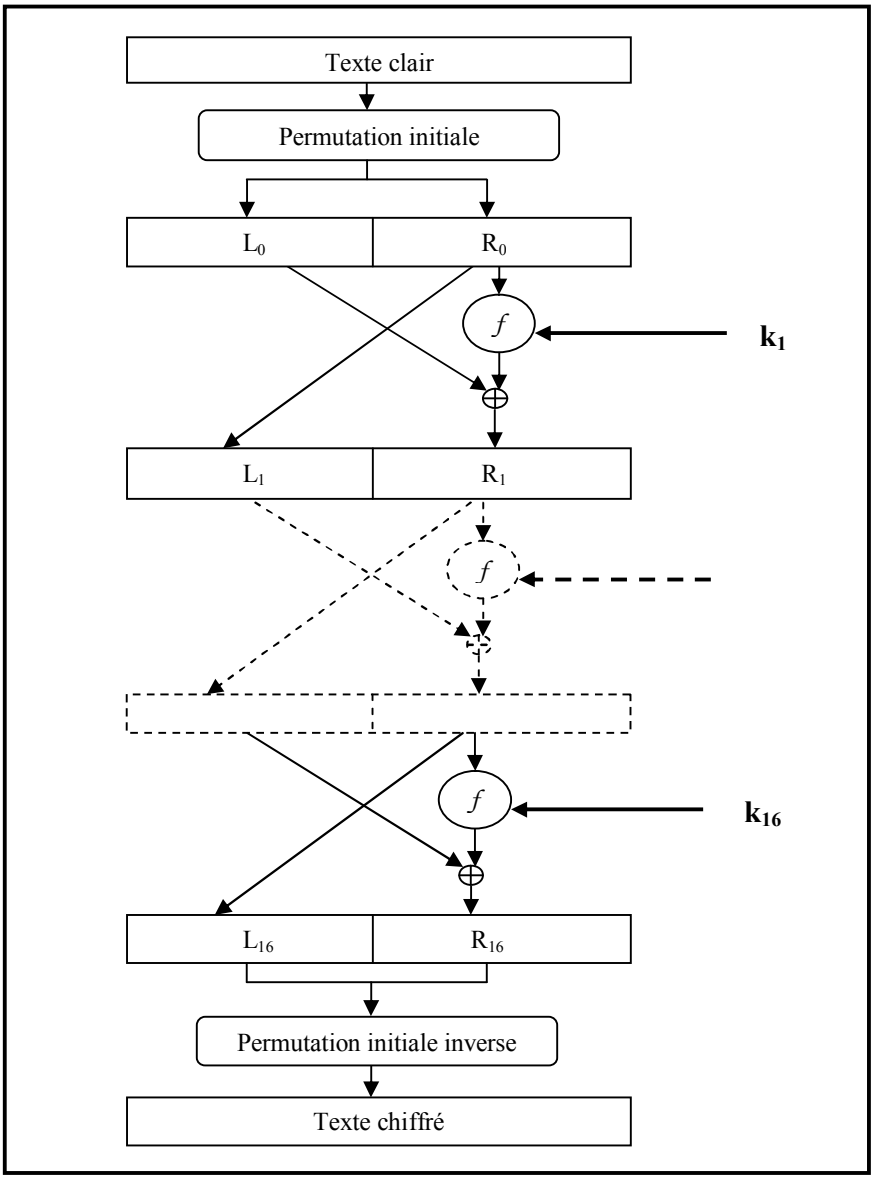


Figure 1.10. *Schéma général de DES*

La figure 1.11 résume la permutation initiale et la permutation initiale inverse, qui se lisent de gauche à droite et de haut en bas.

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

Permutation initiale

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

Permutation initiale inverse

Figure 1.11. *La permutation initiale et son inverse*

Remarque :
Le déchiffrement suit le même algorithme avec la même clé *K*. Seules les sous-clés sont appliquées dans le sens inverse.

b.2.2 Cryptanalyse de DES

En janvier 1998 [21], une attaque dite *exhaustive* a été réalisée contre le DES utilisant une clé secrète de 56 bits, en 39 jours sur 10 000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée comportant 1500 composants DES.

En 1990, *Eli Biham* et *Adi Shamir*, introduisent la méthode de ***cryptanalyse différentielle*** [28]. C'est grâce à cette méthode qu'ils ont pu trouver une attaque à texte clair efficace contre le DES. Cette attaque cherche des paires de texte en clair et des paires de texte chiffré, puis elle les analyse en comparant les différences notables entre ces deux paires. Mais le DES complet à 16 tours est resté hors de portée de cette attaque.

La ***cryptanalyse linéaire*** qui a été introduite par *H.Gilbert* et *M. Matsui* dans le cas du DES [27] est une attaque à messages clairs connus, qui utilise de légers défauts statistiques des étages de substitutions, correspondant aux boites-S dans le cas de DES. Le DES complet n'est pas menacé par cette attaque.

b.3 Triple DES (3DES)

Pour palier à l'insuffisance cryptographique observée du cryptosystème DES, due à la faible longueur de sa clé, il a été indispensable de chercher une solution rapide à cette situation. La première idée qui vienne à l'esprit est de combiner plusieurs chiffrements DES pour obtenir un système ayant une clé plus longue.
En 1978, le **triple DES** (**3DES**) a été conçu par *Whitfield Diffie*, *Martin Hellman* et *Walt Tuchmann*. Il consiste à composer deux chiffrements DES de même clé séparée par un déchiffrement DES avec une autre clé. Donc, ce principe peut être formulé comme suit [27]:

$$\textbf{Triple-DES}_{k1,k2} = \textbf{DES}_{k1} \ \textbf{O} \ \textbf{DES}^{-1}_{k2} \ \textbf{O} \ \textbf{DES}_{k1}$$

Le déchiffrement de son tour est formulé par [27]:

$$\textbf{Triple-DES}^{-1}_{k1,k2} = \textbf{DES}^{-1}_{k1} \ \textbf{O} \ \textbf{DES}_{k2} \ \textbf{O} \ \textbf{DES}^{-1}_{k1}$$

En pratique, on n'utilise que 2 clés différentes (que l'on alterne) car l'utilisation d'une troisième clé ne rajoute aucune sécurité. Cette méthode de chiffrement reste hors portée de l'attaque exhaustive vu la taille de la clé 3DES qui est composée de deux clés DES et donc

composée de 112 bits. Mais l'attaque la plus courante contre le triple DES est « *par le milieu* » (*par dictionnaires*) qui consiste à créer des dictionnaires multiples de façons à scinder le schéma en 2 parties et diminuer ainsi d'autant le nombre de possibilités à tester.

b.4 Blowfish

Blowfish a été conçu par *Bruce Schneier* en 1993 comme étant une alternative aux algorithmes existants, en étant rapide et gratuit [29]. Ce cryptosystème est sensiblement plus rapide que le DES. La grandeur de ses blocs est de 64 bits et il peut prendre une longueur de clé variant entre 32 bits et 448 bits. Ainsi, et depuis sa conception, il a été grandement analysé et est aujourd'hui considéré comme étant un algorithme de chiffrement robuste, mais il n'est pas breveté. Ainsi, son utilisation est libre et gratuite.

b.5 AES

Robert S. Litt (Principal Associate Deputy Attorney General), a assuré le 17 mars 1998, que le FBI n'avait aucune possibilité technologique et financière de décoder un message codé avec un algorithme symétrique dont la clé secrète a une longueur égale à 56 bits [26]. Ce qui fait que, le NIST (National Institute of Standards and Technology) a demandé à la communauté cryptographique de réfléchir au successeur : **AES**.

AES est le sigle d'**Advanced Encryption Standard** (en français, standard de chiffrement avancé). C'est l'algorithme **Rijndael**, du nom de leurs concepteurs Belges *Joan Daemen* et *Vincent Rijmen* [30]. Il a été retenu par le NIST en octobre 2000 pour être l'algorithme AES, et ce, principalement pour des raisons de sécurité, performance, efficacité, facilité d'implémentation et flexibilité. De plus, son utilisation est très pratique car il consomme peu de mémoire. On va voir en détaille cet algorithme dans le chapitre 3.

b.6 Serpent

Serpent, inventé par *Ross Anderson*, *Eli Biham* et *Lars Knudsen*, est un cryptosystème symétrique chiffrant des blocs de 128 bits. Il a été développé en vu d'être un Advanced Encryption Standard. Et malgré que le choix du NIST pour AES s'est porté sur Rijndael, mais ça n'empêche de signaler que Serpent et Rijndael sont similaires, et que la principale différence entre eux, est que Rijndael est plus rapide mais Serpent est plus sûr.

b.7 Twofish

Twofish est un algorithme de chiffrement symétrique par bloc inventé et analysé par *Bruce Schneier*, *Niels Ferguson*, *John Kelsey*, *Doug Whiting*, *David Wagner* et *Chris Hall* au

sein du Counterpane Labs, pour participer au concours AES, où, il a été l'un des cinq finalistes du concours. Ce cryptosystème est conçu pour être très sûr et très flexible, en chiffrant des blocs de 128 bits avec une clé de 128, 192 ou 256 bits, et en reprenant quelques concepts présents dans le Blowfish du même auteur. Cependant, Twofish est légèrement plus lent que Rijndael mais plus rapide que les autres finalistes de l'AES.

En 2005, Counterpane Labs a passé un long temps en évaluant Twofish, qui semble être plus sûre que la version initialement annoncée durant le concours AES. Malgré ça, il reste relativement peu utilisé.

b.8 MARS

MARS est un algorithme de chiffrement symétrique par blocs créé par IBM comme algorithme pour le standard AES. *Don Coppersmith* était l'un des concepteurs de cet algorithme, qui prend en charge des blocs de 128 bits et des clés de dimensions variables entre 128 et 448 bits par incréments de 32 bits. Cet algorithme est unique, car il associe toutes les techniques de cryptage connues dans un seul produit. Ainsi, il utilise deux algorithmes séparés, de façon que si une partie de MARS est cassée, le reste des chiffres restera sécurisé et les données seront sauvegardées. De plus, MARS offre une meilleure sécurité que le triple DES et il est plus rapide que le DES.

b.9 RC6

RC6 est un algorithme de chiffrement par bloc publié en 1998, et conçu au sein de la société RSA Security par *Ron Rivest, Matt Robshaw, Ray Sidney* et *Yiqun Lisa Yin* dans le cadre du concours AES. Il est basé sur un bloc de 128 bits et supporte des clés de 128, 192 et 256 bits.

1.3.3.2.2 La cryptographie asymétrique

a. Historique

Le concept de **cryptographie à clé publique** (**cryptographie asymétrique)**, a été présenté pour la première fois par *Whitfield Diffie* et *Martin Hellman* dans leur article écrit pour le *National Computer Conference* en 1976, puis publié quelques mois plus tard dans *New Directions in Cryptography* sans pouvoir donner un exemple d'un système à clé publique. Il fallut attendre 1978 où la version académique du premier cryptosystème à clé publique a fait l'objet d'un article intitulé : « A Method for Obtaining Digital Signatures and Public-key Cryptosystems » écrit par *Ronald Rivest, Adi Shamir,* et *Leonard Adleman*.

En réalité, *James Ellis*, qui travaillait au service du chiffre britannique (GCHQ, *Government Communications Headquarters*), avait eu cette idée un peu avant [31].

b. Principe

La cryptographie à clé publique évite le partage d'un secret entre les deux interlocuteurs puisque, chaque utilisateur dispose d'un couple de clés : une clé publique qu'il met en général à disposition de tous dans un annuaire, et une clé secrète connue de lui seul. Ces deux clés, en plus d'être distinctes, elles ne peuvent se déduire l'une de l'autre. Alors, pour envoyer un message confidentiel à Bob, Alice chiffre le message clair par la clé publique de Bob. Ce dernier, déchiffrer le message reçu par sa clé secrète correspondante.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de *fonction à sens unique avec trappe* [83]. Sachons qu'une fonction est appelée à *sens unique* si elle est aisément calculée, mais difficile à inverser, ou plus exactement, infaisable en un temps réalisable avec une puissance de calcul raisonnable. Et une telle fonction sera dite *à trappe*, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la *trappe*. Donc, la construction de ce mode de chiffrement à partir d'une telle fonction, sera une chose très simple où la procédure de chiffrement consiste simplement à appliquer la fonction au message clair.

c. Applications

- **Transmission sécurisée de la clé symétrique**

Pour résoudre le problème d'échange de la clé secrète utilisée lors d'un chiffrement symétrique, le chiffrement asymétrique a été envisagé comme solution. Cette dernière consiste à chiffrer la clé secrète en utilisant ce mécanisme assurant, ainsi, un partage sécurisé de cette clé et évitant son interception par une personne tierce non autorisée.

- **Mécanismes d'identification**

Le problème qui se pose avec le mode de chiffrement asymétrique est celui dit d'identification. Lors de la réception puis du déchiffrement d'un message chiffré, on a aucun moyen de vérifier avec certitude son origine. Afin de résoudre ce problème, on utilise des mécanismes d'identification. Ces mécanismes sont fondés sur le chiffrement asymétrique dont nous décrivons ses étapes à travers l'exemple suivant [32] :

Bob souhaite envoyer des données chiffrées à Alice en garantissant la provenance de celles-ci, il utilise le principe d'identification par chiffrement asymétrique :

1) Alice crée une paire de clés asymétriques : clé privée K_{prA}, clé publique K_{puA} ;
2) Alice envoie sa clé publique à Bob pour qu'il puisse l'envoyer des données chiffrées;
3) Bob procèdera à signer numériquement ces informations afin de garantir à Alice que celles-ci proviennent effectivement de lui :
  3.1) Bob doit donc, avant d'envoyer ces données chiffrées, créer une paire de clés asymétriques : clé publique K_{puB}, clé privée K_{prB}
  3.2) Bob envoie sa clé publique K_{puB} à Alice
  3.3) Bob chiffre son message avec sa clé privée K_{prB}, ce qui représente la *signature numérique*, puis chiffre une seconde fois le message précédent avec la clé publique d'Alice K_{puA}. C'est la phase du chiffrement réel du message.
4) Alice reçoit le message chiffré de Bob, le déchiffre avec sa clé privée : K_{prA}. À ce stade le message n'est pas encore lisible car il a été chiffré deux fois de suite.
5) Alice déchiffre une seconde fois le message avec la clé publique de Bob : K_{puB}.

Si le message ainsi déchiffré par Alice sera un message lisible, alors on conclu qu'il provient effectivement de Bob, sinon l'identité de l'expéditeur a été altérée.

d. Exemples d'algorithmes de chiffrement asymétriques

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

d.1 RSA

En cryptographie à clé publique, les trois lettres **RSA** sont certainement les plus célèbres. Ce cryptosystème tire son nom des noms de ses trois inventeurs : *R. Rivest, A. Shamir,* et *L. Adleman* [26]. Ce système, inventé en 1977, est le premier protocole de cryptographie à clé publique.

d.1.1 Description

Ce chiffrement est fondé sur la difficulté de factoriser un nombre qui est le produit de deux grands nombres premiers. Il utilise l'arithmétique de Z_n, qui est un anneau pour tout entier n supérieur à 2, et où n est le produit de deux nombres premiers impairs distincts p et q. Pour un tel n, on a [14] :

$$\emptyset(n) = (p-1)(q-1).$$

La description formelle du système est donnée dans la figure 1.12.

Soit $n = pq$ où : p et q sont premiers, et soit $P = C = Z_n$.

On définit : $K = \{(n, p, q, a, b) : n = pq, p \text{ et } q \text{ premiers} ; ab \equiv 1 \bmod Ø(n)\}$

Pour k = (n, p, q, a, b), on définit :

$$\mathbf{e_k(x) = x^b \bmod n = y}$$

$$\mathbf{d_k(y) = y^a \bmod n = x}$$

(x, y) Є Z_n. Les valeurs n et b sont publiques, et les valeurs p, q et a sont secrètes.

Figure 1.12. *Le chiffrement RSA*

La sécurité de RSA est basée sur l'hypothèse que la fonction $e_k(x) = x^b \bmod n$ est a sens unique, ce qui rend impossible à Oscar de décrypter un texte chiffré.

Pour implémenter le chiffrement RSA, Bob suit les étapes indiquées ci-dessous [14] :

1) Bob engendre deux grands nombres premiers p et q secrets.

2) Bob calcul $n = pq$ et $Ø(n) = (p-1)(q-1)$.

3) Bob choisis un b aléatoire ($1 < b < Ø(n)$), tel que pgcd(b, $Ø(n)$) = 1.

4) Bob calcul $a = b^{-1}$ en utilisant l'algorithme d'Euclide.

5) Bob publie b et n dans un répertoire.

Une attaque évidente à ce système consiste à tenter de factoriser n, alors que, l'intérêt du système RSA repose sur le fait, qu'à l'heure actuelle, il est pratiquement impossible de retrouver dans un temps raisonnable p et q à partir de n si celui-ci est très grand, alors, le bon choix de p et q est un point crucial assurant la bonne sécurisation de ce cryptosystème.

d.2 Chiffrement d'ElGamal

L'algorithme **ElGamal,** créé par *Taher Elgamal*, est un algorithme asymétrique basé sur les logarithmes discrets. Ainsi, et contrairement à RSA, cet algorithme n'a jamais été sous la protection d'un brevet.

d.2.1 Problème du logarithme discret

Le problème du logarithme discret est décrit dans le corps fini Z_p, où p est un nombre premier, sachons que le groupe Z_p^* est cyclique, et que ses générateurs sont appelés racines primitives modulo p [14]. La figure suivante résume le problème du logarithme discret.

Instance du problème : I = (p, α, ß) où p est premier, $\alpha \in Z_p$ est primitif et $ß \in Z_p^*$.

Question : Trouver l'unique a, $0 \leq a \leq p-2$ tel que : $\alpha^a \equiv ß \pmod{p}$

On note cet entier $\log_\alpha ß$

Figure 1.13. *Problème du logarithme discret dans Z_p*

Le problème du logarithme discret dans Z_p, est réputé difficile. Sachons qu'aucun algorithme polynomial n'a été défini pour le résoudre. L'utilité de ce problème en cryptographie provient du fait que calculer des logarithmes discrets est certainement difficile, tandis que calculer l'opération inverse d'exponentiation peut se faire efficacement avec l'algorithme d'exponentiation modulaire [14] qui est une fonction à sens unique.

d.2.2 Description

La figure suivante présente, d'une manière formelle, l'algorithme ElGamal [14].

Soit p un nombre premier tel que le problème du logarithme discret dans Z_p soit difficile, et soit $\alpha \in Z_p^*$ un élément primitif. Soit $P = Z_p^*$. $C = Z_p^* \times Z_p^*$ et $K = \{(p, \alpha, a, ß) : ß \equiv \alpha^a \pmod{p}\}$

Les valeurs p, α et ß sont publiques, et a est secret.

Pour $k = (p, \alpha, a, ß)$, et pour un $k \in Z_{p-1}$ aléatoire (secret) :

La fonction de chiffrement est définit par : $\mathbf{e_k(x, k) = (y_1, y_2)}$

Où $\mathbf{y_1 = \alpha^k \bmod p}$ et $\mathbf{y_2 = x\, ß^k \bmod p}$

Pour $y_1, y_2 \in Z_p^*$, la fonction de déchiffrement est définit par : $\mathbf{d_k(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p}$

Figure 1.14. *Chiffrement d'ElGamal*

- **Comparaison entre les cryptosystèmes symétriques et asymétriques :**

Le tableau ci-dessous présente une comparaison entre les systèmes de chiffrement symétriques et asymétriques [83] :

Méthode	Exemples	Avantages	Inconvénients
À clés Secrètes	DES, AES	▪ Rapidité de calcul en général (dépend de la taille de la clé). ▪ Utilise des clés de tailles plus petites qu'un chiffrement asymétrique ▪ Il est base des fonctions mathématiques simples	▪ Moins sécurisé (DES). ▪ Problème de communication de clés entre émetteur et récepteur. ▪ Une clé pour chacun des correspondants : n personnes => n(n-1)/2 clés.
À clés Publiques	RSA, ElGamal	▪ Très sécurisée à cause de l'utilisation de deux clés distinctes. ▪ Permet la signature électronique. ▪ Un couple de clés publique/privée suffisant pour 'n' correspondants. ▪ La distribution des clés publiques est très simple à gérer	▪ Lente ▪ Utilise des clés de grandes tailles ▪ Nécessite un temps de calcul plus long à cause de la complexité des opérations à effectuer

Tableau 1.1. *Comparaison entre les méthodes de chiffrement symétriques et asymétriques*

1.3.3.2.3 La cryptographie hybride

La cryptographie hybride consiste, comme son nom l'indique, en une association des deux techniques de chiffrement précédentes où on code tout d'abord les données avec une clé privée dite *clé de session*, ensuite cette clé est chiffrée à l'aide d'une clé publique classique. Dans cette politique de cryptage le choix de chiffrer la clé d'une manière publique au lieu de chiffrer les messages est du au fait que, la clé est souvent de petite taille par rapport aux données à chiffrer, donc, elle consomme beaucoup moins de temps lors de son chiffrement par rapport aux données. Ensuite, il ne reste qu'à transmettre le package contenant les données cryptées avec une clé privée, chiffrée de son tour avec une clé publique. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques.

a. Exemples d'algorithmes de chiffrement hybrides

a.1 PGP

Philip Zimmermann, qui est un mathématicien, a commencé à travailler en 1984 sur un système cryptographique aussi sûr mais plus souple que le RSA. Ainsi, il a développé le **PGP** (**Pretty Good Privacy**) en 1991, puis il l'a mis à disposition gratuitement sur Internet sans se préoccuper des détails juridiques qui concernent son utilisation de RSA sans l'accord de son

propriétaire, ou de son vendeur, ViaCrypt [28]. Après trois ans de menaces judiciaires par le gouvernement américain, PGP est à nouveau accessible depuis 1993.

Son principe est comme suit : lorsqu'un utilisateur chiffre un texte avec le système de chiffrement hybride PGP combinant des fonctionnalités de deux modes de la cryptographie, les données sont d'abord compressées. Cela à pour objectif de réduire le temps de transmission de ces données, et d'économiser l'espace disque et, surtout, le renforcement de la sécurité cryptographique du moment où, les cryptanalystes exploitent les modèles trouvés dans le texte en clair pour casser le chiffrement.

Ensuite, l'opération de chiffrement se fait principalement en deux étapes, comme suit [34] :

- ✓ PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé ;
- ✓ Il chiffre cette clé secrète au moyen de la clé RSA publique du destinataire et la transmet.

L'opération de déchiffrement se fait également en deux étapes, qui sont [34] :

- ✓ PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
- ✓ PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

a.2 GPG

GPG (**Gnu Privacy Guard**) est dans le principe un clone de PGP, ou plus exactement une implémentation de l'OpenPGP, mais n'utilise aucun code de PGP [33]. Donc, c'est l'équivalent libre de PGP. Il est entièrement écrit par des développeurs bénévoles, et est complètement libre, sous licence GPL (General Public License).

1.3.3.3 La cryptographie quantique

L'**Informatique quantique** est née de la rencontre de physiciens et de théoriciens de l'informatique, commence à jeter les bases d'un nouvel espace technique [35]. L'idée d'utiliser des lois de la mécanique quantique dans le domaine de la sécurité, remonte au début des années 1970 lorsque *Stephen Wiesner* de l'université de Columbia, a écrit un rapport présentant des idées tout à fait nouvelles [36]. Il s'agit en fait d'échange quantique de clés.

1.4 Conclusion

Dans ce chapitre un état de l'art aussi riche que possible sur la cryptographie, depuis sa première apparition jusqu'à nos jours, a été présenté. D'après cette étude, un système cryptographique est considéré comme sûr si personne n'a encore mis en défaut sa sécurité.

CHAPITRE 2

PDA et Téléphone mobile

CHAPITRE 2

PDA et Téléphone mobile

2.1 Introduction

Depuis ses inventions, les téléphones portables et les ordinateurs de poche (PDA) n'ont pas cessé d'évoluer et de développer dans le monde, de sorte qu'ils sont devenus des outils indispensables aux personnels mobiles. Ils sont désormais courantes dans de nombreux endroits du monde, et devrait progressivement remplacer le téléphone classique.

Dans ce chapitre, nous commencerons par la présentation du dispositif PDA. En suite, nous aborderons un autre terminal qui est le téléphone mobile, où nous détaillerons pour chacun d'eux leur évolution, structure, généralités ainsi que leurs fonctionnements.

2.2 PDA

2.2.1 Définition

Un PDA (Initiales du terme anglais « ***P**ersonal **D**igital **A**ssistant* », littéralement « Assistant Numérique personnel ») est le terme générique employé pour définir un micro-ordinateur ou ordinateur de poche tenant dans la main et offrant la possibilité de réaliser des calculs, de stocker et de récupérer de l'information pour un usage personnel ou professionnel. L'Assistant numérique personnel (PDA) aussi appelés « palmtop », « handheld » [48]. Le terme « handheld » est utilisé pour désigner les ordinateurs de poche dotés d'un clavier.

2.2.2 La naissance des *assistants personnels*

Les constructeurs d'ordinateurs se liant aux concepteurs de plate-forme afin de pouvoir proposer des fonctionnalités à leurs produits.

2.2.2.1 Le Newton de Apple

Le terme de PDA (Personal Digital Assistant), voit le jour pour la première fois au début des années 1990, lorsque Apple commence à rechercher des associés afin de créer le premier ordinateur de poche [38].

Le 2 août 1993 est commercialisé le Newton. Déjà équipé d'un système de reconnaissance d'écriture, et de quelques logiciels tels que la prise de notes et l'agenda, il ne connaît pas un franc succès du fait de sa lenteur et de ses difficultés à reconnaître l'écriture. Apple s'arrête de fabriquer les PDA en 1998, même de développer son système d'exploitation, le NewtonOS.

Peu de temps après la sortie du Newton de Apple, arrive sur le marché le Zoomer en octobre 1993, né de la coopération de plusieurs sociétés telles que Palm, Tandy, Casio, Geoworks, America On Line et Intuit. Mais seulement 10 000 unités sont vendues en 4 mois lors de sa sortie, marquant la fin de son existence.

2.2.2.2 Microsoft

A cette même époque Microsoft propose pour l'informatique mobile deux concepts qui sont le WinPad (qui est une sorte de « super » calculatrice avec des fonctions rajoutées) et Pulsar (qui est un petit ordinateur de poche simple, couplé à une radio-messagerie) ayant comme système d'exploitation le *At Work O.S* [38] qui n'est rien d'autre qu'une version très allégée de Windows 3.1 de Microsoft, mais ces deux ordinateurs mobiles sont abandonnés. Pour cela, MICROSOFT décide de ne plus produire de matériels informatiques mobiles et de recentrer son activité sur la conception des systèmes d'exploitation pouvant fonctionner sur les PDA de différents fabricants. En 1996, le système d'exploitation *At Work OS* est repris et réétudié dans le projet « Pegasus » qui donnera Windows CE 1.0, qui était le système d'exploitation de 1er Handheld PC 1.0 sous la marque CASIO [38].

Au fil des années 90, Microsoft diffusera Windows CE 2.0 avant d'arriver au système d'exploitation actuel : Windows CE 3.0.

2.2.2.3 Palm

En 1994, l'américain Jeff Hawkins crée le concept du PDA Palm [38]. Ce concept se définit ainsi : taille et poids réduits, aspect attractif, alimentation via deux piles au format AAA, doté de quatre applications qui sont l'agenda, et trois gestionnaires : contact, notes, dépenses. L'entreprise 3COM, filiale de US Robitics, est chargée de la commercialisation de ce PDA. Tout de suite le PDA PALM plait beaucoup au grand public. Le premier modèle est le PalmPilot m1000 Handheld [38]. En 1998, US Robotics se sépare de sa filiale 3C0M. Jeff Hawkins quitte alors la filiale et crée une nouvelle entreprise appelée Handspring. Il reprend la base d'un Palm et y rajoute des fonctions multimédias telles que le lecteur de musique au format compressé type MP3. Le PDA Visor de Handspring [38] est né. Pendant ce temps, 3COM devient Palm Computing et vend le système d'exploitation Palm OS sous licence à

IBM qui fabrique le WorkPad. Dans le même temps 3COM développe ses propres produits et voit arriver dans l'univers Palm les écrans couleurs avec sa série III (Palm IIIc).

2.2.2.4 Epoc et Symbian OS

L'entreprise PSION nait en 1980, met sur le marché au milieu des années 80 des agendas électroniques avec une fonction carnet d'adresses (Psion Organizer I et II).

Septembre 1991 voit la sortie de la Série 3 de PSION (qui est un ordinateur de poche qui comporte un clavier et non pas un système de reconnaissance d'écriture, contrairement à APPLE) avec Epoc comme système d'exploitation qui rend PSION la 3ème société à se disputer le marché des ordinateurs de poches avec Microsoft (Windows CE) et Palm (Palm O.S.) dans le milieu et la fin des années 90 [38].

C'est en 1994 que l'entreprise PSION commence à faire évoluer son système d'exploitation de 16 à 32 bits. Cette évolution aboutit à la «série 5» en 1997[38]. Ce dispositif propose un écran tactile tout en conservant son clavier. A cette époque, cette entreprise développe à la fois le système d'exploitation de ses PDA (Epoc) et le matériel informatique.

En 1998 est créé la Symbian O.S. (système d'exploitation construit sur la base de Epoc 32 bits associé à la technologie de la téléphonie sans fil) par un groupe d'entreprises des Télécommunications (Ericson, Nokia, Motorola, Siemens, Panasonic, Samsung, Nec, Lg, Mitsubichi, Kyocara, Alcatel) [38].

2.2.3 Types de PDA

2.2.3.1 PDA Palm

La plupart des dispositifs Palm sont faits par palmOne, qui offre les gammes de produits de Zire et Tungsten [48].

Les PDA Palm OS sont facile à utiliser, et ils ont :

- Une vaste bibliothèque d'applications pouvant ajouter au système.
- Une version mise à jour de l'application de reconnaissance d'écriture de Graffiti
- La synchronisation avec les ordinateurs de bureau Palm.
- Des petits écrans que le Pocket PC pour accommoder une zone de Graffiti.

2.2.3.2 Pocket PC

Pocket PC est le nom générique pour les PDA Windows Mobile. Ses caractéristiques standard comprennent [48] :

- Les versions Pocket des applications de Microsoft telles que Word, Excel et Outlook.
- La synchronisation avec Microsoft Outlook sur un PC Windows.
- Trois applications de reconnaissance d'écriture: Transcriber, Reconnaissance de lettre (similaire à la nouvelle version de Graffiti) et Reconnaissance de Bloc (similaire au Graffiti original)
- Une zone d'écriture virtuelle, ce qui maximise la taille de l'affichage.
- Windows Media Player pour le contenu multimédia.

2.2.3.3 Smartphone

Un Smartphone est un téléphone mobile avec des capacités de PDA ou un PDA traditionnel avec ajout de capacités de téléphone mobile, selon le facteur de forme et de fabricant. Les caractéristiques de ces dispositifs comprennent [48] :

- Un fournisseur de services cellulaires pour gérer le service téléphonique.
- L'accès à Internet par le biais de réseaux de données cellulaires.
- Diverses combinaisons des caractéristiques de téléphone portable et PDA, dépend de dispositif (par exemple, le Smartphone offre des possibilités de reconnaissance d'écriture).
- Un certain nombre de différents systèmes d'exploitation, y compris Windows Mobile Pocket PC Phone Edition, Palm OS, Blackberry OS pour les Smartphone BlackBerry**,** et Symbian OS pour Smartphone de Panasonic, Nokia, Samsung et bien d'autres.

2.2.4 Généralité sur les PDA

2.2.4.1 Format et accessoires

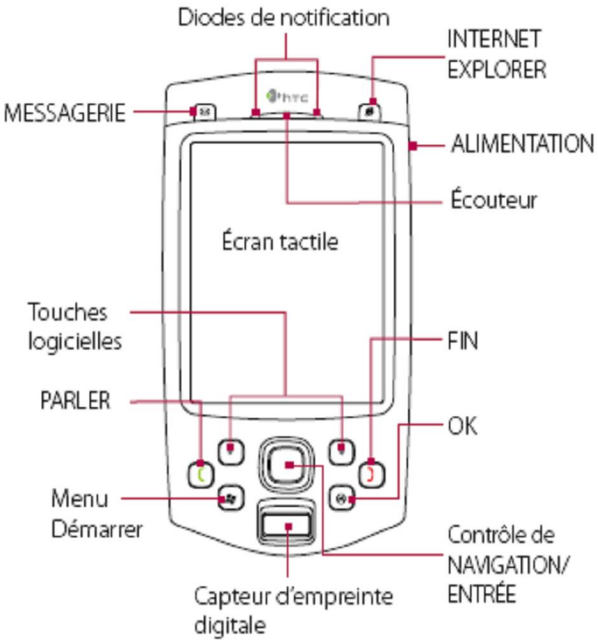


Figure 2.1. *Architecture externe du PDA* [39]

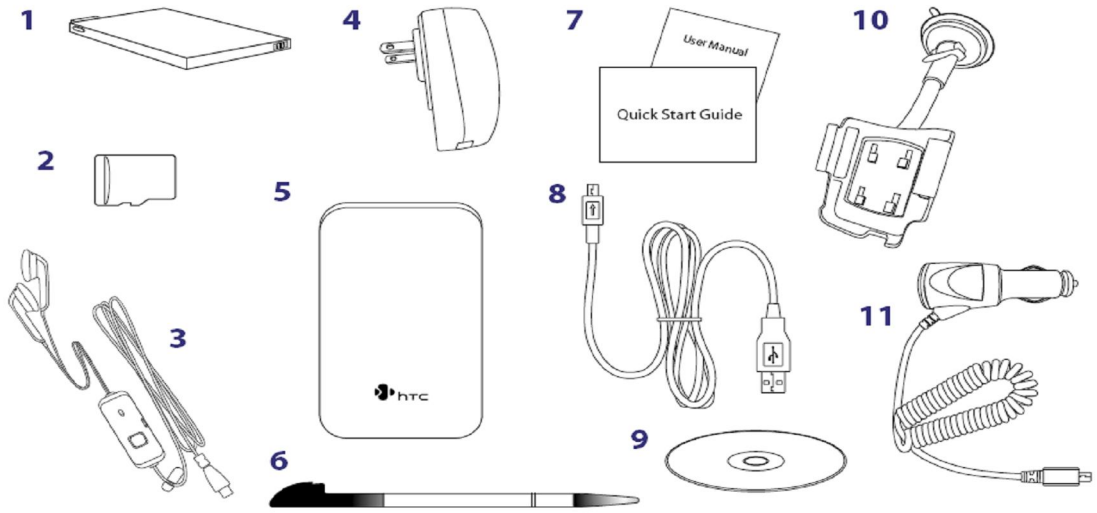


Figure 2.2. *Les accessoires du PDA* [39]

N°	Accessoire	Fonction
1	Batterie	Alimente l'appareil
2	Carte microSD	/
3	Casque stéréo	Permet d'écouter la musique ou d'effectuer/répondre à un appel en mains libres
4	Adaptateur secteur	Recharge la batterie
5	Étui	Sert de sac de transport et de protection pour le PDA
6	Stylet supplémentaire	Utilisé pour appuyer sur les éléments à l'écran.
7	Manuel de l'utilisateur et Guide de mise en route rapide	Références pour utiliser le PDA
8	Câble de synchronisation USB	Connecte le PDA à un PC et synchronise les données
9	Disque de mise en route	Source pour des outils et programmes supplémentaires
10	Support de voiture	Pour monter le PDA dans une voiture
11	Adaptateur de voiture	Pour charger la batterie de PDA lors de l'utilisation en voiture

2.2.4.2 Les matériels

Dans les ordinateurs de poche, il existe une multitude de composants amovibles, utiles pour la vie professionnelle et privée. Nous allons passer en revue les principaux composants indispensables et optionnels des PDA.

2.2.4.2.1 Taille et poids

La taille et le poids sont la raison du développement de ces dispositifs. Le PDA est destiné à être emporté partout et doit donc tenir dans la main ou la poche.

2.2.4.2.2 Clavier

Pour minimiser l'encombrement dans le PDA, le clavier est alors remplacé par un système d'écriture directe sur l'écran tactile en combinaison avec un programme de reconnaissance d'écriture ou au travers d'un clavier qui s'affiche à l'écran. Donc Les PDA ne sont pas tous équipés de clavier, et qui ont un clavier type ordinateur de bureau, sont appelés « Handheld ».

2.2.4.2.3 Ecran

Les écrans sont de tailles différentes suivant le type de PDA. Pour les « handheld », l'écran est disposé en longueur.

Presque tous les PDA offrent maintenant des écrans couleur. Le nombre de couleurs actuelles varie de 256 à 65536 couleurs. Pour la résolution de l'écran, elle dépend du système d'exploitation. Pour Palm OS, la résolution est de 160 pixels sur 160 pixels alors que sous Microsoft Pocket PC 2002, elle est de 320 sur 240 pixels [38].

PDA utilise un écran LCD (écran à cristaux liquides) tactile. À la différence des écrans LCD pour les ordinateurs de bureau ou ordinateurs portables, qui sont utilisés uniquement comme dispositifs de sortie, PDA utilise leur écran pour l'entrée et la sortie [48]. Les écrans LCD peuvent être de deux types: à matrices actives ou passives. Les matrices actives, les plus performantes, emploient le TFT (Thin-Film Translator). Les matrices passives, comme DSTN (Double Layer SuperTwist Nematic) ou CSTN (Color SuperTwist Nematic) [38] proposent cependant une qualité très proche des matrices actives à un prix moindre.

2.2.4.2.4 Batterie

Les PDA sont alimentés soit par des piles ou dans la majorité des cas actuellement par des batteries rechargeables (lithium, nickel) [50]. L'autonomie varie sur ce type de dispositif de 7 à 15 heures, tous dépend du type de PDA [38].

La plupart des PDA ont le système de gestion de puissance pour prolonger l'autonomie de la batterie. Même si les batteries sont si basses, il ya généralement assez d'énergie pour maintenir la mémoire rafraîchie. La majorité des dispositifs ont une batterie de secours interne qui fournit la puissance à court terme (30 minutes ou moins) jusqu'à ce qu'on installe un remplacement. Si toutes les sources d'énergie sont épuisées, les PDA perdent toutes les données dans la RAM. Cela rend la synchronisation de PDA extrêmement important.

2.2.4.2.5 ROM

Un PDA ne dispose pas d'un disque dur. Il stocke les programmes de base (carnet d'adresses, calendrier, bloc-notes et le système d'exploitation) dans une mémoire en lecture seule (ROM). Cette mémoire ne consomme pas d'énergie, donc elle est insensible à toute coupure d'électricité ou au déchargement de la batterie. La taille de cette mémoire est très variable, elle varie actuellement de 1 à 64 mégaoctets [38].

2.2.4.2.6 RAM

La RAM ou mémoire vive est une mémoire volatile qui à la particularité d'offrir un accès simultané entre lecture et écriture. Son gros défaut est de perdre toute information en cas d'absence d'alimentation électrique. C'est dans cette mémoire que sont stockées, de manière temporaire, toutes les données que le processeur utilise. La taille de cette RAM varie de 2 à 64 mégaoctets mais peut être augmentée jusqu'à 1 gigaoctet à l'aide d'extension [38].

2.2.4.2.7 Type de processeurs

On distingue deux grandes familles de processeurs utilisés avec les PDA :

Sous PALM OS, le processeur utilisé était le DragonBall de MOTOROLA [38] cadencé jusqu'à 33 mégahertz. Il est fiable et permet une grande autonomie.

Les seconds types sont les processeurs ARM (Acorn RISC Machine) et StrongARM et XScale [38]. Ces processeurs plus rapides cadencent actuellement entre 100 et 400 mégahertz. Ils sont utilisés par les ordinateurs de marque PSION fonctionnant sous Epoc et par les PDA utilisant le système d'exploitation Windows Mobile et même des PDA sous PALM OS.

2.2.4.2.8 Port Infrarouge

Le port infrarouge est un système d'émission de très basses fréquences de lumière, similaire à celui utilisé par les télécommandes de télévision. Présent sur la majorité des PDA.

Ce système est un moyen de communication entre différents éléments communicants tels que PDA, modem des téléphones portables, ordinateurs équipé. Le débit théorique du port infrarouge est de 115 000 bits par seconde sur les PDA. Ce débit peut varier en fonction des conditions d'éclairage (soleil par exemple) [38].

2.2.4.2.9 Connecteurs d'extension et les extensions

2.2.4.2.9.1 Les connecteurs d'extension

Les connecteurs les plus communs sont les connecteurs dits Compact Flash et SD/MMD (*Secure Digital Cards/MultiMedia Card*), qui permettent d'ajouter des cartes mémoires.

Il existe deux types de Compact Flash. Le *type I* et le *type II* [38]. Ces connecteurs possèdent 50 broches. Le connecteur de type I peut recevoir seulement les cartes Compact Flash de type I de taille 43mm x 36mm x 3.3mm. Le connecteur de type II, peut à la fois recevoir les cartes Compact Flash de type I et II de taille 43mm x 36mm x 5.5mm.

En ce qui concerne le connecteur SD/MMD, il permet de recevoir des cartes de type SD card et de type MMD card.

Les ordinateurs de bureau contiennent également des connecteurs d'extension simples tels que les ports USB, PCMCIA (*Personal Computer Memory Card International Association*) et série. Certains appareils ne possèdent pas tous les connecteurs d'extension disponibles, d'où la création d'une interfaces matérielles «jaquettes», qui permettent de connecter une extension d'un type particulier sur un appareil qui n'est pas prévu pour recevoir ce type de carte [38].

2.2.4.2.9.2 Les extensions

a) La mémoire

Pour améliorer la capacité de stockage, des cartes mémoires peuvent être rajoutée sur la majorité des PDA. Les cartes utilisées pour cet usage sont les cartes SD card, CompactFlash et MultiMedia Card. Le volume de stockage maximal pouvant atteindre 1 gigaoctet.

b) Le clavier

Il est possible de connecter un clavier externe sur la majorité des PDA. Ces claviers peuvent êtres de petite taille et s'enficher sur le PDA, ou peuvent être de taille normale.

Figure 2.3. *Mini Clavier pour PDA Ipaq de Compaq*

Figure 2.4. *Clavier Pliant Logitech pour PDA Palm*

c) Carte LAN (Local Area Network)

Carte d'extension permettant de connecter un PDA à un réseau local par l'intermédiaire d'onde radio selon la technologie IEEE 802.11b. [38].

d) Carte Modem

Carte d'extension permettant de connecter un PDA au réseau téléphonique commuté (RTC). Le débit est faible et dépend du modem utilisé. Il existe aussi des cartes modems RNIS (Réseau Numérique à Intégration de Service) plus rapide que sur le réseau RTC. Actuellement les modems les plus courants ont un débit de 56 000 bits par seconde [38].

e) La technologie Bluetooth

La technologie Bluetooth est basée sur une connexion radio courte distance (2.45 gigahertz) [38], permettant de relier différents appareils dans un rayon de 10 mètres, avec un débit de 1 mégabits par seconde qui permet la transmission d'images et de vidéos, ainsi que la connexion à un réseau. Contrairement aux liaisons infrarouges, les appareils communicants n'ont pas besoin d'être visibles l'un de l'autre.

Cette technologie peut être intégrée dans les nouveaux appareils ou être rajoutée via des extensions CompactFlash ou SD card et PCMCIA.

Figure 2.5. *Carte Bluetooth au format SD card pour Palm à gauche et au format CompactFlash à droite pour Pocket PC*

f) Carte GSM

Carte d'extension permettant d'utiliser le PDA comme téléphone mobile à condition de se trouver dans une zone de couverture d'un opérateur de téléphonie mobile et de pouvoir se connecter à un réseau distant (par l'utilisation d'un protocole de transfert de donnés appelé GPRS permettant d'avoir un débit théorique actuellement de 160.000 bits par seconde par rapport au 9.600 bits par seconde du réseau GSM) ou recevoir des fax par l'intermédiaire du réseau téléphonique GSM [38].

g) GPS

Le GPS (Global Positioning System) [38] existe sous forme de carte d'extension ou sous forme d'appareil indépendant au PDA qu'il est possible de connecter par un connecteur tel le port série. Un appareil GPS associé à une antenne spécifique permet d'avoir un positionnement précis à quelques mètres près sur la planète, il sert par exemple au guidage routier.

Figure 2.6. *Module GPS sur carte Compact Flash*

h) Caméra numérique et appareil photo numérique

Il existe des extensions d'appareils photos numériques et de caméras qui permettent de saisir des images et des séquences vidéo.

Figure 2.7. *Caméra pour PocketPC sur carte CompactFlach.*

Figure 2.8. *Appareil photo numérique pour PDA Sony sous Palm OS*

I) Lecteur Code Barre

Dans l'industrie, un lecteur de code barre peut être utilisé dans un système de référencement en relation avec une base de données.

k) Connexions à un écran ou à un rétroprojecteur

Il est possible de connecter le PDA à un moniteur, un téléviseur ou à un rétroprojecteur afin de faire des présentations.

2.2.4.3 Les systèmes d'exploitation

Le système d'exploitation est une interface entre le matériel et les logiciels. Les plus connus sont ceux qui existent sur les ordinateurs de bureau : Microsoft Windows 95 – 98 – Millenium – XP – 2000 – NT, Linux, Mac OS (Operating System).... [38].

2.2.4.3.1 Palm OS

Le système d'exploitation Palm OS est un système multitâche (capable d'exécuter plusieurs programmes simultanément). C'est est un système d'exploitation pour appareils mobiles édité par PalmSource, séparé en 2003 de la société Palm (PalmOne depuis). Initialement conçus pour les PDA de Palm, Palm OS a été adopté par beaucoup d'autres fabriquants à l'instar de : Handspring, Sony, IBM, Qualcomm, Symbol, Tapwave, AlphaSmart, Fossil, Garmin, HandEra, Acceca, AlphaSmart, GSPda et Samsung [74].

2.2.4.3.2 Microsoft Windows CE, Microsoft Pocket PC et Microsoft Handheld PC

Le système d'exploitation Windows CE est un système d'exploitation multitâches, conçu par Microsoft et qui dans sa dernière version en janvier 2002 s'appelle Window CE.net.
Il inclut des logiciels de traitements de texte, un tableur, un agenda (ils sont tous compatibles avec les programmes de type Office de Microsoft fonctionnant sur les ordinateurs de bureau) ainsi qu'un navigateur Internet et divers lecteurs de fichiers multimédias.

A partir de Windows CE, les PDA offrent la plateforme « Microsoft Handheld PC » qui était destiné à des machines avec clavier. Ainsi, Windows CE a évolué pour finalement devenir « Microsoft Pocket PC ». Il existe actuellement 3 grandes versions de Pocket PC (2000, 2002, 2003) [77]. Puis, Microsoft Pocket PC est devenu Microsoft Windows Mobile.

2.2.4.3.3 Epoc et Symbian OS

Epoc 32 bits est le système d'exploitation des PDA de marque PSION. Il inclut des logiciels de traitement de texte, tableurs, agenda et possède une fonctionnalité de fax et de consultation de données sur Internet.

Symbian OS est un système d'exploitation construit sur la base de Epoc 32 bits [38], il était créé par un groupe d'entreprises des Télécommunications. Ce système équipe les « Smartphones » et les téléphones mobiles possédant des fonctions multimédias.

2.2.4.3.4 Linux

Le système d'exploitation Linux pour PDA est comme Linux pour les ordinateurs de bureau, un système d'exploitation libre. Il s'appuie sur un noyau Linux, et il se décline sous

différentes formes telles que Pocket Linux, ARM Linux, Microwindows [38], avec autant d'interfaces utilisateurs. Dans un premier temps, ce système d'exploitation a été rajouté sur des PDA fonctionnant sous Windows CE ou Palm OS.

2.2.4.4 Les applications

2.2.4.4.1 Agenda

L'agenda est utilisé pour enregistrer toutes les informations concernant les activités du jour pour organiser au mieux un emploi du temps, avec des mécanismes de rappel visuels ou auditifs. Ces fonctions peuvent être regroupées au sein d'une seule application (Microsoft Outlook 2002 sous Microsoft Pocket Pc 2002 ou Action Names Datebook sous Palm OS) ou au sein de multiples petites applications (comme les programmes de base fournis avec le PDA sous Palm OS (Adresses – Agenda – Tâches)) [38].

2.2.4.4.2 Bloc notes

Le Bloc notes est un petit module de prise de notes pour enregistrer des mémos et des tâches. A la différence des traitements de texte, il n'est pas possible en général de générer une mise en page.

2.2.4.4.3 Traitement de texte – tableur

Le traitement de texte sert à rédiger des documents, de les archivés ou de les imprimés et à en faire la mise en page tout comme avec les ordinateurs de bureau. Les logiciels les plus courants (Pocket Word pour Pocket PC, Psion Word pour Epoc, WordSmith pour Palm OS) [38] sont tous compatibles avec Microsoft Word.

Le tableur est destiné à élaborer des feuilles de calculs. Les logiciels les plus courants (Pocket Excel pour Pocket PC, Psion Sheet pour Epoc, Quicksheet pour Palm OS) [38] sont compatibles avec Microsoft Excel.

2.2.4.4.4 Navigateur et messagerie électronique

Le Navigateur est un logiciel qui permet d'avoir accès à des pages internet et intranet en ligne. Il fonctionne comme Microsoft Internet Explorer existant sur les ordinateurs de bureau.

Le logiciel de messagerie adapté au PDA permet l'envoi et la réception des messages électroniques. Les logiciels les plus utilisé : Pocket Outlook pour Pocket PC, le composant E-mail de Mobile Office pour Epoc, Courrier ou Eudora pour Palm OS [38]. Pour que le PDA puisse naviguer sur internet ainsi d'envoyer et de réceptionner de courriers électroniques, il

faut qu'il soit connecté au réseau Internet ou à un réseau intranet (de l'entreprise par exemple).

2.2.4.4.5 Lecteur de documents

Les lecteurs de documents sont des logiciels dont la fonction est d'éditer des documents à l'affichage sans offrir la possibilité d'y apporter des modifications. On peut citer comme exemple Isilo, Mobipocket…

2.2.4.4.6 Outils multimédias

Des outils multimédias avancés permettant : de lire des musiques (notamment au format mp3) ou des animations Flash, de lire des vidéos (dans les différents formats, y compris le format DivX qui est un format de compression/décompression vidéo permettant d'obtenir des vidéos compressées très peu volumineuses avec une perte de qualité très raisonnable [49]).

2.2.5 Le fonctionnement pratique du PDA

2.2.5.1 Méthodes d'entrée des données

2.2.5.1.1 Par clavier

Le clavier est une partie facultative du PDA, afin de diminuer l'encombrement. Sur les « Handheld PC » il est présent sous forme de clavier mécanique et sur les autres PDA, il peut être présent sous forme amovible ou sous forme de claviers digitaux.

2.2.5.1.2 Par écrire sur l'écran

Le clavier est remplacé par un écran tactile dans la majorité des PDA. C'est par l'intermédiaire d'un stylet que l'on applique des pressions sur les différents éléments affichés à l'écran, entraînant une action.

Du fait de l'augmentation de la puissance des dispositifs, aujourd'hui, la reconnaissance de l'écriture fonctionne de façon fiable et permet la saisie rapide d'informations et en tous lieux. Il existe quatre façons de rentrer du texte dans le PDA via l'écran. Elles ne sont pas toutes disponibles sur tous les ordinateurs de poche [38] :

- Clavier virtuel.
- Reconnaissance de caractères et de chiffres simplifiés (Reconnaissance de blocs).
- Reconnaissance de lettres.
- Reconnaissance d'écriture (Transcriber).

- **Clavier virtuel**

Le Clavier virtuel est un clavier de type AZERTY ou QWERTY qui apparaît à l'écran de PDA, similaire au clavier de l'ordinateur de bureau. Il est possible de rentrer des données par pression sur les touches représentées sur l'écran de PDA.

- **Reconnaissance de caractères et de chiffres simplifiés**

Sur les dispositifs Palm, le logiciel qui reconnaît les lettres est appelée Graffiti, et sur les dispositifs Pocket PC il est appelée écriture par reconnaissance de bloc. Ce système de reconnaissance d'écriture exige que chaque lettre soit enregistrée d'une certaine manière, en utilisant un alphabet spécialisé [48]. Sa contrainte est d'apprendre pendant quelques jours la liste des tracés de lettres.

- **Reconnaissance de lettres**

Ce système de reconnaissance est prévu pour reconnaître l'écriture des lettres, des chiffres, et des signes de ponctuation, qui seront convertis en texte tapé [39].

- **Reconnaissance d'écriture (Transcriber)**

Transcriber est un programme de reconnaissance d'écriture manuscrite. Il détecte le tracé des mots qui sont écrit à l'écran, analyse la construction de ce mot lettre par lettre et recherche la correspondance dans son dictionnaire et le retranscrit à l'écran en lettre d'imprimerie.

2.2.5.2 Importations et exportations des données

L'assistant personnel doit être capable de récupérer des données et de les restituer sur des serveurs, ordinateurs de bureau, ou tous autres supports informatiques.

2.2.5.2.1 Synchronisation via un ordinateur de bureau

La synchronisation consiste à rendre identique des éléments prédéfinis sur le PDA et sur l'ordinateur de bureau. Toutes les informations sont actualisées automatiquement, au plus récent, sur les deux supports lors de ce processus. Il existe plusieurs façons pour mettre ces deux dispositifs en communication : par liaison infra rouge, Bluetooth, port série ou USB.

Le logiciel de synchronisation sur le PDA fonctionne avec le logiciel compagnon installé sur l'ordinateur de bureau. Les dispositifs Microsoft Pocket PC utilisent le logiciel de synchronisation ActiveSync et les dispositifs Palm OS utilisent HotSync [38]. L'utilité de synchronisation est qu'on a toujours une copie de sauvegarde de nos données, qui peut être un sauveteur si le PDA est cassé ou volé.

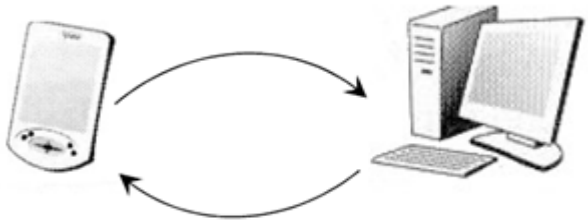


Figure 2.9. *La synchronisation*

2.2.5.2.2 Accès aux réseaux

- **Accès à travers le réseau téléphonique mobile**

La communication via le réseau téléphonique mobile peut se faire selon deux modes [38]:

- Dans le cas du GSM, on doit alors se connecter à chaque fois que l'on a besoin d'une connexion et on utilise le même canal de transmission que la voix. Le PDA va se connecter au réseau de l'entreprise ou le fournisseur d'accès à Internet, via le modem du téléphone mobile.

- Dans le cas du GPRS, le téléphone mobile est raccordé en permanence au réseau. Le mode de transmission des données se fait par paquets avec un débit élevé.

La liaison entre le PDA et le réseau GSM peut s'effectuer de différentes façons [38]:

- Soit directement par le PDA si ce dernier est équipé d'une carte GSM ou GPRS.
- Soit par l'intermédiaire d'un téléphone mobile en mode GSM, GPRS ou UMTS, qu'on le relie avec le PDA par plusieurs façons : avec un cordon de raccordement, par liaison infra rouge, ou par liaison radio « Bluetooth ».

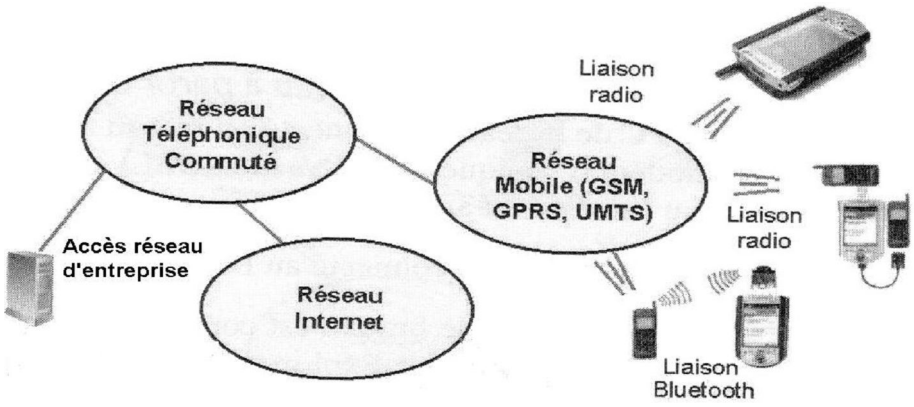


Figure 2.10. *Différentes façons de connecter un PDA à un réseau via le réseau téléphonique*

- **Accès via le réseau téléphonique commuté**

Le réseau téléphonique commuté (RTC) est le réseau de la ligne téléphonique classique. Il permet de se connecter au serveur distant d'une entreprise, de se connecter à Internet pour consulter des pages, ainsi que d'envoyer et consulter des E-mail.

Pour qu'un PDA puisse accéder au réseau téléphonique commuté, il faut lui pouvoir ajouter un modem, ou pourra se faire via l'ordinateur de bureau à partir de la station d'accueil du PDA ou le port infra rouge s'il est disponible.

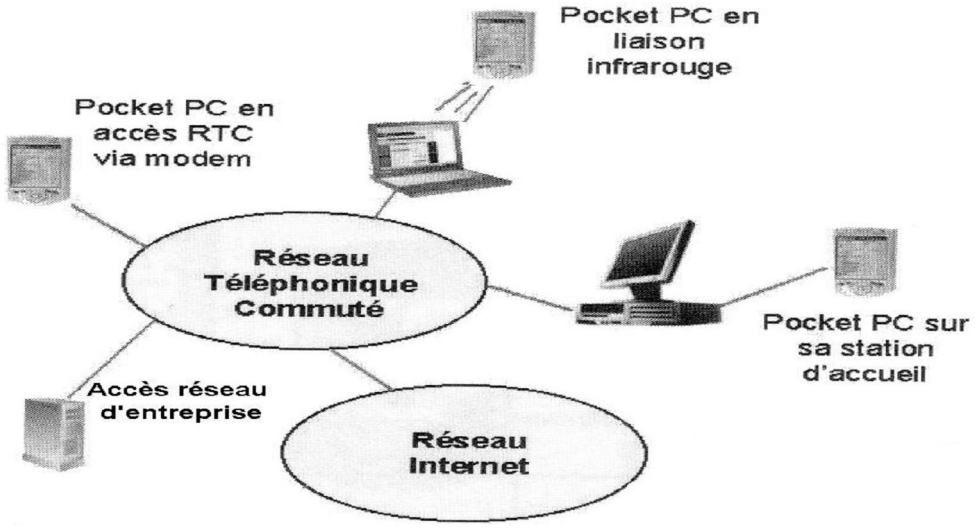


Figure 2.11. *Différentes façons de connecter un PDA à un réseau via le réseau téléphonique*

- **Connexion au travers d'un réseau d'entreprise**

L'accès du PDA au réseau Internet et Intranet peut se faire via un ordinateur de bureau connecté au réseau Ethernet (un protocole de communication de bas niveau permettant à des ordinateurs de communiquer sur un réseau local) de l'entreprise.

L'accès au réseau de l'entreprise peut aussi être sans fil :

- Sois avec une liaison radio « Bluetooth » qui permet de créer un lien entre l'ordinateur de bureau, lui-même relié au réseau d'entreprise et le PDA.
- Sois avec une liaison infra-rouge.
- Sois par une connexion au Wireless Lan (WLan) selon la technologie WiFi (Wireless fidelity), donc le PDA devient un élément sur le réseau d'entreprise comme l'ordinateur de bureau. La liaison physique au réseau se fait grâce à une antenne réceptrice.

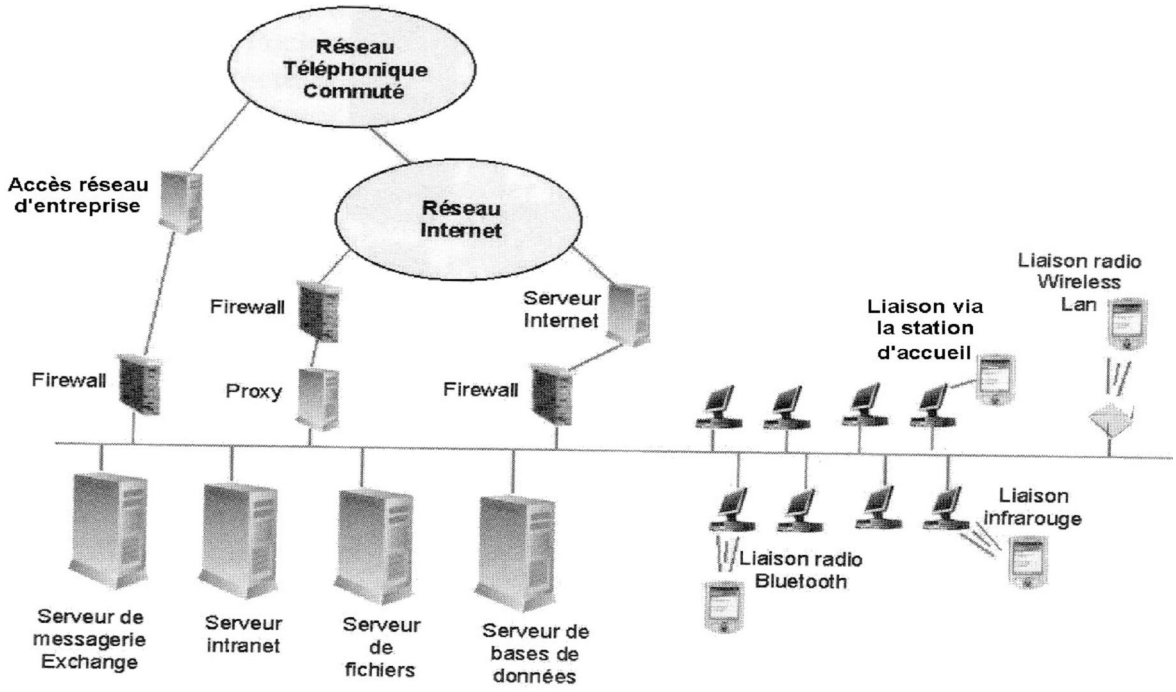


Figure 2.12. *Différentes façons de connecter un PDA à un réseau*

2.2.6 La comparaison entre un PDA, tablette pc et un ordinateur portable

Une tablette PC est une ardoise électronique dispose d'un écran tactile et ne possède pas de clavier. Une comparaison entre un PDA, tablette pc et un ordinateur portable [43] se trouve dans Annexe B.

2.3 Téléphone mobile

2.3.1 Définition

Le téléphone portable, également nommé téléphone mobile ou cellulaire, est un appareil qui repose sur la transmission par ondes électromagnétiques pour transmettre la voix. Il permet de communiquer sans fil sur un réseau de téléphonie mobile.

2.3.2 Histoire et évolution des *téléphones* et *téléphonie mobiles*

Contrairement au téléphone filaire, le téléphone mobile n'est pas une invention mais un objet né de l'association de plusieurs technologies (comme la technologie radio qui a été développée à partir des années 1940), ils utilisent des ondes électromagnétiques, tout comme le réseau pour la radio, la télévision…

Dans les années 20 [41], les systèmes de radiocommunications étaient utilisés par l'armée au Japon et aux États-Unis pour communiquer en temps réel, mais la lourdeur des matériels et leurs utilisation de ces années rendait difficile l'exploitation des télécommunications mobiles. De nombreuses recherches alors mènent d'intégrer le dispositif dans des véhicules pour résoudre le problème occasionnée par : d'une part l'encombrement du dispositif (dont l'antenne mesurait plus d'un mètre) et, d'autre part, les énormes besoins en énergie électrique pour émettre. Le téléphone de voiture était né...

Dès 1947, une invention intervient avec la création des cellules hexagonales pour les téléphones de Bell Labs, qui transmettent et reçoivent des signaux dans trois directions [40].

À la fin de la seconde guerre mondiale apparut pour la première fois un véritable plan de répartition des fréquences permettant son utilisation non archaïque. Ainsi, dans les années 60 [41] se développèrent des techniques permettant une utilisation multiple des ces fréquences.

En 1973, l'ARP (AutoRadioPuhelin : radiotéléphone de voiture), premier réseau de téléphonie mobile, est déployé en Finlande. Il restera opérationnel pendant 27 ans. Au même moment, Le Dr Martin Cooper, le directeur général de Motorola a été inventé le tout premier téléphone portable. Lui même, a été la première personne à faire un appel depuis son téléphone portable, qui a été destinée à son rival Joel Engel, un grand chercheur.

À la fin des années 1970, diverses avancées techniques (la réduction de la taille des batteries) et le développement de la technologie cellulaire, ouvrent l'ère du téléphone mobile dite de première génération (1G). Ces téléphones étaient de taille importante, fonctionnaient

en mode analogique et selon les normes NMT (Nordisk Mobiltelefon-gruppen), AMPS (Advanced Mobile Phone System), TACS (Total Access Communication System)... [51].

Il fallut attendre encore plusieurs années pour que les téléphones soient miniaturisés pour pouvoir être qualifiés de "mobiles". C'est en 1983 que Motorola a lancé aux États-Unis le premier téléphone portable : le Motorola DynaTAC 8000X [40].

Ce n'est qu'au début des années 1990 que les téléphones cellulaires sont devenus assez petits grâce aux progrès de la technologie des batteries et de la puce informatique et ils sont considérés comme étant de seconde génération (2G) où ils fonctionnent en mode numérique. Ils utilisaient notamment la norme GSM [47] (*Global System for Mobile Communications*) établie en 1982. Des extensions de cette norme ont été mises au point afin d'en améliorer le débit. C'est le cas du standard GPRS [47] (*General Packet Radio System*) dite de (2.5G).

Les téléphones mobiles actuellement disponibles sont dits "de troisième génération" (3G), ils utilisaient notamment la norme UMTS [47] (Universal Mobile Telephone System) (une description de ce norme et les normes précédentes est fournie en annexe B). Ils intègrent de nombreuses innovations dans la technologie et les services et permettent d'envoyer des SMS, des images, des photographies, des sons et des vidéos, lire et rédiger des e-mails, naviguer sur Internet, écouter de la musique ou encore regarder la télévision.

2.3.3 Technique de la téléphonie mobile

2.3.3.1 Structure d'un téléphone mobile

La structure d'un téléphone mobile répond à des exigences de taille, de poids, de convivialité et d'autonomie.

Si on démonte un téléphone portable, on constate qu'il ne contient que quelques éléments:

* **Une carte de circuit (ou logique)** : contenant le cerveau du téléphone
* **Une antenne** : elle émet et capte les ondes électromagnétiques
* **Un écran à cristaux liquides (LCD)** : affiche le numéro de téléphone à composer, les messages reçus, les menus de configuration…
* **Un clavier** : permet de composer les numéros de téléphone, taper des messages à envoyer, de configurer l'appareil,...
* **Un microphone** : il est généralement de type électret, il convertit des ondes sonores acoustiques en impulsions électriques.
* **Un haut-parleur** : fait parfois office de *buzzer* (sonnerie).
* **Une batterie** : c'est comme la batterie de PDA.

Figure 2.13. *Constitution d'un téléphone portable* [44]

2.3.3.1.1 Carte de circuit (ou logique)

La carte de circuit est le cœur du système, elle est composé de plusieurs éléments, les plus importants sont :

* **Le microprocesseur :** C'est une pièce importante. Il traite et réachemine le flux d'informations numériques, il sert à orchestrer et faire fonctionner tout ce qui a trait aux commandes et au système d'exploitation du téléphone. Il procède à tous les calculs que nécessitent les fonctions prises en charge par les téléphones modernes (lecture de musique, vidéo, photo et communication de données sans fil, etc).
* **La mémoire ROM :** Elle fournit le stockage pour le système d'exploitation du téléphone et des caractéristiques personnalisables, comme l'annuaire téléphonique.
* **La mémoire RAM :** C'est comme la RAM de PDA.
* **Le socket de la carte SIM *(Subscriber Identification Module)*:** C'est une orbite pour mettre la carte SIM qui est une carte à puce qui mémorise les informations spécifiques à l'utilisateur, son répertoire de numéro et ses messages, le réseau auquel il est abonné. Il en existe deux formats : ISO (*International Organization for Standardization*) (taille d'une carte de crédit) et micro (pour les téléphones les plus petits) [41].

* **DSP (The digital signal processor)** [45] **:** Est un processeur très personnalisée conçue pour transformer le signal numérique envoyé par le processeur en un signal analogique amplifié - à l'aide d'un algorithme de conversion - qui pourra être diffusé par le haut-parleur du téléphone sous forme de son. Le DSP effectue aussi le chemin inverse et traite la voix qui lui parvient sous forme de signal analogique à partir du microphone du téléphone pour aboutir à l'émission du signal par son antenne.

* **Les puces pour communiquer (Bluetooth et infrarouge) :** La communication sans fil Bluetooth ou infrarouge est gérée par des puces dédiées qui collaborent étroitement avec le processeur central.

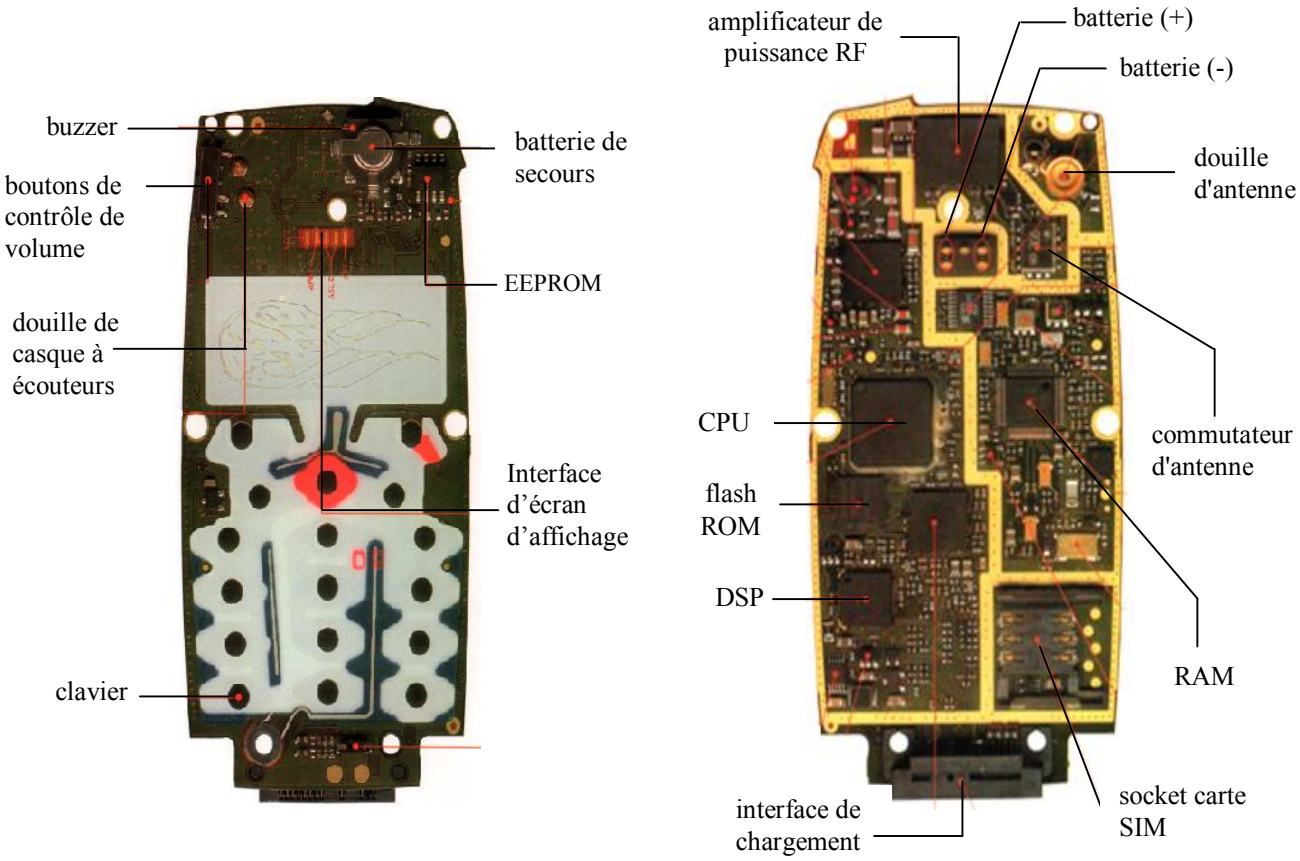


Figure 2.14. *La carte de circuit d'un téléphone ERICSSON (T20).* [46]

2.3.3.2 Caractéristiques d'un téléphone mobile

Sa fonction d'usage est la communication vocale mais le téléphone mobile permet d'envoyer des messages désignés SMS. Avec l'évolution de l'électronique, le téléphone mobile permet aujourd'hui de lire et rédiger des emails, écouter la radio, naviguer sur Internet, écouter de la musique, regarder la télévision, jouer, photographier et enregistrer des vidéos…

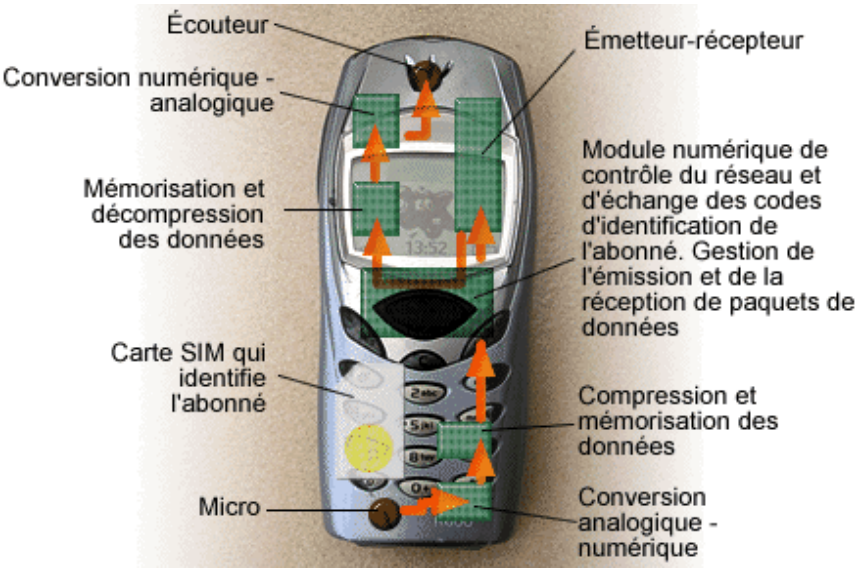


Figure 2.15. *La communication vocale du téléphone mobile.* [47]

2.3.3.3 Fonctionnement d'un réseau de téléphonie mobile

Un réseau de téléphonie mobile est composé d'un ensemble d'émetteurs-récepteurs radioélectriques appelés stations de base (ou BTS) distribués sur un territoire donné. Ces stations, de courte portée, assurent la liaison avec le téléphone mobile par ondes hertziennes autour de 900 MHz et 1800 MHz.

La zone sur laquelle un terminal peut établir une liaison avec une station de base déterminée s'appelle une *cellule* qui ne peut accueillir qu'un nombre limité d'usagers. C'est pourquoi les cellules sont toutes petites en ville, et plus vastes en milieu rural. Les mêmes fréquences ne sont réutilisées qu'à une distance suffisante afin d'éviter les interférences.

La téléphonie mobile fait intervenir deux notions fondamentales : l'*itinérance* et le *handover* [41] :

- L'itinérance, ou *roaming* est une technique qui permet à un abonné d'appeler et être appelé dans n'importe quel endroit, y compris en dehors du territoire national, avec la nécessité de gestion de la localisation des mobiles.
- *Handover* ou inter-cellulaire est une technique qui permet à un utilisateur de passer d'une cellule à une autre sans que la communication soit interrompue, c'est-à-dire de changer de station de base tout en maintenant la continuation du service.

Lorsqu'un téléphone mobile entre dans une cellule et il veut faire un appel vers un autre téléphone mobile, une liaison s'établit avec la station de base et il communique alors son numéro d'identité à cette station qui lui réserve un canal jusqu'à ce qu'il quitte la zone, et qui

interroge ensuite la centrale principale du réseau auquel est abonné le destinataire. La centrale renvoie alors l'adresse de la cellule du destinataire. La communication peut commencer.

Mais la gestion de la communication entre les mobiles, du *handover* et de l'itinérance nécessite des équipements particuliers qui ne sont pas présents dans les réseaux téléphoniques classiques.

2.3.3.4 L'architecture du réseau

Un réseau PLMN (*Public Land Mobile Network)* peut être vu comme un système qui est adéquat pour les communications téléphoniques de parole et qui assure un accès radio au réseau téléphonique général, le RTCP (Réseau Téléphonique Commuté Public). L'architecture d'un réseau PLMN peut être divisée en trois sous-systèmes [42] :

1. Le sous-système radio (BSS, *Base Station Sub-system*) ;
2. Le sous-système réseau ou d'acheminement (NSS, *Network Sub-System*) ;
3. Le sous-système opérationnel ou d'exploitation et de maintenance.

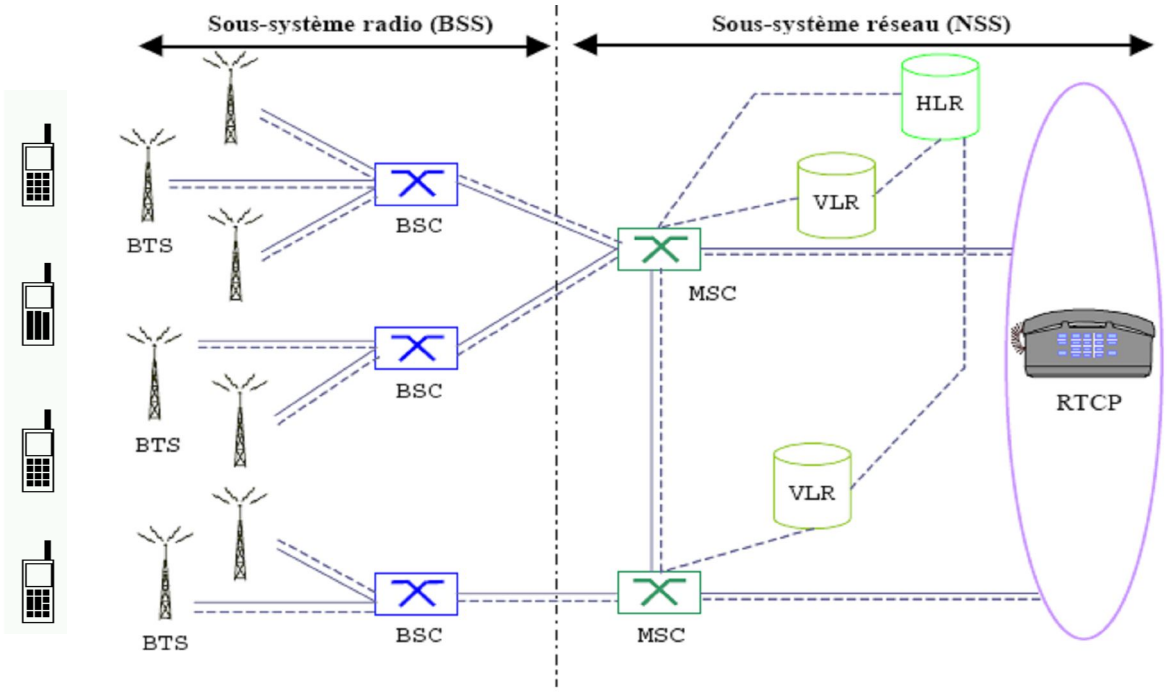


Figure 2.16. *Architecture du réseau GSM* [41]

2.3.3.4.1 L'architecture du sous-système radio (BSS)

Le sous-système radio BSS (*Base Station Subsystem*) gère la transmission radio. Il est composé de un ou plusieurs [53] :

- Stations mobiles (téléphone cellulaire) ;
- BTS (Base Transceiver Station) qui assure l'interface entre structures fixes et mobiles. Elles s'occupent de la transmission radio et réalisent l'ensemble des mesures radio nécessaires pour vérifier qu'une communication se déroule correctement ;
- BSC, contrôleur de station de base (*Base Station Controller*) est l'organe intelligent du BSS. Il a pour fonction de gérer la ressource radio. Il distribue les canaux, prend la décision de l'exécution d'un *handover*, exploite les mesures effectuées par les BTS pour contrôler la puissance d'émission du mobile et/ou des BTS.

2.3.3.4.2 L'architecture du sous-système réseau (NSS)

Le NSS (*Network SubSystem*) s'occupe de toutes les fonctions du niveau réseau (routage, interconnexion) et d'analyse d'informations contenues dans des bases de données. Il est composé de [41] :

- MSC (*Mobile services Switching Center*) gère l'établissement des communications entre un mobile et un autre MSC, et assure la commutation entre les abonnés du réseau mobile et ceux du réseau Téléphonique commuté public (RTCP).
- Les bases de données HLR (*Home Location Register*); il peut être considéré comme la mémoire centralisée du réseau ;
- Les VLR (*Visitor Location Register*) qui peuvent être considérés comme des mémoires temporaires.

2.3.3.4.3 L'architecture d'exploitation et de maintenance

Il s'occupe de différentes fonctions : déclaration des abonnés, des terminaux, facturations, la gestion de la sécurité (détection des intrusions), la maintenance, etc.

2.4 Conclusion

Le long de ce chapitre, nous avons présenté deux dispositifs mobiles qui sont le téléphone mobile et le PDA. Ses usagers prennent toutefois de plus en plus conscience des problèmes de sécurité. Les moyens de communication usuels – téléphone fixe ou mobile, télécopie, messagerie électronique – sont, en effet, les sources les plus riches pour le vol d'informations [37]. Ainsi, un dispositif perdu peut contenir des données sensibles (extrait de la base de données clients, par exemple), ce sont des données à la fois commerciales, et à caractère personnel. Pour protéger ces données sensibles en cas de vol ou de perte, la solution qui consiste à chiffrer l'ensemble des données permet de transformer un PDA ou un téléphone mobile en « coffre-fort numérique ».

CHAPITRE 3

Le standard de chiffrement avancé (AES)

CHAPITRE 3
Le standard de chiffrement avancé (AES)

3.1 Introduction

Le cryptosystème AES est issu d'un appel d'offre international lancé en janvier 1997 et ayant reçu 15 propositions en première ronde [26]. Ces algorithmes ont été évalués par des experts, avec forum de discussion sur Internet, et organisation de conférences. Pour la même raison, 600 CD-ROM portant ces algorithmes ont été distribués dans plus de 50 pays en décembre 1998. Après la deuxième ronde qui a débuté le 22 Mars 1999 lors d'une conférence à Rome [55], Rijndael a été déclaré vainqueur par le NIST le 2 octobre 2000 sur les 5 candidats finalistes : MARS [56], RC6 [57], Rijndael [58], Serpent [59] et TwoFish [60]. Les résultats des votes étaient comme suit [29] :

Rijndael : 86 votes, Serpent : 59 votes, Twofish : 31 votes, RC6 : 23 votes, MARS : 13 votes

Donc, le terme d'AES (***A**dvanced **E**ncryption **S**tandard*) remplace désormais celui de Rijndael (dont le nom est basé sur les noms de ses deux inventeurs, **Joan Daemen** et **Vincent belges Rijmen**) [30] sans que l'algorithme ne soit modifié. Mais en réalité, AES est un sous-ensemble de Rijndael, puisque ce dernier offre des tailles de blocs et de clés qui sont des multiples de 32 compris entre 128 et 256 bits, tandis que AES travaille avec des blocs de 128 bits seulement et emploie trois tailles différentes de clés : 128, 192 et 256 bits, donc c'est un algorithme de chiffrement par bloc. Mais contrairement à DES (Data Encryption Standard), qui est basé sur la structure de Feistel, AES est un réseau de substitution-permutation.

Dans ce chapitre, nous allons détailler l'algorithme cryptographique AES, en présentant les définitions utilisés dans ce standard ainsi que ses préliminaires mathématiques. En suite, nous allons aborder sa méthode de chiffrement, de déchiffrement ainsi que l'opération de génération de clé. Le chapitre se termine par une présentation de l'ensemble de cryptanalyse qui peut menacer cet algorithme.

3.2 Définitions

3.2.1 Glossaire des acronymes, termes, symboles et fonctions

Les définitions suivantes sont utilisées dans ce standard [63] :

AES, Transformation Affine, Bit, Octet, Bloc, Tableau, Clé de chiffrement, Chiffrement, Texte chiffré, Déchiffrement, Texte clair, Expansion de clé, Clé de tour, Rijndael, Etat, S-box, Mot.

Les symboles, et les fonctions suivants sont utilisés dans cette norme [63] :

K, Nb, Nk, Nr, AddRoundKey (), SubBytes (), ShiftRows (), MixColumns (), InvSubBytes (), InvShiftRows (), InvMixColumns (), Rcon [], SubWord (), RotWord (), XOR, $\oplus$, $\otimes$, $\bullet$.

Pour voir ces définitions en détaille, voir Annexe C.

3.3 Notation et structure de données

3.3.1 Entrées et sorties

L'entrée et la sortie pour l'algorithme AES est une séquence de 128 bits (bit binaire). Les bits de ces séquences seront numérotés à partir de zéro. A chaque bit on associe un index i qui sera dans l'une des plages $0 \leq i < 128$, $0 \leq i < 192$ ou $0 \leq i < 256$.

3.3.2 Octets

L'unité de base de traitement dans l'algorithme AES est l'octet qui est une séquence de huit bits et qui est traitée comme une seule entité. L'entrée, la sortie et la clé de chiffrement qui sont notée par '*a*', sont traitées comme des tableaux d'octets. Chaque octet sera référencé par a_n, où n sera être dans l'une des plages suivantes:

Longueur de bloc = 128 bits, $0 \leq n < 16$;
Longueur de clé = 128 bits, $0 \leq n < 16$;
Longueur de clé = 192 bits, $0 \leq n < 24$;
Longueur de clé = 256 bits, $0 \leq n < 32$.

Chaque octet a_n sera présenté comme la concaténation de 8 bits dans l'ordre $\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$. Ces octets sont interprétés comme des éléments de corps finis en utilisant un polynôme de représentation [66] :

$$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \sum_{i=0}^{7} b_i x^i \ . \quad (3.3.1)$$

Par exemple, {01100011} identifie l'élément de corps fini $x^6 + x^5 + x + 1$.
Ainsi, l'élément {01100011} peut être représentée en hexadécimale par {63}.

3.3.3 Tableaux d'octets

On a la séquence d'entrée de 128-bits suivante :

$$input_0\ input_1\ input_2\ \ldots\ input_{126}\ input_{127}$$

Un tableau d'octets est représenté par la forme suivante (pour un bloc de taille 128 bits) :

$$a_0\ a_1\ a_2 \ldots\ldots a_{15}$$

Où :

$$a_0 = \{input_0, input_1, \ldots, input_7\};$$
$$a_1 = \{input_8, input_9, \ldots, input_{15}\};$$
$$\vdots$$
$$a_{15} = \{input_{120}, input_{121}, \ldots, input_{127}\}.$$

D'une façon générale :

$$a_n = \{input_{8n}, input_{8n+1}, \ldots, input_{8n+7}\}. \qquad (3.3.2)$$

Séquence de bit d'entrée	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	…
Numéro d'octet	0								1								2								…
Numéro de bit dans l'octet	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	…

Tableau 3.1. *Indices des octets et des bits* [63]

3.3.4 L'état

Les opérations de l'algorithme AES sont effectuées sur une matrice de quatre lignes et Nb=4 colonnes d'octets appelée l'état, où Nb est la longueur de bloc (128 bits), divisée par 32. Dans l'état désignée par le symbole S, chaque octet $S_{i,j}$ a deux indices, l'indice *i* pour désigner le numéro de ligne $0 \leq i < 4$ et l'indice *j* pour désigner le numéro de colonne $0 \leq j <$ (Nb=4).

Les octets lus en entrée $in_0, in_1, \ldots in_{15}$ y sont copiés colonne après colonne dans la matrice. A la fin des opérations de chiffrement ou déchiffrement, la valeur finale de la matrice d'état est copié dans la sortie – les octets de sortie $out_0, out_1, \ldots out_{15}$.

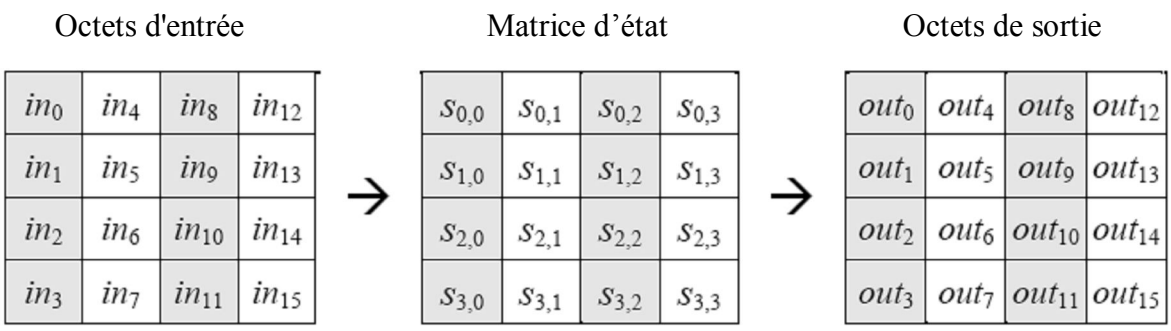


Figure 3.1. *Matrice d'état, l'entrée et la sortie* [63]

Le tableau d'entrée est copié à l'état selon le schéma:

$$s[i, j] = in[i + 4j] \qquad \text{pour } 0 \le i < 4 \text{ et } 0 \le j < Nb, \tag{3.3.3}$$

L'état est copié au tableau de sortie comme suit:

$$out[i + 4j] = s[i, j] \qquad \text{pour } 0 \le i < 4 \text{ et } 0 \le j < Nb. \tag{3.3.4}$$

Par conséquent, l'état peut être considéré comme un tableau de quatre mots de 32 bits, comme suit :

$$\begin{aligned} w_0 &= s_{0,0}\, s_{1,0}\, s_{2,0}\, s_{3,0} & w_2 &= s_{0,2}\, s_{1,2}\, s_{2,2}\, s_{3,2} \\ w_1 &= s_{0,1}\, s_{1,1}\, s_{2,1}\, s_{3,1} & w_3 &= s_{0,3}\, s_{1,3}\, s_{2,3}\, s_{3,3} \end{aligned} \tag{3.3.5}$$

3.4 Préliminaires mathématiques

3.4.1 Les opérations dans $GF(2^8)$

Les opérations effectuées par l'AES (l'addition et la multiplication) le sont dans le corps fini $GF(2^8)$ (*Galois Fields* [64]) [65]. Ce corps est composé de 2^8 éléments (les octets de 0 à 255), où ces éléments sont représentés par des polynômes.

Si on parle de $GF(2^8)$, on parle de $\dfrac{\mathbb{Z}_2[X]}{m(X)\,\mathbb{Z}_2[X]}$, c'est à dire [66]:

$$GF(2^8) \;\Leftrightarrow\; \frac{\mathbb{Z}_2[X]}{m(X)\,\mathbb{Z}_2[X]} \tag{3.4.1}$$

Ceci signifie que les calculs sont effectués sur des polynômes de degré 7 à coefficients dans $\{0, 1\}$, *modulo* un polynôme irréductible $m(X)$ de degré 8 (valant dans l'AES $X^8 + X^4 + X^3 + X + 1$). Un polynôme est irréductible si son seul diviseurs sont 1 et lui même.

$$\mathbb{Z}_2[\mathrm{X}] \Leftrightarrow \{0,1\} \tag{3.4.2}$$

$$m(\mathrm{X}) = \mathrm{X}^8 + \mathrm{X}^4 + \mathrm{X}^3 + \mathrm{X} + 1 \tag{3.4.3}$$

3.4.1.1 L'addition de deux polynômes

L'addition de deux éléments dans un corps fini est équivaut à un ou-exclusif XOR (signalés par $\oplus$) bit à bit entre les octets les représentants. Pour les deux octets $\{a_7a_6a_5a_4a_3a_2a_1a_0\}$ et $\{b_7b_6b_5b_4b_3b_2b_1b_0\}$, la somme est $\{c_7c_6c_5c_4c_3c_2c_1c_0\}$, où chaque $c_i = a_i \oplus b_i$ (c'est-à-dire, $c_7 = a_7 \oplus b_7$, $c_6 = a_6 \oplus b_6$, ... $c_0 = a_0 \oplus b_0$).

Par exemple :

$$\begin{array}{rr} & X^6 + X^3 + X^2 + 1 \\ \oplus & X^7 + X^4 + X^3 + X + 1 \\ \hline = & X^7 + X^6 + X^4 + X^2 + X \end{array} \Leftrightarrow \begin{array}{rr} & 01001101_2 \\ \oplus & 10011011_2 \\ \hline = & 11010110_2 \end{array} \tag{3.4.4}$$

3.4.1.2 La multiplication de deux polynômes

La multiplication de deux polynômes dans GF(2^8) (signalé par $\otimes$) correspond à une multiplication binaire entre octets *modulo* $m(X)$. Pour calculer le reste d'un polynôme $p(X)$ *modulo* $m(X)$ il suffit d'appliquer l'algorithme suivant [66] :

$$\begin{array}{l} \text{tant que } \textit{degré(p)} \geq 8 \text{ faire} \\ \quad p(X) \leftarrow p(X) \oplus m(X) \cdot X^{\textit{degré(p)}-\textit{degré(m)}} \\ \text{fin tant que} \end{array} \tag{3.4.5}$$

L'exemple suivant résume ces propriétés :

$$\begin{array}{rll} & 11101010 & X^7 + X^6 + X^5 + X^3 + X \\ \otimes & 00000101 & X^2 + 1 \\ \hline & 11101010 & \\ \oplus & 1110101000 & \\ \hline = & 1101000010 & \textit{résultat brut} \\ \oplus & 1000110110 & \textit{modulo } X \cdot m(X) \\ \hline = & 101110100 & \\ \oplus & 100011011 & \textit{modulo } m(X) \\ \hline = & 01101111 & \textit{résultat} : X^6 + X^5 + X^3 + X^2 + X + 1 \end{array} \tag{3.4.6}$$

Notons que la multiplication d'un polynôme $p(X)$ par 1 donne $p(X)$:

$$\begin{array}{rrl} & 10000010 & X^7 + X \\ \otimes & 01 & 1 \\ \hline = & 10000010 & X^7 + X \end{array} \qquad (3.4.7)$$

que la multiplication de $p(X)$ par X (soit 10_2) revient à décaler l'octet $p(X)$ d'un bit sur la gauche et à le réduire *modulo* $m(X)$:

$$\begin{array}{rrl} & 10000010 & X^7 + X \\ \otimes & 10 & X \\ \hline = & 100000100 & \\ \oplus & 100011011 & \textit{modulo } m(X) \\ \hline = & 00011111 & X^4 + X^3 + X^2 + X + 1 \end{array} \qquad (3.4.8)$$

et enfin que la multiplication par $X+1$ (soit 11_2) revient à effectuer une multiplication par X et une addition par le polynôme lui-même :

$$\begin{array}{rrl} & 10000010 & X^7 + X \\ \otimes & 11 & X + 1 \\ \hline = & 10000010 & \\ \oplus & 100000100 & \\ \hline = & 110000110 & \textit{résultat brut} \\ \oplus & 100011011 & \textit{modulo } m(X) \\ \hline = & 010011101 & X^7 + X^4 + X^3 + X^2 + 1 \end{array} \quad \Leftrightarrow \quad \begin{array}{rrl} & 10000010 & X^7 + X \\ \otimes & 11 & X + 1 \\ \hline = & 100000100 & X^7 + X \otimes X \\ \oplus & 100011011 & \textit{modulo } m(X) \\ \hline = & 00011111 & X^4 + X^3 + X^2 + X + 1 \\ \oplus & 10000010 & X^7 + X \\ \hline = & 010011101 & X^7 + X^4 + X^3 + X^2 + 1 \end{array} \qquad (3.4.9)$$

3.4.2 Polynômes à coefficients dans $GF(2^8)$

Un mot de 32 bits (4 octets) peut être représenté par un polynôme de degré 3 à coefficients dans $GF(2^8)$, où chaque coefficient représentant un octet, tout comme nous avons vu, un octet peut être représenté par un polynôme de degré 7.

Etant donné deux polynômes $a(X)$ et $b(X)$:

$$\begin{aligned} a(X) &= a_3X^3 + a_2X^2 + a_1X + a_0 \\ b(X) &= b_3X^3 + b_2X^2 + b_1X + b_0 \end{aligned} \qquad (3.4.10)$$

où $\{a_3, a_2, a_1, a_0, b_3, b_2, b_1, b_0\}$ sont des octets.

L'addition de deux mots est alors égale à l'addition des polynômes les représentants, soit un ou-exclusif XOR entre les coefficients de même degré [66].

$$a(X) + b(X) = (a_3 \oplus b_3)X^3 + (a_2 \oplus b_2)X^2 + (a_1 \oplus b_1)X + (a_0 \oplus b_0) \qquad (3.4.11)$$

La multiplication de deux mots se fait en deux étapes. Dans la première étape, on calcule le produit polynomial $c(x) = a(x) \otimes b(x)$, dont on peut calculer les coefficients de ce polynôme de degré 6 :

$$a(X) \times b(X) = c(X) = c_6 X^6 + c_5 X^5 + c_4 X^4 + c_3 X^3 + c_2 X^2 + c_1 X + c_0 \qquad (3.4.12)$$

où

$$\begin{aligned} c_0 &= a_0 \otimes b_0 & c_4 &= a_3 \otimes b_1 \oplus a_2 \otimes b_2 \oplus a_1 \otimes b_3 \\ c_1 &= a_1 \otimes b_0 \oplus a_0 \otimes b_1 & c_5 &= a_3 \otimes b_2 \oplus a_2 \otimes b_3 \\ c_2 &= a_2 \otimes b_0 \oplus a_1 \otimes b_1 \oplus a_0 \otimes b_2 & c_6 &= a_3 \otimes b_3 \\ c_3 &= a_3 \otimes b_0 \oplus a_2 \otimes b_1 \oplus a_1 \otimes b_2 \oplus a_0 \otimes b_3 \end{aligned} \qquad (3.4.13)$$

Afin de rester dans l'espace des mots de quatre octets, la deuxième étape de la multiplication consiste à réduire c(x) *modulo* un polynôme de degré 4 qui vaut pour l'AES $(X^4 + 1)$, formant un groupe que l'on pourrait écrire [66] :

$$A = \frac{GF(2^8)[X]}{(X^4 + 1)GF(2^8)[X]} \qquad (3.4.14)$$

où

$$\begin{aligned} & a(X) \otimes b(X) \\ = \; & c(X) \; modulo \; (X^4 + 1) \\ = \; & d(X) \\ = \; & d_3 X^3 + d_2 X^2 + d_1 X + d_0 \end{aligned} \qquad (3.4.15)$$

avec

$$\begin{aligned} d_0 &= a_0 \otimes b_0 \oplus a_3 \otimes b_1 \oplus a_2 \otimes b_2 \oplus a_1 \otimes b_3 \\ d_1 &= a_1 \otimes b_0 \oplus a_0 \otimes b_1 \oplus a_3 \otimes b_2 \oplus a_2 \otimes b_3 \\ d_2 &= a_2 \otimes b_0 \oplus a_1 \otimes b_1 \oplus a_0 \otimes b_2 \oplus a_3 \otimes b_3 \\ d_3 &= a_3 \otimes b_0 \oplus a_2 \otimes b_1 \oplus a_1 \otimes b_2 \oplus a_0 \otimes b_3 \end{aligned} \qquad (3.4.16)$$

Ces opérations peuvent être mises sous forme matricielle comme suit [66] :

$$\begin{pmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{pmatrix} = \begin{pmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \qquad (3.4.17)$$

Puisque ($X^4 + 1$) n'est pas un polynôme irréductible, A n'est pas un corps et ses éléments ne sont pas forcément inversibles, cependant, l'AES utilise un polynôme $a(\mathrm{X})$ inversible dans A.

$$\begin{aligned} a(\mathrm{X}) &= (0\mathrm{x}03)\mathrm{X}^3 + (0\mathrm{x}01)\mathrm{X}^2 + (0\mathrm{x}01)\mathrm{X} + (0\mathrm{x}02) \\ a^{-1}(\mathrm{X}) &= (0\mathrm{x}0\mathrm{b})\mathrm{X}^3 + (0\mathrm{x}0\mathrm{d})\mathrm{X}^2 + (0\mathrm{x}09)\mathrm{X} + (0\mathrm{x}0\mathrm{e}) \end{aligned} \qquad (3.4.18)$$

3.5 Spécification de l'algorithme AES

Dans l'algorithme AES, la taille du bloc (sois d'entré, de sortie ou d'état) est 128 bits, c'est-à-dire, il est composé de quatre mot de 32 bits d'où *Nb* (nombre de colonne) = 4. La clé de chiffrement, possède trois taille différentes 128, 192 ou 256 bits, d'où elle est composé de *Nk* (nombre de colonne) = 4, 6 ou 8 respectivement de mot de 32 bits.

Chaque bloc de 128 bits subit une séquence de transformations, ces différentes transformations, définissant un *tour Nr*, sont répétées plusieurs fois. Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours, comme le résume ce tableau suivant :

	Nb	***Nk***	**Nombre de tour (*Nr*)**
AES-128	4	4	10
AES-192	4	6	12
AES-256	4	8	14

Tableau 3.2. *Combinaison bloc, clé, tour* [63]

Ces transformations consistent en quatre opérations principales qui sont :

1. une substitution non linéaire ;
2. une permutation circulaire des octets de chaque ligne ;
3. une multiplication dans $\frac{GF(2^8)[X]}{(X4 + 1)GF(2^8)[X]}$ pour chaque colonne ;
4. une addition de l'état par la clé.

3.5.1 Le chiffrement

Le chiffrement commence par le copiage de l'entrée au tableau d'état. Ensuite, un tour initial est appliqué en ajoutant la clé de chiffrement à cet état, qui sera après transformé en itérant 10, 12 ou 14 fois (selon la longueur de clé) quatre transformations sur les octets :

SubBytes(), ShiftRows(), MixColumns() et AddRoundKey() en utilisant le cadencement de clé , avec le tour final différant du premier (Nr-1) où la transformation MixColumns() n'est pas inclus. Le résultat final est ensuite copié dans la sortie.

La figure suivante résume le principe de fonctionnement de cet algorithme de chiffrement :

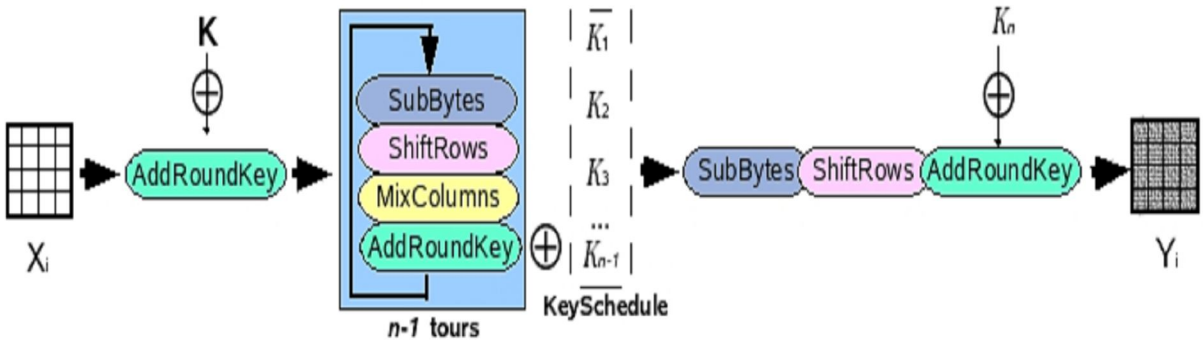


Figure 3.2. *Schéma général de l'AES* [62]

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
 début
  1       byte state[4,Nb]
  2
  3       state = in      // l'entrée est copiée dans le bloc
  4
  5       AddRoundKey(state, w[0, Nb-1])
  6
  7       pour round = 1 à Nr-1 par pas de 1 faire
  8            SubBytes(state)
  9            ShiftRows(state)
 10            MixColumns(state)
 11            AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
 12       finpour
 13
 14       SubBytes(state)
 15       ShiftRows(state)
 16       AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
 17
 18       out = state   // copie du bloc dans la sortie
 fin
```

Figure 3.3. *Pseudo-code - chiffrement* [66]

où le tableau w[] contient le cadencement de clé produit par la routine d'expansion de clé (voir section 3.5.3).

3.5.1.1 La transformation AddRoundKey()

Dans cette opération, une clé de tour est ajoutée à l'état par un simple XOR. La clé de tour est dérivée de la clé de chiffrement à partir de l'algorithme d'expansion de clé, avec une taille 128 bits (4 mots de 4 octets). Les mots sont additionnés suivant la formule [66] :

$$\text{pour } 0 \le \text{tour} \le Nr \quad \text{et} \quad 0 \le \text{c} < (Nb = 4)$$

$$[s'_{0,c}\ ,\ s'_{1,c}\ ,\ s'_{2,c}\ ,\ s'_{3,c}] = [s_{0,c}\ ,\ s_{1,c}\ ,\ s_{2,c}\ ,\ s_{3,c}] \oplus [w_{\text{tour} * \text{Nb}+c}] \qquad (3.5.1)$$

où le $[w_i]$ représente le $i^{ème}$ mot de la clé de tour. Dans le tour initial (tour = 0), (w_0, w_1, w_2, w_3) représente la clé de chiffrement.

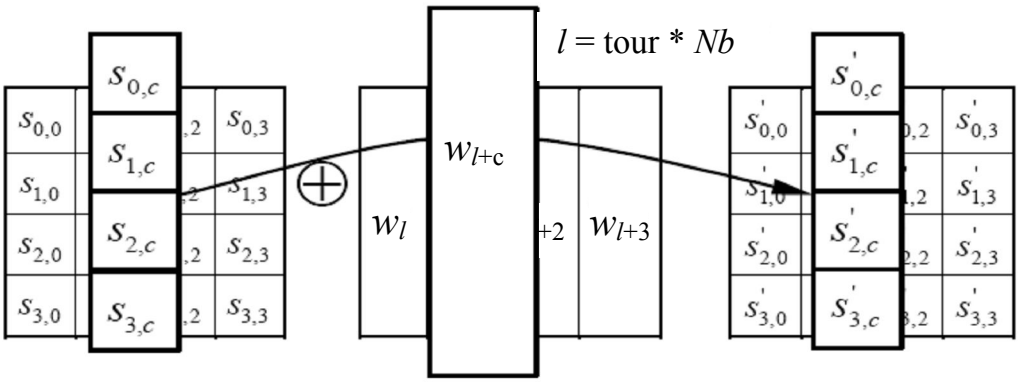


Figure 3.4. *La transformation AddRoundKey()* [63]

3.5.1.2 La transformation SubBytes()

La transformation SubBytes() est une substitution non-linéaire d'octets, qui remplace chaque octet par un autre de façon indépendante utilisant une table de substitution (S-box). Cette table (S-box) est inversible, et elle est construite par la composition des deux transformations :

1. Prendre l'inverse multiplicatif dans le corps fini GF(2^8), avec 0^{-1}=0.
2. Appliquer la transformation affine suivante (dans GF(2^8)) [66]:

$$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i \ , \text{pour} \quad 0 \le i < 8 \qquad (3.5.2)$$

où b_i est le $i^{ème}$ bit de l'octet et c_i le $i^{ème}$ bit d'un octet c qui vaut 01100011_2 (0x63).

Cette transformation peut prendre la forme matricielle suivante [66] :

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \quad (3.5.3)$$

La table S issue du précalcul des valeurs de chacun des 256 polynômes de GF(2^8) est indexée par les 4 bits de poids fort et les 4 bits de poids faible de l'octet. Par exemple, si $s_{1,1}$ = {75}, alors la valeur de substitution sera déterminée par l'intersection de la ligne avec l'indice '7' et la colonne avec l'index '5' de la Fig 3.5. Cela aboutirait que $s'_{1,1}$ ayant une valeur de {9d}.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 3.5. *La table de substitution S-box* [63]

3.5.1.3 La transformation ShiftRows()

Dans cette transformation, les trois dernières lignes de l'état sont cycliquement permutées vers la gauche, selon l'index r de la ligne [66] :

$$s''_{r,c} = s'_{r,(c+shift(r,Nb))mod\ Nb} \quad , \quad \text{pour} \quad 0 < r < 4 \quad \text{et} \quad 0 \leq c < Nb \quad (3.5.4)$$

où (pour l'AES) : $shift(0, 4) = 0$; $shift(1, 4) = 1$; $shift(2, 4) = 2$; $shift(3, 4) = 3$

La figure 3.6. illustre cette transformation :

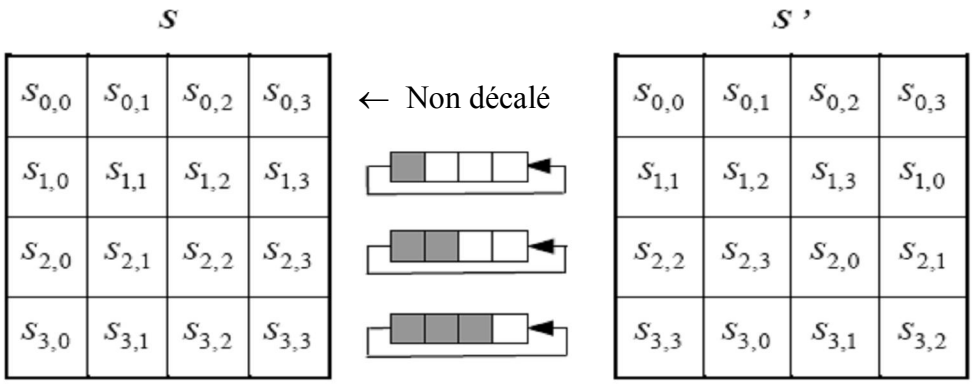


Figure 3.6. *La permutation circulaire sur les trois dernières lignes du bloc* [63]

3.5.1.4 La transformation MixColumns()

La transformation MixColumns() traite chaque colonne comme un élément du corps *A* (voir la section 3.4.2) c'est-à-dire un polynôme de degré 3, on calcule dans ce corps le produit de ce polynôme avec un polynôme fixe *a*(X) inversible [66].

$$a(\mathrm{X}) = (0\mathrm{x}03)\mathrm{X}^3 + (0\mathrm{x}01)\mathrm{X}^2 + (0\mathrm{x}01)\mathrm{X} + (0\mathrm{x}02) \qquad (3.5.5)$$

Ces opérations peuvent être mises sous forme matricielle [63]:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{pour} \quad 0 \leq c < Nb \qquad (3.5.6)$$

La figure 3.7. illustre cette transformation :

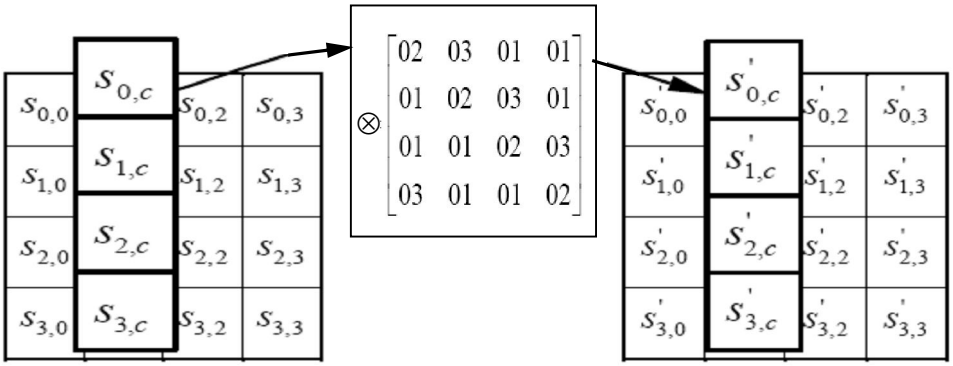


Figure 3.7. *MixColumns() fonctionne sur l'état colonne par colonne* [63].

3.5.2 Le déchiffrement

Le déchiffrement est l'inverse de chiffrement, où les transformations de ce dernier peuvent être inversés et implémentés dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse. L'inverse des transformations de chiffrement sont : InvSubBytes(), InvShiftRows(), InvMixColumns(), avec AddRoundKey() reste la même [63].

Le déchiffrement est décrit dans ce pseudo-code suivant :

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
début
 1       byte state[4,Nb]
 2
 3       state = in      // l'entrée est copiée dans le bloc
 4
 5       AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
 6
 7       pour round = Nr-1 à 1 par pas de -1 faire
 8            InvShiftRows(state)
 9            InvSubBytes(state)
10            AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
11            InvMixColumns(state)
12       finpour
13
14       InvShiftRows(state)
15       InvSubBytes(state)
16       AddRoundKey(state, w[0, Nb-1])
17
18       out = state   // copie du bloc dans la sortie
fin
```

Figure 3.8. *Pseudo-code - déchiffrement* [66]

3.5.2.1 La transformation InvSubBytes()

InvSubBytes() est l'inverse de la transformation SubBytes(), dans laquelle l'inverse de la table S-box est appliqué à chaque octet de l'état. Cette table est obtenue en appliquant l'inverse de la transformation affine (voir équation 3.5.2) puis en prenant l'inverse multiplicatif du résultat dans GF(2^8), ce qui donne :

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 3.9. *L'inverse de la table de substitution S-box* [63]

3.5.2.2 La transformation InvShiftRows()

La transformation InvShiftRows() est l'inverse de ShiftRows(), tels que les trois dernières lignes de l'état sont cycliquement permutées vers la droite, suivant cette formule [66] :

$$S''_{r,(c+shift(r,Nb))mod\ Nb} = S'_{r,c} \quad , \quad \text{pour} \quad 0 < r < 4 \quad \text{et} \quad 0 \le c < Nb \qquad (3.5.7)$$

La figure 3.10. illustre cette transformation :

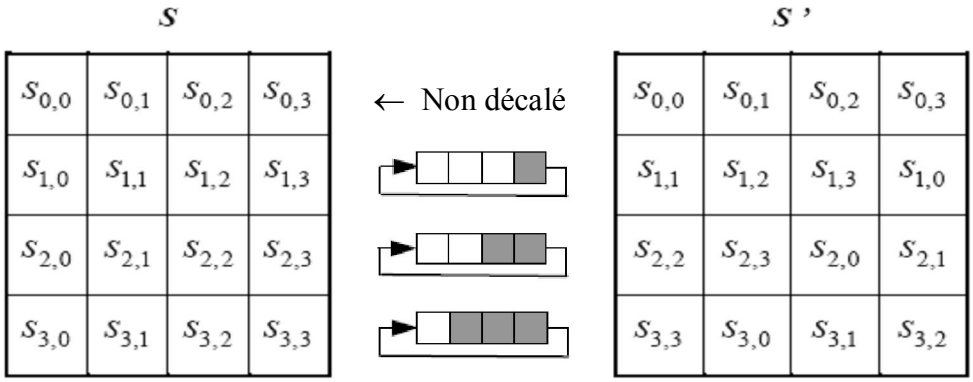


Figure 3.10. *La permutation circulaire des trois dernières lignes du bloc vers la droite* [63]

3.5.2.3 La transformation InvMixColumns()

La transformation InvMixColumns() est l'inverse de MixColumns(), où chaque colonne est traité comme un élément du corps *A* (voir la section 3.4.2) c'est-à-dire un

polynôme de degré 3, on calcule dans ce corps le produit de ce polynôme avec un polynôme fixe $a^{-1}(X)$ [66].

$$a^{-1}(x) = (0x0b)x^3 + (0x0d)x^2 + (0x09)x + (0x0e) \qquad (3.5.8)$$

Ces opérations peuvent être mises sous forme matricielle [63] :

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{pour} \quad 0 \leq c < Nb \qquad (3.5.9)$$

La figure 3.11. illustre cette transformation :

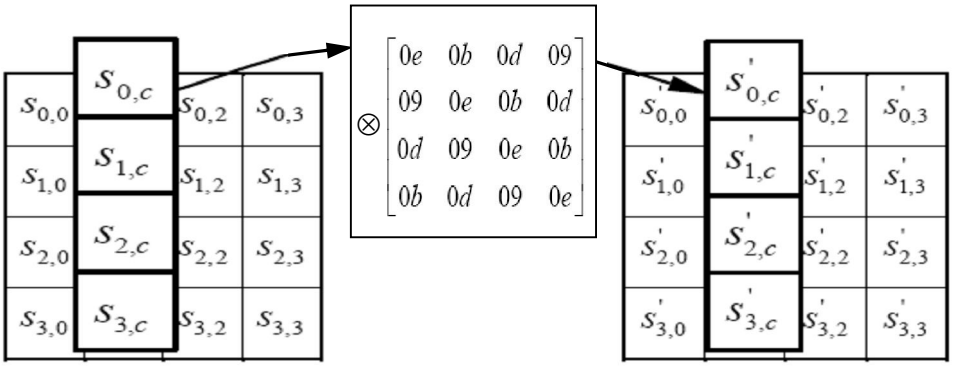


Figure 3.11. *InvMixColumns() fonctionne sur l'état colonne par colonne* [63].

3.5.2.4 Le déchiffrement équivalent

Dans le déchiffrement (pseudo-code 3.8) la séquence de transformations diffère de celle du chiffrement (pseudo-code 3.3). Certaines propriétés du l'AES permettent d'implémenter une routine de déchiffrement qui respecte la séquence de transformations de la routine de chiffrement, et qui nécessite une modification de l'algorithme de cadencement de clé.
Les deux propriétés qui permettent ce changement sont les suivantes [66]:

1. les transformations SubBytes() et ShiftRows() commutent. Il en est de même pour leurs inverses InvSubBytes() et InvShiftRows().

2. les transformations MixColumns() et InvMixColumns() sont linéaires :

InvMixColumns(state XOR Round Key) =
InvMixColumns(state) XOR InvMixColumns(Round Key)

Dans le déchiffrement équivalent, l'ordre des transformations InvSubBytes() et InvShiftRows() peut être inversé grâce aux propriétés cités précédemment. L'ordre des transformations AddRoundKey() et InvMixColumns() peut aussi être inversé si on applique une modification dans l'algorithme de cadencement de clé en utilisant InvMixColumns(). Les premiers et derniers Nb mots de la clé étendue n'ont pas modifié.

Le pseudo-code du déchiffrement équivalent est donné en 3.12.

```
EqInvCipher(byte in[4*Nb], byte out[4*Nb], word dw[Nb*(Nr+1)])
début
 1       byte state[4,Nb]
 2
 3       state = in      // l'entrée est copiée dans le bloc
 4
 5       AddRoundKey(state, dw[Nr*Nb, (Nr+1)*Nb-1])
 6
 7       pour round = Nr-1 à -1 par pas de -1 faire
 8            InvSubBytes(state)
 9            InvShiftRows(state)
10            InvMixColumns(state)
11            AddRoundKey(state,dw[round*Nb, (round+1)*Nb-1])
12       finpour
13
14       InvSubBytes(state)
15       InvShiftRows(state)
16       AddRoundKey(state,dw[0, Nb-1])
17
18       out = state   // copie du bloc dans la sortie
fin

ceci implique l'aménagement suivant dans la routine d'expansion
de la clé :

        pour i = 0 à (Nr+1*Nb-1) par pas de 1 faire
            dw[i]=w[i]
        fin pour
        pour round = 1 à Nr-1 par pas de 1 faire
               InvMixColumns(dw[round*Nb, (round+1)*Nb-1])
        fin pour
```

Figure 3.12. *Pseudo-code - déchiffrement équivalent* [66]

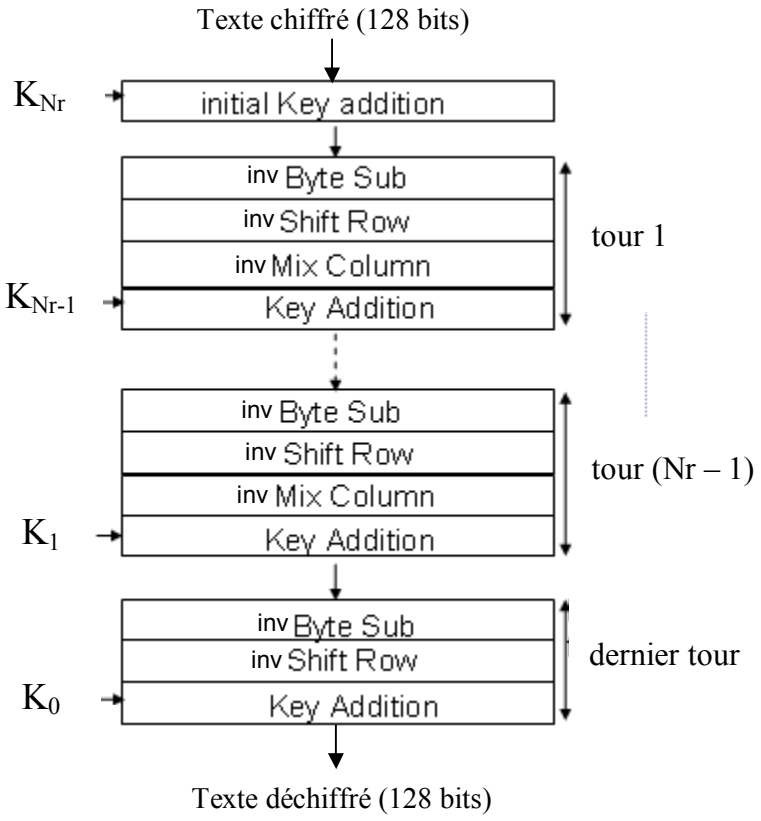


Figure 3.13. *Déchiffrement équivalent*

3.5.3 La génération des sous-clés

Cette opération permet de générer les sous-clés de chiffrement à partir de la clé secrète K. Le cadencement de clé se déroule en deux étapes : d'abord l'expansion de clé, ensuite la sélection des sous-clés générées à partir de la clé étendue.

La routine d'expansion de clé consiste à générer $Nb*(Nr + 1)$ mots nécessaires à chaque tour dans le chiffrement et le déchiffrement à partir de la clé secrète K. La clé étendue est représentée par un tableau de mots de 4 octets, noté w[i], tels que $0 \leq i < Nb*(Nr + 1)$.

La fonction **SubWord()** prend en entrée un mot de 4 octets et substitue chaque octet par sa valeur correspondante dans la table S-Box. la fonction **RotWord()** prend en entrée un mot de 4 octets $[a_0, a_1, a_2, a_3]$ et lui applique la permutation circulaire pour retourner le mot $[a_1, a_2, a_3, a_0]$. Le tableau **Rcon[]** est construit ainsi [66]:

$$\text{Rcon}[j] = [(0x02)^{j-1}, 0x00, 0x00, 0x00] \text{ , } pour \quad j \geq 1 \qquad (3.5.10)$$

On peut constater que les *Nk* premiers mots de la clé étendue sont constitué de la clé de chiffrement K. Chaque mot w[i] qui les suive, est égal au XOR du mot précédent w[i-1] avec le mot w[i-*Nk*]. Pour les mots dans des positions qui sont un multiple de *Nk*, on applique tous d'abord à w[i-1] la fonction RotWord(), suivi par l'application de la fonction SubWord(), suivi par un XOR avec le tableau Rcon[j] avant de faire le XOR avec w[i-*Nk*].

Notons que la routine d'expansion de la clé lorsque la taille de la clé K est 256 bits (*Nk* = 8) est légèrement différente de celle de 128 et 192 bits. Dans ce cas, si i-4 est un multiple de *Nk*, SubWord() est appliquée à w[i-1] avant le XOR avec w[i-Nk].

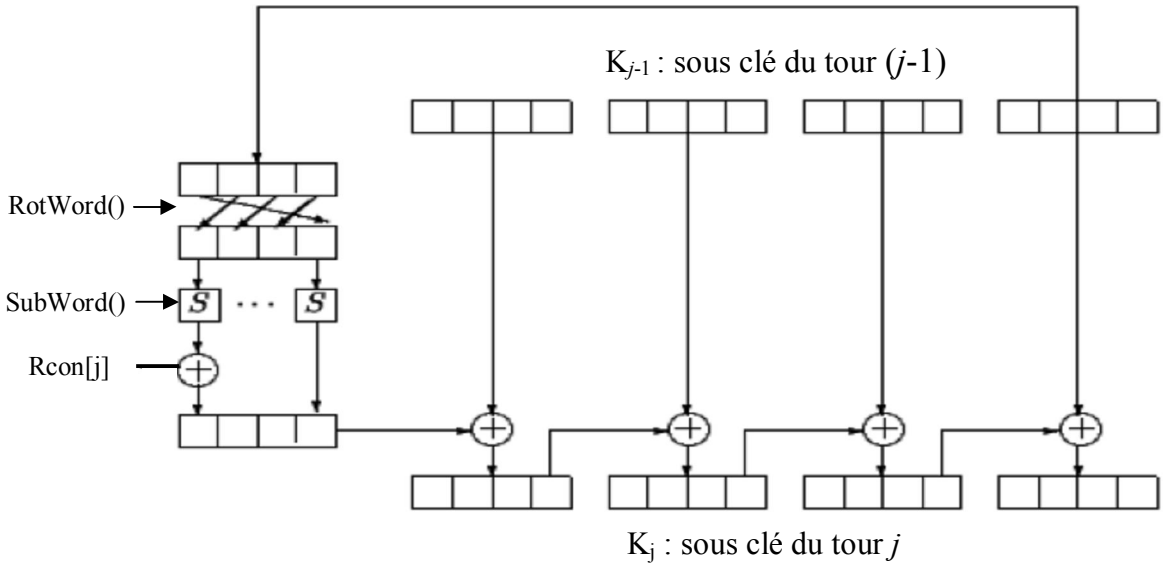


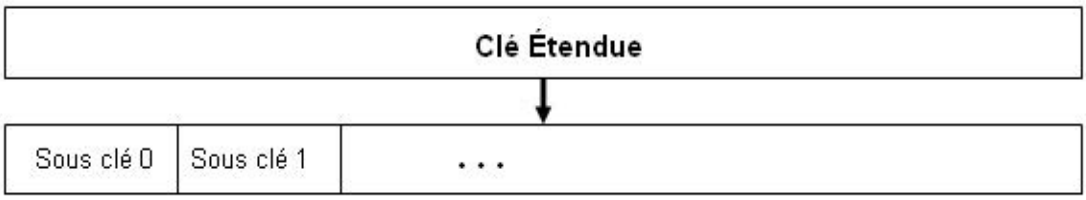
Figure 3.14. *Algorithme d'expansion de clé* [67]

Le pseudo-code d'expansion de clé est donné en 3.15.

```
KeyExpansion (byte key[4*Nk], word w[Nb * (Nr+1)], Nk)
début
   word temp

   i = 0;

   tant que (i < Nk) faire
       w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
       i = i+1
   fin tant que

       i = Nk

   tant que (i < Nb * (Nr+1)) faire
       tmp = w[i-1]
       si (i mod Nk = 0) alors
          tmp = SubWord(RotWord(tmp)) Xor Rcon[i/Nk]
       sinon si (Nk > 6 et i mod Nk = 4) alors
          tmp = SubWord(tmp)
       fin si
       w[i] = w[i-Nk] Xor tmp
       i = i + 1
   fin tant que
fin
```

Figure 3.15. *Pseudo-code – expansion de la clé* [66]

Pour la sélection des sous-clés, la sous-clé i, de taille Nb est simplement donnée par les mots W[Nb*i] à W[Nb*(i+1)−1]. On obtient par exemple le schéma suivant :



3.6 Cryptanalyse de l'AES

L'AES n'a pour l'instant pas été cassé et la recherche exhaustive demeure la seule solution.

a. Attaques sur des versions simplifiées

Niels Ferguson et son équipe ont proposé en 2000 une attaque sur une version à 7 tours de l'AES 128 bits [54]. Une attaque similaire casse un AES de 192 ou 256 bits contenant 8 tours. Un AES de 256 bits peut être cassé s'il est réduit à 9 tours avec une contrainte supplémentaire. En effet, cette dernière attaque repose sur le principe des « related-keys » (clés apparentées), c'est une attaque où la clé demeure secrète mais l'attaquant peut légèrement modifier la clé, chiffrer des textes à sa guise, et regarder comment la sortie de l'AES se comporte.

b. Attaques sur la version complète

Certains groupes ont affirmé avoir cassé l'AES complet, mais après vérification par la communauté scientifique, il s'avérait que toutes ces méthodes étaient erronées. Cependant, plusieurs chercheurs ont essayé des attaques algébriques, notamment l'attaque XL et une version améliorée, la XSL [54]. Ces attaques leur efficacité n'a pas encore été pleinement démontrée, de plus, elles sont impraticables car le XSL demande au moins 2^{87} opérations voire 2^{100} dans certains cas. Le principe est d'établir les équations qui lient les entrées aux sorties et de résoudre ce système qui ne comporte pas moins de 8000 inconnues et 1600 équations pour 128 bits. La solution de ce système reste pour l'instant impossible à déterminer. Donc, l'AES est considéré comme sûr en l'absence d'une preuve formelle sur l'efficacité d'attaques similaires au XSL.

c. Attaques par cryptanalyse différentielle (DC)

L'attaquant choisit des textes clairs présentant une différence fixe, calcule les chiffrés par l'AES et leurs différences puis assigne des probabilités à certains types de clés. Plus le nombre d'essais augmente, plus la probabilité de la bonne clé est forte.

Dans le cas du DES, cette attaque nécessite 2^{47} textes clairs et 2^{47} chiffrements pour retrouver la clé, mais l'AES est lui résistant à ce type d'attaque [68].

d. Attaques par cryptanalyse linéaire (LC)

L'attaquant utilise des approximations linéaires pour décrire les opérations conduisant au chiffré. Plus le nombre d'essais augmente, plus la probabilité de la bonne clé est forte.

Cette attaque ne nécessite que 2^{43} textes clairs et 2^{43} chiffrements pour retrouver une clé DES, mais pour l'AES, il est lui résistant à ce type d'attaque [68].

e. Attaques par le cache

Les implémentations sous forme logicielle d'AES sont particulièrement vulnérables aux attaques dites *par le cache* [61]. L'attaquant observe simplement le comportement et le *timing* d'un de ses *process* (pas directement lié à AES) qui tourne en même temps et sur le même processeur que le *process* du logiciel AES attaqué. Il est montré que les données rassemblées peuvent conduire à des informations concernant la clé secrète AES, qui peut ensuite être connue en une fraction de seconde. Cette attaque, cependant, ne réclame aucun privilège par rapport au process AES, il suffit d'avoir un accès simultané au processeur sur lequel le logiciel AES s'exécute.

Pendant la *compétition* AES, le fait que le cache soit une ressource partagée qui permet des fuites d'information entre les process était soit ignoré soit considéré hors de propos. Actuellement, tout logiciel incluant un process AES est à risque s'il s'exécute sur un processeur qui peut être partagé. C'est le cas des serveurs et de tout ordinateur sans un contrôle d'accès approprié. Des mesures de protection ont été proposées pour faire échouer les attaques par le cache, mais elles ne sont pas fréquemment appliquées et ne font pas partie de l'approche standard de la sécurité informatique.

3.7 Conclusion

A l'heure actuelle l'utilisation du DES est simplement déconseillée du fait de la puissance de calcul offerte par les ordinateurs les plus récents. Toutefois le triple DES, version améliorée du DES, permet de se prémunir des attaques les plus classiques et d'apporter un niveau de sécurité acceptable. Le choix de l'AES reste néanmoins le meilleur, de telles sortes qu'il est un standard important et l'utilisation et la compréhension qu'il permet, accroître sensiblement la fiabilité et la sécurité de nos systèmes informatiques. Dans ce chapitre, nous avons détaillé ce standard qui est, d'après cette étude, incassable dans le sens où aucune attaque connue de cryptanalyse ne peut déchiffrer le texte chiffré sans utiliser une recherche par force brute, dans l'attente d'un remplaçant ou d'une méthode d'attaque efficace qui précipiterait sa remise en cause.

CHAPITRE 4

Réalisation d'un système de chiffrement de PDA et téléphone mobile basé sur l'algorithme symétrique AES

CHAPITRE 4

Réalisation d'un système de chiffrement de PDA et téléphone mobile basé sur l'algorithme symétrique AES

4.1 Introduction

PDA et téléphones mobiles sont largement disponibles et sont entrain de devenir la «norme» chez les gens d'affaires et plus généralement avec tous ceux qui voyagent fréquemment et qui ont besoin d'informations en mouvement. Cette information peut être composée de données sensibles, que l'utilisateur du PDA ou téléphone ne veut pas qu'elles soient disponibles à n'importe qui autre que lui-même. Le risque de vol des informations privées en cas de perte ou de vol de dispositif peut compromettre les données confidentielles.

Dans ce chapitre, nous allons développer une application basée sur l'algorithme AES qui permet le chiffrement et le déchiffrement des données utilisateur (contacts, rendez-vous, e-mails, images, vidéos, audio,…) sur un PDA/téléphone moderne.

Précédemment un tel système n'aurait pas été faisable sur un PDA/téléphone en tant que techniques de chiffrement, car ceci nécessite de grandes durées de traitement. Ces dispositifs mobiles de nos jours tirent bénéfice de la loi de Moore, qui stipule que les processeurs développés sont plus rapides, de plus petites tailles, et consomment moins d'énergie - ainsi leurs insertion dans le PDA/téléphone permettant d'utiliser des algorithmes nécessitant des calculs plus complexes, tels que les algorithmes de chiffrement et de déchiffrement.

Le système est destiné à toute personne ayant besoin de stocker des données sensibles sur un PDA/téléphone en lui assurant une sécurité pour ces données. Le système proposé tient compte les restrictions de la limitation des ressources d'un PDA/téléphone (tels que le stockage), ainsi, il fonctionne sur n'importe quel dispositif de poche de Microsoft. Pour tester ce système, nous utilisons des PDA séries HP (iPAQ) h5550 et le téléphone mobile de type Samsung SGH-i200.

4.2 Evolution du système proposé

4.2.1 Motivations

En quelques années, la multiplication et la diversité des dispositifs mobiles dans nos vies quotidiennes ont entraîné de nouveaux défis pour les développeurs. En plus de vouloir sécuriser le contenu de leurs données par un système de chiffrement puissant, le développement des applications sur ces dispositifs mobiles est fortement contraint par les limitations de ces derniers. Ces limitations incluent [70] :

- Une mémoire limitée dans la plupart des dispositifs mobiles (RAM utilisée par le système, Flash ROM stocke des données d'utilisateur, des programmes et autres fichiers) ;
- Puissance de traitement limitée ;
- Petites claviers numériques ;
- Ecran de petite taille ;
- Consommation d'énergie de processeur.

4.2.2 Solutions possibles

4.2.2.1 *Solutions Java*

Les solutions de Java sont très utilisées en raison des possibilités qui donnent aux programmeurs pour résoudre tous problème de programmation.

a) Sun Microsystems Java	http://java.sun.com

J2ME PP (Java 2 Micro Edition Personal Profil) [71][73] est destiné aux petits dispositifs pour fournir des environnements d'exécution des applications Java et il est toujours en cours de développement par Sun. Il est gratuit, et il a une portabilité multiplateforme. Une application Java écrite en utilisant la J2ME Wireless Toolkit (qui fournit l'environnement d'émulation, de la documentation et des exemples de développement des applications) fonctionne sur n'importe quel dispositif mobile qui dispose d'une machine virtuelle [71].

En conclusion; cette solution n'est pas encore complète et ne fournit pas la fiabilité et la vitesse d'exécution qui seraient très utiles dans ce genre d'environnement, en plus , un ralentissement du développement d'applications , et le ralentissement de ses

performances [71]. *La disponibilité d'outils de développement qui aide à accélérer le développement et le débogage des installations est un peu rare. Ainsi, le J2ME exige aussi une machine virtuelle sur l'appareil de développement, ce qui va ralentir notre application et qui consomme les ressources système déjà limitées* [71]. *Ceci rendre cette solution inadéquate pour le développement de notre système.*

b) Jeode	http://www.embedded.com/192200699?_requestid=346522

Jeode est une solution Java de marché à grand public pour le mobile et les dispositifs existants. L'environnement d'exécution de Jeode™ offre les outils suivants [72] :

- Accélération de la performance, la robustesse, et l'utilisation efficace de la mémoire;
- Une implémentation complète des caractéristiques de PersonalJava et d'EmbeddedJava;
- Support, formation, personnalisation, et instruments de développement.

Jeode ajoute simplement une machine virtuelle à l'iPAQ lui permettant d'exécuter des applications de Java (basée sur le profil personnel, le profil personnel de base et le profil de fondation (CDC)). L'application serait codée avec Java et puis compilée, ensuite convertie avec le logiciel Jeode pour exécuter correctement sur l'iPAQ [72].
Jeode fonctionne avec plusieurs sous-ensembles de l'API Java (les profils ci-dessus).

En conclusion; dans cette solution, premièrement il n'y a aucun outil de développement suggéré, deuxièmement chaque utilisateur de l'application doit payer 50$ pour chaque exécution [73]. *L'avantage de cette solution par rapport à la solution Sun est son fonctionnement sur le système d'exploitation par défaut de dispositif. Cette solution est non adéquate pour notre application.*

c) SuperWaba	http://www.superwaba.com.br/en/default.asp http://www.pocketpcfreeware.com/index.php?soft=785

La Machine Virtuelle SuperWaba est parmi l'une des plus puissantes machines virtuelles Java pour les dispositifs mobiles. Initialement développée pour le système PalmOS, elle a récemment été adaptée au Pocket-PC [87]. SuperWaba permet aux programmeurs Java de développer des applications dans toute IDE Java pour Palm, Pocket PC, Win CE et

Symbian [74]. Ainsi, comparée à d'autres plates-formes de développement mobile comme Compact Framework .NET (Microsoft) et J2ME (Sun Microsystems), les applications SuperWaba possède une bonne performance.

En conclusion; cette solution est open source, gratuite à télécharger *et est très attrayante dans ce sens, ainsi, elle a beaucoup d'intérêt de développement. Mais pour exécuter une application finale sur un PDA/téléphone mobile, la machine virtuelle devrait être installée, en plus il y a toujours le problème avec des outils de développement.*

4.2.2.2 *Solutions de Microsoft*

Microsoft a fourni deux outils différents disponibles aux réalisateurs voulant développer des applications sur les dispositifs Pocket PC, Windows CE, et le Windows Mobile.

a) Studio Visuel .NET 2008	http://msdn.microsoft.com/vstudio/
Type de solution: sélection de langage de programmation & outils de développement	

Studio Visuel .NET 2008 est un outil de développement orienté vers le déploiement rapide des applications pour les systèmes d'exploitation de Microsoft [75]. Il permet aux développeurs de développer et de déboguer des applications qui utilisent le .NET Compact Framework qui est un sous-ensemble du .NET Framework. Il existe aussi d'autres plates-formes comme le Studio Visuel.NET 2003 et 2005.

Pendant les tests, le Studio Visuel .NET nous permet d'exécuter les applications sur un émulateur, ou un dispositif réel. Ainsi, il nous permet de les développer en utilisant les langages C#, C++, et Visual Basic.

L'installation sur le dispositif d'une application créée dans cette plate-forme n'exige aucun fichier supplémentaire, et aucune machine virtuelle.

En conclusion; cette solution est la première solution de bout en bout, elle donne une exécution rapide, un ensemble complet d'outils de développement, et un choix du langage de programmation [75]. *Le coût du développement est faible et l'utilisateur ne nécessite pas une machine virtuelle pour exécuter son application sur le dispositif.*

b) Embedded Visual Tools Version 3	http://msdn.microsoft.com/en-us/magazine/cc301473.aspx

Embedded Visual Tools Version 3 est un outil de développement similaire à celui du Studio Visuel 6. Cet outil est totalement gratuit, il est composé d'un environnement de développement "Embedded Visual Basic" et "Embedded Visual C++", et de 3 SDK (System Development Kits), celui des Pocket PC, des Handheld PC et celui des Palm Size PC [76].

En conclusion; cette solution offre des avantages similaires au Studio Visuel. Toutefois, elle ne fournit pas de support pour le C# [77]*, un langage relativement nouveau avec des similitudes au Java. Cette solution est gratuite, mais elle manque de certains supports et outils de développement comparativement à Studio Visuel.*

4.2.2.3 *Autres Solutions*

a) AppForge MobileVB	http://www.appforge.com/products/mobilevb/index.html

L'outil de développement AppForge MobileVB permet de développer rapidement des applications en Visuel Basic 6.0 et Visuel Basic .Net, qui fonctionnera sur les ordinateurs de poche, les mobiles et les appareils sans fil tournant sur les systèmes d'exploitation Palm OS, Pocket PC et Symbian OS [77].

MobileVB utilise plus de 30 commandes standard, ainsi que des modules et des bibliothèques pour la synchronisation de base de données. Une fois installé sur l'ordinateur de bureau, les applications MobileVB peuvent être développées et testées au sein de l'IDE (Integrated Development Environment) Visual Basic. Une fois ces applications compilées, elles seront transférées à un dispositif mobile [69].

En conclusion; cette solution n'est pas gratuite et coûte trop chère [86] [71]. *Ainsi, le fonctionnement de l'application nécessite l'installation d'un RunTime "AppForge Booster" compatible avec le dispositif. Ce qui signifie en ce qui concerne les PocketPC que certains dispositifs ne sont pas supportés* [69].

b) Embedded QNX on Intelligent Platforms (iPAQ/Pocket/dispositifs personnels)	http://eqip.openqnx.com/ http://www.qnx.com/

QNX [80] est un système d'exploitation temps réel (RTOS). Il est conçu pour le marché des systèmes embarqués. Ce système d'exploitation est disponible en plusieurs versions - les plus populaires est la plate-forme en temps réel QNX RTP. Il peut travailler non seulement comme un système embarqué, mais également comme un serveur ou un système d'exploitation de l'ordinateur de bureau [80].

Le système QNX est comme le système Linux où les utilisateurs peuvent développer des applications en utilisant Java et d'autres langages développées spécifiquement pour le QNX RTP.

En conclusion; cette solution ne fournit pas la portabilité immédiate. L'installation du système QNX nécessite l'effacement du chargeur d'amorçage (bootloader) et du système d'exploitation du dispositif iPAQ [90].*Une fois les nouveaux systèmes d'exploitation et bootloader sont installés, l'application peut être développés en utilisant l'environnement de développement QNX.*

c) Linux Familial pour l'iPAQ	http://familiar.handhelds.org/ http://www.handhelds.org

Le projet familial est composé d'un groupe de développeurs souples qui contribuent tous à créer la prochaine génération de système d'exploitation de PDA. Actuellement, la plupart du temps de développement est réservé pour produire une distribution stable et complète de Linux pour la série HP iPAQ des ordinateurs de poche.

En conclusion; cette solution est actuellement en cours de développement, et son inconvénient est qu'elle ne fournit pas la portabilité immédiate. Afin d'installer linux, il est nécessaire de changer le bootloader du iPAQ [89][52].

4.2.3 Evaluation

Après une étude des différentes solutions, nous pouvons dire que les solutions Java nécessiteraient une machine virtuelle à être installée, et la plupart des machines virtuelles ne fonctionnent pas convenablement sur les dispositifs mobiles en raison de la limitation de la puissance du processeur et de la capacité mémoire.
Les autres solutions, telles que Linux et QNX ne fournissent pas la portabilité immédiate et ne sont pas en conformité avec les dispositifs standard et exigent une expertise pour être configurée.

La solution Microsoft est une solution de bout en bout pour le développement et les tests des applications et elle ne nécessite pas une machine virtuelle. Les environnements sont robustes et complètes et ont un support logiciel complet.

4.2.4 Le système proposé

Après avoir étudié les différentes solutions, nous pouvons dire que la solution la plus fiable est la solution Microsoft. Pour le développement de notre application, nous allons utiliser la solution Studio Visuel de Microsoft, car cette plate-forme offre des excellents équipements de teste, un ensemble complet des outils de développement, et un choix du langage de programmation. Le langage de programmation utilisé est le C# qui a une structure similaire à Java et il est très bien documenté, comparativement à Visual Basic qui ne fournit pas la performance requise (temps d'exécution et consommation mémoire) et à Embedded C++ (langage de Embedded visual version 3) qui est excessive pour ce système.

Le but de notre application sera de chiffrer et de déchiffrer des données sélectionnées par l'utilisateur (contacts, rendez-vous, emails, images, vidéos, audio…) sur le PDA/téléphone mobile.

Les avantages du système proposé sont :

- Le système est portable, peut être installé sur n'importe quel nombre de dispositifs de PDA ou téléphone mobile ;
- Des grandes clés de chiffrement ne devront pas être rappelés à chaque fois que l'application soit exécuté ;
- Le système pourrait importer / exporter les valeurs de clé de chiffrement, ou de la taper directement ;

- Les données chiffrées sur un PDA/téléphone mobile spécifique peuvent être utilisés sur d'autres dispositifs mobiles, en les envoyant par courrier électronique ou par Bluetooth ;
- Les données peuvent être stockées sur la carte mémoire amovible sur le dispositif et utilisé sur les ordinateurs de bureaux et autres PDA et téléphone mobile.

4.3 Analyse des besoins

Le système proposé exige deux types des paramètres. Les paramètres fonctionnels définissent la fonctionnalité qui doit être fournie par le système, et les paramètres non fonctionnels définissent les propriétés et les contraintes du système.

4.3.1 Les paramètres fonctionnels

Fonction	Connexion
Description	Cette fonction permet d'accéder au menu principal du système et de manipuler ses autres fonctions
Entrées	Le nom d'utilisateur et le mot de passe
Sorties	Le menu principal du système
Exigences	La saisie correcte du nom d'utilisateur et du mot de passe
Effets secondaires	Aucun

Tableau 4.1. *Les paramètres fonctionnels de la fonction 'Connexion'*

Fonction	Déconnexion
Description	Mettre l'utilisateur hors connexion
Entrées	Le bouton "quitter l'application" sur le menu principal
Sorties	L'utilisateur est déconnecté
Exigences	L'utilisateur est connecté
Effets secondaires	Aucun

Tableau 4.2. *Les paramètres fonctionnels de la fonction 'Déconnexion'*

Fonction	Chiffrer un nouveau fichier
Description	Sélectionner un nouveau fichier depuis le dispositif afin de le chiffrer, ainsi que son enregistrement approprié après le chiffrement
Entrées	Sélectionner le fichier à chiffrer depuis le dispositif
Sorties	Le fichier est chiffré avec un temps de chiffrement X millisecondes
Exigences	L'utilisateur est connecté
Effets secondaires	Le fichier original est supprimé

Tableau 4.3. *Les paramètres fonctionnels de la fonction 'Chiffrer un nouveau fichier'*

Fonction	Déchiffrer un fichier chiffré
Description	Déchiffrer un fichier déjà chiffré, puis faire le choix de son enregistrement approprié après le déchiffrement
Entrées	Sélectionner le fichier à déchiffrer depuis le dispositif
Sorties	Le fichier est déchiffré avec un temps de déchiffrement X millisecondes
Exigences	l'utilisateur est connecté
Effets secondaires	Le fichier chiffré peu être supprimé ou non

Tableau 4.4. *Les paramètres fonctionnels de la fonction 'Déchiffrer un fichier chiffré'*

Fonction	Changement du mot de passe du système
Description	Permet à l'utilisateur de changer le mot de passe du système
Entrées	L'ancien et le nouveau mot de passe
Sorties	Confirmation de changement du mot de passe
Exigences	L'utilisateur est connecté, et l'ancien mot de passe doit être saisi correctement dans le système
Effets secondaires	Aucun

Tableau 4.5. *Les paramètres fonctionnels de la fonction 'Changement du mot de passe'*

Fonction	Visualiser les fichiers chiffrés
Description	Permet à l'utilisateur de visualiser tous les fichiers déjà chiffrés
Entrées	L'utilisateur sélectionne le bouton "visualiser les fichiers chiffrés" sur le menu de l'application
Sorties	Tous les noms et les emplacements des fichiers chiffrés sont affichés
Exigences	Aucun
Effets secondaires	Aucun

Tableau 4.6. *Les paramètres fonctionnels de la fonction 'Visualiser les fichiers chiffrés'*

Fonction	Changement de la clé de chiffrement / déchiffrement
Description	Permet à l'utilisateur de changer la clé de chiffrement
Entrées	Nouvelle clé de chiffrement
Sorties	Affichage du message : Voulez vous changer votre clé de chiffrement ?, si oui, ceci rendre les fichiers déjà chiffrés inutilisables. Veuillez saisir votre mot de passe pour confirmer cette modification
Exigences	Le correct mot de passe doit être saisi dans le système
Effets secondaires	Les fichiers déjà chiffrés seront inutilisables s'ils ont été chiffrés avec l'ancienne clé.

Tableau 4.7. *Les paramètres fonctionnels de la fonction 'Changement de la clé de chiffrement'*

4.3.2 Les paramètres non fonctionnels

Les contraintes l'application développée sont :

- L'application devrait être compatible avec le système d'exploitation Pocket PC 2003 et Windows Mobile de Microsoft ;
- Son interface devrait être utilisable avec la résolution du dispositif (240*320 Pixel).

Pour les propriétés de l'application on peut citer :

- L'application est portable, ainsi que les fichiers chiffrés ;
- Elle est facile à utiliser ;
- Il n'y a aucune perte de données dans le cas où un fichier ne pourrait pas être chiffré ;
- L'application devrait être chargée au cours de 15 secondes.

4.4 Spécifications fonctionnelles

La figure suivante représente le schéma général du système proposé, où on va expliquer ses fonctions par des diagrammes qui nous montrent les interactions entre l'utilisateur et ce système :

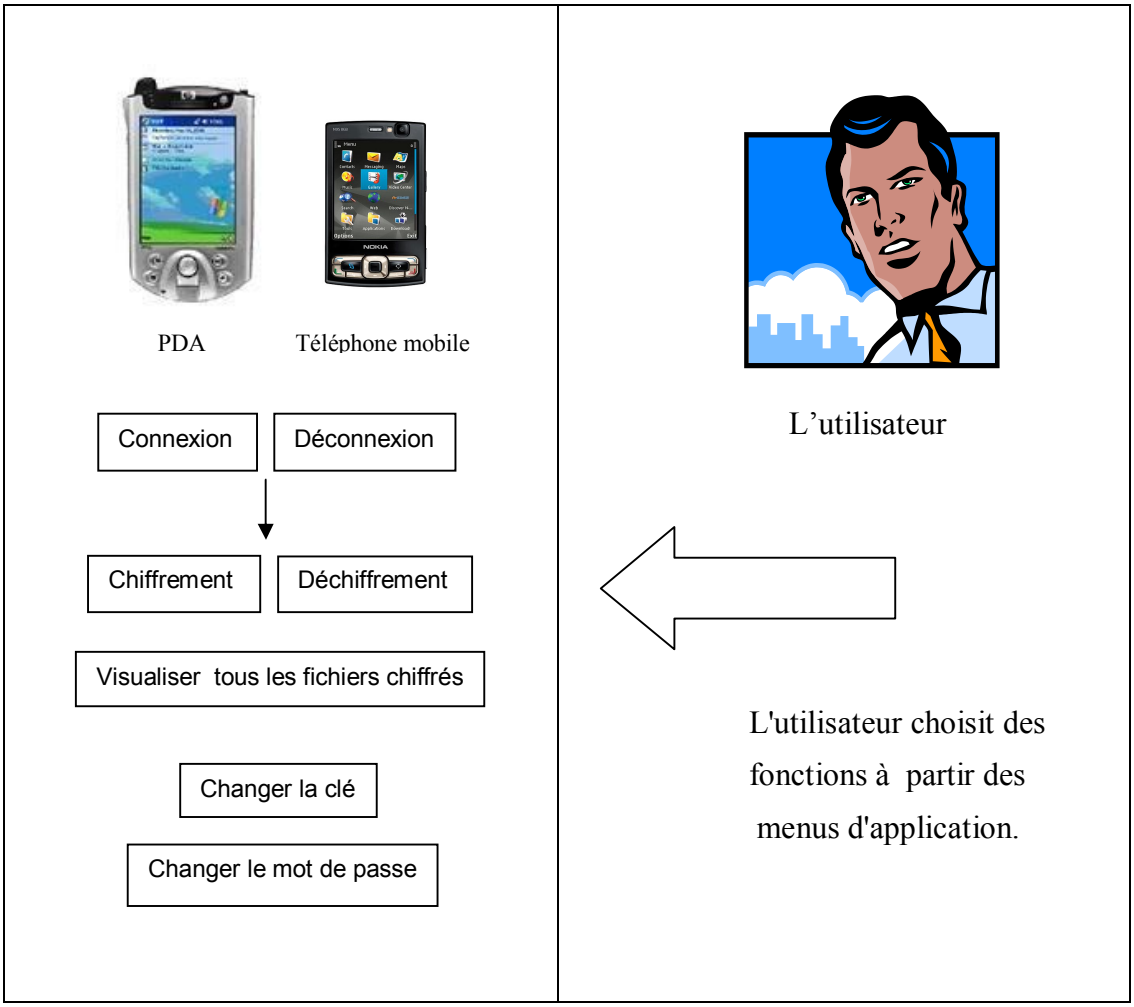


Figure 4.1. *Schéma général du système proposé*

- L'application aura besoin de stocker les données suivantes sur le dispositif : Le nom utilisateur, le mot de passe, la clé de chiffrement/déchiffrement, ainsi que le répertoire pour sauvegarder les fichiers chiffrés.

4.4.1 Diagramme de la fonction 'Connexion'

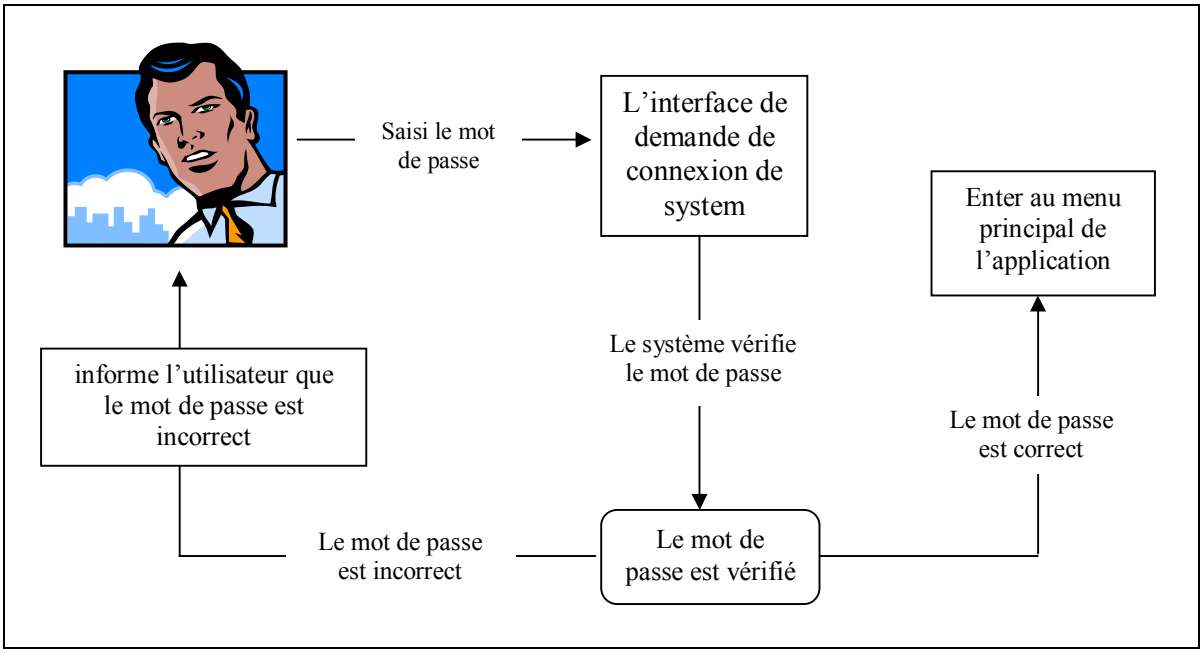


Figure 4.2. *Diagramme de la fonction 'Connexion'*

- Le flux des événements pour le cas d'utilisation de la fonction **Connexion** (Login) :

Flux principal :

1. *L'utilisateur saisie le mot de passe*
2. *Le système vérifie le mot de passe*
3. *Le mot de passe est correct*
4. *Entrer au menu principal de l'application*

Flux alternatif :

1. *L'utilisateur saisie le mot de passe*
2. *Le système vérifie le mot de passe*
3. *Le mot de passe est incorrect*
4. *L'utilisateur est informé que le mot de passe est incorrect*

4.4.2 Diagramme de la fonction 'Déconnexion'

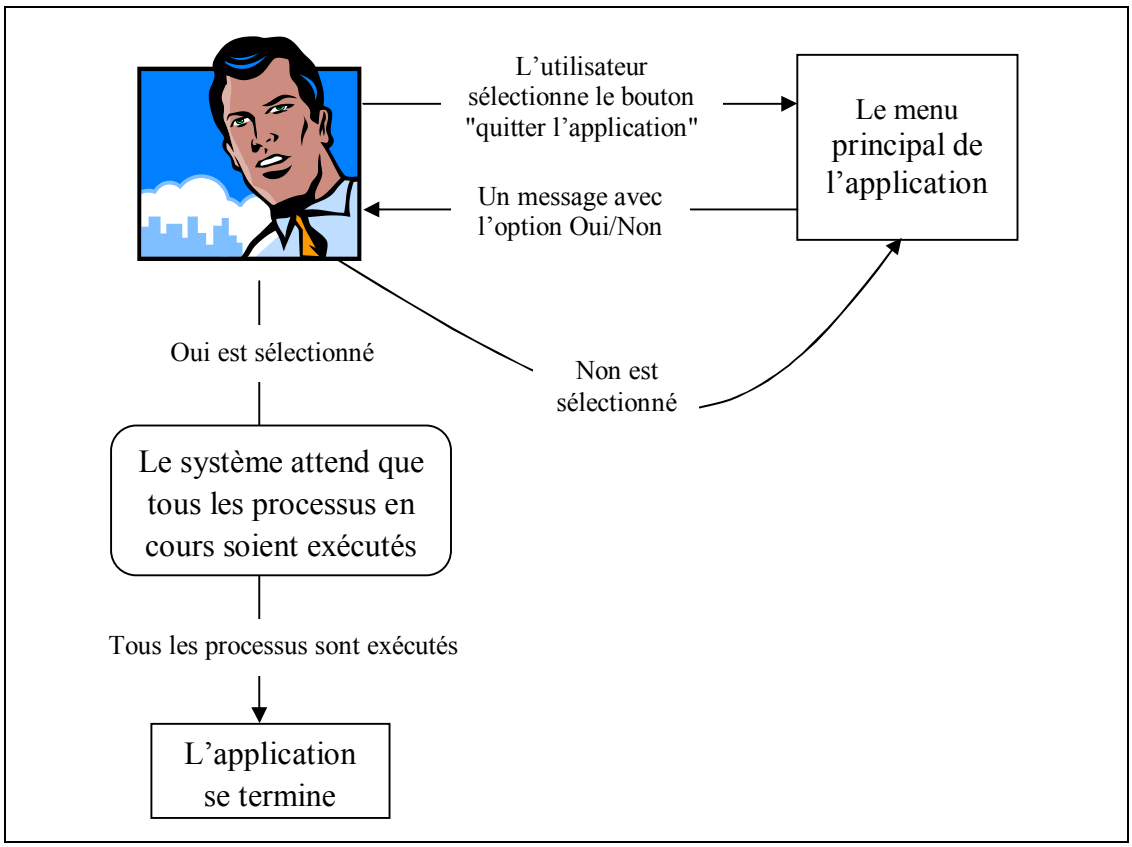


Figure 4.3. *Diagramme de la fonction 'Déconnexion'*

Le flux des événements pour le cas d'utilisation de la fonction **Déconnexion** (Logout) :

Flux principal :

1. *L'utilisateur sélectionne le bouton "quitter l'application"*
2. *L'utilisateur choisit l'option "oui"*
3. *Le système attend que tous les processus en cours soient exécutés*
4. *L'application se termine*

Flux alternatif :

1. *L'utilisateur sélectionne le bouton "quitter l'application"*
2. *L'utilisateur choisit l'option "non"*
3. *Le menu principal de l'application est affiché*

4.4.3 Diagramme de la fonction 'Chiffrer un nouveau fichier'

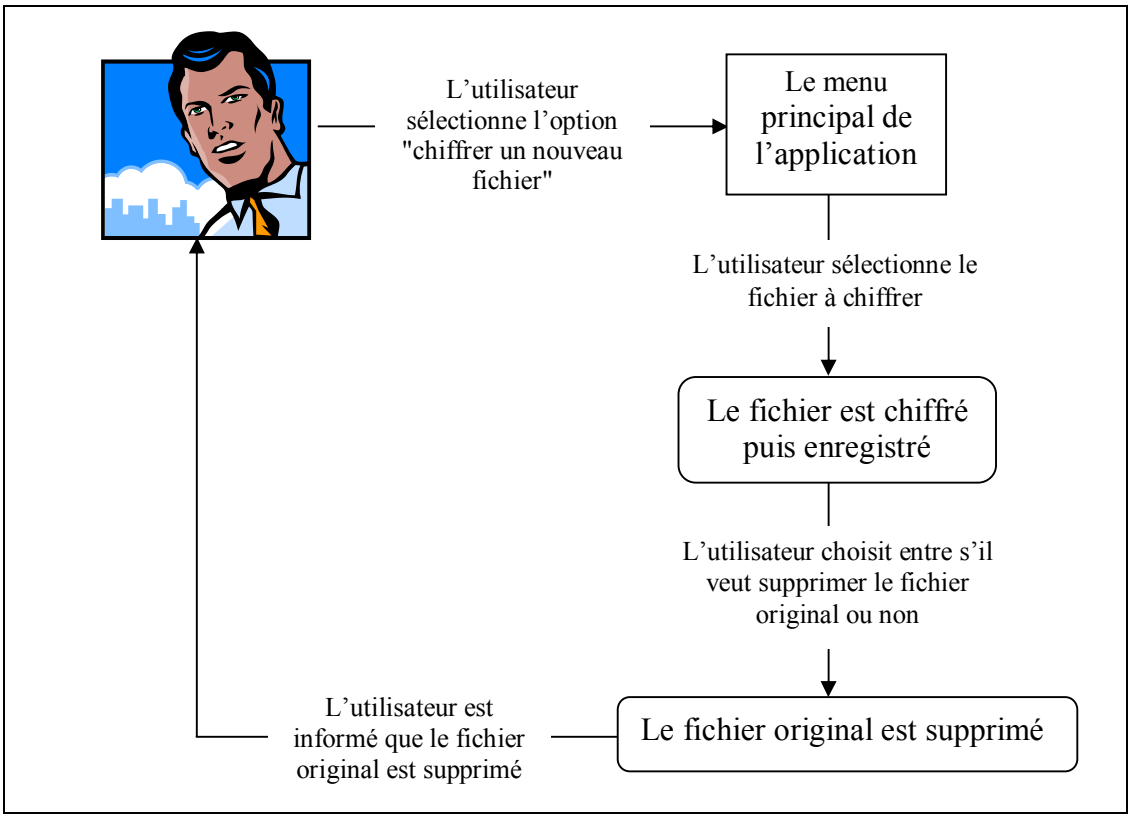


Figure 4.4. *Diagramme de la fonction 'Chiffrer un nouveau fichier'*

Le flux des événements pour le cas d'utilisation de la fonction ***Chiffrer un nouveau fichier*** :

Flux principal :

1. *L'utilisateur sélectionne l'option "chiffrer un nouveau fichier"*
2. *Le fichier est localisé*
3. *Le fichier est chiffré*
4. *Le temps de chiffrement de fichier est affiché*
5. *Le fichier chiffré est enregistré*
6. *L'utilisateur choisit entre s'il veut supprimer le fichier original ou non*
7. *Le fichier original est supprimé*
8. *L'utilisateur est informé que le fichier original est supprimé*

Flux alternatif :

1. *L'utilisateur sélectionne l'option "chiffrer un nouveau fichier"*
2. *Le fichier est localisé*
3. *L'utilisateur sélectionne "Annuler"*
4. *L'utilisateur retourne au menu principal de l'application*

4.4.4 Diagramme de la fonction 'Déchiffrer un fichier chiffré'

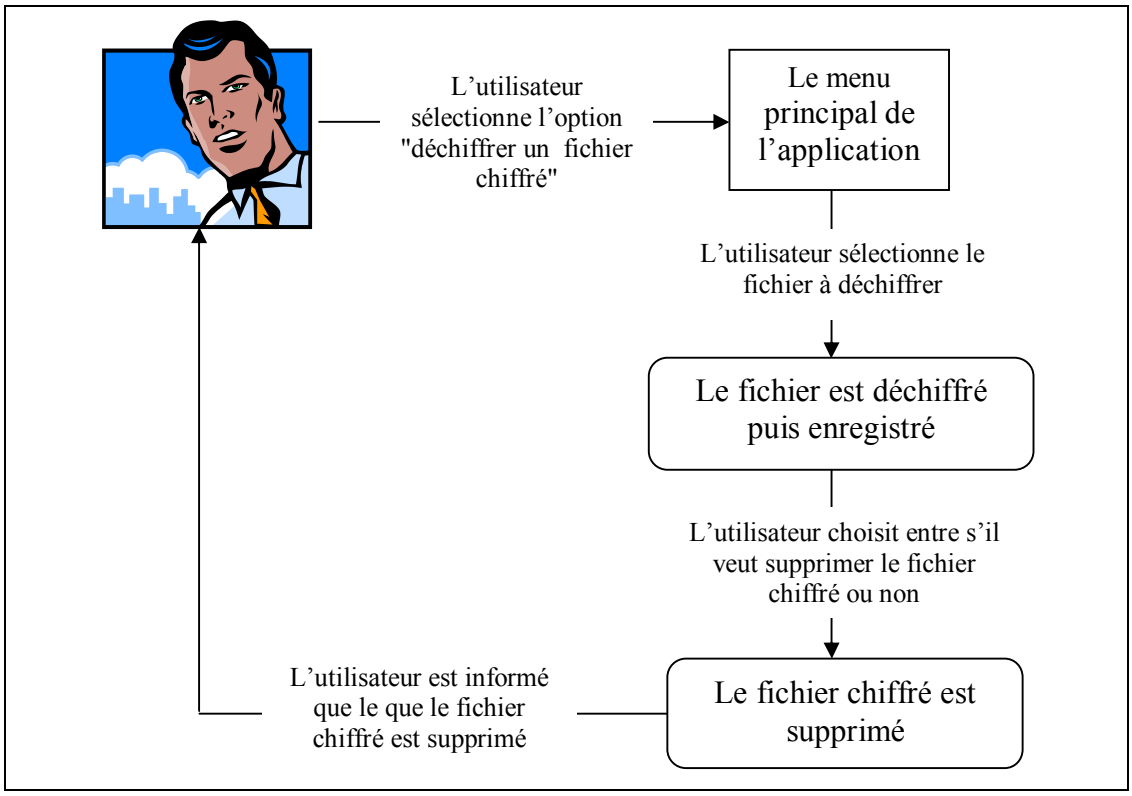


Figure 4.5. *Diagramme de la fonction 'Déchiffrer un fichier chiffré'*

Le flux des événements pour le cas d'utilisation de la fonction ***Déchiffrer un fichier chiffré*** :

Flux principal :

1. *L'utilisateur sélectionne l'option "déchiffrer un fichier chiffré"*
2. *Le fichier est localisé*
3. *Le fichier est déchiffré*
4. *Le temps de déchiffrement de fichier est affiché*
5. *Le fichier déchiffré est enregistré*
6. *L'utilisateur choisit entre s'il veut supprimer le fichier chiffré ou non*
7. *Le fichier chiffré est supprimé*
8. *L'utilisateur est informé que le fichier chiffré est supprimé*

Flux alternatif :

1. *L'utilisateur sélectionne l'option "déchiffrer un fichier chiffré"*
2. *Le fichier est localisé*
3. *L'utilisateur sélectionne "Annuler"*
4. *L'utilisateur retourne au menu principal de l'application*

4.4.5 Diagramme de la fonction 'Changement du mot de passe du système'

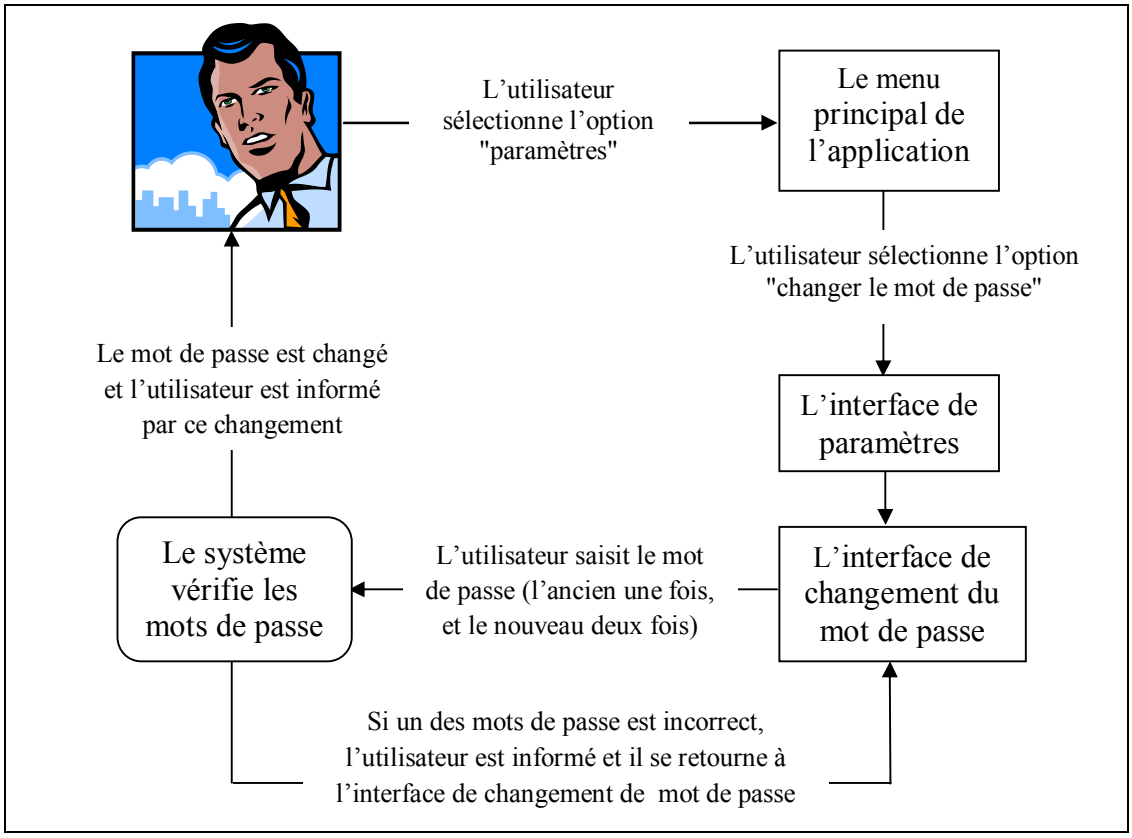


Figure 4.6. *Diagramme de la fonction 'Changement du mot de passe'*

Le flux des événements de la fonction ***Changement du mot de passe du système*** :

Flux principal :

1. *L'utilisateur sélectionne l'option "paramètres"*
2. *L'utilisateur sélectionne l'option "changer le mot de passe"*
3. *L'utilisateur saisit le mot de passe (l'ancien une fois, et le nouveau deux fois)*
4. *Le system vérifie les mots de passe*
5. *Le mot de passe est changé et l'utilisateur est informé par ce changement*

Flux alternatif :

1. *L'utilisateur sélectionne l'option "paramètres"*
2. *L'utilisateur sélectionne ensuite l'option "changer le mot de passe"*
3. *L'utilisateur saisit le mot de passe (l'ancien une fois, et le nouveau deux fois)*
4. *L'utilisateur saisit les mots de passe incorrects*
5. *L'utilisateur est informé que les mots de passe sont incorrects*

4.4.6 Diagramme de la fonction 'Visualiser les fichiers chiffrés'

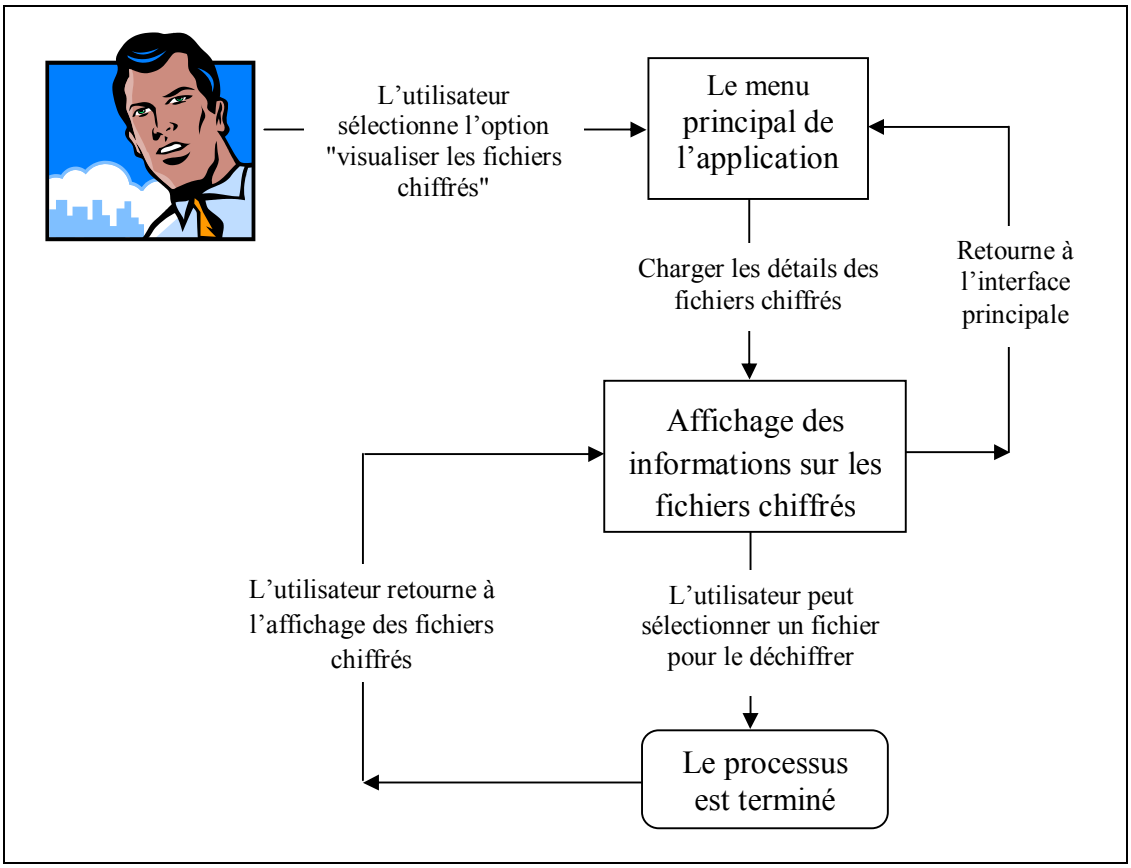


Figure 4.7. *Diagramme de la fonction 'Visualiser les fichiers chiffrés'*

Le flux des événements pour le cas d'utilisation de la fonction ***Visualiser les fichiers chiffré*** :

Flux principal :

1. *L'utilisateur sélectionne l'option "visualiser les fichiers chiffrés"*
2. *Les informations sur les fichiers chiffrés sont affichées*
3. *L'utilisateur peut appeler le processus de déchiffrement*
4. *Après ce processus, l'utilisateur est retourné à l'affichage des fichiers chiffrés*
5. *L'utilisateur peut retourner à l'interface principale*

Flux alternatif :

1. *L'utilisateur sélectionne l'option "visualiser les fichiers chiffrés"*
2. *Les informations sur les fichiers chiffrés sont affichées*
3. *L'utilisateur retourne à l'interface principale d'application*

4.4.7 Diagramme de la fonction 'Changement de la clé de chiffrement'

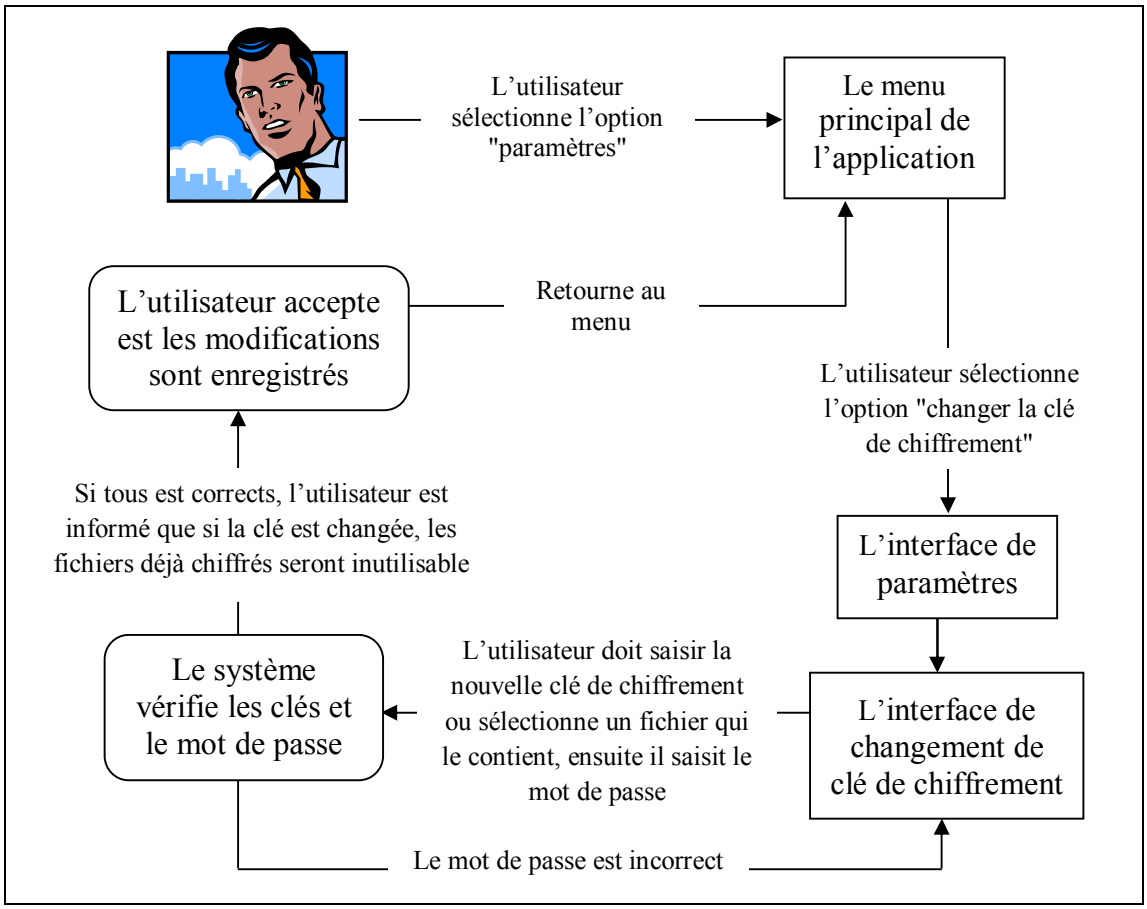


Figure 4.8. *Diagramme de la fonction 'Changement de la clé de chiffrement'*

Le flux des événements pour ce cas d'utilisation :

Flux principal :

1. *L'utilisateur sélectionne l'option "paramètres"*
2. *L'utilisateur sélectionne ensuite l'option "changer la clé de chiffrement"*
3. *L'utilisateur doit saisir la nouvelle clé de chiffrement via un stylet ou la sélectionne depuis un fichier, ensuite il saisit le mot de passe*
4. *Le système demande à l'utilisateur de confirmer ce changement*
5. *L'utilisateur accepte et les modifications sont apportées*

Flux alternatif :

1. *L'utilisateur sélectionne l'option "paramètres"*
2. *L'utilisateur sélectionne ensuite l'option "changer la clé de chiffrement"*
3. *L'utilisateur saisit la nouvelle clé de chiffrement, puis il saisit le mot de passe*
4. *L'utilisateur saisit un mot de passe incorrect*
5. *L'utilisateur retourne à l'interface de changement de clé de chiffrement*

4.5 Spécifications matérielles

a- Caractéristiques des dispositifs de test

Le tableau suivant représente les caractéristiques du dispositif PDA HP iPAQ Pocket PC h5550.

Caractéristiques du HP iPAQ Pocket PC h5550	
Système d'exploitation	Microsoft Pocket PC 2003
Processeur	Processeur Intel XScale PXA250 400 MHz
Type d'affichage	3.8" matrice active TFT - transflectif, 64K couleurs
Ecran tactile	Oui
Résolution écran (pixels)	240 x 320
Dimensions	13,8 x 8,4 x 1,6 cm (h x l x p)
Taille utile de l'image (l x h)	57,6 mm x 76,8 mm
RAM	128 MB- SDRAM
ROM	48 MB- mémoire Flash
Slot mémoire externe	SD Memory Card
Nombre total de connecteurs d'extension (disponibles)	1 (1) x Carte mémoire SD
Interfaces	1 x casque - sortie 1 x USB - USB à 4 broches, type A 1 x série - RS-232
Matériel de connectivité	Station d'accueil
Connectivité sans fil	Bluetooth, IEEE 802.11b
Audio	Haut-parleur, microphone, MP3
Dispositifs de sécurité	lecteur d'empreinte digitale
Batterie	Lithium-Ion polymère 1250 mAh amovible et rechargeable
Poids	207 g

Tableau 4.8. *Caractéristiques du dispositif HP iPAQ Pocket PC h5550* [78]

Le tableau suivant représente les caractéristiques du téléphone mobile SGH-i200.

Caractéristiques du Samsung SGH-i200	
Système d'exploitation	Microsoft Windows Mobile 6.1
Nombre de couleurs	262.144 couleurs
Ecran tactile	Non
Résolution écran (pixels)	240 x 320
Dimensions	116.7 x 50.8 x 11.8 mm (l x h x p)
Technologie écran	262k couleur TFT écran
Mémoire interne	128 MB
Slot mémoire externe	microSD
Mode GSM	GSM -Tri Band (900 / 1800 / 1900 MHz)
Fonction MMS, SMS, WAP, GPRS	oui
Navigateur Internet	oui
Connectivité sans fil	Bluetooth
Lecteur vidéo, lecteur MP3, audio OUT	oui
Synchronisation PC	WINDOWS XP / WINDOWS VISTA
Technologie batterie	Li-ion, avec une autonomie en conversation 7 heures et en veille 300 heures

Tableau 4.9. *Caractéristiques du téléphone mobile SGH-i200* [79]

b- Logiciel

Le système d'exploitation de base installé sur le dispositif de test PDA ainsi que son émulateur est le Pocket PC 2003 [78], aussi nommé Windows mobile 2003 [81], c'est une plate-forme riche et extensible pour le développement d'applications, c'est pour cette raison que nous avons choisi la solution de programmation de Microsoft.

Les avantages d'un Pocket PC 2003 sont les suivantes [81] :

- Il a une meilleure gestion de la mémoire et il est sensiblement plus rapide pour de nombreuses tâches que ses prédécesseurs ;
- Il a une meilleure interface utilisateur ;
- Il simplifie la connexion à de nombreux réseaux sans fil ;
- Il comprend le logiciel ActiveSync qui le rend facile à gérer le transfert des fichiers d'un PC vers un PDA ou l'inverse.

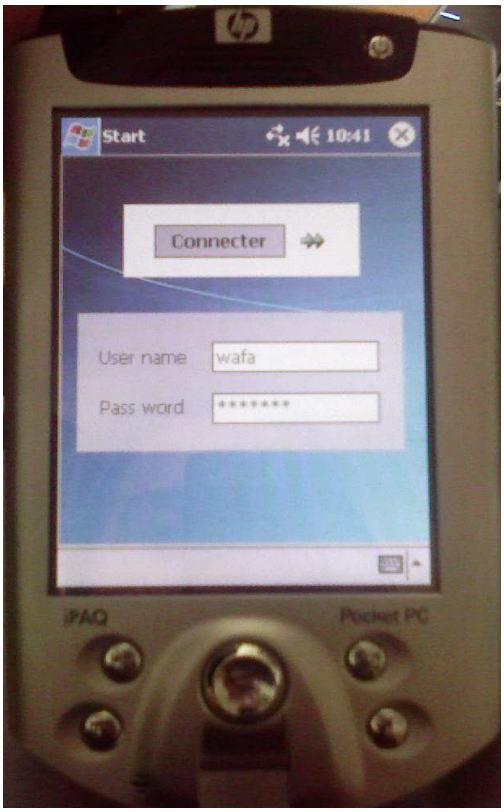
Pour le téléphone mobile et son émulateur utilisé, ils disposent d'un système d'exploitation Windows Mobile 6.1[79], qui est une version évoluée de Windows mobile 2003.

4.6 Résultats de l'exécution de l'application développée sur les dispositifs de test

Après avoir expliqué en détails les différentes fonctions de l'application, nous allons présenter les résultats de l'exécution de cette dernière sur notre PDA de test et son émulateur ainsi que sur l'émulateur de téléphone mobile SGH-i200 pour chiffrer/déchiffrer les données personnelles (fichier, audio, vidéo, image) en utilisant l'algorithme de chiffrement AES.

a- Sur le PDA "HP iPAQ Pocket PC h5550" et son émulateur Pocket PC 2003 SE

L'utilisation de l'application est simple, pour commencer, il suffit de sélectionner la fonction ***"connexion"*** pour pouvoir entrer à l'interface principale, et cela après la saisie correcte du nom d'utilisateur et du mot de passe, comme le montre les quatre figures suivante sur le PDA et son émulateur :



Connexion

Figure 4.9. *Fonction 'connexion' de l'application sur le PDA*

Figure 4.10. *Menu de choix de la fonction appropriée sur le PDA*

Connexion

Figure 4.11. *Fonction 'connexion' de l'application sur l'émulateur*

Figure 4.12. *Menu de choix de la fonction appropriée sur l'émulateur*

Une fois l'utilisateur est connecté, il aura plusieurs choix, soit :

→ Chiffrer/Déchiffrer un fichier : où l'utilisateur aura une option "**Chiffrer**" / "**Déchiffrer**" sur l'interface du système pour chiffrer/déchiffrer son fichier sélectionné. Ceci est fait après la sélection de la taille de la clé de chiffrement.

Pour évaluer le système, l'utilisateur choisit un numéro de téléphone à partir de la liste des contacts. Une fois sélectionné, le numéro est chiffré dans le PDA en utilisant une clé de 128 bits.

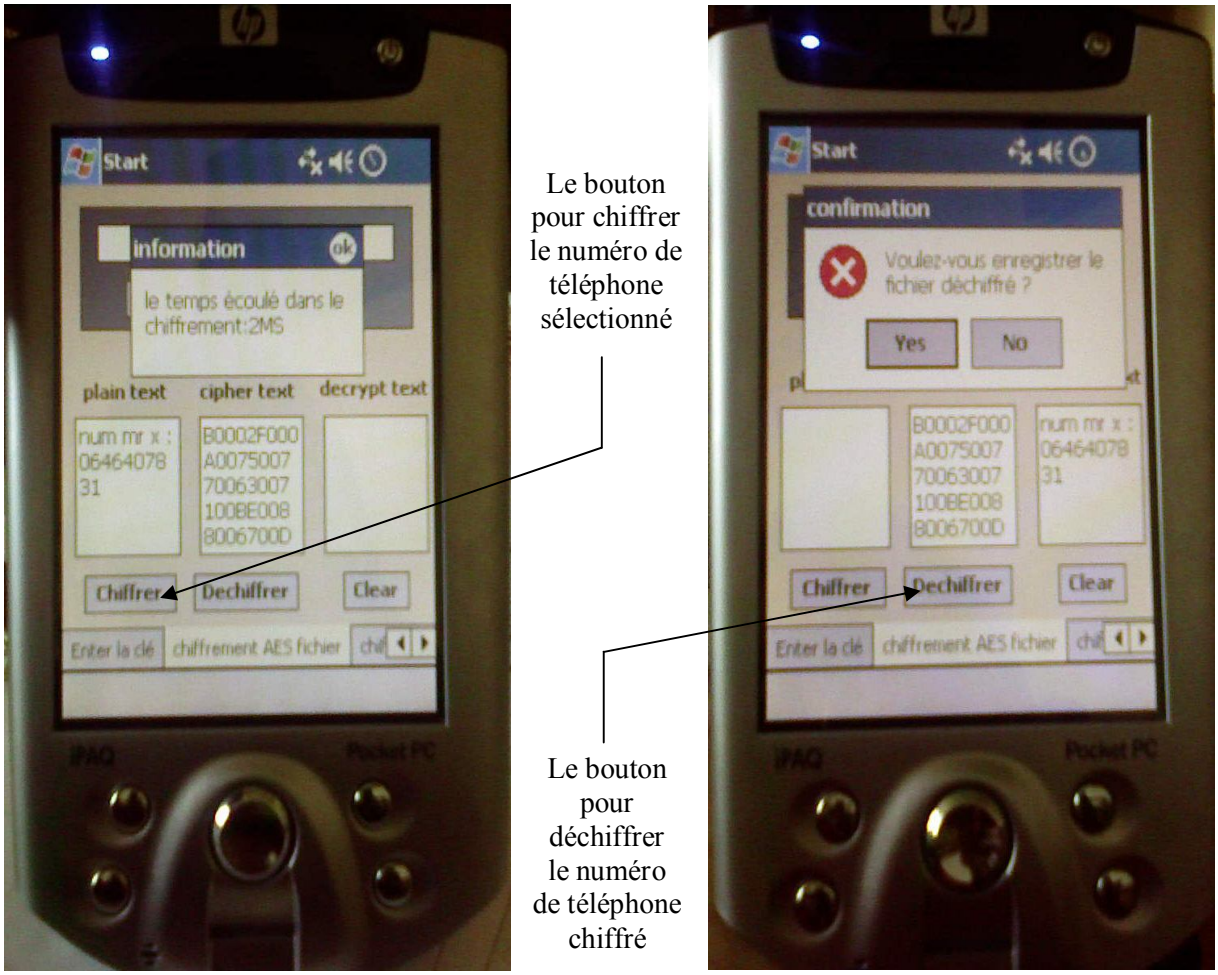


Figure 4.13. *Fonction de chiffrement d'un numéro de téléphone sélectionné sur le PDA*

Figure 4.14. *Fonction de déchiffrement d'un numéro de téléphone sélectionné sur le PDA*

Avec la même taille de la clé (128 bits), nous faisons sur l'émulateur le chiffrement/déchiffrement d'un numéro de téléphone sélectionné à partir de la liste des contacts.

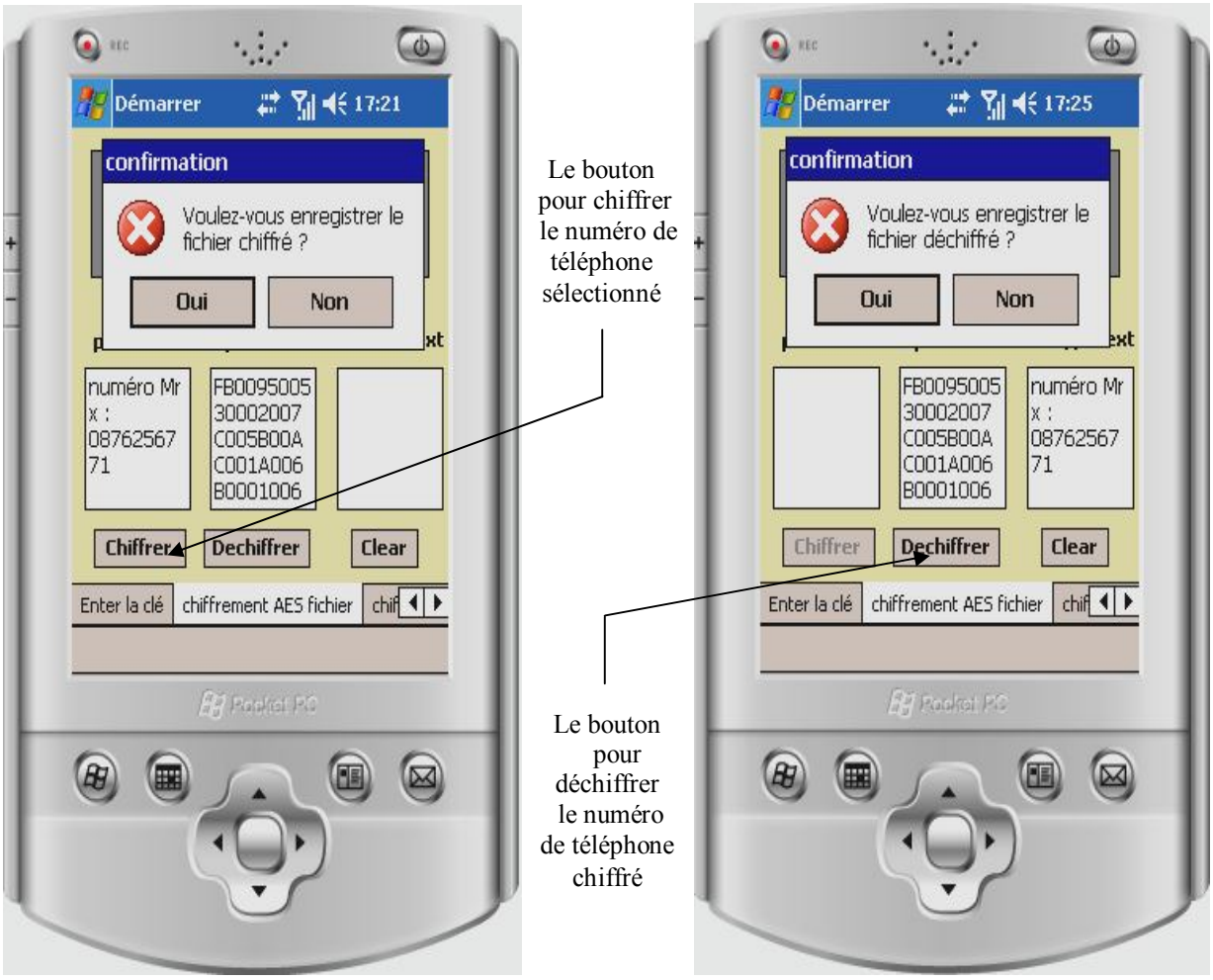


Figure 4.15. *Fonction de chiffrement d'un numéro de téléphone sélectionné sur l'émulateur*

Figure 4.16. *Fonction de déchiffrement d'un numéro de téléphone sélectionné sur l'émulateur*

- Pour le chiffrement des images, des audio, des vidéos, et la fonction de déconnexion, ils sont présentés dans l'annexe D.

→ Changer les paramètres : le changement du mot de passe ou de la clé de chiffrement s'effectue facilement en utilisant l'interface affichée sur le PDA :

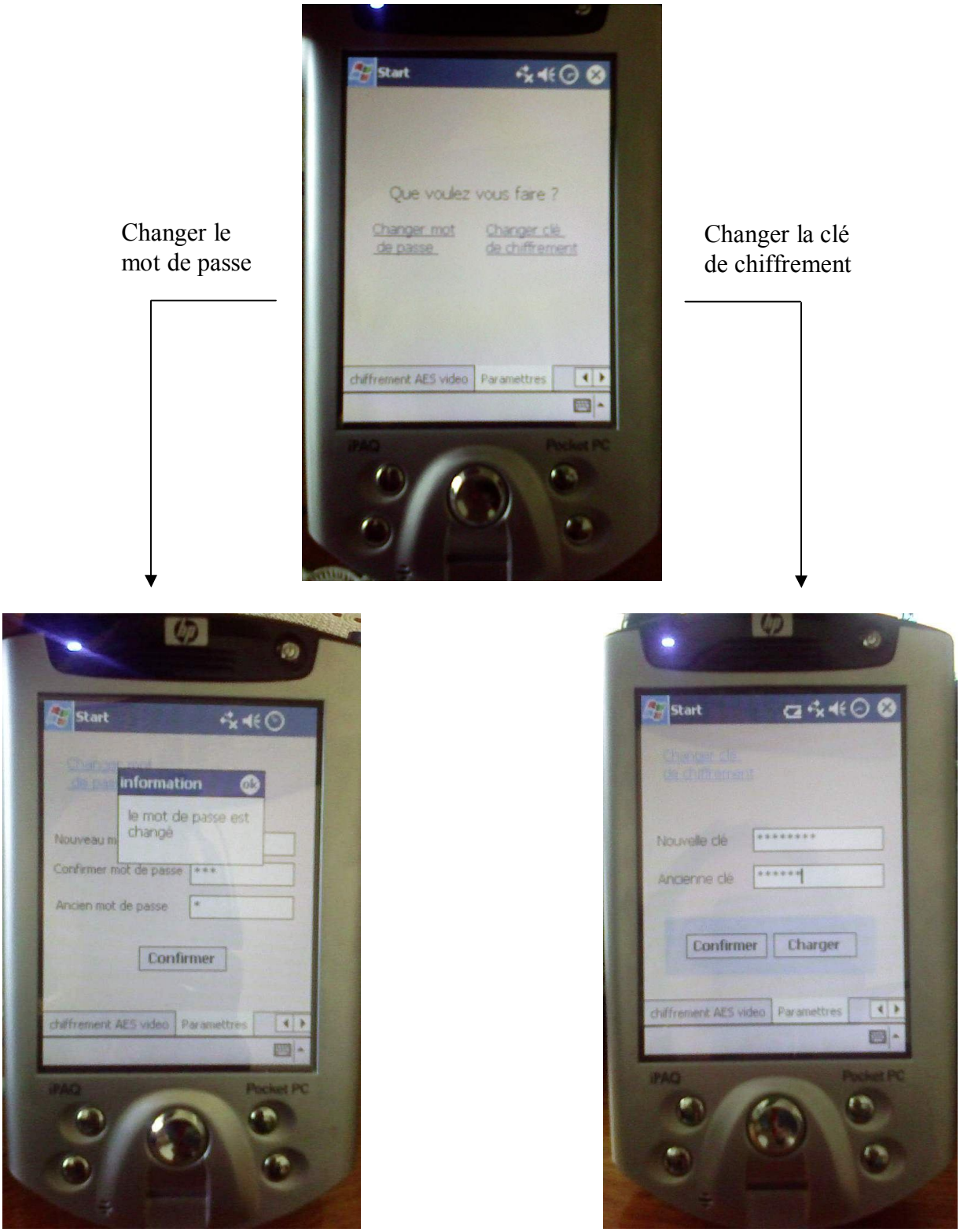


Figure 4.17. *Les fonctions de changement du mot de passe, de changement de la clé de chiffrement du système sur le PDA*

b- Sur l'émulateur du téléphone mobile SGH-i200

- Le chiffrement/déchiffrement d'un fichier sur l'émulateur du téléphone mobile s'effectue de la même manière que dans le cas de PDA.

Pour évaluer le système sur l'émulateur, l'utilisateur choisit un numéro de téléphone à partir de la liste des contacts. Une fois sélectionné, le numéro est chiffré en utilisant une clé de 128 bits.

Figure 4.18. *Fonction de chiffrement d'un numéro de téléphone sélectionné sur l'émulateur du téléphone*

Figure 4.19. *Fonction de déchiffrement d'un numéro de téléphone sélectionné sur l'émulateur du téléphone*

- Pour le chiffrement des images, des audio, des vidéos dans l'émulateur, les résultats sont présentés dans l'annexe D.

Remarque : On note qu'on obtiendra les mêmes résultats (soit pour les fichiers, les images, les audio, ou les vidéo) si on exécute l'application sur les téléphones mobiles ayant comme système d'exploitation le Windows Mobile.

c- Evaluation des performances du système développé sur le PDA

Pour évaluer les performances du système développé, nous avons effectué des expériences en changeant la taille des images et en calculant le temps moyen requis par le chiffrement et le déchiffrement. Pour chaque taille de l'image considérée (32x32, 64x64, 128x128, 256x256, 512x512 pixels), six images différentes ont été utilisées et le temps moyen de chiffrement / déchiffrement était calculé comme le montre le tableau 4.10.

Temps moyen / Taille image (pixels)	Chiffrement (ms)	Déchiffrement (ms)
32 x 32	1606,83	2169,50
64 x 64	5545,83	8122,00
128 x 128	21190,66	33561,16
256 x 256	80736,16	129331,66
512 x 512	334165,83	493241,50

Tableau 4.10. *Temps moyen de chiffrement et de déchiffrement des images sur le PDA*

Comme s'est montré sur la Figure 4.20, le temps moyen de chiffrement et de déchiffrement augmente en augmentant la taille de l'image. La forme de l'augmentation suit une grandeur d'ordre 2 puisque l'image est de type bidimensionnel. En plus, on constate que pour des images de taille moins de 128x128 pixels, le temps est tout à fait acceptable en tenant les contraintes des dispositifs mobiles en termes de puissance, taille de la mémoire, …etc.

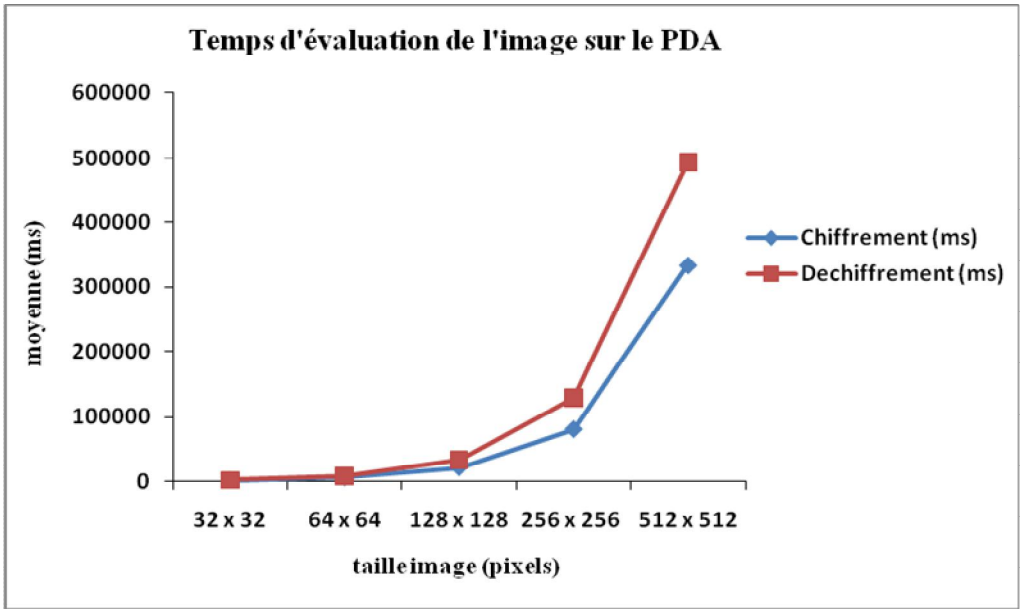


Figure 4.20. *Temps d'évaluation de l'image sur le PDA*

De même, des expériences sur des données audio utilisant des tailles différentes ont été effectuées. Tableau 4.11 et Figure 4.21 montrent le temps moyen requis pour le processus de chiffrement et de déchiffrement et la représentation graphique correspondante, respectivement. Le temps moyen augmente d'une manière presque linéaire comme prévu puisque le signal audio est de type unidimensionnel. Les résultats obtenus sont également acceptables étant donné les contraintes des dispositifs mobiles.

Temps moyen / Taille audio (ko)	Chiffrement (ms)	Déchiffrement (ms)
32	14480,66	21507,16
64	28635,00	43789,50
128	58895,83	88063,16
253	115994,00	170905,66
514	225923,00	348807,50

Tableau 4.11. *Temps moyen de chiffrement et de déchiffrement des audio sur le PDA*

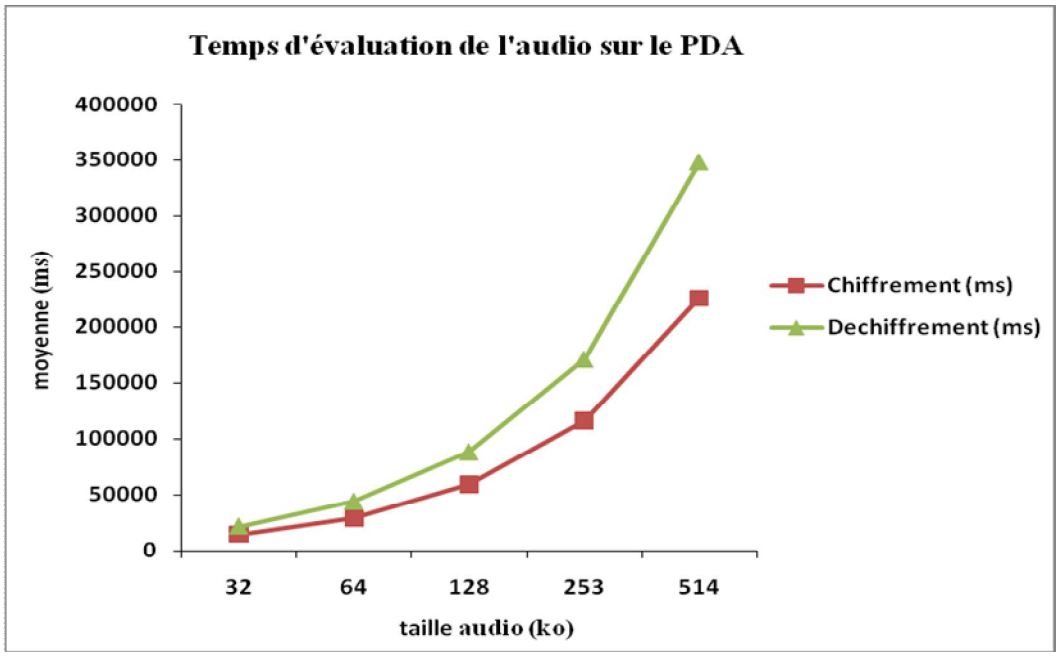


Figure 4.21. *Temps d'évaluation de l'audio sur le PDA*

Pour que le temps de chiffrement et de déchiffrement peut être amélioré, en optimisant le code des opérations effectuées (cas du chiffrement et de déchiffrement) du système.

4.7 Conclusion

Ce chapitre a été consacré au développement d'une application de chiffrement pour la sécurisation des données sensibles sur les dispositifs mobiles (PDA/ téléphones mobiles). L'application proposée est facile à utilisé même par quelqu'un qui n'a pas beaucoup de conaissance pour chiffrer et déchiffrer ses données personnelles (contacts, rendez-vous, emails, images, vidéos, audio,…). Une fois installée sur un dispositif mobile, l'utilisateur peut protéger ses données efficacement. Elle est basée sur l'algorithme de chiffrement AES, qui est le plus fiable, efficace et fort des algorithmes de chiffrement disponibles aujourd'hui, en mode CBC, avec des tailles de clé de 128 bits, 192 bits, 256 bits.

Ainsi, notre application prend peu d'espace mémoire et ne nécessite pas une machine virtuelle pour être exécutée sur le dispositif, de plus qu'elle est portable et les données chiffrées peuvent être utilisées dans les ordinateurs de bureaux, ainsi dans un dispositif autre que le dispositif où les données ont été chiffrées.

CONCLUSION ET PERSPECTIVES

Le travail réalisé dans le cadre de notre mémoire s'intéresse aux problématiques liées à la sécurité des données sensibles sur les dispositifs mobiles qui ont fait l'objet de recherches ces dernières années. Le développement fulgurant des moyens de communication modernes a permis une large distribution des téléphones portables et PDA ces dernières années, et ils sont désormais courants dans de nombreux endroits du monde, en plus, les utilisateurs de ces terminaux mobiles cherchent un accès rapide et direct aux informations personnelles ou de leurs entreprises, partout dans le monde, à n'importe quel moment, comme s'ils étaient dans leurs bureau. Leur objectif étant d'optimiser leurs temps de travail, tout en gardant une facilité d'utilisation. Malgré ces avantages, la miniaturisation de ces dispositifs (téléphone mobile, PDA,…) risque sa perte ou son vol qui peut compromettre ses données confidentielles.

Dans la mesure où la première cause des divulgations et fuites de données est le vol ou la perte de ces dispositifs, l'utilisateur doit impérativement se prémunir contre l'accès à ses données confidentielles stockées sur ces systèmes mobiles dans le cas où ces derniers tomberaient entre les mains d'utilisateurs non autorisés. Le chiffrement est la principale technologie permettant de remédier à ce problème, afin de sécuriser toutes les données stockées sur ces dispositifs.

Dans le chapitre 1, nous avons abordé les différents algorithmes cryptographiques proposés pour le chiffrement, que ce soit classique ou moderne, et comme le but est d'appliquer un algorithme de chiffrement incassable par les intrus, les algorithmes cryptographiques modernes et plus précisément à clé secrète sont apparu les plus approprié par rapport aux algorithmes de chiffrement à clé publique.

En deuxième lieu, notre travail a essentiellement consisté à présenter des généralités sur les terminaux mobiles sur lesquels notre application est implémentée. L'accent est mis particulièrement sur les PDA et les téléphones mobiles, où on a découvert que le développement des applications sur ces dispositifs est contraint par ses limitations : consommation mémoire et d'énergie de processeur, puissance de traitement limitée, taille

d'écran très réduite,….etc. De ce fait, plusieurs outils de développement existent, mais la plupart nécessite une machine virtuelle à être installé par l'utilisateur, et la plupart des machines virtuelles ne fonctionnent pas parfaitement sur les dispositifs mobiles en raison de la puissance limitée du processeur. Pour cette raison, on a proposé l'utilisation de la solution Microsoft "studio visuel" comme une plateforme de développement de notre application, car elle fournit des excellents équipements de teste et un ensemble complet d'outils de développement.

Dans le chapitre 3, on a détaillé l'algorithme cryptographique symétrique AES (Advanced Encryption Standard), qu'on a choisi parmi les différents algorithmes cité dans le chapitre 1 pour développer notre application de chiffrement. Le choix de cet algorithme répond à de nombreux critères dont nous pouvons citer : l'AES est le plus récent algorithme à clé symétrique, il est relativement simple, entraine une grande rapidité de traitement, sa flexibilité d'implémentation inclut des tailles de clés et de blocs supplémentaires, besoins en ressources et mémoire très faibles, en plus, c'est le plus fiable, et fort des algorithmes de chiffrement disponibles aujourd'hui [88].

Dans le dernier chapitre, nous avons développé une application de chiffrement des données sensibles sur les PDA et téléphones mobiles modernes et proposé comme solution de problèmes de sécurité cités précédemment. Cette application repose, d'un coté sur l'algorithme de chiffrement symétrique AES comme un outil de chiffrement, en mode CBC, qui est l'un des meilleurs modes et l'un des plus utilisés [85], et d'un autre coté, sur la plateforme proposée comme un outil de développement qui est le studio visuel de Microsoft. L'application développée peut être utilisée par un utilisateur qui n'a pas beaucoup de connaissance en chiffrement pour chiffrer et déchiffrer ses données personnelles (contacts, rendez-vous, audio, image, vidéo,...) facilement.

Les avantages du système proposé sont :

- Il est portable, peut être installé sur n'importe quel nombre de dispositifs de PDA ou téléphone mobile.
- Des grandes clés de chiffrement ne devront pas rappeler.
- Les données chiffrées sur un PDA/téléphone mobile spécifique peuvent être utilisés sur d'autres dispositifs mobiles, en les envoyant par courrier électronique ou par Bluetooth ;

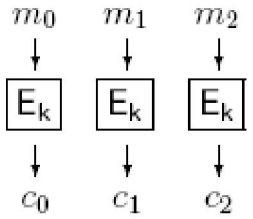
- Le système pourrait importer / exporter les valeurs de clé de chiffrement, ou de la taper directement.
- Les données peuvent être stockées sur la carte mémoire amovible sur le dispositif et utilisé sur les ordinateurs de bureaux et autres PDA et téléphone mobile.
- Il prend un peu d'espace mémoire et ne nécessite pas une machine virtuelle pour l'avoir exécuté sur le dispositif.

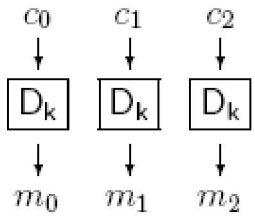
A l'issue de ces travaux, ce mémoire ouvre de nouvelles perspectives de recherche parmi lesquelles nous citons :

- Il serait intéressant de combiner l'algorithme AES, avec un algorithme cryptographique asymétrique (RSA par exemple), afin de sécuriser le partage du mot de passe de notre application, qui sera transféré chiffré et d'une façon sûre au destinataire, via Bluetooth ou SMS ou e-mail. Ensuite cette personne peut utiliser notre application pour le chiffrement ou le déchiffrement. Dans ce cas, la performance temps de calcul doit être considérée afin que l'application ne soit pas assez lente.
- L'optimisation de temps de chiffrement / déchiffrement des audio, des vidéos et des images de taille importante contenus dans le dispositif, afin d'améliorer la performance de l'application.
- Il est intéressant aussi de généraliser le mécanisme de chiffrement aux SMS, pour que la sécurité englobe toute le dispositif.
- L'application du système proposé sur les téléphones mobiles ayant comme système d'exploitation le Symbian.
- Il est aussi intéressant de proposer d'autres méthodes pour alerter l'utilisateur à chiffrer ses nouvelles données.
- Aborder le problème des attaques des données chiffrées et de fonctionnement de notre système développé par des virus existants dans le PAD.
- Trouver une solution pour sauvegarder le mot de passe et le nom d'utilisateur de système en évitant ses pertes par l'utilisateur.

Annexe A

$c_i = E_k(m_i)$



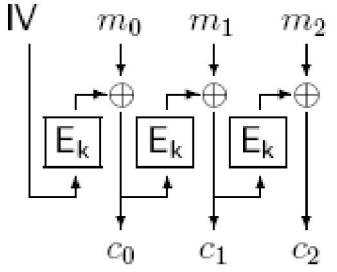


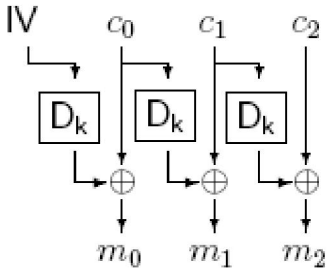
$m_i = D_k(c_i)$

Figure A.1. *Chiffrement et déchiffrement en mode ECB*

$c_{-1} = IV$

$c_i = E_k(c_{i-1}) \oplus m_i$





$m_i = D_k(c_{i-1}) \oplus c_i$

Figure A.2. *Chiffrement et déchiffrement en mode CFB*

$s_{-1} = IV \quad s_i = E_k(s_{i-1})$

$c_i = m_i \oplus s_i$

$m_i = c_i \oplus s_i$

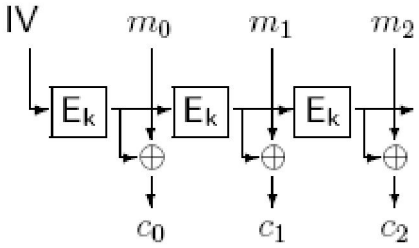


Figure A.3. *Chiffrement ou déchiffrement en mode OFB*

$s_i = E_k(IV + i)$

$c_i = m_i \oplus s_i$

$m_i = c_i \oplus s_i$

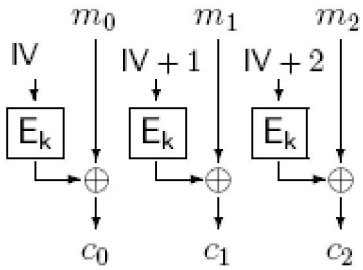


Figure A.4. *Chiffrement ou déchiffrement en mode CTR*

Annexe B

Le tableau comparatif entre un PDA, tablette PC et ordinateur portable est le suivant :

Avantages de ⇒ Par rapport à ⇓	**PDA**	**Tablette PC**	**Ordinateur portable**
PDA		- Taille de l'écran - Facilité d'utilisation - Capacité de stockage	- Possibilité d'utilisation - Performance - Taille de l'écran - Capacité de stockage
Tablette PC	- Autonomie - Prix - Vitesse de démarrage - Maniabilité - Robustesse		- Ratio - Performance/prix - Performance - Capacité de stockage
Ordinateur portable	- Autonomie - Prix - Vitesse de démarrage - Mobilité/encombrement - Maniabilité - Poids - Robustesse	- Facilité de transport - Facilité d'utilisation - Maniabilité	

Le tableau suivant récapitule certaines technologies de téléphonie mobile [54]:

Génération	**Acronyme**	**Intitulé**	**Date d'apparition des offres « grand public » en France**
1G	Radiocom 2000	Radiocom 2000 France Telecom	1986
2G	GSM	Global System for Mobile Communication	1992/1995[8]
2.5G	GPRS	General Packet Radio Service	2001/2002
2.75G	EDGE	Enhanced Data Rate for GSM Evolution	2005
3G	UMTS	Universal Mobile Telecommunications System	2004/2005
3.5G	HSDPA	High Speed Downlink Package Access	2006/2007

Annexe C

Les définitions suivantes sont utilisées dans le standard AES :

AES	Advanced Encryption Standard (standard de chiffrement avancé)
Transformation Affine	Une transformation qui consiste d'une multiplication par une matrice suivie par l'ajout d'un vecteur
Bit	Un chiffre binaire ayant une valeur de 0 ou de 1
Octet	Un groupe de huit bits qui est considéré soit comme une entité unique ou comme un tableau de huit bits individuels.
Bloc	Séquences de bits binaires qui composent l'entrée, la sortie, l'état et la clé de tour. Les blocs sont aussi interprétés comme des tableaux d'octets
Tableau	Une collection énumérée d'entités identiques (par exemple, octets)
Clé de chiffrement	Clé secrète qui est utilisé par la routine d'expansion de clé pour générer un ensemble des clés de tour, on peut la considérer comme un tableau rectangulaire d'octets, ayant quatre lignes et *Nk* colonnes
Chiffrement	Série de transformations qui convertit un texte clair en texte chiffré en utilisant la clé de chiffrement
Texte chiffré	Les données de sortie de chiffrement ou d'entrée de déchiffrement
Déchiffrement	Série de transformations qui convertit un texte chiffré en texte clair en utilisant la clé de déchiffrement
Texte clair	Les données d'entrée de chiffrement ou de sortie de déchiffrement
Expansion de clé	Routine utilisée pour générer une série de clés de tour à partir de la clé de chiffrement

Clé de tour	Clé dérivée de la clé de chiffrement utilisant la routine d'expansion de clé
Rijndael	Algorithme cryptographique spécifique dans l'AES
Etat	Résultat intermédiaire de chiffrement qui peut être décrite comme un tableau rectangulaire d'octets, de quatre lignes et de *Nb* colonnes.
S-box	Table de substitution non-linéaire utilisée dans le chiffrement et à la routine d'expansion de clé pour la substitution des octets des blocs
Mot	Un groupe de 32 bits qui est considérée soit comme une entité unique ou comme un tableau de quatre octets.

Les symboles, et les fonctions suivants sont utilisés dans cette norme :

K	Clé de chiffrement
Nb	Nombre de colonnes (mots de 32-bit) du bloc. Pour l'AES, Nb = 4.
Nk	Nombre de mots de 32-bit dans la clé de chiffrement. Pour l'AES, Nk = 4, 6 ou 8
Nr	Nombre de tours, qui est une fonction de Nk et Nb. Pour l'AES, Nr = 10, 12 ou 14
AddRoundKey ()	transformation lors du chiffrement et du déchiffrement dans lequel une clé de tour qui est de taille 128 bits est ajouté à l'état en utilisant l'opération XOR
SubBytes ()	Transformation non-linéaire qui substitue chaque octet de l'état indépendamment en utilisant la table de substitution (S-box) lors du chiffrement
ShiftRows ()	Transformation lors du chiffrement qui décale cycliquement les trois dernières lignes de l'état
MixColumns ()	Transformation linéaire lors du chiffrement qui prend chaque colonne de l'état et la multiplier par une matrice pour produire de nouvelle colonne
InvSubBytes ()	C'est la transformation inverse de SubBytes (), utilisé dans le déchiffrement

InvShiftRows ()	C'est la transformation inverse de ShiftRows, utilisé dans le déchiffrement
InvMixColumns ()	C'est la transformation inverse de MixColumns (), utilisé dans le déchiffrement
Rcon []	Tableau constant
SubWord ()	Fonction utilisée par la routine d'expansion de la clé qui prend un mot de quatre octet en entrée et applique une substitution non linéaire (S-box) pour chacun de ses octets
RotWord ()	Fonction utilisée par la routine d'expansion de la clé qui applique à un mot de quatre une permutation circulaire
XOR	Opération ou-exclusif XOR
$\oplus$	Opération ou-exclusif
$\otimes$	Multiplication de deux polynôme (avec degré <4) modulo $x^4 + 1$
•	Multiplication de corps fini.

Annexe D

Le cas de PDA :

- Pour chiffrer/déchiffrer une image, l'utilisateur clique sur l'option "**Chiffrer**"/"**Déchiffrer**" affichée sur le PDA.

Pour évaluer le système sur la protection des images, des images à niveau de gris de type BMP ayant une taille de 185x142 pixel sont sélectionnés par l'utilisateur et chiffré avec une clé de 192 bits (ainsi que des clés de 128 et 256 bits et d'autres types d'image peuvent être utilisées).

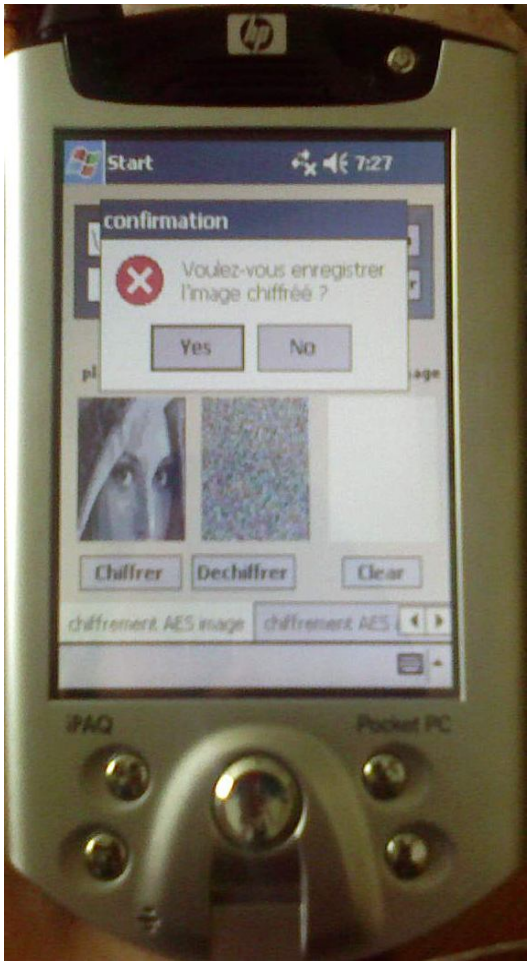


Figure D.1. *Le chiffrement de l'image sélectionnée sur le PDA*

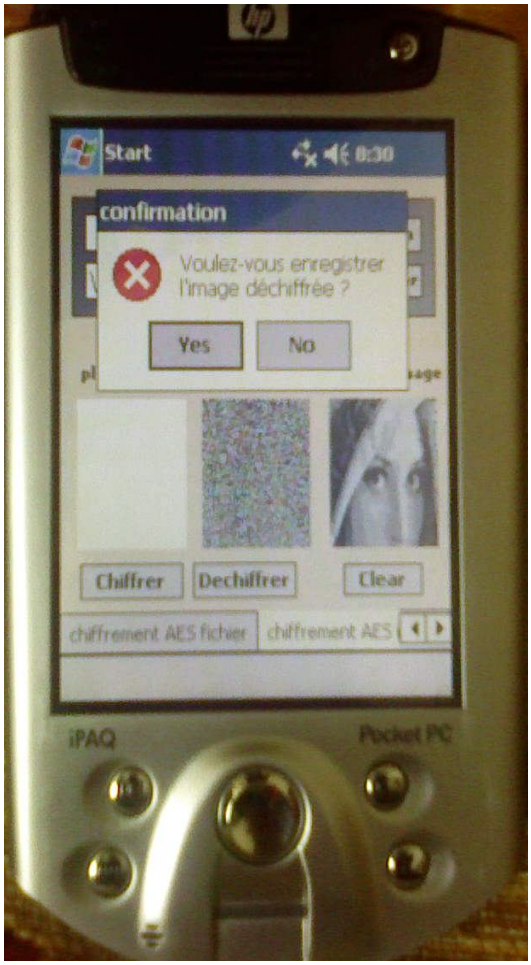


Figure D.2. *Le déchiffrement de l'image sélectionnée sur le PDA*

Avec la même taille de clé, nous faisons le chiffrement/déchiffrement de la même image sur l'émulateur comme le montre les figures (Figure D.3) et (Figure D.4) suivantes :

Figure D.3. *Le chiffrement de l'image sélectionnée sur l'émulateur*

Figure D.4. *Le déchiffrement de l'image sélectionnée sur l'émulateur*

- Pour le chiffrement/déchiffrement des audio, nous procédons de la même manière que dans le cas des fichiers et des images.

Dans le cas de la protection des audio sur le dispositif PDA, nous choisissons un audio de type MP3. L'utilisateur peut ensuite utiliser sa clé (128 bits) pour le chiffrer (ainsi que des clés de 192 et 256 bits et d'autres types d'audio puissent être utilisées).

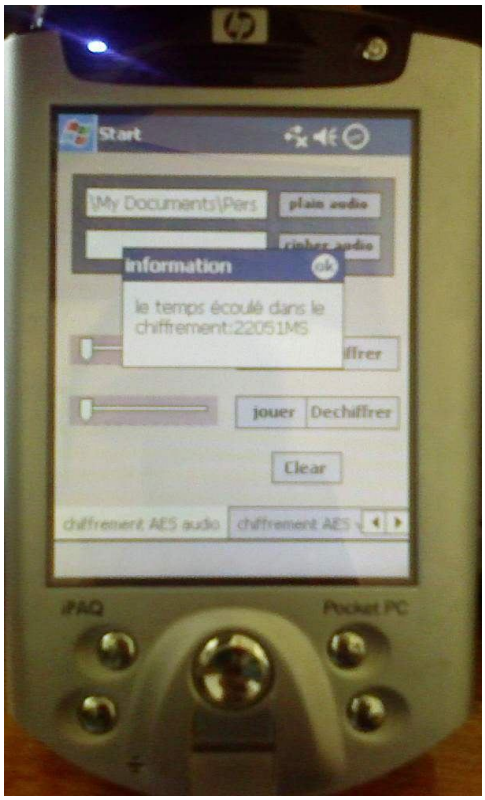


Figure D.5. *Le chiffrement de l'audio sélectionné sur le PDA*

Figure D.6. *Le déchiffrement de l'audio sélectionné sur le PDA*

Nous faisons le chiffrement/déchiffrement d'un audio de type MP3 sur l'émulateur avec la même taille de clé (128 bits):

Figure D.7. *Le chiffrement de l'audio sélectionné sur l'émulateur*

Figure D.8. *Le déchiffrement de l'audio sélectionné sur l'émulateur*

- Pour le chiffrement/déchiffrement des vidéos, nous procédons de la même manière que dans le cas des données chiffrées précédemment.

Dans le cas de la protection des vidéos sur le PDA, nous choisissons une vidéo du type AVI. L'utilisateur peut ensuite utiliser sa clé (192 bits) pour la chiffrer.

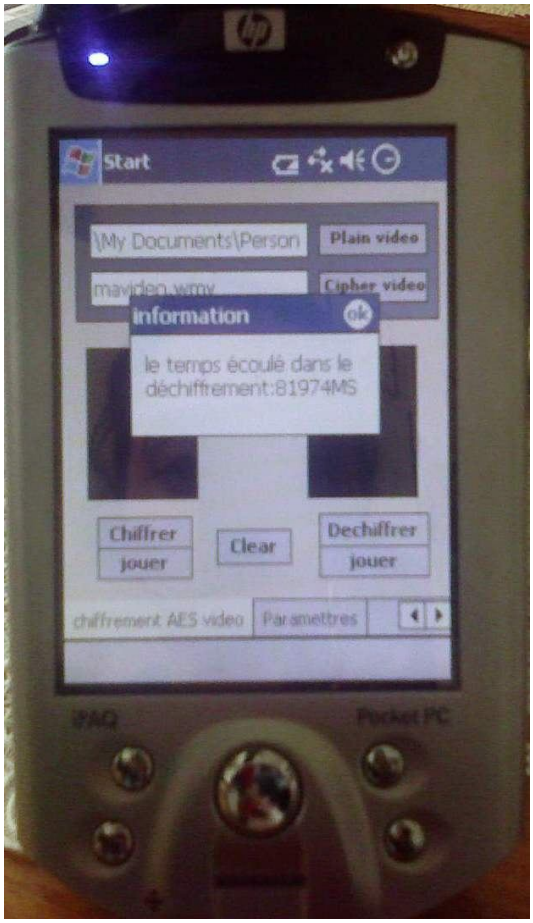


Figure D.9. *Le chiffrement de la vidéo sélectionné sur le PDA*

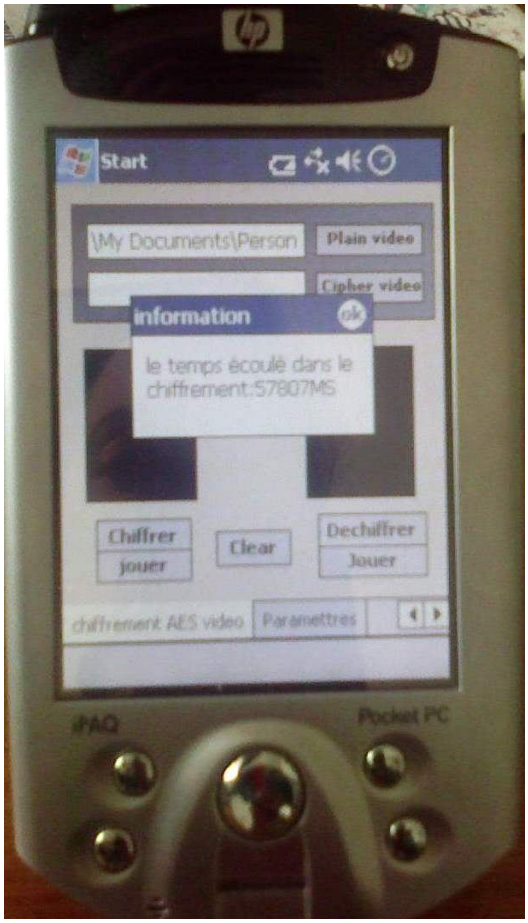


Figure D.10. *Le déchiffrement de la vidéo sélectionné sur le PDA*

Nous faisons le chiffrement/déchiffrement d'une vidéo de type MP4 sur l'émulateur, avec la même taille de clé (192 bits) comme suit :

Figure D.11. *Le chiffrement de la vidéo sélectionné sur l'émulateur*

Figure D.12. *Le déchiffrement de la vidéo sélectionné sur l'émulateur*

- Pour la déconnexion du système, un message s'affiche à l'utilisateur pour lui demander de quitter ou non l'application, comme la montre la figure suivante :

Figure D.13. *Fonction 'déconnexion' de l'application sur PDA*

Figure D.14. *Fonction 'déconnexion' de l'application sur l'émulateur*

Le cas du téléphone mobile :

- Le chiffrement/déchiffrement dans le téléphone mobile se déroule de la même manière que dans le cas de PDA et son émulateur.

Pour évaluer le système sur la protection des images dans l'émulateur du téléphone, une image couleur du type JPG d'une taille de 171x120 pixel est sélectionnée par l'utilisateur et chiffrée avec une clé de 128 bits (des clés de 192 et 256 bits ainsi que d'autres types d'image peuvent être utilisées).

Figure D.15. *Chiffrement de l'image sélectionnée sur l'émulateur du téléphone*

Figure D.16. *Déchiffrement de l'image sélectionnée sur l'émulateur du téléphone*

- Dans le cas de la protection des vidéos dans l'émulateur, on choisit une vidéo du type AVI. L'utilisateur peut ensuite utiliser sa clé (192 bits) pour la chiffrer (des clés de 128 et 256 bits ainsi que d'autres types de vidéo puissent aussi être utilisées).

Figure D.17. *Chiffrement de la vidéo sélectionné sur l'émulateur du téléphone*

Figure D.18. *Déchiffrement de la vidéo sélectionné sur l'émulateur du téléphone*

- Pour évaluer le système dans l'émulateur sur la protection des audio, un audio de type MP3 est sélectionné par l'utilisateur et chiffré avec une clé de 256 bits (des clés de 128 et 192 bits ainsi que d'autres types d'audio peuvent aussi être utilisées).

Figure D.19. *Chiffrement de l'audio sélectionné sur l'émulateur du téléphone*

Figure D.20. *Déchiffrement de l'audio sélectionné sur l'émulateur du téléphone*

Résumé

Le risque de vol des informations privées en cas de perte ou de vol d'un dispositif mobile (PDA, téléphone mobile) peut compromettre des données confidentielles, surtout que ces dispositifs sont largement utilisés par les gens d'affaires et plus généralement avec tous ceux qui voyagent fréquemment et qui ont besoin d'information en mouvement.

Ce travail vise à développer une application de chiffrement basée sur l'algorithme AES qui permet à l'utilisateur de chiffrer et de déchiffrer des données sensibles sur un téléphone mobile moderne / PDA (tel que : contacts, rendez-vous, e-mails, images, audio, et vidéos). Le système est destiné à toute personne ayant besoin de stocker des données sensibles sur un dispositif mobile en lui donnant une sécurité d'une manière que les informations ne seront jamais accédées par aucune personne d'autre. En raison des limitations de ressources d'un PDA/téléphone mobile (telles que le stockage), une partie de la solution serait de fournir un système capable à fonctionner parfaitement dans ces restrictions.
Parmi les avantages du système proposé:

- Système portable, qui peut être installé sur n'importe quel nombre de dispositifs mobiles;
- Des grandes clés de chiffrement ne devront pas être rappelés à chaque exécution;
- Les données chiffrées sur un PDA / téléphone mobile spécifique peuvent être utilisés sur d'autres dispositifs mobiles, en les envoyant par courrier électronique ou par Bluetooth;

Mots-clés : téléphone mobile, PDA, sécurité, cryptographie, AES, chiffrement, déchiffrement.

Abstract

The risk of theft of private information in case of loss or theft of a mobile device (PDA, mobile phone) can compromise confidential data, especially since these devices are widely used by business people and more generally with all those who travel frequently and have the need for information on the move.

This work aims to develop an encryption application based on AES algorithm that allows the user to encrypt and decrypt sensitive data on a modern mobile phone/PDA (such as contacts, appointments, emails, images, audio, and video). The system is designed for anyone who needs to store sensitive data on a mobile device giving security for the information so that it should not be accessed by anyone else. Due to the limited amount of resources of a PDA/mobile phone (such as storage), it will be part of the solution to provide a system that works perfectly under these restrictions.
Among the advantages of the proposed system:

- Portable system, could be installed on any number of mobile devices;
- Large encryption keys will not have to be remembered at each execution;
- The encrypted data on a specific PDA / mobile phone can be used on other mobile devices, by sending them by electronic mail or Bluetooth.

Keywords : mobile phone, PDA, security, cryptography, AES, encryption, decryption.