F.S.D……№ d'Ordre……

Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

Université Saad Dahlab de Blida

Faculté des Sciences

Département d'Informatique



Mémoire présenté par : **Mr Manave Edmundo Fransual**
**Mr Traoré Moussa Bakary**

En vue d'obtenir un diplôme de Master

**Domaine :** Mathématique et Informatique

**Filière :** Informatique

**Spécialité :** Informatique

**Option :** Génie des Systèmes Informatiques

Sujet :

# Plateforme de gestion sécurisée du système Evoucher

**Promotrice :** Mme Zahra Fatma Zohra

**Encadreuse :** Mme Sabiha Tagma

**Organisme d'accueil :** TECHNOLOGIES

Soutenue le : 25/06/2014 devant le jury composé de :

| | | |
|---|---|---|
| Mme | REZZOUG NASHIDA | Présidente |
| Mme | AROUSSI SIANA | Examinatrice |
| Mme | ARKAM MERIEM | Examinatrice |

2013/2014

# *Remerciement*

*Nous remercions ALLAH celui à qui tous les honneurs et louanges reviennent, d'avoir pu permettre la réalisation de ce présent travail.*

*Nous adressons nos vifs remerciements :*

*A nos chers parents du mali, du Mozambique et de l'Algérie*

*Au corps professoral du département d'informatique de l'université Saad Dahlab Blida 1 qui nous ont accompagnés au cours de ces années d'étude.*

*Notre entière gratitude va à l'endroit de notre promotrice* ***Mme Fatma Zahra*** *pour son encadrement sa patience, sa disponibilité, sa confiance. Elle qui par-dessus tout a su garder un regard critique et constructif sur tout le projet, nous vous remercions.*

*Le travail présenté dans ce mémoire a été réalisé au sein de l'entreprise* ***HB Technologie*** *sous la direction de* ***Mme Tagama Sabiha*** *notre encadreuse. Nous tenons à la remercier très vivement du fait de nous avoir accepté et proposer le présent thème mais aussi pour ses précieuses remarques, son sens d'organisation et de rigueur dont nous avons bénéficié.*

*A tous le personnel de l'entreprise HB TECHNOLOGIE.*

*A vous chers membres du jury de nous avoir fait honneur de votre présence pour juger cet humble travail.*

*A tous ces gens que nous n'avons pu évoqué qu'ils trouvent ici l'expression de nos profondes reconnaissances.*

# *Dédicace*

*Je dédie ce modeste travail aux êtres qui ont et qui continuent à contribuer à l'efflorescence de ma personnalité intellectuelle aussi bien qu'humaine.*

*A mon père*

*A mes mères Que la terre vous soit légère.*

*A Mon oncle Bien Aimée l'exemple de ma vie, toi qui a toujours guider et encourager mes pas depuis mon enfance jusqu'à l'heure actuelle qu' « ALLAH » vous accorde longue vie*

*A mes très chères et tendre tante que dieu vous garde longtemps auprès de nous*

*A mes frères et sœurs votre amour et soutient personnel n'a jamais manqué durant tout ce temps.*

*A toute la famille Traoré et Manave*

*A toute mes amis(e)s et camarades de la promotion de 2009 particulièrement Sall Cheick Oumar, Diané Moussa, Sinayogo Moussa, Diawara Mohamed Sidi, Konaté Amadou, Touré Awa, Assane Nerambayé Damien et partout ailleurs en Algérie.*

*A mes amis et cothurnes Traoré Yéra, M. T. Bathily, A. Coulibaly, Sylla P, les frères Abdel Salam.*

*A ma grande famille celle de la communauté estudiantine malienne de Blida vous m'avez fait passer de bons moments.*

*A toute ma promotion de Master 1 et Master 2 de l'université de Blida1*

*A mes chers ami(e)s des Résidence universitaires la joie de vivre n'a jamais manquée.*

*A tous ceux qui de loin ou prêt participent ont participé*

*Traoré M. B*

*Manave E. F.*

# Résumé

Dans ce présent travail nous nous intéressons à la gestion sécurisée du système Evoucher qui représente une solution de distribution automatique et en temps réel de bons électroniques (vouchers) de vente de codes de recharges. Nous nous attaquons aux vulnérabilités encourues par les codes de recharges lors de leurs envoie au TPE (terminal de paiement électronique) pour leurs mise en vente.

Nous proposons une approche de sécurité hybride qui combine les cryptosystèmes symétriques et asymétriques tout en essayant d'exploiter les points forts de chaque cryptosysteme et en préservant les contraintes liées au système. Nous implémentons cette approche dans l'application back office du système Evoucher qui s'installe sur un serveur qui communique avec l'ensemble de TPE via GPRS.



**Mots clés :** EFTPOS, TPE, Evoucher, Algorithme de cryptage



# Abstract

In this work we are interested in the secured management of the voucher system that represents a solution for automatic and real time distribution of electronic tickets (vouchers) of sales of refilled codes. We're focusing on weakness of refilled codes when transmitting to the TPE (electronic payment by terminal) for theirs sale.

We propose a hybrid approach of security that combines symmetrical cryptosystems and asymmetrical cryptosystems by trying to exploit the stronger points of each cryptosystems and by preserving the system's restrictions. We implemented this approach in the Back-Office's application of the Evoucher system that will be installed on a server which communicates with all the TPE by GPRS.



**Keys Word:**EFTPOS, TPE, voucher, Algorithm of cryptage

# Sommaire

**Chapitre III : Sécurité et Approches de cryptages**



**Chapitre IV : Solution proposée**

# Liste de figures

**Chapitre I : Système Evoucher**



**Chapitre II : Modélisation**



**Chapitre III : Sécurité et Approches de cryptages**

# Liste de figures

## Chapitre IV : Solution proposée



## Chapitre V : Implémentation et Test

# Liste de tableaux

**Chapitre III : Sécurité et Approches de cryptages**



**Chapitre IV : Solution proposée**



**Chapitre V : Solution proposée**

## Introduction générale

Le rapprochement de l'informatique et des télécommunications a permis des avancées technologiques et ainsi favoriser l'apparition de nouvelles horizons scientifiques encadrant plusieurs domaines de la recherche et de l'industrie tel que : *e-business*, les télécommunications, le multimédia, les systèmes intelligents, l'aéronautique, l'aérospatiale, l'automobile, etc.

Avec l'arrivée des cartes dites intelligentes pour le domaine banquier et des télécommunications, l'industrie de fabrication des *TPE*(Terminaux de paiement électronique) s'est développée et croitre avec le développement des cartes à puces afin de faciliter les opérations et transactions effectuées sur les TPE, ainsi faciliter et améliorer les procédures économiques et industrielles dans le monde entier.

La communication entre la carte intelligente et le terminal de paiement ainsi que la consolidation des transactions électroniques effectuées par ces TPE a créé les systèmes *EFTPOS* ( Electronic Fund Transfert at Point of Sale) permettant la convergence directe vers le paiement électronique.

Cette évolution loin d'être néfaste, peut être une source d'efficacité économique en particulier dans un secteur tel que les télécommunications où les couts sont importants.

Aujourd'hui, les opérateurs télécom jouent un rôle indispensable dans notre vie quotidienne et professionnelle, cette stratégie des télécoms a conduit à imaginer de nouvelles idées qui se sont concrétisée par des produits réels.

Parmi les perspectives entreprises par cette stratégie et en vue d'alléger la charge d'accès aux produits et services des opérateurs télécom, des systèmes et plateformes sont établies, qu'on retrouve parmi le système *Evoucher* qui représente une solution de distribution automatique et à temps réel de bons électroniques (voucher)de vente de code de recharges.

# Introduction générale

## Problématique

La nécessité de trouver les moyens de protection de l'information digitale lorsqu'elle traverse des réseaux non sécurisés comme les réseaux internet ou les réseaux sans fils et de fournir de services sécurisés est un aspect qui de plus en plus gagne en ampleur dans la sécurité de l'information.

L'objectif de ce travail est d'arriver à proposer une approche de sécurité dans le back office du système Evoucher qui représente une solution de distribution des codes de recharge (vouchers ou tickets électronique). Confronté au défi d'assurer cette charge, la société HB Technologies aimerais une solution informatique pour assurer les contraintes liées aux différents aspects de sécurité des informations envoyées aux terminaux, d'administration, de traçabilité et de suivi par des rapports dans le système Evoucher.

## Objectif

Le but consiste à développer une application qui s'installera sur un serveur communicant avec un ensemble de TPE par GPRS/ADSL. L'application dans un cadre idéal devra permettre la gestion sécurisé des informations envoyées aux terminaux, leur administration mais aussi établir des rapports.

## Organisation du mémoire

Afin d'avoir une bonne vue, ce présent document sera organisé en chapitres définis de façon ordonné en vue de pouvoir suivre méthodiquement le développement de notre sujet mais aussi la compréhension de celui-ci par nos interlocuteurs. Nous distinguons cinq chapitres ordonnés comme suit :

Le premier chapitre définit le cadre de la réalisation du projet. Nous y définirons quelques principes et terminologies liées au thème.

Le second chapitre portera sur la modélisation de notre solution selon une démarche de conception.

Le troisième chapitre contiendra les différents principes et méthodes liées à la sécurité des données, on présentera un survol des différentes méthodes de cryptages ainsi qu'une comparaison des approches de cryptages symétriques et asymétriques.

Le quatrième chapitre sera consacré à la mise en œuvre dans le système Evoucher d'une approche d'authentification et de sécurité moyennant les approches de cryptage étudiées.

Le dernier chapitre sera dédié à la réalisation, l'expérimentation et aux tests de validation de notre solution.

Nous mettrons fin à ce mémoire par une conclusion et nous soulignerons également les perspectives futurs qui pourraient être entreprises pour de nouvelles réalisations et projets du même domaine.

# Entreprise d'accueil

## Présentation de l'entreprise d'accueil HB Technologies

L'entreprise HB Technologies dont le siège social est au niveau de la zone industrielle de Rouïba, est une société familiale de droit algérien créée en 2004 avec la mise en route effective des premières lignes de production en 2006[1].

Totalement opérationnelle depuis 2009, l'entreprise HB Technologies est en substance un centre de compétence dans la conception et développement de solutions à base de cartes intelligentes dans le domaine de la monétique et l'identitaire. Elle emploie actuellement un effectif de 102 personnes dont 50% environ sont des jeunes de l'université Algérienne encadrés par quatre (04) PhD revenant de l'étranger[2]. La force de l'entreprise réside dans son centre de recherche et développement et HB LAB, mis en place dès sa création. Elle est aussi l'une des rares entreprises qui font R&D dont le cout est important pour les équipements et les ressources humaines.

Le domaine de compétence de l'entreprise reste la fabrication de la carte à puce pour la téléphonie mobile, mais également pour le secteur bancaire ainsi que les cartes d'identification et d'authentification. Elle a pour stratégie ainsi qu'une sécurité supérieure et ceci en investissant sur le développement de l'engineering relative à la carte intelligente.

Même si elle n'est pas en situation de monopole, en qualité de fournisseur de puces sur le marché de la téléphonie mobile en Algérie, l'entreprise HB Technologie fournit aux trois opérateurs (03), que son Ooredoo, Mobilis et Djezzy, à hauteur de 65% de leur demande dans ce segment[3]. Faut-il souligner à ce titre, que les puces fabriquées actuellement correspondent à la 128 USIM, soit la 3G++, qui donnent la possibilité d'avoir deux(02) numéros sur le même SIM. L'entreprise est ainsi présente dans le secteur des télécommunications, plus particulièrement à travers, la fourniture de cartes de recharge hautement sécurisées aux principaux opérateurs de téléphonie mobile sur le marché algérien d'une part et, le développement par des compétences en interne et des partenaires de marque de solutions SIM/USIM économiques destinées aux opérateurs les plus avancées d'autre part. En outre l'entreprise HB Technologies, offre des solutions monétiques et toute une gamme de cartes bancaires produites et personnalisées selon les standards Mastercard et Visa International.



[1] Cf. http://WWW.hb-technologies.com.dz/fr/history.html.
[2] Cf. http://www.maghrebemergent.com. 22.01.2014
[3] Cf. http://www.maghrebemergent.com. 22.01.2014

# Chapitre I

# Système Evoucher

## 1. Terminologie et définitions

La compréhension fonctionnelle de notre système « ***Plateforme de Gestion Sécurisée du Système Evoucher***» prévoit une certaine connaissance des éléments qui composent le système globale Evoucher.

### 1.1 La carte à puce

Une carte intelligente munie d'un micromodule (contenant un système embarqué) répondant aux normes fixées par le standard ISO pouvant enregistrer et traiter de l'information.

### 1.2 Le Terminal de paiement électronique(TPE)

Est un appareil électronique capable de lire les données d'une carte à puce, d'enregistrer une transaction, de communiquer avec un serveur d'authentification à distance et aussi de pouvoir imprimer des reçus ou vouchers. Un TPE se compose essentiellement de:

- Microprocesseur
- Mémoire
- Lecteur de carte à puce
- Lecteur de carte magnétique
- Un Écran d'affichage
- Une imprimante
- Interface réseau (filaire ou sans fil)
- Un clavier pour introduire des informations dans le TPE

Dans le large spectre des TPE utilisés et selon les besoins de plusieurs domaines applicatifs, nous pouvons citer plusieurs types de terminaux **[1]**.

#### 1.2.1 Terminaux autonomes :

Les terminaux de paiement autonomes fonctionnent généralement de manière indépendante du système de caisse **[1]**.

#### 1.2.2 Terminaux intégrés

Les terminaux de paiement intégrés sont des solutions qui s'imposent à chaque fois que le terminal de paiement doit être incorporé dans un système de caisse existant **[1]**.

### 1.2.3 Terminaux mobiles

Les terminaux de paiement mobiles constituent la solution de paiement idéale en déplacement et partout où le terminal doit aller vers le client plutôt que l'inverse **[1]**.

## 1.3 GPRS

**General Packet Radio Service** se base sur la commutation par paquets, la norme GPRS (2,5G) permet d'accroître la bande passante sur le réseau GSM.

En théorie 10 fois plus rapide que le GSM, on constate des débits proches des connections RTC environ 50Kbit/s **[2]**.La technologie GPRS est utilisée dans le système Evoucher comme moyen d'échange de données entre le terminal (TPE) connecté au réseau et le serveur de la base des données, ceci est réalisé par une puce GSM placée dans le terminal.

## 1.4 Point of Sale (POS)

Désigne un emplacement de vente attribué à un marchand qui sera muni d'un TPE agrée par le système Back-office. Le POS ou point de vente a pour objectif d'éditer à la demande des tickets (**vouchers**) de recharge ou d'accès **[1]**.

## 1.5 Opérateur télécom

Cet élément de notre système est le producteur de code de recharges qu'il met en vente au niveau d'un point de vente par le biais du Back-office notamment le serveur d'application.

## 1.6 Code de recharge

Une séquence de 14 chiffres générés par des opérateurs télécom pour rechargé un mobile selon les commandes USSD (*xxx*Pin-digit#). Elles sont préalablement stockées au niveau du Back-office, et copié vers un POS sur une demande de celui-ci.

## 1.7 Voucher

Appelé encore ticket ou bon, est le résultat d'une opération électronique effectuée et peut être édité à la demande par le TPE. Le voucher après l'impression contient tous

les renseignements concernant le code PIN (code de recharge), son TPE, le gérant (Marchand).Cela se situe au niveau du Front-Office.

### 1.8 Distribution des codes de recharges dans le système Evoucher

Elle consiste à envoyer des codes de recharges aux terminaux.

### 1.9 Solution Evoucher

Avant d'arriver à destination les codes de recharges passent par les points suivants et ce dans l'ordre de citation qui suit:

- **L'opérateur** : envoie les codes de recharges au back-office dans un fichier qu'on dénote *Fichier de code de recharge.*
- **Le Back-office:** c'est le cœur du système, communique avec le Front-Office reçoit les fichiers de codes de recharges, enregistre dans la base de données.
- **Le Front-Office :** reçoit le fichier de codes de recharges.

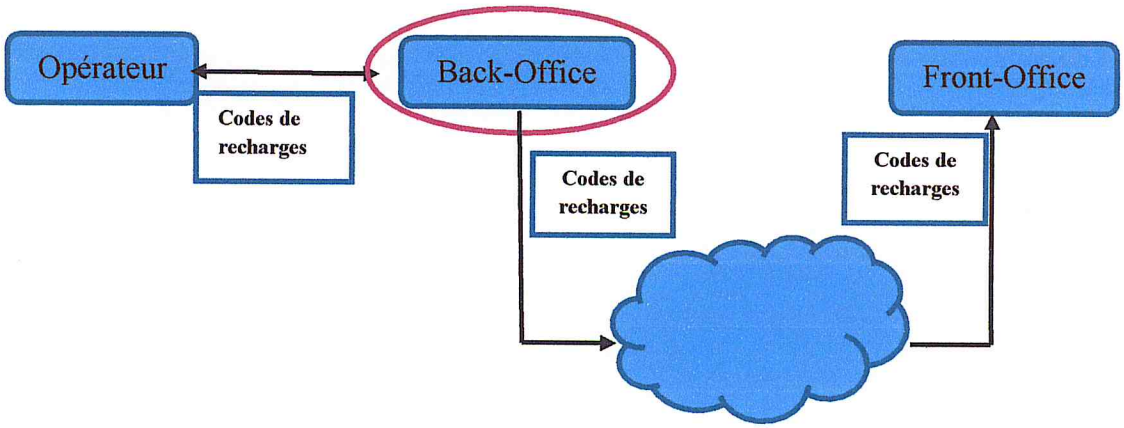



*Figure 1 : Schéma illustratif du système Evoucher*

### 1.10. Conclusion

Dans cette partie nous avons pu définir certaines notions plus particulièrement les éléments essentiels qui constituent le système Evoucher. Cela nous a permis de comprendre, de nous situer par rapport au champ d'étude et d'aborder le thème d'une façon générale et nous servira d'appuis pour les prochains chapitres.

# Chapitre II

# Modélisation

### 2.1. Introduction

Le succès du projet dépend dès lors de l'adéquation du projet au processus de développement qui est une étape importante pour l'élaboration d'une application indépendante de toute plateforme d'exécution et de tout langage de programmation. En effet, le processus de développement est constitué d'une succession de phases (spécification, conception et réalisation)

L'étude conceptuelle vise à mettre en place un système d'information capable d'atteindre des objectifs souhaités en tenant compte des suggestions formulées dans les objectifs au début de ce mémoire.

Nous déciderons dans cette partie du document du choix de la méthode de conception du système et du choix du langage de modélisation et ainsi que les différents aspects que nous aborderons.

## 2.2 Démarche Méthodologique :

### 2.2.1. Choix de la méthode de développement :

Dans le but d'assurer la qualité du produit logiciel plusieurs méthodes de développement furent élaborées. Après de recherches approfondies nous constatons que deux principaux méthodes sont utilisées lors de la conception d'une application : Les méthodes anciennes basées sur la séparation des données et des traitements, et les méthodes basées sur une approche orientée objet.

Pour notre part nous nous intéressons aux approches basées sur les méthodes orientées Objet notamment le modèle en V de UML.

### 2.2.2. Modèle en V :

Pour notre modélisation nous avons opté pour le modèle de *processus en v* qui pour nous est un démarche dite agile. Ce modèle grâce à sa flexibilité nous permettra de modéliser sans crainte, car nous ne pas voir tous les besoins (ceux du système) seulement au long de l'étape modélisation. De ce fait à l'implémentation nous pourrons faire des retours sur les étapes précédentes pour faire les modifications nécessaires

### 2.2.3 Motivation pour la démarche adoptée et langage de modélisation UML:

Nous avons choisi le modèle de processus en V centré autour du langage de conception UML du fait de l'avantage que constituent ces deux. Pour cette combinaison nous nous intéresserons à trois grands aspects : Fonctionnel, dynamique, et statique.

- **L'aspect fonctionnel :** Est composée principalement du diagramme de cas d'utilisation. Elle a pour rôle d'appréhender les interactions entre les différents acteurs(ou utilisateurs) et le système, sous forme d'objectif à atteindre d'un côté et sous forme séquentiel de scénarios d'interaction typiques de l'autre.

- **L'aspect dynamique :** Cette vue contient comme diagramme : d'activité, de séquence, de machine d'états, de collaborations. Cette vue vise à suivre l'état d'évolution des objets du programme tout au long de leur cycle.

- **L'aspect statique (structurelle):** Cette vue est composée des diagrammes de classes et de package. Cette vue est statique car on ne tient pas en compte le facteur temporel du système.

## 2.3 Analyse et conception :

Comme le veut le modèle en V nous allons procéder dans cette étape à l'effectuation du:

- recueil des besoins,
- l'analyse,
- L'implémentation,
- Le test.

### 2.3.1. Recueil des besoins :

Notre plateforme Back office de gestion des ressources du système Evoucher doit satisfaire et assurer un ensemble de fonctionnalités contenant dans ses modules soulignés comme suivant :

- **Module de sécurité :** Il contient un ensemble de fonctions pour la gestion d'accès, la vérification et l'authentification des codes de recharges, ainsi le cryptage des codes Pin et les clés symétriques.

- **Module d'administration :** Ce module sert pour la gestion et la création des objets et entités participant dans le fonctionnement de la solution Back office tel que : les terminaux de paiement ; les opérateurs, les marchands, et les utilisateurs du système.
- **Module de communication :** C'est un module intégré avec le système, il sert essentiellement a assuré la communication entre les différents terminaux et le serveur d'application.
- **Module statistique :** Pour l'édition des états statistique liés aux données (code de recharges), aussi ce module pourra jouer le rôle d'un outil d'aide à la décision.

Ainsi cela nous a permis de déduire les charges auxquelles notre application doit répondre:

- **Gestion des Terminaux ;**
- **Gestion Opérateurs ;**
- **Gestion comptes ;**
- **Etablir les rapports ;**
- **Gestion des utilisateurs ;**

Ce qui nous permet d'avoir le diagramme de cas d'utilisation global suivant :

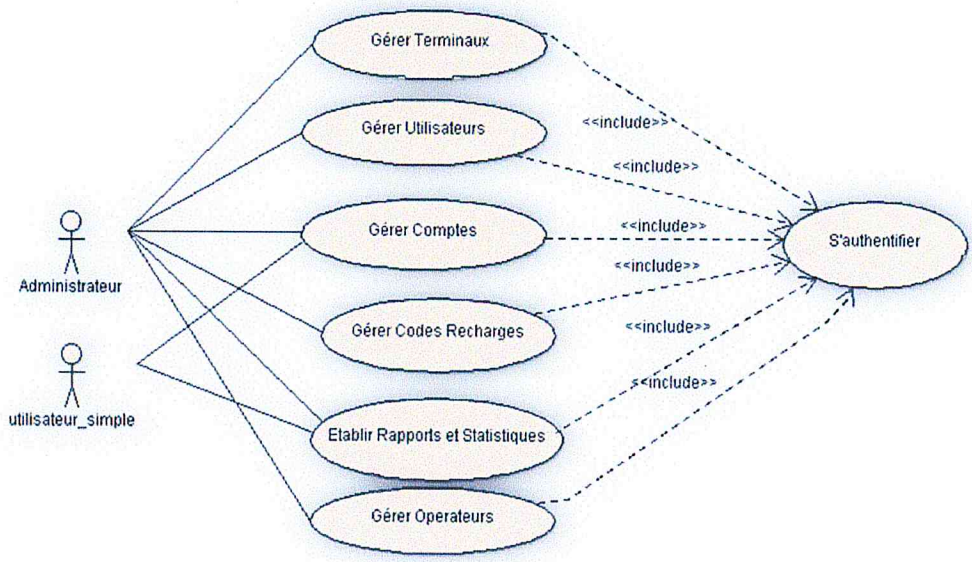



*Figure 2.1 Use case global*

Ce diagramme représente les cas d'utilisation de façon global sans en montrer les détails, chaque cas d'utilisation sera détaillé par la suite.

### 2.3.1. Description Fonctionnelle et structuration détaillée des cas d'utilisations

Dans la description qui suit nous présentons de façon détaillée les diagrammes de cas d'utilisation **[3]**. L'aspect dynamique sera vu dans les diagrammes de séquences par la suite.

#### 2.3.1.1.Cas d'utilisation « Gérer Comptes »

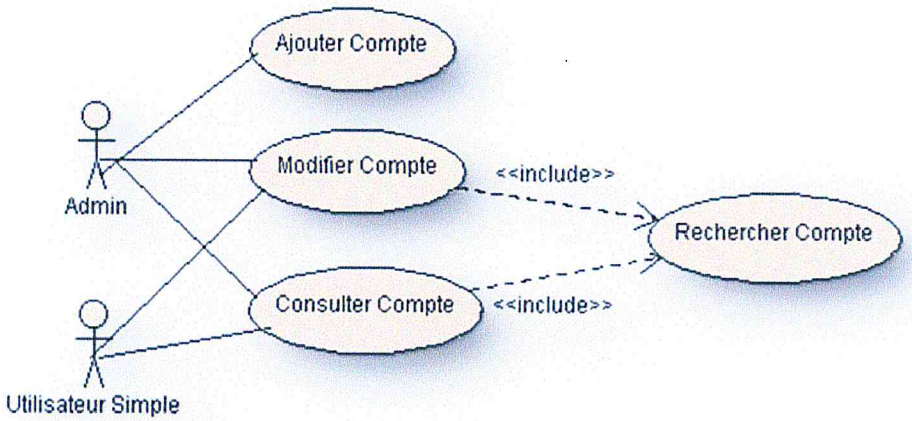



*Figure 2.2 Use case Gérer compte*

#### 2.3.1.2.Cas d'utilisation « Gérer Terminal »

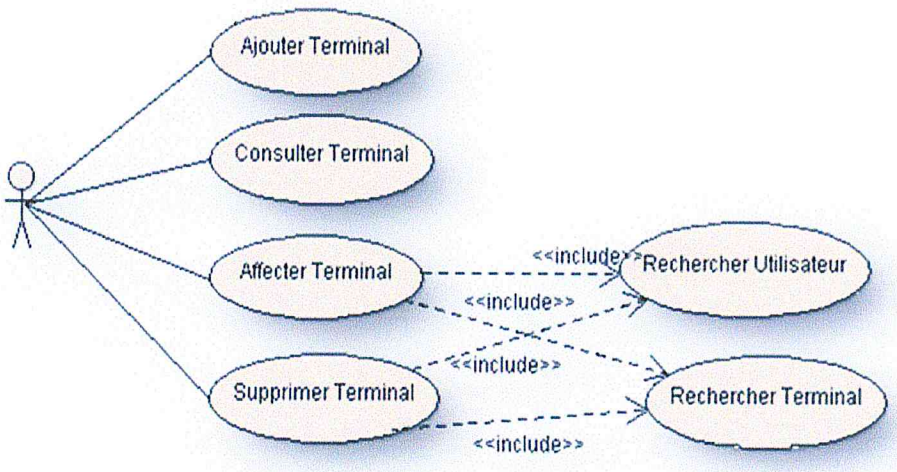



*Figure 2.3Use case Gérer terminal*

### 2.3.1.3.Cas d'utilisation « Gérer Code Recharges »

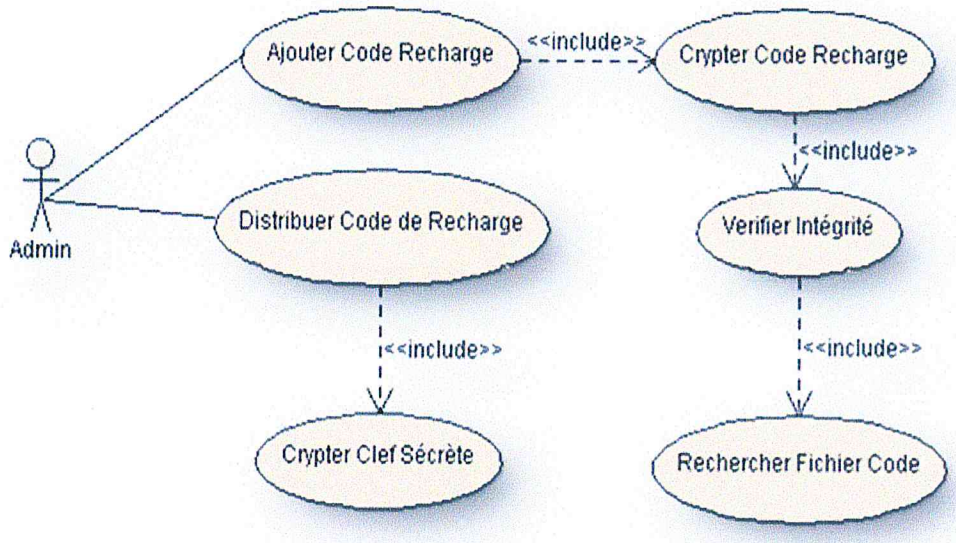



*Figure 2.3 Use case Gérer Code Recharges*

### 2.3.1.4.Cas d'utilisation « Gérer Utilisateur »

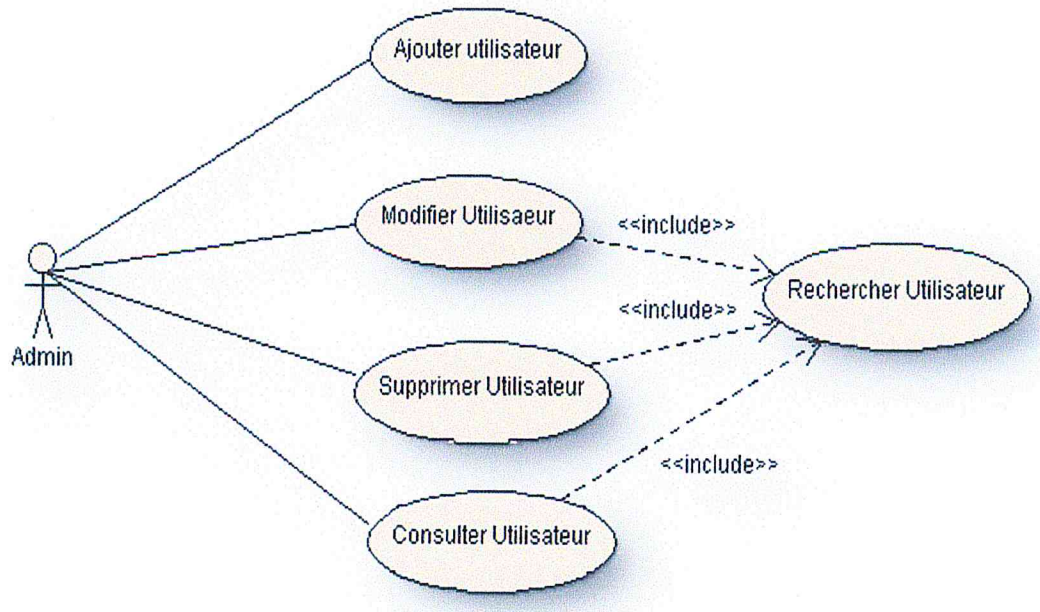



*Figure 2.4 Use case Gérer Utilisateur*

### 2.3.1.5.Cas d'utilisation « établir Rapport Statistique »

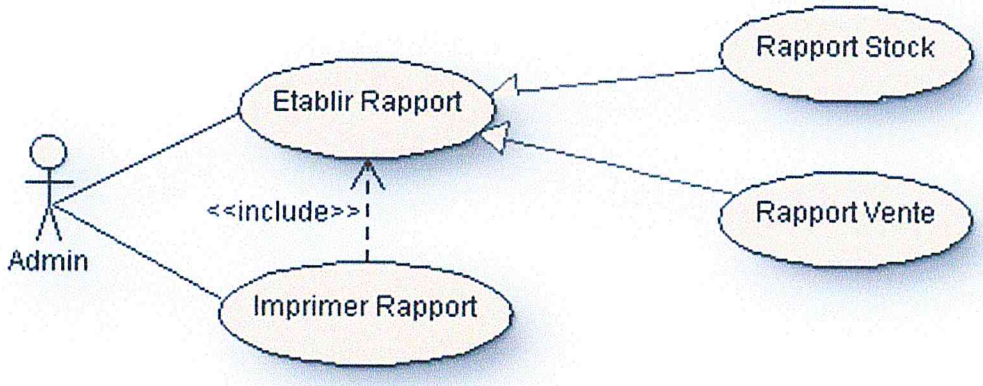



*Figure 2.5 Use case Rapport et statistiques*

### 2.3.1.6.Cas d'utilisation« Gérer Opérateurs»

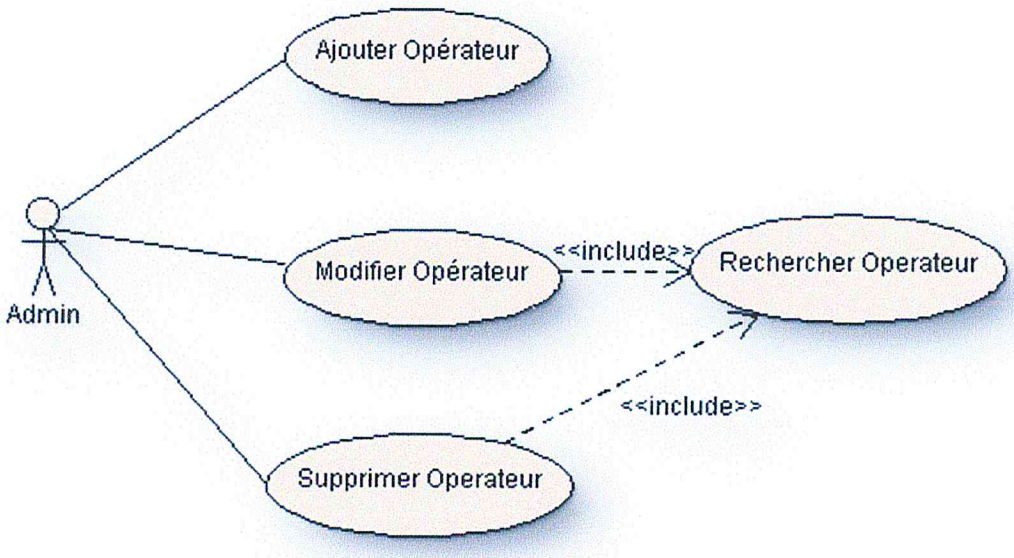



*Figure 2.6 Use case Gérer Operateurs*

## 2.4. Conception

Le diagramme de cas d'utilisation générale permet de voir en gros les fonctionnalités du système. Afin de rentrer en profondeur dans la description de notre système, nous nous proposons de décrire notre système suivant deux grands aspects. Il s'agira de décrire en se référant à la démarche RUP ; une vue statique du système par la modélisation de diagramme de classe puis une vue dynamique par la modélisation

des diagrammes de séquence. Cette description sera guidée par des cas d'utilisations comme l'indique la démarche.

### 2.4.1. Développement du modèle Dynamique :

Cette étape nous permettra de décrire les interactions entre les objets de notre application. Pour appréhender cela nous le matérialisons à l'aide de diagramme de séquence.

#### 2.4.1.1 Diagramme de séquences :

Le diagramme de séquence est un diagramme d'interactions entre objets, qui met l'accent sur l'ordre d'échange des messages aucours de l'exécution du système **[3]**.

Dans notre cas nous nous y intéressons pour représenter des scénarios de cas d'utilisation, des opérations du système.

##### 2.4.1.1.1 Diagramme de séquence du scénario « Authentification »

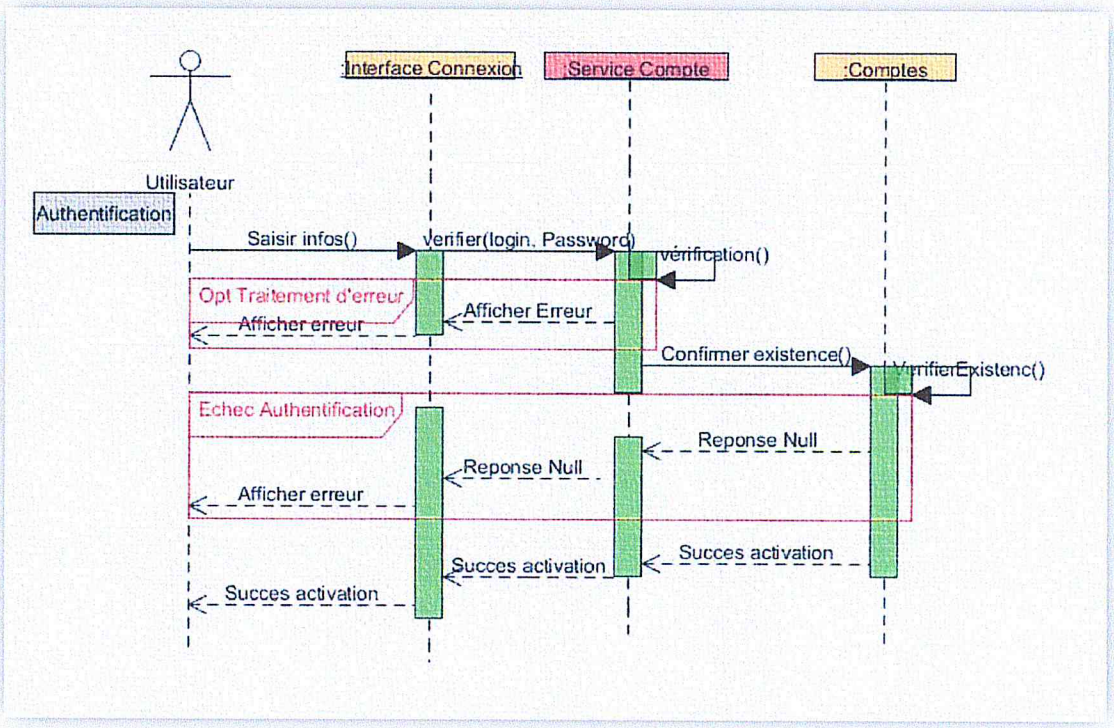



*Figure 2. 7 Schéma d'authentification*

#### 2.4.1.1.2 Diagramme de séquence du scénario « Modifier rechercher compte utilisateur »

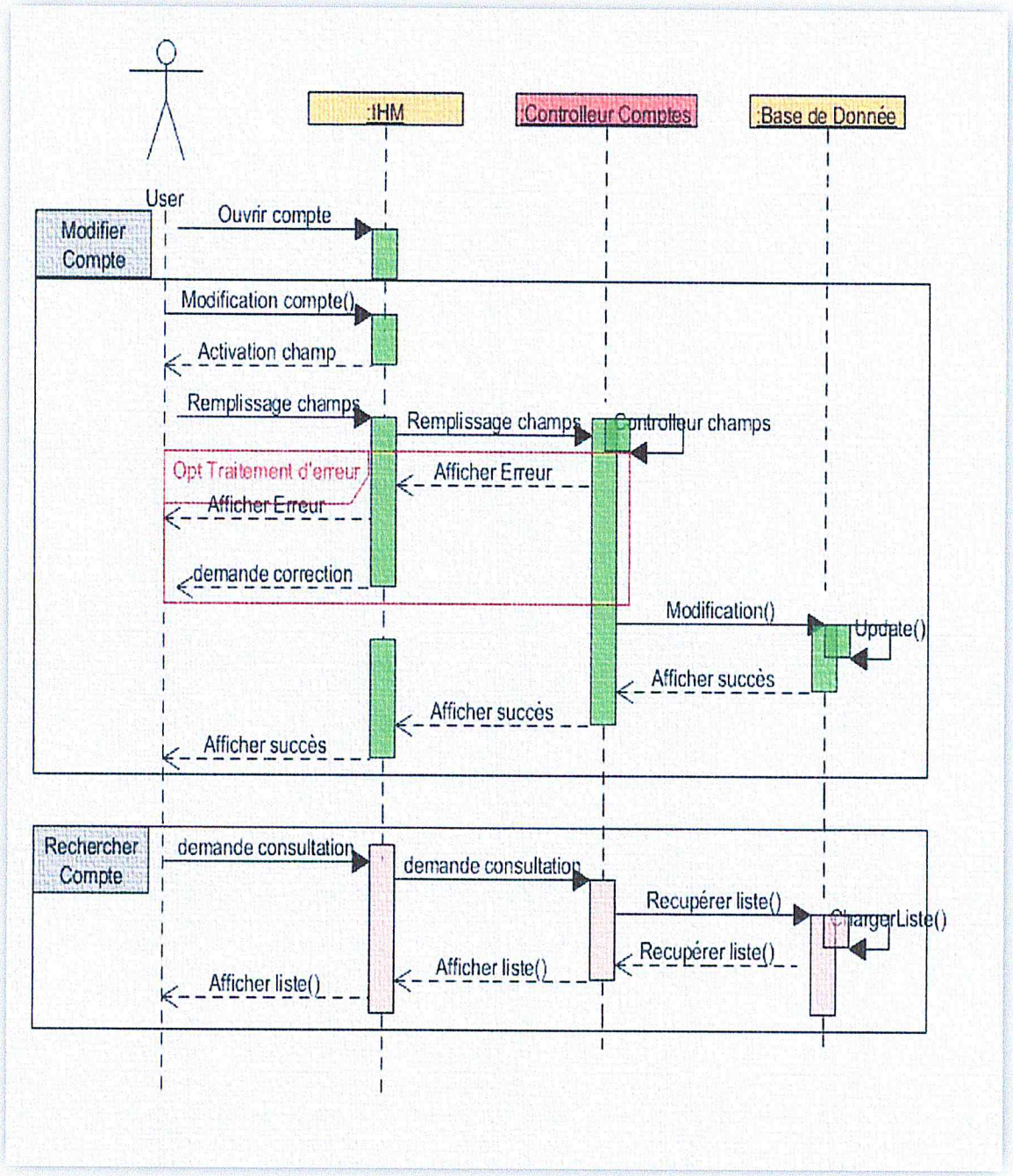



*Figure 2.8 Diagramme de séquence Modifier Compte*

### 2.4.1.1.3 Diagramme de séquence du cas « Gérer Utilisateur »

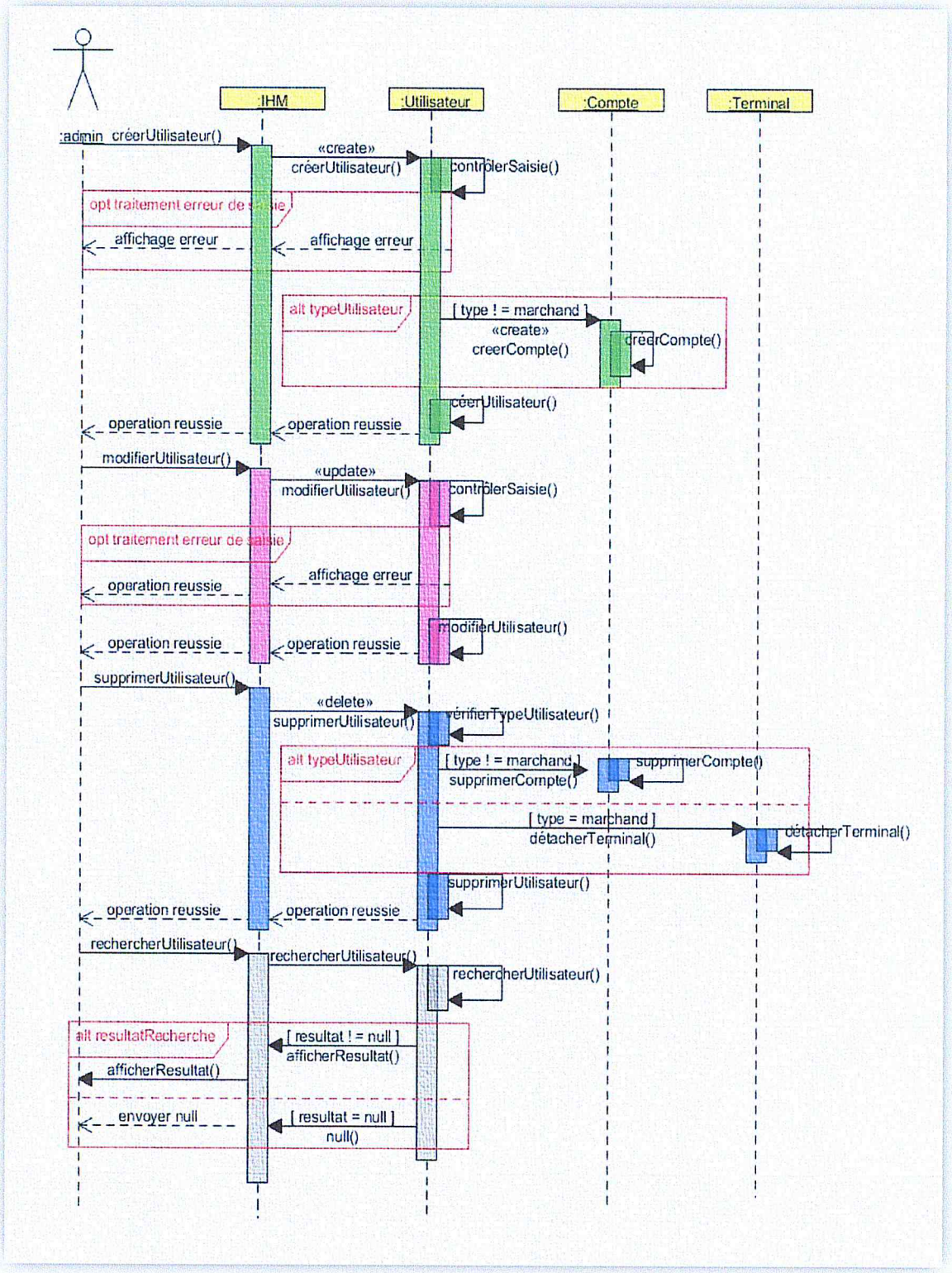
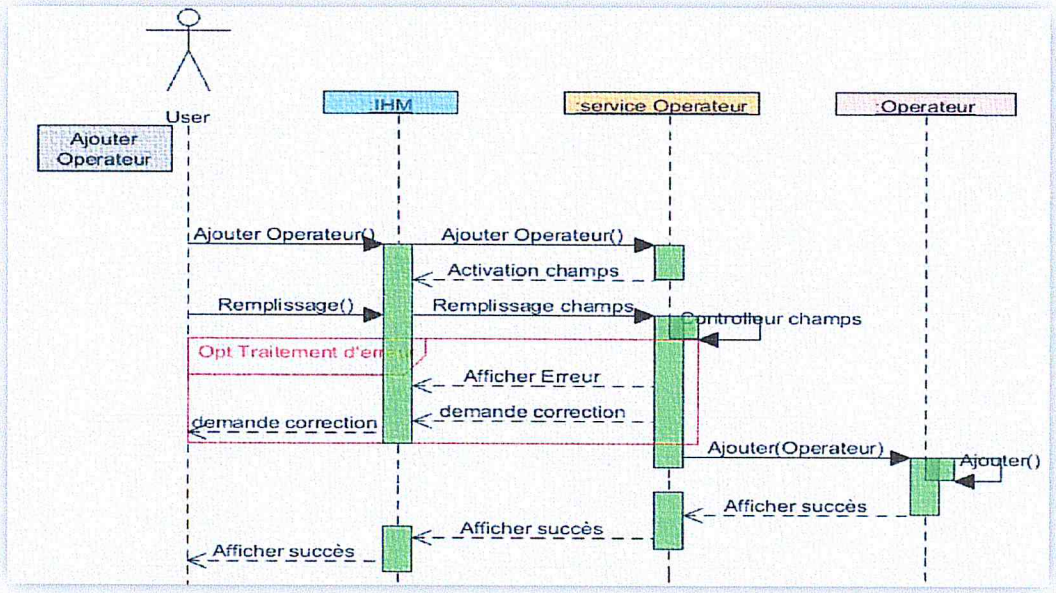



*Figure 2.9Diagramme de séquence Gérer Utilisateur*

#### 2.4.1.1.4 Diagramme de séquence « Gérer opérateur»



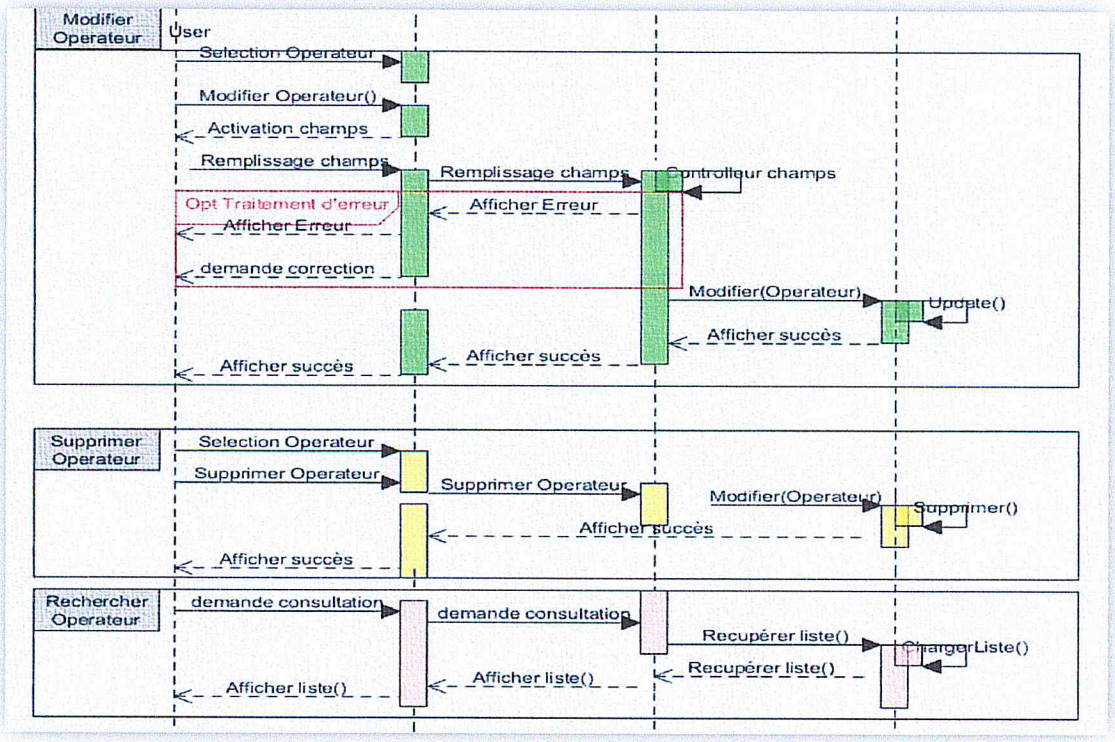




*Figure 2.10 Diagramme de séquences « Gérer Opérateur »*

#### 2.4.1.1.5 Diagramme de séquence « Gérer Code Recharges »

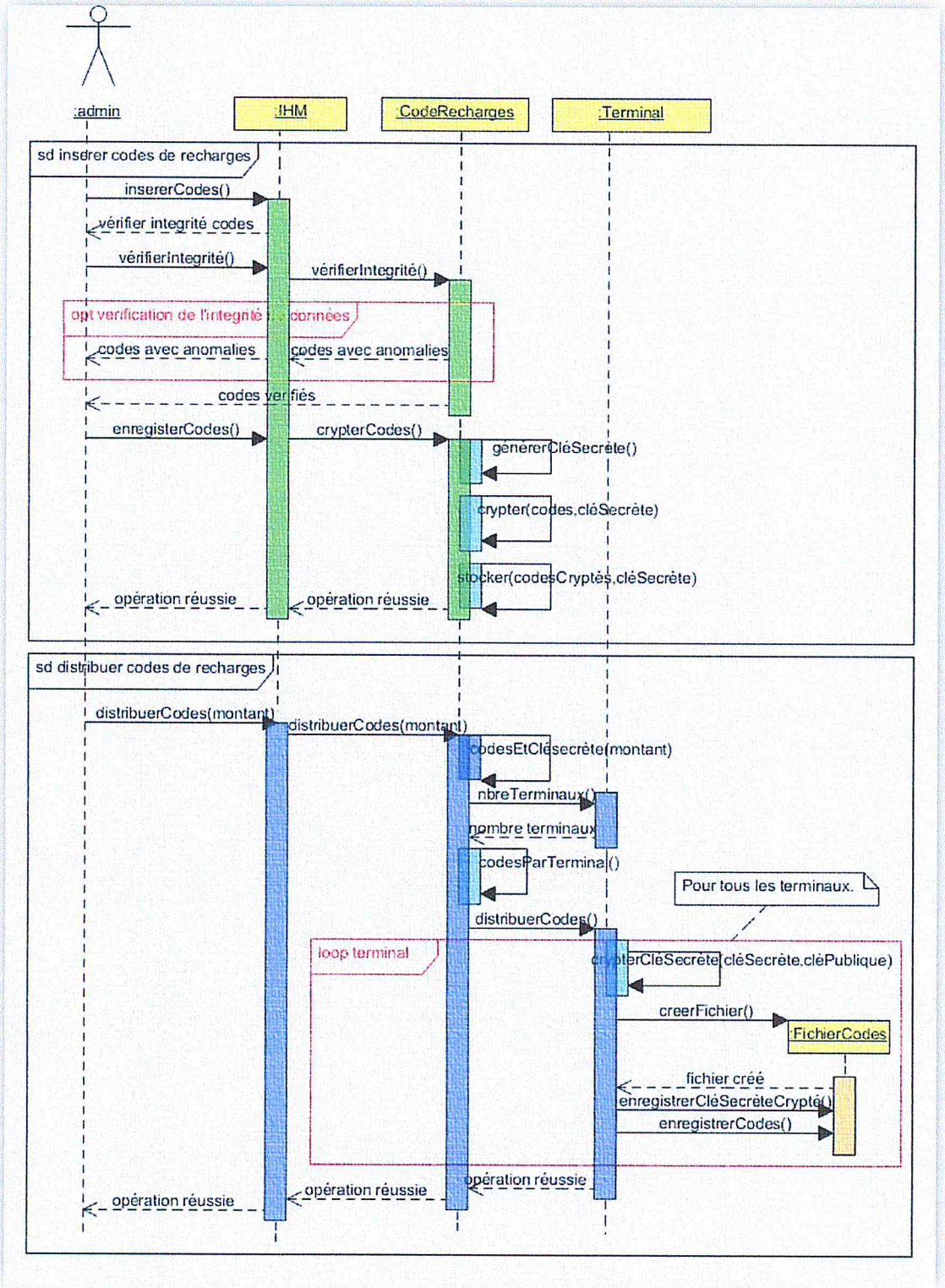



*Figure 2.11Schéma de diagramme de séquence de Gérer Codes recharges*

### 2.4.1.1.6 Diagramme de séquence « Gérer Terminaux »

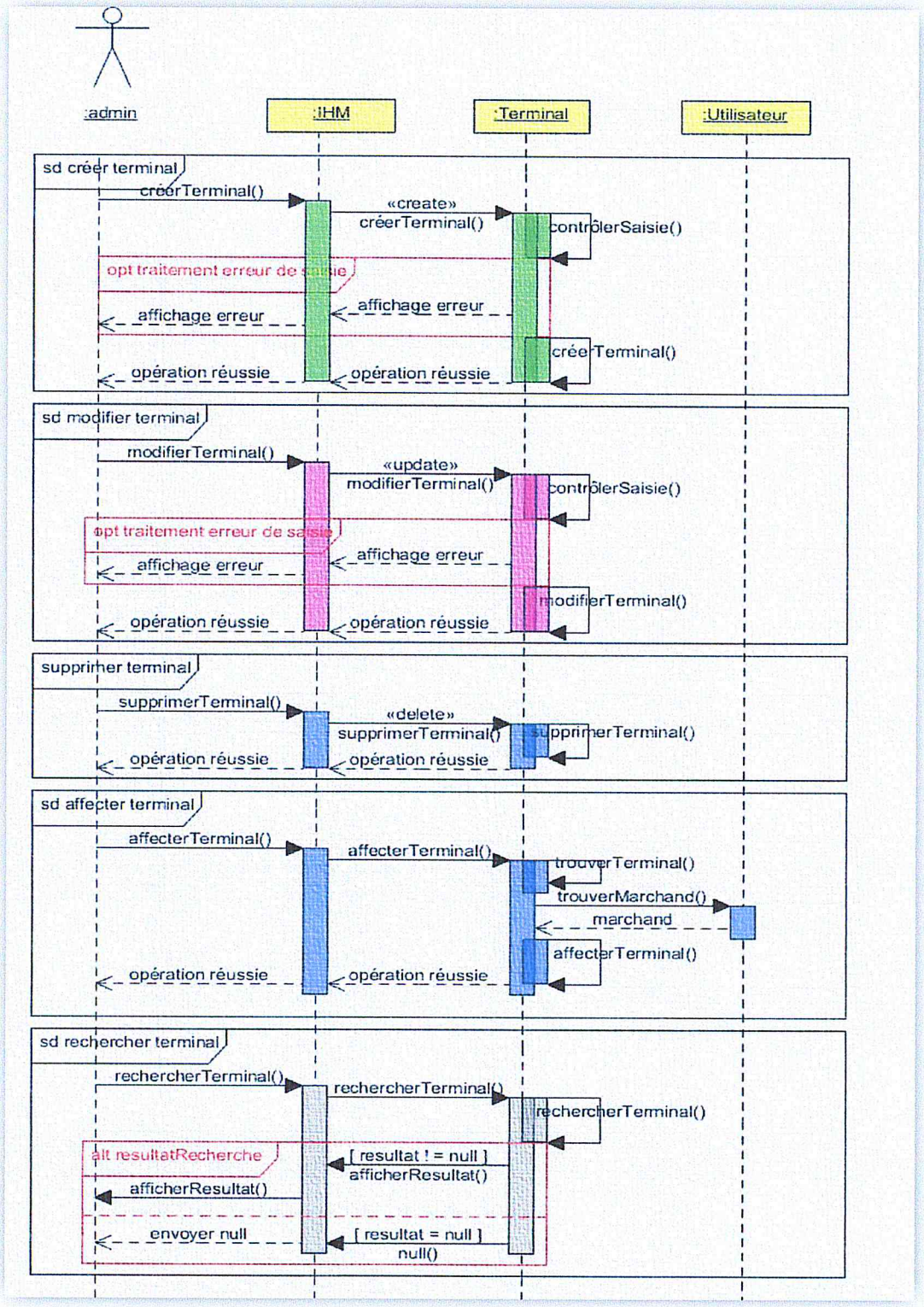



*Figure 2.12Diagramme de séquence Gérer Terminaux*

### 2.4.2. Développement du modèle Statique

C'est la deuxième étape de phase d'analyse après le recueil des besoins. Dans cette étape nous représentons les diagrammes de classes afin d'avoir une vue du système et aussi de savoir les classes entrantes en jeu.

#### 2.4.2.1.Diagramme de classe

Ce diagramme représente la description statique du système en intégrant dans chaque classe la partie dédiée aux données et celle consacrée aux traitements **[4].** C'est le diagramme pivot de l'ensemble de la modélisation d'un système car toutes les opérations de l'application se baseront sur lui. Nous donnons dans le diagramme de classe qui suit les tables qui rentrent dans la conception de notre système.

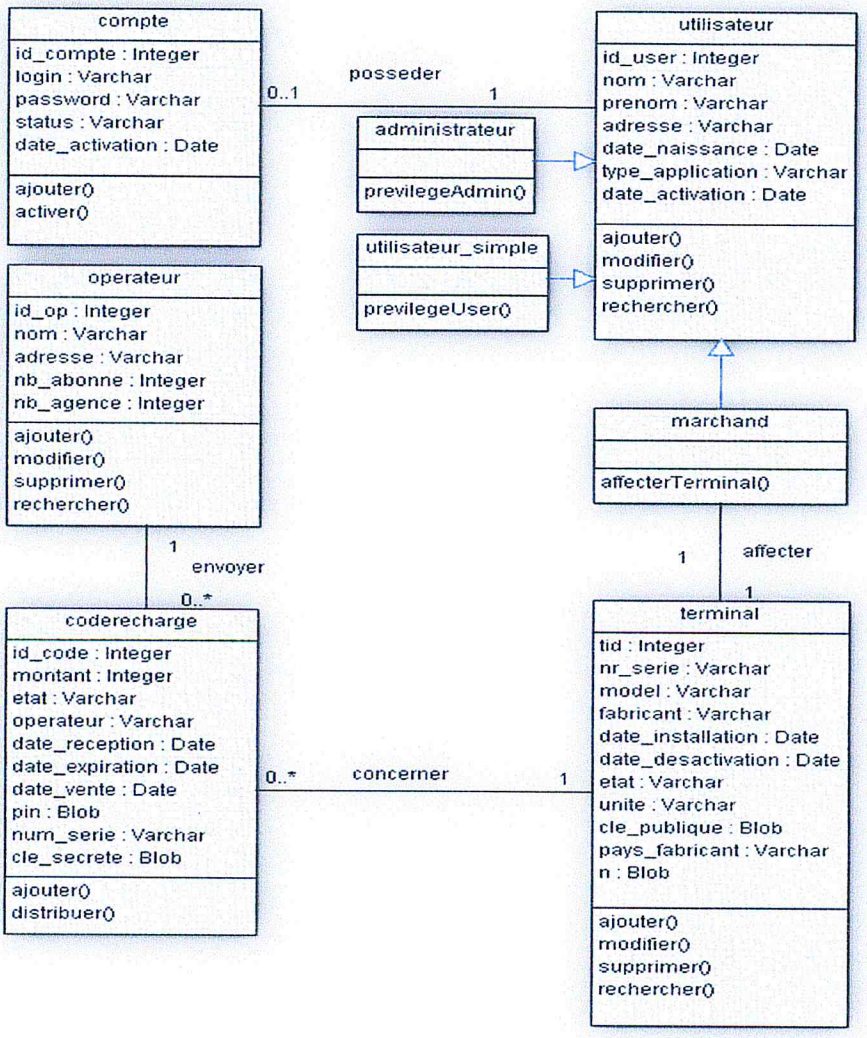



*Figure 2.13 Diagramme de classe du système*

#### 2.4.2.2.Passage vers le modèle relationnel

Dans notre SGBD relationnel nous aurons les classes suivantes :

**Compte (id_compte**, login, password, status,date_activation,*id_user) ;

**Utilisateur(id_user**, nom, prenom, adresse, date_naissance, type_user, type_application, date_activation) ;

**Operateur(id_op,** nom, adresse, Nb_abonne, Nb_agence) ;

**Coderecharge(id_code**, montant, etat, operateur, date_reception,* tid, *id_op, date_expiration, date_vente, pin) ;

**Terminal(tid**, nr_serie,model,fabricant,pays_ fabricant,date_installation, date_desactivation, etat, unite, *id_user,n, cle_publique) ;

## 2.5. Conclusion

Cette description est effectuée suivant plusieurs grands aspects permettant de pouvoir comprendre notre système sur les plans tant fonctionnels, dynamiques, que statiques.

Nous avons décrit le dynamisme de notre système à l'aide des diagrammes de séquences ; l'aspect statique à l'aide du diagramme de classe ; et tous cela guider par la méthode RUP cadré autour des cas d'utilisations que nous décrivîmes aussi.

Nous avons pu étudier notre système afin de pouvoir déceler les lacunes de celui-ci. Parmi ces lacunes nous nous sommes vus obligés d'assurer la sécurité des données transmises aux terminaux. Le prochain titre fera l'objet d'une étude des méthodes de cryptages possibles ainsi qu'une débauche sur une solution proposée.

# Chapitre III

# Sécurité et

# Approches de Cryptage

### 3.1. Introduction

La nécessité de trouver les moyens de protection de l'information digitale lorsqu'elle traverse des réseaux non sécurisés comme les réseaux internet ou les réseaux sans fils et de fournir de services sécurisés est un aspect qui de plus en plus gagne en ampleur dans la sécurité de l'information. Le commerce électronique est une réalité dont l'ampleur ne cesse de s'étendre et même si les problèmes de sécurité, de gestion, d'adaptation, et d'utilisation sont encore nombreux **[5]** ce domaine se développe et la question de transmission sécurisée de l'information digitale se pose car la vente, le paiement, l'achat ne peuvent pas se réaliser avec la procédure traditionnelle.

Le développement croissant de l'ouverture des réseaux filaires, non filaires, la prolifération des terminaux multiplient les points de vulnérabilité **[5] :**

- Usage frauduleux de terminaux ;
- Interception de données ;
- Attaques de nœuds de routage ;

#### 3.1.1. Sécurité des données lors de la transmission [6]

Assurer la sécurité de l'information digitale requiert une vaste gamme de techniques et des compétences. Toutefois, il n'y a pas de garantie que les objectifs de la sécurité jugés nécessaires peuvent être atteints adéquatement. Lors de l'échange d'information dans les architectures à aspect bilatéral, les menaces encourues par les données augmentent. La cryptographie apporte un certain nombre de fonctionnalités Permettant de pallier ces menaces à travers les propriétés: Confidentialité, Authentification, Intégrité des données, et la non répudiation.

- ***Confidentialité :*** La confidentialité des informations stockées ou manipulées par le biais des algorithmes de chiffrement. Elle consiste à empêcher l'accès aux informations qui transitent à ceux qui n'en sont pas les destinataires.
- ***Authentification :*** la transmission de l'information doit être authentifiée. Il faut pouvoir détecter une usurpation d'identité.
- ***Intégrité de données*** **:** Il est nécessaire de s'assurer que les informations seront bien transmises dans leur intégralité sans modification par un tiers.

- ***Non-répudiation*** **:** Il est nécessaire de s'assurer que les intervenants ne vont pas nier d'avoir émis ou reçu une information donnée.

Les moyens techniques sont fournis par la *cryptographie* **[7]**permettent d'assurer de la sécurité. De ce fait nous entamons dans le point suivant du chapitre l'étude de quelques méthodes cryptographiques les plus connues de cryptosystèmes symétriques et asymétriques présentes dans la littérature.

### 3.2 Systèmes cryptographiques

On distingue deux grandes classes de systèmes cryptographiques : *la cryptographie symétrique ou à clé secrète* et *la cryptographie asymétrique ou à clé publique*.

#### 3.2.1 Cryptographie à clé sécrète

Ce système de chiffrement à clef sécrète repose sur un principe tel que la clé de chiffrement et de déchiffrement sont les même c'est-à-dire $K_e=K_d=K$.

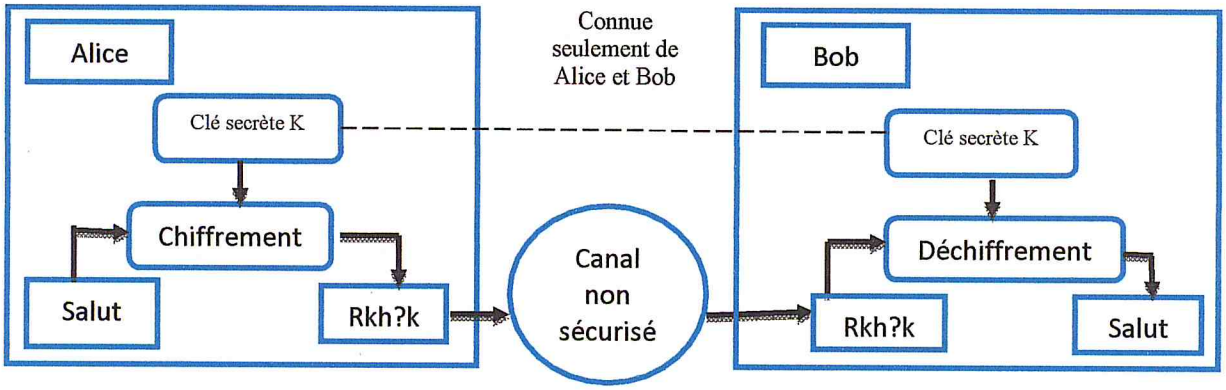



*Figure 3.1Principe de la cryptographie à clef secrète* **[6].**

#### 3.2.1.1.Le chiffrement à clef secrète par flot

Cette catégorie de chiffrement fait un traitement bit à bit des données. Leur utilisation repose sur un générateur de nombre pseudo-aléatoires et un mécanisme de substitution bit-à-bit rapide (figure ci-dessous). Les lettres K, M, C désignant respectivement *la clé, le message clair, le message chiffré.*

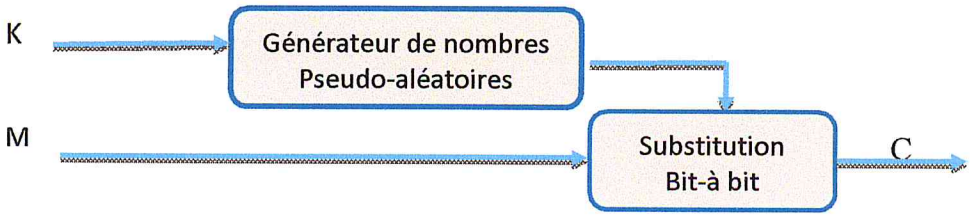



*Figure 3.2Chiffrement à clef secrète par flot***[6].**

### 3.2.1.2.Le chiffrement à clef secrète par blocs

Cette catégorie de chiffrement découpe le message M de n bits en S blocs de r=n/s. On désigne chiffrement à clefs sécrètes par blocs (block-cipher en anglais), tout système de chiffrement dans lequel le message clair est découpé en blocs d'une taille fixée, et chacun de ces blocs est chiffrée **[8].** Un bloc de r bits est encodé comme le montre la figure suivante pour donner un résultat.

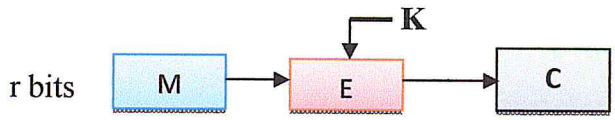



*Figure 3.3Chiffrement à clef secrète par bloc* **[6]**

L'utilisation de blocs pour chiffrer suppose que le message clair soit découpé en blocs de n bits préalablement au chiffrement. Ce découpage pose le problème du dernier bloc, dont la taille est généralement strictement inférieure à n bits. Il faut donc le compléter sans affaiblir la sécurité du message en ajoutant des caractères sans signification afin que sa taille soit un multiple de *r* **[6]**.

### 3.2.1.3.Méthodes cryptographiques à clefs sécrètes

#### 3.2.1.3.1 Data Encryption Standard (DES)[7]

DES (*figure 2.4)* est un algorithme de chiffrement symétrique par blocs qui permet de chiffrer des mots de 64 bits à partir d'une clef 64 bits dont 8 servent à vérifier l'intégrité de la clef. Cet algorithme tire son origine des travaux menés par le groupe cryptographique d'IBM dans le cadre du projet LUCIFER. Le DES a été objet de nombreuses implémentations à la fois en matériel et en logiciel, depuis sa publication.

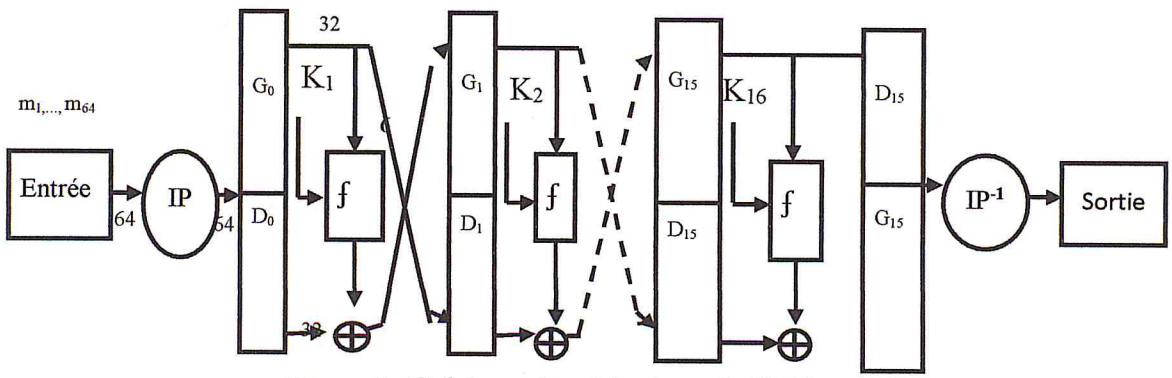



*Figure 3.4Schéma algorithmique de DES*

### A. Description des étapes

L'algorithme DES comprend les étapes suivantes :

a. ***Permutation initiale*** *(figure 2.5)* : Chaque bit d'un bloc est soumis à la permutation initiale, pouvant être représenté par la matrice initiale PI.

b. ***Fractionnement du message*** *(figure 2.5)* : Une fois la permutation initiale réalisée, le bloc de 64 bits est divisé en deux blocs de 32 bits, notées G et D (pour gauche et droit respectivement). On note $G_0$ et $D_0$ l'état initial de ces deux blocs.

| PI | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

| $G_0$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| $D_0$ | | | | | | | |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

*Figure 3.5: Matrice PI et blocs $G_0$, $D_0$*

c. ***Ronde*** *:* Les blocs $G_i$ et $D_i$ sont soumis à un ensemble de transformations appelées rondes. Chaque ronde à son tour comprend les étapes suivantes :

- *Fonction d'expansion (figure 2.6):* Les 32 bits du bloc $D_0$ sont étendus à 48 bits grâce à une table d'expansion dans laquelle 32 bits sont mélangés et 16 d'entre eux sont dupliqués pour former un bloc de 48 bits nommé $D'_0$.
- *XOR avec la clef :* DES procède ensuite à un OU exclusif entre $D'_0$ et la première clé $K_1$ générée à partir de la clé K (celle qui doit être partagée entre l'émetteur et destinataire). On appelle $D''_0$ le résultat de cette opération.
- *Fonction de substitution (figure 2.6):* $D''_0$ est découpé en 8 blocs de 6 bits noté $D''_{0i}$. Chacun de ces blocs passe par des de fonctions substitution.

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

| Fonction de substitution | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 2 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 3 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

*Figure 3.6Fonctions d'expansion et de substitution*

Les premiers et derniers bits de chaque $D''_{0i}$ détermine(en binaire) la ligne de la fonction de sélection et les autres bits détermine la colonne. Grace à cette information, la fonction de sélection prend une valeur codée sur 4 bits.

Chacun des 8 blocs de 6 bits est passe dans la fonction de sélection correspondante, ce qui donne en sortie 8 valeurs de 4 bits chacune. Ces valeurs sont regroupées pour former un bloc de 32 bits.

d. ***Permutation*** *(figure 2.7)* Le bloc de 32 bits est soumis à une transposition P.
e. ***OU exclusif*** *:* Le bloc de 32 bits ainsi obtenu est soumis à un OU exclusif avec le $G_0$ de départ pour donner $D_1$ et $D_0$ initial donne $G_1$.
   L'ensemble de ces étapes est itéré 16 fois.
f. ***Transposition initiale inverse*** *(figure 2.7)* : Au bout des seize itérations, les deux blocs $G_{16}$ et $D_{16}$ sont 'recollés', puis soumis à la permutation initiale inverse $PI^{-1}$. On obtient alors le bloc initial chiffré.

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

| $PI^{-1}$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

*Figure 3.7 Transpositions P et $PI^{-1}$*

### B. Algorithme de cadencement des clefs

Dans un premier temps les bits de parité de la clé sont éliminées afin d'obtenir une clé de longueur de 56 bits. La première étape alors consiste en une permutation notée CP-1. La matrice résultante est découpée pour obtenir 2 blocs de 28 bits (*figure 8).*

| $CP^{-1}$ | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

*Figure 3.8: Transposition $CP^{-1}$*

Ces deux blocs subissent une rotation à gauche, c'est à dire que les bits en seconde position prennent la première position, ceux en troisième position prennent la seconde, ceux en première position prennent la dernière et ainsi de suite. Les deux blocs sont regroupés pour faire un bloc de 56 bits qui passe par une permutation fournissant un bloc de 48 bits représentant la clef $k_i$ (*figure 2.9*).Des itérations de l'algorithme permettent de donner les 16 clefs utilisées dans l'algorithme DES.

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

*Figure 3.9 : Représentation de la clef $k_i$*

#### 3.2.1.3.2 Advanced Encryption Standard(AES)

AES est un algorithme de chiffrement par blocs à plusieurs tours similaire à DES mais avec une taille de blocs et de clefs supérieures et variables, choisis entre 128, 196 et 256 bits. **[9]**. Cet algorithme est un standard de la cryptographie symétrique dénommée Rijndael conçu par deux belges Joan Daemen et Vincent Rijmen. Avant de passer à l'algorithme nous cernerons quelques concepts de celui-ci.

a) **Un état :** On appelle état un bloc vue comme un tableau de $4 \times N_b$ où $N_b$ est égal à la taille du bloc/32(figure 2.10). On représente la clé de la même façon

(figure 2.11). Le nombre de colonnes étant $N_k$=*longueur de la clef/32*. Aussi vue comme étant le résultat d'un chiffrement intermédiaire **[10]**.

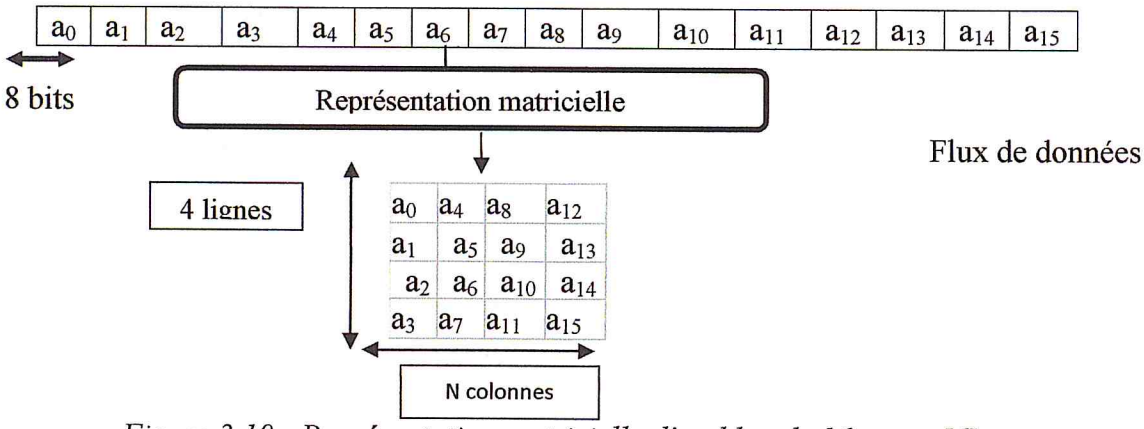



*Figure 3.10 : Représentation matricielle d'un bloc de 16 octets* **[6]**.

| | | | |
|---|---|---|---|
| $K_{0,0}$ | $K_{0,1}$ | $K_{0,2}$ | $K_{0,3}$ |
| $K_{1,0}$ | $K_{1,1}$ | $K_{1,2}$ | $K_{1,3}$ |
| $K_{2,0}$ | $K_{2,1}$ | $K_{2,2}$ | $K_{2,3}$ |
| $K_{3,0}$ | $K_{3,1}$ | $K_{3,2}$ | $K_{3,3}$ |

*Figure 3.11 : Représentation clef de longueur 128 bits* ***[6]***.

### b) Nombre de tours

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

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

*Tableau 3.1 Nombre de Tour en fonction de la taille des blocs et des clés* **[9]**

c) **Propriété fondamentale requise d'un algorithme de Chiffrement :** Un algorithme de chiffrement par bloc doit pouvoir être inversible. Pour que le message chiffré par un émetteur puisse être décrypté par le destinataire il aura besoin d'avoir le même algorithme mais de façon à inverser le processus de cryptage. Nous exposons les deux algorithmes (chiffrement et déchiffrement).

**A. Structure générale de l'algorithme de Chiffrement AES**

AES opère sur des blocs d'octets vus comme une matrice d'éléments. Le chiffrement AES consiste en :

- *Un Tour initial d'ajout de clé (Round key addition) ;*
  - *$N_r$ -1 Tours pour les opérations dans le même ordre sur un bloc;*
  - *Un Tour Final ($N_r$ tour Final Round) dans lequel on omet l'opération MixColumns;*

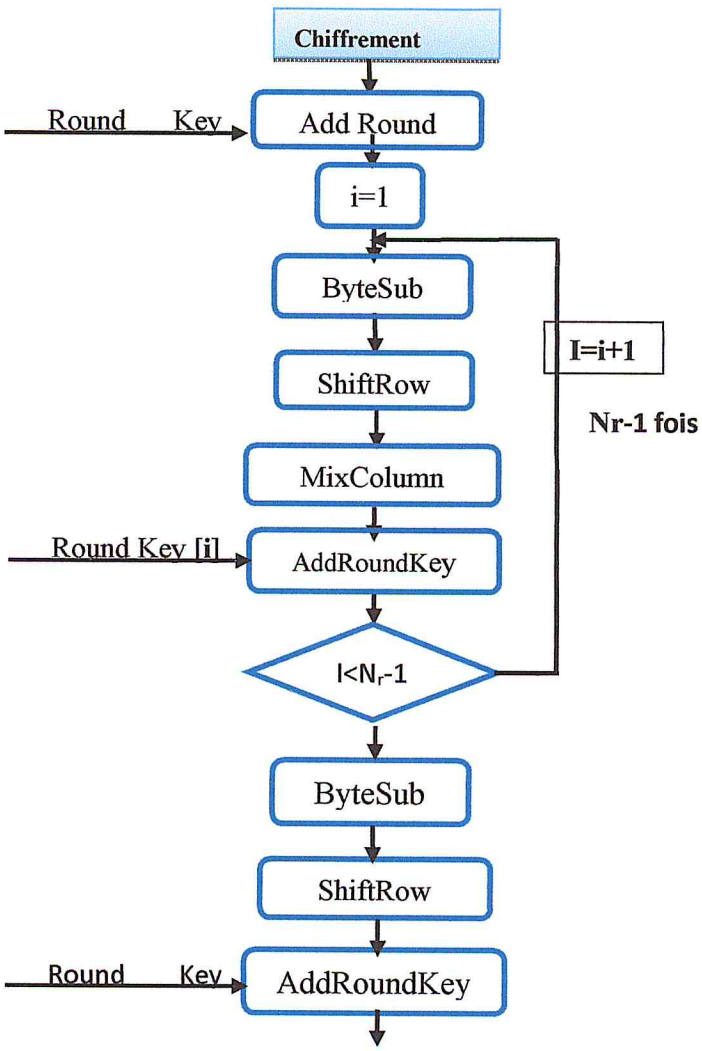



*Figure 3.12 : Structure générale d'un algorithme de chiffrement AES*

### B. Structure générale de l'algorithme de déchiffrement AES

Comme dans le cryptage, le décryptage opère aussi sur des blocs d'octets vus comme une matrice d'éléments. Le chiffrement AES consiste en :

- *Un Tour initial d'ajout de clé (Round key addition) ;*
- *Effectuer le $N_r$ nième Tour sans l'opération MixColumns;*
- *Effectuer les $N_r$ tours*

La figure ci-dessous en donne une explication plus détaillée.

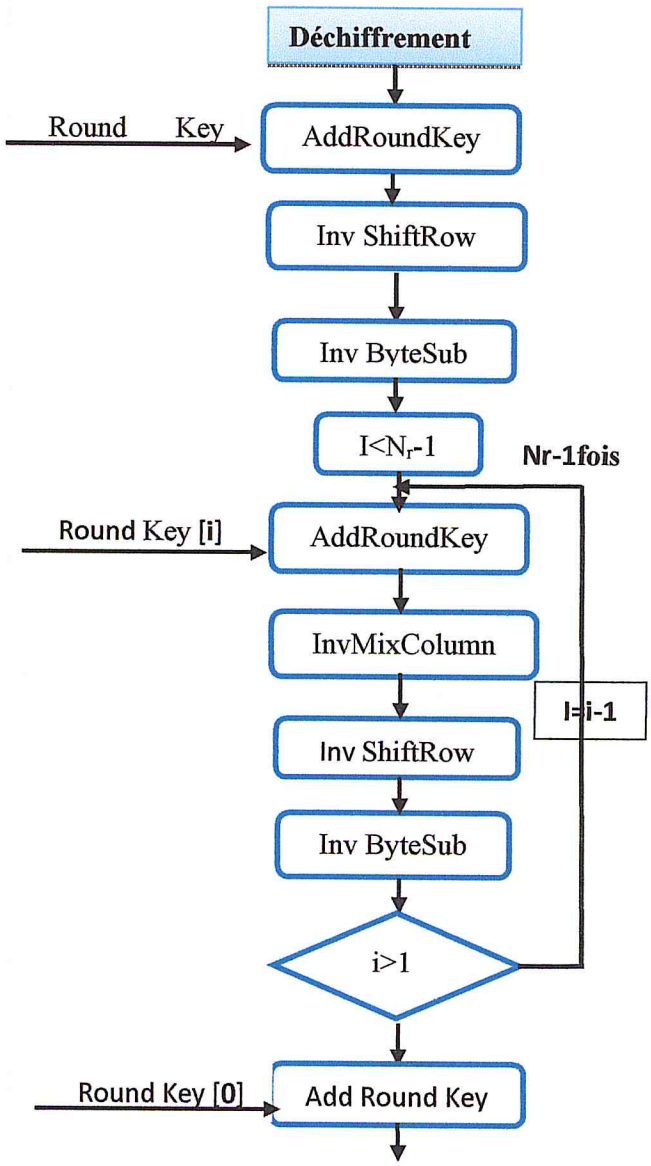



*Figure 3.13Structure générale d'un algorithme de déchiffrement AES*

### C. Description des étapes pour le cryptage

- ***SubBytes (état) :*** SubBytes (*figure 2.13)* est une substitution non-linéaire lors de laquelle chaque octet est remplacé par un autre octet choisit dans la table particulière appelé S-Box (*figure 2.14*) et opérant sur chaque octet de l'état indépendamment.

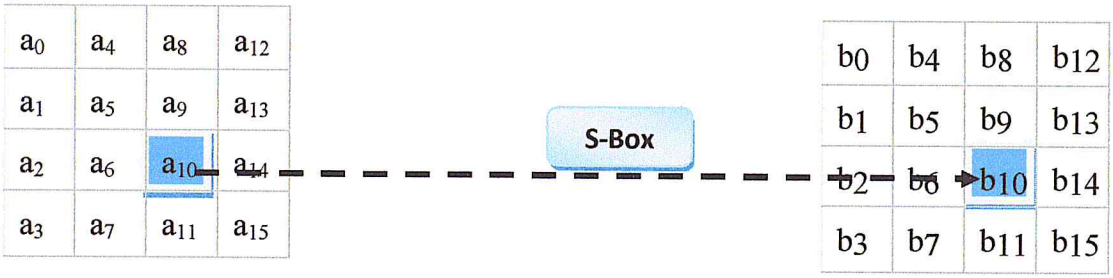



*Figure 3.14Transformation SubBytes.*

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 99 | 124 | 119 | 123 | 242 | 107 | 111 | 197 | 48 | 1 | 103 | 43 | 254 | 215 | 171 | 118 |
| 202 | 130 | 201 | 125 | 250 | 89 | 71 | 240 | 173 | 212 | 162 | 175 | 156 | 164 | 114 | 192 |
| 183 | 253 | 147 | 38 | 54 | 63 | 247 | 204 | 52 | 165 | 229 | 241 | 113 | 216 | 49 | 21 |
| 4 | 199 | 35 | 195 | 24 | 150 | 5 | 154 | 7 | 18 | 128 | 226 | 235 | 39 | 178 | 117 |
| 9 | 131 | 44 | 26 | 27 | 110 | 90 | 160 | 82 | 59 | 214 | 179 | 41 | 227 | 47 | 132 |
| 83 | 209 | 0 | 237 | 32 | 252 | 177 | 91 | 106 | 203 | 190 | 57 | 74 | 76 | 88 | 207 |
| 208 | 239 | 170 | 251 | 67 | 77 | 51 | 133 | 69 | 249 | 2 | 127 | 80 | 60 | 159 | 169 |
| 81 | 163 | 64 | 143 | 146 | 157 | 56 | 245 | 188 | 182 | 218 | 33 | 16 | 255 | 243 | 210 |
| 205 | 12 | 19 | 236 | 95 | 151 | 68 | 23 | 196 | 167 | 126 | 61 | 100 | 93 | 25 | 115 |
| 96 | 129 | 79 | 220 | 34 | 42 | 114 | 136 | 70 | 238 | 184 | 20 | 222 | 94 | 11 | 219 |
| 224 | 50 | 58 | 10 | 73 | 6 | 36 | 92 | 194 | 211 | 172 | 98 | 145 | 149 | 228 | 121 |
| 231 | 200 | 55 | 109 | 141 | 213 | 78 | 169 | 108 | 86 | 244 | 234 | 101 | 122 | 174 | 8 |
| 186 | 120 | 37 | 46 | 28 | 166 | 180 | 198 | 232 | 221 | 116 | 31 | 75 | 189 | 139 | 138 |
| 112 | 62 | 181 | 102 | 72 | 3 | 246 | 14 | 97 | 53 | 87 | 185 | 134 | 193 | 29 | 158 |
| 225 | 248 | 152 | 17 | 105 | 217 | 142 | 148 | 155 | 30 | 233 | 233 | 206 | 85 | 40 | 223 |
| 140 | 161 | 137 | 13 | 191 | 230 | 66 | 104 | 65 | 153 | 15 | 15 | 176 | 84 | 187 | 22 |

*Figure 3.15Transformation S-Box.*

- ***ShiftRows (état) :*** ShiftRows (*figure 2.15*) effectue un décalage des lignes de l'état courant. La ligne 0 n'est pas décalée, la ligne 1 l'est de $C_1$octets, la ligne 2 de $C_2$octets et la ligne 3 de $C_3$octets. Les valeurs de $C_1$ et $C_2$dépendant de la taille du bloc, selon la table (*tableau 3.2*) **[9]**.

| $N_b$ | $C_1$ | $C_2$ | $C_3$ |
|---|---|---|---|
| 4 | 1 | 2 | 3 |
| 6 | 1 | 2 | 3 |
| 8 | 1 | 3 | 4 |

*Tableau 3.2 Valeurs de $C_i$ en fonction de la taille du bloc.*

| Etat initial | | | |
|---|---|---|---|
| $a_0$ | $a_4$ | $a_8$ | $a_{12}$ |
| $a_1$ | $a_5$ | $a_9$ | $a_{13}$ |
| $a_2$ | $a_6$ | $a_{10}$ | $a_{14}$ |
| $a_3$ | $a_7$ | $a_{11}$ | $a_{15}$ |

| Etat après décalage | | | |
|---|---|---|---|
| $a_0$ | $a_4$ | $a_8$ | $a_{12}$ |
| $a_5$ | $a_9$ | $a_{13}$ | $a_1$ |
| $a_{10}$ | $a_{14}$ | $a_2$ | $a_6$ |
| $a_{15}$ | $a_3$ | $a_7$ | $a_{11}$ |

*Figure 3.16ShiftRows*

- ***MixColumns (état)*** : MixColumns (*figure 2.16*) effectue un produit matriciel en opérant sur chaque colonne de la matrice, vue comme vecteur **[6].**La transformation MixColumns consiste à prendre chaque colonne de l'état et à la multiplier par une matrice précise :

$$\begin{bmatrix} X_1 \\ X_5 \\ X_9 \\ X_{13} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \times \begin{bmatrix} X_1 \\ X_5 \\ X_9 \\ X_{13} \end{bmatrix}$$

*Figure 3.17MixColumns*

- ***AddRoundKey (état, $k_i$)*** : AddRoundKey (*figure 2.17*) consiste en un OU exclusif de l'état courant état et de la clef de tour $k_i$.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $a_0$ | $a_4$ | $a_8$ | $a_{12}$ | $k_0$ | $k_4$ | $k_8$ | $k_{12}$ | $b_0$ | $b_4$ | $b_8$ | $b_{12}$ |
| $a_1$ | $a_5$ | $a_9$ | $a_{13}$ ⊕ | $k_1$ | $k_5$ | $k_9$ | $k_{13}$ = | $b_1$ | $b_5$ | $b_9$ | $b_{13}$ |
| $a_2$ | $a_6$ | $a_{10}$ | $a_{14}$ | $k_2$ | $k_6$ | $k_{10}$ | $k_{14}$ | $b_2$ | $b_6$ | $b_{10}$ | $b_{14}$ |
| $a_3$ | $a_7$ | $a_{11}$ | $a_{15}$ | $k_3$ | $k_7$ | $k_{11}$ | $k_{15}$ | $b_3$ | $b_7$ | $b_{11}$ | $b_{15}$ |

*Figure 3.18 AddRoundKey*

### D. Diversification des Clés (KeyExpansion)

L'opération de KeyExpansion dépend de la taille du bloc choisi qui peut être de 128 jusqu'à 256 mais aussi de la taille de la clé aussi allant de 128 à 256.Elle consiste à générer pour chaque tour une clé de tour en fonction de la clé de sécrète k**[11].**La clé de chiffrement k (de $4N_k$ octets) dans une clé étendue W de $4N_b(N_r+1)$ octets. On disposera ainsi de $N_r+1$ clefs de tours (chacune de $4N_b$ octets).
L'algorithme utilisé pour la diversification de clef diffère légèrement selon que $N_k \leq 6$ ou $N_k > 6$. Les $N_k$ premières sont recopiés sans modification aux $N_k$ premières colonnes de W la clef étendue. KeyExpansion utilise notamment les deux fonctions : SubWord et RotWord.

- *SubWord* est une fonction prenant en entrée un mot de 4 octets et applique la boite S-Box**.**
- *RotWord* est une fonction prenant en entrée un mot de 4 octets $a=[a_0, a_1, a_2, a_3]$ et effectue une permutation circulaire pour renvoyer le mot $[a_1, a_2, a_3, a_0]$**.**
- Le tableau de constantes de tours de Rcon[i], indépendant de $N_k$ défini récursivement par : $Rcon[i]= [x^{i-1}, 00, 00,00]$, $\forall i \geq 1$

**E. Propriétés Cryptanalitiques**

D'un autre coté AES lui est très résistant à toutes les attaques connues. Très grande rapidité que ses autres congénères dans le cryptage et le décryptage. AES utilise des méthodes de substitution-permutation et non les diagrammes de Feistel (ou généralisations) ; possède une véritable structure mathématique. Toutes les attaques efficaces contre A.E.S sont des attaques contre de mauvaises implémentations logicielles ou matérielles. Très facile d'implémenter sur des cartes intelligentes en quelques lignes de codes et de petits critères de nécessité matérielle **[10]**.

### 3.2.2. Cryptographie à clé publique

L'idée de base de la cryptographie à clefs publiques a été proposée dans un article fondamental de Diffie et Hellman en 1976. Le principe consiste à utiliser des clefs de chiffrement et déchiffrement différentes, non reconstructibles l'une à partir de l'autre *:* une clé publique pour le *chiffrement* et une clé privée pour *le déchiffrement.*

Nous expliquons ce principe dans la figure (figure 2.17) ci-dessous.

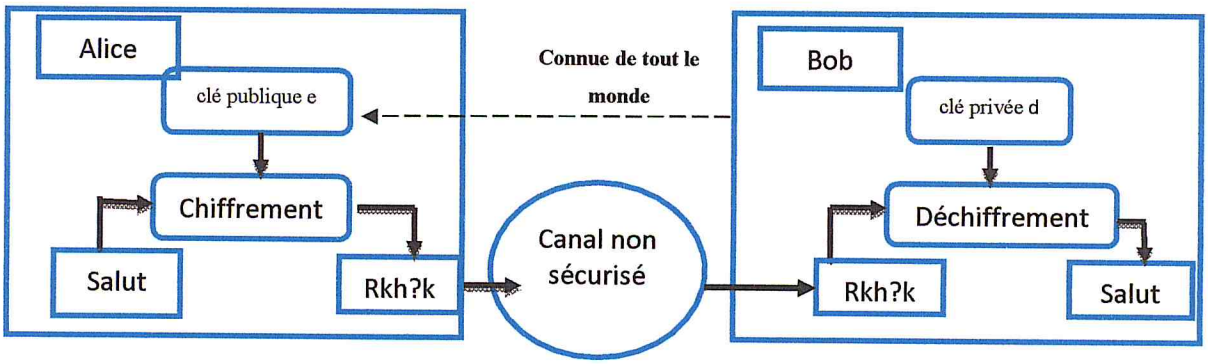



*Figure 3.19 Principe de la cryptographie à clef publique* **[6]**

### 3.2.2.1 La Méthode cryptographique RSA[7]

L'algorithme le plus célèbre d'algorithmes à clef publique a été inventé en 1977 par Ron Rives, Adi Shamir et Adleman, à la suite de la publication de l'idée d'une cryptographie à clef publique par Diffle et Hellman. Il fut appelé RSA, des initiales de ces inventeurs.

RSA est basé sur la difficulté de factoriser un grand nombre en produit de deux grands facteurs premiers.

***a. Génération de la clé publique***

1. Générer au hasard deux grands nombres premiers distincts p et q.
2. Calculer n= pq et $\phi$= (p-1)(q-1).
3. Sélectionner aléatoirement un entier e, 1 <e < $\phi$, tel que pgcd(e, $\phi$)= 1
4. Calculer d, 1 <d <$\phi$, tel que ed≡ 1(mod$\phi$ ) .
5. La clé publique est (n, e) et d est la clé privé

***b. Chiffrement du message***

1. Obtenir la clé publique (n, e).
2. Représenter le message sous la forme d'un ou plusieurs entiers m dans l'intervalle [0, n-1].
3. Calculer c= $m^e$ (mod n)

4. Envoyer le message crypté c.

**c. Déchiffrement**

3.1 Utiliser la clé privé d pour calculer $m=c^d \bmod n$.

### 3.2.2.2 ElGamal [6]

Cet algorithme repose sur l'utilisation du logarithme discret. Il a été publié par Tahar ElGamal en 1984. Cet algorithme est utilisé par le logiciel libre GNU Privacy Guard, PGP, et d'autres systèmes de chiffrement. Aucun brevet n'a été déposé, contrairement à RSA. Il peut être utilisé aussi bien pour le chiffrement que pour la signature électronique.

Soit Alice l'émetteur d'un message et Bob le destinataire. Bob possède deux clés :

- **Une clé privée** : un entier L
- **Une clé publique** : un grand nombre premier $p$ tel que trouver le logarithme discret dans le groupe $(Z_p)$ est difficile, un nombre α premier avec $p$ *et* $P=\alpha^s \bmod p$.

*a. Chiffrement :*

1. Alice tire aléatoirement un nombre k.
2. Elle calcule $C_1=a^k \bmod p$ et $C_2= MP^k \bmod p$

Alors le message chiffré est alors le couple $(C_1, C_2)$.

*b. Déchiffrement :*

A la réception du couple $(C_1, C2)$, Bob doit calculer $R_1=C_1^s mod\ p$ et après calculer la quantité $\frac{C2}{R1} = \frac{MPk}{Pk} = M$

### 3.2.3. Comparaison des cryptosystèmes symétriques et asymétriques

Les systèmes cryptographiques symétriques et asymétriques possèdent des avantages et des inconvénients (tableau 3.3). On peut exploiter les avantages de chaque système pour obtenir encore des meilleures solutions, on parle de *cryptosystèmes hybrides*.

| | Avantages | Inconvénients |
|---|---|---|
| **Cryptographie symétriques** | ✓ Facilement implémentables en Hardware et en Software<br>✓ Tailles de clefs relativement courtes<br>✓ Peuvent être composés pour produire de chiffrements complexes<br>✓ Historique plus élargi | ➢ La clé doit rester secrète entre l'émetteur et destinataire<br>➢ Dans la plupart de réseaux, il y a beaucoup de pairs à gérer |
| **Cryptographie asymétriques** | ✓ Seule la clé privée doit rester secrète<br>✓ Selon l'usage le pair des clés peut rester inchangé pendant des périodes considérables<br>✓ Dans la plupart de réseaux, le nombre de clés nécessaires peut être considérablement petit que dans un scenario à clé symétrique | ➢ Moins rapide que les meilleurs cryptosysteme symétrique connus<br>➢ Tailles de clés typiquement plus grandes que celles nécessaires dans un chiffrement symétrique<br>➢ Aucun système à clé public n'a été démontré (pareil pour chiffrement par bloc). Jusqu'aujourd'hui sécurité basée sur les problèmes de la théorie de nombres |

*Tableau 3.3Systèmes cryptographiques symétriques et asymétriques*

## 3.3. Conclusion

Dans ce chapitre nous nous sommes intéressés à la problématique de la transmission de données digitales et aussi à l'étude des quelques méthodes cryptographiques parmi les plus répandues de la cryptographie symétrique et asymétrique notamment AES, DES, RSA, ElGamal.

La cryptographie en se basant sur la puissance des mathématiques est sans équivoque un outil majeur pour la sureté de l'information surtout lors de sa transmission ou de son acquisition. Ses objectifs qui sont *la confidentialité, l'authentification, l'intégrité et la non répudiation* permettent d'assurer de la sécurité.

Ce chapitre et le précédent nous ont permis de recueillir et comprendre les éléments intervenants dans notre problématique, d'une part les aspects de la solution Evoucher dans son ensemble et d'autre part les aspects de la sécurité de la transmission de données digitales dans ce système. Ainsi nous pouvons dorénavant envisager une solution qui répondra à l'objectif de notre étude dans les prochains chapitres.

# Chapitre IV

# Solution proposée

## 4.1 Introduction

Après une profonde analyse dans le chapitre modélisation, nous avons pu déceler les problèmes que notre système pourrait encourir, spécifiquement sur le plan sécurité d'où la nécessité d'assurer cette dernière par des moyens adaptés.

Pour ce faire, nous avons dû faire une étude des approches de cryptages symétriques, asymétriques, cela dans le but d'avoir un aperçu de chacune.

Ainsi ce chapitre du mémoire sera consacré au choix et à la mise en œuvre de l'approche de cryptage hybride proposée, et des autres mesures de sécurités.

## 4.2.Transmission de données dans le système Evoucher

Après une étude des problèmes des systèmes de vente électroniques, nous avons pu constater que le système Evoucher pourra être aussi confronté aux mêmes dangers. Ci-dessous une vue globale du système Evoucher.

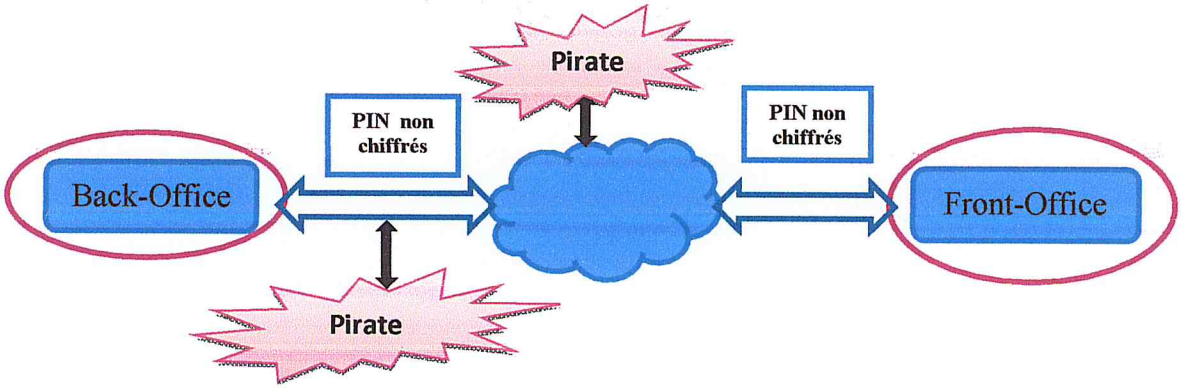



*Figure 4.1 Schéma illustratif de dialogue entre Back-Office et Front-Office*

Nous supposerons que la communication avec l'opérateur est suffisamment sécurisée en nous assurant d'une bonne transmission sans altération des codes PIN au Back-Office.

Afin d'assurer le maximum de sécurité nous tiendrons compte de :

- **L'authentification entre le Front et le Back-Office.**
- **La sécurité des informations dans le Back-Office.**
- **La sécurité des informations lors du transit entre le Front et Back-Office.**
- **La sécurité des informations au niveau du Front-Office.**

Par la suite nous décrirons les points qui furent soulignés récemment.

### 4.2.1. Authentification entre le Front et le Back-Office

Dans l'objet de garantir la validité des informations échangées entre les deux parties, il est nécessaire de vérifier les responsables de l'information échangé.

### 4.2.2. La sécurité des informations dans le Back-Office

Les codes de recharges qui sont fournis par l'opérateur au Back-Office doivent être invisibles jusqu'à leur réception au niveau du Front-Office.

### 4.2.3. La sécurité des informations lors du transit du Front et Back-Office

Pour garantir la qualité de la communication entre le Front-Office et le Back-Office nous devons sécuriser cette dernière.

### 4.2.4. La sécurité des informations au niveau du Front-Office

Une fois les informations présentes au niveau du Front-Office seul le terminal est habilité à déchiffrer et après à faire l'impression des vouchers pour le grossiste.

Les codes de recharges ne sont accessibles qu'à l'impression de voucher.

## 4.3.Solution hybride Proposée

### 4.3.1. Quelques critères à l'origine du choix de l'approche hybride

Après un constat rigoureux nous avons pu situer notre choix de l'approche selon les axes suivants:

- Vitesse de cryptage des données
- Sécurité de l'échange des données
- Disponibilité en temps réduit ou en temps réel des codes de recharges aux niveaux du Front-Office.

Dans le but de pouvoir apporter un maximum de sécurité dans notre système Evoucher et afin de tirer parti des avantages des algorithmes à clé secrète et des algorithmes à clé publique, nous optâmes pour un système hybride constitué à la fois d'algorithmes à clé secrète et à clé publique. L'échange de la clé secrète s'effectue grâce à l'algorithme à clé publique apportant une réponse à la question de l'échange

sécurisé de la clé. La communication qui s'ensuit est chiffrée grâce à l'algorithme à clé secrète, ce qui permet de bénéficier d'un système rapide**[12]**.

### 4.3.2. Principe

L'approche est assez simple à comprendre nous comblons ce lacune des approches de cryptages symétriques en y appliquant un cryptage asymétrique de la clef sécrète.

***Structure générale du principe hybride***

Nous allons décrire le fonctionnement sur deux côtés.

- **l'application du Back-Office**

⇒ *Générer la* ***clef sécrète Ks*** de façon *aléatoire et dynamique* ;

⇒ *Crypter les données à envoyer selon un* ***algorithme de cryptage symétrique*** *(notamment le standard AES) avec la clef $K_s$;*

⇒ *Crypter la* ***clef sécrète Ks*** *à l'aide d'un algorithme de cryptage asymétrique (à savoir)*

- **l'application du Front-Office**

⇒ ***DeCrypterclef sécrète Ks****avec d'un algorithme de cryptage asymétrique avec sa clef privée (celui du terminal).*

⇒ ***DeCrypter****les données reçues selonun****algorithme de cryptage symétrique*** *(notamment le standard AES) avec la clef* ***clef sécrète Ks.***

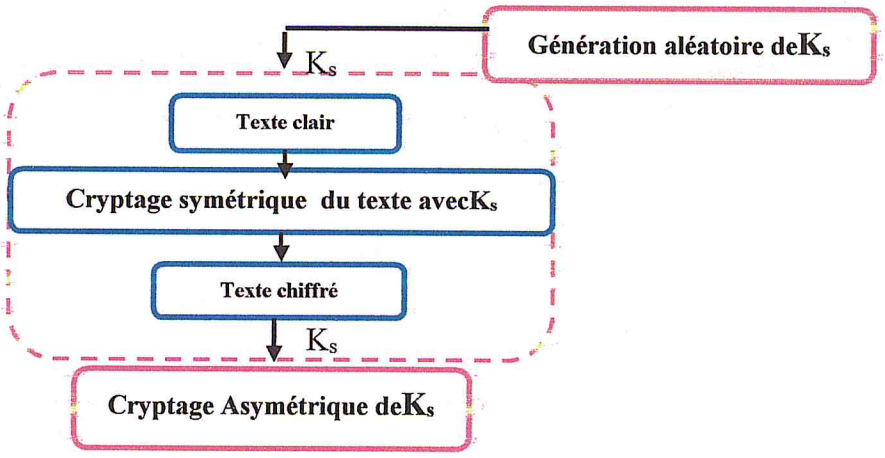



*Figure 4.2Schéma illustratif de principe de fonctionnement de l'approche hybride.*

### 4.3.3. Mise en œuvre de l'approche hybride dans la solution Evoucher

L'algorithme AES a une grande résistivité à toutes les attaques et une très grande vitesse de cryptage et décryptage *entier* **[12]**, nous l'utilisons pour le cryptage symétrique. RSA est le cryptosysteme à clé publique le plus utilisé et sa sécurité repose sur le *problème de factorisation d'un nombre,* nous l'utilisons pour le cryptage asymétrique. Nous définissons aussi des procédures algorithmiques dans le but de renforcer encore la sécurité.

| **Algorithmes cryptographiques** | | |
|---|---|---|
| **Algorithmes** | **Rôle** | **Eléments Evoucher** |
| **AES** | chiffrement de codes de recharge | Application (back-office) |
| | déchiffrement de codes de recharge | Terminal (front office) |
| **RSA** | chiffrement de la clé secrète | Application (back-office) |
| | déchiffrement de la clé secrète | Terminal (front office) |
| **Procédures** | | |
| **Procédures** | **Rôle** | **Eléments Evoucher** |
| **Générer Clé Secrète** | générer une clé secrète | Application (back-office) |
| **Vérifier Intégrité Codes** | vérifier intégrité de codes de recharges | Application (back-office) |
| **Distribuer Code de Recharge** | Distribuer codes de recharges pour chaque terminal | Application (back-office) |

*Tableau 4.1*choix algorithmiques

**Terminologie :**

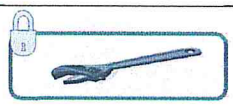
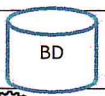
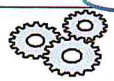
| **Symboles** | **Signification** |
|---|---|
|  | Clé publique |
|  | Clé secrète |
|  | Clé secrète cryptée |
|  | Clé privée |
|  | Base de données |
|  | Exécution des opérations |
|  | Fichier de codes de recharges |
|  | Fichier de codes de recharges cryptés |

*Tableau 4.2terminologie*

**4.3.4. Principe de fonctionnement** Le principe de base de notre approche consiste essentiellement à appliquer l'algorithme Back Office (algorithme 2) et l'Algorithme Front Office (algorithme 3) en faisant intervenir les éléments de la solution Evoucher de la manière suivante :

**4.3.4.1.Opérateur téléphonique** son rôle (Figure 3.3) dans la solution Evoucher envoi les fichiers de codes de recharges à l'application back-office.

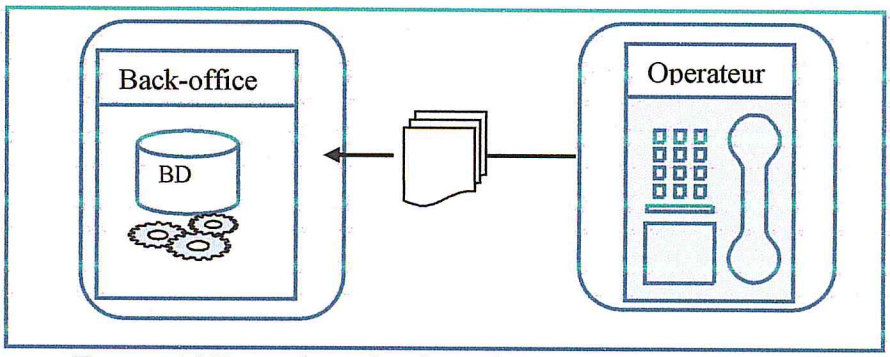



*Figure 4.3Envoi de codes de recharges au Back-office*

**4.3.4.2.Application back-office** : Dans ce processus l'application back-office utilise l'Algorithme Back Office (algorithme 2) pour exécuter les modules suivants :

**A. Module de cryptage et stockage de codes de recharge dans la base de données** (voir figure 3.4)**:** Ce module utilise l'algorithme AES pour crypter et les procédures générer clé secrète (procédure 3) et vérifier intégrité des codes (procédure 1) pour réaliser les taches suivantes :

***a.*** *Recevoir et garder les fichiers de codes de recharges envoyés par l'opérateur ;*
***b.*** *Générer aléatoirement une clé secrète;*
***c.*** *Crypter les codes de recharges reçus en utilisant l'algorithme AES;*
***d.*** *Stocker dans la base de données les codes de recharges cryptés et la respective clé secrète ;*

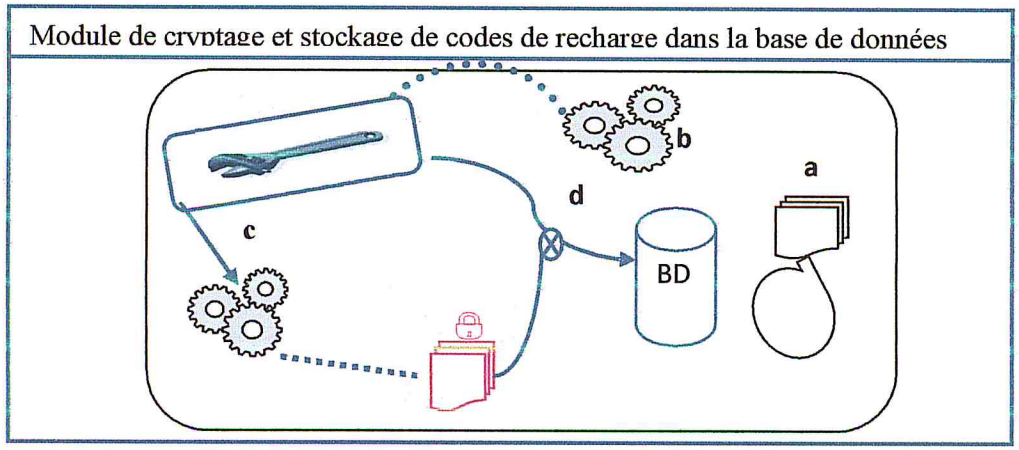


*Figure 4.4Cryptage et stockage de codes de recharge dans la base de données*

**B. Module de distribution de codes de recharge** (voir figure 3.5) :Ce module utilise l'algorithme RSA et la procédure distribuer codes recharge (procédure 2) pour réaliser les taches suivantes :

*Selon le montant précisé par l'utilisateur habilité :*

***a.****Récupérer les codes de recharges en stock dans la base de données et pour chaque terminal :*

***b.****crypter la clé secrète en utilisant l'algorithme RSA et la clé publique du terminal ;*

***c.****créer un fichier ;*

***d.****enregistrer dans le fichier la clé secrète cryptée et les codes de recharges dont le nombre est préalablement calculé ;*

***e.****Transmettre les fichiers de codes de recharges cryptés aux différents terminaux connectés au back-office ;*

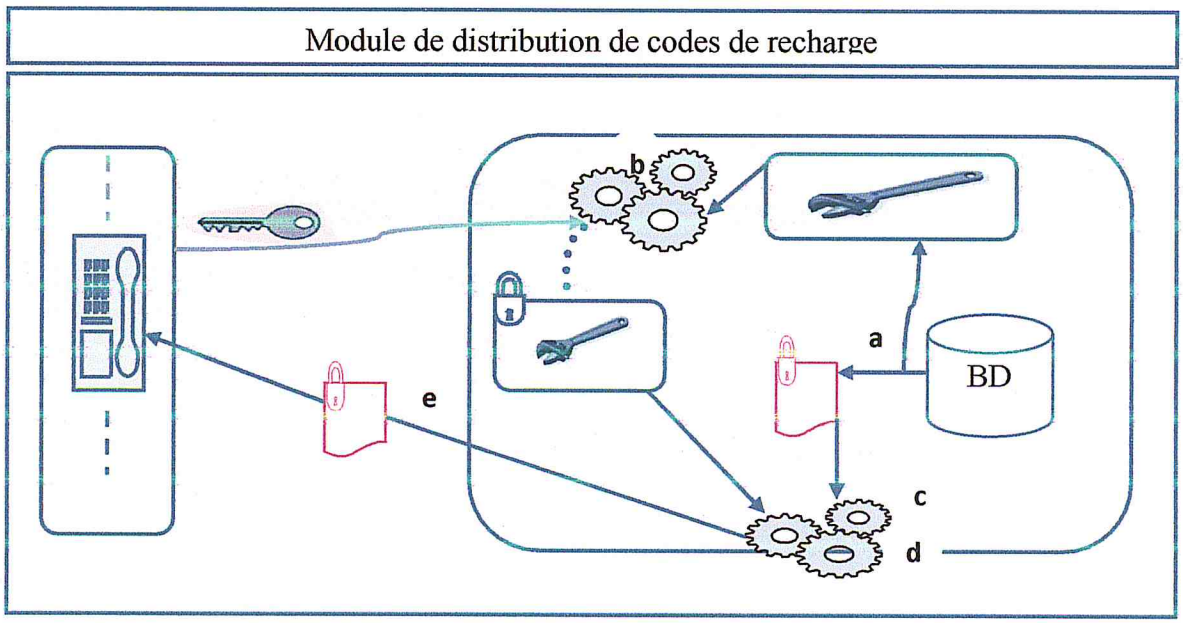



*Figure 4.5Distribution de codes de recharge*

**4.3.4.3.Application Front-Office** : récupère les codes de recharges cryptés et pour trouver l'état initial de codes de recharges utilise l'Algorithme Front Office (algorithme 3) pour réaliser les opérations inverses de l'application back-office:

a. *Accéder au fichier de codes de recharges cryptés et récupérer la clé secrète crypté ;*
b. *Décrypter la clé secrète en utilisant l'algorithme RSA et la clé privé du terminal ;*
c. *Utiliser la clé secrète décryptée pour décrypter les codes de recharges avec l'algorithme AES ;*

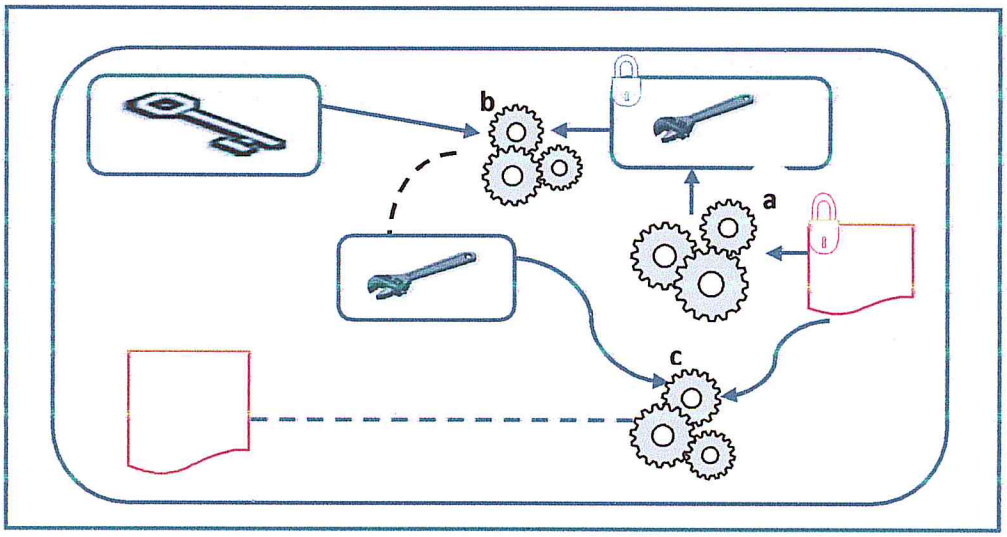



*Figure 4.6Décryptage de codes de recharge*

**Algorithme 2:**Algorithme Back Office

**Nom algorithme_Back_Office**
**Sortie :**fichierCodesCypté
**Début**
/* récupération, cryptage et stockage de codes de recharge dans la base de données */
**Pour chaque** operateur **Faire**
fichierCodes = recupererFichierCodeRecharge() ;
**Si** (procedure_Verifier_Integrite_Codes (fichierCodes)=vrai) **Alors**
cléSecrete = ProcedureGenererCléSecrete () ;
fichierCodesCypté= cryptageAES(cleSecrete, fichierCodes) ;
stockerCodesCryptesBaseDeDonnées (fichierCodesCypté, cléSècrete)
**FSI**
**Fait**

/* Distribution de codes de recharges à l'ensemble TPE selon à montant donné */
**procédure_ Distribuer_CodesRecharges ()** (procédure 2) ;
**Fin**

**Algorithme 3:** algorithme Front Office

```
Nom algorithme_Front_Office
Sortie : fichierCodesDecriptés
Début
fichierCodesCryptés= recupererFichierCodes() ;
cleSecreteCrypte=recupererCleSecreteCrypte (fichierCodesCryptés) ;
cleSecrete= decryptageRSA(clePrivée, cleSecreteCrypte) ;
fichierCodesDecriptés= decryptageAES(fichierCodesCryptés, cleSecrete) ;
Fin
```

**Procédure 1 :** Procédure Vérifier Intégrité de Codes

```
Nom procedure_Verifier_Integrité_Codes
Entrée :fichierCodesrecharges
Sortie : état
Début
état = vrai ;
Tant que (Fin (fichier)=faux) Faire
 Codes= lireCodes (fichierCodesrecharges) ;
Si (Digits (Codes, 1, POSITION_ESPACE_1)=faux)
Alors état=faux ; break ; Fsi
 Si (Digits (Codes, 1+ POSITION_ESPACE_1, POSITION_ESPACE_2)=faux)
Alors état=faux ; break ; Fsi
Si (Digits (Codes, 1+ POSITION_ESPACE_2, POSITION_ESPACE_3)=faux)
Alors état=faux ; break ; Fsi
Si (Digits (Codes, 1+ POSITION_ESPACE_3, POSITION_FINALE)=faux)
Alors état=faux ; break ; Fsi
Fin tant que
Fin
```

**Procédure 2 :** Procédure Distribuer Codes de Recharges

```
Nom  procédure_Distribuer_CodesRecharges
Entrée :montantDonné
Sortie :fichierCodesTerminalCrypté
Début
fichierCodesRechargesADistribuer=recupererCodeBaseDeDonnes(montantDonné
) ;
cléSecrète=récupérerCléSecrète (fichierCodesRechargesADistribuer) ;
nbreTerminaux= recupererNbreTerminauxBaseDeDonnes ();
nbreCodes= compterNbreCodes(fichierCodesRechargesADistribuer) ;
nbreCodesParTerminal=calculerNbreCodes(nbreTerminaux) ;
Pour chaque terminal Faire
cléPublique=recupererClePublique(terminal) ;
cléSecrèteCrypté = cryptageRSA (cléSècrete, cléPublique) ;
fichierCodesTerminalCrypté=creerFichier() ;
enregisterCleSecreteCrypteFichier (cléSecrèteCrypté,
fichierCodesTerminalCrypté) ;
enregisterCodes (fichierCodesTerminalCrypté, nbreCodesParTerminal,
fichierCodesRechargesADistribuer) ;
Fait
Fin
```

**Procédure 3 :** Procédure Générer Clé Secrète

```
Nom procédure_Generer_Clef_Secrete
Sortie :Clef
Entrée :TAILLE_CLEF
Début
Tant que (TAILLE_CLEF) Faire
        Val=PiocherValeur () ;
caractere=genererCaractere(Val);
        Clef=concatener(Clef, caractere);
Fin Tant que
```

### 4.4. Evaluation des performances de la solution proposée

Issus de la combinaison des deux approches, cette solution ne posera pas de problème lié à la transmission de la clef sécrète des méthodes symétriques. L'approche hybride héritera évidement des points fort de ses parents mais ne souffrira que du défaut des approches de cryptages asymétriques.

Nous sommes permis d'évaluer les performances de notre approche hybride sur les bases suivantes:

- La rapidité de cryptage asymétrique et transmission de la clef
- La rapidité de cryptage symétrique et transmission des données
- La résistivité par rapport aux attaques connues des deux approches symétriques et asymétriques.

Nous justifions pas la suite les points soulevés précédemment.

#### 4.4.1. Point de vue rapidité de cryptage et transmission de la clef

Comme la clef sécrète $K_s$ sera chiffré de façon asymétrique cela permettra surement d'éviter des échanges non sécurisés.

#### 4.4.2. La rapidité de cryptage symétrique et transmission des données

Puisque les algorithmes de cryptage symétriques sont les plus rapides ; Notre approche aura sans doute acquis de leur rapidité en matière de cryptage.

#### 4.4.3. La résistivité par rapport aux attaques connues des deux approches

Vu que les deux algorithmes sont utilisés de façon séparer cela ne nuit en rien la robustesse de l'un ou de l'autre. Mais en revanche cela rend l'approche hybride résistante aux attaques connues des deux algorithmes.

### 4.5 Conclusion

A la lumière de ce chapitre nous avons pu mettre en œuvre une solution hybride composé des deux algorithmes de cryptage ce qui rend encore plus fort notre application sur le plan sécurité.

Nous avons détaillé dans cette partie du document notre solution et aussi les algorithmes que nous avons choisis notamment AES et RSA qui sont deux grands dans le domaine de cryptographie puisque jusqu'à l'heure incassable.

Nous avons introduit dans notre solution un algorithme de générations dynamique et aléatoire de la clef sécrète ce qui rend encore quasiment indétectable la clef sécrète et assure donc une sécurité majoritairement parfaite.

Pour passer à la suite de l'élaboration du produit logiciel, nous fixons comme objectif du prochain titre à venir la mise en œuvre et le test de l'application de gestion sécuriséedu système dans son ensemble.

# Chapitre V

# Implémentation

# Et Test

## 5.1. Introduction

Après une modélisation de notre système nous allons dans ce chapitre effectué la mise ne œuvre de l'application. Nous procéderons en premier lieu à une présentation des *outils techniques utilisés*, du choix du *modèle d'agencement du code*, et pour finir nous exposerons quelques *interfaces* de notre application logicielle.

## 5.2 Langage et Outils de programmation utilisés

Suite aux contraintes qu'impose la réalisation et aussi compte tenu des choix des environnements de développement, nous décrivons les langages de programmations utilisées ainsi que les environnements utilisées.

### 5.2.1. Langages de programmation utilisés :

Nous utilisons notamment le *langage SQL* dans le but de dialoguer avec la base de donnée relationnelle sous MySQL.

En vue de pouvoir élaborer une application qui pourra fonctionner sur de multiple plateforme, nous utilisons le *langage JAVA* intégré à l'environnement éclipse. Ce langage de programmation supporté par plusieurs plateformes présente des avantages incontestés.

### 5.3.2 Outils de programmation utilisées

L'objectif est de réaliser une application. Pour ce faire nous aurons besoins essentiellement :

- Un Environnement de développement : dans notre cas ***Java Eclipse***
- Système de gestion de base de données : dans notre cas ***MySQL***.
- D'un outil de réalisation de rapport : dans notre cas ***Jasper Report***

#### 5.3.2.1 Java Eclipse [15]


Java Eclipse est un Environnement intégré de développement qui permet la création des logiciels. Il est extensible, universel et polyvalent, permettant de mettre en œuvre n'importe quel langage de programmation. Eclipse est principalement écrit en java à l'aide de la bibliothèque graphique SWT d'IBM.

#### 5.3.2.2 MySQL[16]

C'est l'un des SGBD les plus utilisé, créée 1994 par David Axmark et Michael Widenius de la société MysqlLab par David Axmark. Sa popularité est dû en grande partie au fait que le logiciel est Open source.

MySQL est un SGBDR (Système de gestion de base de données Relationnel) basée sur le modèle **Client-Serveur**, il utilise le langage SQL pour entretenir le dialogue avec le client.

#### 5.3.2.3 Jasper Report [17]

Jasper Reports est une librairie Java open source dédiée à l'ajout de capacités de reporting aux applications Java, Web ou autre. Elle permet la représentation des données sous format textuelle mais aussi la génération de graphiques divers. Comme Jasper Report est une librairie java, elle permet au programmeur de fournir les données au rapport sous forme de paramètres, de requêtes permettant de récupérées dans une de base de données, par l'intermédiaire d'une connexion JDBC fournie au Rapport.

#### 5.3.2.4 Modèle de Structuration du code : MVC [13]

MVC : désigné pour dire respectivement Modèle, Vue et Contrôleur.

- **Modèle** Définit les données qui sont manipulées par utilisateur ainsi que les méthodes d'accès.
- **Vue** Permet la représentation des données dans les interfaces avec lesquels l'utilisateur agit.
- **Contrôleur** Prend en compte les évènements pour la mise à jour de la vue ou du modèle. Cela joue en une façon une passerelle entre la vue et le modèle.

Nous y associons un schéma explicatif (figure ci-dessous).

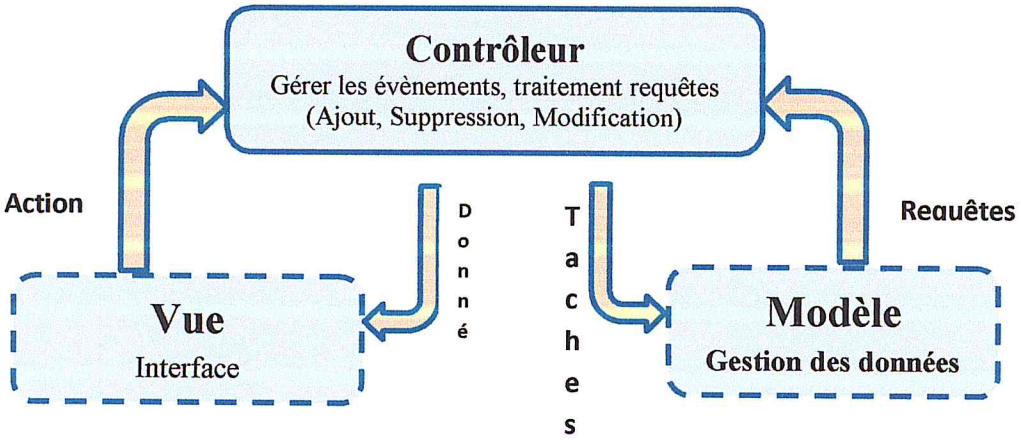



Figure 5.1 Le modèle MVC

L'utilisation du modèle de structuration MVC permet une bonne organisation de celui-ci, une mise à jour facile mais aussi une bonne compréhension même pour un autre intervenant sur le code.

## 5.4 Application

Cette partie nous permet de voir les résultats de notre réalisation c'est-à-dire de l'application. Cette dernière gérera et prendra en compte le cryptage des codes de recharges en vue d'assurer leur sécurité. Nous allons procéder à la présentation de l'application coté back-Office et aussi coté Front-Office.

L'application du Front-Office est à but de simulation du terminal. Contrairement à l'application du Back-Office.

### 5.3.1 Présentation de l'organigramme de l'application Back-Office :

Dans le but d'avoir une vue macroscopique des fonctions de l'application élaborés nous introduisons cet organigramme. Elle est une image de l'application de façon représentative.

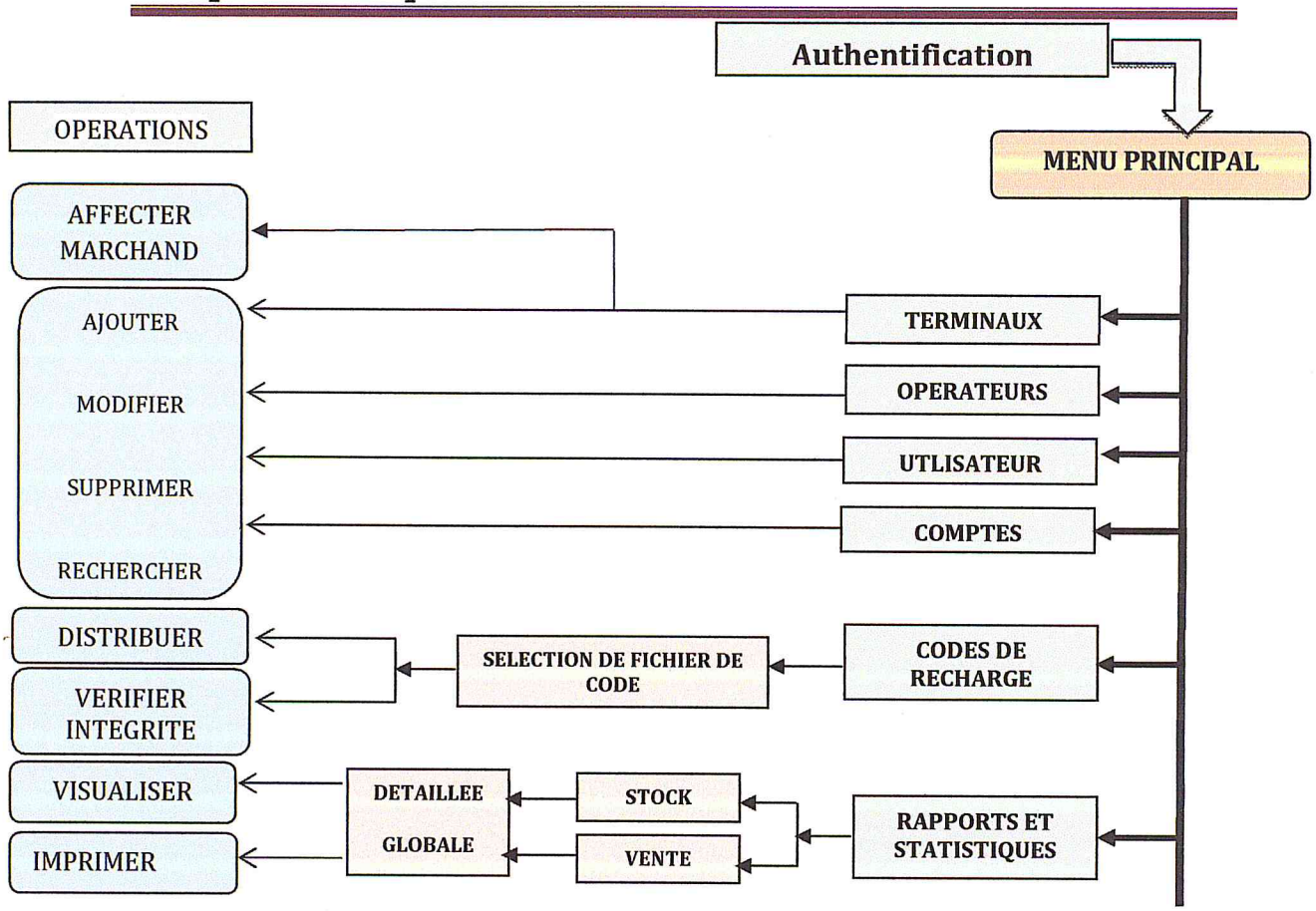



*Figure 5.2Organigramme de l'application*

### 5.3.2 Présentation des interfaces

Nous présentons dans cette section quelques interfaces principales de notre réalisation qui illustrent les différents cas d'utilisation déjà élaborés dans le chapitre précédent celui de modélisation.

#### 5.3.2.1 Interface d'authentification

Au démarrage de l'application nous aurons une interface d'authentification comme la montre la figure suivante :

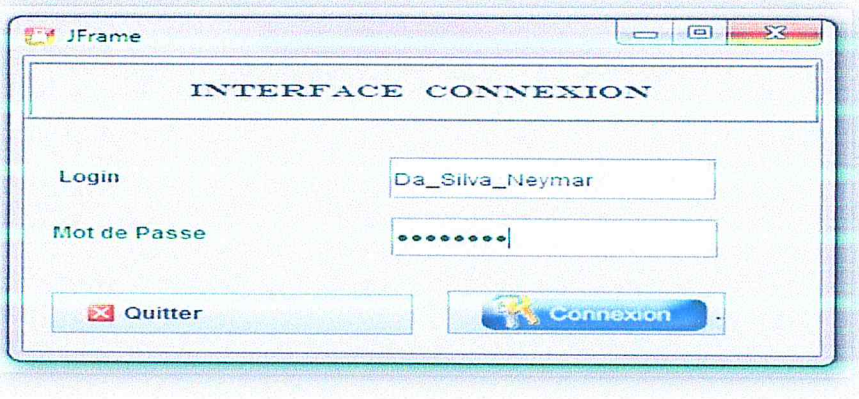



*Figure 5.3Interface d'authentification*

Une fois authentifié correctement l'utilisateur aura accès à l'interface principale de l'application

### 5.3.2.2 Interface principale

Cette interface montre les grandes fonctionnalités de l'application accessible à l'utilisateur connecté.

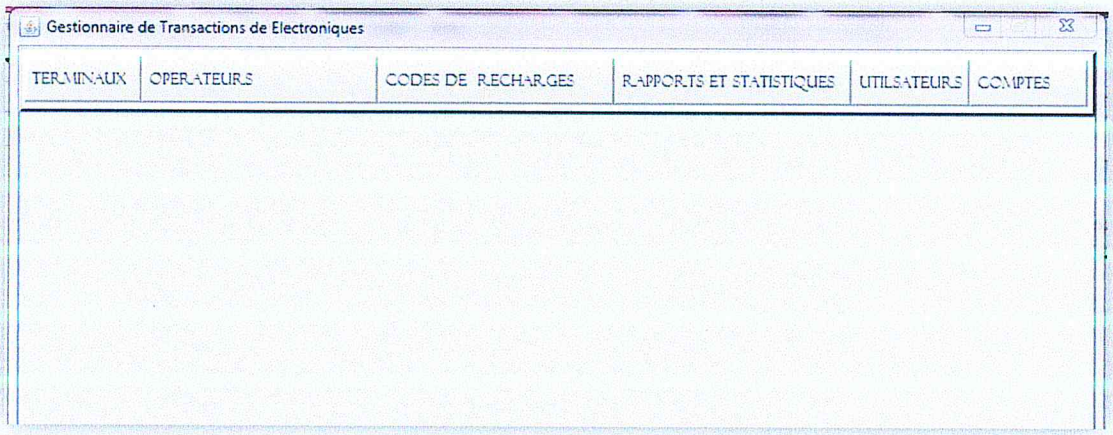



*Figure 5.4Interface d'authentification*

Selon qu'il soit de type Administrateur ou utilisateur simple, l'usager connecté pourra effectuer certaines taches selon les privilèges accordés.

**Simulation et Test de l'approche de sécurité proposée dans le système Evoucher**

Pour entamer la simulation nous aurons besoin qu'un opérateur nous envoie un fichier de code de recharge. Nous testerons sur les bases suivantes :

- 21 codes de recharges envoyés par l'opérateur au Back-Office.
- La plateforme contient 04 Terminaux

Avec 04 terminaux et 21 codes de recharges la plateforme par une politique de distribution des codes procèdera comme suit :

| Terminal | Nombre de codes |
|---|---|
| 1 | 5 |
| 2 | 5 |
| 3 | 5 |
| 4 | 6 |

*Tableau 1 Affectation de code de recharges au TPE*

### 5.4.2.3 Fichier de codes envoyé à la plateforme Back-Office

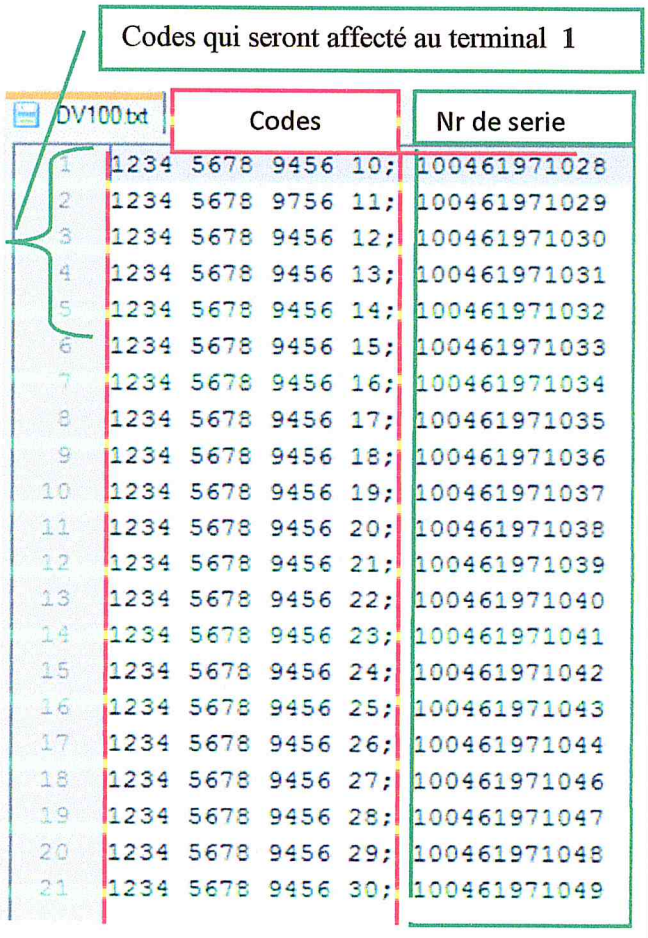



*Figure 5.5 Fichier de codes envoyé à la plateforme back office*

### 5.4.2.4 Sélection du fichier envoyé par l'opérateur

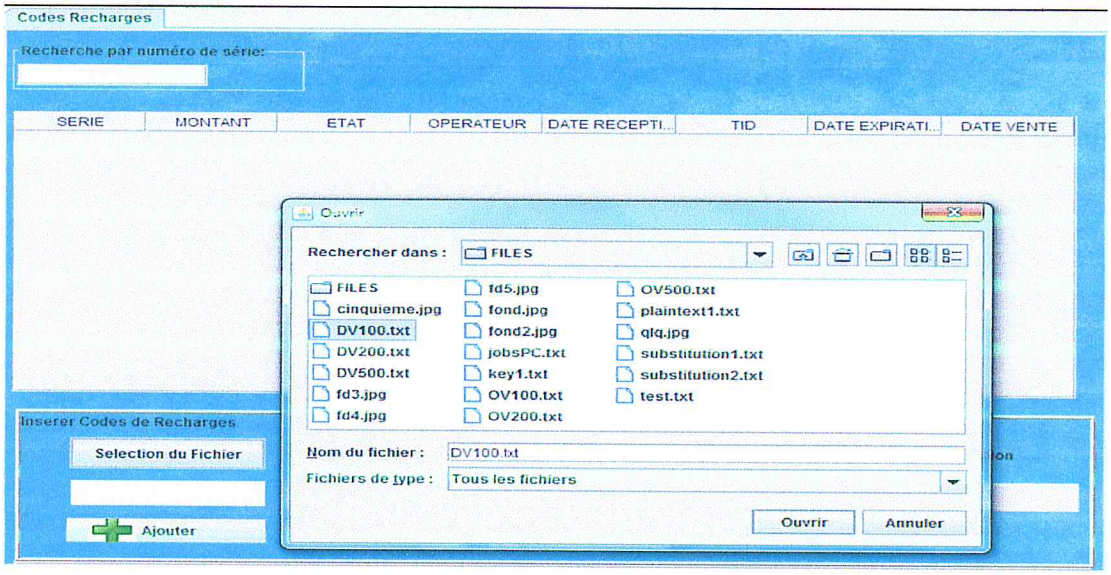



*Figure 5.6 Sélection du fichier de codes de recharges*

### 5.4.2.5 Insertion de codes de recharges dans la base de donnée

Cette insertion implique :

- Une vérification de l'intégrité de codes ;
- Une génération aléatoire d'une clé sécrète et cryptage de ces codes ;

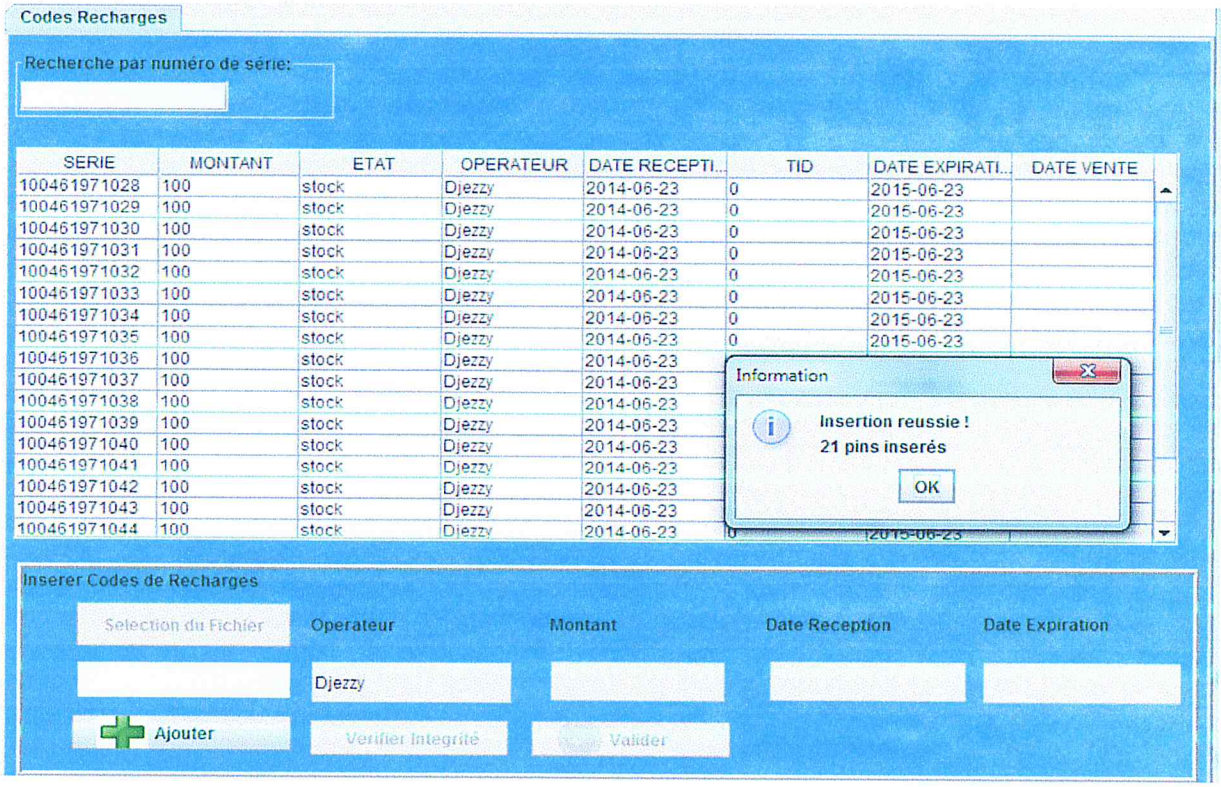



*Figure5.7 Insertion de codes de recharges*

### 5.4.2.6 Distribution de codes de recharges à l'ensemble de TPE

La distribution de codes de recharges implique :

- Pour chaque opérateur la création d'un fichier.
- Enregistrement de codes et de la respective clé secrète préalablement cryptée ;

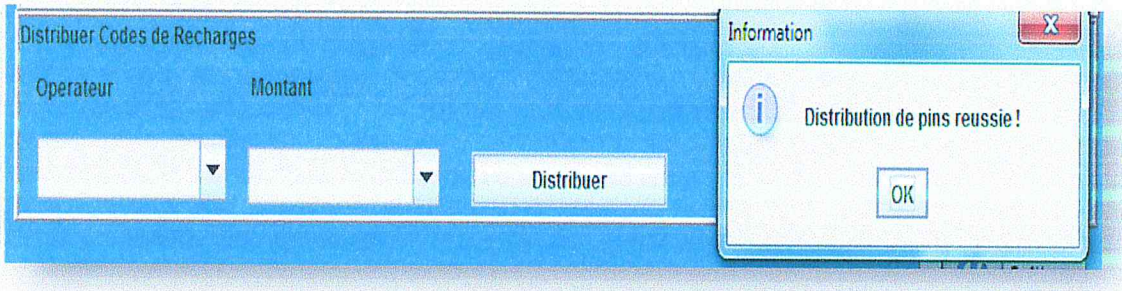



*Figure 5.8 Distribution de codes de recharges*

La distribution fut un succès alors les terminaux pourront dès à présent télécharger le fichier de code chiffré.

### 5.4.2.7 Vue du fichier de codes cryptés du terminal 1

Les codes de recharges sont cryptés selon AES, Mais la clef secrète est cryptée selon RSA

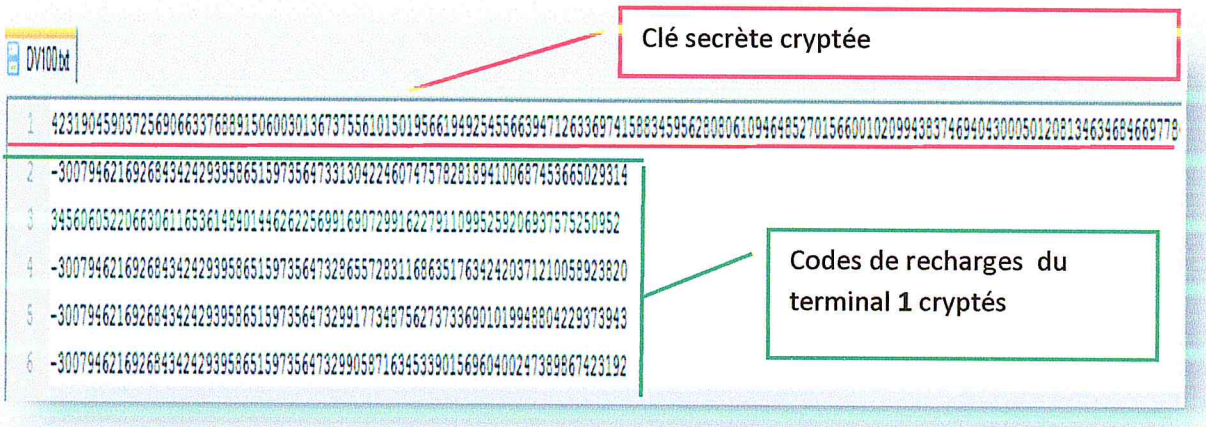



*Figure 5.9 Fichier codes cryptés du terminal 1*

### 5.4.2.8 Front-Office

Le Terminal télécharge à distance le fichier dans un répertoire dédié à lui seul, le chemin est unique. Une fois le fichier téléchargé alors il procède au décryptage du fichier chiffré.

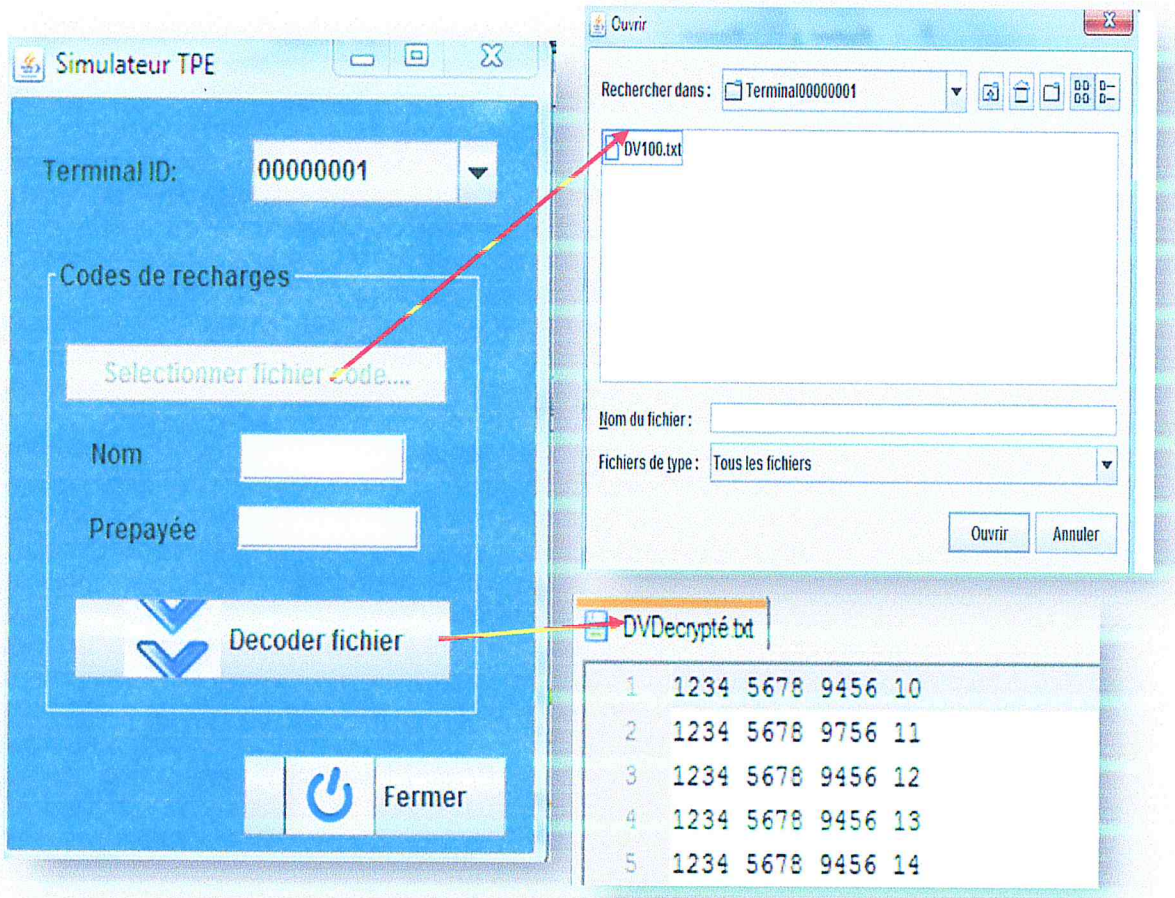



*Figure 5.10 Simulateur Front-Office*

Fichier bien reçu et bien décrypté au niveau du Front-Office.

Nous présentons quelques autres interfaces de l'application par la suite.

### 5.3.2.9 Interface Utilisateurs

Dans cette interface la gestion des utilisateurs est faite tout en tenant compte du type d'utilisateur lors de la création.

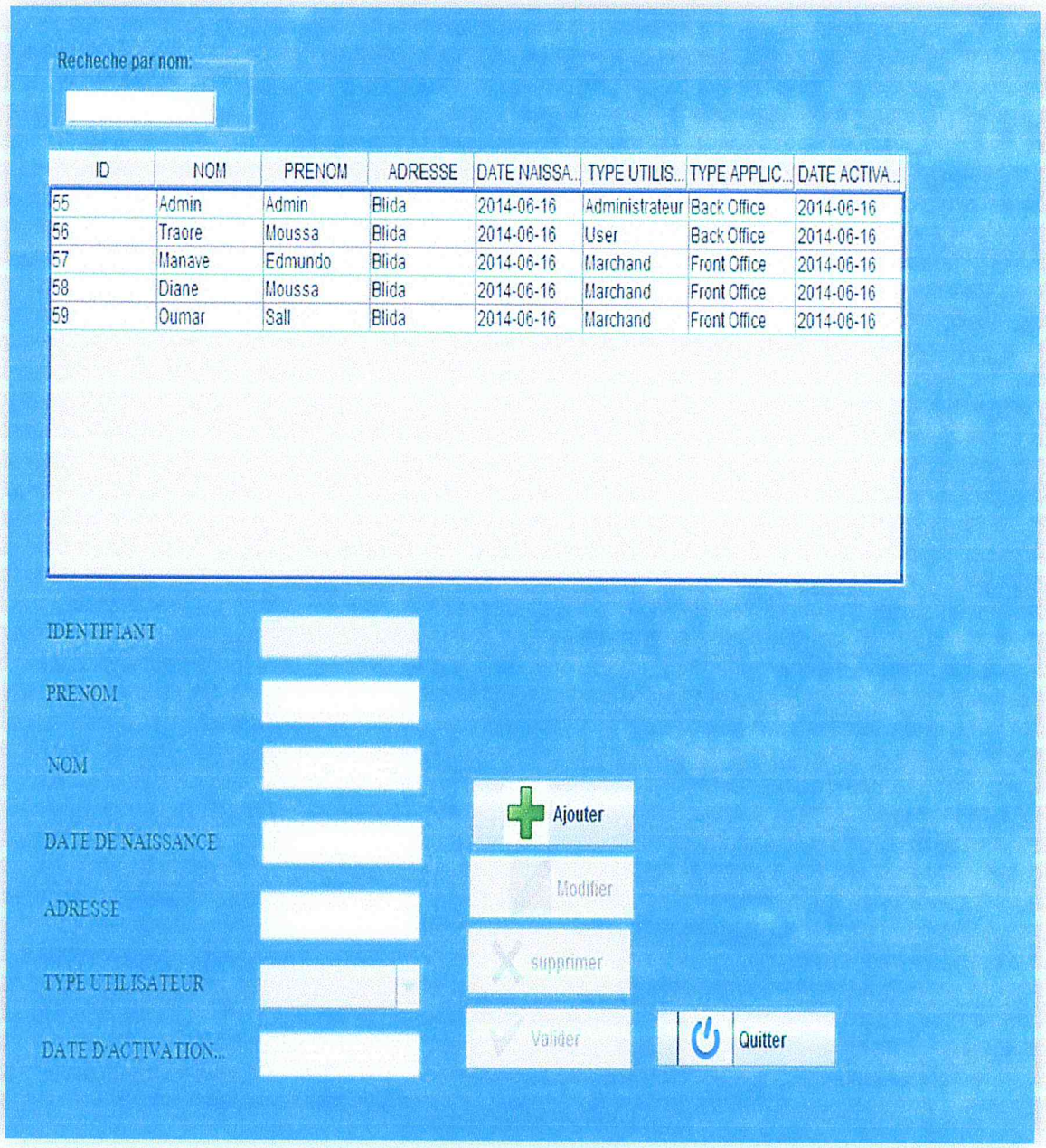



*Figure 5.11Interfaceutilisateur*

### 5.3.2.10 Interface Rapport et Statistiques

L'utilisateur pourra préciser les conditions d'édition de rapports de type détaillés ou global de stock ou de vente des codes de recharges d'un terminal.

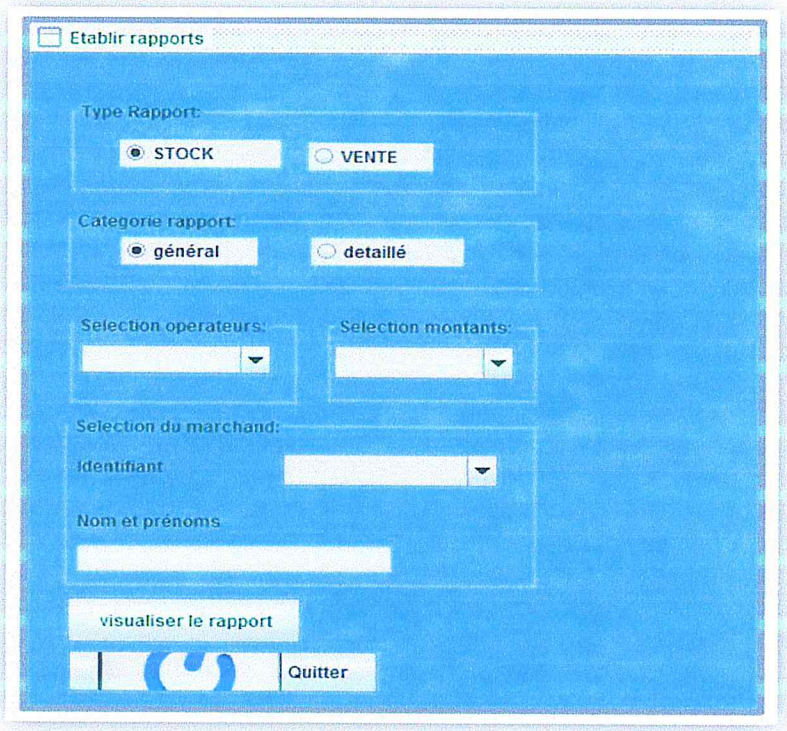



*Figure 5.12Interface Rapport et Statistiques*

Une fois les choix du type et de la catégorie du rapport effectué, le rapport pourra être visualisé.

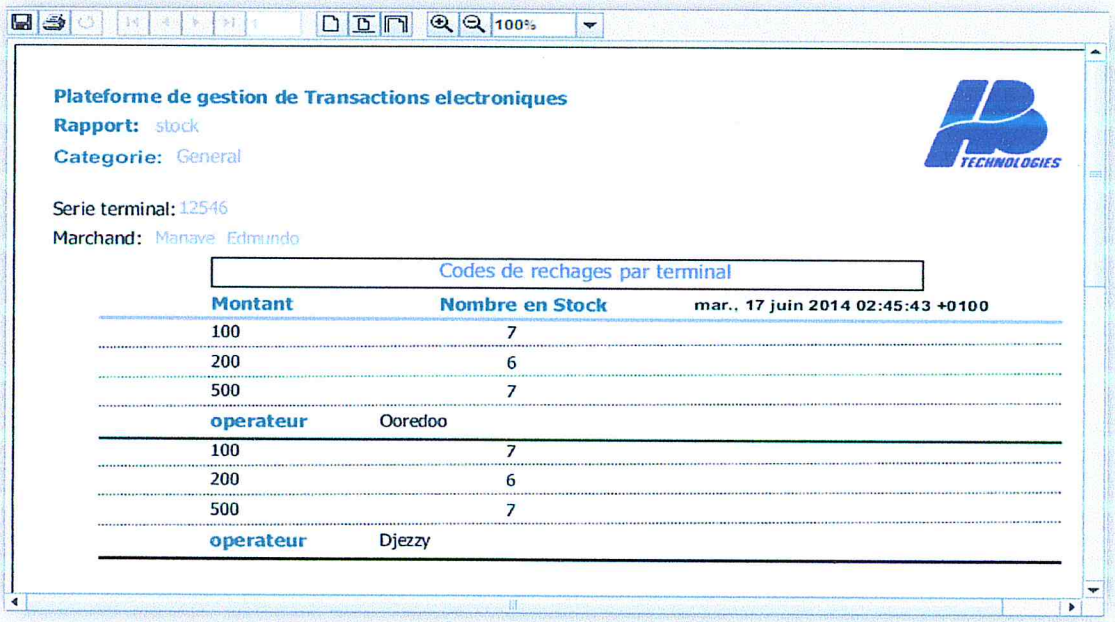



*Figure 5.13Un Rapport général de stock du Terminal 12546*

Nous avons aussi l'opportunité d'éditer éventuellement un rapport global

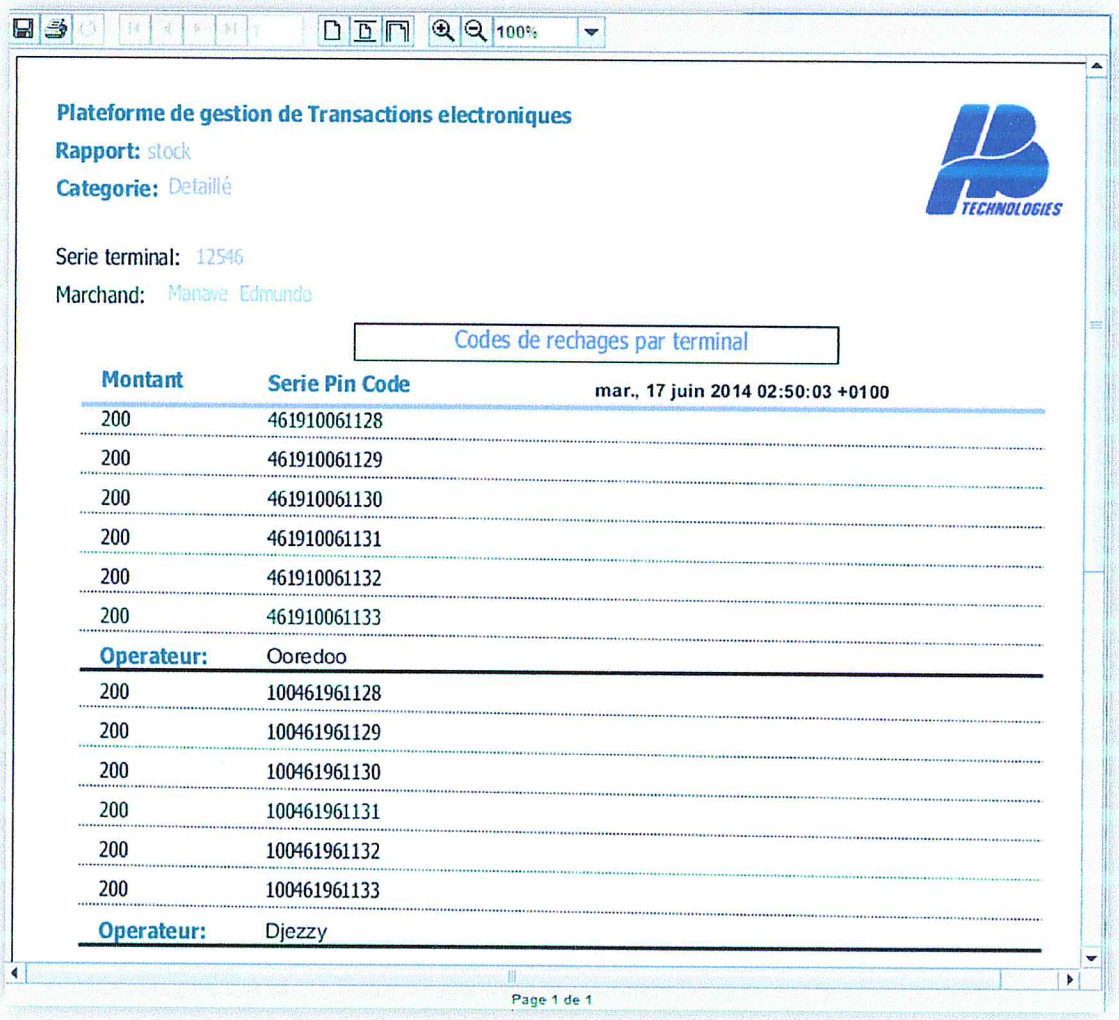



*Figure 5.14Rapport détaillée de stock du Terminal 12456*

### 5.3.2.11 Interface Terminaux

L'utilisateur après un clic sur le bouton Terminaux sera en face de l'interface ci-dessous. Elle permettra essentiellement d'effectuer des affectations et aussi d'autres opérations.

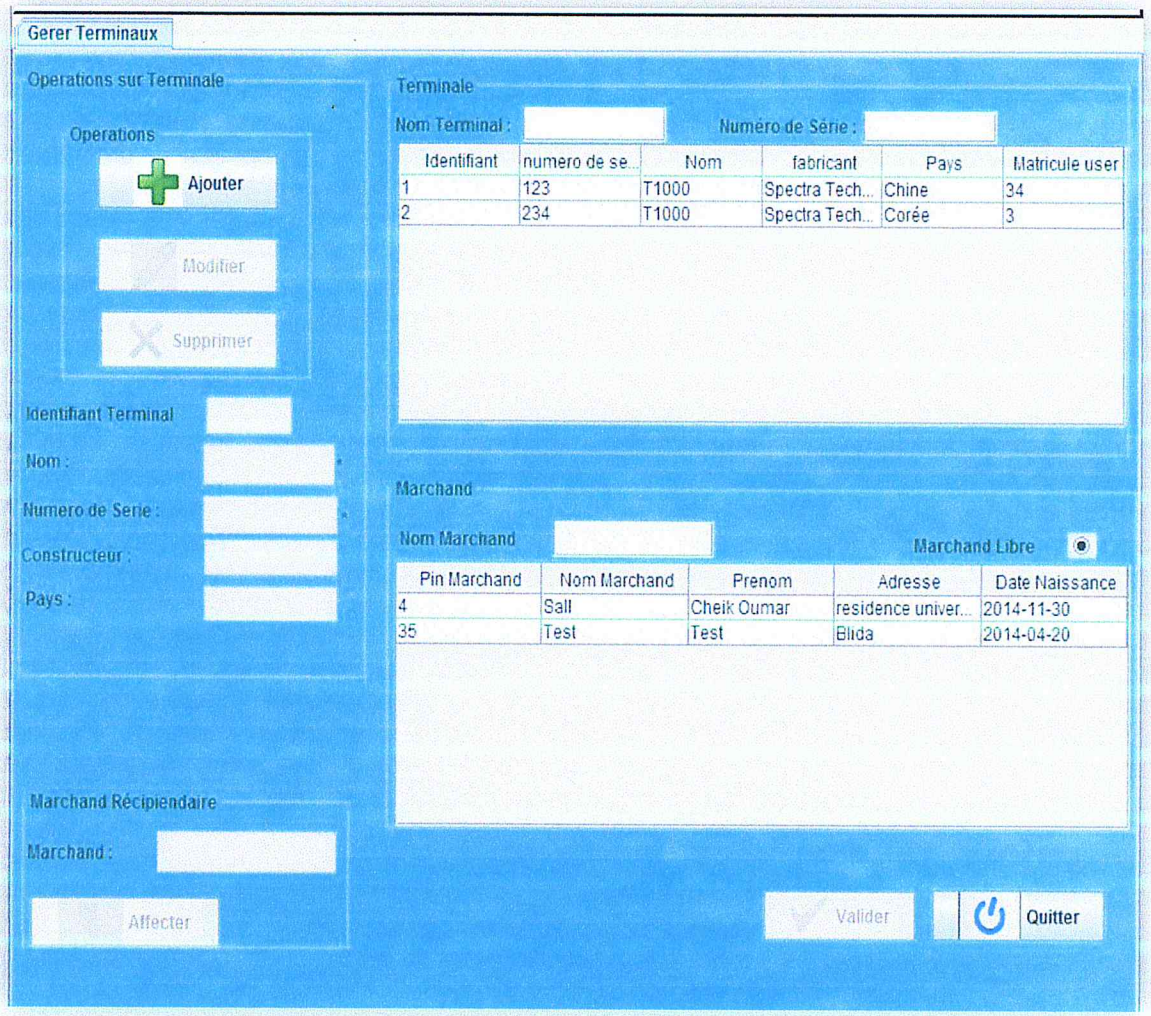



*Figure 5.15*Interface Gestion Terminaux

## 5.4. Conclusion

Après une brève introduction des choix de l'environnement et des outils de programmations utilisés, nous présentâmes quelques interfaces de notre application.

Nous avons fait un jeu de test pour montrer la simulation du fonctionnement de notre système (aussi bien du côté Back-Office que Front-Office) cela par rapport à notre problématique de sécurité, suivie de quelque autres interfaces de notre application.

# Conclusion générale

Dans ce travail nous avions comme objectif principal le développement d'une plateforme de gestion sécurisée du système Evoucher qui comportait la réception et transmission de codes de recharges aux terminaux, l'administration des entités (terminaux, operateurs, utilisateurs, ...), traçabilité et suivi de transactions électroniques par des rapports, entre autres. Le défi était alors de proposer une solution de sécurité pour la transmission de codes de recharge aux TPE tout en essayant de respecter les contraintes liées au système.

Pour la sécurisation des données transmises nous avons proposé une approche de sécurité hybride combinant deux méthodes cryptographiques, une asymétrique et autre symétrique tout en exploitant les points forts de chacune et aussi en respectant les aspects propres au système(vente en temps réel de codes de recharges).

Nous mettons en œuvre l'approche hybride dans le système Evoucher en faisant interagir le back office et le front office. Le back office crypte les codes de recharges avec l'algorithme symétrique AES dont la clé secrète est générée de façon aléatoire de manière à rendre encore plus improbable une éventuelle cryptanalyse de cette dernière, ensuite pour chaque TPE nous créons un fichier et enregistrons ces codes(en respectant un principe de distribution équitable) avec leurs clés secrètes respectives préalablement cryptées avec un algorithme de cryptage asymétrique RSA en utilisant la clé publique du TPE. Le front office fait un double décryptage pour trouver l'état normal des codes, il décrypte la secrète et ensuite l'utilise pour décrypter les codes de recharges en utilisant respectivement dans cet ordre les algorithmes RSA et AES.

Cette hybridation présente beaucoup d'avantages incontestés tant sur le plan rapidité de cryptage de données que sur la sécurité de transmission de la clef sécrète.

Lors de l'implémentation de cette approche nous nous sommes confrontés à de mainte problèmes. Celui d'encodage des codes de recharges cryptés parmi tant d'autre soit sur la base de données ou sur des fichiers. On a dû trouver des stratégies pour résoudre ce problème qui corrompait d'une manière les codes et amenait des incohérences.

**Perspectives et Recommandations**

Dans ce travail nous avons supposé que la transmission de codes de recharges de l'opérateur au back office du système Evoucher est sécurisée. En réalité cette transmission encourt les mêmes risques que la transmission de codes de recharges back office et TPE, d'où nous proposons une extension de l'approche de sécurité hybride pour cette transmission. Les opérateurs télécom devront alors crypter leurs codes de recharges selon une méthode symétrique et crypter les respectives clés secrètes avec une méthode asymétrique et puis envoyer ces données au back office, qu'à son tour réalisera les opérations inverses pour obtenir l'état normal de codes de recharges.

Le présent travail pourra être enrichie et mieux exploiter dans un environnement faisant beaucoup d'échanges de données comme l'internet où la communication bilatérale s'effectue à grand échelle.
Ce travail pourrait être amélioré par un approfondissement d'étude d'intégration dans de multiple environnement notamment accentué sur ceux ne disposant pas de beaucoup de sources d'énergie.

# Bibliographie

[1] **Aduno SA.** Aduno _ Pratique et commode - le terminal de carte Verdi Comfort.htm. *Aduno payment services.* [En ligne] www.aduno.com.

[2] **Chu, Nicolas et Thomas, Jean-Marie.***Réussir un projet de site web.* Paris : ÉDITIONS EYROLLES, 2006.

[3] **GAZZAH, WAHID.** *Rapport de projet de fin d'étude: Développement d'un lecteur de code à barre universel pour Android.* s.l. : Ecole Polytechnique Privée sousse Tunisie, 2012.

[4] *UML2 Analyse et conception*ParisDunod2008978-2-10-053567-5

[5] **Cochard, Gérard Michel, et al.** Paiement électronique et sécurisation des échanges. Paris, France : s.n., 27 Setembre 2006.

[6] **Jean-Guillaume Dumas, Jean-Louis Roch, Eric Tannier, Sébastien Varette.** *Théorie des codes Compression,cryptage,correction.* Paris : DUNOD, 2007.[7]*Chiffrement par blocs* Limoges .

[7]**Menezes, A, Van Oorschot, P et Vanstone, S.** Handbook of Applied Cryptography. s.l. : SRS Press, 1996.

[8]*Chiffrement par blocs* Limoges

[9]**Jonathan BLANC, Adrien De Georges.** *Techniques de cryptographie.* 2004.

[10] **Joan Daemen, Vincent Rijmen.** *The Rijndael Block Cipher : AES proposal.* Bruxelle, heverlee : s.n., 1999.

[11] **Barsky, Daniel.** *Cours de cryptographie.* 2006.

[12 ] **M. VIDEAU,** «CRITÈRES de SÉCURITÉ des ALGORITHMES de CHIFFREMENT à CLÉ SECRÈTE,» université de Paris 6, Paris, 2005

[13] **GAZZAH, WAHID.** *Rapport de projet de find d'étude: Développement d'un lecteur de code à barre universel pour Android.* s.l. : Ecole Polytechnique Privée sousse Tunisie, 2012.

[14] *UML2 Analyse et conception*ParisDunod2008978-2-10-053567-5

[15] *Introduction à Eclipse*IUT Orsay département informatique2008

[16] **gribaumont(Taguan), Chantale.** *Administrez vos Bases de Données avec MySQL.* s.l. : Le site du Zero, 2012.

[17] **Wilsh, JP.** *Guide d'utilisation JasperReport et iReport.* Californie : Communauté Adullact, 2008.