République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A. Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Fin d'études

En vue de l'obtention du diplôme de Master Profssionnel en Informatique

Option : Génie Logiciel

Thème

Développement d'une application mobile pour le transfert sécurisé des images médicales

Présenté par :

BEDOUHENE Rania & BACHIR Lynda

Devant le jury composé de :

Président :	Pr. H. SLIMANI	Professeur U.A/Mira Béjaia
Examinateur :	Dr. F. BOUCHEBAH	M.C.B U.A/Mira Béjaia
Promotrice :	Dr. L.HAMZA	M.C.A U.A/Mira Béjaia

Année Universitaire : 2021/2022

Remerciements

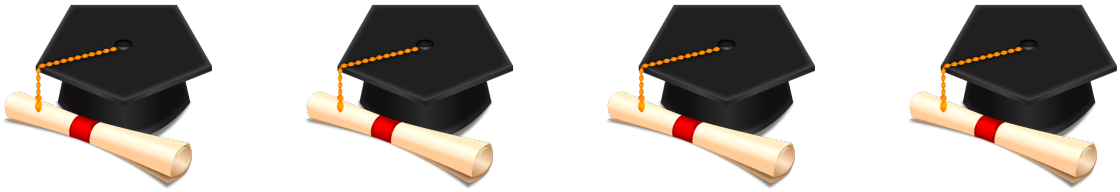
Tout d'abord, nous tenons à remercier ***Dieu****, de nous avoir donné la santé, la volonté et la patience à terme notre formation de master et pouvoir réaliser ce travail.*

Nous tenons à remercier notre promotrice Mme ***HAMZA Lamia*** *pour la confiance et l'intérêt quelle nous a témoigné durant toute la période de travail, pour son aide, ses précieux conseils, sa patience .*

Nous sommes très honorées de la participation de Professeur ***SLIMANI Hachem*** *, et Docteur* ***BOUCHEBAH Fateh*** *dans notre jury de soutenance. On les remercions vivement pour avoir accepté de juger ce travail.*

Nous tenons à remercier tous nos enseignantes qu'on a eu le plaisir de côtoyer pendant la période de notre formation à l'université de Béjaia.

Un grande merci à nos familles pour leurs soutiens aussi bien moral que financier et pour leurs sacrifices.



Dédicace

C'est avec une grande modestie et un immense plaisir que je dédie ce modeste travail :

À ma seule source de patience, de tendresse, et d'amour, ma chére et adorable mére ***Khedra****.*

À mon guide, ma source de courage, mon cher père ***Hamou*** *qui représente le symbole de la bonté pour moi.*

À mes chers soeurs et frères ***Fouad, Ahlam, Karim, Imane, Assia, Yousra, Oussama*** *et mes belles soeurs* ***Mounira, Chahinaz*** *pour leurs présence à mes côtés et leurs soutiens moraux.*

À mon fiancé ***Halim*** *en signe d'amour et de gratitude de m'avoir supporté, soutenu encouragé à poursuivre mes études, et aussi à sa famille.*

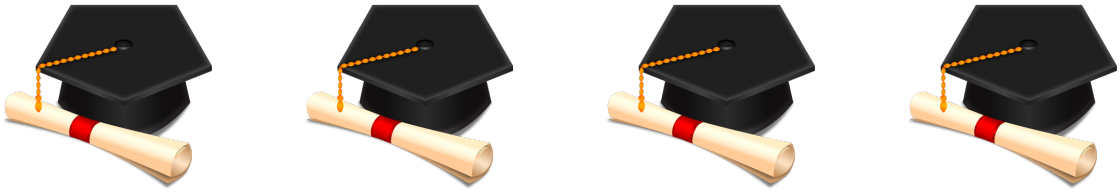
À mes chers neveux et niéces ***Mousaab, Ranim, Maissame, Anaïs, Oubaï, Asil, Ayla****.*

À mon cher binôme ***Lynda*** *pour tous les moments qu'on a passé ensemble.*

À mes amies ***Sonia, Romaissa, Khaoula, Imane, Chahira, Mouna, Fairouz, Nihad, Yousra, Nassima*** *pour leurs encouragements et fidélité.*

À tous ceux qui été à mes côtés durant la réalisation de ce travail.

Rania



Dédicace

dédie ce modeste travail à :

Ma famille : mon père ***MOHAND*** *(paix a son âme), à ma chère Mère* ***Houria****,* ***mes sœurs, mes frères*** *.*

A Mon cher binome ***Rania*** *avec qui j'ai partagé de belles années d'études.*

*A Mes amis :****Ryma, Lynda, Thiziri*** *a tous ceux que j'aime...*

Lynda

Table des matières

Liste des tableaux

Table des figures

List of Algorithms

Liste des abréviations

2TUP *2 Track Unified Process*

AI *Adobe Illustration*

ASCII *American Standard Code For Information Interchange*

AES *Advanced Encryption Standard*

DES *Data Encryption Standard*

DPI *Dots Per luch*

DICOM *Digital Imaging and Communications in Médicine*

EPS *Encapsulated Postscript*

GIF *Graphique Interchange Format*

IRM *Image Résonance Magnétique*

JPEJ *Joint Photographe Experts Group*

JDK *Java Development Kit*

JRE *Java Runtime Environment*

NIST *National Institute of Standards and Technology*

PIXEL *Picture Element*

PPP *Point Par Pouce*

PDF *Portable Document Format*

PNG *Portable Network Graphics*

PSD *Document Photoshop*

RX *Rayons X*

RVB *Rouge Vert Bleu*

RSA *Rivest, Shamir, Adleman*

RC4 *Rivest Cipher 4*

RUP *Rational Unified Process*

SDK *Software Development Kit*

SVG *Scalade Vector Graphics*

SMS *Short Message Service*

TIF *Tagged Image File Format*

UP *Unified Process*

UML *Unified Modeling Language*

XML *Extensible Markup Languag*

XOR *eXclusive OR*

Xp *Extreme Programming*

Introduction générale

Actuellement, le monde connaît une avancée technologique considérable dans tous les secteurs, et cela grâce à l'informatique qui est une science étudiant les techniques de traitement automatique de l'information. Celle-ci joue un rôle important dans le développement des entreprises et d'autres établissements.

Avec l'accélération significative du développement des technologies, la révolution numérique a permis de simplifier le traitement et la transmission de l'information multimédia, notamment la transmission des données médicales en général, et des images en particulier.
L'imagerie médicale est l'un des outils les plus importants. Elle joue un rôle clé dans la communication entre les patients et les médecins. Les images médicales constituent un moyen universel qui permet à chacun d'entrer en contact avec des professionnels de la santé et de recevoir un diagnostic à distance. Cependant, pour des raisons éthiques, le transfert d'images médicales présentent certains risques. Assurer leur protection est donc primordiale.

Dnans le but de contribuer à la protection de ces données et d'assurer une transmission sans risques, nous avons fait appel aux concepts de base des algorithmes de cryptage pouvant garantir la confidentialité, l'authentification et l'intégrité en cas de modification. Dans ce contexte, nous avons utilisé le codage de base 64, du fait de sa disponibilité dans la majorité des systèmes informatiques et son rôle dans la conversion d'image en texte. Nous avos également appliqué l'algorithme de cryptage AES (Advanced Encryption Standard) pour crypter et décrypter notre image médicale et assurer la protection de ses informations. Ce partage d'images se faire par internet, mais que faire s'il n'y a pas de connexion Internet et que l'image doit être partagée. Dans ce cas, la technique du SMS (Short Message Service) peut être utilisée pour envoyer l'image.

Ainsi, ce travail a pour objectif de contribuer à la sécurisation du contenu des images médicales, mais également du processus de transmission entre les utilisateurs. Pour cela, un intermédiaire entre les parties est nécessaire. Celui-ci peut être représenté par l'application de clés de sécurité entre les deux appareils pendant le processus.

Le présent travail s'articule autour de quatre chapitres :

- Le chapitre 1 intitulé "Imagerie médicale", où des informations générales sur les images, ainsi que différentes techniques et caractéristiques d'imagerie médicale sont présentées.
- Le chapitre 2 intitulé "Cryptographie", comprenant les concepts de base de la cryptographie, des techniques de cryptage des images médicales et une présentation du codage base64.
- Le chapitre 3 intitulé "Analyse et conception", illustre les besoins fonctionnels et non fonctionnels de notre application. Suivie d'une identification des acteurs et de leurs fonctionnalités, à travers le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme de classes.
- Le chapitre 4 intitulé "Implémentation", dans lequel les différents outils techniques de l'implémentation et les langages de programmation utilisés pour le développement sont présentés. Ce chapitre comprend également des exemples de quelques interfaces de notre application.
- Nous terminons ce mémoire par une conclusion générale et quelques perspectives qui peuvent aider à améliorer le système dans le futur.

Chapitre 1

Imagerie Médicale

1.1 Introduction

La photographie est l'une des techniques les plus populaires aujourd'hui, utilisée par presque tout le monde depuis des décennies pour présenter ou représenter un objet, une personne ou un événement. Dans ce chapitre, nous définirons les images en général, les caractéristiques des images, les types d'images numériques et leurs différents formats les plus importants, puis nous aborderons les images médicales, les principales techniques d'imagerie médicale, les caractéristiques de l'imagerie médicale, et enfin nous définirons par le format le plus utile pour les images médicales.

1.2 Image

Une image est une représentation visuelle voire mentale de quelque chose (objet, être vivant et/ou concept). Elle peut être naturelle (ombre, reflet) ou artificielle (peinture, photographie), visuelle ou non, tangible ou conceptuelle (métaphore) [1].

Une image numérique est une représentation numérique (valeur binaire de 0 et 1). Elle peut être définie comme une fonction $f\ (x,\ y)$ où x et y sont des coordonnées spatiales (plan) pour chaque pixel [2] , tel que x représente la largeur et y représente la hauteur, la Figure 1.1 présente un exemple d'une image numérique :

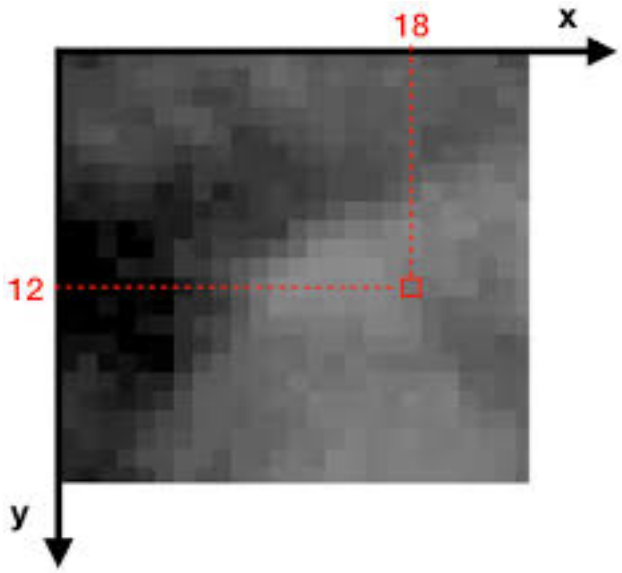


FIGURE 1.1 – Image numérique avec coordonnées [26]

1.3 Caractéristiques de l'image

Une image a des nombreuses caractéristiques, les plus importantes sont les suivantes :

1.3.1 Pixel

Un pixel (Picture Element) est le plus petit élément d'une image numérique, quand on examine l'image, il se présente comme un carré de couleur uniforme [3]. La Figure1.10 présente les pixels d'une image numérique noir et blanc ou chaque pixel et codé avec un seul bit (0 (noir) ou 1 (blanc)).

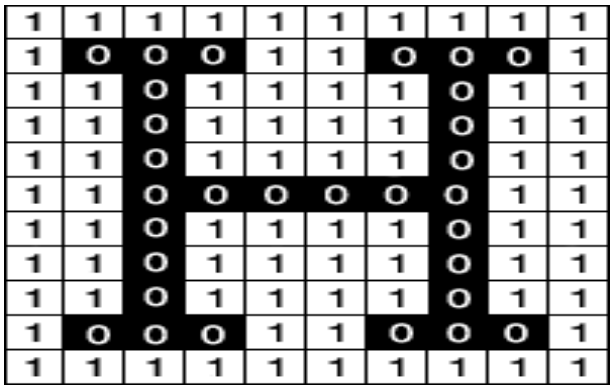


FIGURE 1.2 – Pixels d'une image numérique [4]

1.3.2 Définition d'une image

La définition d'une image est le nombre de points (pixels) qui comporte une image numérique en largeur et en hauteur (le nombre de colonnes et le nombre de lignes).

Dans la Figure 1.3 la définition de l'image est : 10 (colonnes) * 11(lignes)= 110 pixels [27].

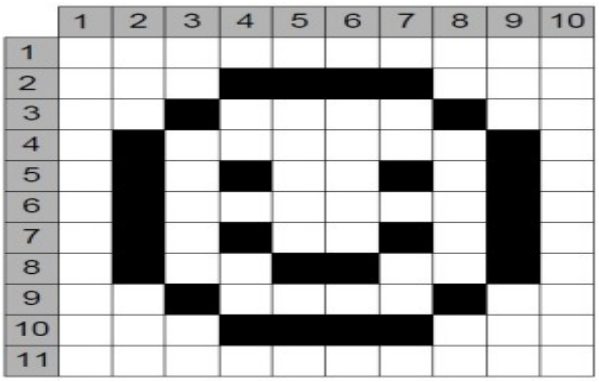


FIGURE 1.3 – Définition de l'image [27]

1.3.3 Résolution d'une image

La résolution d'une image est le nombre de pixels contenus dans l'image par unité de longueur elle s'exprime souvent en PPP (Point Par Pouce) ou en anglais DPI (Dots Per luch). Sachant qu'un **1 pouce (inch)=2,54 cm** et la **résolution=définition/dimension**. Plus la résolution est grande plus l'image est précise dans les détails.

1.4 Types d'images numériques

1.4.1 Image Matricielle (Bitmap)

Une image bitmap est composée en mode point (matrice de pixel). Plus on zoom, plus les pixels deviennent apparents [5]. On obtient également des images matricielles à l'aide d'un appareil photo numérique ou d'une caméra vidéo numérique.

1.4.2 Image vectorielle

Formée de lignes calculées de manière géométrique [4] permettant un redimensionnement de l'image sans aucune dégradation. Elle est utilisée surtout pour les dessins et les animations.

La Figure 1.4 illustre les images matricielles et vectorielles :

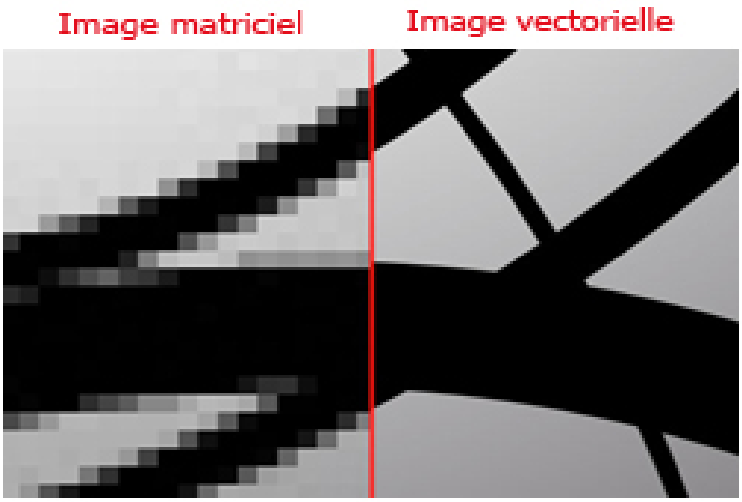


FIGURE 1.4 – Image matricielle VS image vectorielle [28]

1.5 Formats de l'image

Les formats d'images les plus utilisable sont les suivants [29] :

1.5.1 Formats vectoriels

1. AI

AI (Adobe Illustration) est un format développé par Adobe pour représenter un dessin vectoriel soit en EPS (Encapsulated Postscript) ou en PDF (Portable Document Format) et sont utilisés par les graphistes et les imprimeurs pour créer des images à utiliser sur tous les supports de communication. L'objectif de ce format est de pouvoir redimensionner l'image à volonté sans aucun effet d'escalier ou pixellisation.

2. EPS

Les fichiers de format (EPS) sont le plus couramment utilisés pour transférer une image ou une illustration, généralement d'un fichier vectorielle dans une autre application. Les fichiers vectoriels (EPS) s'adaptent à toutes les tailles.

3. SVG

Le SVGs (Scalade Vector Graphics) ou graphique vectoriel adaptable en français est un format de données conçu pour décrire des ensembles de graphiques vectoriels. Il est utilisé pour les éléments visuels d'une entreprise : logo, icônes, illustrations, etc.

4. PDF

PDF est un format de fichier universel qui conserve les polices, les images, la mise en page et les graphiques du document source. Les fichiers de format (PDF) peuvent être partagés, visualisés et imprimés par n'importe qui avec le logiciel gratuit Adobe Reader.Ce format est utilisée pour des formulaires en ligne et documents d'impression commerciale, numériques et /ou de bureau.

1.5.2 Formats matriciels

1. GIF

Les fichiers de format GIF (Graphique Interchange Format) sont des fichiers de basse résolution. Il sont couramment utilisés à des fins web et pour les e-mails. Ce format utilise un système de compression pour maintenir la taille du fichier et est utilisée pour les courtes animations sur les réseaux sociaux, comme Facebook et Twitter.

2. JPEJ

JPEJ (Joint Photographe Experts Group) est un format d'enregistrement et de compression numérique. Il est utilisé pour des images Web, des images de réseaux sociaux.

3. PNG

Les fichiers de format PNG (Portable Network Graphics) sont des images (bitmap) qui emploient la compression sans perte de données. Ce format est le plus utilisé pour une utilisation en ligne et sur les sites en raison de leur faible résolution.

4. PSD

Le format PSD (Document Photoshop) est un format bitmap qui contient des graphiques et des photos créés dans Adobe Photoshop, logiciel de retouche d'image.

5. TIF

Le TIF(Tagged Image File Format) est le format de fichier le plus utilisé pour stocker des images et des photographies pour l'impression commerciale.

Dans ces nombreux formats les avantages et inconvenants sont déterminants quant à la décision de leur utilisation. Le tableau 1.1 illustre les avantages et les inconvénients des formats d'images :

Formats	Avantages	Inconvenantes
AI	Redimensionner des images sans altérer la qualité	Il faut avoir Adobe Illustrator sur votre appareil
EPS	Peut s'ouvrir avec plusieurs logiciels	Fichier assez lourd
SVG	Peut être redimensionné sans perte de qualité	Est un format insupportable dans les réseaux
PDF	La mise en page quel que soit l'appareil sur lequel il est affiché est conservé les polices	Il faut installer PDF Reader sur votre appareil
GIF	Supporte le fond transparent et l'animation basique	Les dégradés de couleurs ne sont pas bien affichés
JPEG	Fichier léger, visible sur presque tous les programmes	Perte de qualité quand le fichier est sauvegardé plusieurs fois
PNG	Bonne qualité des images, supporte les fonds transparents	Fichier lourd (prend de l'espace de stockage et met du temps à charger)
PSD	Les images peuvent être travaillées très précisément	Il reste un fichier lourd
TIF	L'excellente qualité de l'image	Très lourd

TABLE 1.1 – Avantages et inconvenantes des formats d'images

1.6 Système des couleurs des images numériques

Les couleurs ont une grande importance dans l'image et le graphisme, elles ont plusieurs système dans les plus importants sont :

1.6.1 Couleurs noir et blanc

L'image de ce type est en noir et blanc, ou le pixels est de 1 bit. Ce mode est adapté pour numériser des textes, des logos, des schémas, des plans, des microfiche [3]. La Figure 1.5 présente un exemple d'une image en noir et blanc :

FIGURE 1.5 – Image en noir et blanc [6]

1.6.2 Niveaux de gris

L'image se compose d'une échelle de couleurs, allant du blanc au noir à différents degrés, et le pixel est représenté par 8 bits. La Figure 1.6 représente une image niveaux de gris :

FIGURE 1.6 – Image niveaux de gris [6]

1.6.3 Couleur(RVB)

L'image consiste à fusionner le rouge R, le vert V et le bleu B dans certains proportions, pour produire de nombreuses couleurs, qui peuvent atteindre jusqu'à 16777216 couleurs, et chaque pixel est représenté par 24 bits. La Figure 1.7 montre une image en couleurs (RVB (Rouge Vert Bleu)) :

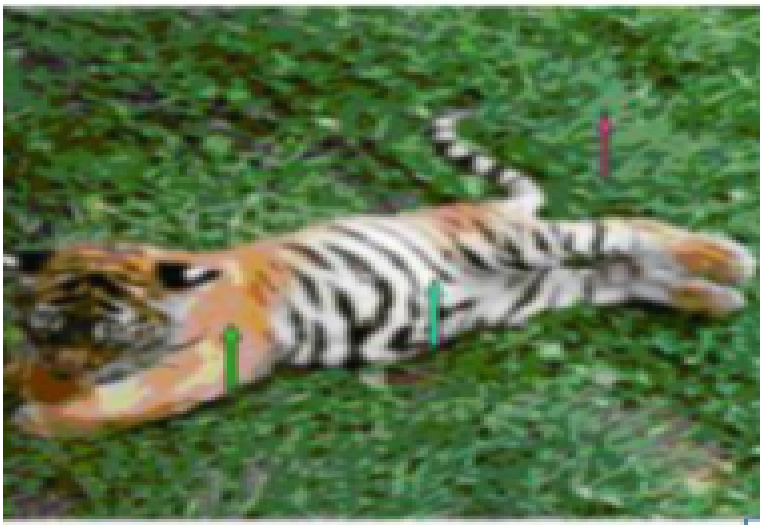


FIGURE 1.7 – Image en couleurs RVB [6]

1.7 Imagerie médicale

L'imagerie médicale est l'un des domaines de la médecine qui a le plus progressé ces vingt dernières années. Permettent non seulement le meilleur diagnostic mais offre aussi de nouveaux espoirs de traitement pour de nombreuses maladies. Cancer, épilepsie, etc. Ce domaine regroupe différentes techniques permettant d'observer l'intérieur du corps humain.
L'magerie médicale a pour objectifs :

- Aide à la décision et au diagnostique.
- Traitement d'un grand nombre de données.
- Geste médical et chirurgical assisté par ordinateur.
- Visualisation scientifique.

1.8 Principales techniques d'imagerie médicale

L'imagerie médicale se divise en deux groupes d'images, les images de la structure qui se concentrent sur l'aspect anatomique des organes du corps humain et les images de la fonction qui montrent le fonctionnement des organes. Dans cette section nous présentons les principales techniques d'imagerie médicale.

1.8.1 Champ magnétique

Les scanners d'imagerie par résonance magnétique IRM (Image Résonance Magnétique) utilisent des champs magnétiques d'environ 10 000 à 60 000 fois plus puissant que le champ magné-

tique terrestre [1]. C'est une technique de recherche des anomalies des vaisseaux, ainsi que des tumeurs, des lésions infectieuses ou inflammatoires. Cette technique d'imagerie repose généralement sur l'interaction des protons du corps humain avec un champ magnétique.

1.8.2 Radiographie

La radiographie a été la première technologie d'imagerie médicale, elle est réalisée avec une source de rayons X sur un côté du patient et une radiographie (généralement plate) détecteur de l'autre côté [7]. C'est une technique utilisée pour observer les os et le squelette, et notamment pour déceler les fractures en cas de traumatisme. Elle est également utilisée pour rechercher des tumeurs mammaires et pulmonaires. La Figure 1.8 présente un exemple d'une radiographie :

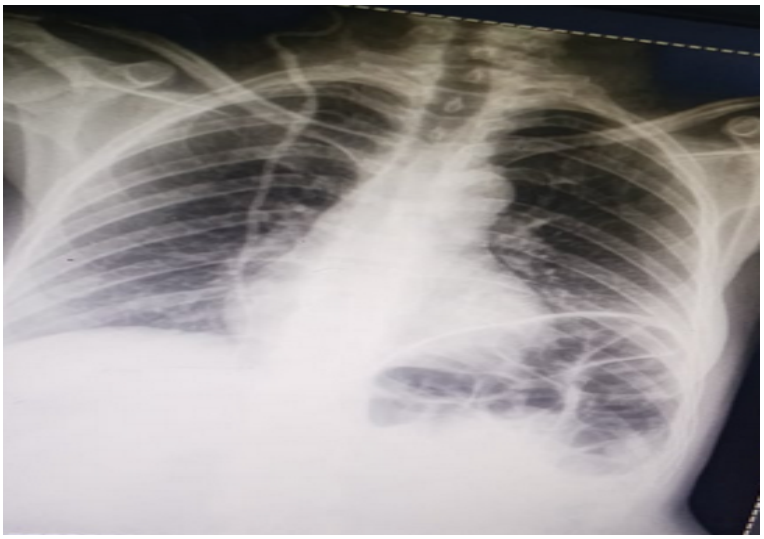


FIGURE 1.8 – Radiographie

1.8.3 Rayons X

Les rayons X sont des ondes électromagnétiques (de même nature que les ondes de lumière mais plus énergétiques). Ils ont la propriété d'être atténués par toutes sortes de substances, y compris les liquides et les gaz [8], les rayons X sont utilisés en radiographie et sont constitués de rayons gamma.

1.8.4 Ultra-sons

Les ultra-sons sont des ondes sonores imperceptibles à l'oreille humaine. Comme toutes les ondes sonores, les ultra-sons sont absorbés ou réfléchis par les substances qu'ils rencontrent [8]. Ils permettent une coupe de l'organe ou l'étude du flux sanguin dans les vaisseaux, même ceux du cœur. L'imagerie par ultra-sons utilise une technique "d'écho d'impulsion" pour synthétiser une

image tomographique en niveaux de gris des tissus basées sur l'interaction mécanique de courtes impulsions d'ondes sonores à haute fréquence et de leurs échos de retour [7].

1.8.5 Échographie

L'échographie est une modalité qui utilise l'énergie ultra-sonore et les propriétés acoustiques du corps pour produire une image à partir de tissus immobiles et en mouvement [7], qui permettent de visualiser l'intérieur du corps. C'est une technique très utilisée pour le diagnostic de nombreuses pathologies mais aussi pour le guidage visuel lors d'autres examens, comme des biopsies. La Figure 1.9 présente une échographie :

FIGURE 1.9 – Échographie

1.8.6 Rayons lumineux

Les rayons lumineux permettent de voir des coupes de tissu et la quantité d'oxygène contenue dans ceux-ci. Ils permettent d'étudier le travail cellulaire.

1.9 Caractéristiques des images médicales

Dans cette sections nous présentons les caractéristiques les plus importantes des images médicales.

1.9.1 Taille des images médicales

En imagerie médicale, la taille de l'image dépend souvent du capteur utilisé lors de l'acquisition et de la zone anatomique à imager.

1.9.2 Résolution spatial

La résolution spatiale est la capacité de distinguer la structure fine et les détails d'une image. C'est le nombre de paires de lignes distinctes dans l'image par unité de longueur.

1.9.3 Bruit

Phénomène aléatoire qui se surajoute à l'image idéale et qui pourra induire une détérioration de la résolution spatiale, même rendre les images complètement illisible.

1.9.4 Contraste dans les images médicales

Le contraste est la différence dans l'échelle de gris de l'image entre des régions étroitement adjacentes sur l'image [7].

1.10 Manipulations et les attaques sur les images médicales

Différents paramètres permettent de mesurer le niveau de cryptage des images médicales en calculant ou analyser certains paramètres. Tous ces paramètres aident à détecter le niveau de algorithme pour résister à une attaque particulière comme indiqué .

1.10.1 MSE (Mean Square Error)

MSE est l'estimateur d'échelle de mesure de qualité d'image le plus courant. C'est une échelle La référence complète et les valeurs les plus proches de zéro sont les meilleures. C'est le deuxième moment d'erreur. Estimation de la variance et du biais les deux sont combinés avec rms.
MSE est un contraste L'estimateur est dans le cas d'un estimateur sans biais. Ils ont les mêmes unités de mesure du carré de la quantité calculée comme la variance. MSE présente l'erreur L'écart quadratique moyen (RMSE) ou l'écart quadratique moyen (RMSD) est souvent appelé l'écart type de la variance. On peut également dire que MSE est l'écart moyen de la racine carrée (MSD) d'un estimateur. L'estimateur fait référence à la procédure de mesure de la quantité d'image non observée. Les mesures MSE ou MSD signifient des erreurs au carré. L'erreur est la différence entre le résultat estimé et le résultat estimé. C'est une fonction du risque, du calcul Compte tenu de la valeur attendue de la perte d'erreur au carré ou de la perte au carré. L'erreur quadratique moyenne (MSE) entre deux images comme g(x,y) g(x,y) et g^(x,y) $g^(x, y) est définie comme$

$$\boxed{\text{MSE=1MNMn=0Nm=1[g^(n,m)g(n,m)]2MSE=1MNn=0Mm=1N[g^(n, m)g(n, m)]2.....(1)}}$$

À partir de l'équation (1), nous pouvons voir que MSE est une représentation de l'erreur absolue. [24]

1.10.2 SSIM(Structure Similarity Index Method)

La méthode de l'indice de similarité structurelle est un modèle basé sur la perception. Dans cette méthode, la dégradation de l'image est considérée comme le changement dans la perception des informations structurelles. Il coopère également avec d'autres faits importants basés sur la perception tels que le masquage de luminance, le masquage de contraste, etc. Le terme informations structurelles met l'accent sur les pixels fortement interconnectés ou les pixels spatialement fermés. Les pixels hautement interconnectés indiquent des informations plus importantes sur les objets visibles dans le champ de l'image. Le masquage de luminance est un terme dans lequel la partie déformée d'une image est moins visible sur les bords de l'image.
D'autre part, le masquage de contraste est un terme dans lequel les distorsions sont moins perceptibles dans la texture de l'image.
SSIM valorise la qualité perçue des photos et vidéos. Mesure la similarité entre deux images : l'originale et l'image récupérée. [24]
La Figure 1.10 ci-dessous montre une Comparaison de SSIM et MSE pour l'image "Barbara" modifiée avec un bruit gaussien blanc additif. (a) Image originale ; (b) bruyant image ; (c) carte d'erreur absolue (plus claire indique une meilleure qualité/une plus petite différence absolue) ; (d) Carte d'indice SSIM (plus clair indique mieux qualité/valeur SSIM plus élevée).

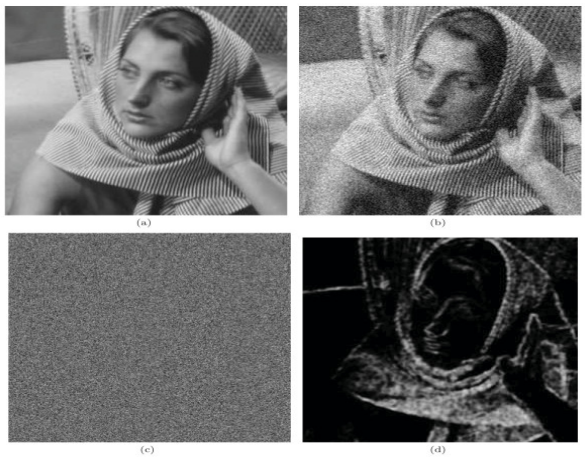


FIGURE 1.10 – Comparaison de SSIM et MSE
[25]

1.11 Formats d'images médicales

En image médicale le format standard est le format de fichier DICOM (Digital Imaging and Communications in Médicine), il est accepter à l'échelle internationale pour visualiser, partager, stocker, récupérer et collaborer les images médicales [30]. Les données de l'imagerie médicale DICOM ne peuvent pas être ouvertes par un logiciel d'imagerie standard, une version DICOM Médicale spéciale doit être installée pour récupérer, afficher, traiter et accéder aux fichiers d'image

médicale DICOM.
La Figure 1.11 présente la composition d'images médicales DICOM qui se composent de deux parties : un en-tête et l'image elle-même. L'en-tête se compose de données qui décrivent l'image et toutes les informations sur le patient (nom, âgé, sexe, etc.).

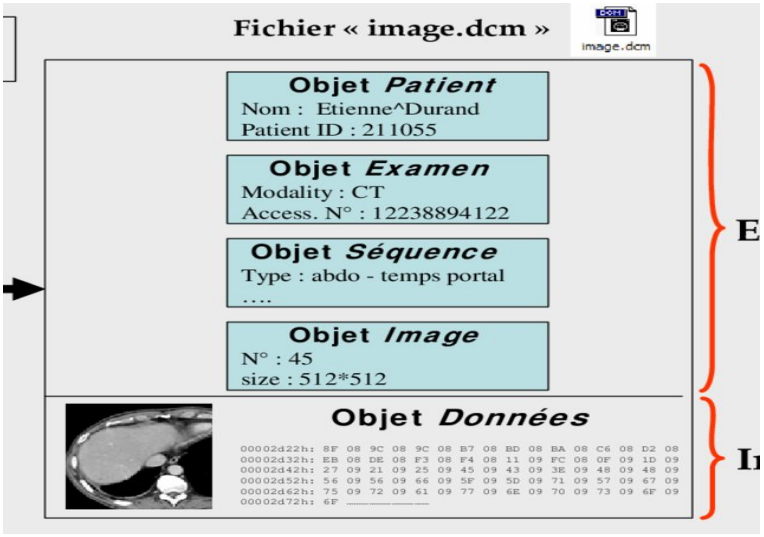


FIGURE 1.11 – Composition d'une image médicale DICOM [31]

Les principaux avantages d'utilisation du fichier DICOM :

- Amélioration de la qualité de l'image.
- Reconstruction d'images.
- Faire des mesures.
- Combinaison d'images médicales.

Les principaux inconvénients d'utilisation du fichier DICOM :

- Espace de stockage important pour ce type de format.

1.11.1 Compression d'image médicale sous DICOM

La compression permet de réduire la taille d'une image et de stocker plus d'images dans un même espace et de les transférer plus rapidement d'un point A à un point B. Ainsi, il y a deux grandes familles d'algorithmes :
Compression avec perte : le taux de compression est amélioré en contrepartie d'une perte d'information généralement indiscernable par l'œil humain.
Compression sans perte : il n'y a aucune perte d'information, lors de la décompression nous retrouvons exactement les mêmes données qu'au départ (avant compression).

1.12 Conclusion

Le domaine de l'imagerie médicale regroupe toutes les techniques utilisées en médecine pour diagnostiquer et traiter un grand nombre de maladies, qui fournissent des représentations visuelles basées sur des caractéristiques physiques ou chimiques spécifiques. Dans ce chapitre, nous avons présenté les images en général, puis les images médicales et leurs techniques importantes.

Chapitre 2

Cryptographie

2.1 Introduction

Le transfert d'images médicales est devenu une préoccupation majeure en médecine, et ce en raison des problèmes liés à la sécurité des données et à la confidentialité des patients. L'objectif de ce deuxième chapitre est d'apporter des solutions à ces problèmes, et de contribuer à la protection des données. Pour ce faire, nous utiliserons le concept de base des algorithmes de cryptage et nous présenterons une méthode de codage des données binaires permettant de faciliter le cryptage des images.

2.2 Objectifs de la cryptographie

Les principaux objectifs de la cryptographie se résument comme suit [13] :

- La confidentialité : Le texte chiffré ne doit être lisible que par les destinataires légitimes. Il ne doit pas pouvoir être lu par un intrus.
- L'authentification : Le destinataire d'un message doit pouvoir s'assurer de son origine. Un intrus ne doit pas être capable de se faire passer pour quelqu'un d'autre.
- L'intégrité : Le destinataire d'un message doit pouvoir vérifier que celui-ci n'a pas été modifié en chemin. Un intrus ne doit pas être capable de faire passer un faux message pour légitime.
- La non répudiation : Un expéditeur ne doit pas pouvoir, par la suite, nier à tort avoir envoyé un message.

2.3 Concepts fondamentaux de la cryptographie

Cette section présente les concepts fondamentaux de la cryptographie :

2.3.1 Cryptologie

Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse [9].

Cryptologie = Cryptographie + Cryptanalyse

2.3.2 Cryptographie

La cryptographie est l'art de chiffrer et coder les messages afin d'éviter les menaces et garantir la protection. Elle est utilisée pour stocker des informations sensibles.

2.3.3 Cryptanalyse

La cryptanalyse est la science ou l'art d'étudier des textes-chiffrés ou systèmes cryptographiques en utilisant des techniques mathématiques en vue d'identifier des faiblesses qui permettront la récupération du texte clair à partir du texte chiffré, sans nécessairement connaître la clé du chiffrement.

2.3.4 Cryptosystème

Le cryptosystème est composé de deux algorithmes, un pour le chiffrement et un autre pour le déchiffrement. Le cryptosystème est décrit par 5 quintuples (P, C, K, E, D) comme suit :

- P : Ensemble de textes clairs.
- C : Ensemble de textes chiffrés.
- K : Ensemble de clés.
- E : Ensemble de fonctions de chiffrement (cryptage) de P vers C.
- D : Ensemble de fonctions de déchiffrement (décryptage) de C vers P

2.3.5 Cryptogramme (texte chiffré)

Le cryptogramme est un message écrit à l'aide d'un système de chiffrement.

2.3.6 Chiffrement (Cryptage)

Le chiffrement est un processus de transformation d'un texte clair en une forme incompréhensible, pour toute personne non autorisée. Il désigne parfois le texte chiffré lui-même [10].

2.3.7 Déchiffrement (Décryptage)

Le déchiffrement est le processus de transformation inverse permettant d'avoir un message clair à partir d'un message crypté.

2.3.8 Message clair

Le message clair est le message réel avant toute modification de transformation.

2.3.9 Message chiffré

Le message chiffré est le résultat de chiffrement d'une donnée ou d'un message.

2.3.10 Clé

La clé est un paramètre impliqué dans les opérations de chiffrement et de déchiffrement. Elle est partagée entre l'émetteur et le récepteur.

2.3.11 Canal

Le canal est un moyen de transport de l'information.

2.3.12 Casser un code

Casser un code représente l'opération réalisée pour trouver la clé du code.

2.3.13 Algorithme cryptographique

Les algorithmes cryptographiques permettent de convertir des messages en clair (informations) à l'aide d'informations secrètes (clés de cryptage) pour générer des messages cryptés. La Figure 2.1 présente le protocole de chiffrement :

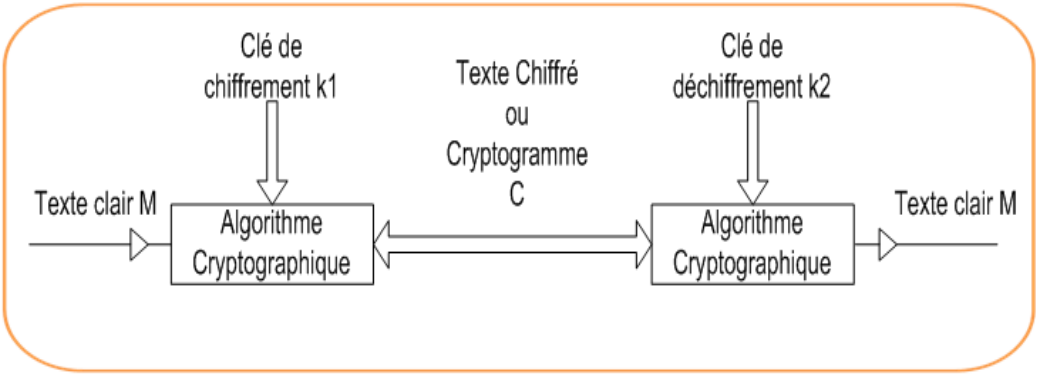


FIGURE 2.1 – Protocole de chiffrement

2.3.14 Confusion

La "confusion" signifie essentiellement que les équations qui produisent le texte chiffré sont trop compliquées, et ne peuvent pas être exploitées pour remonter des informations sur les messages clairs ou sur les clés utilisées dans le chiffrement [11].

2.3.15 Diffusion

La "diffusion" signifie que le changement d'un seul bit, du texte en clair ou de la clé, doit engendrer un changement majeur est imprévisible sur les bits du texte chiffré (plus de la moitié des bits du texte chiffré doivent changer de valeur) [11].

2.4 Techniques de cryptage classiques

Les chiffrements classiques sont des chiffrements antérieurs aux ordinateurs et fonctionnent donc sur des lettres plutôt que sur des bits [15]. La plupart des méthodes de chiffrement classiques reposent sur deux principes essentiels : la substitution et la transposition.

2.4.1 Chiffrement par substitution

Ce type de chiffrement consiste à remplacer chaque lettre par une autre lettre, selon une règle convenue. On distingue 3 types de chiffrement par substitution :

Mono-alphabétique

Le chiffrement mono-alphabétique consiste à remplacer chaque lettre du message d'origine par une autre, le principe de ce type de chiffrement permet de chiffrer un message avec seulement 26 façons. De plus, les fréquences d'apparition des lettres ne sont pas masquées, ce qui facilite sa cryptanalyse. Le chiffrement de César est un exemple de chiffrement mono-alphabétique [16].

Poly-alphabétique

Ce type de chiffrement consiste à remplacer chaque lettre du message par plusieurs lettres différentes. On citera le chiffrement le plus courant qui est celui de Vigenère, représentant une amélioration du chiffrement de César où une même lettre sera chiffrée de différentes manières en utilisant le carré de Vigenère [17].

Polygrammique

Un cryptage de substitution polygrammique est celui où les lettres sont remplacées par un bloc de plusieurs lettres. Le chiffrement Playfair [18] et celui de Hill [17] sont deux exemples du chiffrement polygrammique.

2.4.2 Chiffrement par transposition

Le chiffrement par transposition (ou par permutation) consiste à modifier l'ordre des lettres. Il exige la découpe du texte clair en blocs de taille identique, la même permutation est alors utilisée sur chacun des blocs. On distingue deux types de chiffrements par transposition :

Transposition simple par colonnes

Dans ce type de chiffrement, les lettres de message claire sont disposées dans un tableau de N colonnes. Le message est écrit horizontalement et les lettres sont lues verticalement. Le déchiffrent est le processus inverse du chiffrement.

Transposition complexe par colonnes

Dans ce type de chiffrement, une clé de caractères différents est utilisée pour constituer une séquence de chiffres représentant l'ordre d'apparition croissant des lettres alphabétiques. Le texte clair est ensuite écrit par ligne dans une matrice et le texte chiffré est lu par colonnes suivant l'ordre croissant de la séquence de chiffres.

2.5 Techniques de cryptage modernes

Les techniques de la cryptographie classique ne garantissent pas une forte sécurité et ne permettent que le chiffrement des données textuelles. Avec l'avancement des ordinateurs, les techniques cryptographiques ont considérablement évolué et peuvent manipuler des séquences de bits, comparés aux ordinateurs qui ne manipulent que des données numériques. Cela rend les techniques de cryptage actuelles plus sûres voire incassables avec certaines techniques.
Le chiffrement moderne peut généralement être classé en deux catégories : chiffrement symétrique ou à clé privée, chiffrement asymétrique ou à clé publique.

2.5.1 Chiffrement symétrique (à clé secrète)

La cryptographie symétrique est la plus ancienne. Dans ce type d'algorithme, la même clé est utilisée à la fois pour le chiffrement et le déchiffrement. Les clés doivent être distribuées secrètement entre l'émetteur et le récepteur. La Figure 2.2 représente le cryptosystème symétrique .

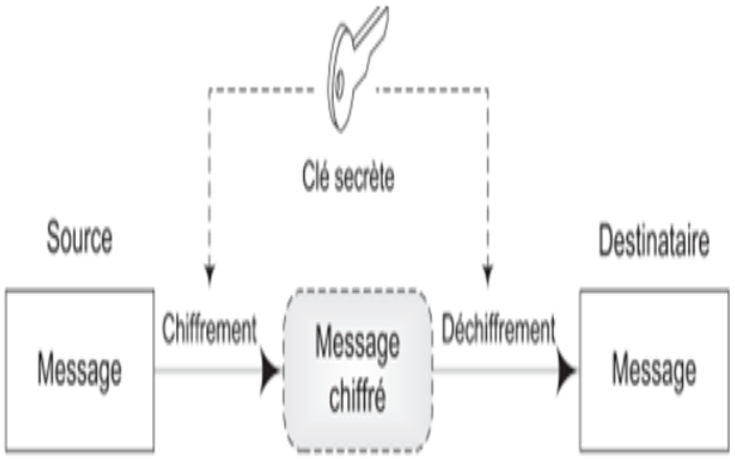


FIGURE 2.2 – Crypto-système symétrique

Les algorithmes de chiffrement symétrique sont de deux types :

1. Chiffrement par bloc

Le chiffrement par bloc est un schéma de chiffrement qui décompose les messages en clair à transmettre en chaînes (appelées blocs) d'une longueur fixe, et crypte un bloc à la fois. Les algorithmes de chiffrement symétrique par bloc les plus utilisés en pratique sont DES (Data Encryption Standard) [19] et AES [20] :

- **Algorithme de chiffrement DES**
  La méthode DES est adoptée comme standard par le gouvernement américain en 1977. Sa sécurité réside uniquement dans le secret de la clé, l'algorithme opère sur des blocs de 64 bits et utilise une clé de 56 bits. L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé.

- **Algorithme de chiffrement AES**
  L'AES a été adopté par le NIST (National Institute of Standards and Technology) en 2001, il remplace le DES qui devenait obsolète car il utilisait des clefs de 56 bits seulement.
  L'algorithme AES prend en entrée un bloc de 128 bits (16 octets). La clé est composée de 128, 192 ou 256 bits. Les 16 octets en entrée sont placés dans une matrice de 4x4 éléments.Les différentes opérations de chiffrement AES sont répétées plusieurs fois et définissent un (tour). Pour une clé de 128, 192 ou 256, AES nécessite respectivement 10, 12 ou 14 tours. Comme le montre la Figure 2.3 l'algorithme, AES est divisé en 3 blocs :

  1. Initial Round : C'est la première étape et la plus simple. Il n'y a qu'une seule opération : AddRoundKey.
  2. N Rounds : N est le nombre d'itérations. Celui-ci varie en fonction de la taille de la clé utilisée. N=9 est de 128 bits, N=11 est de 192 bits et N=13 est de 256 bits. La deuxième étape consiste en N itérations, dont chacune comprend les quatre opérations suivantes : Sub Bytes, Shift Rows, Mix Columns, Add Round Key.
  3. Final Round : Cette étape est quasiment identique à l'une des N itérations de la deuxième étape. La seule différence est qu'il n'y a pas d'opération Mix Columns.

  Sachant que :

  – SubBytes : signifie que chaque entrée est remplacée par un autre mot de 8 bits donné par un tableau de correspondance (Table 2.1) .

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

TABLE 2.1 – Tableau de substitution (Sbox)

– ShiftRows : signifie que les entrées sont décalées suivant un décalage circulaire à gauche d'un nombre de cases dépendante de la ligne. La première ligne n'est jamais décalée étant décalée de 0, la seconde de 1 et ainsi de suite.

– MixColumns : signifie que chaque colonne est remplacée par une nouvelle colonne obtenue en multipliant la colonne par une ligne d'une matrice fixée comme suite :

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

– AddRoundKey : signifie que chaque entrée est remplacée par le XOR entre cette entrée et l'entrée correspondante dans une matrice 4 * 4 construite à partir de la clé.

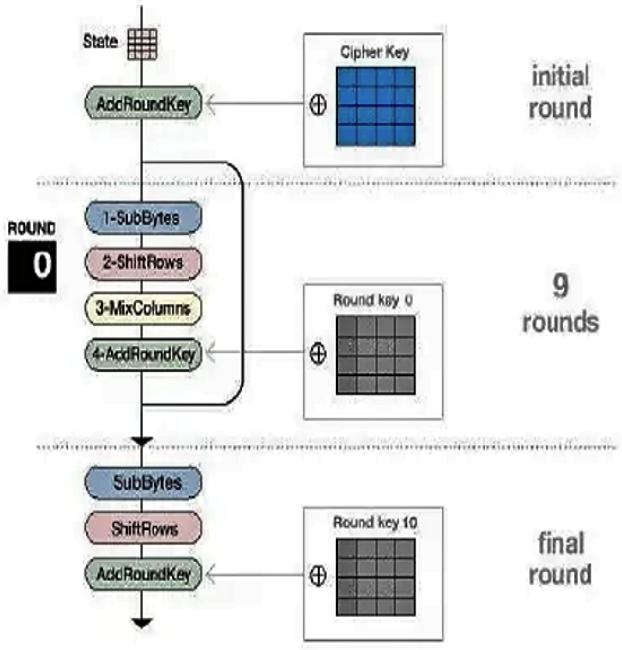


FIGURE 2.3 – Chiffrement AES [6]

L'algorithme AES peut être décrit comme suit :

Algorithm 1 Chiffrement AES

Input key, state

1: KeyExpansion(key,RoundKeys) ;
2: AddRoundKey($state$,RoundKeys[0]) ;
3: **for** $r = 1$ to N_r-1 **do**
4: SubBytes($state$) ;
5: ShiftRows($state$) ;
6: MixColumns($state$) ;
7: AddRoundKey($state, RoundKeys[r]$) ;
8: **end for**
9: SubBytes($state$) ;
10: ShiftRows($state$) ;
11: AddRoundKey($state, RoundKeys[N_r]$) ;

La génération (expansion) des clés

La procédure d'obtention de toutes les clés rondes à partir de la clé d'entrée originale de 16 octets est définie par l'expansion de clé. Dans le chiffrement, les clés d'origine sont la clé ronde initiale et dans le déchiffrement, les clés d'origine seront le dernier groupe généré par

l'expansion de clé. Avant que les itérations ne commencent le chiffrement ou le déchiffrement, les entrées sont ajoutées avec la clé de tour initiale. La clé initiale se présente sous la forme d'une matrice 4*4. Les trois étapes d'expansion de la clé sont : RotWord, SubWord, XOR.
RotWord : signifie le décalages des octets, chaque octet est remonté cycliquement dans la quatrième colonne.
SubWord : les résultats du RotWord sont ensuite substitués avec la table S-Box.
XOR : pour obtenir la première sous-clé de la première colonne, une opération XOR est effectuée avec la première colonne de la clé de chiffrement et le Rcon(1) (Tableau 2.2). Les résultats de l'opération sont ensuite utilisés pour obtenir la colonne suivante en effectuant un XOR sur la colonne avec la clé de chiffrement correspondante et ainsi de suite.

Rcon(1)	01	00	00	00
Rcon(2)	02	00	00	00
Rcon(3)	04	00	00	00
Rcon(4)	08	00	00	00
Rcon(5)	10	00	00	00
Rcon(6)	20	00	00	00
Rcon(7)	40	00	00	00
Rcon(8)	80	00	00	00
Rcon(9)	1b	00	00	00
Rcon(10)	36	00	00	00

TABLE 2.2 – Tableau de Rcon(i) avec i=[1..10]

2. Chiffrement par flot

Dans le chiffrement par flot, les données sont traitées en flux (traitement bit par bit). Le principe consiste à générer un flux pseudo aléatoire et le combiner avec l'information bit à bit par l'opération XOR. La taille de la clé est égale à la taille du message. Les algorithmes de chiffrement symétrique par flot les plus célèbres sont RC4 (Rivest Cipher 4).

- **Algorithme de chiffrement RC4**
  Conçu par Ron Rivest de RSA Security en 1987, RC4 fonctionne comme suit : La clé RC4 est utilisée pour initialiser une table de 256 octets en répétant la clé autant de fois que nécessaire pour remplir la table. Ensuite, de simples opérations sont effectuées : déplacement d'octets dans le tableau, réalisation d'additions, etc. Le but étant de mélanger autant de planches que possible. Enfin, nous aurons une séquence de bits pseudo-aléatoire qui peut être utilisée pour chiffrer les données via XOR.

2.5.2 Chiffrement asymétrique (à clé publique)

La cryptographie à clé publique, ou cryptographie asymétrique est une méthode de chiffrement qui s'oppose à la cryptographie symétrique. Elle utilise deux clés différentes : une clé publique (mises à la disposition de quiconque) qui sert à chiffrer et une clé privée (gardée secrète) qui sert à déchiffrer. La Figure 2.4 représente le principe de chiffrement asymétrique.

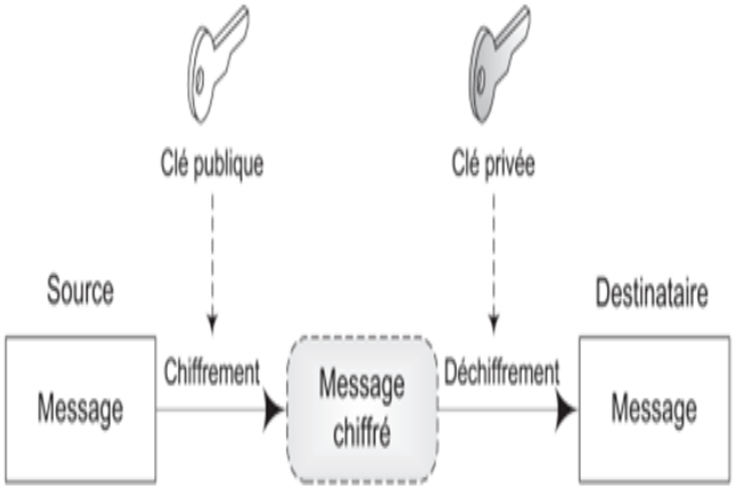


FIGURE 2.4 – Crypto-système asymétrique

Il existe plusieurs systèmes asymétriques, les plus célèbres sont : RSA (Rivest, Shamir, Adleman) [21] et Elgmal [22].

- **Chiffrement de RSA**

  L'algorithme repose sur le principe de chiffrement asymétrique selon lequel deux clés différentes doivent être utilisées. La clé publique peut être partagée avec quiconque, tandis que la clé privée doit rester secrète.
  L'algorithme de chiffrement de RSA tire sa sécurité du problème de factorisation des grands nombres entiers. Il est souvent utilisé pour la sécurisation des données confidentielles, en particulier lorsqu'elles sont transmises sur un réseau peu sûr comme Internet.

- **Chiffrement d'Elgamal**

  L'algorithme de chiffrement d'Elgamal tire sa sécurité du problème du Logarithme discret.

2.5.3 Cryptographie symétrique vs asymétrique

Les schémas de chiffrement à clé symétrique et à clé publique ont divers avantages et inconvénients, dont certains sont communs aux deux [12].

Avantages de la cryptographie symétrique

- Les algorithmes de chiffrements symétriques peuvent être conçus pour avoir des taux élevés de débit de chiffrement de données. Certaines implémentations matérielles atteignent des taux de plusieurs centaines de mégaoctets par seconde, alors que les implémentations logicielles peuvent atteindre des débits de mégaoctets par seconde.
- Les clés dans les cryptosystèmes symétriques sont relativement de petite taille.
- Les algorithmes de chiffrement à clé symétrique peuvent être utilisés comme des primitives pour construire divers mécanismes cryptographiques, y compris des générateurs de nombres pseudo-aléatoires, des fonctions de hachage et des schémas de signature numérique.
- Les chiffres à clé symétrique peuvent être composés pour produire des chiffres plus forts. Ils sont basés sur des transformations simples qui sont faciles à analyser. Toutefois, des chiffres forts peuvent être construits en utilisant leur propre faiblesse.

Inconvénients de la cryptographie symétrique

- Dans une communication entre deux entités, la clé doit rester secrète aux deux extrémités.
- Dans un grand réseau, il y a beaucoup de paires de clés à gérer. Par conséquent, une gestion efficace des clés nécessite l'utilisation d'une autorité de confiance.
- Dans une communication à deux parties entre deux entités A et B, la pratique cryptographique dicte que la clé doit être changée fréquemment, et ce pour chaque session de communication.
- Les mécanismes de signature numérique provenant du cryptage à clé symétrique nécessitent généralement de grandes clés pour la fonction de vérification ou l'utilisation d'une autorité de confiance.

Avantages de la cryptographie asymétrique

- Seule la clé privée doit être gardée secrète.
- Selon le mode d'utilisation, une paire de clé peut rester inchangée pour des périodes de temps considérables, par exemple, de nombreuses sessions (voire plusieurs années)
- Dans un grand réseau, le nombre de clés nécessaires pourrait être considérablement plus petit que dans le scénario de clé symétrique.

Inconvénients de la cryptographie asymétrique

- Les débits pour les méthodes les plus populaires de chiffrement à clé publique sont plus lents que les schémas à clés symétriques les plus connus.
- Les tailles de clé sont généralement beaucoup plus grandes que celles requises pour le chiffrement à clé symétrique, et la taille des signatures à clé publique est plus grande que celle des étiquettes fournissant l'authentification de l'origine des données à partir des techniques symétriques.
- Aucun système à clé publique n'a été prouvé pour être sûr (la même chose peut être dite pour le chiffrement par blocs). La sécurité des systèmes de chiffrement à clé publique les plus efficaces à ce jour repose sur la difficulté supposée d'un petit ensemble de problèmes numériques théoriques.
- La cryptographie à clé publique n'a pas un aussi vaste historique que le cryptage à clé symétrique, puisque elle a été découverte seulement dans le milieu des années 1970.

2.5.4 Signature numérique

La signature numérique se base principalement sur la cryptographie asymétrique. Elle consiste à générer un condensé à partir d'un message et le chiffrer en utilisant la clé privée de l'émetteur. Ce dernier, envoie ensuite le message en clair avec la signature. Afin de vérifier la validité de cette signature, le récepteur recalcule le condensé à partir du message en clair et déchiffre la signature. Puis, il vérifie l'égalité des deux résultats.

2.5.5 Fonctions de hachage

Une fonction de hachage convertit des séquences de caractères de longueurs différentes en séquences de même longueur. Employées pour l'authentification et les signatures numériques.

2.6 Codage base 64

L'encodage Base 64 est un algorithme qui convertit des données binaires (image, son, vidéo) en une représentation ASCII (American Standard Code for Information Interchange). Ce n'est pas un algorithme de cryptage, mais peut être utilisé pour coder les données avant le cryptage. Le codage en Base 64 commence par découper le message en groupe de 6 bits (sextets), complété avec des 0 si besoin. Puis, il remplace chaque sextet par son caractère correspondant (Table 2.3). Le Tableau 2.3 montre les caractères (26 lettres majuscules (A, ..., Z), 26 lettres minuscules (a, ..., z), 10 chiffres décimaux (0, ..., 9), deux caractères + et /.) associés à chaque sextet (64 sextets).

Valeur	**sextet**	**caractère**	Valeur	**sextet**	**caractère**
0	000000	A	**1**	000001	B
2	000010	C	**3**	000011	D
4	000100	E	**5**	000101	F
6	000110	G	**7**	000111	H
8	001000	I	**9**	001001	J
10	001010	K	**11**	001011	L
12	001100	M	**13**	001101	N
14	001110	O	**15**	001111	P
16	010000	Q	**17**	010001	R
18	010010	S	**19**	010011	T
20	010100	U	**21**	010101	V
22	010110	W	**23**	010111	X
24	011000	Y	**25**	011001	Z
26	011010	a	**27**	011011	b
28	011100	c	**29**	011101	d
30	011110	e	**31**	011111	f
32	100000	g	**33**	100001	h
34	100010	i	**35**	100011	j
36	100100	k	**37**	100101	l
38	100110	m	**39**	100111	n
40	101000	o	**41**	101001	p
42	101010	q	**43**	101011	r
44	101100	s	**45**	101101	t
46	101110	u	**47**	101111	v
48	110000	w	**49**	110001	x
50	110010	y	**51**	110011	z
52	110100	0	**53**	110101	1
54	110110	2	**55**	110111	3
56	111000	4	**57**	111001	5
58	111010	6	**59**	1110011	7
60	111100	8	**61**	111101	9
62	111110	+	**63**	1111111	/

TABLE 2.3 – Tableau de codage base 64

En parcourant les données binaires de gauche à droite, des groupes de 24 bits sont créés en concaténant des blocs de 3 données de 8 bits. Chaque groupe de 24 bits est ensuite divisé en 4

groupes de 6 bits, correspondant à 4 caractères de l'alphabet Base64.

L'encodage Base64 est prévu pour des données formant un multiple de 24 bits. Ainsi, si le volume des données à coder ne forme pas un multiple de 24 bits, le résultat du codage Base64 doit être complété par 0 à 3 caractères (=) afin d'obtenir un multiple de 24 bits. Ce 65ème caractère ne peut ainsi être présent qu'à la fin des données encodées.

Par ailleurs, afin de garantir une compatibilité avec l'ensemble des systèmes, les données Base64 sont formatées avec des retours à la ligne pour que chaque ligne ne dépasse pas 76 caractères.

Exemple 1

Le mot ABC, codé en ASCII, a une longueur de $3 \times 8 = 24$ bits(multiple de 6) :

01000001 01000010 01000011

Ce qui donne un découpage de 4 sextets :

010000 010100 001001 000011

Le résultat de l'encodage selon la table de codage base 64 (Tableau 3.1) est comme suit :

QUJD

Exemple 2

Le mot salut, codé en ASCII, a une longueur de $5 \times 8 = 40$ bits (n'est pas multiple de 6) :

01010011 01100001 01101100 01110101 01110100

Ce qui donne un découpage de 6 sextets + 4 bits. Il convient donc d'ajouter un sextet obtenu en ajoutant 2 bits nuls :

010100 110110 000101 101100 011101 010111 010000

Le résultat de l'encodage selon la table de codage base 64 (Tableau 2.1) est comme suit :

U2FsdXQ=

2.7 Conclusion

Dans ce chapitre, nous avons présenté les notions de base de la cryptographie, les techniques de cryptage classique et moderne, ainsi que la technique d'encodage base 64.

Chapitre 3

Analyse et conception

3.1 Introduction

L'analyse des besoins est la première étape de la conception, elle consiste à analyser la situation pour prendre en considération des contraintes et des risques. Ce chapitre décrit la problématique et les objectifs de notre application, les solutions proposées, le processus de développement, le langage de modélisation choisis, ainsi que les besoins fonctionnels et non fonctionnels de l'application. Un diagramme de cas d'utilisation global est établit avec une description détaillée des cas d'utilisation. Enfin, la conception de notre application est présentée, en commençant par le diagramme de séquence, puis le diagramme de classe.

3.2 Problématique et Objectifs

L'imagerie médicale joue un rôle important dans le diagnostic de l'état des patients. Il est essentiel de protéger les images médicales lors de leur transmission entre hôpitaux, médecins, etc. En effet, leur détérioration ou divulgation peut être à l'origine de dommages pour la santé d'un patient. Bien qu'il existe de nombreux systèmes de transmission d'images médicales, celles-ci présentent de nombreux problèmes liés à la sécurité de la transmission d'images, dont les plus importants sont :

- Transmission non sécurisée d'images médicales, elles peuvent être manipulées intentionnellement par des utilisateurs non autorisés.
- Les données peuvent être capturées et modifiées.
- Risques liés à la confidentialité des patients.
- Les images sont transmises par e-mail ou d'autres applications utilisant une connexion Internet, mais s'il n'y a pas de connexion Internet dans des situations critique(cas des urgences) comment pouvons-nous envoyer les images médicales aux personnels médicales ?

Dans ce contexte, une question se pose : comment peut-on concevoir un système pour assurer la sécurité de ce type de données ? et comment partager sans avoir besoin d'une connexion Internet ?

Ce présent travail permettra d'y répondre, à travers le développement et l'implémentation d'un système pouvant résoudre les problèmes susmentionnés, à savoir ceux de sécurité et de confidentialité.

3.2.1 Solutions

Pour faciliter au personnel médical (patients, médecins, etc) le transfert d'images médicales en toute sécurité, nous développerons à travers ce projet une application Android simple à utiliser. Celle-ci permet :

- L'envoi d'images médicales après les avoir rendues incompréhensibles à l'aide des mécanismes de sécurité actuels, tels que les techniques de chiffrement qui offrent une protection pour empêcher les utilisateurs non autorisés d'accéder au contenu des images médicales.
- La réception des images médicales après les avoir décryptées pour les rendre compréhensibles pour une personne autorisée.
- L'enregistrement des images médicales dans le smartphone des récepteurs.
- Assurer la sécurité des images médicales et le respect de la confidentialité des patients ainsi que le secret médical.
- Permettra à tous les téléphones de transférer des images d'un téléphone à un autre, même s'ils sont éloignés et n'ont pas de connexion Internet.
- Diagnostiquer l'état du patient sans déplacement.

3.3 Méthode de développement

Il existe plusieurs méthodes de développement logiciel parmi eux : UP(Unified Process), RUP (Rational Unified Process), 2TUP (2 Track Unified Process), XP (Extreme Programming),etc. Suivant la nature de notre projet, la méthode utilisée est celle de UP. Cette dernière serait la mieux adaptée à la réalisation de notre application.

3.3.1 Processus Unifié

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, différents types d'entreprises, différents niveaux de compétences et différentes tailles d'entreprises [32].

3.3.2 Caractéristiques de Up

1. UP est itératif

Chaque étape de la méthodologie UP comprend des itérations. Une itération est un cycle de développement logiciel complet, de la collecte des exigences à la mise en œuvre et aux tests. La Figure 3.1 illustre le déroulement du processus unifié :

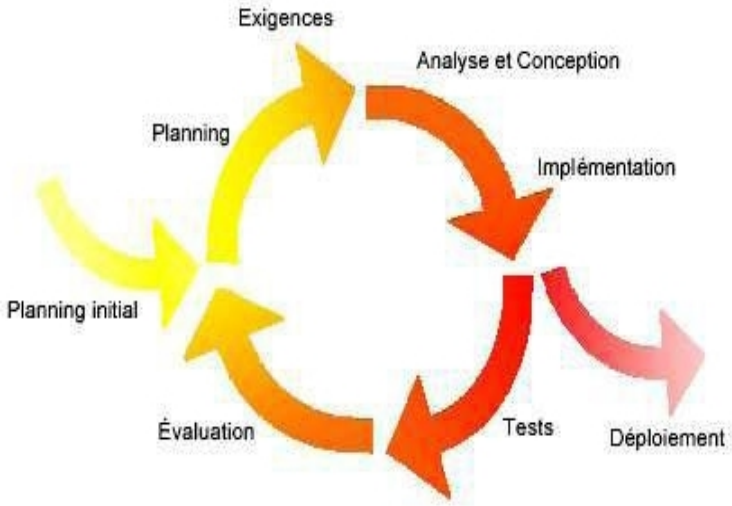


FIGURE 3.1 – Déroulement du processus unifié [32]

2. UP est centré sur l'architecture

Dès le démarrage du processus, on aura une vue sur l'architecture à mettre en place. Celle-ci est dérivée des exigences métier, exprimées par les utilisateurs et d'autres parties intervenants reflétés par les cas d'utilisation.

3. UP est piloté par les cas d'utilisation d'UML

Le but principal d'un système informatique est de satisfaire les besoins du client. Le processus de développement sera donc accès sur l'utilisateur.
Les cas d'utilisation permettent d'illustrer les besoins : Ils détectent et décrivent les exigences fonctionnelles (du point de vue de l'utilisateur), et leur ensemble forme un modèle de cas d'utilisation qui indique les fonctionnalités complètes du système.

3.3.3 Phases de UP

La méthode UP se base sur quatre phases :

1. L'analyse des besoins

L'analyse des besoins comprend le cahier des charges et d'écrie les spécifications internes présentant la manière d'implémenter le système.
Cette phase porte essentiellement sur les utilisateurs de système, l'architecture générale du système, les risques majeurs, les délais et les coûts.

2. Élaboration

L'élaboration reprend les éléments de la phase précédente (analyse des besoins) et les précise pour arriver à une spécification détaillée de la solution à mettre en œuvre. Elle permet d'analyser le domaine du problème, de construire l'architecture de base, de résoudre les éléments à haut risque et de définir la plupart des exigences.

3. Construction

La construction est le moment où l'on construit le produit. Elle assure la finalisation de l'analyse, la conception, l'implémentation et les tests , ainsi que la transformation de l'architecture de référence en un produit complet. Et ce en veillant à respecter son intégrité.

4. Transition

La transition est la phase de livraison du produit au client afin de permettre des essais par les utilisateurs et de détecter les anomalies et défauts.

3.3.4 Activités du processus unifié

Chaque phase est constituée d'une succession d'activités. Les activités du processus UP sont les suivantes :

1. Expression des besoins

C'est la compréhension et l'expression des besoins fonctionnelles et non fonctionnelles, ainsi que la livraison d'une liste comprenant les exigences du client.

2. Analyse

L'analyse est l'activité de préparation à la conception. Elle permet d'accéder à une compréhension des besoins et des exigences du client, et aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.

3. Conception

La conception permet d'acquérir une compréhension approfondie des contraintes liées au langage de programmation, à l'utilisation des composants et au système d'exploitation. Elle détermine les principales interfaces.

4. Implémentation

C'est le résultat de la conception, on implémente le système sous forme de composants(de code source, de scripts, d'exécutable et d'autres éléments du même type.). Elle a pour objectif de planifier l'intégration et de produire les classes et les sous-systèmes sous formes de codes sources.

5. Tests

Les tests permettent de vérifier les résultats de l'implémentation de toutes les exigences en testant la construction, et de s'assurer de la bonne intégration de tous les composants dans le logiciel.

La Figure 3.2 décrit les différentes activités et les phases de UP :

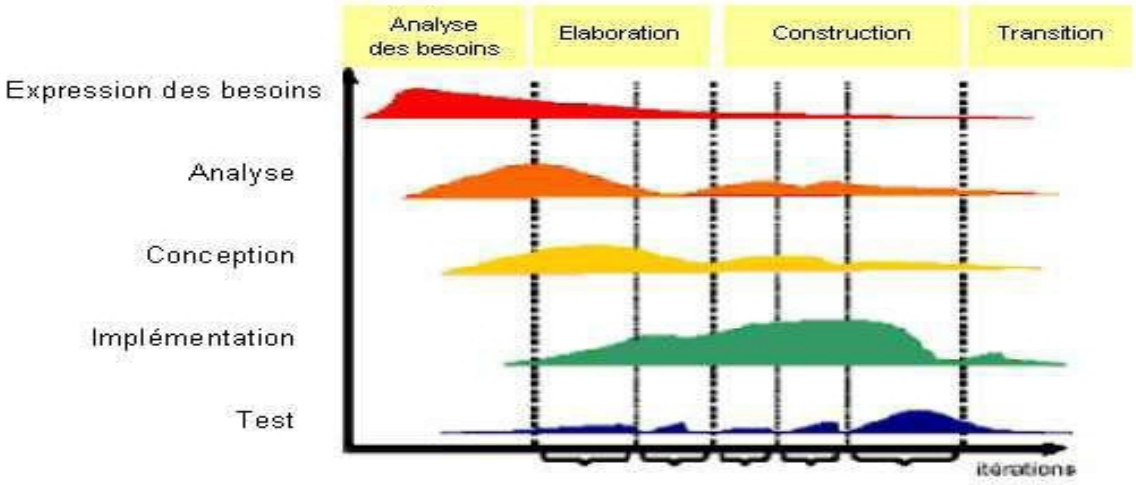


FIGURE 3.2 – Description du processus unifié [32]

3.4 Langage de modélisation

Comme le processus unifié exige l'utilisation d'UML (Unified Modeling Language), notre modélisation de la solution se fera en utilisant des diagrammes UML.

3.4.1 UML

UML ou Langage de Modélisation Unifié, est un langage de modélisation utilisé pour former des diagrammes. Ces derniers sont utilisés pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML est apparu dans le cadre de la conception orientée objet, il est dit universel car il est indépendant des langages de programmation.

3.5 Spécification des besoins

Les besoins de notre application se divisent en deux types : besoins fonctionnels et besoins non fonctionnels.

3.5.1 Besoins fonctionnels

Le système à réaliser comportera un ensemble de fonctionnalités qui doivent être mises en relation avec un ensemble de besoins utilisateur.
L'application doit accomplir les traitements suivants :
Pour le chiffrement :

- Sélectionner l'image à crypter.
- Crypter l'image.
- Envoyer le code d'image crypté (la clé de sécurité et le hashcode).

Pour le déchiffrement :

- Recevoir le code d'image crypté(la clé de sécurité et le hashcode).
- Décrypter.
- Enregistrer l'image.

3.5.2 Besoins non fonctionnels

- L'extensibilité de l'application .
- L'ergonomie de l'interface du fait qu'elle doit permettre aux utilisateurs d'apprendre à l'utiliser facilement.
- La sécurité.

3.6 Analyse des besoins

L'analyse des besoins vise à identifier les exigences ainsi que les acteurs du système et les tâches associées à chacun. On parle également de cadrage de projet.

3.6.1 Identification des acteurs

Un acteur représente un rôle joué par une entité externe (utilisateur humain, dispositif matériel ou autre système) qui interagit directement avec le système étudié.
Un acteur peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [14].
L'étude préliminaire des besoins fonctionnels a révélé la présence de 2 acteurs :
Émetteur : Le responsable du cryptage et de l'envoie des images médicales.
Récepteur : Le responsable de la réception et du décryptage des images médicales.

3.6.2 Identification des cas d'utilisations

Un cas d'utilisation (use case) représente un ensemble de séquences d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier.
L'ensemble des cas d'utilisation doit décrire exhaustivement les exigences fonctionnelles du système. Chaque cas d'utilisation correspond donc à une fonction métier du système, selon le point de vue d'un de ses acteurs [14].
Le Tableau 3.1 illustre l'ensemble des cas d'utilisations de notre système et l'acteur de chaque cas.

Num de cas	Acteur	Cas d'utilisation
1	Sélectionner une image médicale	Émetteur
3	Crypter l'image	Émetteur
4	Envoyer le code de l'image crypté	Émetteur
5	Recevoir le code de l'image crypté	Récepteur
7	Décrypter l'image	Récepteur
8	Enregistrer l'image	Récepteur

Table 3.1 – Tableau des cas d'utilisations

3.6.3 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation décrivent les utilisations requises d'un système, ou ce qu'un système est supposé faire. La Figure 3.3 présente le diagramme de cas d'utilisation de notre application.

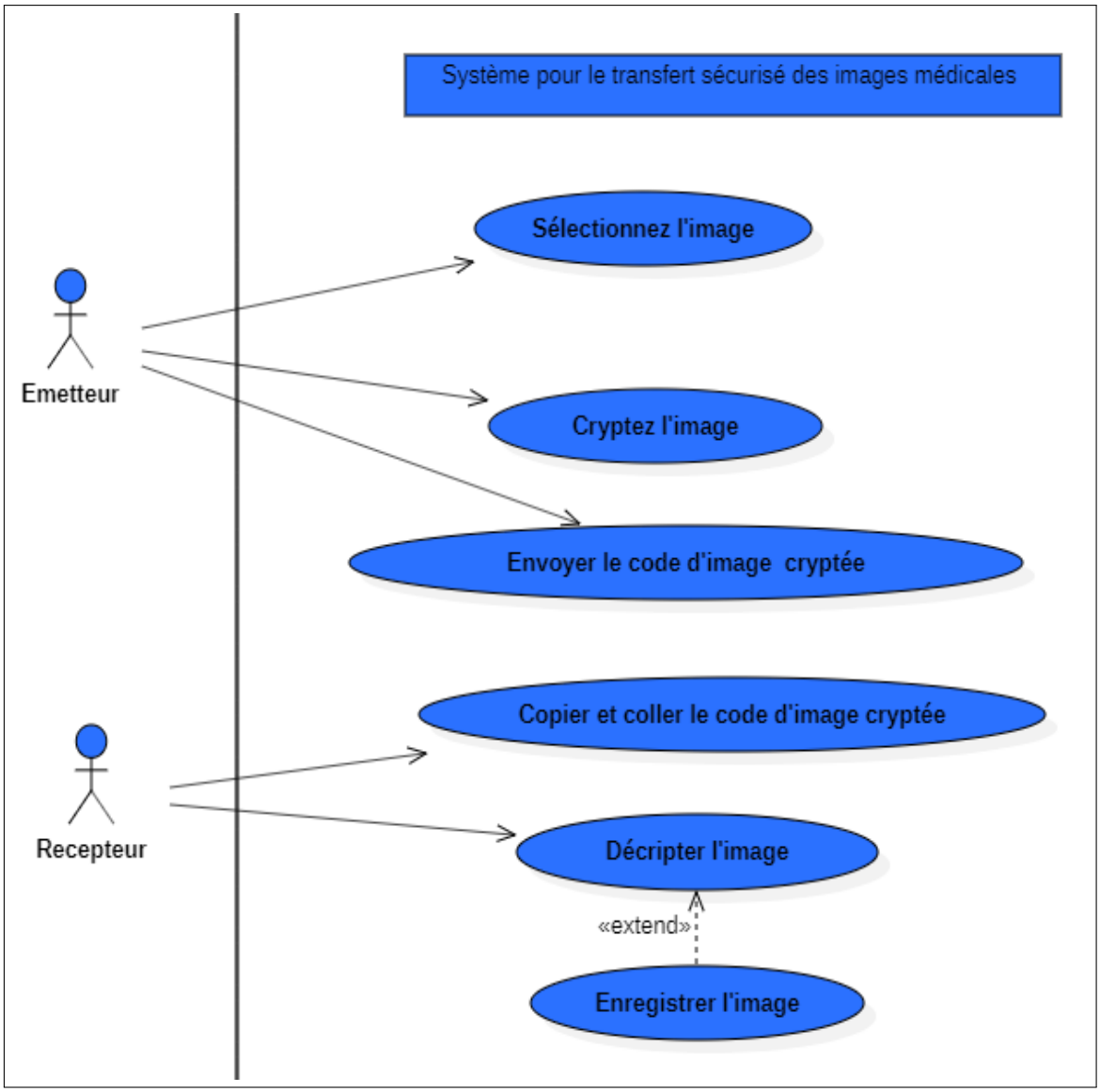


FIGURE 3.3 – Diagramme de cas d'utilisation

3.7 Description des cas d'utilisation

3.7.1 Sélectionner une image médicale

Objectif	Permet de choisir une image médicale à chiffrer
Acteur	Émetteur
Scénario nominal	1-L'émetteur sélectionne une image médicale 2-L'application affiche l'image sélectionnée

TABLE 3.2 – Description du cas d'utilisation (Sélectionner une image médicale)

3.7.2 Crypter l'image médicale

Objectif	Permet le chiffrement d'une image médicale pour la rendre incompréhensible
Acteur	Émetteur
Pré-conditions	Sélectionne une image médicale
Enchaînement nominal	1-L'émetteur clique sur le bouton crypter 2-L'application crypte l'image à l'aide de l'algorithme proposé 3-L'application affiche l'image cryptée (deux codes l'un est l'image cryptée et l'autre est la clé) à l'émetteur

TABLE 3.3 – Description du cas d'utilisation (Crypter l'image médicale)

3.7.3 Envoyer l'image

Objectif	Permet l'envoie d'une image par SMS
Pré-conditions	Crypter une l'image
Acteur	Émetteur
Enchaînement nominal	1-L'émetteur clique sur le bouton envoyer 2-L'application affiche l'interface de l'envoie 3-L'émetteur saisi le numéro de la personne à qui envoyer l'image crypté (que le hashcode que nous obtenons dans le cryptage). 4-L'émetteur clique sur OK pour envoyer l'image ou Annuler pour annuler l'envoi. 5-L'émetteur envoie le clé de sécurité au récepteur après la confirmation de la réception de code de l'image par le récepteur par un message de confirmation.
Enchaînement Alternative	A1 : L'enchaînement démarre après le point 4 de la séquence nominale : Le système indique par un message d'erreur qu'il faut introduire le numéro de téléphone du récepteur L'enchaînement reprend à l'étape 3 du scénario nominal

TABLE 3.4 – Description du cas d'utilisation (Envoyer l'image)

3.7.4 Recevoir le code de l'image chiffrée

Objectif	Permet de recevoir le code de l'image crypté reçu
Acteur	Récepteur
Enchaînement nominal	1-Le récepteur reçoit le code de l'image cryptée dans un SMS. 2-Le récepteur confirme la réception de code de l'image cryptée. 3-Le récepteur reçoit la clé de sécurité dans un SMS.

TABLE 3.5 – Description du cas d'utilisation (Recevoir le code de l'image chiffré)

3.7.5 Décrypter l'image

Objectif	Permet le déchiffrement d'une image médicale pour la rendre claire
Acteur	Récepteur
Pré-conditions	Recevoir le code de l'image cryptée
Enchaînement nominal	1-Le récepteur colle le code de la clé et le hashcode de l'image crypté 2-Le récepteur clique sur le bouton décrypter 3-L'application décrypte la chaîne de caractères en la convertissant en image claire. 4-L'application affiche l'image médicale.
Enchaînement Alternative	A1 : L'enchaînement démarre après le point 2 de la séquence nominale : Le système indique par un message d'erreur que la clé de cryptage ou le code d'image crypté est incorrect L'enchaînement reprend à l'étape 1 du scénario nominal

TABLE 3.6 – Description du cas d'utilisation (Décrypter l'image)

3.7.6 Enregistrer l'image

Objectif	Permet d'enregistrer l'image médicale dans le smartphone de récepteur
Acteur	Récepteur
Pré-conditions	Décrypter l'image
Enchaînement nominal	1-Le récepteur clique sur le bouton enregistrer 2-L'application enregistre l'image dans le smartphone de récepteur

TABLE 3.7 – Description du cas d'utilisation (Enregistrer l'image)

3.8 Conception

3.8.1 Diagrammes de séquence

Un diagramme de séquence est un diagramme UML d'interaction qui expose en détail les lignes de vie, les processus et les objets , ainsi que la séquence de messages transmis entre des objets.
La Figure 3.4 représente le diagramme de séquence de notre application :

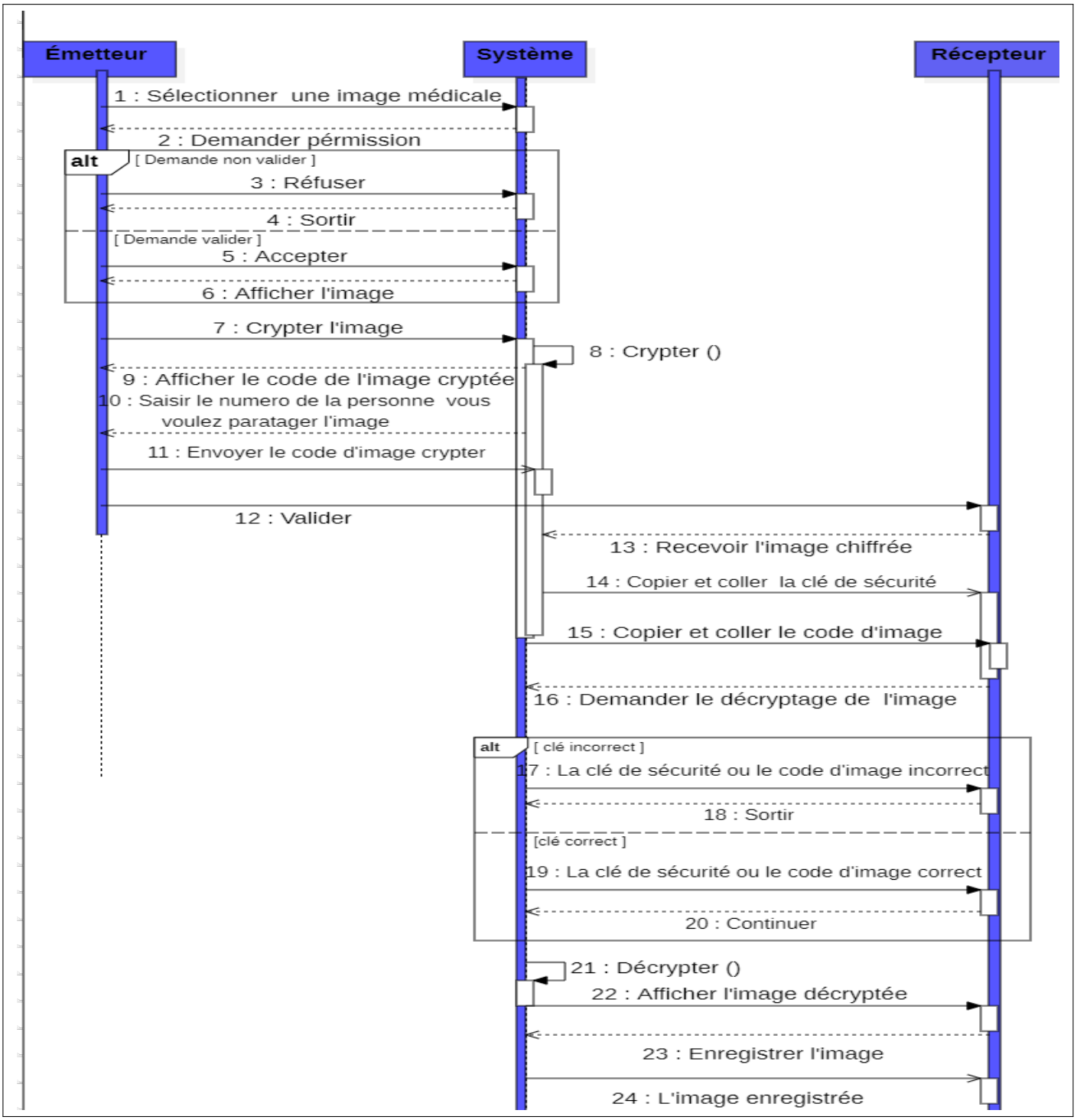


FIGURE 3.4 – Diagramme de séquence

3.8.2 Dictionnaire de données

Le tableau suivant d'écrit et explique toutes les données relatives aux classes de notre application.

Classe	**Codification**	**Type**	Désignation	Méthodes
Code de l'image	cle	string	la clé de chiffrement	EnvoyerImage()
	hashcode	byte[]	le code de l'image cryptée	
Emetteur	numero	int	le numero de telephone de l'emetteur	ChargerImage()
	objetpatient	string	contient le nom et id du patient	
	objetexamen	string	contient le modèle de l'examen	
	objetequence	string	séquence d'élements	
	objetimage	string	contient le num et la taille de l'image	
	objetdonnees	string	contient les pixels d'image	
Recepteur	numero	int	le numéro de telephone de recepteur	Enregistrer()
Cryptage	cle	string	la clé sucret de chiffrement	Crypter()
	image	bitmap	l'image a crypter	
Decryptage	cle	byte[]	la clé sucret de dechiffrement	Decrypter()
	codeimage	byte[]	le code de l'image obtenu par cryptage	

3.8.3 Diagramme de classes

Le diagramme de classes est le point central dans un développement orienté objet. En analyse, il a pour objectif de décrire la structure des entités manipulées par les utilisateurs. En conception, le diagramme de classes représente la structure d'un code orienté objet ou, à un niveau de détail plus important, les modules du langage de développement. Le diagramme de classe met en œuvre des classes, contenant des attributs et des opérations, et reliées par des associations ou des généralisations [14].
le Figure 3.5 illustre le diagramme de classes de notre système.

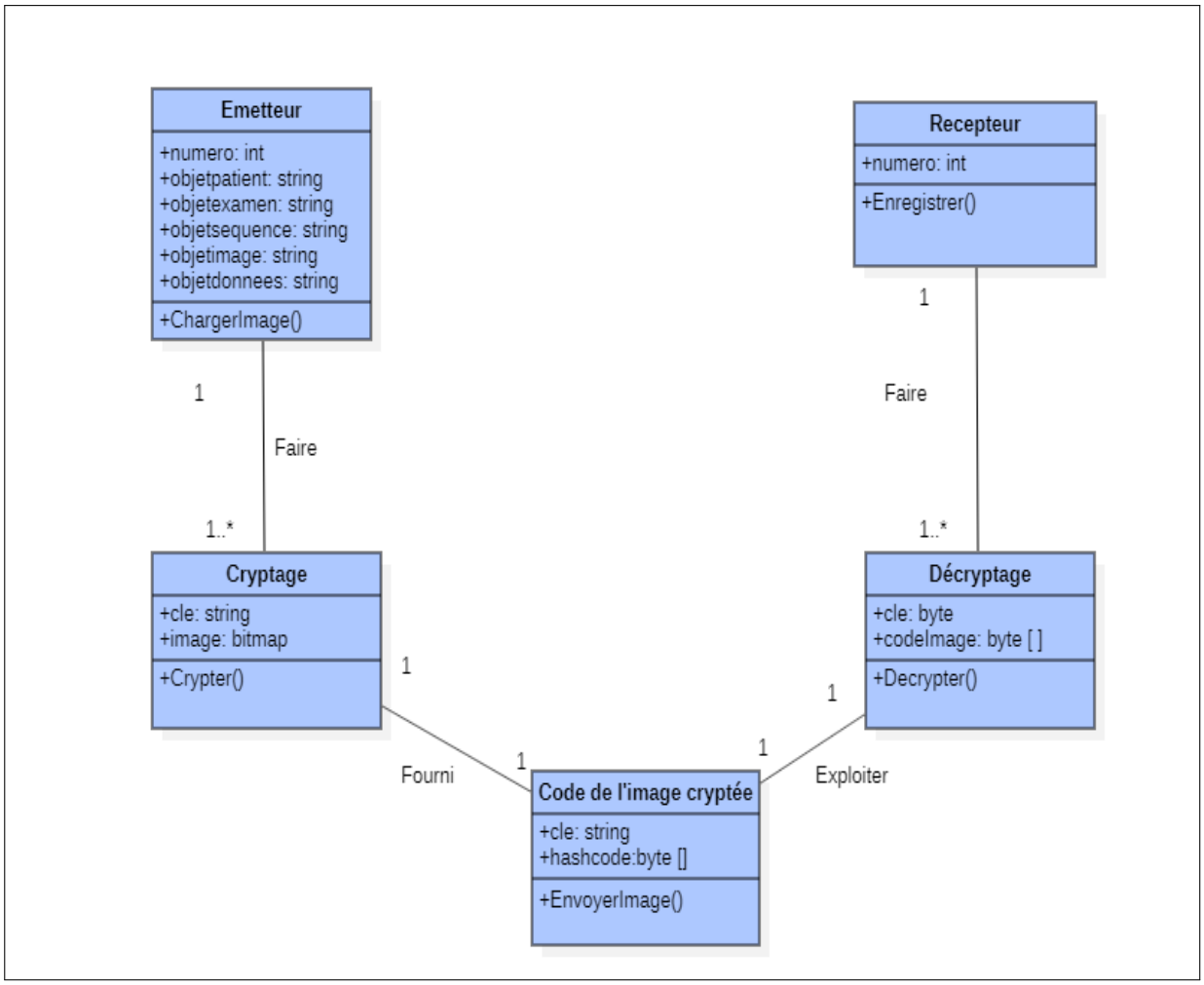


FIGURE 3.5 – Diagramme de classe

3.9 Conclusion

Dans ce chapitre, nous avons présenté les besoins fonctionnels et non fonctionnels de notre application, le langage de modélisation UML et le processus de développement que nous avons adopté. Par la suite, nous avons entamé la phase de spécification et d'analyse des besoins qui nous a permis d'identifier les acteurs de notre application et les différents cas d'utilisations, suivi par des descriptions textuelles. Enfin, nous avons présenté la phase de conception de notre application en introduisant le diagramme de séquence, le diagramme de classes et son dictionnaire de données. Dans le chapitre suivant, nous allons entamer les phases d'implémentation et de réalisation de notre système.

Chapitre 4

Implémentation

4.1 Introduction

Ce chapitre est consacré à la partie pratique de la mise en œuvre de notre application mobile assurant le transfert sécurisé des images médicales. Il comprend une description des outils de développement utilisés, tels que l'environnement de travail Android Studio avec le langage de programmation Java et le langage de description XML (Extensible Markup Languag) ainsi q'une description de l'algorithme proposé. Enfin, nous présenterons quelques interfaces de notre application.

4.2 Application mobile

Les applications mobiles sont des programmes téléchargeables gratuitement ou de manière payante. Elles sont exécutées depuis le système d'exploitation d'un smartphone ou d'une tablette. Les applications mobiles s'adaptent aux différents environnements techniques des smartphones et à leurs limites et possibilités ergonomiques (écrans tactiles notamment).

4.2.1 Application Android

Les applications Android sont des applications mobiles spécialement développées pour les smartphones utilisant le système d'application Android, acheté et développé par Google. Ce type d'application s'obtient sur GooglePlay.

4.2.2 Avantages des applications mobiles

- Contenus adaptés aux spécificités de chaque smartphone.
- Rapidité d'exécution (l'application mobile exploite au mieux les capacités du téléphone).
- Facilité d'installation et d'accès.
- Interface plus riche avec de meilleures performances.

- Pas besoin d'avoir accès à l'internet pour que l'application fonctionne.

4.3 Environnement de développement

Dans cette partie, nous allons citer l'environnement matériel (Hardware) et logiciel (Software) utilisés.

4.3.1 Environnement matériel

Notre application est réalisée sur un PC portable avec un smartphone pour l'exécution dont les caractéristiques sont résumées dans le Tableau 4.1 :

	Micro-ordinateur(Développement)	Smartphone(Exécution)
Marque	TOSHIBA	Samsung F12
Processeur	Intel(R) Core(TM) i5	4x2.0 GHz Cortex-A55 & 4x2.0 GHz Cortex-A55
Disque dur	500 Go	64 Go
RAM	4 GO	4 Go
Système	Windows 10	Android 11

TABLE 4.1 – l'environnement matériels utilisé

4.3.2 Environnement logiciels

Les logiciels utilisés pour la réalisation du projet sont les suivants :

Android studio

Android Studio est un environnement de développement utilisé pour développer des applications mobiles Android. Il permet principalement d'éditer les fichiers Java et les fichiers de configuration XML d'une application Android.

JDK

Le JDK (Java Development Kit) est un kit de développement logiciel permettant de développer des applications en Java. Il comprend l'environnement JRE (Java Runtime Environment), le compilateur Java et les API Java. Le JDK contient également plusieurs outils de développement (compilateurs, JavaDoc, Java Debugger, etc.).

SDK

Le SDK (Software Development Kit) est un ensemble d'outils fournis par le fabricant (généralement) d'une plate-forme matérielle, d'un langage de programmation. Le SDK Android permet

de développer des applications uniquement pour Android.

4.3.3 Langage de programmation

Afin de développer notre système, nous avons choisi les langages suivants :

JAVA

Est un langage de programmation populaire, utilisé à grande échelle dans le monde entier pour le développement d'applications. Il présente des avantages tels que l'extensibilité, la gestion de la mémoire, la haute sécurité, le support communautaire, etc.

XML

Le XML est un langage informatique qui sert essentiellement à stocker/transférer des données de type texte structurées en champs arborescents. Ce langage est qualifié d'extensible car il permet à l'utilisateur de définir des marqueurs (balises) qui facilitent le parcours au sein du fichier et donc la lecture de l'information.

4.4 Algorithme proposé

La Figure 4.1 montre l'algorithme de cryptage d'images médicales proposé. Tout d'abord, nous encodons l'image d'origine à l'aide de l'algorithme d'encodage en base 64 (cité en chapitre 2 ,page **27**), qui convertit l'image du format binaire en une chaîne (représentation ASCII). Nous chiffrons ensuite le texte résultant à l'aide de l'algorithme de chiffrement AES (cité en chapitre 2 ,page **21**) à l'aide d'une clé de 128 bits, 192 bits ou 256 bits. Enfin, nous obtenons le texte chiffré (image chiffrée).
Le décryptage d'image est le processus de cryptage inverse. Nous utilisons l'algorithme de cryptage AES pour décrypter le texte chiffré en utilisant la même clé que celle utilisée pour le cryptage afin d'obtenir le texte en clair, puis décodons le texte en clair via l'algorithme d'encodage base 64 pour obtenir l'image d'origine.

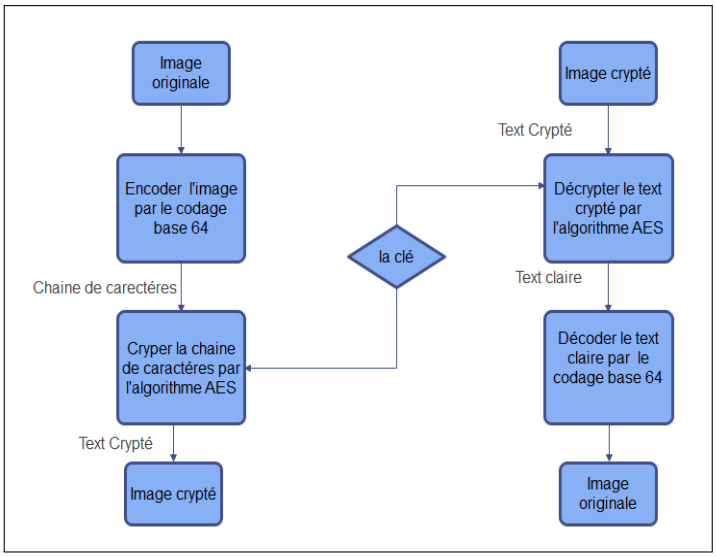


FIGURE 4.1 – Organigramme de l'algorithme proposé

Le choix de l'algorithme AES répond à un certain nombre de critères plus généraux, parmi lesquels on peut citer les suivants [23] :

- Simple.
- Excellent Sécurisé
- Implémentation rapide.
- Très faibles besoins en ressources.
- Le cryptage et le décryptage plus rapide que les autres algorithmes.

Pour le transfert d'images médicales, nous utiliserons l'application SMS, qui est un service populaire pour transférer et échanger des messages textes courts entre téléphones mobiles car il présente les avantages suivants, notamment :

- Le gain de temps.
- Une fois installée, l'application est simple et rapide .
- Pas besoin de s'inscrire, ce qui élimine la possibilité de perte de données.
- L 'application fonctionne en mode réduit même s'il n'y a pas de connexion Internet .

4.5 Quelques interfaces de l'application

La Figure 4.2 représente l'arborescence de notre application nommée " Transfert sécurisé des images médicales " :

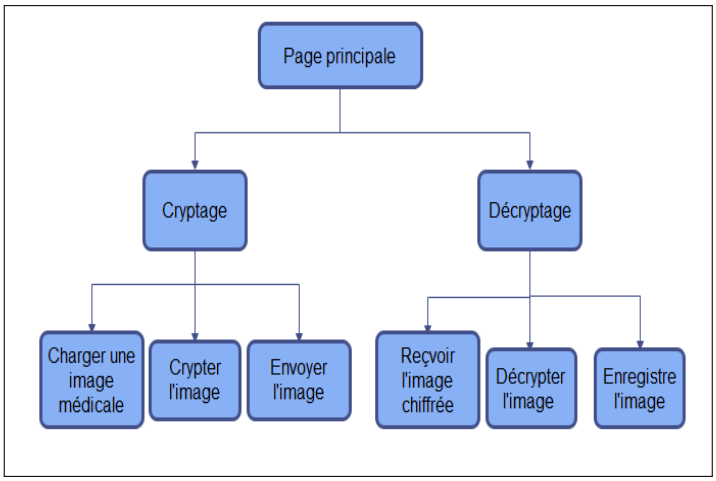


FIGURE 4.2 – Arborescence de l'application

La Figure 4.3 représente le logo de notre application. Au démarrage, le système affiche l'interface de notre application présentée dans la Figure 4.4 :

FIGURE 4.3 – Logo de l'application

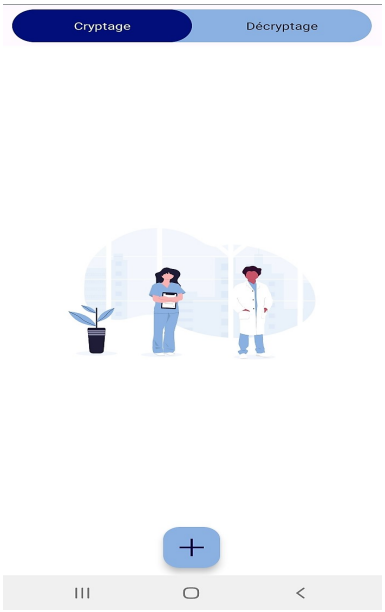


FIGURE 4.4 – Interface principale

Cette interface contient un TableLayout comprenant deux fragments. L'un pour le cryptage et l'envoie d'images, l'autre pour le décryptage et l'enregistrement d'images .

4.5.1 Cryptage d'images

Le cryptage des images par notre application se fait de la manière suivante :
Au début, l'utilisateur clique sur le bouton (plus) pour charger l'image, puis l'application affiche un message lui demandant d'autoriser l'accès à ses photos et à son contenu multimédia, comme

le montre la Figure 4.5. L'interface (Figure 4.6) montre un message au cas où l'utilisateur refuse la demande d'accès :

FIGURE 4.5 – Message d'une demande d'autorisation d'accès au stockage

FIGURE 4.6 – Message d'un refus de la demande d'autorisation d'accès au stockage

Si l'utilisateur autorise la demande d'accès, l'application lui demande de sélectionner le format d'image souhaité (Figure 4.7, l'image sélectionnée s'affichera sur l'écran (Figure 4.8).

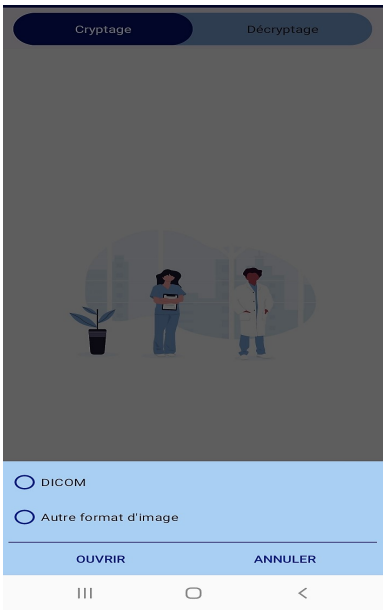


FIGURE 4.7 – Interface de sélection du format d'image

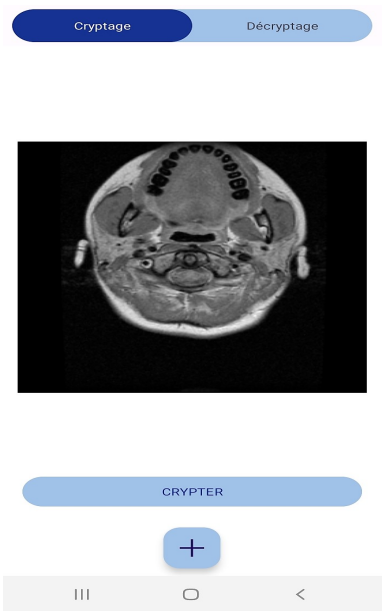


FIGURE 4.8 – Image sélectionnée

Après avoir sélectionné l'image, l'utilisateur appuie sur le bouton de cryptage pour crypter l'image, l'application crypte l'image selon l'algorithme proposé. Puisque nous avons utilisé l'algorithme de cryptage symétrique AES, l'application génère une clé de cryptage d'une manière aléatoire par l'utilisation d'une classe publique SecureRandom extends Random. Cette classe fournit un générateur de nombres aléatoires cryptographiquement fort. L'interface présenté dans la Figure 4.9 montre le résultat du chiffrement, qui est composé de deux chaînes, l'une est l'image et l'autre est la clé.

FIGURE 4.9 – Résultat de cryptage

Pour l'envoi d'images, nous avons utilisé l'algorithme d'envoi par **SMS** et la méthode **sendMultipartTextMessage** qui permet de scinder le message en plusieurs parties (sous-messages), puisque le SMS a une capacité d'environ 164 caractères, et que notre message est très long. Une fois que l'utilisateur appuie sur le bouton d'envoi(l'utilisateur envoie d'abord le code de hachage, puis la clé de sécurité après que le destinataire confirme la réception), l'application lui demandera la permission d'envoyer des SMSs (Figure 4.10), puis si l'utilisateur l'autorise, l'application affichera une interface (Figure4.11) (Figure4.11) lui demandant d'entrer le numéro de la personne avec laquelle vous voulez partager l'image, puis il suffit d'appuier sur OK pour envoyer l'image ou annuler l'envoie :

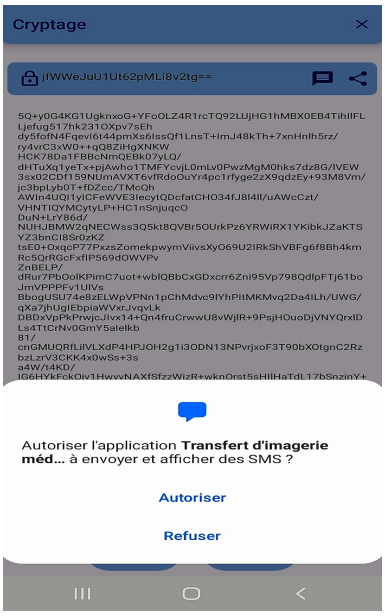


FIGURE 4.10 – Message d'une demande d'autorisation d'envoyer des SMSs

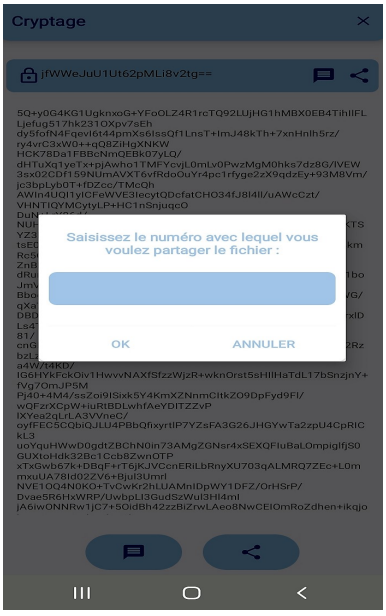


FIGURE 4.11 – Envoie le code d'images

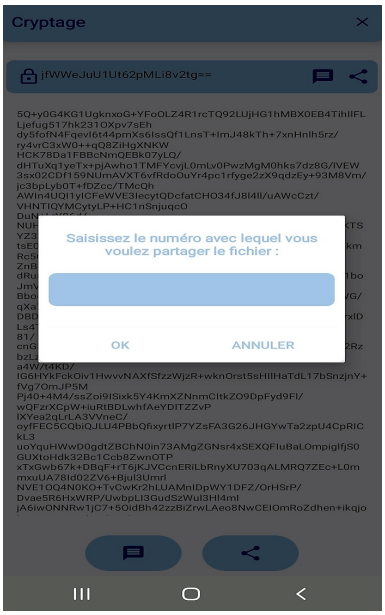


FIGURE 4.12 – Envoie la clé

4.5.2 Décryptage d'images

Après que l'émetteur envoie l'image chiffrée, le récepteur reçoit le hashcode (le code de l'image) par SMS (Figure 4.13, puis confirme la réception en envoyant un message de confirmation à l'émetteur, puis reçoit la clé de sécurité comme indiqué la Figure 4.14 :

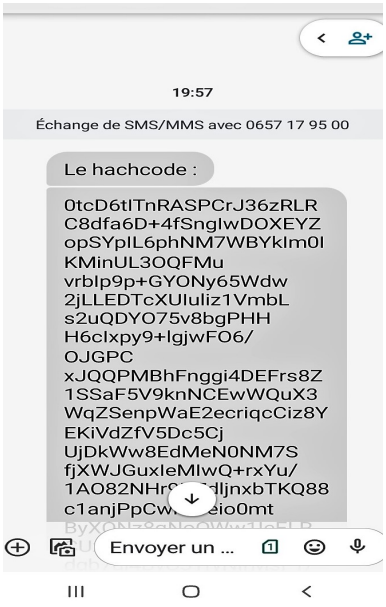


FIGURE 4.13 – Réception de code de l'image par SMS

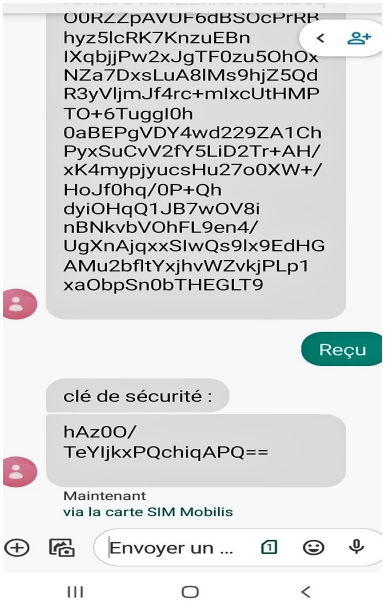


FIGURE 4.14 – Message de confirmation et la réception de la clé

L'utilisateur copie et colle les deux codes dans l'interface de décryptage d'images comme le montrent les Figures 4.15 et 4.16 :

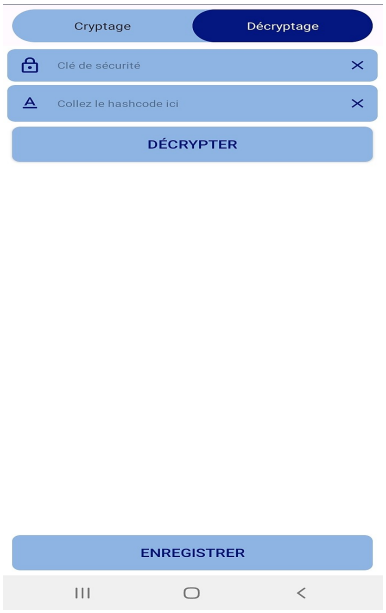


FIGURE 4.15 – Interface de décryptage d'images

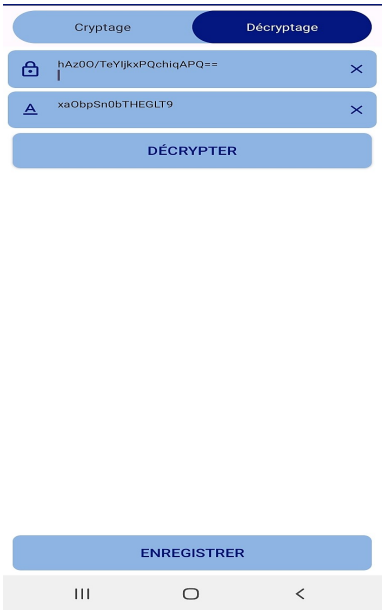


FIGURE 4.16 – Copier coller le code de la clé et l'image

Ensuite, l'utilisateur appuie sur le bouton de décryptage pour décrypter l'image. Si la clé de sécurité ou le code de hachage est incorrect ou incomplet, l'application affichera un message d'er-

reur comme indiqué dans l'image 4.17, sinon l'application affichera l'image cryptée à l'utilisateur, qui l'enregistrera ensuite. La Figure 4.18 montre le résultat du déchiffrement :

FIGURE 4.17 – Erreur de décryptage

FIGURE 4.18 – Résultat de décryptage

4.6 Conclusion

Dans ce chapitre, nous avons abordé les aspects pratiques liés à la réalisation de notre projet, à savoir les outils de développement nécessaires et les langages de programmation. Nous avons présenté par la suite l'organigramme de l'algorithme proposé. Enfin, nous avons exposé quelques captures d'écran expliquant les fonctions de base et les étapes pour envoyer et recevoir en toute sécurité des images médicales via notre application.

Conclusion générale et perspectives

Les travaux réalisés dans ce mémoire ont permis à la conception et la réalisation d'applications mobiles visant à sécuriser la transmission d'images médicales. Nous avons entamé le travail par une phase de description de notre système interactif. Par la suite nous avons présenté la conception détaillée à l'aide du langage UML en utilisant le processus UP, ainsi que les trois types de digrammes : de séquence, d'utilisateurs et de classe.

En effet, en raison du manque de sécurité, le partage d'images médicales est souvent une méthode indispensable pour les professionnels. Assurer la sécurité de ces données est aujourd'hui le devoir de tous professionnel de santé, incluant la confidentialité, la disponibilité et la fiabilité, exprimé en termes de sécurité des informations médicales.

Sachant que cette application peut être envoyée par SMS et ne nécessite pas l'utilisation d'Internet, elle est accessible à toutes les couches de la société et facile à utiliser en cas d'urgence.

Au terme de ce travail, quelques perspectives peuvent être envisagées. Il serait intéressant :

- De traiter le cas des images DICOM compressées.
- De rendre l'application plus facile à manipuler et le transfert plus rapide.
- Évaluer la similarit de l'image avant et après cryptage .

Bibliographie

[1] M. Bergounioux. *Introduction au traitement mathématique des images*. Springer-Verlag Berlin Heidelberg, 1ère édition, 2015.

[2] R.C. Gonzalez and R.E. Woods. *Digital image processing*. Pearson, 3éme édition, 2007.

[3] E. Bacquet. *Préparer des images numériques*. Eyrolles, 2009.

[4] L. Robichaud. *L'image numérique Pixels et couleurs*. Support de cours, Département d'histoire, Université de Sherbrooke, 2019.

[5] K.Hadjer. *Méthode de cryptage d'image basée sur la permutation et la matrice de Householder*. Mémoire de master, Université Kasdi-Merbah Ourgla, 2019.

[6] M.Benabdellah. *Outils de compression et de cryptocompression : Application aux images fixes et vedio* . Thèse de doctorat, Universite mohammed v-agdal, 2007.

[7] J.T. Bushberg, J.A.Seibert, E.M. Leidholdt Jr, J.M. Boone. *The Essential Physics of Medical Imaging*. Lippincott wiliams, 2émé édition, 2001.

[8] C.Delpas ,G.Frija ,B.Mazoyer. *L'imagerie médicale*. Texte rédigé pour le site web de la fondation pour la Recherche Médicale, 54 rue de Varenne - 75007 Paris, 2002.

[9] R. Dumont. *Cryptographie et Sécurité informatique*. Support de cours, Université de Liège, 2010.

[10] T. Mekhaznia. *Analyse cryptographique par les méthodes heuristiques*. Thèse doctorat, Université de Batna, 2017.

[11] Z. Farah. *Cours sécurité, 1 ère année Master*. Université de béjaia, 2021.

[12] A.J. Menezes, P.C. Van Oorschot and S.A. Vanstone. *Handbook of applied cryptography*. Mit Press, 1996.

[13] B.Rabab. *Sécurité des images Numériques compressées JPEG*. Thèse de doctorat, Université Djilal Liebes Sidi Bel Abbes, 2019.

[14] P. Roques. *UML 2 par la pratique : Etudes de cas et exercices corriges*. Eyrolles, 5ème édition, 2006.

[15] J.P. Aumasson, *Serious Cryptography : A Practical Introduction to Modern Encryption*, No Starch Press, 4ème édition, 2017.

[16] L. Bryant and J. Ward. *Caesar ciphers : An introduction to cryptography*. Support de cours, Université de Portugal, 2007.

[17] B.A. Forouzan. *Cryptography Network Security*. McGraw-Hill. 1ère édition, 2007.

[18] S. William. *Cryptography and Network Security*. Pearson Education India, 4ème édition, 2006.

[19] Fips Pub *Data Encryption Standard (DES)*. National Institute of Standards and Technology, 46-3, 1999.

[20] Fips Pub. *Advanced Encryption Standard (AES)*. ,National Institute of Standards and Technology, 197, 2001.

[21] R.L. Rivest and A.Shamir and L.Adleman. *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2) :120–126, 1978.

[22] T. ElGamal. *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*. IEEE Transactions on Information Theory, 31 :469–472, 1985.

[23] M. Prerna, A. Sachdeva and P. Mahajan. *A Study of Encryption Algorithms AES, DES and RSA for Security*. Int. Res. J. Publ. Glob. Journals Inc, vol.13, 2013.

[24] U. Sara and M. Akter and M. S. Uddin. *Évaluation de la qualité d'image par FSIM, SSIM, MSE et PSNR - une étude comparative*. Journal of Computer and Communications, Vol.07, 2019.

[25] A. Rehman and Z. Wang and M. Rostami. *SSIM-inspired image restoration using sparse representation*. Article dans la revue EURASIP sur les progrès du traitement du signal, vol.16, 2012.

[26] https ://interstices.info/tout-ce-que-les-algorithmes-de-traitement-dimages-font-pour-nous/. *Tout ce que les algorithmes de traitement d'images font pour nous*. Consulté le 20/04/2022.

[27] https ://sites.google.com/site/maliinformaticaalameda/home/1a-evaluacion.*L'image numérique :Composition et caractéristiques*. Consulté le 20/04/2022.

[28] https ://www.baches-publicitaires.com/blog/actualites/vectorisation-cest/. *Quel est la différence entre une image matricielle ou vectorielle ?*. Consulté le 21/04/2022.

[29] https ://www.webmarketing-com.com/2012/11/06/16580-quels-sont-les-9-formats-differents-pour-une-image. *Quels sont les 9 formats differents pour une image ?*. Consulté le 21/04/2022.

[30] https ://www.imaios.com/fr/Societe/blog/. *Les-5-meilleurs-DICOM-Viewer*. Consulté le 28/04/2022.

[31] https ://www.researchgate.net/. *Representation shematique dun fichier DICOM et des differents objets DICOM quil peut contenir*. Consulté le 01/05/2022

[32] https ://sabricole.developpez.com/uml/tutoriel/unifiedProcess/. *UP : Unifed process*. Consulté le 25/05/2022.

Résumé Au cours de ce projet nous avons conçu et mis en place une application mobile dédiée à la sécurité du transferts des images médicales. Cette application facilite la communication avec les professionnels de la santé et les parties intéressées. Notre objectif est d'aider à protéger le contenu de l'image médicale avec le cryptage symétrique AES et le cryptage Base 64 et d'assurer l'envoi des images cryptes par SMS qui ne nécessitent pas Internet et sont donc disponibles pour tous les intéressés avec une simple manipulation.
Ce travail a été réalisé en utilisant le processus de développement Processus Unifie (UP) et le langage de modélisation unifié (UML) pour la planification de la solution. Et nous avons choisi de programmer l'applications en Java (Android Studio).

Mots-clés– Cryptographie, Image médicale, AES, Base 64, MLobilité, java, Uml.

Abstract During this project we designed and implemented a mobile application dedicated to the security of medical image transfers. This application facilitates communication with healthcare professionals and interested parties. Our goal is to help protect the content of the medical image with symmetric AES encryption and Base 64 encryption and to ensure the sending of encrypted images via SMS that do not require the Internet and are therefore available to all interested parties with a simple manipulation.
This work was carried out using the Unified Process (UP) development process and the Unified Modeling Language (UML) for planning the solution. And we chose to program the application in Java (Android Studio).

Key-words– Cryptography, Medical Image, AES, Base 64, Mobility, Java , Uml.