République Algérienne Démocratique et Populaire

الجمهورية الجزائرية الديمقراطية الشعبية

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Chadli Bendjedid
Faculté des Sciences et de la Technologie
Département d'Informatique

وزارة التعليم العالي والبحث العلمي
جامعة الشاذلي بن جديد
كلية العلوم و التكنولوجيا
قسم الاعلام الالي

جامعة الشاذلي بن جديد
UNIVERSITE CHADLI BENDJEDID

MEMOIRE

Présenté par

HannaniLouiza

Pour l'obtention de diplôme de

MASTER

Filière : Informatique

Spécialité : Systèmes Informatiques Intelligents

Thème

Réalisationd'une application mobile de stéganographie

Soutenu le : 29 / 06 / 2022 Devant le Jury composé de :

Qualité	Nom et Prénom	Grade	Université
Président	Mme. FeroumAssia	MCB	Chadli Bendjedid El-Tarf
Rapporteur	**Mme. Bougarne Imane**	**M**	
Examinateur	Mme. Ahmed Malek Nada	MAA	Chadli Bendjedid El-Tarf

Année Universitaire : 2021/2022

REMERCIMENT

Je remercie en premier bien le dieu qui par sa grâce a permis la réalisation de ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'avais pas pu voir le jour sans l'aide de Mme Bougarne Imane, je le remercie pour la qualité de son encadrement durant notre préparation de ce mémoire,

Je remercie tous l'ensemble des enseignants du département informatique pour ces directives et ses précieux conseils.

Tous les membres de jury à :

- *Mme Feroum maitre de conférences qui nous a fait l'honneur de présenter le jury de cette soutenance.*
- *Mme Ahmed Maled mettre assistante qui en l'amabilité d'examiner ce mémoire.*

Mes vifs remercîments s'adressent également à tous les personnels administratifs (Rachida, Nadia, Imane …) du département informatique.

Dédicace

Je dédie ce modeste travail

A celle qui a attendu avec patience les fruits de sa bonne éducation et de ses dévouements ;

A ma chère mère.

A celui qui ma aider et encourager pour m’assurer les bonnes conditions ;

A mon mari

A ma petite famille qui m’a toujours soutenue

A mes enfants

A tous mes collègues et amis.

SOMMAIRE

LISTE DES FIGURES

LISTE DES TABLEAUX

Liste des acronymes

JPEG (Joint Photographic Experts Group)

BMP : Bit Map

DCT : Discrète CosineTransform

DWT : Discrète WaveletTransform

GIF : Graphics Inter change Formats

LSB: Least Significant Bit

PNG: Portable Network Graphic

RGB : Red, Green, Blue

Wifi : Wireless fidelity

Introduction générale

La technologie ne cesse d'évoluer considérablement ces dernières années et semble ne pas vouloir s'arrêter, ce qui conduit à la banalisation de celle-ci. En effet, la plupart des gens actuellement y ont accès : par exemple, une grande majorité peut s'offrir un téléphone portable doté de capacité de traitement de plus en plus performante, et cela à un prix plus ou moins abordable, contrairement à auparavant.

En outre, ces progrès en termes de technologies apportent un nouveau concept : l'importance de l'information, l'oxygène des temps modernes. L'information peut être une source de richesses et de pouvoirs pour une entreprise, ultraconfidentielle dans le domaine de la communication militaire, vitale ou tout simplement privée pour certaines personnes.

À la lumière de ce constat, donner une protection adéquate à l'information devient un enjeu primordial pour tous les acteurs économiques (entreprises, gouvernements, mais aussi les utilisateurs/consommateurs).

La cryptographie, une première solution, consiste à chiffrer l'information, c'est-à-dire, à appliquer une certaine transformation à celle-ci, de façon à la rendre illisible pour les personnes non concernées. Toutefois, un problème se pose : l'information chiffrée attire trop l'attention. En effet, certes les personnes malintentionnées ne savent pas encore sur le moment comment la déchiffrer, cependant, elles savent que c'est une information chiffrée suspecte, et donc importante. Ces personnes peuvent ensuite essayer de la décrypter en utilisant la cryptanalyse.

C'est là qu'intervient la stéganographie, un tout autre art, une autre science souvent confondue avec la cryptographie. La stéganographie arrive en renforcement au chiffrement de données. Quelle meilleure protection y a-t-il que de faire passer une information sans que les personnes susceptibles de l'intercepter aient même conscience de son existence ?

Ceci étant, le but de ce travail est donc axé sur la conception et le développement d'une application mobile de stéganographie, une technique qui permet de répondre à notre besoin, c'està-dire de protéger l'information d'une certaine façon en la dissimulant dans un autre support anodin, d'où le titre de ce mémoire : « Développement d'application mobile de stéganographie ».

Le contenu de ce mémoire est divisé en trois parties :

- La première partie, consiste à présenter l'art de la stéganographie, à décrire son principe afin de mieux cerner son intérêt, et les techniques de stéganographie seront utilisées par l'application ; et on parle plutôt d'une comparaison des meilleurs systèmes d'exploitation mobiles actuels, et explique le choix de la plateforme Android comme plateforme de développement.
- La deuxième partie relate les phases de conception et de développement de l'application, ainsi que de son implémentation.
- La dernière partie est réservée aux différents tests et évaluations de la performance de l'application.Et enfin, une conclusion et des perspectives possibles pour notre travail sont énoncées.

CHAPITRE 1 : L’ART DE LA STEGANOGRAPHIE

1.1Introduction

Dans ce chapitre, nous présentons tout d'abord quelques définitions de la stéganographie, un petit historique des techniques utilisées pour bien cerner la philosophie du domaine, et les concepts d'emploi. Nous posons ensuite les bases de la stéganographie moderne et mettons en évidence les propriétés intrinsèques des schémas de la stéganographie. Une notion de stéganalyse y sera présentée. Nous en déduirons ainsi les services de sécurité offerts par de telles techniques, ainsi que les règles fondamentales de leur mise en œuvre. En outre, nous verrons brièvement une technique dérivée de la stéganographie appelée tatouage numérique, mais pourtant qui diffère de celle-ci par l'utilisation et le but recherché.

1.2Définitions de la stéganographie

La stéganographie est l'art de la dissimulation de communications. Contrairement à la cryptographie, la stéganographie n'a pas pour objectif de sécuriser une communication, ***mais d'en cacher l'existence même***. Les deux disciplines ont donc chacune leurs propres domaines de compétence. Dans certaines situations, le fait même de vouloir transmettre des données de manière chiffrées sera jugé comme ***suspect*** [1].

La stéganographie provient étymologiquement de la combinaison des mots grecs ***Stéganô***, signifiant Je ***couvre***, et ***Graphô***, signifiant J'***écris***, soit littéralement traduisible par ***Je couvre ce que j'écris***. Ainsi l'objectif premier de la stéganographie est de dissimuler une information, d'encacher son existence, là où la cryptographie cherche à rendre illisible cette information.

Pour bien comprendre ce concept, comparons-la avec la cryptographie plus connue. Dans le cas de la cryptographie, un message est chiffré avec une clé de chiffrement qui reste secrète, et garantit le secret du message. Une fois chiffré le message est illisible, ce message chiffré peut circuler librement sans risquer d'être dévoilé, car son intégrité est liée au secret de la clé de chiffrement uniquement connue par les personnes qui sont censées accéder au contenu du message. Dans ce cas de figure les personnes qui voient le message chiffré savent qu'il contient une information, mais ne peuvent la lire. Concernant la stéganographie, c'est l'existence de ce message qui est cachée. C'est-à-dire que le secret du message est préservé par le fait que personne, hormis les personnes auxquelles le message est destiné, ne doit savoir qu'un message est transmis.

Si dans le cas de la cryptographie, la sécurité de l'information est garantie par la clé de chiffrement, dans le cas de la stéganographie ***la sécurité de l'information est garantie par le secret de la manière dont est dissimulé le message***.

En stéganographie on cherche à cacher un message secret (« secret message » en anglais) au sein ***d'un objet de couverture(coverobject)***. La compilation du message secret et de l'objet de couverture donnera un ***objet stégo*** ou encore ***stégo-objet(stegoobject)***, qui devra être d'apparence similaire à l'objet de couverture, et contenir le message secret afin de pouvoir le transmettre sans que personne hormis les personnes pour lesquelles le message secret est destiné, ne suspecte l'existence de ce message secret au sein de l'objet de couverture [2].

1.3Historique

1.3.1Une technique antique

La première forme de stéganographie répertoriée nous vient d'une histoire Grecque signée Hérodote et datant du 5ème siècle avant Jésus-Christ. L'auteur nous relate la révolte contre les lois Perses. Les Grecs utilisaient certains esclaves pour transmettre les messages. Ceux-ci étaient écrits sur les crânes des messagers, et passaient donc inaperçus lorsque les cheveux repoussaient. Une fois ses cheveux suffisamment longs, le messager pouvait être envoyé, avec l'ordre de se faire raser le crâne, une fois arrivé à destination. Le principal désavantage de cette méthode était l'attente pour l'envoi d'un message. Une autre technique était d'utiliser des tablettes de cire. Une fois la cire raclée, on gravait le message dans le bois de la tablette. Il suffisait ensuite d'y remettre de la cire, et le message était parfaitement caché.

Figure 1.Tablette de cire antique [1]

En Chine ancienne, les messages étaient écrits sur de la soie, qui était ensuite roulée en boule, ellemême recouverte de cire. Un messager devait enfin avaler cette boule.

Dès le Ier siècle avant Jésus-Christ, les romains utilisaient l'encre invisible, qui fut la plus utilisée des méthodes de stéganographie à travers les siècles. On écrit, au milieu des textes écrits à l'encre, un message à l'aide de jus de citron, de lait ou de certains produits chimiques. Il est invisible à l'œil, mais une simple flamme, ou un bain dans un réactif chimique, révèle le message [1].

1.3.2Du moyen-âge à l'avant-guerre

Au cours du moyen-âge et de la renaissance se développèrent des techniques de stéganographie linguistiques dont la plus connue est l'acrostiche, qui repose sur l'écriture d'un message caché qui sera lu, en lisant la première lettre ou le premier mot d'une ligne de haut en bas et non de gauche à droite. Parmi les plus célèbres acrostiches, on peut citer les correspondances d'Alfred De Musset et Georges Sand. Lewis Carroll en est aussi friand dans ses livres et plus particulièrement dans le livre *A travers le miroir* où il révèle sous forme d'acrostiche le nom de son héroïne Alice [2].

D'autres techniques consistent à cacher un message en jouant sur la ponctuation du texte, l'espacement entre les mots, ou des erreurs volontaires sur le style de l'écriture (typographies, taille de polices de caractères, etc…). On voit aussi apparaître au cours de cette période des techniques utilisant la musique pour envoyer des messages de manière stéganographique. Un scientifique allemand, Gaspart Schott (1608-1666) explique dans son livre « *Schola Steganographica* » comment dissimuler des messages en utilisant des notes de musique. Les notes des partitions de musique correspondent alors à des lettres qui forment le message caché, pouvant ainsi être transmis en récupérant la partition ; ou bien en écoutant les notes jouées par un musicien comme expliqué par John Wilkins en 1694, qui montre par ailleurs comment transmettre un message avec l'utilisation de figures géométriques, chaque figure (ligne, rectangle, carré, etc…) dans sa position correspondant à une lettre de l'alphabet [1].

1.3.3Des 2 guerres mondiales à aujourd'hui

La période correspondant aux deux guerres mondiales et la guerre froide est en quelque sorte l'apogée de l'utilisation de la stéganographie. Lors de la première guerre, les espions dissimulaient des microfilms contenant des images réduites par réductions successives sous leurs ongles, dans les oreilles, etc….

Les techniques d'encres invisibles refont aussi leurs apparitions de manière améliorée, avec des procédés chimiques qui nécessitent un premier produit chimique pour cacher un message, et un deuxième différent pour le révéler.

Les services secrets se sont aussi servis de la technique des micros points, qui consiste à réduire un message ou une image à une taille microscopique, et à l'insérer dans un signe de ponctuation d'un texte contenu par exemple dans un magazine ou un journal.

Ces dix dernières années ont vu un regain d'intérêt pour l'utilisation de la stéganographie, principalement suite à l'attaque des deux tours du World Trade Center du 11 Septembre 2001.

Les services secrets américains soupçonnent vigoureusement Oussama Ben Laden et ses hommes, d'avoir échangé des informations via l'utilisation d'images pornographiques sur Internet cachant des messages.

Le succès de romans tels que le *Da Vinci Code* de Dan Brown ou la série américaine *Prison Break* n'est pas non plus étranger à cette attention nouvelle.

Par ailleurs, de la stéganographie est née une variante, à savoir le tatouage numérique, qui consiste à inscrire des données sur des supports numériques pour en certifier l'auteur, technique qui intéresse au plus haut point les majors du disque et du cinéma qui cherchent à l'utiliser pour limiter le piratage numérique.

1.4Alice, Bob et Wendy

Pour expliquer le principe et l'utilité de la stéganographie appelée aussi communication invisible, Simmons [3] a été le premier à proposer le problème des prisonniers avec pour acteurs de ce problème : Alice, Bob et Wendy.

Alice et Bob sont deux personnes arrêtées et placées en prison dans deux cellules de prison différentes. Ces deux prisonniers cherchent à s'évader et pour cela elles doivent planifier leur action en s'échangeant des informations (heure et date de la tentative de l'évasion par exemple).

Malheureusement leurs seules communications possibles doivent se faire par l'intermédiaire de Wendy le gardien de prison. Ainsi ils ne peuvent s'échanger des lettres, où sont par exemple écrites clairement des informations qui doivent rester secrètes. Wendy a la possibilité de lire ses lettres et donc de ne pas les transmettre s'il pense détecter une information compromettante. De la même manière si Alice et Bob tentent de crypter les informations pour parer à ce problème, Wendy considérera comme suspect le message de par son caractère illisible et ne le transmettra pas.

Alice et Bob doivent alors trouver un subterfuge pour que Wendy transmette le message sans y détecter la présence d'une information sur leur future évasion. Pour cela ils doivent créer ce que Simmons appelle un *canal subliminal*, autrement dit un canal secret de communication dont seuls Alice et Bob connaissent l'existence, malgré le fait que Wendy fasse partie intégrante de la transmission de ce message. La figure 1.02 représente la configuration de la transmission de message stéganographique entre Alice, Bob et Wendy [2].

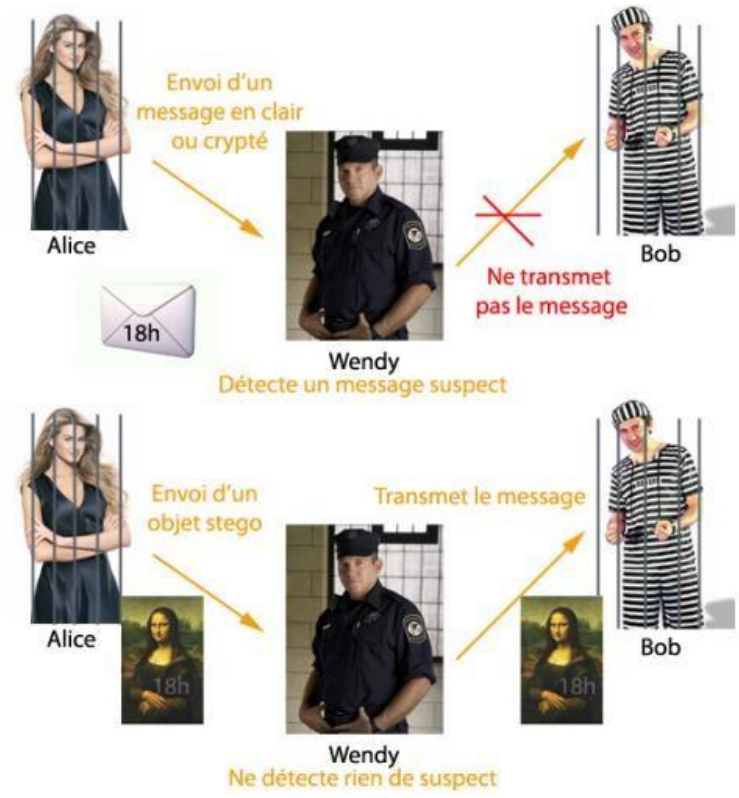


Figure 2 : Le problème des prisonniers [2]

Pour arriver à créer ce canal subliminal, Alice peut par exemple colorier une image d'une certaine façon, pour que les couleurs correspondent à un code compréhensible par Bob, et qui décrit la date et l'heure de l'évasion. Ainsi si Alice envoie cette image par l'intermédiaire de Wendy, Wendy pensera seulement transmettre une image à Bob, et en aucun cas une information. A la réception de cette image, Bob connaissant la méthode de lecture du message secret, sera alors capable de

connaître les informations envoyées par Alice. Ils pourront ensuite continuer à se transmettre des images coloriées pour créer ce canal subliminal.

Bien entendu cette approche nécessite que le moyen de communication, à savoir la technique de stéganographie employée pour créer le canal subliminal, ait été défini avant l'emprisonnement d'Alice et Bob [2].

Le modèle des prisonniers peut facilement être transposé à toute communication entre deux dispositifs qui cherchent à communiquer de manière invisible, et où Wendy serait un attaquant qui écoute le réseau.

Par ailleurs l'attaquant en stéganographie peut être de trois types :

- **Passif :** On parlera d'un attaquant passif quand l'attaquant ne cherche qu'à détecter une information circulant entre Alice et Bob. Il ne cherche pas forcément à découvrir le message secret mais seulement à découvrir son existence.
- **Actif :** un attaquant actif est un attaquant qui modifie l'objet de couverture, et peut altérer dans certains cas le message secret.
- **Malicieux :** un attaquant malicieux va modifier le message secret contenu dans l'objet de couverture, il peut par exemple essayer de se faire passer pour un des deux prisonniers.

1.5Architectures stéganographiques

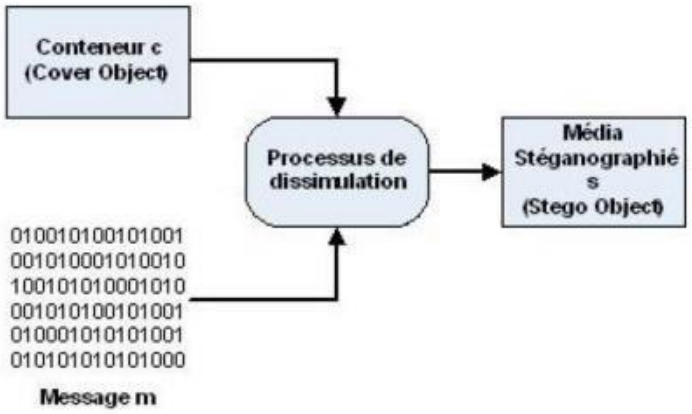


Figure 3 ; Schéma simplifié de stéganographie [1]

Dans une architecture stéganographique, il y a principalement deux éléments. D'un côté ***un processus de dissimulation***, de l'autre ***un processus de recouvrement***. Un processus de dissimulation simplifié peut être donné par le schéma de la figure 3 [1].

Il existe trois types d'architecture de stéganographie, correspondant de près à ce qui existe en cryptographie.

1.5.1Stéganographie pure

La stéganographie pure correspond à une utilisation de la stéganographie, où le secret du message ne correspond qu'à la méthode de dissimulation de ce message et à la méthode de récupération de ce message secret.

Le quadruplet $G = \{ C, M, D, E \}$, où C est l'ensemble des possibilités de couverture, M l'ensemble des messages secrets avec $| C | \geq | M |$, $E : C \times M \rightarrow C$ la fonction d'insertion et

$D : C \rightarrow M$ la fonction d'extraction, avec la propriété que $D(E(c, m)) = m$ pour tout $m \in M$ et $c \in C$ est appelé : ***système de stéganographie pure***.

Pour cette méthode, deux correspondants n'ont besoin d'échanger que la méthode de dissimulation et de récupération du message. Cela sous-entend que si un attaquant connaît ces méthodes, il n'y a plus de secret et il peut lire directement le message secret. Selon la méthode de stéganographie utilisée, l'attaquant peut trouver facilement la méthode pour découvrir le message. C'est donc le niveau de sécurité le plus faible pour l'utilisation de la stéganographie [3].

1.5.2Stéganographie à clé secrète

La stéganographie à clé secrète correspond à la combinaison de l'utilisation d'une méthode de stéganographie pour dissimuler le message, et au cryptage préalable de ce message avant son incrustation dans l'objet de couverture.

Le quintuplé $G = \{ C, M, K, D_K , E_K \}$, où C est l'ensemble des possibilités de couverture, M l'ensemble des messages secrets avec $| C | \geq | M |$, K l'ensemble des clés secrètes, $E_K : C \times$

$M \times K \rightarrow M$ la fonction d'insertion et $D_K : C \times K \rightarrow M$ la fonction d'extraction, avec la propriété que $D_K (E_K(c, m, k), k) = m$ pour tout $m \in M$, $c \in C$ et $k \in K$ est appelé : ***système de stéganographie à clé secrète***.

Pour mettre en place cette technique, il est nécessaire que deux correspondants connaissent d'une part la méthode de dissimulation et de récupération du message secret, et d'autre part qu'ils partagent une clé commune de chiffrement. Ainsi un attaquant qui trouverait la méthode stéganographique pour dissimuler l'information, ne pourrait pas pour autant lire le message secret s'il ne connait pas la clé de chiffrement [2].

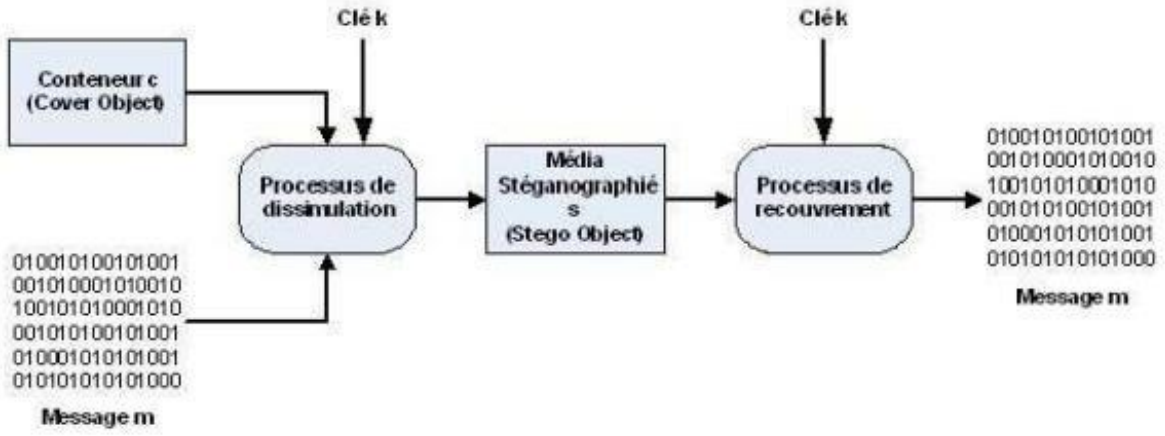


Figure 4Schéma de stéganographie à clé secrète [2]

La sécurité est donc globalement accrue par rapport à l'utilisation d'une technique de stéganographie pure. Cependant l'utilisation de clé symétrique ne permet pas de garantir l'authenticité de l'envoyeur du message, et peut permettre à un attaquant malicieux connaissant la clé de chiffrement de se faire passer pour l'expéditeur du message.

1.5.3 Stéganographie à clé publique

La stéganographie à clé publique reprend la définition de la stéganographie à clé symétrique hormis le fait que le chiffrement ne se fait plus à l'aide d'une clé partagée, mais grâce à l'utilisation de clés, publique et privée.

Ainsi si Alice souhaite envoyer un message secrètement à Bob, elle va préalablement chiffrer ce message avec la clé publique de Bob, et éventuellement le signer avec sa clé privée pour en assurer son authenticité. Puis elle le dissimulera avec une technique de stéganographie dans un objet de couverture, qu'elle fera transmettre à Bob. Celui-ci n'aura plus qu'à récupérer le message chiffré, à le déchiffrer, avec sa clé privée et éventuellement à vérifier la signature d'Alice avec la clé publique d'Alice [4].

La stéganographie à clé publique est la technique de stéganographie qui garantit la sécurité maximale. Si un attaquant découvre la méthode de stéganographie utilisée, il ne pourra pas déchiffrer le message car il ne connait pas la clé privée du destinataire du message. De plus l'utilisation de la signature par clé publique ajoute l'authenticité du message. Cependant il peut tout de même se poser le problème d'une attaque de type *man in the middle*, si la cryptographie à clé publique n'est pas couplée avec un tiers de certification des clés publiques [6].

1.6Caractéristiques de Schéma Stéganographique

Trois critères permettent de classer les algorithmes stéganographiques : La capacité, l'invisibilité et la robustesse [7].

- ***La capacité*** correspond à la taille de données pouvant être incorporées dans l'objet de couverture, relativement à la taille de celui-ci,
- ***L'invisibilité*** ou ***la transparence*** ou encore ***l'imperceptibilité*** qui dépend directement de la distorsion introduite par le processus de dissimulation pendant l'insertion de données ; la distorsion étant tout simplement le nombre de modifications ou de changements dans l'objet de couverture,
- ***La robustesse*** signifie la résistance de notre stégo-objet, c'est à dire rester normale même s'il subit des transformations (filtrage, etc....).

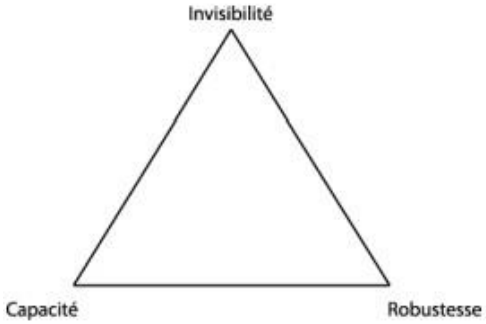


Figure 5 :Compromis entre capacité, invisibilité et robustesse[5]

Ces trois critères ne peuvent pas être maximisés simultanément. Chacun d'entre eux aura une influence sur l'autre. Par exemple, la capacité va en contradiction avec la transparence.

1.7La Stéganalyse

1.7.1Définition

La steganalyse est l'ensemble des techniques qui permettent de détecter si un objet est un objet stégo ou non, c'est-à-dire s'il contient un message secret ou non, puis dans la mesure du possible de décoder ce message secret.

Pour contrer cette stéganalyse, un algorithme de stéganographie doit trouver un compromis entre l'invisibilité, c'est-à-dire la probabilité qu'un objet de couverture soit déclaré non stégo, la capacité, soit la quantité d'informations dissimulées dans un objet de couverture, et la robustesse, c'est-àdire qu'un message secret reste intègre malgré des modifications de l'objet de couverture.

Dans le cadre de la stéganographie, ***l'invisibilité et la capacitésont les deux principales caractéristiques nécessaires***. La robustesse est quant à elle importante dans l'utilisation d'algorithmes de tatouage que nous décrirons plus tard[8].

1.7.2Sécurité parfaite

La sécurité parfaite d'un système de stéganographie correspond à un système qui garantit l'invisibilité des messages secrets. Cette invisibilité dépend de la capacité. Ainsi plus la taille des messages secrets augmente par rapport à la taille des objets de couverture, plus grands sont les risques d'être détectés.

La définition formelle de la sécurité d'un système stéganographique est donnée par Cachin [9]. L'idée principale de sa définition est la sélection d'un objet de couverture en fonction d'une variable aléatoire C, avec une probabilité de distribution P_C. La dissimulation d'un message secret correspond à une fonction définie dans C.

Soit P_S la probabilité de distribution de E_K (c, m, k), qui est l'ensemble des stégo-objets produits par le système stéganographique. Si un objet de couverture c n'est jamais utilisé comme stégoobjet, alors $P_S = 0$. Voici la définition de l'entropie relative $D(P_1 \| P_2)$ entre deux distributions P_1 et

P_2définies sur l'ensemble $\mathbb{Q}$, qui détermine l'inefficience de la distribution P_2 sur la distribution P_1, où P_1 représente une probabilité de distribution aléatoire [7].

P_C correspond alors à la distribution aléatoire P_1 et P_2 représente la probabilité P_S de distribution réelle du système de stéganographie. On peut ainsi représenter la sécurité d'un système de stéganographie en termes d'entropie relative $D(P_C \parallel P_S)$.

Sécurité parfaite pour un système de stéganographie : Soit G un système de stéganographie, P_S la probabilité de distribution de stégo-covers (médium de couverture) envoyés sur le canal de communication, et P_C la probabilité de distribution de C. G est dit ***-securisé*** contre les attaquants passifs, si $D(P_C \parallel P_S) \leq$, et parfaitement sécurisé

Puisque $D(P_C \parallel P_S)$ est nulle, cela signifie que l'entropie relative est égale à zéro, et donc que les deux probabilités de distribution sont égales, ce qui garantit que le système de stéganographie est théoriquement parfaitement sécurisé [8].

1.7.3Détection des stégo-objets

Dans le cas de la détection des messages, Wendy doit décider si un objet de couverture c envoyé entre Alice et Bob contient un message secret ou non. Pour cela Wendy définit une fonction f : C $\rightarrow$ 0, 1

$$f(c) = \begin{cases} 1 & c \text{ contient une secret} \\ 0 & \text{sinon} \end{cases} \qquad (1.02)$$

Dans certains cas, Wendy va correctement détecter un stégo-objet ; dans d'autres cas, il ne va pas détecter un stégo-objet, on parle ***d'erreur de type II***. S'il détecte un objet non stégo comme un objet-stégo, on parle ***d'erreur de type I***.

En pratique, les systèmes de stéganographie cherchent à maximiser la probabilité β qu'un attaquant passif fasse des erreurs de type II, le système idéal étant pour β = 1 [9].

1.7.4Techniques de stéganalyse

Les techniques de stéganalyse cherchent à minimiser la probabilité β qu'une attaque sur un système de stéganographie fasse des erreurs de type II. Ce qui correspond en fin de compte à maximiser la probabilité de détection de tous les stégo-objets.

Afin de mettre à jour l'utilisation de la stéganographie, les attaques possibles par un stéganalyste sur un système de stéganographie sont caractérisées en 6 types d'attaques[10]:

- ***attaque sur objet stégo uniquement :*** seul l'objet stégo est connu par le stéganalyste.
- ***attaque sur objet de couverture connu :*** l'objet de couverture original (sans insertion de messages secret) est connu ainsi que l'objet stégo.
- ***attaque sur message connu :*** le stéganalyste connait le message
- ***attaque sur stéganographie choisie :*** l'algorithme de stéganographie ainsi que l'objet stégo sont connus.
- ***attaque sur messages secrets choisis :*** le stéganalyste est capable de générer des objets stégos avec des messages secrets choisis.

- ***attaque sur stéganographie connue :*** l'algorithme de stéganographie est connu, et le stéganalyste dispose de l'objet de couverture initial ainsi que l'objet stégo.

1.8Tatouage numérique ou Watermarking

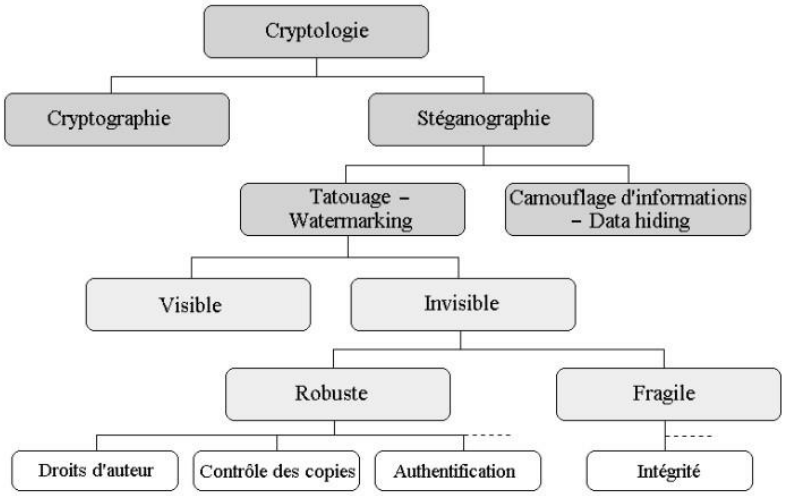


Figure 6 : Diagramme de méthodes de sécurisation[10]

La ***cryptologie*** est la science qui permet de protéger des données. Elle regroupe les deux méthodes existantes de protection de l'information : la Cryptographie et la Stéganographie. Pour la stéganographie, deux types d'approches sont envisageables. La première consiste à cacher l'information à l'intérieur d'un autre document : ***Camouflage d'informations*** ou ***Data Hiding***. La seconde méthode d'utilisation de la stéganographie est d'intégrer une signature dans le document traité. Cette partie est appelée ***Tatouage*** ou ***Watermarking***.

1.8.1Technologie du watermarking

Vers les années 1990, une nouvelle technologie est apparue à savoir le marquage numérique des signaux, appelée aussi tatouage numérique ou watermarking. La plus grande partie des applications de cette technologie est reliée à des applications de sécurité des données de médias audio et visuels, tel que la protection de droit d'auteur, le contrôle de l'intégrité des données, l'authentification, etc. Le processus de marquage comporte trois étapes [11] :

1. ***La génération de la marque :*** En général, la marque est un message transformé en une matrice W dont les valeurs sont binaires {±1} ou ternaires {1, 0, -1}.
2. ***L'insertion de la marque :*** Étant donné une image hôte I, l'insertion se fait soit dans le domaine spatial de l'image (dans les pixels), soit dans le domaine transformé de l'image (DCT ou DiscreteCosineTransform, DWT ou DiscreteWaveletTransform, …). De façon générale, l'insertion est une fonction qui prend en entrée l'image hôte I (ou image originale) et la marque générée W, et délivre en sortie l'image marquée I'. Ce processus est illustré dans les figures 7 et 8.
3. ***La détection/extraction de la marque :*** Étant données une image test I' et une marque W, cette étape consiste à analyser l'image test I' et vérifier si la marque W est

présente dans cette image. La vérification de la présence de la marque se fait par détection ou extraction :

a. ***Détection :*** Mesure du degré de présence de la marque par une fonction de corrélation entre I' et W. La comparaison à un seuil de corrélation aboutit à une décision binaire (marque détectée/marque non détectée) (figure 9).

b. ***Extraction :*** Extraire les bits de la marque W' contenue dans l'image test I' et la comparer à la marque originale W en utilisant une métrique donnée tel que le TDR (TrueDetection Rate), signifiant le nombre de bits correctement extraits sur le nombre de bits total de la marque. Une autre métrique utilisée est le BER (Bit Error Rate) indiquant le ratio du nombre de bits extraits et erronés sur le nombre de bits total de la marque. Ce processus est illustré par la figure10 [12].

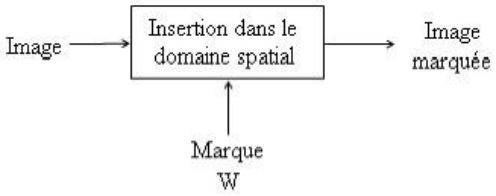


Figure 7 : Représentation du processus d'insertion de la marque dans le domaine spatial[12].

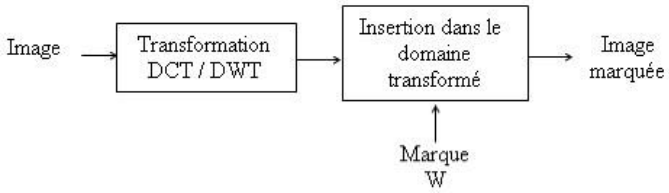


Figure 8 : Processus d'insertion de la marque dans le domaine transformé DCT ou DWT

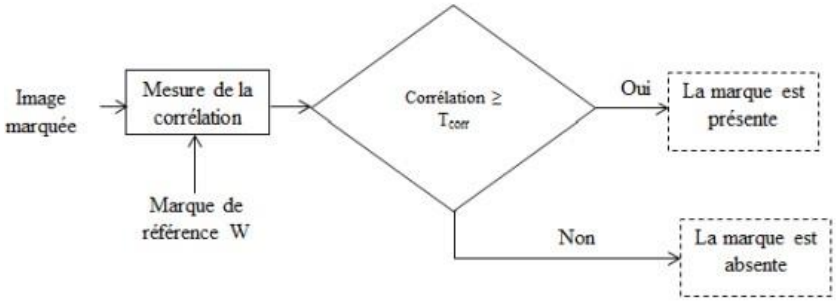


Figure 9 : Représentation du processus de détection de la marque par corrélation [13].

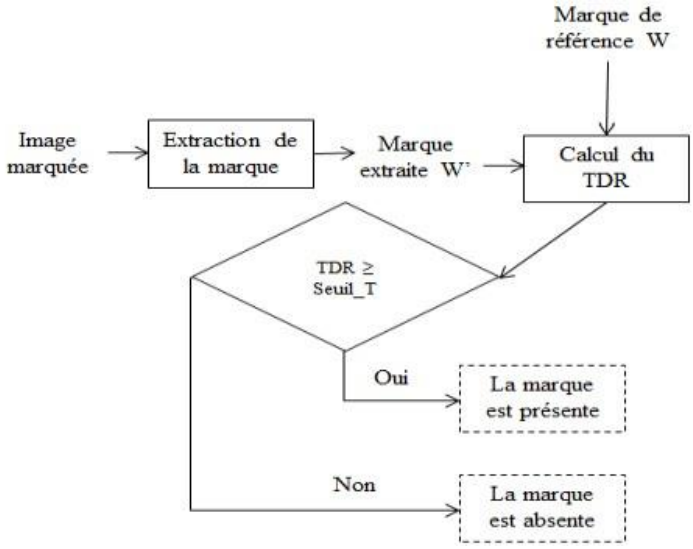


Figure 10 : Processus d'extraction de la marque et comparaison avec la marque de référence

1.8.2Qualités d'un tatouage

Les performances d'un tatouage sont appréciées sous les critères principaux suivants :

- l'invisibilité,
- la robustesse,
- l'inversibilité,
- le ratio,
- la complexité,
- les informations nécessaires lors de la détection.

L'invisibilité : l'invisibilité d'une marque est sa capacité à être dissimulée sur un support. Cette invisibilité se traduit aussi par le respect de la qualité du document. Par exemple, dans le cas où le support serait une image, celle-ci ne doit pas être dégradée. Dans certains cas, une modification importante du contenu numérique peut avoir des conséquences limitées sur la perception visuelle que nous avons de l'image, et inversement.

Robustesse : Nous pourrions séparer cette rubrique en deux parties : ***la robustesse et la sécurité***. Ces deux caractéristiques sont souvent confondues surtout dans le cas du tatouage. Nous parlons de ***robustesse*** pour définir la ***résistance du tatouage face à des transformations de l'image tatouée***. Ces transformations peuvent être de type géométrique (rotation, zoom, découpage). Elles peuvent modifier certaines caractéristiques de l'image (histogramme des couleurs, saturation). Il peut aussi s'agir de tous les types de dégradations fréquentielles de l'image (compression avec pertes, filtres passe-haut ou passe-bas, passage analogique-numérique-analogique, etc.). Une marque est robuste si elle est capable de résister aux attaques. En général, cette robustesse est plus ou moins importante suivant le choix du facteur d'insertion utilisé. Plus la force d'insertion est grande, plus la robustesse de la marque devrait être importante [14].

La sécurité caractérise la façon dont le marquage va résister à des attaques « malicieuses ». Nous pouvons faire des parallèles avec la cryptanalyse. Le pirate va chercher à laver l'image de façon intelligente. Il est sensé connaître l'algorithme et va, en général, chercher la clé qui lit le tatouage. Cela demande souvent une analyse approfondie de la technique de marquage employée.

L'inversibilité : L'inversibilité est la capacité d'un algorithme à extraire la marque de façon à restituer exactement l'image originale. Cette opération peut être utile par exemple en indexation. Les informations insérées dans le document peuvent être modifiées sans ajouter de dégradations au support ou de conflits dans les données insérées.

La complexité : La complexité indique le "nombre" et la nature des instructions algorithmiques nécessaires pour effectuer l'insertion de la marque ainsi que son extraction. Cette complexité va bien évidement indiquer le temps de calcul nécessaire à l'opération de tatouage. En particulier dans le cas de la vidéo, il sera préférable d'utiliser un algorithme peu complexe pour que les opérations d'un marquage ne soient pas coûteuses en temps de calcul.

La capacité ou ratio : Le ratio d'un système de tatouage numérique désigne le rapport : « nombre de données » à dissimuler sur « taille du document hôte ». Dans le cas du tatouage, généralement 16 à 64 bits sont suffisants pour assurer un service de droit d'auteurs. Pour des applications telles que l'indexation, l'algorithme de marquage devra être capable d'intégrer au message une marque contenant beaucoup plus d'informations. Dans certains cas, nous chercherons donc à intégrer une marque de grande capacité. De façon générale, plus le ratioestfaible, plus la robustesse et l'imperceptibilité peuvent être élevées [15].

Informations nécessaires lors de la détection :

- ***Le tatouage privé***, aussi appelé ***non aveugle***, où la donnée originale est nécessaire à l'extraction. Ce système fonctionne en deux étapes : d'abord il compare le support marqué avec le support original ; ensuite un algorithme de décision est appliqué pour répondre si oui ou non la marque détectée correspond bien à la marque appliquée à l'origine. L'intérêt de ce marquage est très limité.
- ***Le tatouage semi-privé***, aussi appelé ***semi-aveugle***, n'utilise pas la donnée originale, mais a besoin de la signature lors de l'extraction. Dans ce cas, il s'agit de répondre à la question : « telle marque est-elle dans l'image ? ». La majorité des algorithmes de tatouage actuels utilise ce mode de fonctionnement.
- ***Le tatouage public***, aussi appelé ***aveugle***, ne nécessite ni la marque d'origine, ni le support d'origine. Ce type d'algorithme peut être difficile à mettre en place.

Afin de faire une synthèse, nous pouvons noter que l'invisibilité, la robustesse et la capacité sont fortement liées. En effet, d'une manière générale plus la marque est robuste et plus elle est visible. Ceci explique la grande difficulté à élaborer des techniques de tatouage, en particulier pour des applications liées à la sécurité.

1.8.3Visibilité

Le principe fondamental du tatouage visible consiste à dissimuler partiellement une image. Pour ce faire, il faudra utiliser un nombre indéterminé de marques visibles, qui ne pourront être efficacement effacées que si l'on possède une "clé secrète" adéquate. Ce type de tatouage est étudié actuellement pour gérer tout ce qui concerne les contrôles d'accès d'un unique document, correspondant en quelque sorte à une distribution de permission. On entend par contrôle d'accès la possibilité de restreindre la divulgation d'un document, en fonction de l'appartenance d'un utilisateur ou non la classe des "*ayants droit*" à la lecture de ce document [16].

Figure 11 Exemple de tatouage visible

En pratique, l'intérêt d'un tatouage efficace réside dans son invisibilité. Et c'est d'ailleurs l'un des trois principaux critères d'un algorithme de marquage : faire en sorte que la différence avec l'original soit la plus minime possible [6].

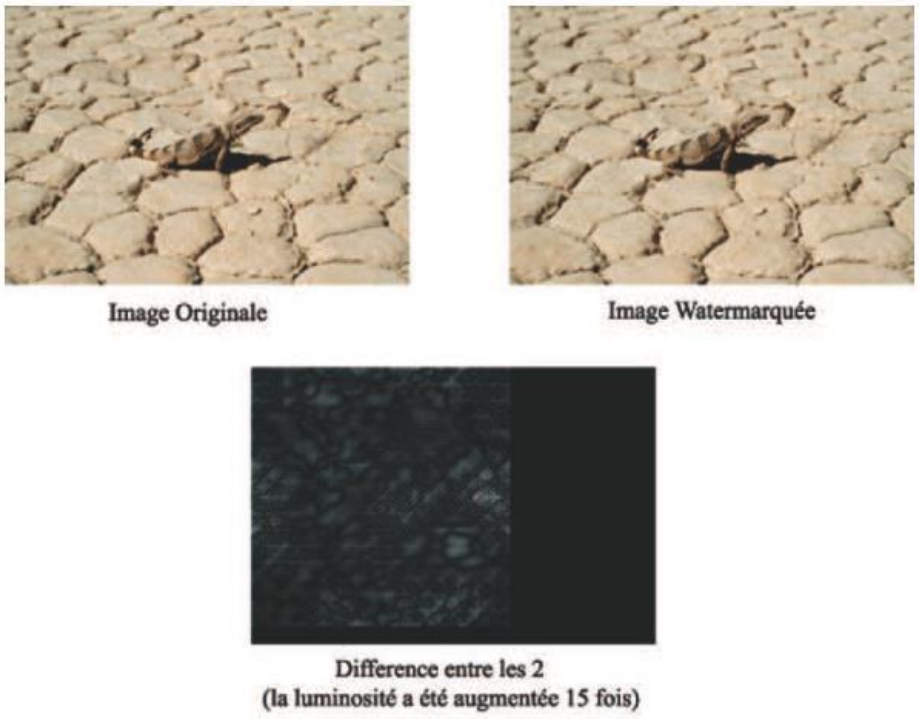


Figure 12Exemple de tatouage (pseudo) invisible

1.8.4Robustesse et fragilité

Le deuxième principal critère de qualité d'un algorithme de tatouage concerne sa robustesse face à des manipulations de l'image : celui-ci doit pouvoir conserver l'information stockée dans le marquage en dépit de diverses transformations.

Or très peux d algorithme résiste à une simple compression ou un changement de format, mais ils sont sensés résister tout de même à des attaques basiques telle que des translations ou des rotations de l'image (souvent utilisées pour la mise en page).

Il est néanmoins intéressant de remarquer qu'il peut être utile, dans certain cas, de favoriser une fragilité plutôt qu'une robustesse (figure 13).

Pour s'assurer par exemple de l'intégrité d'un document, le fait de le tatouer avec un algorithme fragile permettra, par la suite de vérifier si l'information marquée est toujours présente, ce qui sous-entend donc qu'elle n'a subi aucune modification malveillante (par exemple, une modification brutale de certaine partie textuelle). Cela permet donc une certaine falsification de l'image. Ce critère de robustesse et de fragilité s'applique surtout à un marquage invisible (il n'y a aucun intérêt à se poser ce genre de question pour un tatouage visible) [16].

Un cas intermédiaire, les Semi-fragiles : les tatouages semi-fragiles combinent à la fois les propriétés des marquages robustes et fragiles. Comme les robustes, ils tolèrent certains changements de l'image, comme des rotations, translations ou addition de bruit. Et comme les tatouages fragiles, ils sont capables de déterminer les régions où l'image a été brutalement modifiée et celles où elle reste authentique. Par conséquent les tatouages semi-fragiles arrivent à différencier les changements "légers" comme l'ajout d'un bruit à des changements « destructeurs ».

Ces algorithmes sont surtout très utiles par exemple dans le cas où une image marquée doit être diffusée sur le net, ou des types de compression comme le JPEG (Joint Photographic Experts Group), vont être employés. Un algorithme fragile ne supporterait pas ce type de transformation, et un robuste permettrait à quiconque récupèrerait cette image sur le net d'en modifier des parties sans que le marquage (donc le pseudo copyright servant à la falsification d'un document) soit altéré : tout le monde peut se présenter comme ayant un document original en sa possession.

L'exemple d'un tatouage fragile (figure13) permet d'illustrer aussi ce cas [17].

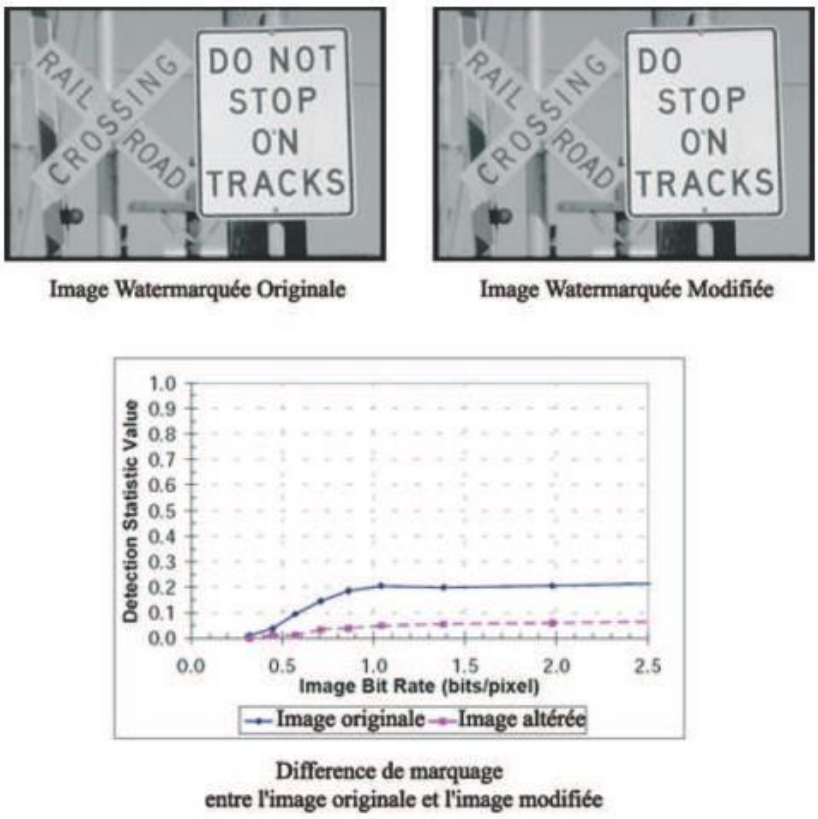


Figure 13Exemple de tatouage fragile [17]

1.8.5Applications

Plusieurs applications peuvent utiliser les techniques de tatouage. Pour chacune d'elles, les critères de qualité sont particuliers. Nous présentons ici les principales applications :

- droits d'auteurs,
- traçabilité (fingerprinting en anglais),
- protection contre les copies,
- authentification,
- indexation.

Droits d'auteurs : L'application la plus évidente du tatouage est le droit d'auteur. Le but est d'insérer une signature permettant d'identifier le propriétaire, de façon très robuste. Les deux principales qualités à respecter sont la robustesse et l'invisibilité de la marque.

Traçabilité (fingerprinting) : Le but de la traçabilité est de pouvoir contrôler et faire le suivi des copies de document. Cela implique de créer une marque originale pour chaque document distribué. Les qualités requises en termes de tatouage peuvent être alors classiquement la robustesse, l'invisibilité et le ratio afin d'éviter de fortes corrélations entre les diverses marques utilisées. Toutefois, nous pouvons envisager que dans certaines problématiques de traçabilité, le document transmis ne subit pas d'attaques malveillantes (restant dans un circuit privé). Dans ce cas, les qualités recherchées sont surtout l'invisibilité, mais aussi que l'insertion d'une nouvelle marque (par exemple pour chaque étape d'un processus), ne gomme pas les marques précédentes [18].

Protection contre les copies : La protection contre les copies consiste à intégrer au document une marque "intelligente", ou plus précisément un environnement matériel et/ou logiciel capable de communiquer avec la marque. Cela nécessite l'utilisation de matériels particuliers. En effet, les

appareils doivent pouvoir détecter la marque et agir en conséquence, c'est-à-dire en permettant ou non la lecture, ou la copie du document. Par exemple dans le cadre de la copie de DVD (Digital Versatile Disc), le lecteur et enregistreur de DVD doit être en mesure de lire et modifier la marque du document. Dans ce cas, pour une marque particulière, l'appareil acceptera l'enregistrement, mais modifiera le contenu de la marque pour interdire toute copie ultérieurement. Cela permet de contrôler le nombre et les utilisateurs des copies [19].

Authentification : Dans le cadre d'application telle que l'authentification, le but est de détecter les modifications effectuées sur une donnée. Ce marquage est qualifié de "fragile". Il doit être résistant à des attaques classiques mais doit être détruit en cas de modification de la donnée. Dans ce cas, une des solutions consiste à intégrer la marque sur les objets principaux de la donnée. Si un de ces objets est modifié ou supprimé, la marque est alors détruite.

Indexation

La marque intégrée au document dépend du contenu de celui-ci. L'algorithme utilisé devra permettre d'intégrer une marque de grande capacité. Par exemple, dans le cadre de la vidéo, la marque pourrait contenir la date d'enregistrement, le titre de la séquence, les noms des personnages ou objets principaux, etc. La marque devra être de grande capacité et invisible. La robustesse est à définir selon l'application mais n'est à priori qu'un problème secondaire.

1.9Comparaison de la stéganographie, du tatouage et du fingerprinting

Voici une petite comparaison de la stéganographie (Data Hiding), du tatouage et de l'empreinte [19] :

Caractéristiques	STÉGANOGRAPHIE *Steganography*	TATOUAGE *Watermarking*	EMPREINTE *Fingerprinting*
Données	Le message à transmettre	Une marque dépendant du support et/ou du propriétaire	Une empreinte dépendant du support et de son utilisateur
Support	Sans importance, le plus « banalisé » possible	Le document hôte dont on veut protéger les droits	Le document hôte dont on souhaite prévenir la diffusion de copie illégale
But de l'utilisateur	Cacher de l'information	Identifier l'émetteur (l'ayant droit)	Identifier le destinataire (l'utilisateur)
But de l'attaquant	Détecter les données et les extraire	Supprimer les données	Supprimer les données
Utilisation courante	Échapper à la censure	Copyrights, monitoring	Maîtrise de la diffusion

Tableau 1 Comparaison entre stéganographie, tatouage et empreinte [19]

1.10Conclusion

Nous avons pu cerner à travers ce chapitre ce qu'est l'art de la stéganographie, et les points à savoir sur le sujet. En outre, nous avons pu distinguer la différence entre la stéganographie, le tatouage ou watermarking, et l'empreinte ou fingerprinting. Le regain d'intérêt actuel pour la stéganographie provient des restrictions imposées à la cryptographie. D'une manière générale, la stéganographie arrive en renforcement au chiffrement de données. Pour permettre de garantir une

confidentialité maximum, les données sont tout d'abord chiffrées avant d'être dissimulées à l'aide d'un processus stéganographique. Le chapitre suivant concerne les techniques de stéganographie et de cryptage utilisées par notre application.

CHAPITRE 2 :CONCEPTION ET REALISATION DE L'APPLICATION

2.1Introduction

Ce chapitre est certainement un des plus importants de tous, étant donné qu'il concerne la conception et le développement de l'application proprement dite. C'est dans le prochain chapitre que nous décrirons les tests effectués et l'interprétation des résultats. Mais d'abord, certaines difficultés du développement pour des systèmes embarqués vont être abordées pour mieux cerner la délicatesse du projet, ainsi que quelques notions à savoir pour programmer sur Android, le système d'exploitation choisi pour diverses raisons expliquées dans le chapitre précédent.

2.2Les difficultés du développement pour des systèmes embarqués

Il existe certaines contraintes pour le développement Android, qui ne s'appliquent pas au développement habituel, par exemple au niveau des mémoires vives RAM (Random Access Memory) des téléphones généralement un peu faibles par rapport aux ordinateurs.

Voici les principales contraintes à prendre en compte quand on développe pour un environnement mobile [19]:

- Il faut pouvoir interagir avec un système complet sans l'interrompre. Android entreprend des activités pendant que notre application est utilisée, il reçoit des SMS et des appels, entre autres. Il faut respecter une certaine priorité dans l'exécution des tâches, par exemple il serait futile de bloquer les appels de l'utilisateur pour qu'il puisse terminer d'utiliser l'application.
- Le système n'est généralement pas aussi puissant qu'un ordinateur classique, il faudra donc exploiter tous les outils fournis afin de débusquer les portions de codes qui nécessitent des optimisations.
- La taille de l'écran est réduite, et il existe par ailleurs plusieurs tailles et résolutions différentes. Notre interface graphique doit s'adapter à toutes les tailles et toutes les résolutions.
- En outre, les interfaces tactiles sont peu pratiques en cas d'utilisation avec un stylet et/ou peu précises en cas d'utilisation avec les doigts, d'où des contraintes liées à la programmation événementielle plus rigides. En effet, il est possible que l'utilisateur se trompe souvent de bouton. Très souvent s'il a de gros doigts.
- Enfin, en plus d'avoir une variété au niveau de la taille de l'écran, on a aussi une variété au niveau de la langue, des composants matériels présents et des versions d'Android. Il y a une variabilité entre chaque téléphone et même parfois entre certains téléphones identiques. C'est un travail en plus à prendre en compte.

Les conséquences de telles négligences peuvent être terribles pour l'utilisateur. Si nous saturons le processeur, il ne pourra plus rien faire excepté redémarrer. Faire crasher une application ne fera en général pas complètement crasher le système, cependant il pourrait bien s'interrompre quelques temps et irriter profondément l'utilisateur.

Le développement de programmes pour un téléphone portable est donc différent de l'écriture d'applications pour des machines de bureau, du développement de sites web ou de la création de programmes serveurs.

A cela, Android essaie de nous faciliter les choses :

- Il fournit un langage de programmation connu (Java), avec des bibliothèques relativement classiques (certaines API d'Apache, par exemple), ainsi qu'un support pour les outils auxquels nous sommes habitués (Eclipse, notamment).
- Il nous offre un framework suffisamment rigide et étanche pour que nos programmes s'exécutent "correctement" sur le téléphone, sans interférer avec les autres applications ou le système d'exploitation lui-même [20].

2.3Android Studio IDE

Android Studio est un environnement de développement pour développer des applications mobiles Android. Il est basé sur IntelliJ IDEA et utilise le moteur de production Gradle. Il peut être téléchargé sous les systèmes d'exploitation Windows, macOS, Chrome OS et Linux.

Android Studio permet principalement d'éditer les fichiers Java/Kotlin et les fichiers de configuration XML d'une application Android.

Il propose entre autres des outils pour gérer le développement d'applications multilingues et permet de visualiser rapidement la mise en page des écrans sur des écrans de résolutions variées simultanément. Il intègre par ailleurs un émulateur permettant de faire tourner un système Android virtuel sur un ordinateur.

2.4Android SDK

Le kit de développement (SDK) d'Android est un ensemble complet d'outils de développement. Il inclut un débogueur, des bibliothèques logicielles, un émulateur basé sur QEMU, de la documentation, des exemples de code et des tutoriels. Les plateformes de développement prises en charge par ce kit sont les distributions avec le noyau Linux, Mac OS X 10.5.8 ou plus, Windows XP ou version ultérieure. L'IDE officiellement supporté était Eclipse combiné au module d'extension avec les outils de développement d'Android (ADT) mais depuis 2015, Google officialisa Android Studio, qui devient alors l'IDE officiel pour ce kit de développement d'Android. Les développeurs peuvent utiliser n'importe quel éditeur de texte pour modifier les fichiers Java et XML, puis utiliser les outils en ligne de commande (Java Development Kit et Apache Ant sont obligatoires) pour créer, construire et déboguer les applications Android ainsi que contrôler des périphériques Android (pour déclencher un redémarrage, installer un logiciel à distance ou autre).

2.5Le Java Development Kit

Il existe deux plateformes en Java :

- Le **JRE** (Java RuntimeEnvironment), qui contient la JVM (Java Virtual Machine), les bibliothèques de base du langage ainsi que tous les composants nécessaires au lancement d'applications ou d'applets Java. En gros, c'est l'ensemble d'outils permettant d'exécuter des applications Java.
- Le **JDK** (Java Development Kit), qui contient le JRE (afin d'exécuter les applications Java), mais aussi un ensemble d'outils pour compiler et déboguer le code.

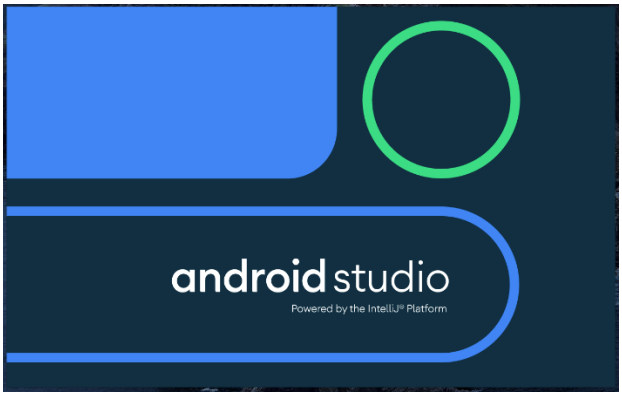


Figure 14 Splash screen d'Android Studio

Figure 6La barre d'outils d'Android Studio

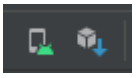


Figure 7Icônes réservées à la SDK et à l'Android Virtual Device AVD

Cliquer sur le bouton Android SDK Manager permet d'ouvrir l'outil de gestion du SDK d'Android.

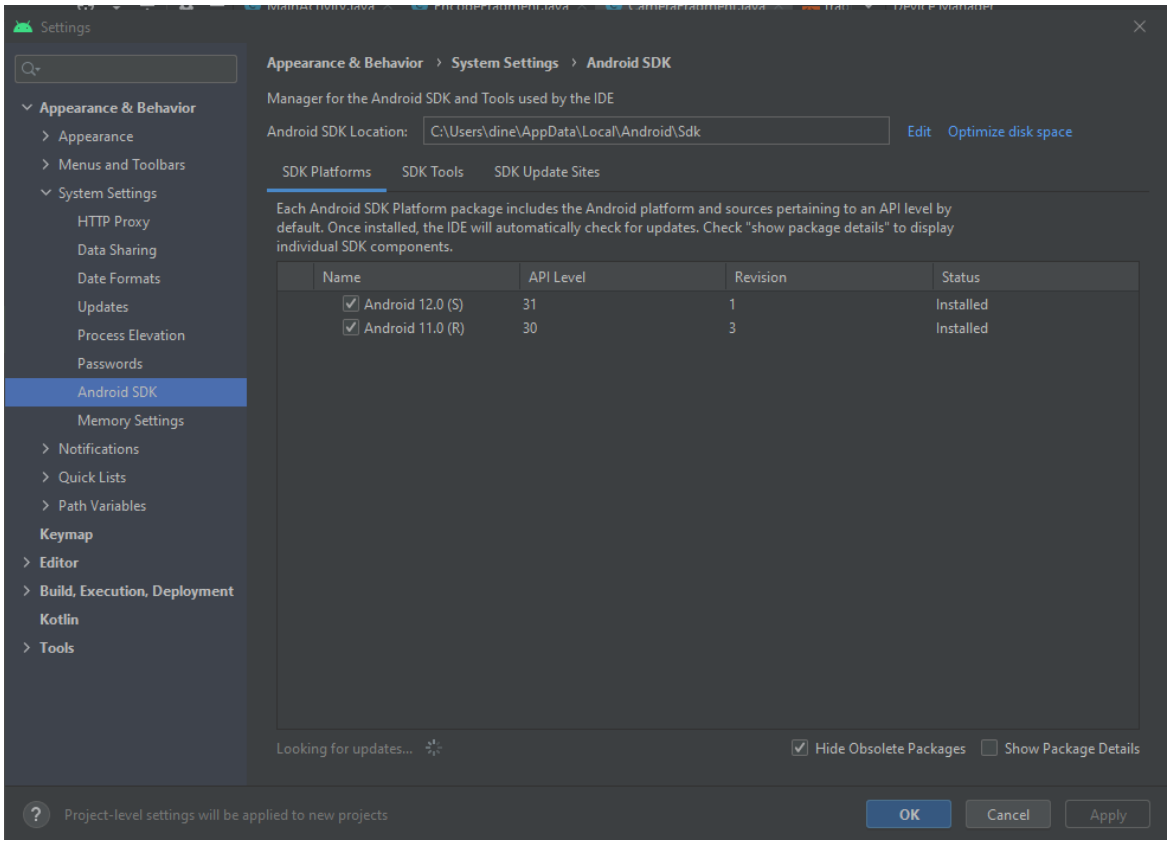


Figure 8SDK Manager d'Android

En ce qui concerne le nom des paquets, nous pouvons remarquer qu'ils suivent tous un même motif. Il est écrit à chaque fois Android [un nombre] (API [un autre nombre]). La présence de ces nombres s'explique par le fait qu'il existe plusieurs versions de la plateforme Android qui ont été développées depuis ses débuts et qu'il existe donc plusieurs versions différentes en circulation.

Le premier nombre correspond à la version d'Android et le second à la version de l'API Android associée. Quand on développe une application, il faut prendre en compte ces numéros, puisqu'une application développée pour une version précise d'Android ne fonctionnera pas pour les versions précédentes.

Nous avons choisi de délaisser les versions précédant la version 12 (l'API 31), de façon à ce que l'application puisse fonctionner pour la plupart des nouveaux smartphones.

Il est peut être injuste de laisser de côté les personnes qui sont contraintes d'utiliser encore ces anciennes versions, mais il faut savoir qu'ils ne représentent que 0,5 % du parc mondial des utilisateurs d'Android. De plus, les changements entre la version 4.4 et la version 12 sont trop importants pour être ignorés. Ainsi, l'application que nous développerons fonctionnera sous Android 4.4 minimum.

2.6Configuration de l'émulateur de téléphone AVD et du terminal réel

L'Android Virtual Device, aussi appelé **AVD**, est un émulateur de terminal sous Android utile pour développer et tester l'application. Voir la figure 18pour l'icône de lancement.

Figure 9Terminal virtuel

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth. La machine est lourde à utiliser, voire très lourde sur les machines les plus modestes ; Il est beaucoup moins confortable à manipuler qu'un vrai terminal sous Android. Il faut aller dans la configuration des paramètres du téléphone virtuel, entrer dans l'option « Développement » puis cocher « Débogage USB » pour rendre le terminal apte à la programmation.

Configuration du terminal réel : Il faut configurer le téléphone de la même façon que l'émulateur. En plus, il faut indiquer que nous acceptons les applications qui ne proviennent pas du Market dans Configuration > Application > Source inconnue.
Pour les utilisateurs de Windows, dans notre cas, il faut d'abord télécharger les drivers adaptés à notre terminal.

2.7Contenu d'un programme Android

Android utilise les mêmes concepts que la programmation classique, mais proposés de façon différente, avec une structure permettant de mieux protéger le fonctionnement des téléphones. Les composants principaux d'une application Android sont:

Les activités (activities) : Ce sont les briques de base de l'interface utilisateur. Une activité peut être considérée comme l'équivalent Android de la fenêtre ou de la boîte de dialogue d'une application classique. Bien que des activités puissent ne pas avoir d'interface utilisateur, un code "invisible" sera délivré le plus souvent sous la forme de fournisseurs de contenus (content provider) ou de services.

Les fournisseurs de contenus (content providers) : Ils offrent un niveau d'abstraction pour toutes les données stockées sur le terminal, accessibles aux différentes applications. Le modèle de développement Android encourage la mise à disposition de ses propres données aux autres programmes ; construire un fournisseur de contenus permet d'obtenir ce résultat tout en gardant un contrôle total sur la façon dont on accédera aux données.

Les services : Les activités et les fournisseurs de contenus ont une durée de vie limitée et peuvent être éteints à tout moment. Les services sont en revanche conçus pour durer et, si nécessaire, indépendamment de toute activité. Nous pouvons, par exemple, utiliser un service pour vérifier les mises à jour d'un flux RSS (le terme RSS ou Really Simple Syndication signifie que le contenu du fichier est informatiquement codé selon le standard RSS, qui s'appuie lui-même sur le langage informatique XML ouExtensible MarkupLanguage) ou pour jouer de la musique, même si l'activité de contrôle n'est plus en cours d'exécution.

Les intentions (intents) : Ce sont des messages du système émis par le terminal pour prévenir les applications de la survenue de différents événements, que ce soit une modification matérielle (comme l'insertion d'une carte SD, SD pour Secure Digital) ou l'arrivée de données (telle la réception d'un SMS), en passant par les événements des applications elles-mêmes (notre activité a été lancée à partir du menu principal du terminal, par exemple). Nous pouvons non seulement répondre aux intentions, mais également créer les nôtres afin de lancer d'autres activités ou pour nous prévenir qu'une situation particulière a lieu.

2.8Cycle de vie d'une activité

Pour développer d'une application sur Android, nous devons comprendre le cycle de vie d'une activité. Le cycle de vie d'une activité est exprimé par la figure suivant :

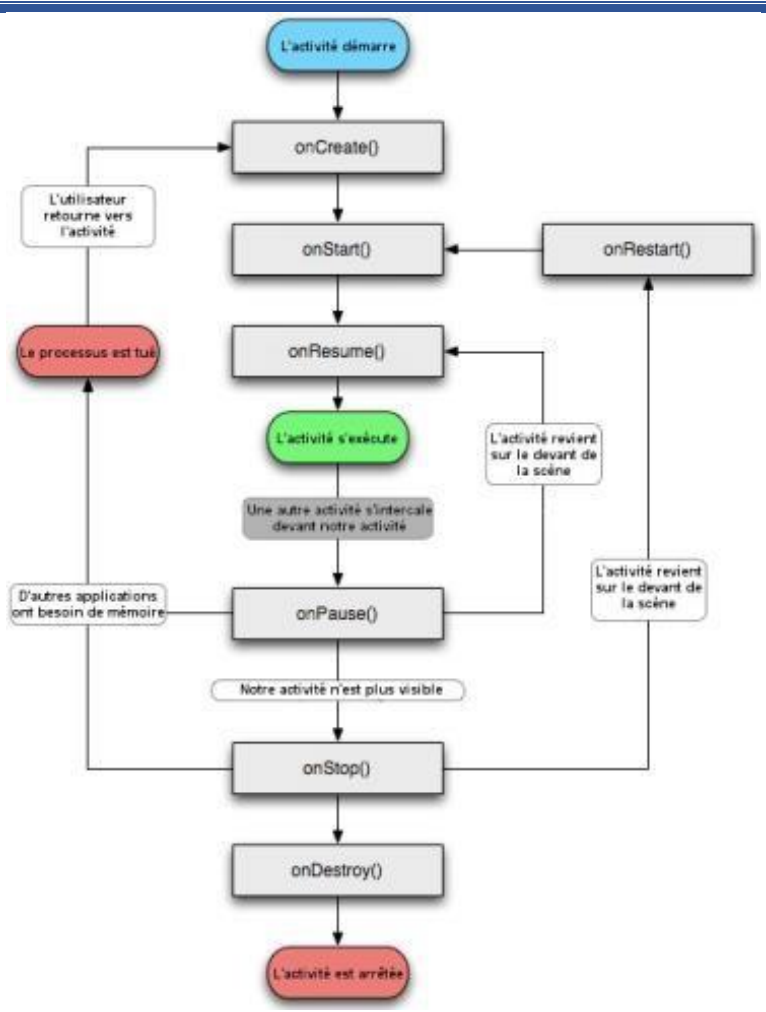


Figure 10Cycle de vie d'une activité

En général, une activité se trouve toujours dans l'un des quatre états suivants:

- ***Active :*** L'activité a été lancée par l'utilisateur, elle s'exécute au premier plan. C'est à cet état que l'on pense quand on évoque le fonctionnement d'une activité.
- ***En pause :*** L'activité a été lancée par l'utilisateur, elle s'exécute et elle est visible, mais une notification ou un autre événement occupe une partie de l'écran. Pendant ce temps, l'utilisateur voit l'activité mais peut ne pas être capable d'interagir avec elle. Lorsqu'un appel téléphonique est reçu, l'utilisateur a l'opportunité de prendre cet appel ou de l'ignorer, par exemple.
- ***Stoppée :*** l'activité a été lancée par l'utilisateur, elle s'exécute mais est cachée par d'autres activités qui ont été lancées ou vers lesquelles le système a basculé. Notre application ne pourra rien présenter d'intéressant à l'utilisateur directement : elle ne peut passer que par une notification.
- ***Morte :*** l'activité n'a jamais été lancée (le téléphone vient d'être réinitialisé, par exemple) ou elle a été tuée, éventuellement à cause d'un manque de mémoire.

Il existe trois boucles principales :

La durée de vie d'une activité se passe entre le premier appel à ***OnCreate ()*** et l'appel à ***onDestroy ()***. Une activité met en place tous les états globaux dans la méthode ***onCreate ()*** et libère toutes les ressources restantes à ***onDestroy ()***.

La durée de vie visible d'une activité se passe entre un appel à ***onStart ()*** jusqu'à un appel correspondant à ***onStop ()***, durant laquelle l'utilisateur peut voir l'activité sur l'écran, même si elle n'est pas à l'avant et à l'interaction avec l'utilisateur. Entre ces deux méthodes, les ressources qui sont nécessaires pour montrer l'activité de l'utilisateur sont conservées.

La durée de vie d'une activité en avant-plan se passe entre un appel à ***onResume ()*** jusqu'à un appel correspondant à ***onPause ()***, durant laquelle l'activité est en face de toutes les autres activités afin d'interagir avec l'utilisateur. Une activité peut souvent changer son état entre l'état de reprise et l'état en pause.

2.9Analyse et Conception

Idéalement, implémenter quelques techniques de stéganographie sur Android est l'objectif principal. L'application Android développée devra être capable de cacher un message secret ou une image secrète dans une image de couverture. L'application devra permettre de choisir deux algorithmes de stéganographie, DWT et F5 pour cacher le texte, préalablement chiffré AES, et un seul algorithme, donner le choix a l'utilisateur de selectioné un algorithme pour cacher un texte sur une image déjà enregistre sur le téléphone ou une image prise par l'appareil photo du téléphone sur l'application.

2.9.1Identification des besoins fonctionnels

L'application est construite à l'aide de groupe de modules. Chaque module effectue une certaine tâche. Les besoins basés sur chaque tâche sont appelées besoins fonctionnels.

En analysant les spécificités de l'application, et en gardant à l'esprit les besoins impératifs de l'utilisateur, l'idée est de développer une application simple, intuitive, design et exécutant les tâches le plus rapidement possible, et donc comportant six fonctions :

- Extraire un texte à partir d'un stégo-image encodé DWT.
- Encoder un texte dans une image suivant l'algorithme F5.
- Extraire un texte à partir d'un stégo-image encodé F5.
- Encoder une image dans une image suivant la technique DWT.
- Extraire une image à partir d'un stégo-image encodé DWT

Ces principales fonctions étant définies, voici d'autres fonctionnalités utiles de l'application :

Cryptage du texte en AES à l'aide d'un mot de passe Partage de l'image traitée après opération.

La spécification et la validité des entrées seront expliquées plus loin.

2.9.2Identification des besoins non fonctionnels

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumise l'application pour sa réalisation et son bon fonctionnement :

- ***Ergonomie et souplesse*** : l'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur.
- ***Rapidité :*** l'application doit optimiser les traitements pour avoir un court temps de réponse.
- ***Efficacité et fiabilité :*** l'application doit remplir ses fonctions indépendamment de toutes circonstances pouvant entourer l'utilisateur de manière fiable, et ainsi garantir la confidentialité, l'imperceptibilité et l'intégrité de l'information cachée.
- ***Maintenabilité et évolutivité :*** le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

Pour décrire la conception de l'application, nous verrons le diagramme de cas d'utilisation et le diagramme d'activité. Par la suite nous passerons au diagramme de classe dans la section implémentation.

2.9.3Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation est une représentation du comportement du système de point de vue de l'utilisateur, c'est une définition des besoins qu'attend un utilisateur du système, il contient tous les cas d'utilisation en liaison directe ou indirecte avec les acteurs. Il aide à gérer la complexité de l'application. Toutes les fonctionnalités requises sont décomposées en plusieurs petits scénarios.

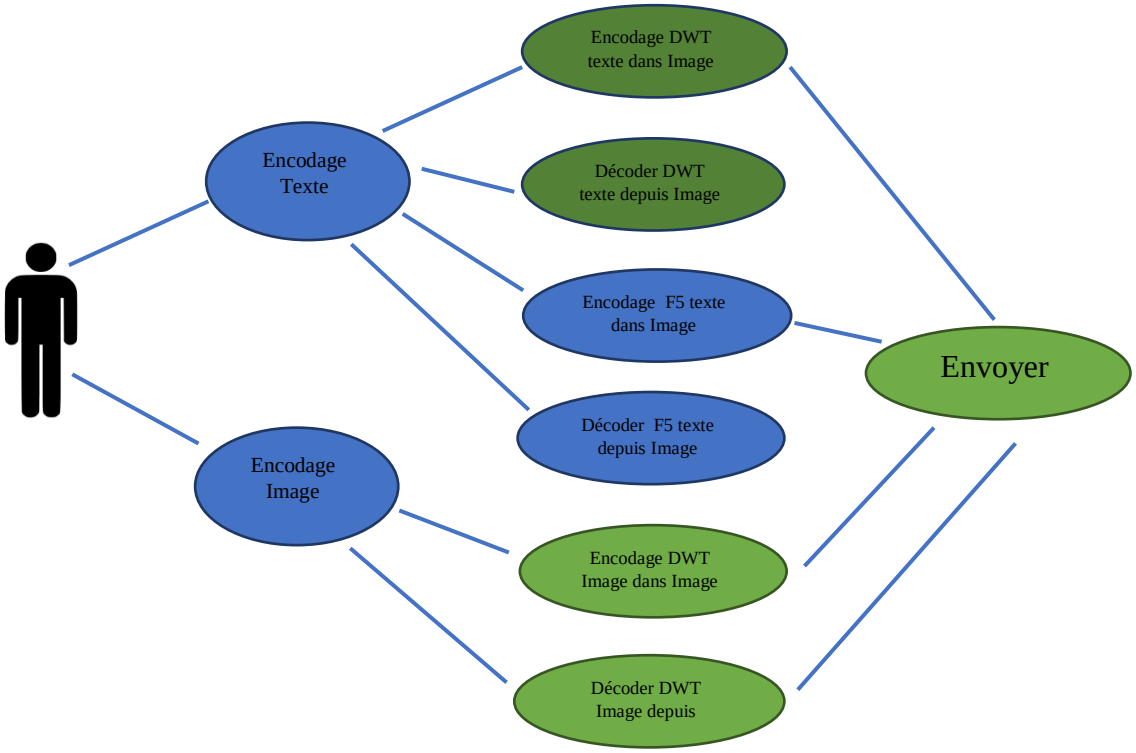


Figure 20Diagramme de cas d'utilisation de l'application

Cas d'utilisation 1	**Encodage Texte**
Intérêts et objectif	L'onglet « Encodage Texte » affiche les fonctionnalités disponibles pour la stéganographie de texte dans une image de l'application. En outre, l'utilisateur peut naviguer librement vers l'onglet « Encodage Image ».
Préconditions	
Post-conditions réussis	Une vue générale des options pour cette rubrique de l'application est présentée avec une navigation facile.
Post-conditions en échec	
Scénario nominal	Etape 1 : Après le lancement de l'application ou à partir de l'onglet « Encodage Image », l'utilisateur arrive à cet onglet. Etape 2 : L'utilisateur peut passer à l'étape suivante ou aller vers l'onglet « Encodage d'image ».

Tableau 2Cas d'utilisation n°1

Cas d'utilisation 2	**Encodage Image**
Intérêts et objectif	L'onglet « Encodage » Image affiche les fonctionnalités disponibles pour la stéganographie d'image dans une image de l'application. En outre, l'utilisateur peut naviguer librement vers l'onglet « Encodage Texte ».
Préconditions	
Post-conditions réussis	Une vue générale des options pour cette rubrique de l'application est présentée avec une navigation facile.
Post-conditions en échec	
Scénario nominal	Etape 1 : Après le lancement de l'application, l'utilisateur choisit l'onglet « Encodage Image ». Etape 2 : L'utilisateur peut passer à l'étape suivante ou aller vers l'onglet « Encodage Texte ».

Tableau 3Cas d'utilisation n°2

Cas d'utilisation 3	**Encoder DWT Texte dans Image**
Intérêts et objectif	Cacher un message secret dans une image avec la technique DWT.
Préconditions	
Post-conditions réussis	Génération d'un stégo-image au format png, contenant le message
Post-conditions en échec	Message d'erreur pour une image invalide ou champs incomplets
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage», Etape 3 : choisit l'image de couverture, soit dans la galerie d'images, soit en cliquant sur le bouton « Appareil » pour pendre une photo à l'aide de l'appareil photo du terminal Etape 4 : entre le message secret, Etape 5 : entre un mot de passe si nécessaire, Etape 6 : choisit l'algorithme DWT, Etape 7 : choisit l'opération « Encoder ». Etape 8 : clique le bouton « Lancer ».

Tableau 4Cas d'utilisation n°3

Cas d'utilisation 4	**Décoder DWT Texte depuis Image**
Intérêts et objectif	Extraire un message secret dans un stégo-image DWT.
Préconditions	
Post-conditions réussis	Affichage du message secret
Post-conditions en échec	Renvoi d'un message d'erreur pour une image ou un mot de passe invalide, ou absence de message dans l'image
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Texte », Etape 3 : choisit le stégo-image dans la galerie d'images, Etape 4 : entre un mot de passe si nécessaire, Etape 5 : choisit l'algorithme DWT, Etape 6 : choisit l'opération « Décoder », Etape 7 : clique le bouton « Lancer ».

Tableau 5Cas d'utilisation n°4

Cas d'utilisation 5	**Encoder F5 Texte dans Image**
Intérêts et objectif	Cacher un message secret dans une image avec l'algorithme F5.
Préconditions	
Post-conditions réussis	Génération d'un stégo-image au format jpg, contenant le message
Post-conditions en échec	Renvoi d'un message d'erreur pour une image invalide ou des champs incomplets
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Texte », Etape 3 : choisit l'image de couverture, soit dans la galerie d'images, soit en prenant une photo à l'aide de l'appareil photo, Etape 4 : entre le message secret, Etape 5 : entre un mot de passe si nécessaire, Etape 6 : choisit l'algorithme F5, Etape 7 : choisit l'opération « Encoder ». Etape 8 : clique le bouton « Lancer ».

Tableau 6Cas d'utilisation n°5

Cas d'utilisation 6	**Décoder F5 Texte depuis Image**
Intérêts et objectif	Extraire un message secret dans un stégo-image F5.
Préconditions	
Post-conditions réussis	Affichage du message secret
Post-conditions en échec	Renvoi d'un message d'erreur pour une image ou un mot de passe invalide, ou absence de message dans l'image
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Texte », Etape 3 : choisit le stégo-image dans la galerie d'images, Etape 4 : entre un mot de passe si nécessaire, Etape 5 : choisit l'algorithme F5, Etape 6 : choisit l'opération « Décoder », Etape 7 : clique le bouton « Lancer ».

Tableau 7Cas d'utilisation n°6

Cas d'utilisation 7	**Encoder DWT Image dans Image**
Intérêts et objectif	Cacher une image dans une image avec l'algorithme DWT.
Préconditions	
Post-conditions réussis	Génération d'un stégo-image au format png, contenant l'image secrète
Post-conditions en échec	Renvoi d'un message d'erreur pour des images invalides
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Image », Etape 3 : choisit l'image de couverture dans la galerie d'images en cliquant sur le bouton correspondant, Etape 4 : choisit l'image secrète à cacher dans la galerie d'images en cliquant sur le bouton correspondant, Etape 5 : choisit l'opération « Encoder », Etape 6 : clique le bouton « Lancer ».

Tableau 8Cas d'utilisation n°7

Cas d'utilisation 8	**Décoder DWT Image depuis Image**
Intérêts et objectif	Extraire une image secrète dans un stégo-image DWT.
Préconditions	
Post-conditions réussis	Génération et affichage de l'image secrète cachée au format png.
Post-conditions en échec	Renvoi d'un message d'erreur pour l'image invalide, ou absence d'image secrète dans l'image choisie
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Image », Etape 3 : choisit le stégo-image dans la galerie d'images, Etape 4 : choisit l'opération « Décoder », Etape 5 : clique le bouton « Lancer ». Etape 6 : choisit l'opération « Décoder », Etape 7 : clique le bouton « Lancer ».

Tableau 9Cas d'utilisation n°8

Cas d'utilisation 9	**Envoyer**
Intérêts et objectif	Partager le stégo-image généré ou une image extraite.
Préconditions	Au moins une application de partage, par exemple via WiFi, Bluetooth, email, MMS ou autres, est installée sur le terminal.
Post-conditions réussis	L'image partagée est effectivement reçue par un autre terminal ou prise en charge par une application choisie par l'utilisateur.
Post-conditions en échec	Aucune application permettant de prendre en charge l'image à partager n'est installée sur le terminal
Scénario nominal	Etape 1 : L'utilisateur démarre l'application, Etape 2 : entre dans l'onglet « Encodage Image » ou « l'onglet Encodage Texte », Etape 3 : effectue une opération, Etape 4 : clique sur le bouton « Envoyer », Etape 5 : choisit une application proposée par l'application

Tableau 10Cas d'utilisation n°9

2.9.4Diagramme d'activité

Le diagramme d'activité doit représenter l'ensemble des actions réalisées par le système, avec tous les branchements conditionnels et toutes les boucles possibles. C'est un graphe orienté d'actions et de transitions. Les transitions sont franchies lors de la fin des actions ; des étapes peuvent être réalisées en parallèle ou en séquence.

L'application commence par l'utilisateur qui choisit l'encodage de texte ou d'image. Ensuite, si l'utilisateur, après avoir choisi l'encodage texte désire encoder un message secret, choisit d'abord une image de couverture, un éventuel mot de passe, un algorithme de stéganographie, F5 ou LSB, l'option encodage, puis lance le traitement.

Le stégo-image généré peut être ensuite partagé par Wi-Fi (Wireless Fidelity), Bluetooth ou pris en charge par d'autres applications installées sur le terminal. L'utilisation des autres fonctionnalités est assez intuitive. Il faut seulement remarquer que pour l'encodage d'image dans une image, seul l'algorithme LSB est implémenté en raison de sa simplicité et de la rapidité de la technique.

Des entrées invalides ou manquantes font appel à un message d'erreur.

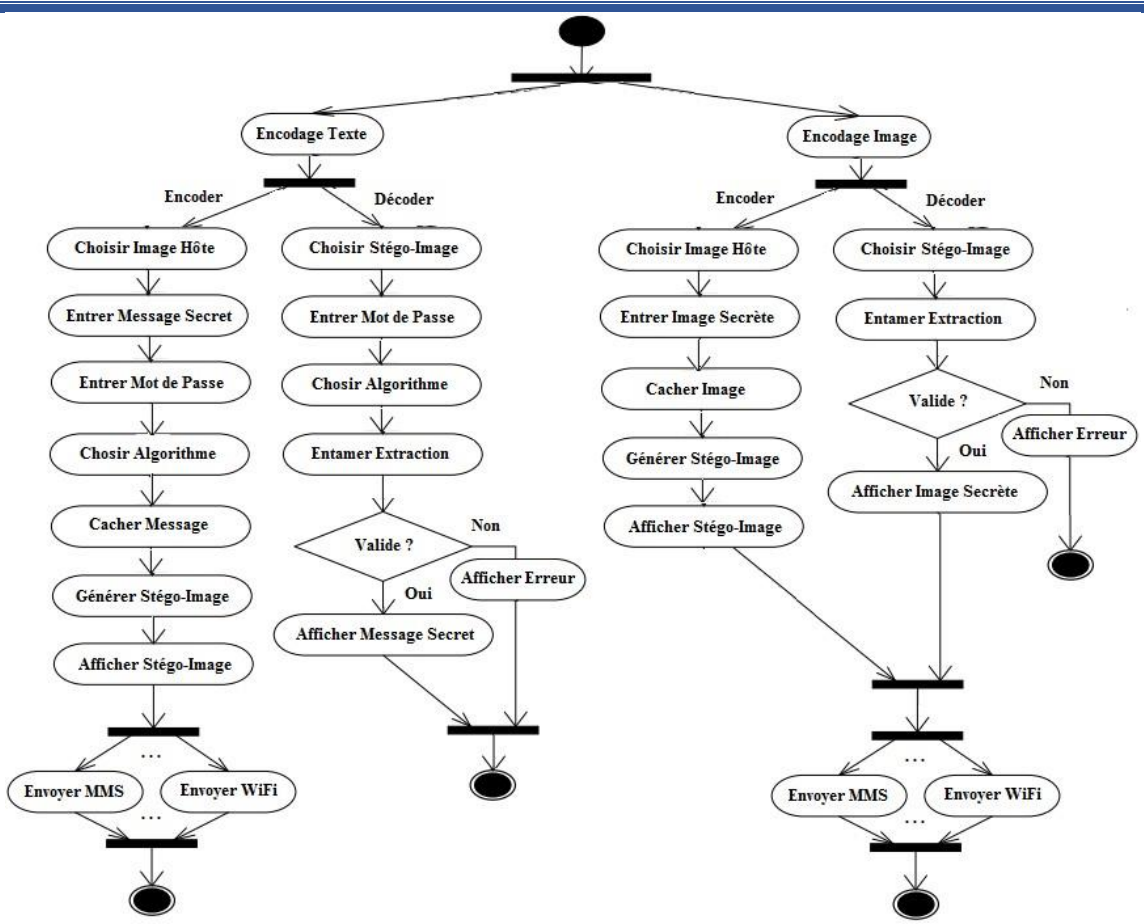


Figure 11Diagramme d'activité de l'application

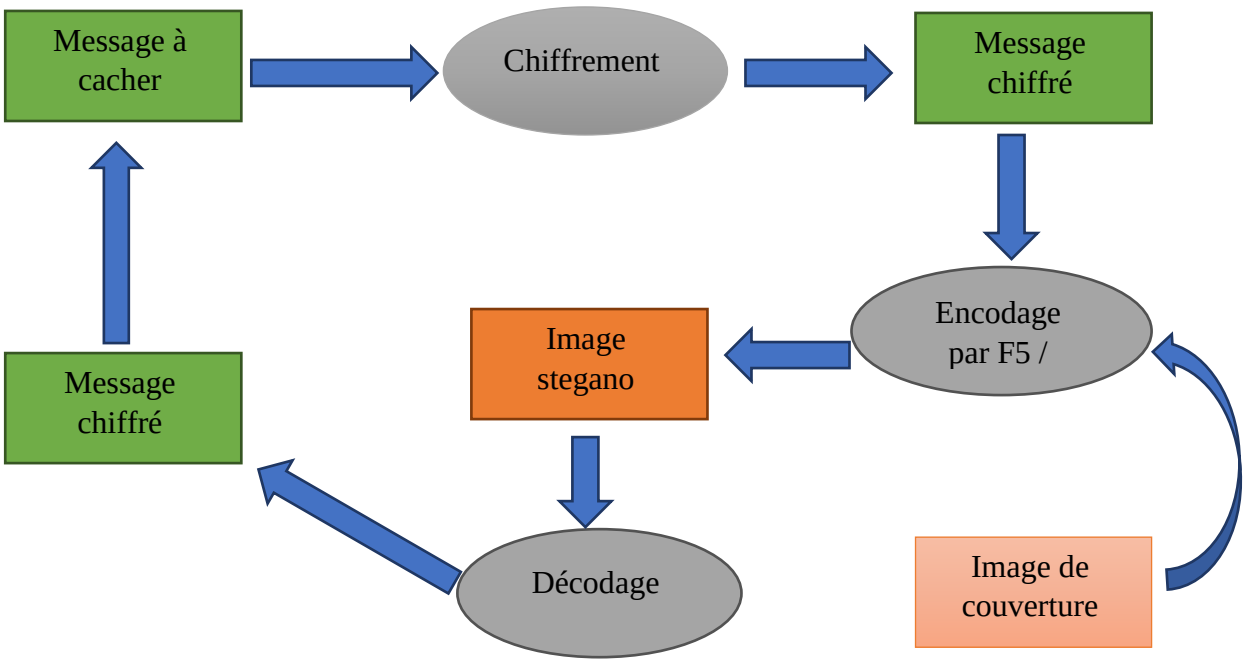


Figure 22 Processus stéganographique d'encodage et de décodage

2.9.5Techniques utilisées

L'application aura donc comme principales fonctions l'encodage de textepréalablement chiffré dans une image par l'algorithme F5 et DWT, le décodage de texteavec déchiffrement à partir d'un stégo-image, l'encodage d'image dans une image, et l'extraction d'image secrète à partir d'un stégo-image.

Le texte sera crypté en AES, et les algorithmes de stéganographie utilisés : le DWT ou F5 pour l'encodage de texte, et DWT pour l'encodage d'image.

Chiffrement AES : La méthode de chiffrement utilisée pour le texte sera AES dont la longueur de la clé est de 128 bits. Cette clé sera générée à partir du mot de passe fourni par l'utilisateur.

Technique DWT :Premièrement, les images secrètes et de couverture sont divisées en sous-bandes à l'aide de DWT. La plupart des informations sont obtenues en bande LL. Par conséquent, SVD a été exécuté sur la sous-bande LL de l'image secrète pour obtenir trois matrices individuelles U, S et V. Ajoutez ces matrices individuelles à la bande LL de l'image de couverture pour générer trois images stego. La qualité des images stego est évaluée par des métriques d'image telles que PSNR, SC, etc., l'algorithme proposé donne les meilleurs résultats par rapport aux algorithmes existants.

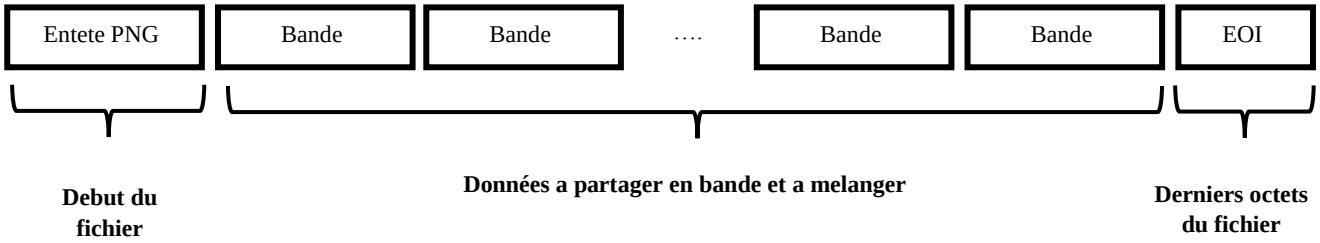


Figure 23 Format du paquet encode DWT

Il y a quelques points essentiels à savoir :

- Nous insérons des délimiteurs de début (156h) et de fin (1A7h) pour faire savoir à l'application qu'un texte (ou une image) est caché dans l'image choisie par l'utilisateur, et plus précisément dans quelle partie de l'image, ce qui sera utile pour le décodage.
- Pour l'encodage de texte, le message sera toujours chiffré pour assurer une certaine confidentialité, même si l'utilisateur n'entre pas de mot de passe (un mot de passe par défaut sera utilisé pour générer la clé).
- Pour l'encodage d'image, la largeur et la hauteur de l'image à cacher seront encodées en premier, suivi de ses données proprement dites, dans l'image de couverture.

Algorithme F5 : cette technique un peu plus complexe a aussi été déjà détaillée auparavant. Elle utilise la compression JPEG, elle cache le message (préalablement chiffré) à travers les coefficients DCT de l'image de couverture. Le paramètre p et la longueur du texte chiffré seront encodée en premier par une technique LSB, puis le message chiffré proprement dit sera encodé par la technique de matrix embedding. Voici la structure du paquet encodé F5 :

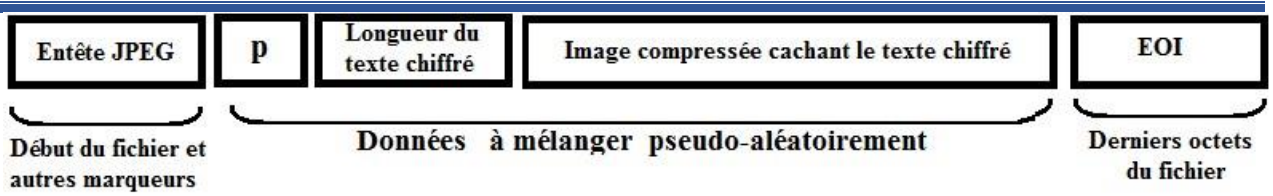


Figure 24 :*Format du paquet encodé F5*

Conclusion

Le travail présenté dans ce chapitre consiste à expliquer la technique de dissimulation d'informations (stéganographie) sur une application mobile base sur le système d'exploitation Android. L'image est l'une des supports les plus largement utilisés par la stéganographie. La raison de l'importance de ce type de support dans le domaine de dissimulation secrète réside dans les différents types d'images numériques.

CHAPITRE 3 : DEVELOPPEMENT DE L'APPLICATION

3.1. Introduction

Ce chapitre a pour but d'expliquer l'objectif et quelques options techniques de l'application, ainsi la conception et l'architecture générale de l'application et les outils nécessaires. De plus, savoir comment faire configurer et saisir les paramètres et les autorisations nécessaires sur l'appareil ou la machine virtuel pour faire fonctionner l'application de stéganographie pour s'assurer un bon fonctionnement de l'application.

3.2. Outils Utilisés

Pour développer cette application, nous avons utilisés :

- **L'environnement de développement Android Studio**

  Android Studio est l'environnement de développement intégré de la plate-forme Android de Google. Les versions d'Android Studio sont compatibles avec certains systèmes d'exploitation Apple, Windows et Linux. Avec la prise en charge de Google Cloud Platform et de l'intégration d'applications Google, Android Studio offre aux développeurs une boîte à outils bien fournie pour créer des applications Android ou d'autres projets, et fait partie intégrante du développement Android depuis 2013.

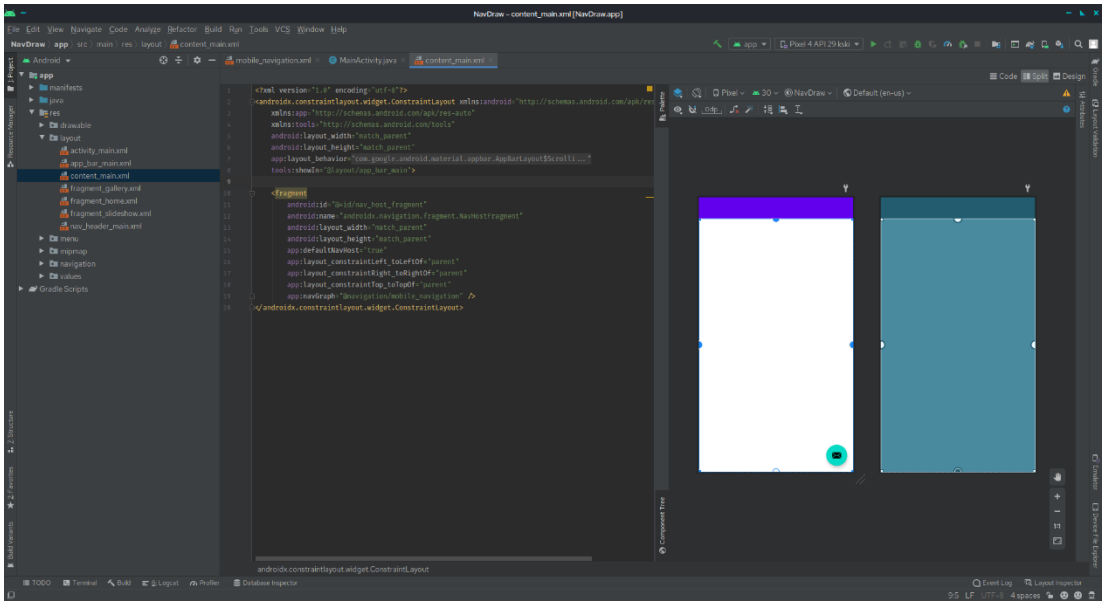


Figure 25 Environnement Android Studio

- **Le SDK de l'android :**

  Le SDK Android (kit de développement logiciel) est un ensemble d'outils de développement utilisés pour développer des applications pour la plate-forme Android qui est devenue le plus grand rival d'Apple dans le domaine des smartphones. Le SDK Android inclut les éléments suivants :

  - Bibliothèques requises.
  - Débogueur.

- Un émulateur.
- Documentation pertinente pour les interfaces de programme d'application (API) Android.
- Exemple de code source.
- Tutoriels pour le système d'exploitation Android.

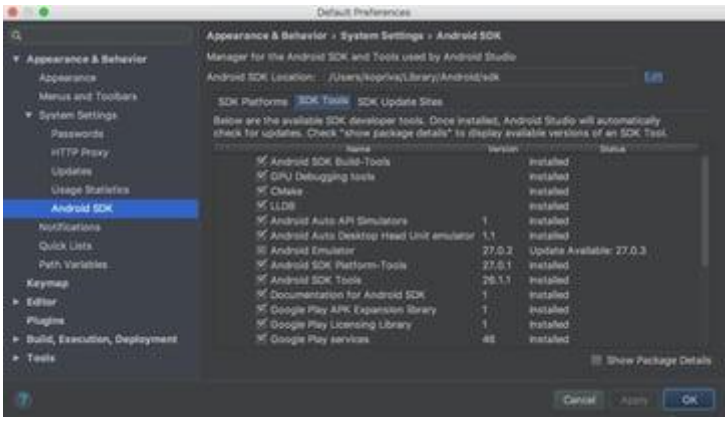


Figure 26 Android SDK

3.3. Limitations

Notre application est compatible seulement avec les Smartphones sous le système d'exploitation Android qui peuvent supporter les versions API 19 ou plus.

3.4. Description de l'application

Nous allons présenter les interfaces principales de notre application

a. L'interface principal

Cette interface présente quelques informations à l'accueil indiquant les propriétés et les fonctionnalités disponibles dans l'application, un clic sur chaque bouton pour entamer chaque option.

Pour pouvoir accéder a la caméra et s'assurer d'en avoir une sur l'appareil il faut ajouter l'autorisation dans le fichier manifest en ajoutant ces trois ligne :

```
<uses-permission
android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.CAMERA"/>
```

La fonction d'insertion d'un message F5 sur image crypté par une clé

```
binding.buttonEncode.setOnClickListener(new View.OnClickListener()
{
@Override
public void onClick(View view) {
        String key = binding.textKey.getText().toString();
String message = binding.textMessage.getText().toString();
ImageSteganographystegano = new ImageSteganography(message,
key,image);
TextEncodingtextEncoding = new
TextEncoding(EncodeFragment.this.getActivity(),
EncodeFragment.this);
textEncoding.execute(stegano);
}
});
```

Puis la fonction d'enregistrement de l'image stegano :

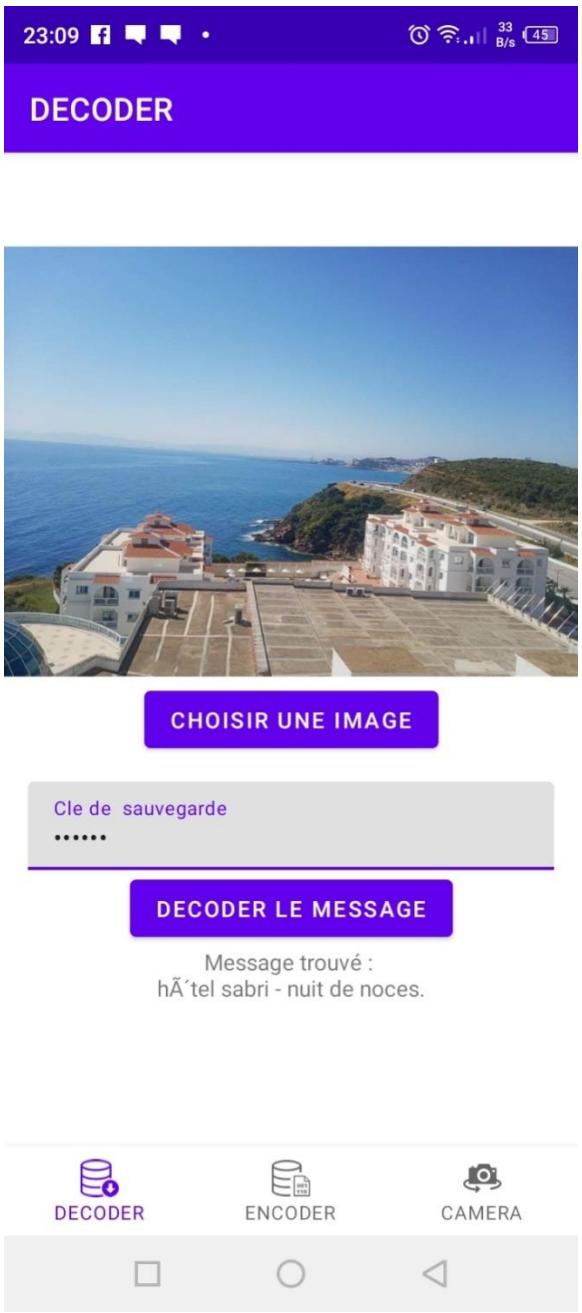
```
@Override
public void onCompleteTextEncoding(ImageSteganography result) {
if (result != null &&result.isEncoded()){

//encrypted image bitmap is extracted from result object
Bitmap img = result.getEncoded_image();

File saveFile = new
File(Environment.getExternalStoragePublicDirectory(
Environment.DIRECTORY_DOWNLOADS), "stegano" + ".PNG");

        try {
FileOutputStreamfo = new FileOutputStream(saveFile);

img.compress(Bitmap.CompressFormat.PNG, 100, fo);
//img.compress(Bitmap.CompressFormat.JPEG, 1000, fo); // saving
the Bitmap to a file compressed as a JPEG with 85% compression
rate
fo.flush(); // Not really required
fo.close();
Toast.makeText(EncodeFragment.this.getContext(),"Image
```

```
crypteenregistrer", Toast.LENGTH_LONG).show();
} catch (FileNotFoundException e) {
e.printStackTrace();
} catch (IOException e) {
e.printStackTrace();
}
    }
}
```

Cette interface sert à ajouter un message a une image sélectionné crypté par une clé prive

Apres avoir choisis une image et saisir une clé prive on pourrait l'enregistrer sur le téléphone, ainsi on pourrait l'envoyer via les réseaux sociaux ou par email

Cette interface représente le décodage d'un message à partir d'une image on introduisant la clé prive utiliser l'or de l'encodage



En cliquant sur le boutton de decodage on obtient le message (si la cle est valide) sur la même interface

On pourrait tout de même utiliser directement l'appareil photo de l'appareil pour capturer une photo puis insérer un message dessus en allant sur l'onglet « camera ».

3.5. Conclusion

Lors de ce chapitre, nous avons présenté premièrement l'objectif de notre application mobile, puis quelque choix technique de développement, encore on a présenté le système de la stéganographie des image et systèmes de cryptages.

Ensuite, nous avons présenté la conception et la construction globale de différents processus ou fonctions fournis par l'application, ainsi on a exposé quelques codes sources qui ont été utilisé durant la programmation.

Finalement, on a présenté le développement de l'application; les outils utilisés, les limitations et le mode d'utilisation.

CONCLUSION GENERALES

Ce travail de mémoire a été consacré au problème de la dissimulation d'information qui nous a permis de comprendre, tout sur l'intérêt de la stéganographie. Nous avons tout d'abord présenté une introduction à la stéganographie qui contient un historique de la stéganographie depuis son apparition, des définitions des concepts utilisés par la sténographie linguistique et technique, puis les différents éléments intervenant dans la dissimulation d'informations, ainsi que les buts et les intérêts de cette technique.Nous avons pu alors choisis deux algorithmes d'insertion (F5 et DWT) du support (soit image ou message) sur une image de couverture sur une application mobile Android. Ensuite, nous avons abordé le concept de la cryptographie qui chiffre les données pour les rendre intelligibles, mais cependant suspectes et le tatouage qui sert à insérer une marque dans un support, la stéganographie offre l'avantage de dissimuler l'information dans un support innocent, de telle sorte qu'aucune autre personne ne saurait suspecter l'existence même de l'information à l'intérieur. Puis une présentation de processus utilisés dans la stéganographie.Finalement, pour concrétiser les recherches faites sur l'application mobile destéganographie , nous avons réalisé une application qui permette de dissimuler un message crypté dans une image de couverture en utilisant deux technique d'insertion (F5 et DWT)

Perspectives

Dans nos futurs travaux plusieurs idées peuvent se concrétiser :

• Faire une étude avec d'autres supports d'information

• Etudier l'influence d'augmentation de volume des données à dissimuler.

• Améliorer l'application pour qu'elle soit plus sécurisées.

Références

• [1] https://www.merriam-webster.com/dictionary/steganography.

• [2] Fridrich, Jessica; M. Goljan; D. Soukal (2004).Delp Iii, Edward J; Wong, Ping W (éd.). "Recherche de la clé Stego" (PDF). Proc. SPIE, imagerie électronique, sécurité, stéganographie et filigrane de contenus multimédias VI. Sécurité, stéganographie et filigrane des contenus multimédias VI. **5306**: 70–82. Bibcode:2004SPIE.5306 … 70F. 23 janvier 2014.

• [3] Pahati, JO (2001-11-29). "Confondre le carnivore: comment protéger votre vie privée en ligne". AlterNet. Archivé de l'original sur 16/07/2007. Récupéré 02/09/2008.

• [4] Mazurczyk,W.,&Szczypiorski,K.(2009).Steganographyinhan-dling oversized IP packets. InProc.offirst international workshopon network steganography (IWNS 2009), November 18–20, 2009,Wuhan, China

• [5] B. Khaled, « *Approche par marquage pour l'évaluation de la qualité d'image dans les applications multimédias* », mémoire (INF6021) de maîtrise en informatique, Dép. Informatique et ingénierie, Univ. Québec en outaouais, Nov 2012

• [6] D. Martins, « *Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance* », Thèse,U.F.R. des Sciences et Techniques, Univ. Franche-Comt, 29 Nov 2010

• [7] F.M.M. Bouyé, « *Application des codes correcteurs d'erreurs en stéganographie* », Thèse, Faculté des Sciences, Univ. Mohammed V Agdal, Rabat, 11 juillet 2012.

• [8] C. Cachin, « *An information-theoretic model for steganography* », Inf. Comput., 192(1), pp.

• [9] https://fr.vikidia.org/wiki/Tablette_de_cire_romaine.

• [10] « Le tatouage d'images ou Watermarhing», Université de Nice - Sophia Antipolis Juin 2004

• [11] kartavoir.blogspot.com/2014/10/n077-le-disque-de-phaistos-env-1700-av.html

• [12] « Insertion adaptative en stéganographie ,application aux images numériques dans le domaine spatial » Sarra KOUIDER

• [13] https://digitalwatermarking.wordpress.com/2012/05/15/exemples/

• [14] « *Développement d'application mobile de stéganographie* », RAKOTOARISON Sitraka Hasinarivo, Université d'antananrivo école supérieur.

• [15] « Jenny Dentand, **Stéganographie**, travail de diplôme », Haute Ecole de Gestion de Genève (HEG-GE), 2005.

• [16] J. Simmons, «The prisioner's problem and the subliminal chanel», Plenum Press, 1983.

• [17] Kévin Hoarau «Communication stéganographiques basée sur la consommation énergétique des Smartphones », Université de la Réunion UFR Sciences et Technologies, Juin 2017.

•[18] Hugo ALATRISTA SALAS «LA STEGANOGRAPHIE MODERNE :L'ART DE LA COMMUNICATION SECRETE », Université Montpellier Sciences et Techniques du Languedoc,

juin 2010.
•[19] A. Westfeld : *High Capacity Despite Better Steganalysis (F5 - A Steganographic Algorithm)* ; In : LNCS, vol. 2137, Springer-Verlag, New York,pp. 2001, 289-302
• [20] www.nbs-system.com/blog/introduction-a-la-securite-informatique.html.(03/01/2013).
• [21] http ://www.mathgon.com/Cours/ESMISAB/CM4.pdf. (03/01/2013).
•[22]R. Watrigant. « *Utilisation des codes correcteurs d'erreurs en stéganographie : De l'algorithmeF5 et sa stéganalyse aux codes à papier mouillé* ». Rapport de projet en L3, Nîmes, France. 2009.
• [23]A. ALI-PACHA, N. HADJ-SAID, A. BELGORAF, A. M'HAMED. "Stéganographie : Sécurité par Dissimulation". Article RIST Vol, 16 n°01. 2006.
• [24] David Martins. « *Sécurité dans les réseaux de capteurs sans fils : Stéganographie et réseaux de confiance* ». Thèse de doctorat. Universite de Franche-comte. Novembre 2010.
• [25] F. HartungetM.Kutter« Multimedia watermarking techniques. Proceedings of the IEEE», 87(7) :1079-1107 . Juillet 1999.
• [26] Rémy ALLAIS, François DUMONT, Alexandre PELTIER. « *La Stéganographie* ». Projet de veille Technologique de Maîtrise informatique et réseaux, Institut Universitaire Professionnel de Blois. 23 Juin 2004.
• [27] Publié par digitalwatermarking le 15 mai 2012 Publié dans: Uncategorized.
• [28] Philippe Perret, MSI Groupe Arkoon et Serge Richard. "Cryptographie". IBM France, CLUSIR, Alpes. 28 Février 2007.
• [29] G.Labouret. "Introduction à la cryptographie". Support de cours du cabinet Hervé Schauer Consultants.09 Février 2001.
• [30] Federal information processing standards publication(fips pub 46-3). "Data Encryption Standard (DES)". U.S. Department of commerce/national institute of standards and technology. Reaffirmed 25 Octobre 1999.
• [31] Joan Daemen, Vincent Rijmen. "AES Proposal : Rijndael". Document version 2. 03 Septembre 1999.
• [32] Ronald L Rivest. "The RC Encryption Algorithm". Article .MIT Laboratory for Computer Science. 20 Mars 1997.

Résumé

La stéganographie est un mécanisme secret qui peut être utilisé pour dissimuler des informations importantes sur un réseau d'une manière qui viole la politique de sécurité et qui peut être difficile à détecter. Dans ce projet, une méthode est proposée pour le système de stéganographie par application mobile.La méthode proposée utilise Android Studio pour couvrir le message secret ; à l'aide de deux algorithmes (F5 et DWT) d'intégration/extraction spécialement conçus afin de rendre le système plus complexe pour être vaincu par les attaquants. ainsi on pourrait l'envoyer via les réseaux sociaux ou par email .

Abstract

Steganography is a covert mechanism that can be used to conceal important information on a network in a way that violates security policy and may be difficult to detect. In this project, a method is proposed for the mobile application steganography system. The proposed method uses Android Studio to cover the secret message; using two specially designed integration/extraction algorithms (F5 and DWT) in order to make the system more complex to be defeated by attackers. so we could send it via social networks or by email.

ملخـــــــــص

إخفاء المعلومات هي آلية سرية يمكن استخدامها لإخفاء المعلومات المهمةعلى شبكة بطريقة تنتهك سياسة الأمان وقد يكون من الصعب اكتشافها .في هذا المشروع ، تم اقتراح طريقة لنظام إخفاء المعلومات لتطبيقات الهاتف المحمول. تستخدم الطريقة المقترحة Android Studio لتغطية الرسالة السرية ؛ باستخدام اثنين من خوارزميات التكامل / الاستخراج المصممة خصيصًا F5) و (DWT من أجل إنشاء النظامأكثر تعقيدًا ليتم هزيمته من قبل المهاجمين. حتى نتمكن من إرسالها عبر الشبكات الاجتماعية أو عبر البريد الإلكتروني.