Université Abdelhamid Ibn Badis Mostaganem

Faculté des sciences et de la technologie

DEPARTEMENT DE GENIE ELECTRIQUE

MEMOIRE

En vue d'obtention du diplôme de MASTER ACADEMIQUE

Spécialité : Système des télécommunications

THEME

Synthèse et implémentation de techniques de stéganographie d'images pour la sécurisation de l'information

Présenté par :

BOUBEGRA Youcef

ALLALI Hadj Hocine

Soutenu le 29/06/2025

Membres du jury :

- Dr. OULDALI Abdelaziz Président
- Dr. BENCHELLAL Amel Examinatrice
- Dr. ZELLAGUI Amine Encadrant

Année universitaire 2024 / 2025

Remerciements

REMERCIEMENTS

Avant toute chose, nous tenons à adresser nos plus sincères et profonds remerciements à nos parents. Leur amour inconditionnel, leur patience, leur bienveillance et leur soutien indéfectible ont été le socle de notre réussite. Ils ont toujours cru en nous, même dans les moments de doute, et nous ont transmis les valeurs de persévérance, d'honnêteté et de travail bien fait. Leur confiance, leurs encouragements constants et leur présence rassurante ont été pour nous une source inestimable de motivation.

Nous leur sommes infiniment reconnaissants pour les sacrifices qu'ils ont consentis afin de nous permettre de poursuivre nos études dans les meilleures conditions. Sans leur appui moral, matériel et affectif, ce mémoire, et plus largement ce parcours, n'auraient pas été possibles. Ce travail est en grande partie le fruit de leur dévouement et de leur amour. Merci du fond du cœur.

Nous remercions chaleureusement notre encadrant, Monsieur Amine Zellagui, pour sa disponibilité, sa patience, ses conseils avisés et son accompagnement tout au long de l'élaboration de ce mémoire. Sa rigueur scientifique et ses remarques pertinentes ont été d'une grande richesse.

Nous souhaitons également remercier l'ensemble des enseignants et membres du corps pédagogique de l'Université des Sciences et Technologies Abdelhamid Ibn Badis de Mostaganem pour la qualité de l'enseignement dispensé et pour leur accompagnement tout au long de notre formation.

Nous n'oublions pas nos amis et camarades de promotion, pour leur soutien, leur bonne humeur et les nombreux moments partagés, qui ont rendu cette expérience encore plus agréable.

Enfin, merci à toutes les personnes, de près ou de loin, qui ont contribué à la réalisation de ce mémoire.

Résumé

الملخص

الستيغانوغرافيا هي تقنية في أمن المعلومات تهدف إلى إخفاء بيانات سرية داخل وسائط رقمية مثل الصور، بطريقة لا يمكن إدراكها بالعين المجردة. يقدم هذا البحث نظامًا هجينًا للإخفاء يعتمد على ضغط هوفمان، وتشفيرRC4 ، واستخدام تسلسلات فوضوية (خريطة هينون والخريطة اللوجيستية) لتوجيه إدراج البيانات في البتات الأقل أهمية (LSB) في صورة ملونة .(RGB) يسمح النظام أيضًا بتخزين قاموس هوفمان في طبقة أخرى من الصورة مما يجعل الاستخراج تلقائيًا دون معلومات مسبقة. تم تقييم الأداء باستخدام مؤشرات بصرية وكمية مثل PSNR وSSIM و MSE وMAE ، وأظهرت النتائج مستوى عالٍ من عدم القابلية للكشف والصلابة ضد الهجمات.

Résumé

La stéganographie est une technique de sécurité de l'information visant à dissimuler des données secrètes dans des supports numériques, tels que les images, de manière imperceptible à l'œil humain. Ce mémoire présente un système hybride de stéganographie basé sur la compression Huffman, le chiffrement RC4, et l'utilisation de séquences chaotiques (Henon et logistique) pour diriger l'insertion des données dans les bits de poids faible (LSB) d'une image RGB. Le système proposé permet également de stocker le dictionnaire de décompression dans une autre couche de l'image, rendant l'extraction autonome. Les performances ont été évaluées à l'aide d'indicateurs visuels et quantitatifs (PSNR, SSIM, MSE, MAE, etc.), démontrant un haut niveau d'imperceptibilité et de robustesse face aux attaques.

Abstract

Steganographyis an information security technique used to conceal secret data within digital media such as images, in a waythatis imperceptible to the humaneye. This thesispresents a hybridsteganographic system based on Huffman compression, RC4 encryption, and chaoticsequences (Henon and logisticmaps) to guide the insertion of data into the least significant bits (LSB) of an RGB image. The proposed system alsoallows the Huffmandecodingdictionary to beembedded in a different image layer, enablingfullyautonomous extraction. The performance wasevaluatedusingvisual and quantitative metrics (PSNR, SSIM, MSE, MAE, etc.), showing high imperceptibility and robustnessagainstattacks.

Table des matières

Table des matières

Liste des figures

Liste des figures

Liste des

Tableaux

Liste des Tableaux

Introduction Générale

Introduction Générale

À l'ère du numérique, l'information constitue une ressource stratégique et précieuse. Elle circule en grande quantité à travers les réseaux informatiques, et sa sécurité est devenue un enjeu majeur dans des domaines aussi variés que la défense, la santé, la finance ou encore la communication personnelle. Dans ce contexte, la protection des données sensibles est une priorité absolue, nécessitant le recours à des techniques avancées garantissant à la fois la confidentialité, l'intégrité et l'authenticité de l'information.

Parmi ces techniques, la stéganographie se distingue par son approche subtile : elle ne se contente pas de rendre le message illisible, comme le fait la cryptographie, mais vise à cacher l'existence même du message. Ainsi, elle permet de dissimuler une information secrète au sein d'un contenu numérique anodin, tel qu'une image, une vidéo ou un fichier audio, sans éveiller de soupçons. Cette capacité à "cacher dans la transparence" rend la stéganographie particulièrement pertinente dans les environnements à risque, où toute trace de communication confidentielle peut être suspecte.

Le présent mémoire s'inscrit dans cette problématique. Il vise à étudier en profondeur les techniques de stéganographie appliquées aux images numériques, tant sur le plan théorique que pratique. L'objectif est double : comprendre les fondements scientifiques de ces méthodes, puis concevoir et implémenter un système efficace et sécurisé de dissimulation d'information. Pour cela, nous explorerons les principes des domaines spatial et fréquentiel, ainsi que leur combinaison avec d'autres mécanismes de sécurité tels que la cryptographie et la compression.

Ce travail s'articule autour de trois grands axes. Le premier chapitre présente les bases du traitement d'image numérique et les enjeux liés à la sécurité des images. Le deuxième chapitre est consacré à une étude approfondie des différentes techniques de stéganographie, leurs classifications, avantages et limites. Enfin, le troisième chapitre décrit l'implémentation d'un système hybride combinant stéganographie, cryptographie et compression, suivie d'une évaluation expérimentale des résultats obtenus.

À travers cette démarche, nous cherchons à contribuer au renforcement de la sécurité de l'information dans les environnements numériques, en montrant comment des approches combinées peuvent offrir des solutions à la fois discrètes, robustes et efficaces

Chapitre 1 :

Introduction aux images numérique

1 Chapitre 1 : Introduction aux images numérique

1.1 Introduction :

Avec l'évolution des technologies, les images numériques sont devenues un vecteur essentiel d'information, de communication et d'analyse dans de nombreux domaines. Ce chapitre explore les fondements des images numériques, en abordant leurs applications, classifications, caractéristiques physiques et représentation en couleur. Il présente également les principaux formats d'image et les méthodes de traitement utilisées pour améliorer ou analyser leur contenu. Enfin, une attention particulière est portée aux aspects de sécurité des images, incluant la cryptographie, le watermarking et la stéganographie.

1.2 Images numériques :

Une image numérique est la représentation d'une image bidimensionnelle sous forme d'un ensemble fini de valeurs numériques, appelées pixels. Chaque pixel possède une valeur qui détermine sa couleur et sa luminosité, et l'ensemble de ces pixels constitue l'image complète (1).

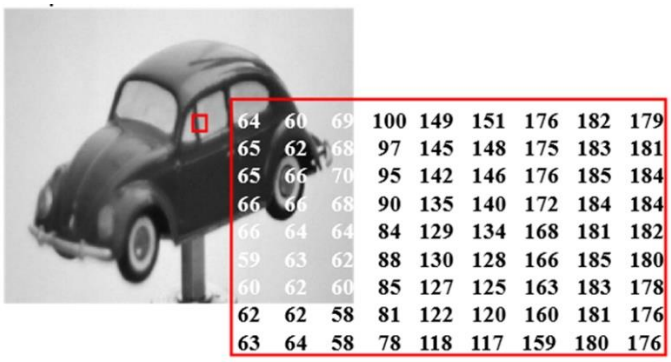

Figure 1-1 Image numérique

Les images numériques sont à la base de nombreuses technologies modernes. Elles jouent un rôle fondamental dans des domaines variés, allant de la photographie et de l'imagerie médicale à la télédétection et à la vision par ordinateur.

Applications des images numériques

a) Photographie

La photographie numérique a largement supplanté la photographie argentique traditionnelle. Les appareils photo numériques permettent :

– une visualisation immédiate des clichés,

– une édition facile via des logiciels de traitement d'image,

– un partage instantané sur Internet ou les réseaux sociaux.

Les photographes professionnels utilisent des appareils haute résolution couplés à des logiciels spécialisés pour capturer, corriger et améliorer leurs images.

b) Imagerie médicale

Les images numériques sont essentielles dans le domaine médical pour le diagnostic et le suivi des patients. Des techniques telles que :

– les radiographies (rayons X),

– l'IRM (imagerie par résonance magnétique),

– le scanner (tomodensitométrie – TDM),

Génèrent des images permettant d'examiner l'intérieur du corps humain sans intervention chirurgicale. Ces images peuvent être améliorées, filtrées ou analysées automatiquement pour aider au diagnostic et à la planification des traitements.

Figure 1-2 Images médicaux (rayon x)

c) Télédétection

Les satellites et drones capturent des images numériques de la surface terrestre à des fins diverses :

– agriculture de précision,

– surveillance environnementale (forêts, eau, pollution),

– planification urbaine.

Ces images sont utilisées pour la cartographie, le suivi de l'évolution des milieux naturels et la prise de décisions stratégiques.

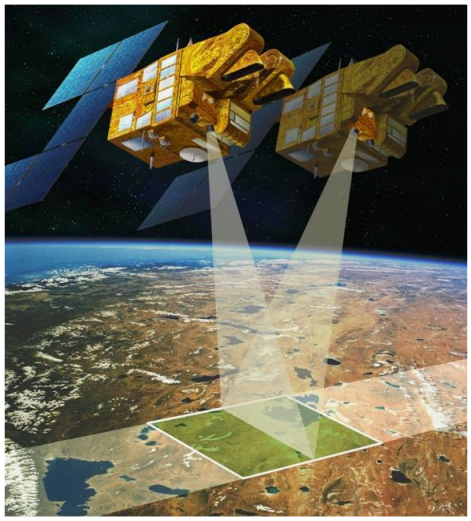

Figure 1-3 Satellite

d) Vision par ordinateur

La vision par ordinateur est un domaine de l'intelligence artificielle permettant aux machines d'interpréter et de comprendre des images numériques. Elle est utilisée dans :

– la reconnaissance faciale,

– les véhicules autonomes,

– les moteurs de recherche d'images.

Les techniques courantes incluent la détection d'objets, la segmentation d'images et la classification automatique (1).

e) Divertissement

Les images numériques sont omniprésentes dans l'industrie du divertissement :

– jeux vidéo,

– films et effets spéciaux,

– réalité virtuelle (VR) et réalité augmentée (AR).

La création d'animations 2D/3D, d'effets visuels et de modélisations virtuelles repose fortement sur les technologies d'imagerie numérique.

1.3 Types d'images numériques

Les images numériques se divisent principalement en deux grandes catégories : les images matricielles (ou raster) et les images vectorielles. Comprendre leurs différences est essentiel pour choisir le type le plus adapté à une application donnée.

1.3.1 Images matricielles (Raster)

Les images matricielles sont composées d'une grille de pixels, chaque pixel représentant une petite unité de couleur et de luminosité. Ce type d'image est bien adapté aux photographies ou aux visuels riches en détails.

Caractéristiques :

- Chaque pixel est stocké avec des informations de couleur (souvent sur plusieurs bits).
- Plus l'image contient de pixels, plus la résolution est élevée.
- L'agrandissement entraîne une pixellisation (perte de qualité).
- Elles occupent souvent une grande taille mémoire, surtout sans compression.

Formats courants :

- BMP, .TIF, .GIF, .JPG, .PNG

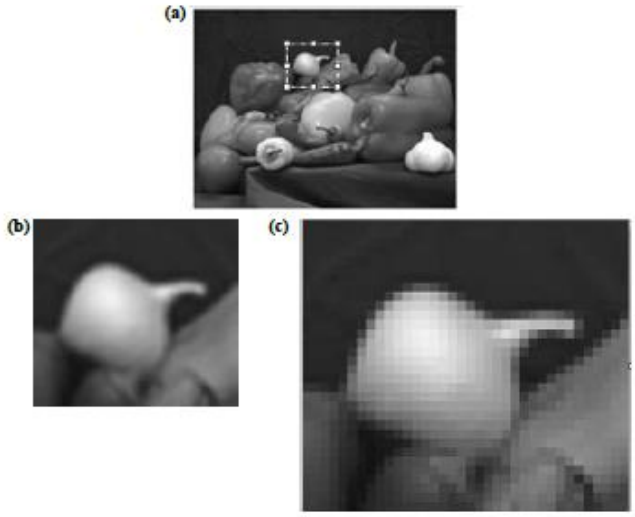

Figure 1-4 Image matricielle

Exemple : Une photo de 640 × 480 pixels contient 307 200 pixels, tandis qu'une image 3072 × 2048 (≈6,3 mégapixels) contient plus de 6 millions de pixels. Des algorithmes de compression comme JPEG ou GIF sont utilisés pour réduire la taille de ces fichiers.

1.3.2 Images vectorielles:

Les images vectorielles ne stockent pas les pixels, mais une série d'instructions mathématiques qui décrivent des formes géométriques : lignes, cercles, courbes, etc. Cela permet de redimensionner l'image à l'infini sans perte de qualité, ce qui les rend idéales pour les logos, icônes, illustrations ou plans techniques.

Figure 1-5 Image vectorielle

Caractéristiques

- Basées sur des vecteurs (formules mathématiques).
- Résolution indépendante : parfaites pour l'impression.
- Taille de fichier généralement plus faible que les images matricielles.
- Moins adaptées pour les détails complexes ou les dégradés riches (comme les photos).

Formats courants :

- .SVG, .EPS, .PDF, .AI, .DXF

1.3.3 Conversions entre images vectorielles et matricielles

A. Vectoriel vers matriciel

Les écrans et imprimantes fonctionnent selon un affichage matriciel. Ainsi, pour pouvoir afficher ou imprimer une image vectorielle, elle doit être convertie en format matriciel.

- La conversion consiste à rastériser l'image vectorielle, en fonction d'une résolution choisie.
- La taille de l'image vectorielle d'origine ne change pas, mais le fichier matriciel généré peut être lourd si la résolution est élevée.

- Cette conversion est simple et courante dans les logiciels de dessin ou d'édition graphique.

B. Matriciel vers vectoriel :

La conversion inverse s'appelle la vectorisation ou image tracing. Elle est plus complexe et implique une analyse automatique ou manuelle des formes présentes dans l'image.

- Elle est souvent utilisée pour restaurer ou simplifier des images anciennes.
- Toutefois, le résultat est souvent approximatif, surtout si l'image contient beaucoup de détails ou de bruit visuel.
- Par exemple, un dessin bitmap dans Paint (comme une ligne irrégulière) peut être converti en tracé vectoriel, mais la qualité finale est rarement identique à l'original.

1.4 Caractéristiques d'une image numérique

Une image numérique possède plusieurs caractéristiques qui permettent de la représenter, de l'analyser et de la manipuler informatiquement. Ces caractéristiques découlent directement du processus de numérisation.

1.4.1 Échantillonnage :

L'échantillonnage est la première étape de la conversion d'une image analogique (continue) en image numérique (discrète). Il consiste à mesurer l'intensité lumineuse à des intervalles réguliers sur l'image, créant ainsi une grille de pixels (2).

Principe :

- Une grille est superposée à l'image analogique.
- À chaque intersection, un pixel est défini, représentant une zone de l'image par une valeur d'intensité.

Impact du taux d'échantillonnage :

- Taux élevé : Plus de pixels → meilleure résolution et plus de détails (ex. : appareil photo moderne).
- Taux faible : Moins de pixels → moins de détails, image floue ou pixelisée (ex. : caméra VGA ancienne).

1.4.2 Quantification:

Après l'échantillonnage, la quantification consiste à convertir les valeurs d'intensité continues obtenues en valeurs numériques discrètes. C'est une forme de simplification contrôlée, nécessaire pour le stockage informatique.

Étapes :

- Division de la plage d'intensité en intervalles (niveaux de quantification).
- Affectation de chaque valeur mesurée à l'intervalle le plus proche.
- Remplacement des valeurs d'origine par les valeurs discrètes correspondantes.

Exemples :

- 8 bits par pixel → 256 niveaux → nuances de gris détaillées.
- bits par pixel → 4 niveaux → image très simplifiée, effet "postérisé".

1.4.3 La "définition" d'une image :

On appelle définition le nombre de pixels constituant l'image, c'est-à-dire sa "dimension informatique" (le nombre de colonnes de l'image que multiplie son nombre de lignes).

Une image possédant 640 pixels en largeur et 480 en hauteur aura une définition de 640 pixels par 480, notée « 640x480 ».

1.4.4 La "Résolution" d'une image :

La résolution, terme souvent confondu avec la "définition", détermine par contre le nombre de points par unité de surface, exprimé en points par pouce (PPP, en anglais DPI pour Dots Per Inch); un pouce représentant 2.54 cm.

La résolution permet ainsi d'établir le rapport entre le nombre de pixels d'une image et la taille réelle de sa représentation sur un support physique. Une résolution de 300 dpi signifie donc 300 colonnes et 300 rangées de pixels sur un pouce carré ce qui donne donc 90 000 pixels sur un pouce carré (3) (4).

1.4.5 Dynamique :

Pour une image en niveaux de gris, la dynamique représente le nombre de niveaux de gris présents dans celle-ci. Pour une image couleur, nous parlons de dynamique par canal.

Nous pouvons aussi exprimer la dynamique par le nombre de bits utilisé pour coder le niveau de gris.

1.5 Couleurs des images numériques :

Les couleurs dans une image numérique sont représentées à l'aide de modèles de couleur mathématiques qui traduisent la perception humaine de la couleur en valeurs numériques. Chaque pixel d'une image couleur contient des informations codées selon l'un de ces modèles.

1.5.1 Image binaire :

Les images binaires représentent la forme la plus simple d'image numérique. Elles sont très courantes dans le domaine du traitement d'image, en particulier pour des tâches telles que la segmentation, la reconnaissance de formes ou encore la détection d'objets.

Une image binaire ne contient que deux couleurs : le noir et le blanc.

- Le noir (valeur 0) représente généralement l'arrière-plan.
- Le blanc (valeur 1) représente les objets ou régions d'intérêt.

Chaque pixel de l'image peut donc prendre uniquement deux valeurs : 0 ou 1.

Figure 1-6 Image binaire.

La binarisation :

Pour obtenir une image binaire à partir d'une image en niveaux de gris ou couleur, on applique un seuillage (binarisation) :

Étapes de la binarisation :

- Conversion en niveaux de gris si l'image est en couleur (RVB).
- Choix d'un seuil (T) : une valeur entre 0 et 255.
- Comparaison pixel par pixel :
  - Si la valeur du pixel est inférieure au seuil, il devient 0 (noir).
  - Si elle est supérieure ou égale au seuil, il devient 1 (blanc).

Le résultat est une image binaire simplifiée, mettant en évidence les régions importantes.

1.5.2 Image en niveau de gris :

Une image en niveaux de gris est une image numérique dans laquelle chaque pixel représente une intensité lumineuse sur une échelle de gris, sans aucune information de couleur.

Chaque pixel prend une valeur d'intensité entre :

- 0 représente le noir,
- 255 représente le blanc,

Avectoutes les nuances de grisintermédiairescomprises entre ces deux extrêmes**.**

La figure ci-dessous montre une image en niveaux de gris, où les différentes intensités lumineuses sont représentées par des nuances allant du noir au blanc.

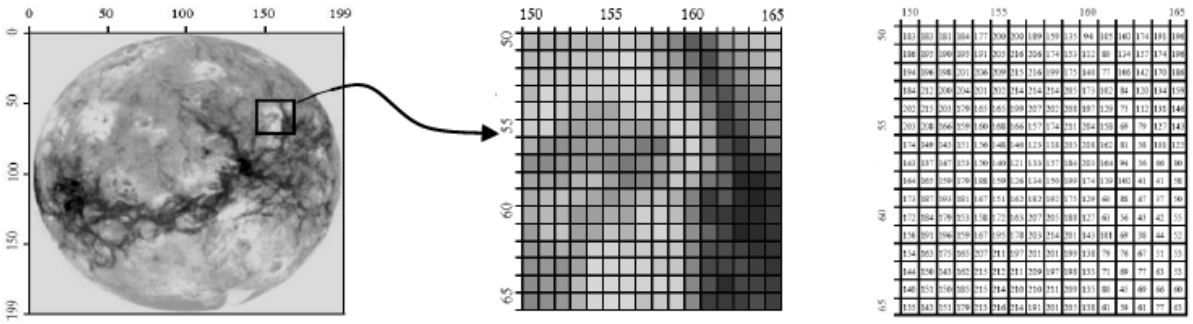

Figure 1-7 Image en niveau de gris

Les images en niveaux de gris sont très utilisées en traitement d'image (filtrage, détection de contours...), en vision par ordinateur ou dans des applications où la couleur n'est pas nécessaire, permettant de réduire la complexité de traitement et d'économiser de la mémoire (5).

Profondeur de bits (bit depth) :

La profondeur de bits détermine le nombre de niveaux de gris possibles. Plus elle est élevée, plus l'image peut représenter finement les variations d'intensité lumineuse.

Tableau 1 Profondeur de bits

Profondeur	Niveaux de gris	Applications typiques
8 bits	256 niveaux	Standard pour la plupart des applications
16 bits	65 536	Imagerie médicale, photographie de haute qualité
Virgule flottante	Illimité (théorique)	Imagerie HDR, calcul scientifique avec haute précision

1.5.3 Image couleur :

Imagerie HDR, calcul scientifique avec haute précision Le traitement d'image couleur repose sur la représentation numérique des informations colorimétriques, ce qui influe directement sur la qualité visuelle, la taille des fichiers et la complexité du traitement. Deux principaux formats sont utilisés : le format 8 bits par canal (24 bits par pixel) et le format 16 bits par canal (48 bits par pixel), chacun adapté à des contextes d'utilisation spécifiques.

1.5.3.1 Format couleur 8 bits

Dans ce format, chaque pixel est composé de trois composantes – rouge, vert et bleu (RVB) – codées chacune sur 8 bits. Cela permet une représentation de 256 niveaux par canal, soit environ 16,7 millions de couleurs possibles (256^3). Ce format est souvent désigné sous l'appellation « truecolor »

| *R (8 bits)* | *G (8 bits)* | *B (8 bits)* | → Total : **24 bits/pixel**

Il existe également le format à couleur indexée, où l'image utilise une palette de 256 couleurs, chaque pixel étant une référence à une couleur de cette table. Ce format est plus léger mais moins précis.

L'image en couleur ci-dessous illustre la répartition des composantes RVB dans un format 8 bits par canal, permettant une visualisation fidèle des teintes de la scène représentée.

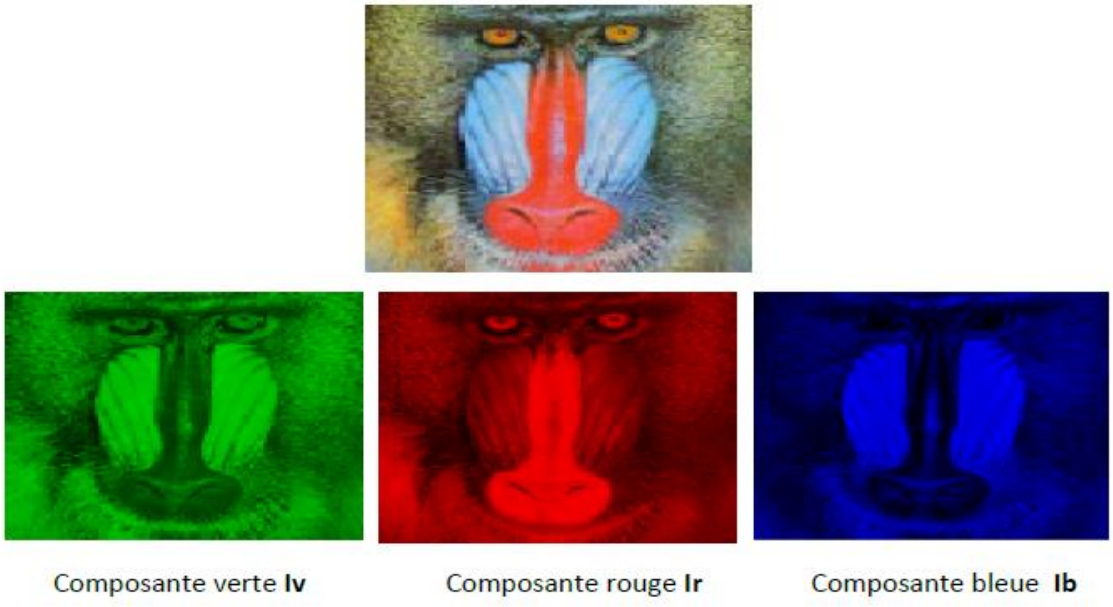

Figure 1-8 Image en couleur RGB

- **Applications du format 8 bits**:

— Conception web et affichage rapide sur les navigateurs ;
— Services de streaming vidéo classiques (ex. YouTube) ;
— Retouche photo de base (Photoshop Elements, GIMP) ;
— Jeux vidéo grand public (ex. *Assassin'sCreed IV*) ;
— Affichage sur téléviseurs et écrans standard.

1.5.3.2 Format couleur 16 bits

Dans un format 16 bits, chaque canal RVB est codé sur 16 bits, offrant 65 536 niveaux d'intensité par canal, soit 281 mille milliards de couleurs possibles ($65\,536^3$). Ce format permet une plage dynamique plus étendue et des transitions de couleurs plus précises, idéales pour les applications exigeantes (6).

| *R (16 bits)* | *G (16 bits)* | *B (16 bits)* | → Total :**48 bits/pixel**

- **Applications du format 16 bits**:

— Photographie professionnelle (modèles, publicités) ;
— Cinéma et production vidéo 4K ou HDR ;
— Imagerie médicale (scanners, IRM) ;
— Visualisation scientifique (simulations, mesures précises) ;
— Archivage d'images sans perte de qualité.

1.6 Les formats des images :

Les formats d'image désignent les types de fichiers utilisés pour enregistrer et diffuser des photographies, illustrations ou graphiques numériques. Le choix du bon format est crucial, car il affecte la qualité visuelle, la taille du fichier, le temps de chargement ainsi que la compatibilité avec les différents supports (web, impression, réseaux sociaux, etc.).

Parmi les formats les plus répandus, on trouve JPEG, PNG, GIF ou encore SVG.

- Le JPEG est privilégié pour les photographies grâce à son excellent compromis entre qualité et compression.
- Le PNG est adapté aux images nécessitant de la transparence.
- Le GIF est utilisé pour les animations simples.
- Le SVG est idéal pour les illustrations vectorielles, évolutives sans perte de qualité.

Bien connaître les spécificités de chaque format permet d'optimiser à la fois le rendu visuel et les performances techniques.

1.6.1 TIFF (.tif, .tiff)

Tagged Image File Format : format sans perte, utilisé pour stocker des images de très haute qualité. Ne compressant pas les données, les fichiers TIFF sont volumineux, mais parfaits pour l'impression professionnelle, les archives ou la publication.

1.6.2 JPEG (.jpg, .jpeg)

Joint Photographic Experts Group : format avec perte, utilisé massivement pour les photographies numériques. La compression réduit la taille du fichier au détriment d'une légère perte de qualité. Il reste le format de référence pour :

- les appareils photo numériques,
- le partage sur Internet,
- les présentations (PPT),
- les impressions non professionnelles.

1.6.3 GIF (.gif)

Graphics Interchange Format : format limité à 256 couleurs, utilisé pour les graphiques simples et les animations. Il prend en charge la transparence, mais de manière rudimentaire.

Avantages : poids léger, compatibilité web universelle.

1.6.4 PNG (.png)

Portable Network Graphics : format sans perte, conçu pour remplacer le GIF. Supporte la transparence avancée et peut représenter jusqu'à 16 millions de couleurs. Idéal pour :

- Les icônes,
- Les interfaces graphiques,
- Les images nécessitant un fond transparent.

1.6.5 WebP

Développé par Google, ce format combine les avantages du JPEG et du PNG. Il permet une compression efficace (avec ou sans perte) pour un rendu de haute qualité tout en réduisant la taille des fichiers, optimisant ainsi le chargement des pages web.

1.6.6 HEIF (High Efficiency Image File Format)

Standardisé par MPEG, ce format moderne repose sur le codec HEVC (High EfficiencyVideoCoding). Il permet de stocker des images uniques ou des séquences tout en conservant une grande efficacité de compression, avec une qualité visuelle supérieure au JPEG.

1.6.7 AVIF (AV1 Image File Format)

Basé sur le codec vidéo AV1, ce format libre de droits offre une compression très performante et une qualité d'image élevée. Il repose également sur le conteneur HEIF et devient progressivement un nouveau standard pour le web, grâce à son efficacité.

1.6.8 Bitmap (.bmp)

Bitmap, format développé par Microsoft, est similaire au TIFF : sans perte, sans compression. Toutefois, il est peu utilisé aujourd'hui en raison de son poids élevé et de son manque de portabilité. Le TIFF est souvent préféré dans les usages professionnels.

1.6.9 Fichiers RAW (.raw, .cr2, .nef, .orf, .sr2)

Fichiers bruts générés directement par les capteurs d'appareils photo (DSLR, hybrides). Chaque constructeur propose ses variantes (.cr2 pour Canon, .nef pour Nikon, etc.).

- Ces fichiers contiennent toutes les données capturées, sans traitement ni compression.
- Ils permettent une retouche précise dans des logiciels spécialisés (Photoshop, Lightroom).
- Utilisés exclusivement dans la photographie professionnelle.

1.7 Images 2D et 3D

Les images peuvent être classées selon leur dimensionnalité : 2D (deux dimensions) ou 3D (trois dimensions).

Figure 1-9 Image 2D/3D

1.7.1 Images 2D

Les images 2D (deux dimensions) ont une longueur et une largeur. Elles sont plates et peuvent être dessinées ou affichées sur une surface plane (comme une feuille de papier ou un écran) sans masquer de parties.

Exemples : un cercle, un carré, une photographie, une illustration, etc.

Ce sont les images les plus courantes, que nous voyons quotidiennement sur nos smartphones, ordinateurs, ou documents imprimés.

1.7.2 Images 3D

Les images 3D (trois dimensions) ont une longueur, une largeur et une profondeur (ou hauteur). Elles représentent des objets dans l'espace et nécessitent souvent un logiciel de modélisation 3D (comme Blender, SolidWorks, etc.) pour être créées.

Bien que les images 3D soient affichées sur des supports 2D (écrans, papier), elles simulent la profondeur par des techniques de perspective, ombrage ou à l'aide de dispositifs optiques (comme les lunettes 3D) pour donner l'illusion de relief.

Remarque : Nos yeux captent en réalité deux images 2D légèrement différentes (grâce à la distance entre les yeux). C'est notre cerveau qui reconstitue la profondeur et crée la perception 3D. La rétine, comme un écran, ne capte que des images planes.

1.8 Traitement d'image :

Le traitement d'image désigne un ensemble de techniques utilisées pour modifier, améliorer ou analyser des images numériques. L'objectif peut être d'en améliorer la qualité visuelle, d'extraire des informations pertinentes (formes, objets, textures…), ou encore de transformer l'image vers un autre format utile pour l'analyse ou la visualisation.

Il s'applique aussi bien aux photographies qu'aux images extraites de vidéos et repose sur des algorithmes variés capables d'effectuer des opérations simples (comme le filtrage ou l'ajustement de contraste) ou complexes (comme la segmentation, la détection de motifs, ou la reconnaissance d'objets) (7).

En général, le processus de traitement d'image suit ces étapes :

- Acquisition de l'image.
- Prétraitement (filtrage, correction, mise à l'échelle…).
- Traitement principal (segmentation, détection, analyse…).
- Sortie ou exploitation des résultats.

1.8.1 Types de traitement d'image

1.8.1.1 Traitement d'image analogique

Le traitement d'image analogique concerne les opérations effectuées sur des images dans leur format non numérique, comme :

- Les photographies argentiques,
- Les clichés sur pellicule,
- Les images imprimées.

Ce traitement repose sur des procédés physiques ou chimiques (ex. : développement en chambre noire, retouche manuelle sur film, etc.).

Bien qu'il soit aujourd'hui moins courant, le traitement analogique conserve une valeur patrimoniale et reste utilisé dans certains domaines artistiques ou scientifiques spécifiques.

1.8.1.2 Traitement d'image numérique

Le traitement d'image numérique s'effectue sur des images converties en données numériques, c'est-à-dire composées de pixels.Chaque pixel a une valeur qui peut être modifiée par des algorithmes informatiques afin de transformer l'image selon des objectifs définis (amélioration, détection, compression, etc.).

Ce type de traitement est plus flexible, plus rapide, plus précis, et largement automatisable grâce aux logiciels modernes.

Il est devenu incontournable dans des domaines comme : la médecine (imagerie médicale), la surveillance (vidéosurveillance, détection de mouvement), les réseaux sociaux (filtres photo, reconnaissance faciale), ou encore l'industrie (contrôle qualité par vision artificielle).

1.8.2 Techniques de traitement d'image

Le traitement d'image numérique fait appel à un ensemble de techniques spécialisées visant à améliorer, restaurer, analyser ou compresser des images. Ces techniques sont adaptées à des objectifs précis selon les besoins de l'application.

1.8.2.1 Amélioration d'image (Image Enhancement)

L'objectif de l'amélioration d'image est de rendre une image plus claire, plus nette ou plus lisible sans en modifier le contenu fondamental.

A. Ajustement du contraste

Cette technique améliore la visibilité des détails en accentuant les différences entre les zones claires et sombres. L'étirement de l'histogramme (ou "stretching") est souvent utilisé pour étendre la gamme des intensités sur toute l'échelle disponible (0 à 255 pour une image en niveaux de gris).

B. Égalisation d'histogramme

Elle consiste à redistribuer les niveaux de gris d'une image pour obtenir un histogramme plus uniforme, ce qui améliore le contraste global, notamment lorsque les détails sont peu visibles à cause d'une plage dynamique réduite.

C. Réduction du bruit

Le bruit est constitué de variations aléatoires indésirables de l'intensité. Des filtres tels que :

- Filtre médian (efficace pour le bruit impulsionnel),
- Filtre gaussien (pour un lissage général),
- Filtre bilatéral (conserve les contours),

Permettent de réduire le bruit tout en préservant les détails de l'image (8).

1.8.2.2 Restauration d'image (Image Restoration)

La restauration vise à reconstruire ou corriger une image dégradée par des facteurs physiques comme le flou, le bruit ou la perte de données.

A. Défloutage (Deblurring)

Le défloutage restaure la netteté d'une image affectée par un mouvement de la caméra ou une mise au point incorrecte. On utilise des techniques telles que : filtrage inverse et filtre de Wiener, pour reconstituer une approximation fidèle de l'image originale.

B. Retouche (Inpainting)

Cette méthode permet de remplir ou reconstruire des zones manquantes ou détériorées dans une image. Elle est utilisée dans la restauration d'anciennes photos, ou pour supprimer des objets indésirables. On distingue :les approches par patchs (ex. : exemplar-basedinpainting),et celles basées sur les équations différentielles partielles (EDP) (9).

C. Dénoyautage (Denoising)

Le dénoyautage cherche à éliminer le bruit tout en préservant les détails. Parmi les méthodes efficaces :

- Ondelette (Waveletthresholding),
- Moyennes non locales (Non-Local Means),

On favorise une réduction adaptative du bruit, avec une meilleure conservation des bords.

1.8.2.3 Segmentation d'image (Image Segmentation)

La segmentation consiste à diviser une image en régions significatives, facilitant ainsi l'analyse et l'interprétation du contenu.

a. Seuillage

Méthode simple pour créer une image binaire, où un seuil fixe ou adaptatif est appliqué pour séparer les objets du fond. Tous les pixels supérieurs au seuil sont mis à 1 (blanc), les autres à 0 (noir).

b. Détection de contours

Elle permet d'identifier les frontières entre différentes régions de l'image. Les opérateurs classiques sont : Sobel, Prewitt, Canny (plus précis et robuste au bruit)
Ces techniques s'appuient sur des variations abruptes de l'intensité.

c. Segmentation basée sur les régions

Cette approche consiste à regrouper les pixels similaires par croissance de régions (expansion à partir de pixels de départ homogènes), ou par lignes de partage des eaux (watershed), qui modélisent l'image comme une topographie et segmentent selon les lignes de crête.

1.8.2.4 Compression d'image

La compression réduit la taille des fichiers image pour optimiser le stockage et la transmission, tout en conservant une qualité acceptable.

a) Compression avec perte (Lossy)

Elle supprime de manière irréversible certaines informations, souvent invisibles à l'œil nu. La méthode JPEG, très répandue, offre un bon compromis entre qualité visuelle et réduction de taille.

b) Compression sans perte (Lossless)

Aucune information n'est perdue, ce qui permet une reconstruction exacte de l'image originale. Des formats comme PNG ou TIFF (avec compression LZW) sont utilisés quand la fidélité des données est cruciale (imagerie médicale, documents, etc.).

1.8.2.5 Extraction de caractéristiques (Feature Extraction)

L'extraction de caractéristiques est une étape clé en traitement d'image, qui vise à extraire des informations pertinentes permettant de représenter, analyser ou reconnaître les objets présents dans une image. Ces caractéristiques peuvent concerner la forme, la texture, la couleur, ou des motifs visuels.

A. Analyse de forme et de texture

Cette analyse permet de décrire la structure géométrique (forme) ainsi que l'apparence de surface (texture) des objets.

- Analyse de forme : basée sur des mesures comme la surface, le périmètre, la compacité, l'excentricité, elle permet de reconnaître les contours d'objets, détecter des formes géométriques (cercles, rectangles, etc.) ou différencier des structures selon leur configuration spatiale.
- Analyse de texture : la texture décrit la variation d'intensité ou de motif dans une région. Des filtres tels que ceux de Gabor, LoG (Laplacian of Gaussian) ou des matrices de cooccurrence (GLCM – Gray LevelCo-occurrence Matrix) sont utilisés pour quantifier la texture selon des critères comme l'homogénéité, le contraste ou l'entropie.

B. Détection de couleur

Les techniques de détection de couleur comprennent le seuillage colorimétrique, qui permet d'isoler des objets en définissant des plages de couleur spécifiques, comme par exemple la détection d'un objet rouge dans une scène. Elles incluent également l'utilisation des histogrammes de couleur, qui décrivent la répartition des couleurs dans une image ou une région donnée, ce qui les rend particulièrement utiles pour la comparaison d'images ou le suivi d'objets (10).

C. Reconnaissance de motifs

La reconnaissance de motifs vise à classer des régions ou objets extraits d'une image en différentes catégories prédéfinies, à partir des caractéristiques extraites précédemment.

Cette étape implique des algorithmes d'apprentissage automatique tels que :

- Réseaux de neurones (ANN, CNN) pour l'identification d'objets complexes ou en contexte,
- SVM (Support Vector Machine), efficace pour des séparations linéaires ou non linéaires entre classes,
- Appariement de modèles (templatematching), méthode plus simple basée sur la corrélation entre une région de l'image et un modèle connu.

Ces techniques sont largement utilisées dans des applications telles que la reconnaissance faciale, la lecture automatique de plaques d'immatriculation (LAPI), ou encore la détection d'objets dans des vidéos de surveillance.

1.9 La sécurité des images numériques :

Les images numériques représentent une part croissante des données échangées à travers les réseaux, que ce soit dans la télémédecine, la vidéosurveillance, les réseaux sociaux, la cartographie satellitaire ou encore le commerce électronique. Elles véhiculent souvent des informations sensibles, personnelles, médicales ou stratégiques, dont la protection est devenue une nécessité absolue.

Contrairement aux données textuelles, les images possèdent une forte redondance spatiale, des structures visuelles complexes et des caractéristiques perceptuelles spécifiques. Ces particularités rendent leur sécurisation non seulement indispensable, mais aussi techniquement plus délicate.

L'objectif de la sécurité des images numériques est de garantir les principaux piliers de la cybersécurité :

- La confidentialité, en empêchant l'accès non autorisé aux données visuelles.
- L'intégrité, en assurant que l'image n'a pas été altérée ou falsifiée pendant le stockage ou la transmission.
- L'authentification, en permettant de vérifier l'identité de l'émetteur ou de l'auteur de l'image.
- La non-répudiation, en empêchant un utilisateur d'en nier l'envoi ou la création ultérieurement.

Les images numériques sont exposées à plusieurs types de menaces. L'une des plus préoccupantes est la falsification de contenu, où une image est intentionnellement modifiée afin de tromper, comme dans le cas de la manipulation de clichés de vidéosurveillance ou de documents médicaux. La violation de la confidentialité constitue une autre menace majeure : elle survient lorsqu'une image est interceptée ou extraite sans autorisation, révélant potentiellement des informations sensibles. La violation de la propriété intellectuelle est également fréquente, notamment par la copie, la réutilisation

ou la diffusion illégale d'images protégées. Enfin, l'altération involontaire peut survenir au cours de traitements numériques tels que la compression, le recadrage ou le changement de format, ce qui peut dégrader la qualité ou l'exactitude de l'image d'origine (6).

Face à ces risques, de nombreuses techniques de sécurité sont mises en œuvre, telles que le chiffrement, le tatouage numérique (watermarking), la stéganographie, ou encore les signatures numériques, afin d'assurer une protection efficace des images dans les environnements numériques (2).

1.9.1 Cryptographie

La cryptographie est la science qui étudie et développe des techniques assurant une communication sécurisée en présence de tiers malveillants, appelés adversaires. Son objectif est de garantir que les informations échangées entre deux parties ne puissent être ni interceptées, ni lues, ni altérées sans autorisation. Elle constitue un pilier essentiel de la sécurité de l'information, en protégeant la confidentialité, l'intégrité, l'authenticité et la non-répudiation des données.

En cryptographie, un adversaire est toute entité cherchant à accéder illégalement à des données sensibles, à les modifier ou à usurper une identité. La cryptographie permet de contrecarrer ces tentatives à travers des mécanismes mathématiques robustes (11).

1.9.1.1 Fonctionnement général du chiffrement

Considérons deux parties : un émetteur et un récepteur. Lorsque l'émetteur souhaite transmettre un message, appelé texte clair (plaintext), il le chiffre à l'aide d'une clé afin de le transformer en une version illisible appelée texte chiffré (ciphertext). Ce processus s'appelle le chiffrement (encryption) (8).

Une fois le message reçu, le récepteur utilise une clé pour convertir le texte chiffré en texte clair. Cette opération est appelée déchiffrement (decryption). Le type de clé utilisée dépend du type de cryptographie.

Figure 1-10 Chiffrement d'une image (département génie électrique)

Exemple : Le chiffre de César (Chiffrement par décalage)

Le chiffre de César est un exemple historique de cryptographie par substitution. Il consiste à remplacer chaque lettre d'un message par une autre, située à un certain nombre de positions plus loin dans l'alphabet.

Par exemple, avec un décalage de 3 positions :

Texte clair : mostaganem

Texte chiffré : prvwdjdqhp

Même si un adversaire sait que le chiffre de César est utilisé, il lui faut connaître le décalage (la clé) pour retrouver le message original.

1.9.1.2 Classification des algorithmes cryptographique :

La cryptographie moderne repose principalement sur deux approches fondamentales : la cryptographie symétrique et la cryptographie asymétrique. Une troisième approche, appelée cryptographie hybride, combine les deux pour profiter de leurs avantages respectifs.

La cryptographie symétrique repose sur l'utilisation d'une seule et même clé pour le chiffrement et le déchiffrement des données. L'émetteur et le récepteur doivent donc partager cette clé secrète avant toute communication. Ce type de cryptographie est rapide, léger et particulièrement adapté aux systèmes embarqués ou aux communications en temps réel.

Cependant, le principal inconvénient réside dans la gestion sécurisée de la clé : si un tiers parvient à l'intercepter, la confidentialité de toute la communication est compromise. Parmi les algorithmes les plus connus de cryptographie symétrique, on trouve le DES (désormais considéré comme obsolète), l'AES (utilisé aujourd'hui comme standard de chiffrement), Blowfish et ChaCha20 (8).

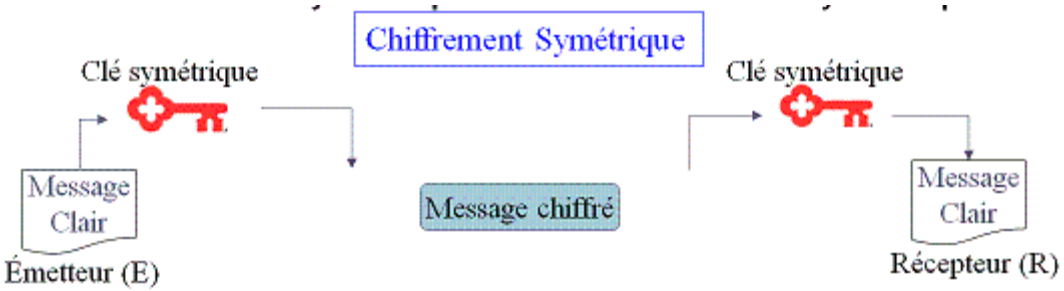

Figure 1-11 Chiffrement symétrique

La cryptographie asymétrique, aussi appelée cryptographie à clé publique, repose quant à elle sur une paire de clés : une clé publique, partagée avec tous, et une clé privée, conservée secrète par son propriétaire. Lorsqu'une personne souhaite envoyer un message confidentiel, elle utilise la clé publique du destinataire pour le chiffrer. Seule la clé privée du destinataire pourra ensuite le déchiffrer. Ce mécanisme permet non seulement d'assurer la confidentialité des données, mais aussi de garantir l'authenticité du message à travers l'usage de signatures numériques. La cryptographie asymétrique résout le problème de distribution des clés rencontré en cryptographie symétrique, mais elle est plus lente et consomme davantage de ressources. Les algorithmes asymétriques les plus utilisés sont RSA, basé sur la difficulté de factorisation des grands nombres, ECC (cryptographie sur les courbes elliptiques), qui offre une sécurité équivalente avec des clés plus petites, et le protocole Diffie-Hellman, utilisé pour l'échange de clés (12).

Figure 1-12 Chiffrement asymétrique

En pratique, les systèmes modernes adoptent souvent une approche hybride. Ce modèle combine la rapidité de la cryptographie symétrique avec la sécurité de la cryptographie asymétrique. Par exemple, lors d'une connexion HTTPS, le navigateur utilise d'abord un algorithme asymétrique (comme RSA ou ECC) pour échanger une clé secrète temporaire, appelée clé de session. Ensuite, cette clé est utilisée dans un algorithme symétrique (souvent AES) pour chiffrer rapidement les données échangées pendant toute la durée de la session.

1.9.2 Filigrane numérique (Watermarking)

Le filigranage numérique consiste à intégrer de manière discrète un marquage à l'intérieur d'une image afin d'en assurer l'authentification, la traçabilité ou la protection contre la copie et la falsification. Ce marquage, appelé filigrane, peut contenir des informations telles qu'un identifiant unique, un logo, une signature numérique ou un code de propriété intellectuelle.

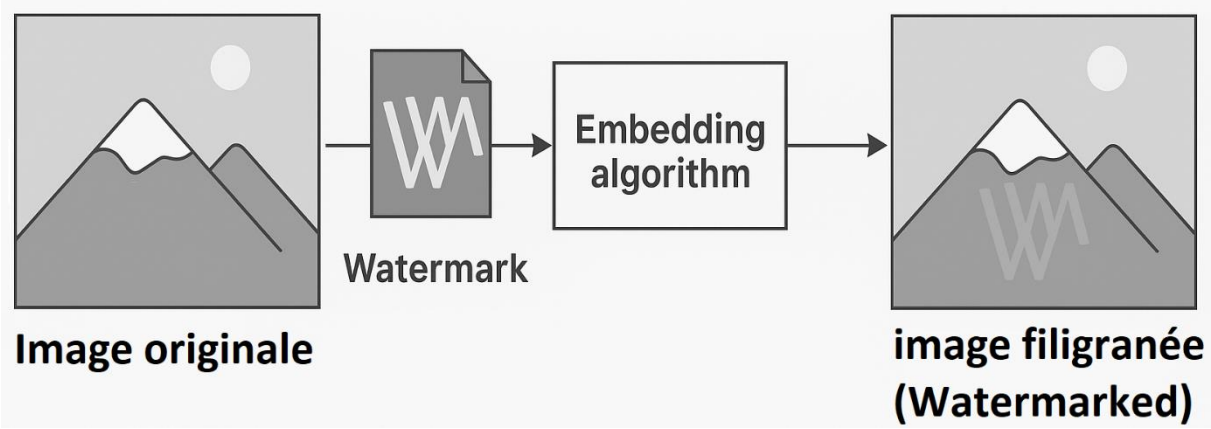

Figure 1-13 Watermarking

Selon le niveau de résistance souhaité face aux altérations, on distingue plusieurs types de filigranage :

- Le filigranage robuste : conçu pour résister aux manipulations courantes comme la compression (notamment JPEG), le recadrage, l'ajout de bruit ou encore les rotations. Il est souvent utilisé dans la protection des droits d'auteur.
- Le filigranage fragile : sensible à la moindre modification de l'image. Il permet ainsi de détecter toute tentative de falsification ou de retouche, ce qui le rend idéal pour la vérification de l'intégrité d'un contenu.
- Le filigranage semi-fragile : intermédiaire, il tolère certaines opérations bénignes comme la compression ou l'ajustement de luminosité, mais détecte les altérations sémantiques plus significatives. Il est adapté aux environnements où un certain niveau de traitement d'image est attendu, sans compromettre la sécurité.

Le filigrane peut être inséré dans le domaine spatial, en modifiant directement les valeurs des pixels, ou dans le domaine fréquentiel, en modifiant les coefficients issus de transformations comme la DCT (DiscreteCosineTransform) ou la DWT (DiscreteWaveletTransform). Le choix du domaine dépend du compromis recherché entre invisibilité, robustesse et capacité d'insertion.

Exemples d'applications concrètes :

- Dans le domaine médical, les images issues de l'imagerie par résonance magnétique (IRM) ou du scanner peuvent être filigranées afin de garantir leur intégrité et d'authentifier les diagnostics associés.

- Les bases de données biométriques, contenant des empreintes digitales ou des images faciales, utilisent le filigranage pour protéger l'identité des individus contre toute forme de falsification ou de fuite de données.

- Sur les réseaux sociaux et les plateformes de partage en ligne, un filigrane automatique peut être inséré pour prévenir le vol ou l'utilisation non autorisée d'images publiées.

1.9.3 Stéganographie

La stéganographie est l'art de dissimuler une information (texte, image, fichier audio ou autre) au sein d'une image porteuse, de manière à ce que sa présence passe inaperçue, notamment à l'œil humain. Cette technique repose souvent sur l'exploitation des bits de poids faible (LSB, Least Significant Bits) ou sur des transformations fréquentielles telles que la DCT (DiscreteCosineTransform) ou la DWT (DiscreteWaveletTransform).

Utilisée principalement pour la communication secrète, la stéganographie offre un niveau de discrétion élevé, mais reste sensible aux traitements appliqués à l'image (compression, redimensionnement, filtrage), qui peuvent altérer ou détruire le message dissimulé sans modifier visuellement l'image.

Une étude détaillée des méthodes de stéganographie, de leurs principes, classifications, performances et vulnérabilités, sera présentée dans le chapitre 2.

1.10 Conclusion

Ce premier chapitre a posé les fondations théoriques essentielles à la compréhension du traitement et de la sécurité des images numériques. Nous avons exploré les caractéristiques fondamentales des images, les étapes de leur acquisition, de leur traitement et de leur analyse, ainsi que les techniques avancées d'extraction de caractéristiques visuelles.

Une attention particulière a été accordée aux enjeux de sécurité des images, un domaine devenu critique avec la multiplication des échanges numériques. Les concepts de cryptographie, de filigranage numérique et de stéganographie ont été introduits comme moyens de garantir la confidentialité, l'intégrité, l'authenticité et la protection contre la répudiation des contenus visuels.

Ce cadre général constitue une base solide pour aborder, dans le chapitre suivant, une étude approfondie de la stéganographie, ses méthodes, ses performances et ses applications concrètes.

Chapitre 2 :

Stéganographie d'image : principe et état de l'art

2 Chapitre 2 : Stéganographie d'image : principe et état de l'art

2.1 Introduction

À l'ère numérique, les ordinateurs et Internet jouent un rôle central dans la communication, connectant les différentes régions du globe au sein d'un monde virtuel unifié. Cette interconnexion a grandement facilité l'échange d'informations, rendant la distance géographique presque insignifiante. Toutefois, cette facilité de communication s'accompagne de nouveaux défis, notamment en matière de sécurité des données, particulièrement lorsque celles-ci sont sensibles ou confidentielles.

Dans ce contexte, des méthodes de protection de l'information ont vu le jour, parmi lesquelles la stéganographie occupe une place singulière. Cette dernière se distingue par sa capacité à cacher l'existence même du message, contrairement à la cryptographie, qui rend le contenu illisible sans clé mais reste détectable. Dans certaines situations, le simple fait de transmettre un message chiffré peut susciter la méfiance. À l'inverse, un message dissimulé dans une image, une vidéo ou un fichier sonore peut passer totalement inaperçu.

La stéganographie constitue ainsi une solution complémentaire à la cryptographie : utilisée conjointement, ces deux approches permettent un niveau de sécurité accru. Même si le message est découvert, il reste protégé grâce au chiffrement. Cette combinaison garantit à la fois la discrétion (stéganographie) et la confidentialité (cryptographie).

Il existe un large éventail de techniques permettant d'insérer des données secrètes dans un fichier image, chacune présentant ses avantages, contraintes et domaines d'application spécifiques. Les approches modernes peuvent être classées selon deux grands critères :

- Le type de support utilisé (image, audio, vidéo, etc.)
- La technique d'insertion employée (dans le domaine spatial, fréquentiel, statistique, etc.)

Ce chapitre a pour objectif de fournir une vue d'ensemble claire et structurée de la stéganographie, en abordant son évolution historique, ses différentes classifications, ainsi que les techniques classiques et avancées utilisées dans le cadre du traitement des images numériques.

2.2 Historique de la stéganographie :

La dissimulation de messages en masquant leur existence n'est en rien une pratique moderne. Depuis l'Antiquité, des méthodes ingénieuses ont été mises en œuvre pour transmettre des

informations secrètes sans éveiller les soupçons. Ces techniques, bien que rudimentaires comparées aux approches numériques actuelles, constituaient déjà les prémices de la stéganographie.

2.2.1 Stéganographie sur tablette de cire (Grèce antique)

Un des premiers exemples documentés de stéganographie remonte à la Grèce antique. L'historien Hérodote relate un épisode célèbre impliquant le roi spartiate Démarate, qui souhaitait avertir secrètement Sparte d'une invasion imminente par les Perses.

Pour contourner la censure ennemie, Démarrât grava le message directement sur le bois d'une tablette, avant de recouvrir celui-ci d'une couche de cire vierge. Une fois scellée, la tablette semblait parfaitement ordinaire et pouvait contenir un message anodin à sa surface, destiné à détourner l'attention. Une fois arrivée à destination, le destinataire n'avait qu'à gratter la cire pour révéler le véritable message caché en dessous.

Cette méthode ingénieuse démontre que la stéganographie, même à ses débuts, ne visait pas seulement à cacher l'information, mais à cacher l'existence même de la communication.

Figure 2-1 Stéganographie sur tablette de cire (Grèce antique)

Un autre exemple ingénieux de stéganographie antique nous vient des Guerres Médiques, au début du Ve siècle av. J.-C. Durant cette période troublée, Histiaeus, tyran de Milet, était retenu à Suse par le roi perse Darius Ier. Désireux d'inciter une révolte contre l'autorité perse, mais privé de tout moyen de communication sécurisé, il conçut une méthode de dissimulation particulièrement audacieuse.

Selon les récits historiques, Histiaeus fit raser le crâne d'un esclave loyal, puis y fit tatouer un message secret destiné à ses alliés. Une fois le tatouage terminé, il attendit que les cheveux repoussent,

masquant totalement le contenu inscrit sur la peau. L'esclave, désormais porteur d'un message invisible, fut ensuite envoyé à Milet sous l'apparence anodine d'un simple messager. À son arrivée, ses cheveux furent rasés, révélant le message dissimulé qui appelait à l'insurrection.

Ce procédé démontre avec éloquence que la stéganographie, bien avant l'ère numérique, exploitait déjà des stratégies ingénieuses pour contourner la surveillance et transmettre des informations sensibles.

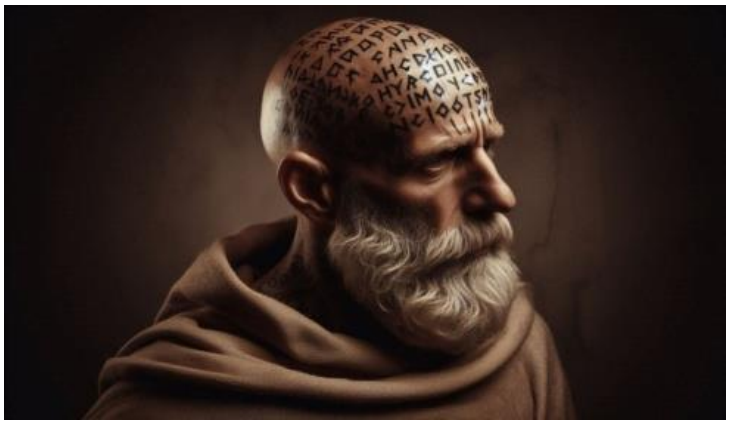

Figure 2-2 Messages tatoués sur le crâne (Guerres Médiques)

2.2.2 Encre invisible (Moyen Âge):

Durant le Moyen Âge, l'utilisation de l'encre invisible s'est largement répandue comme méthode de dissimulation d'information. Cette technique reposait sur des substances naturelles et facilement accessibles, telles que le jus de citron, le lait, ou encore le vinaigre blanc. Ces liquides, une fois appliqués sur du papier à l'aide d'une plume, nissaient aucun indice visible à l'œil nu, donnant l'impression d'une feuille vierge.Pour révéler le message, le destinataire devait simplement chauffer délicatement le papier — par exemple en le tenant au-dessus d'une bougie, d'un fer chauffé, ou d'une lampe à huile. La chaleur provoquait une réaction chimique faisant brunir l'encre, rendant ainsi le message lisible.

Ce procédé, simple mais efficace, illustre bien la volonté constante, à travers les siècles, de communiquer secrètement sans éveiller les soupçons. Il représente une forme primitive mais ingénieuse de stéganographie chimique, encore utilisée de manière symbolique dans certaines pratiques modernes d'espionnage ou de jeux éducatifs.

Figure 2-3 Encre invisible (Moyen Âge)

2.2.3 La stéganographie pendant la Seconde Guerre mondiale :

Au cours de la Seconde Guerre mondiale, la stéganographie a joué un rôle stratégique essentiel dans les opérations de renseignement, la transmission de messages secrets et la coordination des mouvements de résistance. Dans un contexte où la surveillance était omniprésente, il ne suffisait pas de chiffrer un message : il fallait dissimuler son existence même. Les techniques utilisées à cette époque incluaient :

- Les encres invisibles, toujours largement utilisées pour écrire entre les lignes de lettres apparemment ordinaires ;
- Les microfilms, souvent dissimulés dans des objets usuels (briquets, boutons, doublures de vêtements), permettant de transporter de grandes quantités d'information sous un format minuscule ;
- Les messages camouflés dans des illustrations, des lettres d'amour ou des cartes postales à double lecture ;
- L'utilisation de codes visuels ou de langages codés intégrés dans des ouvrages, des livres pour enfants ou même des motifs textiles.

Ces méthodes ont permis aux espions alliés, aux agents secrets et aux membres des réseaux de résistance de transmettre des informations sensibles sans éveiller les soupçons de l'ennemi, assurant ainsi la sécurité, la continuité et l'efficacité des opérations clandestines.

2.3 Classification de la stéganographie :

La stéganographie peut être classée selon deux grandes approches : linguistique et technique. Chacune regroupe différentes méthodes et sous-catégories, en fonction du type de message, du support utilisé, ou de la méthode d'insertion du message caché.

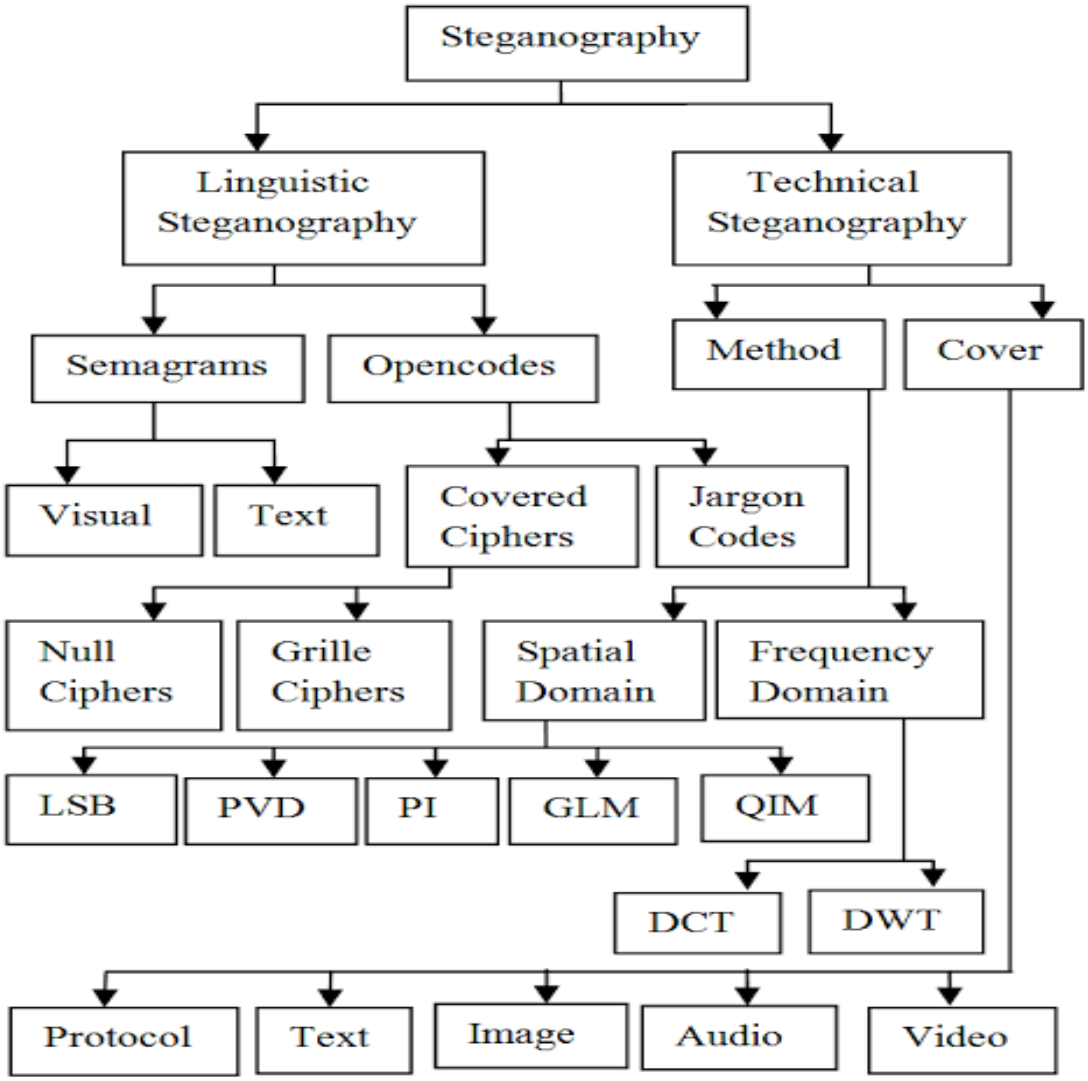

Figure 2-4 Digramme de la classification de la stéganographie

2.3.1 Caractéristiques de la stéganographie linguistique :

La stéganographie linguistique, basée sur la structure du langage, repose sur la modification de documents textuels en agissant sur leurs propriétés lexicales, syntaxiques et sémantiques, tout en préservant autant que possible le sens général du contenu.

Ces méthodes sont plus robustes que celles qui se limitent à altérer l'apparence visuelle des éléments textuels, telles que la police de caractères, l'espacement entre les lignes ou les mots, pour produire un message dissimulé.

Il existe plusieurs formes de stéganographie linguistique, que l'on peut classer comme suit :

2.3.1.1 Sémagrammes

Les sémagrammes sont des moyens de dissimulation d'informations à travers des signes, symboles ou modifications visuelles subtiles. On les classe généralement en deux catégories : visuels et textuels.

a) Sémagrammes visuels

Ce type de stéganographie consiste à transmettre un message à travers des gestes ou des arrangements physiques spécifiques. Cela peut inclure :

- Agiter la main d'une certaine manière,
- Placer un objet à un endroit précis sur un bureau,
- Modifier discrètement l'apparence d'un site web ou d'une affiche.

Ces signaux, souvent interprétés comme des comportements ordinaires dans un contexte donné, sont difficiles à détecter sans connaissance préalable du code.

Dans des situations sensibles ou dangereuses, les sémagrammes visuels peuvent constituer le seul canal de communication discret, à condition qu'un système codé ait été défini à l'avance.

b) Sémagrammes textuels

Ils consistent à intégrer un message caché directement dans un texte, à travers des éléments typographiques inhabituels tels que :

- L'usage sélectif de lettres majuscules ou soulignées,
- Des styles d'écriture particuliers,
- Des espacements inhabituels entre les mots ou les lignes.

Par exemple, des messages subliminaux peuvent se dissimuler dans des rapports ou des discussions anodines. Une phrase apparemment banale comme « le ciel est sombre aujourd'hui » pourrait, selon un code préétabli entre deux personnes, signaler une alerte ou un appel à l'aide.

2.3.1.2 Codes ouverts (Opencodes)

La stéganographie par codes ouverts consiste à dissimuler un message dans un texte légitime, sans altérer son apparence ou son sens de manière évidente. Le message caché reste invisible à un lecteur non averti, car il est intégré dans des structures linguistiques ordinaires.

Cette méthode s'appuie sur une observation essentielle : les ordinateurs et les humains n'ont pas les mêmes capacités en matière de stéganalyse (détection des messages dissimulés). Un texte peut paraître naturel pour un humain tout en déjouant les filtres automatisés grâce à des variantes lexicales, syntaxiques ou stylistiques (13).

L'objectif de cette approche est souvent de :

- Tromper les filtres électroniques, qui cherchent des mots-clés suspects dans les e-mails ou les communications ;
- Tester la robustesse des systèmes de surveillance automatisés.

Cependant, il est important de noter que ces techniques ne doivent pas être utilisées pour transmettre des informations sensibles, car elles n'offrent aucune garantie de sécurité réelle. Elles servent principalement à :

- Illustrer la faiblesse des systèmes de détection automatique,
- Ou à tester les limites de tels dispositifs.

Supposons que certains mots déclenchent le blocage automatique d'un e-mail. Si l'on connaît ces mots interdits, on peut les remplacer ou reformuler le message de manière à contourner le filtre. Par exemple, remplacer "attaque" par "opération" ou "ciel sombre" pour évoquer une menace, selon un code préétabli.

Ces ajustements peuvent suffire à faire passer un message inaperçu par les algorithmes, mais pas nécessairement par un lecteur humain entraîné.

2.3.2 Caractéristiques de la stéganographie technique :

Les approches techniques de la stéganographie exploitent les supports numériques (images, sons, vidéos, fichiers, etc.) pour dissimuler des messages dans les données binaires. Contrairement aux méthodes linguistiques, elles nécessitent des compétences en informatique, traitement du signal et mathématiques. Ces approches sont cependant bien mieux adaptées aux environnements numériques modernes.

L'efficacité de ces techniques dépend largement de leur robustesse face aux transformations telles que la compression, les modifications accidentelles ou intentionnelles, et les processus de transmission. Deux éléments fondamentaux définissent cette approche : le support (ou cover) et la méthode d'insertion utilisée pour dissimuler les données.

2.3.2.1 Le support (cover)

Le support agit comme un véhicule qui transporte les données secrètes d'un point à un autre. Il peut être constitué de bits individuels ou de blocs de données, selon la nature du média utilisé. Les supports les plus courants en stéganographie technique sont les suivants :

A. Stéganographie textuelle

Elle repose sur la modification de certaines propriétés du texte, telles que les espacements, les polices ou la mise en forme. Moins courante dans les environnements numériques modernes, elle reste utile dans certains contextes légers ou restreints.

B. Stéganographie d'image

C'est la forme la plus répandue. Elle consiste à cacher des données dans une image porteuse, en intégrant l'information de façon à préserver l'apparence visuelle de l'image. Cela peut se faire dans le domaine spatial (ex. : LSB) ou dans le domaine fréquentiel (ex. : DCT, DWT).

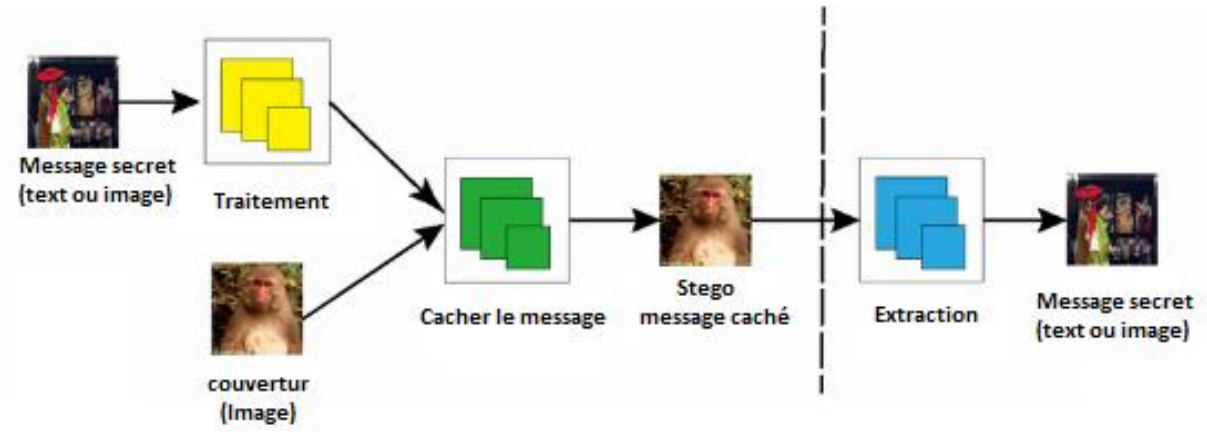

Figure 2-5 Technique LSB (Least Significant Bit)

Les données secrètes sont insérées dans un signal audio numérique. Plusieurs techniques sont disponibles, telles que la manipulation des échantillons sonores, des fréquences inaudibles ou encore l'exploitation des redondances perceptuelles du signal audio (14).

C. Stéganographie vidéo

Elle combine les approches audio et image, permettant de cacher une grande quantité d'informations dans les frames successifs d'une vidéo. La difficulté réside dans la sélection judicieuse des images dans lesquelles le message sera inséré, sans affecter la qualité visuelle globale.

2.3.2.2 Méthodes d'insertion

Les techniques de stéganographie appliquées aux images numériques se divisent généralement en deux grandes catégories :

- les méthodes du domaine spatial,
- et celles du domaine fréquentiel.

Dans le **domaine spatial**, l'insertion du message se fait directement dans les valeurs des pixels de l'image porteuse. C'est la méthode la plus simple et la plus utilisée. L'exemple le plus courant est la substitution du bit de poids faible (LSB), qui consiste à remplacer le ou les derniers bits de chaque pixel par ceux du message secret, sans affecter significativement l'apparence de l'image.

Dans le **domaine fréquentiel**, l'image est d'abord transformée à l'aide d'un outil mathématique tel que la transformée en cosinus discrète (DCT) ou la transformée en ondelettes discrètes (DWT).

Le message est ensuite inséré dans les coefficients de cette transformation. Ce procédé est généralement plus robuste contre les attaques ou traitements tels que la compression JPEG, les filtrages, ou les modifications mineures.

2.4 Techniques utilisées en stéganographie d'image

Les méthodes de stéganographie dans les images numériques se divisent principalement en deux grandes familles, selon le domaine de traitement de l'image utilisé : le domaine spatial et le domaine fréquentiel.

Chaque approche dispose de techniques spécifiques permettant d'insérer des informations secrètes tout en minimisant l'impact visuel sur l'image porteuse (15).

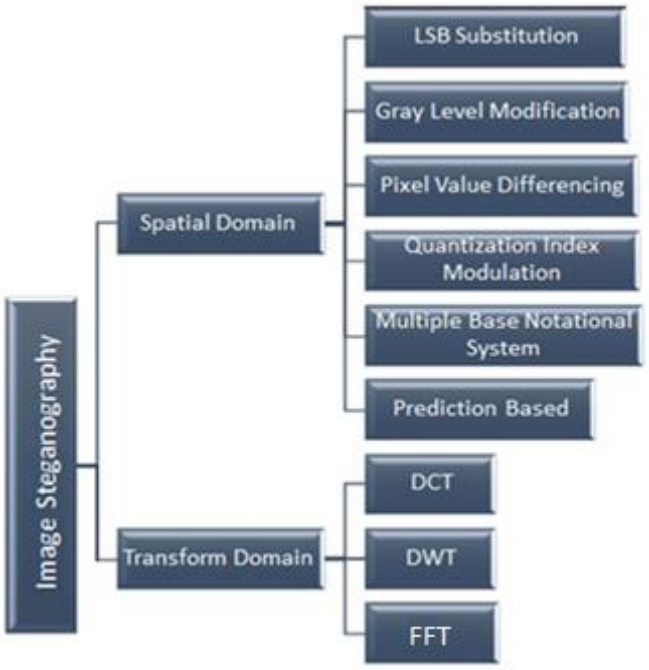

Figure 2-6 Les techniques utilisées en stéganographie d'image

2.4.1 Techniques utilisées dans le domaine spatial

Les techniques du domaine spatial consistent à modifier directement les valeurs des pixels de l'image pour y insérer des données cachées. Elles sont faciles à implémenter mais vulnérables à certaines attaques ou transformations (compression, redimensionnement, etc.).

2.4.1.1 LSB (Least Significant Bit) :

La méthode LSB (Least Significant Bit), ou méthode du bit de poids faible, est l'une des techniques les plus simples et les plus utilisées en stéganographie d'image. Elle consiste à remplacer les bits de poids faible des pixels d'une image porteuse par les bits du message secret à dissimuler.

Dans une image en niveaux de gris codée sur 8 bits (256 niveaux de gris), chaque valeur de pixel est représentée par une séquence binaire de 8 bits. En modifiant le dernier bit (le moins significatif) de chaque pixel, on peut insérer un bit de message sans provoquer de changement visuellement perceptible dans l'image. Cette faible variation ne modifie que très peu la valeur du pixel, ce qui garantit une bonne imperceptibilité (16).

Par exemple :

Pixel original : 11001010 (202)

Après insertion : 11001011 (203)

Le changement est minime et pratiquement indétectable à l'œil humain.

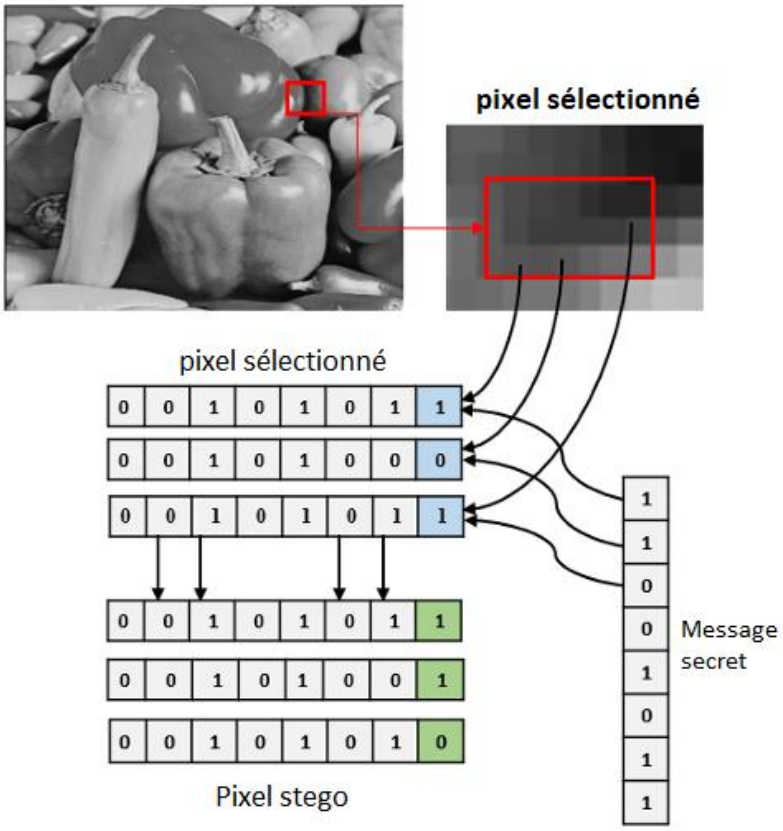

Figure 2-7 Techniques utilisées en stéganographie d'image

2.4.1.2 LSB adaptatif (LSB Matching)

La méthode LSB Matching (ou LSB adaptatif) est une amélioration de la technique LSB classique visant à réduire la détection statistique du message inséré. Contrairement à la méthode LSB simple, où le bit de poids faible d'un pixel est directement remplacé par un bit du message, la méthode LSB Matching modifie la valeur du pixel de manière plus subtile (17) (18).
Si le bit de poids faible du pixel est différent du bit à insérer, la valeur du pixel est augmentée ou diminuée aléatoirement de 1, au lieu de simplement remplacer le bit.

$$p' = \begin{cases} p+1, & siLSB(p) \neq metp < 255 \\ p-1, & siLSB(p) \neq metP > 0 \\ p, & siLSB(p) = m \end{cases}$$

Avec : m est le bit de message secret, p est le pixel de l'image et p' est le pixel Stego (modifiée)

Cela permet d'éviter certaines anomalies statistiques (comme la distribution anormale des paires de valeurs voisines), ce qui rend le message plus difficile à détecter via des techniques de stéganalyse.

Exemple :Soit un pixel ayant une valeur de 100 (en binaire 01100100, donc LSB = 0) et un bit de message 1 à insérer :

- Plutôt que de forcer le LSB à 1, on choisira de modifier la valeur du pixel à 101 (binaire 01100101) pour obtenir un LSB égal à 1.
- Si le LSB avait déjà été égal à 1, le pixel serait resté inchangé.

2.4.1.3 Modification des niveaux de gris

Cette technique repose sur une approche simple mais efficace, utilisant les niveaux de gris des pixels pour dissimuler des données, sans nécessiter un encodage complexe. Elle repose sur le principe de parité :

- Un pixel de valeur paire représente un bit 0.
- Un pixel de valeur impaire représente un bit 1.

Ainsi, pour encoder un bit dans un pixel, on ajuste simplement sa valeur afin qu'elle respecte la parité correspondant au bit secret à insérer. Par exemple, pour insérer un bit 0, on s'assure que la valeur du pixel est paire. Si elle est impaire, on l'ajuste (par exemple, en la décrémentant de 1). Inversement, pour un bit 1, la valeur du pixel doit être impaire.

Exemple : Supposons que tu veux cacher les bits 0 1 1 0 dans les niveaux de gris suivants (en 8 bits, valeurs entre 0 et 255) :

Pixel original	Bit à insérer	Pixel après modification
120 (pair)	0	120 (inchangé)
133 (impair)	1	133 (inchangé)
144 (pair)	1	145 (on ajoute 1)
157 (impair)	0	156 (on enlève 1)

Tableau 2 Exemple de modification par parité (pair/impair) dans les niveaux de gris

2.4.1.4 Pixel Value Differencing (PVD) :

La méthode Pixel Value Differencing (PVD) est utilisée pour offrir un bon compromis entre capacité d'insertion et robustesse dans la stéganographie d'image. Elle consiste à diviser l'image de couverture en blocs de deux pixels consécutifs. L'idée centrale est d'utiliser la différence de valeur entre ces deux pixels pour déterminer la quantité d'information pouvant y être dissimulée.

Le principe repose sur l'observation suivante :

- Une grande différence entre deux pixels indique une zone à fort contraste (par exemple les contours), où l'œil humain est moins sensible aux petites modifications, ce qui permet d'insérer plus de bits.
- Une petite différence signale une zone homogène, où l'on insère moins de bits pour éviter d'introduire des artefacts perceptibles (19).

Étapes de la méthode :

- Prendre deux pixels consécutifs : p1, p2
- Calculer la différence absolue : d = |p1 - p2|
- Déterminer l'intervalle Ri = [li, ui] dans lequel tombe d. Cet intervalle est prédéfini dans une table (par exemple : [0–7], [8–15], [16–31], etc.)
- Calculer le nombre de bits à insérer : $\tau=[\log2 (ui -li+1)]$
- Prendre τ bits du message secret, les convertir en une valeur décimale $b \in [0, 2^{\tau} - 1]$
- Calculer la nouvelle différence : d'= li + b
- Modifier p1 et p2 de manière à ce que leur nouvelle différence soit d′ tout en maintenant la moyenne de p1 et p2 pour limiter la distorsion visuelle.

Exemple :

- p1 = 100, p2 = 105 → d = 5
- d∈ [0, 7] → on peut insérer τ = log2(8) = 3 bits
- Message à insérer : 011 → b = 3
- Nouvelle différence : d′ = li + b = 0 + 3 = 3
- Modifier p1 et p2 pour que |p1 - p2| = 3

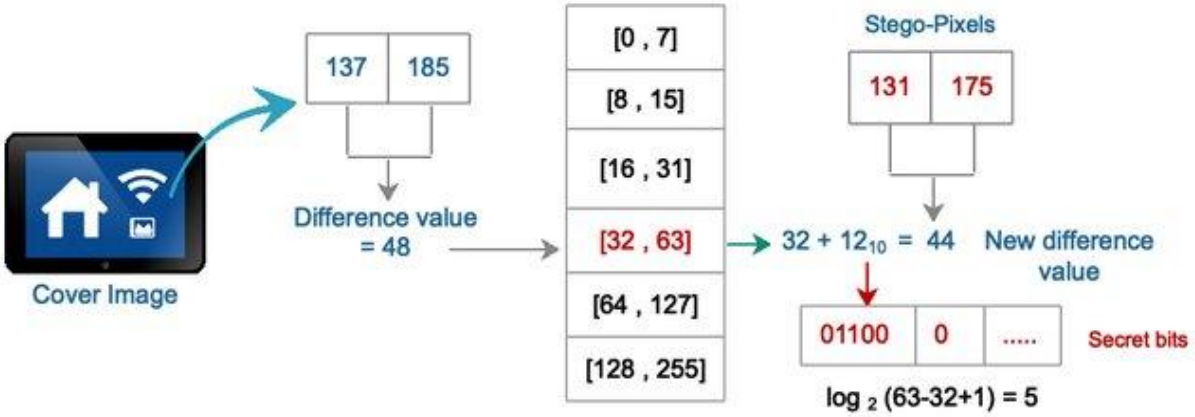

Figure 2-8 Pixel Value Differencing (PVD)

2.4.1.5 Lastéganographie basée sur la prédiction :

La stéganographie basée sur la prédiction repose sur l'utilisation d'un prédicteur pour estimer la valeur d'un pixel à partir de ses voisins. L'idée est de cacher les données dans la différence (résidu) entre la valeur réelle du pixel et sa valeur prédite, ce qui permet de réduire la distorsion tout en maintenant une capacité d'insertion élevée (16).

Certaines variantes de cette méthode atteignent une capacité allant jusqu'à 99,85 %, en particulier lorsqu'elles sont combinées avec des techniques d'optimisation ou de compression.

Principe de fonctionnement :

Soit une image I de taille M x N , et I (i,j) l'intensité du pixel situé en position(i,j)

a. Estimation par prédiction :P(i,j) = valeur prédite du pixel à partir de ses voisins
b. Calcul du résidu (erreur de prédiction) :e(i ,j) = I(i, j) – P (i,j)
c. Insertion d'un bit secret : Un bit du message est inséré en modifiant légèrement le résidu e (i,j), par exemple : en ajustant sa parité (pair/impair),

d. Reconstruction du pixel stéganographié : I' (i , j) = P(i , j) + e' (i , j), Ou e'(i, j) est le résidu modifié contenant l'information secrète.

Exemple :

a. Image locale (4 pixels d'une ligne) :

Supposons une ligne de pixels avec les intensités suivantes (valeurs en niveaux de gris 8 bits):

… | 120 | 122 | ? | 130 | …

On veut insérer un bit secret dans le pixel inconnu I (i,j) = ?

b. Prédiction de la valeur du pixel :

On utilise un prédicteur simple, comme la moyenne des deux voisins horizontaux :

$$P(i, j) = (122 + 130) / 2 = 126$$

c. Valeur réelle du pixel :

Supposons que la valeur réelle du pixel soit :

$$I(i,j)=125$$

Donc le résidu est :

$$e(i,j)=I(i,j)-P(i,j)=125-126=-1$$

d. Insertion du bit secret :

Disons qu'on souhaite insérer le bit 0.

Méthode : on encode ce bit dans la parité du résidu.

Actuellement, e(i,j)=−1e(i,j) = -1 (impair)

Pour insérer un bit 0, on veut que le résidu soit pair

On modifie e (i,j) en e' (i,j) = 0 ,(valeur la plus proche avec parité paire)

e. Calcul du nouveau pixel :

$$I'(i,j) = P(i,j) + e'(i,j) = 126 + 0 = \mathbf{126}$$

Donc, le pixel I(i,j) devient 126 au lieu de 125, ce qui est une modification légère et quasi imperceptible, tout en transportant un bit d'information.

2.4.2 Techniques utilisées dans le domaine fréquentiel :

Les méthodes de stéganographie en domaine fréquentiel consistent à transformer l'image depuis son domaine spatial (pixels) vers un domaine mathématique fréquentiel, dans lequel l'information secrète sera insérée dans les coefficients transformés. Cette approche permet une meilleure robustesse face aux compressions (comme JPEG), aux filtrages ou aux modifications accidentelles.

Il existe plusieurs techniques basées sur des transformations mathématiques, parmi lesquelles

- La Transformée en Cosinus Discrète (DCT).
- La Transformée en Ondelette Discrète (DWT).
- La Transformée de Fourier Discrète (DFT).

2.4.2.1 Transformation en Cosinus Discret (DCT)

La transformation en cosinus discrète (DCT) repose sur la conversion d'un signal ou d'une image du domaine spatial (pixels) vers le domaine fréquentiel. Elle permet de décomposer une image en plusieurs composantes fréquentielles, chacune représentant un certain niveau de détail ou d'énergie dans l'image.

La DCT est largement utilisée dans les standards de compression d'image, comme JPEG, en raison de sa capacité à concentrer l'énergie visuelle dans un petit nombre de coefficients.

L'image est généralement divisée en blocs (souvent de 8×8 pixels), puis chaque bloc est transformé à l'aide de la DCT. Cela génère une matrice de coefficients DCT, où :

- Le coin supérieur gauche représente les basses fréquences (éléments visuels globaux),
- Les coins inférieurs droits représentent les hautes fréquences (détails fins).

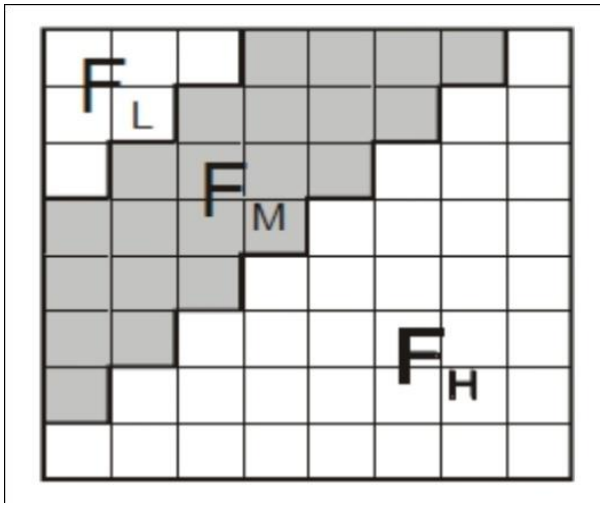

Figure 2-9 Transformation en Cosinus Discret (DCT)

Choix des positions d'insertion :

- **Coefficients de basses fréquences** :Mauvais choix pour la dissimulation, car ces coefficients influencent fortement l'apparence visuelle de l'image. Toute altération peut être perceptible par l'œil humain.
- **Coefficients de hautes fréquences** :Peu robustes, car ils sont souvent supprimés lors de la compression (par exemple, en JPEG) ou altérés par le bruit.
- **Coefficients de fréquences moyennes** :Représentent le meilleur compromis entre invisibilité et robustesse. Ce sont les zones privilégiées pour insérer les données secrètes de manière discrète tout en assurant une certaine résistance aux transformations.

a. Formule générale de la DCT :

Soit une image ou un bloc de taille N×N, avec des valeurs de pixels f(x,y), la DCT est définie comme :

$$F(u,v) = \frac{1}{4} \cdot C(u) \cdot C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cdot \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cdot \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

Où :

- u,v=0,1,2,...,N−1 : indices des fréquences (coordonnées dans le domaine DCT),
- f(x,y) : intensité du pixel à la position(x,y),
- F(u,v) : coefficient DCT à la position (u,v),
- C(k) = $\frac{1}{\sqrt{2}}$ sik=0, et C(k)=1 sinon (pour normalisation).

b. Formule inverse (IDCT)

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} C(u)\cdot C(v)\cdot F(u,v)\cdot \cos\left[\frac{(2x+1)u\pi}{2N}\right]\cdot \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

La figure suivante montre la transformation d'un bloc 8×8 de l'image du domaine spatial vers le domaine fréquentiel à l'aide de la DCT.

$$f = \begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix} \Rightarrow F = \begin{bmatrix} 1260 & -1 & -12 & -5 & 2 & -2 & -3 & 1 \\ -23 & -17 & -6 & -3 & -3 & 0 & 0 & -1 \\ -11 & -9 & -2 & 2 & 0 & -1 & -1 & 0 \\ -7 & -2 & 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 2 & 0 & -1 & 1 & 1 \\ 2 & 0 & 2 & 0 & -1 & 1 & 1 & -1 \\ -1 & 0 & 0 & -1 & 0 & 2 & 1 & -1 \\ -3 & 2 & -4 & -2 & 2 & 1 & -1 & 0 \end{bmatrix}$$

Figure 2-10 Exemple de transformation DCT d'un bloc 8×8

Pour chaque bloc 8×8 de l'image, un bit du message secret est généralement inséré. L'insertion ne se fait pas sur l'ensemble des coefficients DCT du bloc, mais seulement sur un sous-ensemble ciblé, en particulier les coefficients de fréquences moyennes, afin d'assurer un bon compromis entre invisibilité et robustesse.

La capacité d'insertion dans l'image peut être estimée par le rapport suivant :

Capacité = (M1 x M2)/ (N1 x N2)

Où :

- M1×M2 représente la taille du message à insérer (en bits),
- N1×N2 est la taille totale de l'image ou du nombre total de blocs utilisés.

En pratique, pour améliorer la robustesse du système (face aux compressions, au bruit, ou à d'autres attaques), il est souvent nécessaire de réduire la quantité d'informations insérées.

Un équilibre est donc à trouver entre capacité, invisibilité et robustesse, en fonction de l'application visée.

2.4.2.2 Transformation en Ondelettes Discrètes (DWT)

La transformée en ondelettes discrètes (DWT) est largement utilisée dans le traitement du signal et la compression d'images. Elle permet de décomposer un signal en un ensemble de fonctions de base appelées ondelettes. La DWT est reconnue pour sa flexibilité et son efficacité dans la décomposition des signaux en sous-bandes fréquentielles. Ces dernières années, le comité JPEG a d'ailleurs adopté cette méthode dans le cadre du standard JPEG-2000 (20).

La DWT est utilisée dans de nombreuses applications : compression audio, vidéo, et image.

A. Décomposition d'image en sous-bandes

Lorsqu'une DWT est appliquée à une image, celle-ci est décomposée en quatre sous-bandes :

- **LL (Low-Low) :** composantes de basse fréquence horizontale et verticale
- **HL (High-Low)** : hautes fréquences horizontales, basses fréquences verticales
- **LH (Low-High)** : basses fréquences horizontales, hautes fréquences verticales
- **HH (High-High)** : composantes de hautes fréquences dans les deux directions

Comme illustré dans la figure 2.11(a), L correspond à un filtrage passe-bas et H à un filtrage passe-haut.

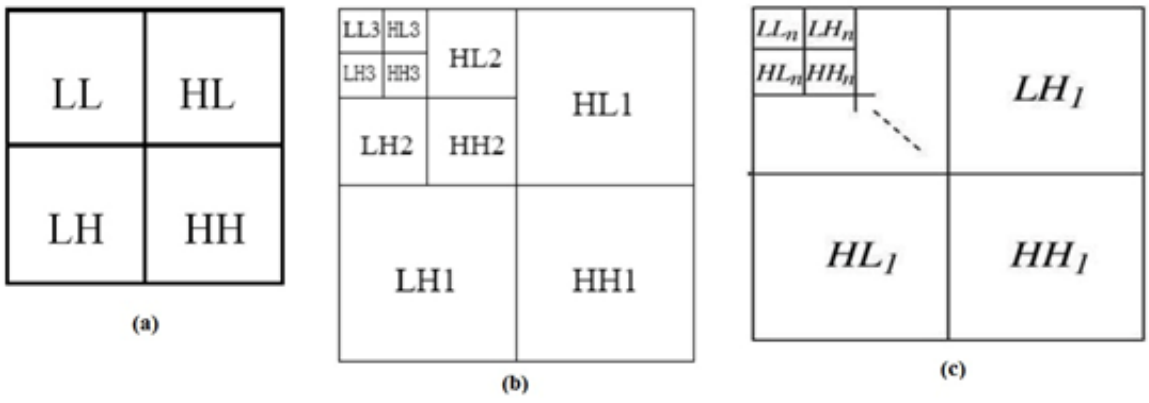

Figure 2-11 La Décomposition d'imageen sous-bandes

B. Décompositions multi-niveaux

Dans le cas d'une décomposition multi-niveaux, illustrée par la figure 2.13(b), l'image subit plusieurs applications successives de la DWT. À chaque niveau de décomposition :

- La sous-bande LL1 est elle-même décomposée en LL2, LH2, HL2, HH2
- En répétant cette opération N fois, on obtient une structure hiérarchique comportant 3N+1 sous-bandes :

$$LL_N,\ LH_x,\ HL_x,\ HH_x,\ ou`\ x = 1 \text{ à } N$$

Cette approche permet une représentation multi-résolution de l'image.

C. Choix des zones d'insertion

La sous-bande LL contient la majorité de l'énergie visuelle de l'image. Modifier cette région altérerait considérablement sa qualité.

Les sous-bandes HH, LH, HL contiennent les détails fins (textures, bords), moins sensibles à l'œil humain.

Par conséquent, l'insertion du message secret est généralement effectuée dans les sous-bandes de hautes fréquences (HHx, LHx, HLx) afin d'assurer un bon compromis entre robustesse et invisibilité.

Une fois les données insérées, on applique la transformée en ondelettes inverse (IDWT) pour reconstruire l'image filigranée.

Formule mathématique de la DWT

La DWT d'un signal f(t) est exprimée par la projection de ce signal sur une ondelette mèreψ (t) et une fonction d'échelle ϕ(t), selon :

$$c_{j,k} = \int f(t)\,\phi_{j,k}(t)\,dt \qquad \text{(coefficients d'approximation)}$$

$$d_{j,k} = \int f(t)\,\psi_{j,k}(t)\,dt \qquad \text{(coefficients de détail)}$$

Avec:

$$\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k), \quad \psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k)$$

- j : : niveau d'échelle (ou résolution)

- k : paramètre de translation
- Cj,k : approximation (basses fréquences)
- dj,k : détails (hautes fréquences)

Pour une image I(x,y), on applique la DWT ligne par ligne, puis colonne par colonne. Cela donne quatre sous-bandes après une décomposition à un niveau :

DWT (I)={LL LH, HL, HH}

- LL : approximation (basse fréquence horizontale et verticale)
- LH : détails verticaux
- HL : détails horizontaux
- HH : détails diagonaux

2.4.2.3 Transformation de Fourier Rapide (FFT)

La transformée de Fourier permet de convertir une image du domaine spatial (valeurs de pixels) au domaine fréquentiel, où chaque valeur représente une composante sinusoïdale à une certaine fréquence et amplitude. Cela permet d'identifier les zones à haute ou basse fréquence dans l'image (21).

La version discrète de cette transformée, appelée Transformée de Fourier Discrète (DFT), est souvent calculée de manière efficace à l'aide de l'algorithme FFT (Fast Fourier Transform).

Formule mathématique de la DFT 2D

Pour une image 2D f(x,y) de taille M x N, la DFT est définie par :

$$F(u,v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \cdot e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

- f(x,y) : valeur de pixel à la position (x,y)(x,y)(x,y)
- F(u,v) : coefficient fréquentiel correspondant aux fréquences u,vu,vu,v
- j : l'unité imaginaire

L'image peut ensuite être reconstruite par la transformée inverse :

$$f(x,y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \cdot e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

Dans la stéganographie par FFT, Le message secret est inséré dans les coefficients fréquentiels de l'image porteuse.

Généralement, les fréquences moyennes sont utilisées pour éviter une altération visuelle importante (basses fréquences) tout en maintenant une certaine robustesse (hautes fréquences sont vulnérables à la compression ou au bruit).

Après l'insertion, une transformée inverse (IFFT) est appliquée pour reconstruire l'image stéganographiée.

La figure 2.12 illustre l'application de la FFT sur une image en niveaux de gris, avec un centrage du spectre (fftshift) et une visualisation logarithmique pour une meilleure lisibilité.

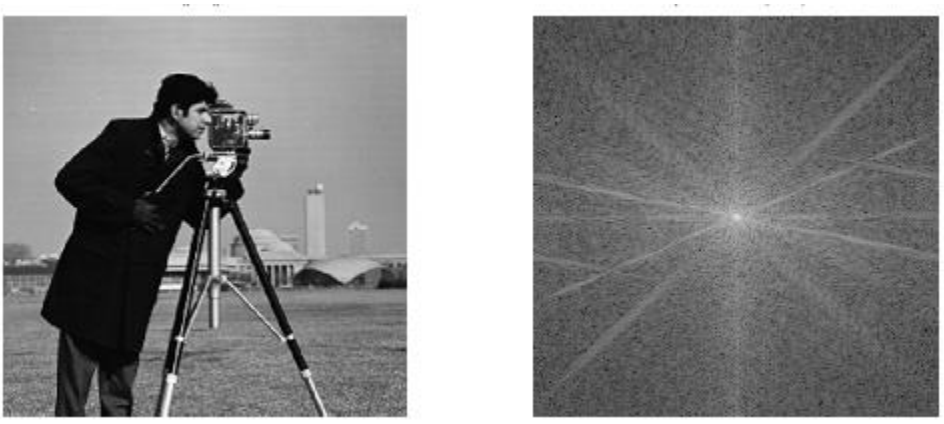

Figure 2-12L'application de la FFT sur une image en niveaux de gris

2.5 Combinaison de la cryptographie et de la stéganographie :

La cryptographie et la stéganographie, bien que puissantes individuellement, ne suffisent pas à elles seules pour garantir une sécurité totale de l'information. En effet, un message chiffré attire souvent l'attention, tandis qu'un message dissimulé sans chiffrement peut être extrait en cas de compromission du support. Ainsi, la combinaison de ces deux approches permet de concevoir un mécanisme de sécurité plus robuste et fiable.

En intégrant ces deux techniques, il devient possible d'assurer une protection renforcée des données confidentielles, tout en répondant simultanément aux exigences de confidentialité, de robustesse et d'intégrité lors de la transmission d'informations sensibles à travers des canaux ouverts.

La figure 2.13 illustre une stratégie typique de cette combinaison, dans laquelle la cryptographie est d'abord utilisée pour chiffrer les données sensibles, avant que celles-ci ne soient dissimulées dans un média hôte (image, audio, vidéo, etc.) via une technique de stéganographie.

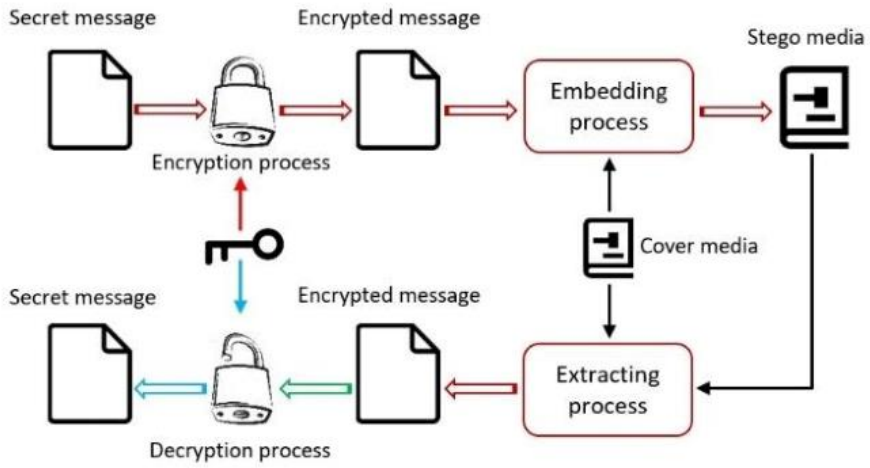

Figure 2-13 Exemple de combinaison la cryptographie et de la stéganographie

Avantages de la combinaison :

La combinaison de la cryptographie et de la stéganographie présente de nombreux avantages. Elle permet d'atteindre un niveau de sécurité accru, en associant deux couches complémentaires de protection. Tandis que la cryptographie rend les données illisibles sans clé, la stéganographie rend leur présence même invisible, réduisant ainsi la probabilité de détection. Cette double protection renforce également la résistance face aux attaques : une attaque de cryptanalyse ne suffit plus à compromettre le message, et une attaque de stéganalyse, même réussie, ne garantit pas l'accès au contenu sans la clé de déchiffrement. En outre, cette approche permet une plus grande flexibilité d'utilisation, en autorisant le chiffrement et la dissimulation de données sensibles dans divers types de médias tels que les images, les fichiers audio ou les vidéos. Elle constitue ainsi une solution efficace pour la transmission discrète et sécurisée d'informations à travers des canaux publics, tout en assurant la confidentialité, la robustesse, et l'intégrité des données échangées (20).

2.6 Recherches récentes en stéganographie d'image

La stéganographie d'image a connu un essor remarquable au cours des deux dernières décennies, en réponse aux besoins croissants en matière de confidentialité des données et de cybersécurité. De nombreuses techniques ont été développées dans le but de dissimuler efficacement des messages au sein d'images numériques, tout en maintenant une qualité visuelle acceptable et en garantissant la sécurité des informations cachées.

Les premières approches, notamment celles fondées sur la méthode LSB (Least Significant Bit), ont été largement adoptées pour leur simplicité de mise en œuvre et leur capacité d'insertion relativement importante. Toutefois, ces méthodes ont montré leurs limites face aux attaques d'analyse statistique. À ce titre, Cheddad et al. (2010) ont mené une étude comparative soulignant la vulnérabilité du LSB classique. Pour pallier ces faiblesses, plusieurs variantes ont été proposées, comme le LSB Matching ou le LSB Replacement amélioré, présentés par Hussain & Hussain (2013), visant à renforcer la robustesse.

En parallèle, les techniques opérant dans le domaine fréquentiel, telles que la DiscreteCosineTransform (DCT) et la DiscreteWaveletTransform (DWT), ont gagné en importance. Ces approches permettent une meilleure résistance aux altérations causées par la compression JPEG ou les filtrages. Par exemple, Thanki et al. (2018) ont combiné la DCT avec des algorithmes d'optimisation afin d'améliorer l'imperceptibilité, tandis que Nag et al. (2010) ont associé la DCT à un encodage de Huffman pour augmenter la capacité d'insertion.

Plus récemment, l'intelligence artificielle a ouvert de nouvelles perspectives. Des travaux comme ceux de Shi et al. (2017) sur le modèle SSGAN, basé sur les GenerativeAdversarial Networks (GANs), démontrent la possibilité de générer des images porteuses visuellement indiscernables, tout en intégrant des messages cachés. De même, Sharma et al. (2019) ont exploité les réseaux de neurones convolutifs (CNN) pour permettre un apprentissage adaptatif des zones d'insertion, augmentant ainsi l'efficacité de la dissimulation.

D'autres recherches innovantes ont exploré des métaheuristiques, comme l'algorithme Firefly étudié par Nokhwal et al. (2023), ou des approches hybrides associant chiffrement (AES, RC4) et compression (Huffman, RLE) pour renforcer à la fois la sécurité, la capacité et l'efficience.

Une analyse globale de ces travaux révèle que chaque méthode présente des compromis entre capacité d'insertion, imperceptibilité et robustesse. La tendance actuelle se dirige vers la combinaison

de plusieurs techniques — qu'elles soient spatiales, fréquentielles, chaotiques ou cryptographiques — afin de satisfaire les exigences des applications modernes : protection des données médicales, tatouage numérique, ou encore communications militaires.

Dans cette optique, notre projet s'inscrit dans cette dynamique évolutive. Il vise à développer un système de stéganographie robuste, imperceptible et autonome, en combinant la compression Huffman, un système de chiffrement (ex : RC4), des cartes chaotiques (Henon et Logistique) et une insertion basée sur le LSB (16) (22).

2.7 Conclusion :

Ce chapitre a permis d'explorer en profondeur l'univers de la stéganographie, en retraçant son évolution historique, en identifiant ses principaux types (linguistique et technique), et en détaillant les méthodes modernes d'insertion dans les images numériques, aussi bien dans le domaine spatial que fréquentiel.

Une comparaison entre cryptographie et stéganographie a été établie, mettant en évidence l'intérêt de leur combinaison pour renforcer la sécurité des communications dans un environnement numérique de plus en plus vulnérable.

En outre, un état de l'art approfondi a révélé les tendances actuelles du domaine, notamment l'intégration de plusieurs techniques (chaotiques, cryptographiques, compressives...) pour concevoir des systèmes à la fois robustes et intelligents.

Chapitre 3 :

Proposition d'un nouveau système de stéganographie

3 Chapitre 3 : Proposition d'un nouveau système de stéganographie

3.1 Introduction

Dans un contexte où les besoins en confidentialité et en protection de l'information sont de plus en plus exigeants, les techniques classiques de stéganographie, bien qu'efficaces, présentent certaines limites en termes de sécurité, de capacité et de résistance aux attaques.

Dans ce chapitre, nous proposons un système de stéganographie d'image qui combine plusieurs techniques complémentaires afin de renforcer la sécurité, la confidentialité et l'imperceptibilité des données dissimulées. La méthode repose principalement sur l'insertion dans les bits de poids faible (LSB), précédée d'un processus de compression du message à l'aide du codage de Huffman, d'un chiffrement par l'algorithme RC4, et de l'utilisation de cartes chaotiques (Henon et Logistique) pour introduire une dimension d'aléa et de complexité.

Ce système vise à tirer profit des avantages de chaque technique tout en minimisant leurs limites individuelles. Il permet ainsi une dissimulation plus efficace et plus sûre des données dans les images numériques.

Enfin, le chapitre se termine par une série de tests pratiques permettant d'évaluer les performances du système proposé à travers des critères tels que la qualité visuelle, l'analyse d'histogramme, la robustesse (PSNR), et la capacité d'extraction correcte du message.

3.2 Proposition d'un nouveau système de stéganographie

Nous proposons un nouveau système de stéganographie combinant plusieurs approches complémentaires afin de garantir une meilleure robustesse, imperceptibilité et sécurité.

Notre méthode repose principalement sur la technique LSB (Least Significant Bit) pour l'insertion des données dans l'image hôte. Pour renforcer la sécurité, le message secret est d'abord compressé à l'aide de l'algorithme de codage de Huffman, ce qui permet de réduire la taille du message tout en optimisant la capacité d'insertion.

Ensuite, nous utilisons deux cartes chaotiques – Henon et logistique – pour générer des séquences pseudo-aléatoires. Celles-ci permettent de choisir de manière non déterministe les

positions des pixels destinés à recevoir les bits du message compressé, rendant l'analyse statistique plus difficile. Ces cartes sont également utilisées pour cacher le dictionnaire Huffman dans l'image.

En complément, un système de chiffrement de type RC4 est intégré dans certaines variantes de notre approche, apportant une couche supplémentaire de protection avant l'insertion du message.

3.2.1 Techniques utilisées dans le système proposé

Le système que nous proposons repose sur l'intégration cohérente de plusieurs techniques issues de domaines complémentaires : la compression, le chiffrement, le chaos et l'insertion en domaine spatial. Chacune de ces techniques joue un rôle spécifique dans l'amélioration de la sécurité, de la robustesse et de l'efficacité de la stéganographie.

Dans cette sous-section, nous présentons brièvement les principales méthodes utilisées dans notre système.

3.2.1.1 Le codage de Huffman :

Le codage de Huffman est une méthode de compression sans perte basée sur l'attribution de longueurs de codes binaires variables aux symboles d'un message, en fonction de leur fréquence d'apparition. L'idée principale est d'optimiser la représentation binaire d'un message en attribuant des codes plus courts aux symboles les plus fréquents, et des codes plus longs à ceux qui apparaissent rarement. Cette approche permet de réduire efficacement la taille totale des données à transmettre ou à dissimuler.

Considérons un ensemble de symboles S={S1,S2,…,Sn}avec des probabilités associées P={P1,P2,…,Pn} telles que $\sum_{i=1}^{n} Pi = 1$. L'algorithme de Huffman construit un arbre binaire optimal de la manière suivante :

- Chaque feuille de l'arbre représente un symbole du message.
- Le chemin de la racine jusqu'à une feuille détermine le code binaire associé à ce symbole.
- Plus la probabilité d'un symbole est élevée, plus la profondeur de sa feuille dans l'arbre est faible, ce qui se traduit par un code plus court.

Le gain en compression réalisé grâce à cette méthode peut être mesuré par la réduction de l'entropie moyenne du message. L'entropie H, qui représente la quantité minimale d'information requise pour coder le message, est donnée par la formule :

$$H = -\sum_{i=1}^{n} = P_i \log_2(P_i)$$

La longueur moyenne du code produit par Huffman est toujours supérieure ou égale à cette entropie, mais reste proche de l'optimum, ce qui garantit une compression efficace (23).

Exemple : Soit le message suivant : HELLO

Le tableau suivant montre les fréquences d'apparition de chaque caractère dans ce message :

Tableau 3 Les fréquences d'apparition

char	*quant*	*p*
L	2	0.400
H	1	0.200
E	1	0.200
O	1	0.200

À partir de ces fréquences, l'algorithme de Huffman génère les codes binaires suivants :

Tableau 4 Les codes binaires (dictionnaire)

char	*code*
L	0
H	10
E	111
O	110

Le message HELLO est alors encodé comme suit :H → 10, E → 111, L → 0, O → 110. Le message final encodé en binaire est : 1011100110

Ce message compressé ne contient que **10 bits**, au lieu de **5 × 8 = 40 bits** si chaque caractère avait été représenté par un encodage standard de 8 bits (ASCII). Le gain de compression est donc significatif dans ce cas simple.

Dans le cadre de notre système, le codage de Huffman est utilisé pour compresser le message secret avant son insertion dans l'image porteuse, ce qui permet de réduire la taille des données à cacher et d'augmenter la capacité d'insertion tout en maintenant la discrétion.

3.2.1.2 Algorithme de chiffrement :

Le chiffrement est une étape essentielle dans tout système de dissimulation d'information visant à garantir la confidentialité des données, même si celles-ci venaient à être extraites. Dans notre

approche, un algorithme de chiffrement est utilisé en amont de l'insertion du message, afin de transformer le message secret en une forme illisible.

Le système proposé est flexible et permet l'utilisation de tout algorithme de chiffrement, qu'il soit symétrique (comme AES, DES, RC4, etc.) ou asymétrique (comme RSA, ElGamal, etc.), en fonction des exigences de sécurité, de performance et de complexité souhaitées. Cette modularité permet d'adapter le système à différents contextes d'application.

Dans le cadre de nos travaux, nous avons choisi d'illustrer cette étape à l'aide de l'algorithme RC4, un chiffrement symétrique à flot, pour sa simplicité de mise en œuvre, sa rapidité d'exécution et sa compatibilité avec des environnements à ressources limitées. Bien que des faiblesses aient été identifiées dans certains usages de RC4, il reste pertinent pour des applications spécifiques où la combinaison avec d'autres techniques, comme la stéganographie, renforce la sécurité globale (24).

Le message compressé à l'aide du codage de Huffman est donc chiffré par RC4 avant son insertion dans l'image hôte, ce qui ajoute une couche de sécurité supplémentaire et rend le message inutilisable même en cas d'extraction non autorisée.

Rappel sur l'algorithme RC4 :

RC4 est un algorithme de chiffrement à flot synchrone, simple et rapide, qui utilise une clé secrète de longueur variable (généralement 128 bits) pour produire un flux de clés pseudo-aléatoires. Ce flux est ensuite combiné avec le texte en clair à l'aide d'une opération XOR, assurant ainsi le chiffrement des données.

Son fonctionnement repose sur deux étapes principales :

- KSA (Key SchedulingAlgorithm) : initialise une permutation des octets à l'aide de la clé.
- PRGA (Pseudo-RandomGenerationAlgorithm) : génère un flux de bits pseudo-aléatoires utilisé pour le chiffrement et le déchiffrement.

RC4 ne nécessite pas de vecteur d'initialisation (IV) et partage le même processus pour le chiffrement et le déchiffrement, ce qui le rend facile à implémenter. Malgré ses failles connues dans certains contextes, il reste un bon choix pour des systèmes combinés, comme dans notre cas, avec la stéganographie (24).

3.2.1.3 System chaotique :

Les systèmes chaotiques sont des systèmes dynamiques déterministes et non linéaires qui présentent un comportement apparemment aléatoire, mais totalement régi par des équations précises.

Leur complexité et leur imprévisibilité en font des outils puissants dans les domaines de la sécurité, notamment en cryptographie et en stéganographie.

Les signaux chaotiques se distinguent par plusieurs propriétés clés :

- **Non-linéarité** : un système chaotique est intrinsèquement non linéaire. C'est cette non-linéarité qui engendre la complexité de son comportement dynamique.
- **Déterminisme** : bien qu'imprévisibles à long terme, les systèmes chaotiques sont gouvernés par des équations déterministes. À partir d'un état initial connu, leur évolution est entièrement définie.
- **Sensibilité aux conditions initiales** : une très légère variation dans les conditions de départ peut provoquer une trajectoire totalement différente — un phénomène souvent désigné comme effet papillon (25) (26).

Ces caractéristiques permettent de générer des séquences pseudo-aléatoires difficiles à prédire, utiles notamment pour :

- Choisir de manière dynamique les positions des pixels dans l'image hôte,
- Brouiller l'ordre des bits du message,
- Ou encore sécuriser la transmission d'informations auxiliaires comme le dictionnaire de Huffman.

Dans notre système, deux cartes chaotiques ont été utilisées : la carte de Hénonetlacartelogistique, en raison de leur simplicité de mise en œuvre et de leur efficacité démontrée.

a. Carte de Hénon

La carte de Hénon est un système chaotique bidimensionnel défini par :

$$\begin{cases} x_{n+1} = 1 - ax_n^2 + y_n \\ y_{n+1} = bx_n \end{cases}$$

Où, a et b sont des paramètres de contrôle avec a > 0, b > 0 et x_n,y_n représente la variable d'état du système. Le système génère une séquence chaotique stable et complexe. Elle est utilisée pour produire des coordonnées de pixels dans l'image porteuse.

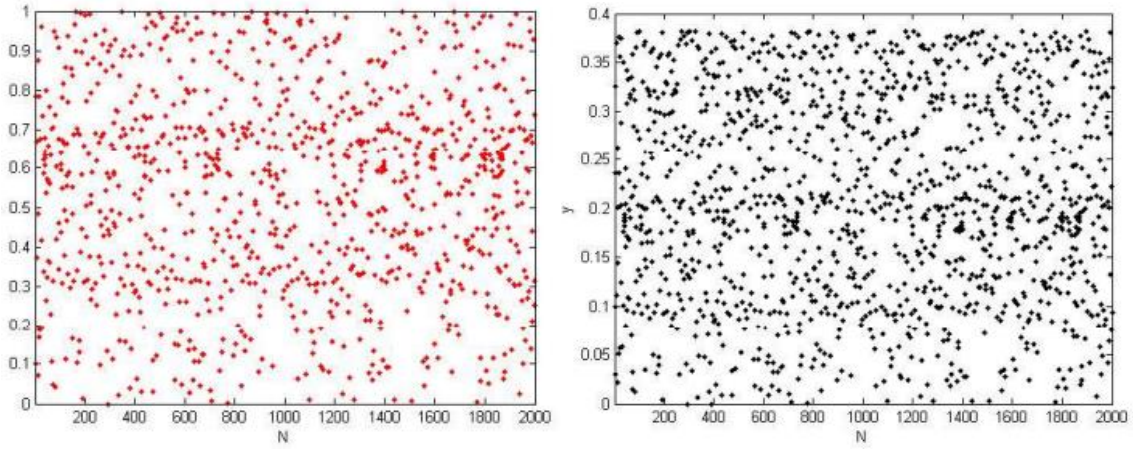

Figure 3-1 Tracée de x et y

b. Carte Logistique

La carte logistique est un modèle unidimensionnel simple mais puissant, exprimé par :

$$x_{n+1} = r x_n (1 - x_n)$$

Avec r∈[3.57,4] pour un comportement chaotique et$x_n$représente la variable d'état du système. Elle permet de générer rapidement une suite pseudo-aléatoire servant à brouiller les données du message compressé ou à sélectionner l'ordre d'insertion (27).

La figure 3.2 montreletracédes données x de la carte logistique avec le paramètre r

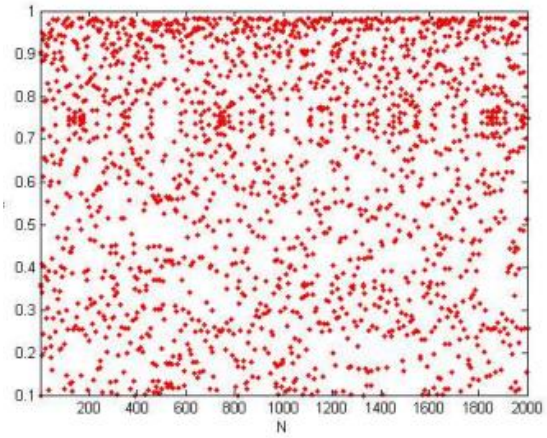

Figure 3-2 Tracée de x

3.3 Architecture générale du système proposé :

Dans cette section, nous présentons l'architecture globale du système de stéganographie développé. Cette approche vise à garantir la confidentialité, l'imperceptibilité et la sécurité du message secret, tout en assurant une extraction autonome et fiable. Le système repose sur une combinaison de techniques avancées couvrant plusieurs aspects complémentaires de la sécurité de l'information.

Le processus global se déroule selon les étapes suivantes :

1) Compression du message,
2) Chiffrement du message compressé.
3) Génération de séquences chaotiques
4) Insertion du message chiffré.
5) Insertion parallèle du dictionnaire Huffman
6) Extraction autonome à la réception.

L'architecture du système est illustrée à travers la Figure 3.3, qui schématise l'ensemble des modules impliqués dans le processus d'insertion et d'extraction du message secret.

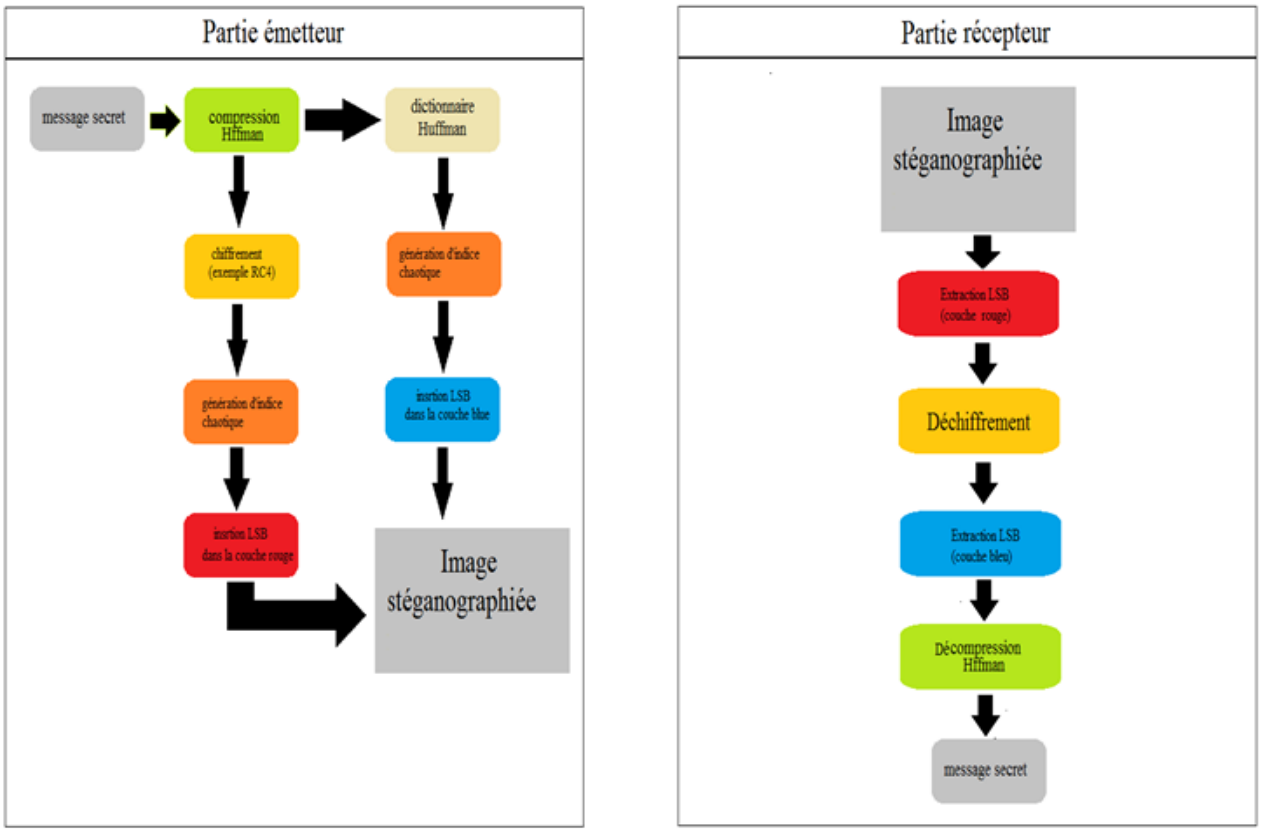

Figure 3-3 Diagramme du système de stéganographie proposé

3.3.1 Etape 1 : Compression du message

L'émetteur commence par convertir le message secret en une séquence numérique, en utilisant les codes ASCII des caractères. Une analyse de fréquence est ensuite réalisée sur cette séquence afin de construire un dictionnaire de Huffman, dans lequel chaque symbole est associé à un code binaire variable selon sa fréquence d'apparition.

Le message est alors compressé à l'aide du codage de Huffman, produisant une suite binaire plus compacte. Pour permettre une extraction correcte, la longueur du message compressé (en bits) est calculée, convertie en binaire sur 16 bits, et ajoutée au début de la séquence. Enfin, si nécessaire, des zéros de bourrage (padding) sont insérés afin que la séquence finale soit un multiple de 8(voir figure 3.4), ce qui facilitera sa conversion en octets pour l'étape de chiffrement.

Le résultat est nommé « New_sequence»

New_sequence = 16 bits | message_compressée | 00…0

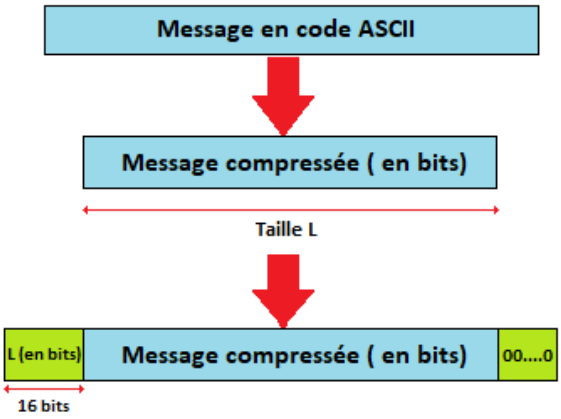

Figure 3-4 Etape de compression et de padding.

3.3.2 Etape 2 : Chiffrement du message compressé.

Après la compression du message à l'aide du codage de Huffman et du padding, une étape de chiffrement est appliquée afin de garantir la confidentialité des données avant leur insertion dans l'image hôte. Cette protection est cruciale, notamment en cas d'extraction non autorisée du message.

La séquence « **New_sequence** », sous forme binaire, est d'abord segmenté en octets (8 bits) pour être compatible avec les algorithmes de chiffrement. Ces octets sont ensuite chiffrés à l'aide de l'algorithme RC4, sélectionné ici pour sa simplicité de mise en œuvre (voir la Figure 3.5).

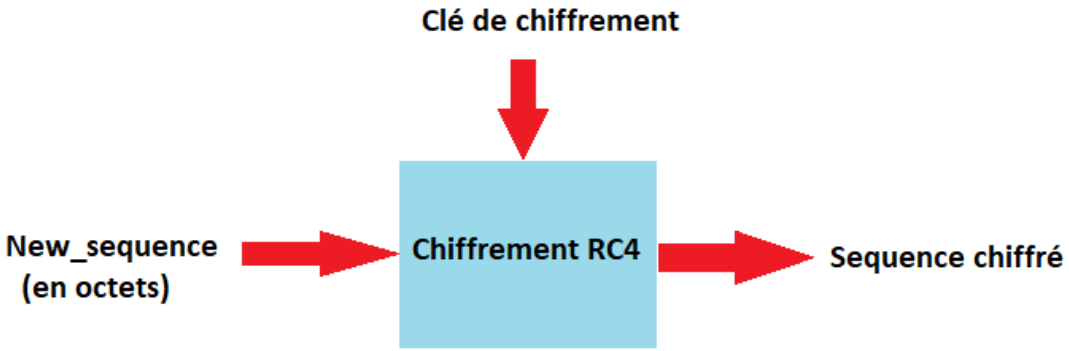

Figure 3-5 Processus de chiffrement

Enfin, la séquence chiffrée est reconvertie en une suite de bits afin de pouvoir être insérée ultérieurement dans l'image porteuse.

Il est important de souligner que, bien que RC4 ait été utilisé à titre d'exemple, des algorithmes plus robustes tels qu'AES peuvent être adoptés pour des applications nécessitant un niveau de sécurité renforcé.

3.3.3 Etape 3 : Génération de séquences chaotiques

Avant d'insérer la séquence chiffrée dans l'image hôte à l'aide de la technique LSB, deux séquences pseudo-aléatoires sont générées à partir de cartes chaotiques afin de déterminer de manière sécurisée les positions des pixels dans lesquels les données seront dissimulées.

Séquence 1 : Générée à partir de la carte de Hénon, cette séquence permet de produire une liste d'indices correspondant aux positions des pixels dans lesquels seront insérés les bits du message chiffré. Ce processus est décrit dans l'algorithme 3.1.

Algorithme 3.1

- Entrée :a ;b ;x(1) ;y(1) ;len ;imsize
- Sortie : Seq_henon
- Début :
- pouri allant de 1 à len, faire
- x(1,i+1) = 1-(a*x(1,i)^2) + y(1,i);
- y(1,i+1) = b*x(1,i);
- result=mod((x.*y)*10^7,imgsize);

- Seq_henon=result(2:end);
- fin

avec :

a et b: paramètres de la carte de Hénon, x(1) et y(1) : valeurs initiales, len : nombre d'itérations à effectuer, imgsize : nombre total de pixels et seq_henon: vecteur de positions pseudo-aléatoires.

Séquence 2 : Produite par la carte logistique, cette séquence est utilisée pour sélectionner les indices des pixels destinés à contenir le dictionnaire de Huffman, également inséré à l'aide de la méthode LSB. Ce mécanisme est défini dans l'algorithme 3.2.

Algorithme 3.2

- Entrée : x(1), r
- Sortie : Seq_logistique
- Début :
- pouri allant de 1 à N, faire
- x(i+1)=r*x(i)*(1-x(i));
- Seq_henon=round(mod(x*10^7,imgsize));
- fin

avec :

x(1) , r : paramètres de contrôle, seq_logistique : vecteur de positions pseudo-aléatoires et N : la taille de dictionnaire de huffman.

Ces séquences chaotiques assurent une dispersion aléatoire et difficilement prédictible des données insérées, ce qui renforce la sécurité du système face aux tentatives de détection ou d'extraction non autorisées.

3.3.4 Etape 4 : Insertion du message chiffré.

Dans cette étape, l'émetteur sélectionne une image couleur RGB comme support de dissimulation. Pour une insertion efficace, il est recommandé de choisir une image de taille suffisante, contenant des variations visuelles riches (textures, contrastes, détails) afin d'augmenter la capacité de dissimulation et de limiter la détection.

La séquence binaire du message chiffré, obtenue après compression et chiffrement, est insérée dans le canal rouge de l'image. La position de chaque bit à insérer est déterminée par une séquence

chaotique générée à l'aide de la carte de Hénon (seq_henon), qui joue le rôle de clé pseudo-aléatoire. Pour chaque bit, une valeur de seq_henon désigne l'indice du pixel cible dans le canal rouge, et le bit est alors inséré dans le bit de poids faible (LSB) de ce pixel.

3.3.5 Etape 5 : Insertion parallèle du dictionnaire Huffman

En parallèle de l'insertion du message chiffré, cette étape consiste à dissimuler le dictionnaire de Huffman dans l'image hôte, afin de permettre au récepteur de reconstituer correctement le message compressé. Contrairement à l'étape précédente, le canal bleu (B) est utilisé pour cette opération, et la séquence chaotique logistique (seq_logistique) sert de clé pseudo-aléatoire pour déterminer les positions d'insertion.

Avant l'insertion, l'émetteur prépare le dictionnaire en le transformant en une séquence binaire compacte. Pour chaque ligne du dictionnaire contenant un symbole et son code Huffman, les informations suivantes sont converties en binaire et concaténées :

- Le symbole ASCII codé sur 8 bits ;
- La taille du code Huffman associé, sur 8 bits ;
- Le code Huffman, sous forme binaire (de longueur variable).

Ces blocs sont concaténés successivement pour tous les symboles présents dans le dictionnaire, formant ainsi une longue séquence binaire compacte représentant l'intégralité du dictionnaire.

Dict = | symbole 1 | taille de code 1 | code 1 | symbole 2 | taille de code 2 | code 2 |……| code n |

Ensuite, la taille totale de cette séquence binaire (en nombre de bits) est calculée, puis convertie en binaire sur 16 bits. Cette information est préfixée au début de la séquence binaire compacte. Elle permettra au récepteur de savoir exactement combien de bits extraire pour reconstituer le dictionnaire lors de la phase de détection.

New_Dict = | taille totale de Dict | Dict |

Cette séquence (New_Dict) est ensuite insérée dans le canal bleu de l'image à l'aide de la technique LSB, en suivant les positions spécifiées par la séquence chaotique seq_logistique. Chaque bit du dictionnaire est ainsi dissimulé dans le bit de poids faible d'un pixel sélectionné de manière pseudo-aléatoire.

L'usage de canaux et de séquences chaotiques distincts pour le message et pour le dictionnaire renforce la sécurité globale du système, tout en assurant une organisation claire entre les différentes données cachées.

Une fois cette étape finalisée, l'image devient une image stéganographiée complète (stego image) contenant à la fois le message chiffré (dans le canal rouge) et le dictionnaire Huffman (dans le canal bleu). Cette image est alors transmise au récepteur via un canal ouvert.

3.3.6 Etape 6 : Extraction autonome à la réception.

À la réception de l'image stéganographiée, le récepteur peut extraire de manière autonome à la fois le message secret et le dictionnaire Huffman, en suivant les étapes inverses du processus d'insertion.

a) Extraction du dictionnaire via le canal bleu (B)

Le récepteur commence par appliquer la même séquence chaotique logistique utilisée à l'émission (seq_logistique) pour retrouver les positions d'insertion dans le canal bleu de l'image. Grâce à cetteséquence, ilextrait :

- D'abord les 16 premiers bits, qui indiquent la taille totale (en bits) de la séquence contenant le dictionnaire ;
- Puis, il extrait cette séquence binaire de la taille indiquée.

La séquence ainsi récupérée est ensuite décodée ligne par ligne pour reconstituer le dictionnaire de Huffman, en lisant :

- Le symbole ASCII (8 bits),
- La taille du code (8 bits),
- Et le code Huffman correspondant.

Une fois le dictionnaire reconstruit, le système est prêt à décoder le message compressé.

b) Extraction du message chiffré via le canal rouge (R)

Le récepteur utilise ensuite la séquence de Henon (seq_henon) pour retrouver les positions dans le canal rouge où a été inséré le message chiffré. Il commence par extraire les 16 premiers bits, qui indiquent la longueur du message compressé, puis il récupère les bits correspondants.

Les bits extraits sont ensuite regroupés en octets afin de pouvoir être déchiffrés correctement.

c) Déchiffrement et décompression

Les octets obtenus sont déchiffrés à l'aide du même algorithme (par exemple RC4) et de la clé secrète partagée. Le flux binaire clair résultant est alors décompressé à l'aide du dictionnaire Huffman précédemment reconstitué, ce qui permet de retrouver le message secret original.

3.4 Évaluation expérimentale du système proposé :

Afin de valider l'efficacité et la robustesse du système de stéganographie développé, une série de tests expérimentaux a été menée à l'aide d'un ensemble d'images RGB standards largement utilisées dans la littérature. Ces expériences ont pour objectif d'évaluer les performances du système selon trois dimensions essentielles : la transparence visuelle, la stabilité statistique, et la fidélité numérique après l'insertion du message secret.

Images de test : Les images utilisées pour les simulations sont :peppers.png, fabric.png, football.jpg,onion.png, et université.jpg.Elles présentent des textures, couleurs et structures variées, ce qui permet une évaluation représentative du système dans différents contextes.

Taille du message inséré : Le message secret utilisé pour les tests est volontairement choisi de manière à représenter plus de 10 % de la capacité totale de l'image (en nombre de pixels). Cela permet d'évaluer les performances du système dans un scénario de forte charge d'insertion, où la robustesse et l'imperceptibilité sont particulièrement mises à l'épreuve.

L'analyse repose sur trois axes complémentaires :

- **Analyse visuelle** : comparaison directe entre images originales et stéganographiées pour vérifier l'imperceptibilité du message.
- **Analyse statistique** : comparaison des histogrammes et les contours pour détecter d'éventuelles anomalies statistiques.
- **Analyse quantitative**: calcul de métriques objectives (MAE, MSE, PSNR, SSIM) pour quantifier la distorsion induite.

3.4.1 Analyse visuelle directe

Ce test permet une inspection visuelle directe des images (originale et stego). L'objectif est de vérifier qu'aucune modification perceptible n'a été introduite lors de l'insertion du message secret. Une bonne technique de stéganographie doit garantir une imperceptibilité visuelle parfaite à l'œil humain.

La comparaison entre les images originales et stéganographiées (voir Figures 3.6 à 3.12) ne révèle aucune altération perceptible à l'œil nu. Chaque image conserve son apparence globale, ce qui témoigne d'une excellente transparence visuelle du système proposé.

Figure 3-6 Analyse visuelle directepeppers.png

Figure 3-7 Analyse visuelle directeonion.png

Figure 3-8 Analyse visuelle directefabric.png

Figure 3-9 Analyse visuelle directefootball.jpg

Figure 3-10 Analyse visuelle directe université.jpg

3.4.2 Séparation des canaux de couleur R, G, B :

En séparant les canaux Rouge, Vert et Bleu, on peut localiser plus précisément les modifications introduites dans chaque couche. Cela permet de :

- Détecter si l'information a été insérée dans un seul canal (souvent le canal Bleu ou Rouge),
- Évaluer l'impact de l'insertion sur chaque composante chromatique.

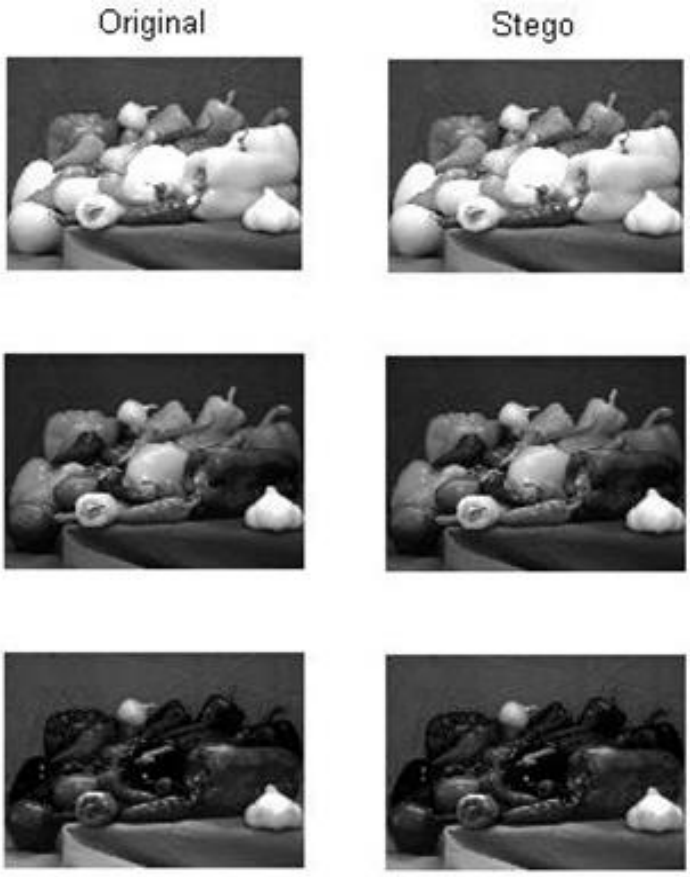

Figure 3-11 Séparation des canaux de couleur R, G, B peppers.png

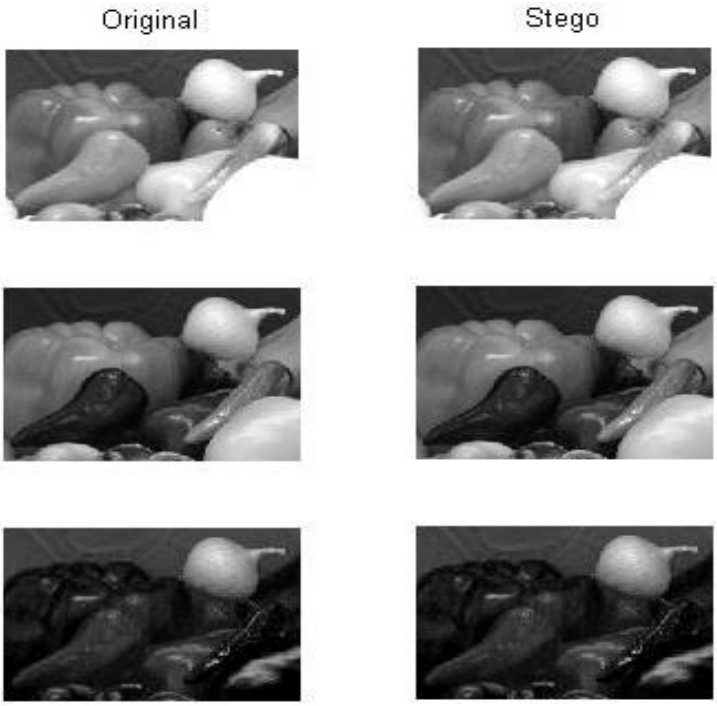

Figure 3-12Séparation des canaux de couleur R, G, B onion.png

Figure 3-13Séparation des canaux de couleur R, G, B université.jpg

Les figures 3.13 à 3.15 présentent la séparation des canaux Rouge, Vert et Bleu pour chaque paire d’images (originale et stégo). Aucune modification notable n’est détectée dans les composantes individuelles, confirmant que l'insertion du message reste visuellement imperceptible même à un niveau chromatique.

3.4.3 Analyse statistique :

Comparer les histogrammes permet de vérifier la conservation statistique de l’image. Un bon algorithme de stéganographie doit conserver une distribution d’intensité similaire à celle de l’image originale, afin d’éviter une détection par analyse statistique

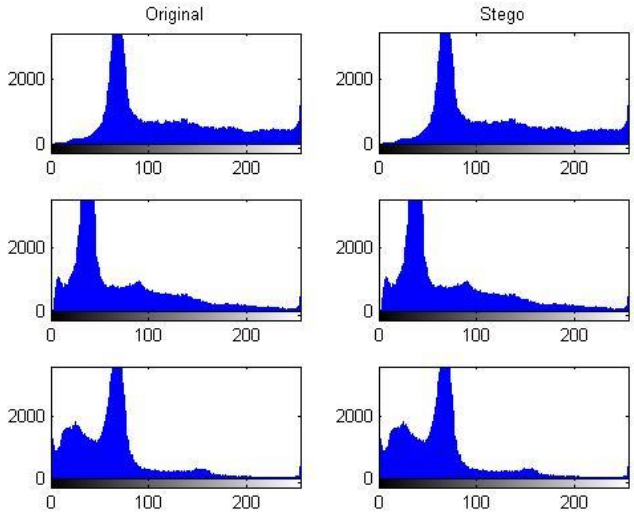

Figure 3-14 Analyse par histogrammes peppers.png

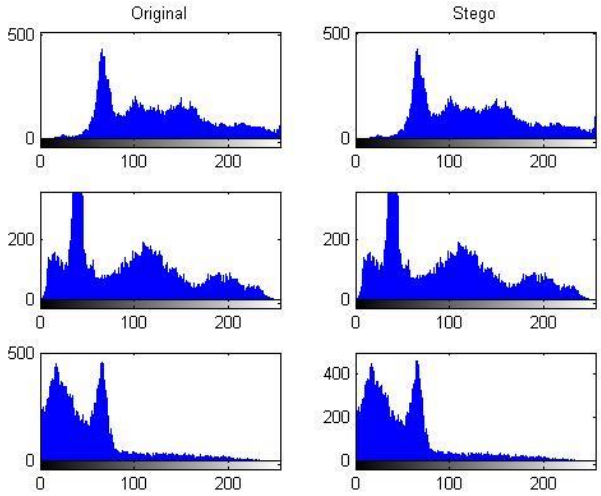

Figure 3-15 Analyse par histogrammes onion.png

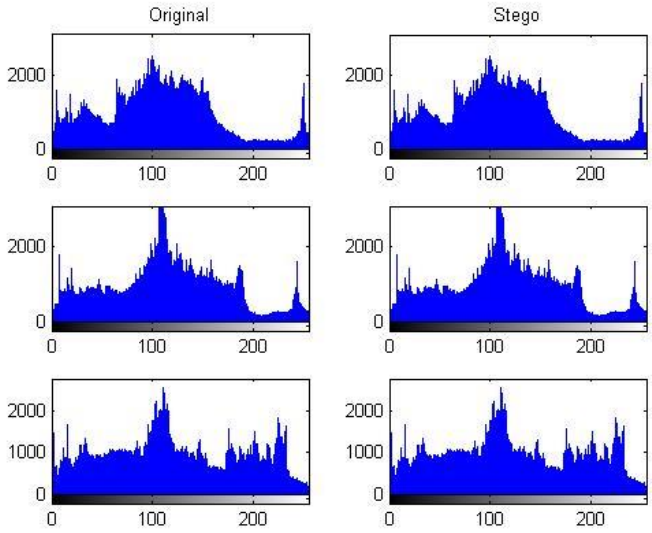

Figure 3-16 Analyse par histogrammes université.jpg

L'étude des histogrammes des canaux R, G, B des images originales et stégo (figures 3.16 à 3.18) révèle une superposition quasi parfaite des courbes. Cela signifie que la distribution statistique des intensités n'a pas été perturbée de manière significative, empêchant ainsi la détection par les techniques de stéganalyse classique.

3.4.4 Détection des contours par filtre Sobel :

Dans le cadre de l'évaluation de la sécurité de notre méthode de stéganographie, la détection des contours constitue un indicateur pertinent. En effet, si les modifications introduites par l'insertion du message secret sont perceptibles au niveau des contours de l'image, cela pourrait compromettre la discrétion du système. L'application du filtre de Sobel permet ainsi de vérifier si les structures visuelles essentielles de l'image (telles que les bords et transitions) sont altérées ou non. Une conservation fidèle des contours après insertion témoigne d'une bonne imperceptibilité et renforce la robustesse du système contre une détection visuelle ou automatisée.

Les résultats obtenus à l'aide du filtre Sobel sont présentés dans les figures suivantes. Ils mettent en évidence la stabilité des contours entre les images originales et stéganographiées, confirmant que notre méthode n'introduit pas de perturbations détectables au niveau des structures visuelles critiques.

Figure 3-17Détection des contours par filtre Sobel peppers.png

Figure 3-18Détection des contours par filtre Sobel onion.png

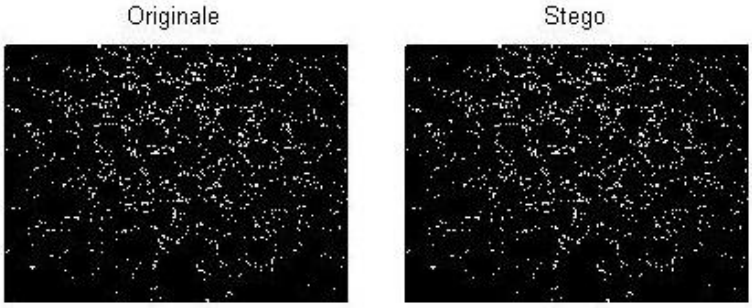

Figure 3-19Détection des contours par filtre Sobel fabric.png

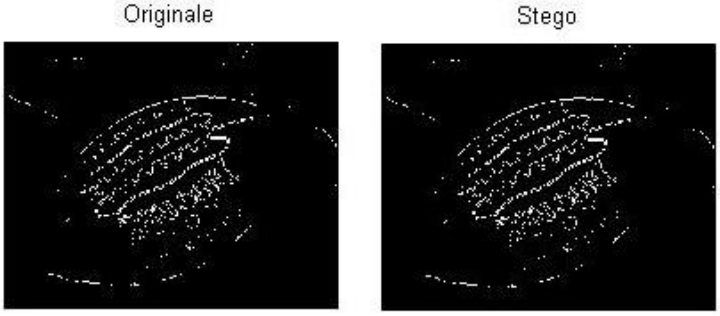

Figure 3-20Détection des contours par filtre Sobel football.jpg

Figure 3-21 Détection des contours par filtre Sobel université.jpg

3.4.5 Analyse quantitative

Dans le but d'évaluer objectivement la qualité des images stéganographiées, plusieurs métriques quantitatives standards ont été calculées entre chaque image originale et son équivalent stégo. Ces mesures permettent de quantifier la distorsion introduite par le processus d'insertion des données, et fournissent une base de comparaison rigoureuse.

A. MAE (Erreur absolue moyenne) :

La MAE calcule l'erreur moyenne absolue entre l'image originale I et l'image stégoI′.Elle est définie par

$$\mathrm{MAE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} |I(i,j) - I'(i,j)|$$

Plus la MAE est faible, plus l'image stégo est proche de l'image originale.

B. MSE (MeanSquaredError)

Le MSE mesure l'erreur quadratique moyenne, c'est-à-dire la moyenne des carrés des différences entre les deux images :

$$\mathrm{MSE} = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - I'(i,j))^2$$

Une faible MSE indique une faible distorsion introduite par l'insertion du message.

C. PSNR (Peak Signal-to-Noise Ratio)

Le PSNR exprime le rapport entre la puissance maximale d'un signal (l'image originale) et le bruit introduit (différences avec l'image stégo), exprimé en décibels :

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{MAX^2}{\text{MSE}} \right)$$

où MAX est la valeur maximale possible d'un pixel (255 pour une image sur 8 bits).

Un PSNR supérieur à 40 dB est généralement considéré comme excellent (image quasi-indistinguable).

D. SSIM (Structural Similarity Index)

Le SSIM mesure la similarité perceptuelle entre deux images, en prenant en compte la luminance, le contraste et la structure locale :

$$\text{SSIM}(I, I') = \frac{(2\mu_I \mu_{I'} + C_1)(2\sigma_{II'} + C_2)}{(\mu_I^2 + \mu_{I'}^2 + C_1)(\sigma_I^2 + \sigma_{I'}^2 + C_2)}$$

où :

- μ_I , $\mu_{I'}$: moyennes locales
- σ_I^2 , $\sigma_{I'}^2$: variances locales
- $\sigma_{I.I'}$: covariance
- C1,C2 : constantes de stabilisation

Le SSIM varie entre -1 et 1, où 1 signifie une similarité parfaite.

E. Coefficient de corrélation (r)

Le coefficient de corrélation mesure la similarité linéaire entre les deux images :

$$r = \frac{\sum_{i,j}(I(i,j) - \mu_I)(I'(i,j) - \mu_{I'})}{\sqrt{\sum_{i,j}(I(i,j) - \mu_I)^2} \cdot \sqrt{\sum_{i,j}(I'(i,j) - \mu_{I'})^2}}$$

r proche de 1 signifie une forte corrélation (images très similaires), tandis qu'une valeur proche de 0 indique peu ou pas de relation.

3.4.6 Présentation des résultats

Le tableau suivant regroupe les résultats obtenus pour chaque image testée selon les cinq indicateurs mentionnés. Ces résultats permettent une comparaison quantitative de la qualité visuelle des images stégo par rapport aux originales :

Tableau 5 Résultats des tests

Image	MAE	MSE	PSNR (db)	SSIM	Corrélation
peppers.png	0.0015	0.0015	76.2763	1.0000	1.0000
onion.png	0.0111	0.0111	67.6781	1.0000	1.0000
fabric.png	0.0010	0.0010	78.1195	1.0000	1.0000
football.jpg	0.0036	0.0036	72.5177	1.0000	1.0000
université.jpg	0.0012	0.0012	77.4083	1.0000	1.0000

Les résultats obtenus témoignent d'une très faible distorsion entre les images originales et stéganographiées, avec des valeurs de MAE et MSE proches de zéro, un PSNR supérieur à 67 dB, et un SSIM égal à 1. Cela démontre l'efficacité du système proposé en matière d'imperceptibilité et de conservation de la qualité visuelle.

3.5 Adaptabilité du système aux environnements bruités et compressés

Bien que le système proposé montre d'excellentes performances en conditions normales, il peut être vulnérable à certaines perturbations comme le bruit, les compressions (JPEG) ou les erreurs de transmission. Pour renforcer sa robustesse, une amélioration envisageable consiste à intégrer un codage correcteur d'erreurs, tel que le codage de Hamming ou le codage Reed-Solomon, avant l'étape de chiffrement. Ces codes permettent de détecter et corriger automatiquement les erreurs introduites lors de la transmission ou de la manipulation de l'image, augmentant ainsi la fiabilité du système.

Par ailleurs, il est également possible d'adapter notre système au domaine fréquentiel, en conservant le même mécanisme général (compression Huffman, chiffrement, insertion contrôlée par séquences chaotiques), mais en insérant les bits du message dans les coefficients transformés de l'image, comme ceux issus de la DCT, DWT, ou FFT. Une telle adaptation offrirait une meilleure résistance aux attaques de compression ou aux transformations géométriques, tout en conservant l'efficacité du schéma proposé.

3.6 Conclusion :

Ce chapitre a présenté un système de stéganographie complet et autonome, combinant l’insertion LSB avec la compression Huffman, le chiffrement symétrique (RC4) et l’utilisation de cartes chaotiques (Henon et Logistique) pour une sélection sécurisée des positions d’insertion. L’ensemble du processus, de la compression du message jusqu’à son extraction, a été conçu pour garantir un bon équilibre entre imperceptibilité, sécurité et capacité d’insertion.

Les résultats expérimentaux valident l’efficacité de notre méthode, avec une excellente transparence visuelle, une stabilité statistique des histogrammes, et des performances quantitatives élevées (PSNR, SSIM, MAE, MSE, corrélation), confirmant la robustesse et la fiabilité du système proposé pour des applications nécessitant une forte confidentialité.

Conclusion

Générale

Conclusion Générale :

Dans un monde de plus en plus interconnecté, la sécurité de l'information représente un défi majeur pour les chercheurs, les ingénieurs et les utilisateurs. Les échanges de données, qu'ils soient personnels, professionnels ou institutionnels, exigent des garanties solides contre les menaces telles que l'espionnage, la falsification ou la fuite d'informations sensibles. Face à ces enjeux, la stéganographie offre une approche complémentaire et discrète, qui consiste non pas à brouiller l'information, mais à dissimuler son existence même.

Ce mémoire a permis de mener une étude approfondie des techniques de stéganographie d'images, en mettant en évidence les méthodes les plus utilisées, tant dans le domaine spatial que fréquentiel. Des techniques classiques comme le LSB ou la DCT ont été analysées, ainsi que des méthodes plus avancées intégrant des algorithmes chaotiques, des mécanismes de compression et de chiffrement. L'objectif n'était pas seulement théorique, mais aussi pratique : concevoir et implémenter un système de stéganographie hybride, combinant plusieurs techniques afin d'optimiser à la fois la sécurité, la capacité d'insertion et la qualité de l'image finale.

Les résultats expérimentaux ont montré que l'approche proposée atteint un bon équilibre entre imperceptibilité, robustesse et efficacité, tout en restant accessible en termes de complexité computationnelle. Grâce à l'intégration de la compression de Huffman, du chiffrement RC4 et de la génération chaotique, le système développé offre un niveau de sécurité renforcé et une adaptabilité aux différentes images hôtes.

En conclusion, cette étude confirme que la stéganographie, lorsqu'elle est intelligemment combinée à d'autres techniques de sécurité, constitue une solution puissante pour la protection de l'information dans les environnements numériques. Elle ouvre également la voie à des perspectives futures prometteuses, telles que l'utilisation de l'intelligence artificielle pour optimiser l'insertion, ou encore l'adaptation aux nouveaux formats multimédias émergents. Ce travail constitue ainsi une base solide pour de futures recherches et applications concrètes dans le domaine de la cybersécurité et de la confidentialité des données.

Bibliographie

Bibliographie :

1. **Buades, A., Coll, B., & Morel, J. M.** *A non-local algorithm for image denoising. International Journal of Computer Vision, 106(2), 199–212.* 2019.

2. **Pitas, I.** *Digital image processing algorithms and applications.* s.l. : Wiley, 2016.

3. **Elharrouss, O., Almaadeed, N., Al-Maadeed, S., & Akbari, Y.** *Image inpainting: A review. Neural Processing Letters, 53, 2005–2028.* 2021.

4. **Mukherjee, R., & Mitra, S.** *Edge detection techniques for image segmentation: A review. Journal of Electronic Imaging, 28(2), 023017.* 2019.

5. **Zhu, Y., & Liu, F.** *A comparative study on color image filtering techniques. Multimedia Tools and Applications, 80(1), 593–616.* 2021.

6. **Cheng, H. D., & Shi, X. J.** *A simple and effective histogram equalization approach to image enhancement. Digital Signal Processing, 20(7), 1808–1820.* 2015.

7. **Wang, W., Shen, J., & Porikli, F.** *Saliency-aware geodesic video object segmentation. IEEE Transactions on Image Processing, 26(4), 1950–1964.* 2017.

8. **Zhang, Y., & Yang, J.** *Deep learning-based image segmentation: A survey. Pattern Recognition, 108, 107359.* 2021.

9. **Zhang, L., Shen, Y., & Li, H.** *A review on image enhancement techniques. Signal Processing, 175, 107288.* 2020.

10. **Gonzalez, R. C., & Woods, R. E.** *Digitale image processing (4 th ed).* s.l. : pirson, 2018.

11. **Schneier, B.** *Applied cryptography: Protocols, algorithms, and source code in C (2nd ed.).* 2015. Wiley.

12. **Stinson, D. R., & Paterson, M. B.** *Cryptography: Theory and practice (4th ed.).* s.l. : CRC Press, 2018.

13. **Islam, M. T., & Hossain, M. M.** *A comparative analysis of image steganography techniques. International Journal of Computer Applications, 183(23), 10–15.* 2021.

14. **Bamatraf, A., Ibrahim, R., & Abdulla, S.** *A new LSB image steganography approach for hiding text in RGB images. International Journal of Computer Applications, 178(7), 11–17.* 2019.

15. **Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P.** *Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3), 727–752.* 2017.

16. **Husain, M. I., & Hussain, M. 2017.** *A survey on image steganography techniques. International Journal of Computer Applications, 143(10), 1–9.* 2017.

17. **Mielikainen, J.** *LSB matching revisited. IEEE Signal Processing Letters, 13(5), 285–287.* 2017.

18. **Luo, W., Huang, F., & Huang, J.** *Edge adaptive image steganography based on LSB matching revisited. IEEE Transactions on Information Forensics and Security, 5(2), 201–214.* 2019.

19. **Provos, N., & Honeyman, P.** *Hide and seek: An introduction to steganography. IEEE Security & Privacy, 1(3), 32–44.* 2015.

20. **Khan, A., Khan, S., & Sadiq, M. A.** *A novel hybrid image steganography technique for secure data hiding. Multimedia Tools and Applications, 79(25–26), 18149–18174.* 2020.

21. **Johnson, N. F., & Jajodia, S.** *Exploring steganography: Seeing the unseen. Computer, 31(2), 26–34.* 2016.

22. **Hussain, M., & Hussain, M.** *A survey of image steganography techniques. International Journal of Advanced Science and Technology, 54, 113–124.* 2016.

23. **Sayood, K.** *Introduction to data compression (5th ed.).* s.l. : Morgan Kaufmann, 2017.

24. **Kumar, P., & Bala, R.** *A secure image steganography based on RC4 encryption and DCT. International Journal of Information Technology, 10(4), 431–438.* 2018.

25. **Fu, C., Lin, Q., & Zou, H.** *A novel chaos-based image encryption using DNA sequence operation and hyper-chaotic system. Signal Processing, 128, 144–157.* 2017.

26. **Wang, X., & Zhao, H.** *Image encryption algorithm based on Henon chaotic map and RC4. Multimedia Tools and Applications, 78(13), 18721–18737.* 2019.

27. **Li, S., Mou, X., & Cai, Y.** *Pseudo-random bit generator based on couple chaotic systems and its application in stream-cipher cryptography. In Progress in Cryptology – INDOCRYPT, 316–329.* 2016.