2024/2025

Rapport du Projet de Sécurité Informatique

Amine Aichane & Hamza Chraim

Table des matières

I – INTRODUCTION

La stéganographie est l'art de dissimuler des informations dans un support numérique de manière à ce que seul le destinataire prévu puisse détecter et extraire ces informations. Contrairement au chiffrement, qui rend le message illisible tout en laissant paraître son existence, la stéganographie masque l'existence même de l'information. Elle s'inscrit dans une logique de sécurité par dissimulation plutôt que par transformation.

L'étymologie du mot vient du grec ancien : *steganos* (caché) et *graphein* (écrire). Elle remonte à l'Antiquité, où des messages étaient tatoués sur des têtes rasées puis recouverts par les cheveux repoussés. Avec le développement du numérique, cette pratique ancestrale a évolué pour intégrer des supports électroniques tels que les images, les vidéos, les fichiers audio ou encore les protocoles réseaux.

Aujourd'hui, la stéganographie numérique est particulièrement pertinente dans les domaines de la cybersécurité, de l'espionnage, de la protection des données personnelles, ainsi que dans la lutte contre la censure dans certains pays. Dans un contexte où la surveillance massive est devenue une réalité, l'importance de transmettre des messages sans attirer l'attention devient cruciale.

Le principe de base repose sur la modification imperceptible du support. Les images numériques, par exemple, possèdent une vaste quantité de données inutilisées aux yeux de l'utilisateur, notamment dans les bits de poids faible de chaque pixel. Ces bits peuvent être modifiés sans altérer visuellement l'image, rendant la dissimulation presque indétectable à l'œil nu.

Notre projet vise à exploiter cette technique, plus précisément le **LSB (Least Significant Bit)**, tout en intégrant une **approche innovante basée sur l'intelligence artificielle**. L'idée n'est pas seulement de cacher les données, mais de le faire intelligemment, en sélectionnant les zones les plus appropriées de l'image grâce à des techniques d'analyse automatique. Cela permet d'optimiser à la fois la sécurité, la discrétion et la robustesse de l'encodage.

Dans ce rapport, nous allons présenter les fondements théoriques et techniques de la stéganographie, décrire le système que nous avons développé, détailler son fonctionnement, analyser les défis rencontrés et proposer des pistes d'amélioration. Ce travail s'inscrit dans une démarche pédagogique, mais aussi expérimentale, visant à repousser les limites des approches traditionnelles.

II – Contexte Général

À l'ère du numérique, la prolifération des données personnelles, professionnelles et institutionnelles soulève d'importants enjeux de sécurité et de confidentialité. Qu'il s'agisse de communications interpersonnelles, d'échanges bancaires, de données médicales ou encore d'informations sensibles dans le cadre gouvernemental, la nécessité de protéger l'information devient un impératif constant. Si le chiffrement constitue un rempart efficace pour rendre les données inintelligibles aux tiers, il présente néanmoins un inconvénient majeur : il signale l'existence d'une communication confidentielle, ce qui peut éveiller des soupçons ou attirer l'attention de personnes malveillantes.

C'est ici qu'intervient la stéganographie, discipline complémentaire à la cryptographie, qui propose de **masquer la présence même de l'information**. Elle permet de dissimuler un message à l'intérieur d'un autre contenu – souvent banal – de façon à ce que seul un destinataire averti puisse détecter sa présence et l'extraire. Dans la pratique, les supports numériques les plus utilisés sont les images, les fichiers audio et vidéo, ou encore le trafic réseau. Parmi ces supports, l'image numérique offre une capacité et une flexibilité exceptionnelles.

Les images sont omniprésentes sur internet, partagées chaque jour sur les réseaux sociaux, les messageries et les sites web. Elles représentent donc un vecteur idéal pour dissimuler des données sans éveiller la moindre suspicion. En effet, l'œil humain est incapable de percevoir des modifications minimes dans les millions de pixels qui composent une image. En exploitant cette faiblesse perceptive, la technique dite du "Least Significant Bit" (LSB) s'impose comme un choix populaire dans le domaine.

Cependant, à mesure que les outils d'analyse et de détection progressent, les techniques de stéganographie doivent également évoluer. Les approches naïves ou trop simples sont aujourd'hui insuffisantes pour faire face aux systèmes de détection automatisés basés sur l'intelligence artificielle ou l'analyse statistique. C'est pourquoi il devient nécessaire de **développer des systèmes intelligents**, capables de s'adapter à la structure de l'image et d'optimiser les zones de dissimulation en fonction du contexte visuel.

Dans ce cadre, notre projet propose une méthode avancée qui combine la simplicité de la stéganographie LSB avec la puissance de l'analyse d'image assistée par IA. L'objectif est de **renforcer la robustesse et la discrétion** de la dissimulation, tout en facilitant l'usage grâce à une interface utilisateur conviviale. Cette approche vise également à démontrer comment des

technologies modernes peuvent être intégrées dans des méthodes classiques pour en améliorer la sécurité et la pertinence dans des contextes variés.

III – Problématique

La stéganographie, bien qu'efficace dans son principe fondamental, souffre de **limites structurelles** lorsqu'elle est implémentée de manière naïve. Les techniques classiques de type LSB (Least Significant Bit), très répandues dans les milieux académiques et amateurs, présentent plusieurs **failles critiques** qui compromettent la sécurité, la robustesse et l'efficacité globale du système.

Premièrement, la **vulnérabilité à la détection** constitue un enjeu majeur. Lorsqu'un message est caché de manière séquentielle dans les bits de poids faible d'une image, les modifications introduites peuvent, à l'échelle statistique, générer des motifs ou anomalies détectables par des outils d'analyse spécialisés. Des logiciels de stégano-analyse comme StegExpose ou des algorithmes d'apprentissage automatique peuvent détecter des déviations subtiles dans la distribution des bits. Cela rend les méthodes de LSB classiques peu adaptées à des contextes nécessitant un haut niveau de discrétion.

Deuxièmement, la **fragilité du support** est un autre facteur limitant. Une image contenant des données cachées peut perdre tout ou partie de ces informations lorsqu'elle est compressée (notamment en JPEG), redimensionnée, ou simplement modifiée. Puisque les bits modifiés sont souvent répartis de manière uniforme, ils peuvent être facilement altérés par une simple opération de traitement d'image. Cela rend les données dissimulées très sensibles aux manipulations même banales.

Troisièmement, la **capacité d'encodage limitée** représente un obstacle technique. Le LSB ne permet d'insérer qu'un faible volume de données, surtout si l'on souhaite préserver la qualité visuelle de l'image. Tout compromis en faveur de la capacité risque de détériorer l'apparence de l'image, ce qui va à l'encontre du principe fondamental de la stéganographie : rester invisible.

Enfin, un des problèmes les plus souvent négligés est **la sélection non pertinente des pixels**. Dans la plupart des implémentations basiques, les pixels sont sélectionnés séquentiellement ou aléatoirement, sans tenir compte du contenu de l'image. Cela revient à modifier aussi bien des zones uniformes (comme un ciel bleu ou un mur blanc) que des zones complexes, alors qu'il est bien plus sûr de concentrer les modifications dans des régions texturées, où l'œil humain comme les outils d'analyse auront plus de mal à percevoir les altérations.

Face à ces contraintes, il devient impératif d'explorer des solutions capables de dépasser les limites des approches traditionnelles. Notre projet cherche à répondre à ces problématiques de manière innovante et efficace. En particulier, nous avons choisi d'intégrer des techniques d'analyse d'image assistées par intelligence artificielle pour **détecter les zones les plus propices à la dissimulation**, c'est-à-dire des zones visuellement complexes et robustes aux manipulations.

Nous nous concentrons également sur l'amélioration de la robustesse face aux traitements courants en mettant en place des mécanismes de redondance, des terminators fiables, et en veillant à préserver la structure binaire des messages même en cas d'altération du support.

Ainsi, la problématique de notre projet peut se formuler de la manière suivante : **comment améliorer la sécurité, la robustesse et la capacité de la stéganographie LSB tout en assurant sa discrétion et sa compatibilité avec des images standards** ? La réponse que nous proposons repose sur la synergie entre techniques classiques et outils modernes d'analyse visuelle intelligente.

IV – Description du projet

Le projet que nous avons conçu repose sur la création d'un système de **stéganographie numérique intelligent** capable d'insérer et d'extraire des messages textuels dans des images, tout en garantissant discrétion, robustesse et accessibilité. Il s'appuie sur deux approches complémentaires : une implémentation classique de la méthode LSB (Least Significant Bit) et une version améliorée intégrant une **analyse d'image assistée par intelligence artificielle**. L'objectif principal est de proposer un outil sécurisé, fiable et simple d'utilisation pour dissimuler des messages sans attirer l'attention.

Le système est structuré autour de **deux modules principaux** :

1. **Le module de stéganographie classique**, qui implémente la technique LSB de manière séquentielle. Il permet d'insérer un message dans une image en modifiant le bit de poids faible des canaux de couleur des pixels. Cette méthode constitue la base du système, assurant un fonctionnement simple, compréhensible et rapide, adapté aux environnements à faible exigence de sécurité.
2. **Le module intelligent basé sur l'IA**, qui repose sur l'analyse du contenu visuel de l'image. Il utilise un algorithme de détection de contours (comme le filtre de Sobel) pour générer une **carte d'importance** indiquant les zones les plus riches en textures. Le message est alors inséré préférentiellement dans ces zones complexes, réduisant ainsi le risque de détection et de dégradation.

Le système est également doté d'une **interface utilisateur graphique intuitive** (GUI), développée avec des outils Python, permettant à des utilisateurs non spécialisés d'utiliser la solution sans avoir à interagir avec le code source. Cette interface offre les fonctions suivantes :

- **Chargement de l'image support** depuis le disque
- **Saisie du message à dissimuler**
- **Choix entre l'encodage standard ou IA**
- **Visualisation de l'image modifiée**
- **Sauvegarde et extraction du message caché**

Notre solution prend en charge différents formats d'image (notamment PNG et BMP, qui conservent fidèlement les données), et s'assure que le processus d'encodage ne détériore pas la qualité visuelle de l'image, grâce à l'ajout d'un **terminateur binaire (11111110)** indiquant

la fin du message. Ce mécanisme garantit une extraction précise même si la taille exacte du message n'est pas connue à l'avance.

Par ailleurs, le système permet une **sauvegarde automatique de l'image originale** avant encodage, afin de faciliter les comparaisons et les vérifications. Il a été conçu de manière modulaire : chaque composant (encodage, analyse IA, interface, sécurité) fonctionne de façon indépendante, ce qui simplifie les tests, les mises à jour et les éventuelles évolutions futures du projet.

En résumé, notre projet est une solution hybride et complète qui **combine simplicité, accessibilité et sophistication technique**. Il constitue une avancée par rapport aux implémentations de stéganographie classiques, tout en restant adaptable à différents niveaux d'exigence. Il peut aussi bien servir de **support pédagogique** pour illustrer les principes fondamentaux de la stéganographie que de **base de développement** pour des applications plus avancées, dans des contextes de sécurité réels.

V – Fonctionnement du projet

Le fonctionnement du système repose sur une **architecture modulaire et hiérarchisée** permettant une dissimulation d'information efficace tout en restant accessible et facilement maintenable. L'ensemble du processus est divisé en deux grandes étapes : **l'encodage** (insertion du message dans l'image) et **le décodage** (extraction du message), chacune étant disponible en version **standard (LSB simple)** ou **améliorée (avec analyse IA)**. Le système intègre plusieurs fichiers Python spécialisés, interconnectés, pour assurer une exécution fluide et cohérente.

1. Architecture générale du système

Le projet est structuré autour des modules suivants :

- **lsb_encoder.py** : effectue l'encodage classique en insérant les bits du message dans les bits de poids faible du canal rouge de chaque pixel.
- **lsb_decoder.py** : extrait les bits cachés dans l'image selon la même logique.
- **ai_analyzer.py** : génère une carte d'importance à l'aide de la détection de contours (algorithme de Sobel), afin d'identifier les zones de l'image les plus propices à la dissimulation.
- **security.py** : ajoute des fonctionnalités de sécurité supplémentaires, telles que l'ajout d'un identifiant unique, de hachage ou d'encodage du message.
- **utils.py** : fournit des fonctions utilitaires comme la conversion texte-binaire, l'insertion de terminateur, ou la gestion des formats d'image.
- **app.py** : compose l'interface graphique du système et gère les interactions utilisateur.

2. Processus d'encodage

Encodage standard :

- L'image est chargée et, si nécessaire, convertie en mode RGB.
- Le message saisi est transformé en une chaîne binaire.
- Chaque bit est inséré dans le bit de poids faible du canal rouge des pixels, un par un, en suivant un parcours linéaire.
- Un terminateur spécial (`11111110`) est ajouté à la fin du message.
- L'image modifiée est sauvegardée sur le disque.

Encodage intelligent (IA) :

- Une **carte d'importance** est calculée via détection de contours : les zones riches en variations (bords, textures) sont identifiées.
- Les pixels sont triés selon cette importance : plus une zone est détaillée, plus elle est priorisée pour le stockage.
- Le message binaire est inséré dans ces zones de forte densité visuelle.
- Ce procédé rend les modifications bien plus **difficiles à détecter**, même par des outils automatisés.

3. Processus de décodage

Décodage standard :

- L'image suspecte est chargée.
- Les bits de poids faible du canal rouge sont extraits de manière séquentielle.
- Le système s'arrête automatiquement lorsque le terminateur est détecté.
- Les bits sont convertis en texte lisible par l'utilisateur.

Décodage intelligent (IA) :

- Une nouvelle carte d'importance est générée (la même méthode que lors de l'encodage).
- Les pixels sont triés dans l'ordre exact utilisé pour l'encodage.
- Les bits sont extraits en suivant cet ordre spécifique.
- En cas d'erreur (ex. : altération de l'image), un **mode de secours** permet de tenter un décodage standard.

4. Interface et sécurité

Le système est prévu pour fonctionner sur des machines de bureau classiques. L'interface utilisateur guide l'usager à travers toutes les étapes (chargement, saisie du message, choix de la méthode, sauvegarde). Un **système de journalisation** permet de suivre les opérations effectuées et d'assurer un meilleur débogage. En outre, le module de sécurité peut être activé pour ajouter une couche de protection (hachage, identifiant, vérification d'intégrité).

VI – Réalisation

La phase de réalisation du projet a été structurée en plusieurs étapes méthodiques, de l'analyse des besoins techniques à l'implémentation concrète des différentes fonctionnalités. Le développement a été effectué en langage Python, en tirant parti de bibliothèques puissantes pour la manipulation d'images, le traitement des signaux visuels, l'intelligence artificielle légère et l'interface graphique. Chaque étape a fait l'objet de tests rigoureux, dans un souci de performance, de compatibilité et de simplicité d'usage.

1. Technologies utilisées

- **Python** : Langage principal pour sa souplesse, sa lisibilité et la richesse de ses bibliothèques.
- **PIL (Pillow)** : Manipulation et traitement de base des images (chargement, conversion RGB, lecture et modification de pixels).
- **OpenCV (cv2)** : Traitement avancé d'image, notamment pour les filtres, la détection de contours et la génération de cartes d'importance.
- **NumPy** : Manipulations numériques sur les tableaux de pixels, essentiel pour optimiser les performances.
- **scikit-image** : Utilisé pour les opérations spécifiques de traitement d'image, comme l'implémentation du filtre de Sobel.
- **Tkinter** : Création de l'interface utilisateur graphique, légère et intégrée dans Python par défaut.

2. Étapes de développement

1. **Conception initiale du système** : Définition des modules nécessaires, identification des formats d'image à supporter et rédaction des spécifications techniques de chaque composant.
2. **Développement du module LSB standard** : Implémentation du premier niveau d'encodage et de décodage par substitution du bit de poids faible, avec ajout d'un terminateur binaire fiable.
3. **Ajout du module IA** : Création d'un analyseur de contours basé sur le filtre de Sobel, développement de la logique de tri des pixels par importance, et intégration dans le processus d'encodage.

4. **Mise en place de l'interface graphique** : Développement de menus simples pour permettre aux utilisateurs de charger une image, saisir un message, sélectionner le mode de dissimulation et sauvegarder le résultat.
5. **Développement du module de sécurité** : Ajout de fonctionnalités facultatives comme l'identification du message, l'intégrité des données ou l'encodage des chaînes avant insertion.
6. **Tests fonctionnels** : Vérification du bon fonctionnement de chaque composant sur différents types d'images (résolution, format, niveau de complexité visuelle), validation de la compatibilité entre encodage et décodage, et mesure du taux de réussite.
7. **Tests de robustesse** : Simulation d'altérations sur les images encodées (compression, redimensionnement, rotation) pour observer l'impact sur l'extraction du message. Des cas de stress ont été utilisés pour évaluer les limites du système.
8. **Optimisation et refactoring** : Réduction de la redondance dans le code, amélioration des performances, et documentation interne pour assurer la maintenance.

3. Défis rencontrés

Plusieurs obstacles techniques et logiques ont été identifiés et résolus au cours du projet :

- **Alignement entre encodage et décodage IA** : Il était essentiel de garantir que l'ordre de tri des pixels soit strictement identique lors de l'encodage et du décodage. Un simple décalage aurait rendu le message illisible.
- **Détection fiable du terminateur** : Le terminateur binaire devait être suffisamment distinct pour ne pas être confondu avec le contenu du message.
- **Compatibilité des formats** : Certaines images compressées perdaient les données LSB. Nous avons restreint l'usage à des formats sans perte comme PNG.
- **Équilibre entre performance et qualité** : L'analyse IA demande un traitement supplémentaire. Nous avons dû optimiser le code pour éviter les ralentissements sur des images de grande taille.

VII – Conclusion

Le projet de stéganographie que nous avons réalisé s'inscrit dans une volonté de dépasser les limitations des techniques classiques d'insertion de données dans des images numériques, en combinant des approches éprouvées comme le LSB avec des outils modernes d'analyse d'image. À travers ce travail, nous avons conçu, implémenté et testé un système capable de dissimuler efficacement des messages textuels dans des images, tout en assurant une extraction fiable et une robustesse renforcée face aux attaques d'analyse.

Parmi les principales avancées du projet, on peut citer l'**utilisation de cartes d'importance générées par détection de contours**. Cette approche permet une dissimulation adaptative, concentrée dans les zones visuellement complexes de l'image, là où les modifications sont le moins détectables. Cette idée a été au cœur de notre démarche, combinant logique de sécurité et optimisation visuelle. Ce choix technique s'est avéré pertinent, notamment dans les tests de résistance aux outils de stégano-analyse.

Un autre point fort réside dans la **conception modulaire du système**, qui permet de séparer clairement les responsabilités de chaque composant : encodage, décodage, analyse IA, interface utilisateur, et sécurité. Cette organisation a grandement facilité le développement, les tests et la maintenance. Elle ouvre également la voie à des améliorations futures sans nécessiter une refonte complète de l'architecture.

D'un point de vue pédagogique, ce projet nous a permis de mettre en pratique des notions clés de programmation Python, de traitement d'image, d'intelligence artificielle et d'ergonomie logicielle. Il a aussi renforcé notre compréhension des enjeux liés à la confidentialité de l'information dans un contexte numérique omniprésent. La dimension pratique du projet, couplée à une réflexion constante sur les risques et les contraintes, a contribué à faire de cette expérience un véritable **exercice d'ingénierie logicielle appliquée à la cybersécurité**.

Bien que le système actuel soit pleinement fonctionnel, plusieurs **perspectives d'amélioration** s'ouvrent à nous. Nous envisageons notamment l'intégration de mécanismes de chiffrement complémentaires pour protéger le message avant sa dissimulation. De plus, l'usage de techniques de machine learning plus avancées pourrait permettre de détecter automatiquement les motifs visuels les plus adaptés, voire d'échapper à des algorithmes de détection entraînés. Enfin, l'extension du système à d'autres types de médias – comme l'audio ou la vidéo – représenterait un défi technique ambitieux, mais très pertinent dans le cadre d'un projet de recherche ou d'un produit de sécurité avancé.

En conclusion, ce projet nous a permis de comprendre les **limites, les forces et les enjeux contemporains** de la stéganographie. Nous avons montré qu'il était possible, avec des outils open-source et une démarche rigoureuse, de concevoir un système complet et performant, répondant aux besoins de sécurité dans un monde de plus en plus numérisé. Ce travail représente une base solide pour des développements futurs dans le domaine de la dissimulation d'information intelligente.