Thèse : Système de Cryptage Visuel et Stéganographie par Mimétisme Géométrique

Auteur : Adam Planque

Date : 24 Décembre 2025

Sujet : VisualEncryption - Une approche nouvelle de la dissimulation de données par reconstruction d'images procédurales.

Résumé

Ce document présente une méthode innovante de cryptage visuel et de stéganographie. Contrairement aux méthodes traditionnelles qui cachent des données dans les bits de poids faible d'une image existante (LSB), notre approche *construit* une image entièrement à partir des données cryptées. Le système transforme des données brutes en pixels colorés, puis réorganise ces pixels pour former des motifs géométriques cohérents (cercle, carré, triangle) avec des gradients de luminosité spécifiques. Cette technique permet de produire des images visuellement structurées et esthétiques tout en stockant une charge utile de données arbitraire, récupérable uniquement via une clé maîtresse de reconstruction.

---

1. Introduction

1.1 Contexte

La protection et la dissimulation de données sont des enjeux majeurs de la cybersécurité moderne. La stéganographie, l'art de dissimuler l'existence même d'un message, complète souvent la cryptographie traditionnelle. Cependant, la plupart des techniques stéganographiques nécessitent un "support" (une image de couverture) et altèrent ce support, ce qui peut parfois être détecté par analyse statistique.

1.2 Problématique

Comment transformer des données arbitraires (texte, fichiers binaires) en une représentation visuelle qui ne ressemble pas à du "bruit" aléatoire suspect, sans pour autant dépendre d'une image source externe ?

1.3 Objectif

L'objectif de cette thèse est de détailler le système **VisualEncryption**, qui convertit des données en une "soupe de pixels" puis utilise un algorithme de tri spatial pour forcer ces pixels à adopter une forme géométrique reconnaissable (Cercle, Carré ou Triangle).

---

2. Fondements Théoriques

2.1 Représentation des Données en Couleurs

Tout fichier numérique est une suite d'octets. En convertissant ces octets en valeurs hexadécimales, nous pouvons mapper chaque valeur ou groupe de valeurs à une couleur spécifique dans l'espace RGB.

- Exemple simple : `0xFF` -> Rouge, `0x00` -> Noir.
- Cette étape transforme l'information abstraite en information visuelle brute (bruit coloré).

2.2 Stéganographie Générative

Notre approche relève de la stéganographie générative. Au lieu de modifier une image existante, nous générons l'image de toutes pièces. L'image *est* la donnée. La sécurité repose sur la permutation secrète des positions des pixels.

---

3. Méthodologie et Architecture

Le processus se divise en deux phases principales : l'Encryption (Génération) et la Décryption (Reconstruction).

3.1 Phase 1 : Conversion Données-Couleurs

1. **Lecture** : Le fichier source est lu en binaire.
2. **Hexadécimal** : Conversion en chaîne hexadécimale.
3. **Mapping** : Chaque unité de donnée est associée à une couleur via une palette prédéfinie.
    - *Résultat intermédiaire* : Une image de "bruit" où les pixels sont placés séquentiellement. L'entropie visuelle est maximale.

3.2 Phase 2 : Génération de la Cible (Rotation de Formes)

Pour éviter l'aspect suspect du bruit aléatoire, le système impose une structure visuelle. Il utilise un cycle de rotation entre trois formes primitives :

1. **Cercle Dégradé**
2. **Carré Dégradé**
3. **Triangle Dégradé**

Logique du Dégradé : Chaque forme est générée mathématiquement avec une contrainte de luminosité stricte :

- **Centre** : Pixels clairs (haute luminosité).
- **Extérieur** : Pixels sombres (basse luminosité).

Cette structure prévisible sert de "moule" pour l'étape suivante.

3.3 Phase 3 : Algorithme de Mimétisme (Rearrangement)

C'est le cœur du système. L'algorithme ne modifie pas la *valeur* (couleur) des pixels de données (ce qui corromprait les données), mais modifie leur *position*.

1. **Analyse** : Le système analyse la luminosité de chaque pixel de données généré en Phase 1.
2. **Tri** : Les pixels de données sont triés par luminosité.

3. **Mapping Spatial** : Les positions de l'image cible (la forme géométrique) sont également classées par luminosité théorique (du centre vers les bords).
4. **Assignment** : Les pixels de données les plus clairs sont déplacés vers les coordonnées centrales de la forme. Les pixels les plus sombres sont déplacés vers les bords.

Résultat : Une image qui contient 100% des données originales, mais qui ressemble visuellement à la forme géométrique choisie.

3.4 La Clé Maîtresse (Master Key)

Puisque les pixels ont été mélangés, l'ordre original est perdu. Une clé de décryptage est générée simultanément. Elle contient :

- **La Carte de Permutation** : Un tableau reliant la position finale `(x, y)` à l'index original `i` dans le flux de données.
- **Métadonnées** : Nom du fichier original, taille, type MIME.

---

4. Scénario d'Utilisation

Prenons l'exemple d'un fichier texte contenant le mot "SECRET".

1. **Conversion** : "SECRET" devient une série de couleurs (ex: 6 pixels).
2. **Rotation** : Le système choisit la forme "Triangle".
3. **Génération Cible** : Un triangle virtuel est créé en mémoire, lumineux au centre.
4. **Rearrangement** :
    - Le pixel correspondant à 'E' (supposons qu'il est jaune clair) est déplacé au centre du triangle.
    - Le pixel correspondant à 'S' (supposons qu'il est bleu foncé) est déplacé vers une pointe du triangle.
5. **Output** : Une image BMP d'un triangle pixelisé.
6. **Clé** : Un fichier `.key` est sauvegardé pour permettre de remettre le pixel 'E' et 'S' à leur place lors de la lecture.

---

5. Analyse de Sécurité et Limitations

5.1 Forces

- **Discrétion Visuelle** : Les données ne ressemblent pas à du bruit cryptographique standard, mais à de l'art abstrait ou des icônes géométriques.
- **Indépendance** : Pas besoin de transporter une image source originale.

5.2 Faiblesses Actuelles

- **Substitution Simple** : Si un attaquant comprend que `Couleur X = Lettre A`, il peut lire le contenu simplement en analysant l'histogramme des couleurs, peu importe la position des pixels.
- **Dépendance à la Clé** : La sécurité repose entièrement sur la clé de permutation. Si la clé est interceptée, le fichier est trivial à reconstruire.

5.3 Améliorations Futures (Recommandées)

Pour un usage en production, il est impératif d'ajouter une couche de chiffrement standard (AES-256) **avant** la conversion en couleurs.

- *Flux Sécurisé* : Données -> AES -> Hex -> Couleurs -> Forme. Ainsi, même si l'attaquant "lit" les couleurs, il n'obtient que du cyphertext AES indéchiffrable sans la clé AES.

---

6. Conclusion

Le système **VisualEncryption** propose une approche originale de la manipulation de données. En traitant les données comme de la matière première visuelle (pixels) et en utilisant des algorithmes de tri pour les sculpter en formes géométriques (Cercles, Carrés, Triangles), nous obtenons un objet numérique hybride : à la fois image visible et conteneur de données. Bien que nécessitant un chiffrement préalable pour une sécurité militaire, cette méthode offre une solution élégante pour le stockage discret ou le transport ludique d'informations.

Document généré pour le projet VisualEncryption - PoC v1.0