La stéganographie : quand des secrets se glissent incognito

Guillaume Boudarham
Docteur en sciences physiques
Diplôme universitaire en criminalistique
Directeur du Laboratoire pluridisciplinaire de criminalistique

La stéganographie est une pratique ancienne qui consiste à dissimuler des informations à l'intérieur d'un support, comme un texte, une image, un fichier audio ou vidéo. Cet article offre une exploration de l'évolution de la stéganographie, depuis l'Antiquité jusqu'à nos jours. Il présente les fondements d'une image numérique et expose en détail une méthode de stéganographie récemment explorée par l'auteur lors d'une expertise privée qui permet de dissimuler une image à l'intérieur d'une autre image.

La stéganalyse consiste à détecter, analyser et décoder les informations dissimulées ou les techniques de stéganographie utilisées dans un médium de couverture afin de déterminer s'il contient des données cachées. Contrairement à la cryptographie, où le message est chiffré pour le rendre illisible, la stéganographie se focalise sur la dissimulation du message dans un support apparent [1-5]. Ce procédé remonte à l'Antiquité et a été utilisé de différentes manières au fil des siècles pour transmettre des messages secrets ou sensibles de manière discrète.

L'historien grec Hérodote relate dans ses Histoires (440 av. J.-C.) deux récits qui illustrent cette pratique [6]. Dans le premier récit, il raconte qu'au VIe siècle av. J.-C., Histiée utilisa une méthode singulière pour transmettre un message confidentiel à Aristagoras, dans le but de déclencher une révolte. Soucieux de préserver absolument le secret de cette instruction, Histiée fit raser complètement le crâne de son serviteur le plus fidèle et y inscrivit le message. Une fois que les cheveux de son messager avaient repoussé, Histiée l'envoya vers Aristagoras avec pour consigne de se raser à nouveau le crâne une fois arrivé à destination.

Dans le deuxième récit, Hérodote relate comment Démarate, roi de Sparte, put avertir les Grecs d'une invasion perse en faisant parvenir un message gravé sur une tablette en bois, ensuite recouverte d'une cire qui semblait vierge. Au Moyen Âge, les alchimistes ont également utilisé des techniques de stéganographie pour dissimuler des informations. Ils ont incorporé des symboles cachés dans leurs illustrations afin de transmettre des connaissances secrètes à d'autres initiés.

À la Renaissance, le moine et cryptographe allemand Johannes Trithemius a rédigé en 1499 un ouvrage intitulé *Steganographia*. Dans cet ouvrage, il expose différentes méthodes permettant de dissimuler des messages au sein de textes ordinaires. Bien que son livre ait été principalement centré sur des thèmes magiques et occultes, il a jeté les fondements de la stéganographie moderne.

Au XIXe siècle, dans leur correspondance amoureuse, le poète Alfred de Musset et la romancière George Sand ont utilisé un moyen subtil pour communiquer leurs désirs érotiques : l'acrostiche. Ils se sont envoyé des lettres contenant des poèmes dans lesquels les premiers mots de chaque vers, lus verticalement, révélaient un message caché [7].

Au cours des guerres mondiales du XXe siècle, la stéganographie a connu un essor particulier en tant qu'outil de communication clandestine. Les espions utilisaient des techniques de stéganographie pour dissimuler des informations sensibles dans des photographies, des lettres ou des objets du quotidien. Des microfilms étaient cachés à l'intérieur de stylos ou de boutons, et des messages étaient codés à l'aide d'encre invisible.

De nos jours, avec l'avènement de l'informatique, la stéganographie moderne offre de multiples possibilités, telles que cacher un message dans une image, un fichier audio ou vidéo. Elle est également utilisée dans des applications commerciales telles que le tatouage numérique pour l'identification des origines et la protection contre la duplication non autorisée.

En 2004, l'Electronic Frontier Foundation (EFF), une organisation à but non lucratif dédiée à la protection des droits numériques et de la vie privée, a mené

des recherches sur une forme de marquage numérique utilisée par certaines imprimantes laser et photocopieurs couleur. L'EFF a analysé les impressions provenant d'imprimantes de différents fabricants et a découvert l'existence de petits points jaunes microscopiques présents sur les pages imprimées [8]. Ces points n'étaient pas perceptibles à l'œil nu mais ils devenaient visibles grâce à une lumière bleue ou lorsqu'on agrandissait la page. Ils étaient disposés de façon à créer un motif particulier qui se répétait sur la page imprimée et renfermait des informations codées. Ces informations peuvent inclure des données telles que le numéro de série de l'imprimante, la date et l'heure de l'impression, ainsi que d'autres identifiants spécifiques à l'imprimante. Cette pratique a été mise en place par Xerox au milieu des années 1980 sur leurs photocopieurs couleur pour lutter contre la contrefaçon et le faux-monnayage.

Dans le contexte d'une cyber-attaque, la stéganographie peut également être utilisée pour dissimuler des logiciels malveillants.

1. REPRÉSENTATIONS D'UNE IMAGE NUMÉRIQUE

Une image est une représentation visuelle ou graphique d'un objet, d'une scène ou d'une idée. Dans le contexte numérique, une image est une composition de pixels, qui sont de petits éléments d'affichage qui combinés ensemble créent une représentation visuelle. Chaque pixel est associé à des informations de couleur qui déterminent sa teinte, sa luminosité et sa saturation [9].

La profondeur de bits est le nombre de bits utilisés pour représenter la couleur d'un pixel dans une image. Un bit est l'unité de base de l'information en informatique. Il représente la plus petite unité de stockage de données et peut prendre deux valeurs distinctes : 0 ou 1. Une plus grande profondeur de bits permet une plus grande échelle de nuances dans les couleurs. Une image binaire ou deux tons est composée de pixels qui utilisent 1 bit chacun pour représenter deux teintes, généralement le noir et le blanc. Le bit 0 est utilisé pour représenter le noir, tandis que le bit 1 est utilisé pour représenter le blanc. Une image en niveaux de gris est créée à partir de pixels qui stockent plusieurs bits d'informations, généralement entre 2 et 8 bits, voire davantage. Le nombre de bits détermine le nombre de teintes différentes qui peuvent être représentées. Par exemple, un pixel avec une profondeur de 2 bits peut représenter 4 teintes distinctes, correspondant aux quatre combinaisons possibles : 00, 01, 10 et 11.

Si nous associons la valeur 00 au noir et la valeur 11 au blanc, alors 01 correspondra à un gris foncé et 10 à un gris clair. En augmentant la profondeur de bits à 8, un pixel peut représenter 256 nuances différentes (2^8). Chaque valeur de 8 bits correspondra à une nuance de gris spécifique dans cette gamme, allant du noir au blanc et avec une variété de nuances de gris intermédiaires. Une image en couleur est une représentation visuelle composée de différentes couleurs. Le modèle de couleur RVB (rouge, vert, bleu) est l'un des modèles les plus couramment utilisés pour représenter les images en couleur sur les écrans d'ordinateur, les téléviseurs et autres dispositifs d'affichage. Dans ce modèle, chaque pixel de l'image est composé de trois composantes de couleur : rouge (R), vert (V) et bleu (B). Ces images sont généralement représentées avec une profondeur de bits allant de 8 à 24 bits, voire plus. Dans une image de 24 bits, la représentation des couleurs se fait en répartissant les bits en trois groupes distincts : 8 bits pour le rouge, 8 bits pour le vert et 8 bits pour le bleu.

Chaque composante est donc représentée par un nombre entier compris entre 0 et 255, où 0 représente l'absence de cette couleur et 255 représente l'intensité maximale de cette couleur. En combinant différentes intensités de rouge, de vert et de bleu, il est possible de créer un large spectre de couleurs. Par exemple, si les composantes rouge, vert et bleu sont toutes définies à leur intensité maximale (255, 255, 255), le pixel apparaîtra comme blanc. Si toutes les composantes sont définies à leur intensité minimale (0, 0, 0), le pixel apparaîtra comme noir. Un pixel rouge pur aura des valeurs RVB de (255, 0, 0), tandis qu'un pixel violet aura des valeurs RVB de (128, 0, 128) (voir figure 1).

2. LES BASES DÉCIMALE ET BINAIRE

La représentation décimale d'un nombre est couramment utilisée dans la vie quotidienne pour compter, effectuer des calculs et représenter des quantités. Elle utilise une base de 10, ce qui signifie qu'elle utilise 10 chiffres différents, de 0 à 9. Chaque position dans un nombre décimal représente une puissance de 10. Par exemple, dans le nombre décimal 123, le chiffre 3 représente la valeur de 3 fois 10^0, le chiffre 2 représente la valeur de 2 fois 10^1, et le chiffre 1 représente la valeur de 1 fois 10^2 ($123 = 1\times10^2 + 2\times10^1 + 3\times10^0$).

La représentation binaire est utilisée dans les systèmes informatiques pour représenter et manipuler l'information sous forme de bits. Elle utilise une base de 2, ce qui signifie qu'elle utilise uniquement deux chiffres, 0 et 1. Chaque position dans un nombre binaire représente une puissance de 2. Par exemple, dans le nombre binaire 10110, le dernier chiffre 0 représente la valeur de 0 fois 2^0, le quatrième chiffre 1 représente la valeur de 1 fois 2^1, le troisième 1 représente la valeur de 1 fois 2^2, le deuxième chiffre 0 représente la valeur de 0 fois 2^3 et le premier chiffre 1 représente la valeur de 1 fois 2^4 ($10110 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 0\times2^0 = 22$).

3. PRINCIPE DE LA STÉGANOGRAPHIE PAR SUBSTITUTION DE LSB (LSB REPLACEMENT)

La stéganographie par substitution de LSB (Least Significant Bit) est une

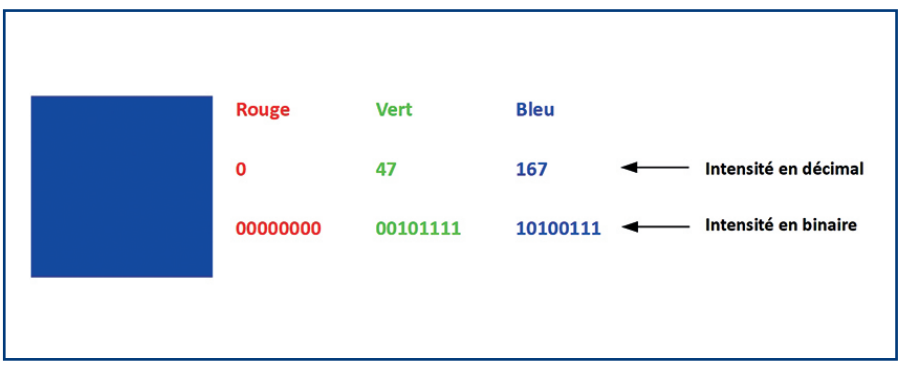

Figure 1 - Code couleur du bleu Klein en décimal et binaire.

technique facile à mettre en œuvre à l'aide de l'outil informatique permettant de dissimuler une image confidentielle à l'intérieur d'une autre image appelée "couverture" [10-12].

L'idée est d'utiliser les bits de poids faible de la couverture pour stocker les bits de poids fort de l'image à cacher. Cette méthode est utilisée pour minimiser les perturbations visuelles de la couverture, qui peut alors être transmise à un tiers sans éveiller les soupçons, car les bits de poids faible ont moins d'impact sur la perception visuelle de l'image. Les bits de poids faible (LSB - Least Significant Bit) sont les bits situés à l'extrémité droite d'un nombre binaire. Ils ont la valeur la moins significative et contribuent le moins à la valeur totale du nombre. Les bits de poids fort (MSB - Most Significant Bit) sont les bits situés à l'extrémité gauche d'un nombre binaire. Ils ont la valeur la plus significative et contribuent le plus à la valeur totale du nombre. Par exemple, dans le nombre binaire 10110, le premier 1 est le bit de poids le plus fort et le dernier 0 est le bit de poids le plus faible.

4. MISE EN ŒUVRE DÉTAILLÉE

4.1. Dissimulation d'une image secrète

Dans cet exemple simplifié, nous allons considérer deux images composées chacune d'un seul pixel. L'image A représente notre pixel de couverture, tandis que l'image B représente notre pixel à cacher. Pour cacher le pixel B dans le pixel A, nous allons remplacer le dernier bit (bit de poids le plus faible) du pixel A par le premier bit (bit de poids le plus fort) du pixel B pour chaque composante (rouge, vert, bleu). Cela signifie que le bit le moins significatif du pixel A sera modifié en utilisant le bit le plus significatif du pixel B. Les autres bits du pixel A resteront inchangés ce qui va permettre de dissimuler l'information contenue dans le pixel B au sein du pixel A (voir figure 2). Comme prévu, le pixel A modifié qui dissimule le pixel B présente une apparence pratiquement indiscernable de celle du pixel A de couverture. Nous montrons en figure 3 le résultat obtenu en dissimulant une image multicolore de quatre pixels dans une image de même taille (2×2) dont la couleur a été choisie uniforme dans un but pédagogique.

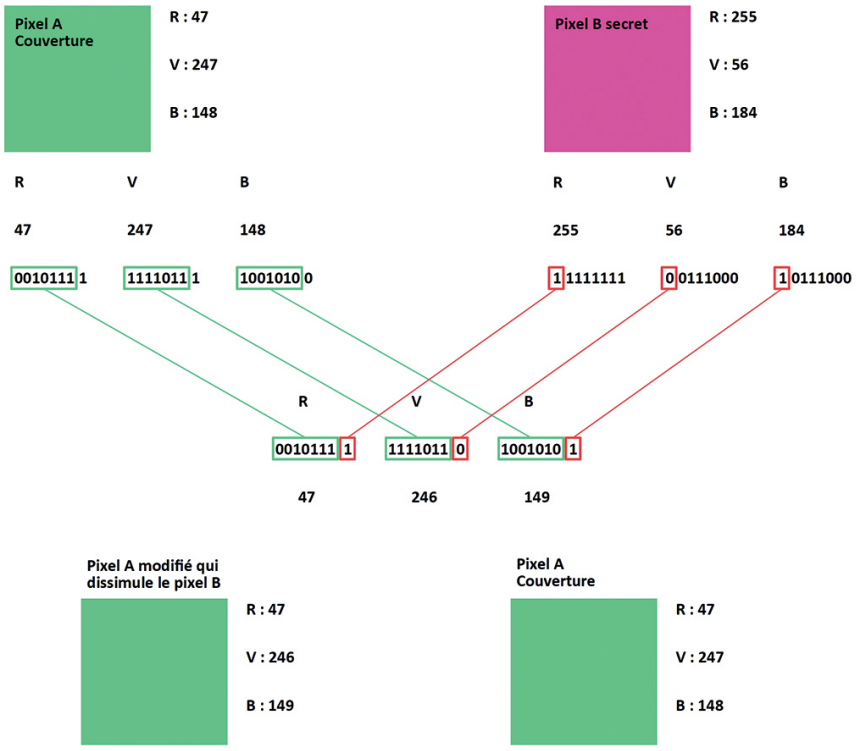

Figure 2 - Dissimulation du pixel B à l'intérieur du pixel A par stéganographie LSB.

Nous invitons les lecteurs téméraires à retrouver ces résultats par eux-mêmes. Encore une fois, l'image finale qui dissimule l'image secrète est pratiquement indiscernable de l'image de couverture. Ainsi, elle peut être transmise à un tiers dans un format préservant l'intégrité des données sans éveiller de soupçons, et ce tiers pourra ensuite révéler l'image dissimulée.

4.2. Révélation de l'image secrète

Pour dévoiler l'image cachée, la tierce personne qui détient l'image finale doit générer une nouvelle image distincte

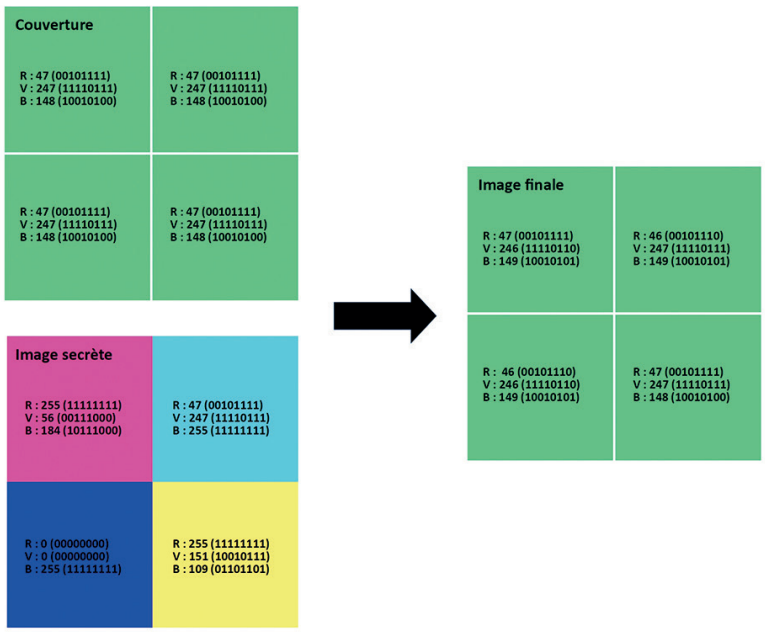

Figure 3 - Dissimulation d'une image secrète multicolore à l'intérieur d'une image de couverture de couleur uniforme par stéganographie LSB.

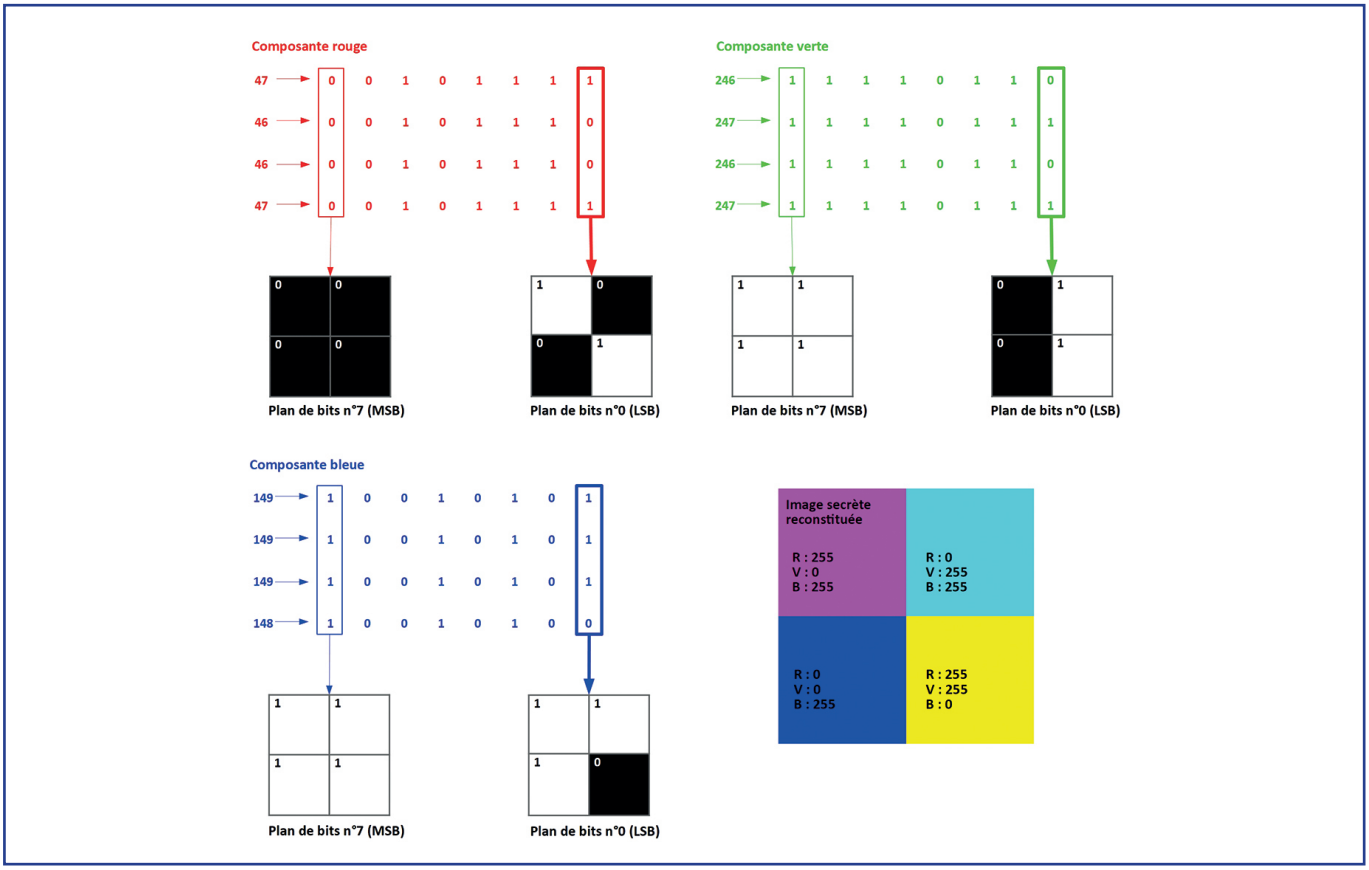

Figure 4 - Extraction des plans de bits pour reconstituer l'image secrète.

pour chaque composante (rouge, vert, bleu) de cette dernière.

Dans chacune de ces images, chaque pixel est représenté par le bit le plus à droite (plan de bits n°0) associé au pixel de même position dans l'image finale. Un pixel étant alors représenté par un seul bit (0 ou 1), nous obtenons trois images binaires (noir et blanc). Les pixels avec une valeur de 0 seront représentés en noir, tandis que les pixels avec une valeur de 1 seront représentés en blanc. Pour révéler l'image cachée, il suffit ensuite de combiner les trois images précédentes en associant aux valeurs 0 et 1 une intensité nulle (0) et maximale (255) respectivement pour la composante concernée.

Figure 5 - Image de couverture servant à dissimuler une image secrète.

Nous montrons en figure 4 le résultat obtenu à partir de l'image finale précédente. À titre indicatif, nous présentons les images binaires obtenues à partir des bits les plus à gauche (plan de bits n°7) et les plus à droite (plan de bits n°0), bien qu'il suffise de combiner les images obtenues en considérant uniquement les bits les plus à droite pour dévoiler l'image cachée. Nous observons que l'image reconstituée est extrêmement similaire à l'image initiale dissimulée.

5. APPLICATION

En utilisant un code développé dans Matlab[1], l'auteur a effectué une substitution LSB pour dissimuler une image secrète (2560 × 1440) à l'intérieur de l'image d'un chaton servant de couverture (voir figure 5). Afin de révéler l'image cachée, comme le ferait une tierce personne ayant récupéré l'image modifiée, l'auteur a employé un autre code dans Matlab afin d'extraire les plans de bits de 0 à 7 pour chaque composante (rouge, vert, bleu). Les images binaires correspondantes sont présentées de gauche à droite dans les tableaux ci-dessous par ordre décroissant des plans de

Composante rouge

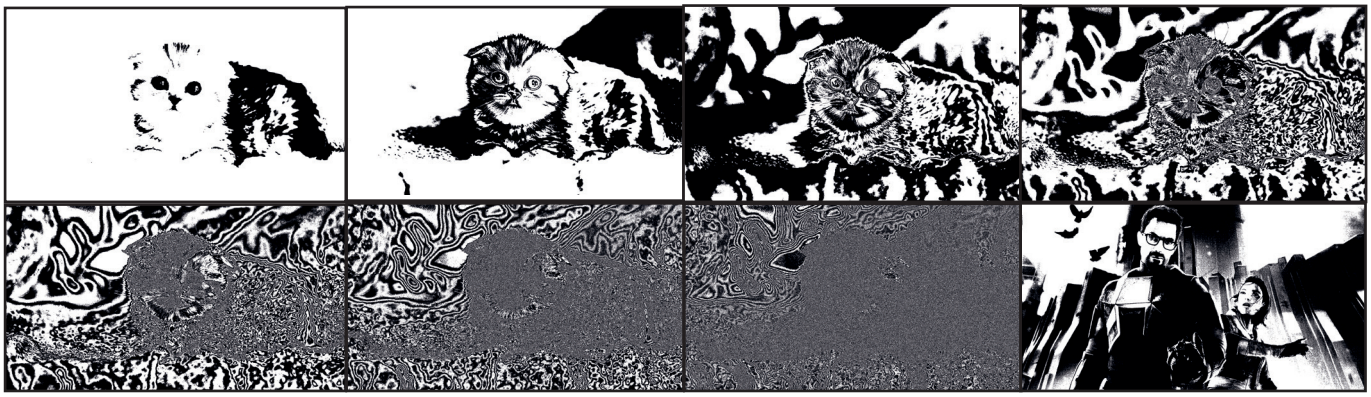

Figure 6 - Extraction des plans de bits pour la composante rouge.

Composante verte

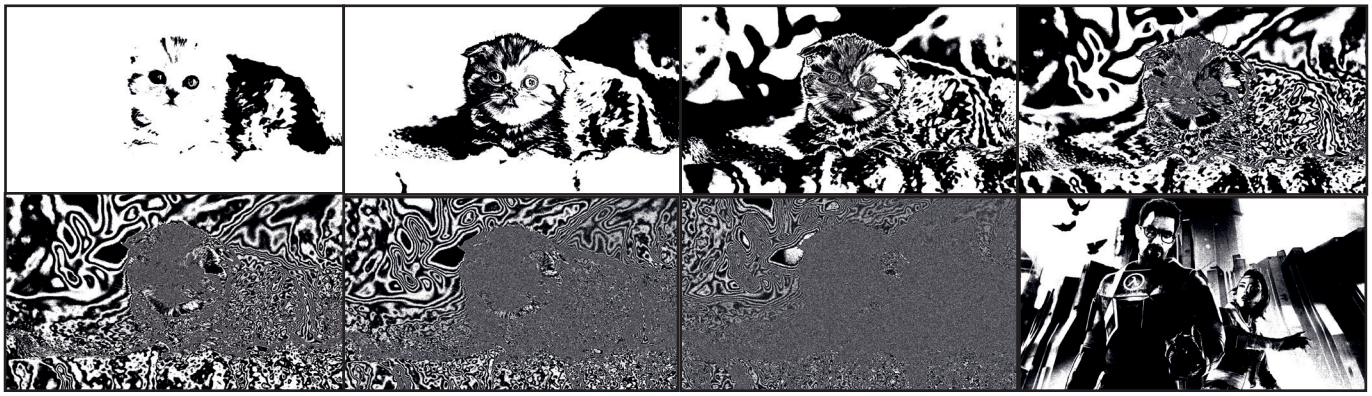

Figure 7 - Extraction des plans de bits pour la composante verte.

Composante bleue

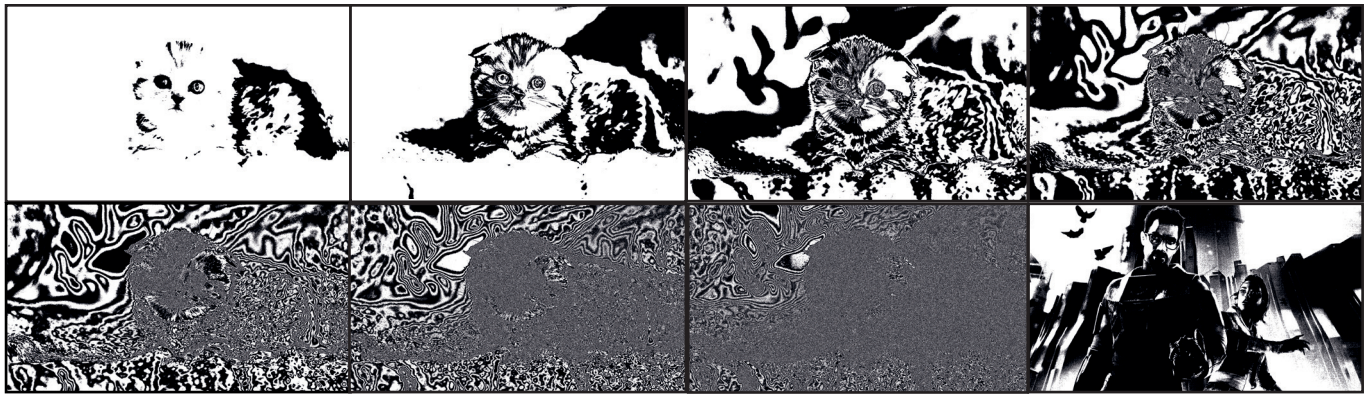

Figure 8 - Extraction des plans de bits pour la composante bleue.

Figure 9 - Image secrète dissimulée à l'intérieur de l'image du chaton.

bits (voir figures 6-8). L'image dissimulée apparaît en noir et blanc à la huitième position (plan de bits n°0).

CONCLUSION

La stéganographie est une pratique ancienne qui consiste à dissimuler des informations à l'intérieur d'un média comme un texte, une image, un fichier audio ou vidéo. Il existe plusieurs méthodes de stéganographie qui opèrent dans le domaine spatial (LSB Replacement, LSB Matching) et des variantes plus robustes dans le domaine fréquentiel (algorithmes Jsteg, F5, nsF5) [13-14].

Les modèles d'intelligence artificielle peuvent également être utilisés pour détecter les informations dissimulées dans un support. En utilisant des techniques d'apprentissage automatique, ces modèles peuvent être entraînés à reconnaître les anomalies ou les schémas spécifiques associés à la stéganographie [15]. Il existe une multitude de logiciels gratuits disponibles sur Internet, qui fonctionnent aussi bien sur les systèmes d'exploitation Linux que Windows. Ces logiciels sont conçus spécifiquement pour dissimuler et révéler des données cachées dans divers types de fichiers. Parmi les logiciels populaires, on peut citer Audiostego, LSB-Steganography, Steghide et Stegsolve. Que ce soit pour dissimuler des informations confidentielles dans des images, des fichiers audio ou d'autres supports, ces logiciels offrent la possibilité d'explorer les aspects fascinants de la stéganographie.

NOTES

1. Matlab (Matrix Laboratory) est une plateforme de calcul numérique et matriciel, et un environnement de développement, disposant de son propre langage de programmation.

BIBLIOGRAPHIE

- 1. GALAND, Fabien, *Introduction à la stéganographie*, Techniques de l'ingénieur, 2015.
- 2. YAHYA, Abid, *Steganography Techniques for Digital Images*, Springer, 2018.
- 3. SUBRAMANIAN Nandhini, ELHARROUSS Omar, S. AL-MAADEED Somaya et A. BOURIDANE Ahmed, *Image Steganography : A Review of the Recent Advances, IEEE Access*, vol. 9, 2021.
- 4. *Steganography*, <https://en.wikipedia.org wiki/Steganography>, consulté le 09 juin 2023.
- 5. JOHNSON Neil et JAJODIA Sushil, *Exploring steganography : Seeing the unseen*, Computer, vol. 31, no. 2, 1998.
- 6. *Histoires d'Hérodote* (Neuvième édition), traduction nouvelle avec une introduction et des notes par Pierre Giguet, Bibliothèque nationale de France, 1913.
- 7. *Correspondance de George Sand et d'Alfred de Musset*, publiée intégralement et pour la première fois d'après les documents originaux par Félix Decori, Bibliothèque nationale de France, 1904.
- 8. *Electronic frontier foundation*, <https:// www.eff.org/>, consulté le 09 juin 2023).
- 9. VENETSANOPOULOS Anastasios N et P TANIOTIS Konstantinos N, C*olor Image Processing and Applications*, Springer, 2014.
- 10. GUPTA Shilpa, GUJRAL Geeta, AGGARWAL Neha, *Enhanced least significant bit algorithm for image steganography, Int. J. Comput. Eng. Manage.*, vol. 15, no. 4, 2012.
- 11. SINGH Amritpal et SINGH Harpal, *An improved LSB based image steganography technique for RGB images, Proc. IEEE Int. Conf. Electr. Comput. Commun. Technol.*, 2015.
- 12. TARUN Venkata Sai, RAO Venkateswara, MAHESH Naga, REDDY Srikanth, VENKATESH M, *Digital video steganography using LSB technique, Red*, vol. 100111, 2020.
- 13. QIAO Tong, ZITZMANN Cathel, RETRAINT Florent et COGRANNE Rémi, *Statistical detection of Jsteg steganography using hypothesis testing theory*, IEEE International Conference on Image Processing, 2014.
- 14. LIU Jiufen, YANG Chunfang, WANG Junchao et SHI Yanan, *Stego key recovery method for F5 steganography with matrix encoding*, J Image Video Proc., 40, 2020.
- 15. PLACHTA Mikolaj, KRZEMIEN Marek, SZCZYPIORSKI Krzysztof et JANICKI Artur, *Detection of Image Steganography Using Deep Learning and Ensemble Classifiers*, Electronics, 11, 1565, 2022.

Mots-clés : Guillaume Boudarham / Cybercriminalité / Expertise / Image / Informatique / Intelligence artificielle / Least Significant Bit / LSB-Replacement / Stéganalyse / Stéganographie / Yellow-dots - RÉF. : ST, G, 13 . www.revue-experts.com