État de l'art

STÉGANOGRAPHIE

Réalisé par BLQIS

Novembre 2023

Table des matières

Table des figures

Chapitre 1

Introduction

La stéganographie, ou l'art antique de la dissimulation, est une science mature qui a résisté à l'épreuve du temps. L'avènement de l'ère numérique a permis d'offrir au domaine de nouveaux vecteurs et supports pour véhiculer des données dématérialisées, et ainsi renouveler la pertinence de la pratique. Au-delà des débats éthiques qui l'entourent, la stéganographie représente un enjeu non négligeable dans la protection des données mais aussi leur compromission.

Une méthode millénaire

Le tout premier exemple recensé de stéganographie remonte au 5ème siècle avant Jésus-Christ, lorsque Aristagoras, gouverneur de la ville de Milet, avait l'intention d'envoyer un message confidentiel. Pour ce faire, il fit tatouer le message sur le crâne rasé d'un esclave, attendant que les cheveux repoussent avant d'expédier la missive.

La Seconde Guerre Mondiale a également vu l'utilisation de techniques de communication secrètes, comme le micro-point, où une photographie de la taille d'un simple signe de ponctuation était astucieusement dissimulée au sein d'un texte légitime.

Nous pouvons également étudier cette missive envoyée par un espion allemand :

Apparently neutral's protest is thoroughly discounted and ignored. Ismam hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils.	Apparemment la protestation des pays neutres est totalement ignorée. Isman frappe fort. L'issue du blocus donne des prétextes pour un embargo sur certains produits, mis à part graisses animales et huiles végétales.

FIGURE 1.1 – Missive stéganographiée envoyée par un espion allemand

À première vue, ce message semble totalement anodin. Cependant, en prenant la deuxième lettre de chaque mot, il devient possible de reconstituer le message suivant : "Pershing sails from NY June 1" (Le Pershing part de New-York le 1er juin).

Un exemple plus contemporain de stéganographie est illustré dans une série de fiction populaire, "Prison Break", où le tatouage du personnage principal, Michael Scofield, semble représenter une bataille entre un ange et un démon, mais dissimule en réalité le plan de la prison qui lui permettra de s'échapper.

La popularité croissante d'Internet a conduit à une prolifération du partage d'images, avec des réseaux sociaux et des forums de discussion rassemblant des millions d'utilisateurs qui utilisent abondamment des images pour illustrer leurs conversations. En conséquence, les fichiers image sont devenus les supports privilégiés pour la stéganographie.

Applications Informatiques

À l'ère d'Internet, la stéganographie est l'apanage des personnes malintentionnées. Une application notoire de cette technique réside dans l'intégration d'*exploits* (un élément de programme qui prend l'avantage d'une vulnérabilité d'un logiciel) dans des images, souvent en conjonction avec des régies publicitaires. Ces régies inséraient des images infectées dans les publicités en ligne, qui en réalité servaient de support à la stéganographie. Lorsque les utilisateurs cliquaient sur ces bannières publicitaires, ils étaient redirigés vers des sites web contenant des failles de sécurité dans Adobe Flash Player. Si l'ordinateur de la victime utilisait une version non mise à jour de Flash, elle pouvait être infectée par un logiciel malveillant.

Cependant, il est important de noter que la stéganographie est loin d'être aussi répandue que la cryptographie à l'heure actuelle. Le *MITRE ATT&CK Framework* décrit la stéganographie comme une technique d'obscurcissement généralement utilisée pour dissimuler des activités malveillantes. Sa dernière utilisation significative remonte à décembre 2020 lors de l'attaque *SUNBURST* contre le logiciel *Orion* de l'éditeur de solutions de sécurité *SolarWinds*, considérée comme l'une des attaques les plus sophistiquées de l'histoire.

De nos jours, la stéganographie est couramment employée dans le cadre du tatouage numérique (*watermarking*), où un copyright est intégré discrètement au sein d'une œuvre protégée. En cas de diffusion non autorisée de cette œuvre sur Internet, la *watermark* permet d'identifier l'origine de la fuite et de prouver l'authenticité de l'œuvre.

Bien que la stéganographie soit une technique ancienne, elle reste pertinente pour des besoins contemporains. Il convient donc d'étudier son fonctionnement réel à travers les différents types de fichiers (image) qui lui servent de support, ainsi que les vulnérabilités et attaques connues à ce jour.

Chapitre 2

La stéganographie numérique

Chaque jour, de nouvelles méthodes de dissimulation de données émergent, toutes visant à modifier des bits de poids faible de manière imperceptible à l'œil humain.

L'utilisation de formats de fichiers riches en bits de poids faible est donc privilégiée, notamment dans le contexte de la stéganographie. Les images numériques sont particulièrement populaires à cet égard, car elles permettent de dissimuler des objets au sein de structures beaucoup plus vastes, profitant de la multitude de pixels présents.

LSB *Least Significant Bit* (LSB) consiste en l'altération du bit du poids faible d'un octet de données en fonction des bits du secret. Dans un fichier image, les bits concernés par les algorithmes de stéganographie se trouvent généralement dans ceux décrivant les pixels de l'image.

Compression Internet pilule d'images compressées : des techniques de stéganographie sont parfois utiliser pour y immiscer un secret durant la compression. Ces algorithmes sont appelés *JSteg* et *Discrete Wavelet Transform* (DWT).

Mode palette Une altération s'opère également lorsqu'une image change de mode de couleur (*RGB* à *Niveaux de Gris* par exemple), il est alors possible de dissimuler un secret durant cette étape. Un mode de couleur en particulier est concerné par cela : le mode *palette*.

Dans le système *Steganographic Nature Of Whitespace* (SNOW), les espaces vides à la fin des lignes de texte sont exploités pour encoder des messages dans des fichiers textes en code ASCII. Ces espaces et retours à la ligne restent invisibles dans les éditeurs de fichiers texte, assurant ainsi la discrétion totale de l'information dissimulée.

La méthode du Least Significant Bit (LSB), ou en français le bit de poids faible, est essentielle pour appréhender le fonctionnement de l'ensemble des techniques actuellement disponibles. Malgré le très large éventail des solutions proposées de nos jours, chaque approche repose finalement sur l'utilisation du LSB pour dissimuler des informations sensibles.

2.1 Formats de fichiers image

Pour mieux comprendre la pratique du LSB, il nous faut dans un premier temps comprendre comment est composé un fichier image classique. Les formats les plus utilisés sur internet sont le Windows BitMap (.BMP), le Joint Photography Expert Group (.JPEG) et le Portable Network Graphic (.PNG). Il est également possible d'enregistrer des images en Graphics Interchange Format (.GIF), qu'elles soient animées ou statiques.

BMP Chaque pixel est associé à une couleur spécifique (noir, blanc, en niveaux de gris ou en couleurs) sur une matrice de pixels. Cependant, les images BMP ne peuvent être compressées et occupent ainsi beaucoup d'espace de disque.

JPEG Couramment utilisé sur le web et apprécié pour son taux de compression inégalé, le JPEG est le standard des formats d'image sur internet. La compression avec perte de données réduit la taille du fichier, mais affecte la qualité visuelle de l'image. Autre inconvénient : l'absence de canal alpha. En effet, le JPEG utilise le mode RGB (*Red, Green, Blue*) et ne gère pas la transparence, contrairement au PNG.

GIF Ce format utilise une palette de couleurs limitée (256 couleurs au maximum), ce qui le rend particulièrement adapté aux images simples. Les images GIF sont compressées sans perte, la qualité de l'image reste donc inchangée.

PNG Le PNG est prisé pour sa qualité d'image élevée. Contrairement au JPEG, il utilise une compression sans perte, préservant ainsi la qualité de l'image. Même si cela peut entraîner des fichiers plus volumineux, le PNG reste une excellente alternative.

2.1.1 Composition d'un fichier image

Un fichier image se compose généralement d'une entête et d'un bloc de données. L'entête contient des informations telles que la taille de l'image, les méta-données et l'interprétation du bloc de données.

Dans le cas des fichiers GIF, l'entête inclut une palette de 256 couleurs utilisées pour représenter les couleurs dans l'image, permettant une compression efficace pour les images à faible profondeur de couleur. Les images PNG, tout comme les GIF, peuvent également utiliser une palette de couleurs, mais le format PNG est plus flexible, prenant en charge des profondeurs de couleur variables, offrant ainsi une plus grande polyvalence pour la représentation des couleurs dans les images.

2.1.2 Corps des fichiers

Dans le mode standard de PNG comme avec JPEG et BMP, le bloc de données comporte la liste des pixels présents dans l'image.

Il existe plusieurs modes de couleurs pour les images PNG :

Nombre de bits	Nombre de canaux	Taille d'un canal en bits	Couleurs
1	1	1	2 : noir et blanc
2	1	2	4 : basiques
8	1	8	256 : niveaux de gris
8	1	8	256 : mode palette
24	3		16 777 216
32	4	8	4 294 967 296
64	4	16	281 474 976 710 656

FIGURE 2.1 – Propriétés des modes de couleurs de PNG

Les modes les plus couramment utilisés sont les 24 et 32 bits. Dans le mode 24 bits, chaque pixel d'une image est composé de trois canaux de 8 bits (1 octet) chacun, correspondant respectivement à la quantité de Rouge, de Vert et de Bleu (dénommés RVB en français et RGB en anglais) présente dans

ce pixel. La mesure de la présence de ces couleurs varie de 0 à 255.

Par exemple, pour représenter la couleur blanche, on utilise le triplet (255, 255, 255), qui peut également être noté (11111111, 11111111, 11111111). En revanche, la couleur noire est représentée par le triplet (0, 0, 0), équivalent à (00000000, 00000000, 00000000). Cette configuration signifie que pour modifier la quantité de rouge dans un pixel, on ajuste le premier canal de 0 à 255. De même, le deuxième canal contrôle la variation de la présence de vert, et le troisième canal permet d'ajuster la quantité de bleu. Dans le mode 32 bits, un quatrième canal, appelé "canal alpha", est ajouté avec 8 bits supplémentaires pour représenter la transparence du pixel.

Constituer une image PNG 24 bit nécessite ainsi une très grande quantité de pixels. Pour représenter une image de 4 pixels dans ce format là par exemple, il nous faudra alors 4 x 3, soit 12 octets.

2.2 LSB : Least Significant Bit

2.2.1 Changement du bit le moins significatif

Avec une palette de plus de 16 millions de couleurs possibles, il devient très ardu de discerner deux teintes extrêmement proches à l'œil humain. Par conséquent, en altérant le bit le moins significatif d'un octet décrivant un composant de couleur d'un pixel, la valeur de ce composant s'accroîtra ou diminuera de ± 1.

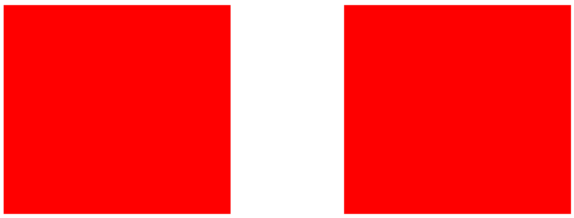

FIGURE 2.2 – Deux nuances de rouge différentes

Il est relativement difficile de pouvoir distinguer ces deux teintes de rouge à l'œil nu, pourtant, elles sont belles et bien différentes. Dans ce cas précis, le bit de poids faible du canal rouge de la couleur dans le carré à gauche a été altéré. En binaire, la couleur à gauche est représentée par (11111111, 0, 0), soit (255, 0, 0), la couleur à droite est représentée par (11111110, 0, 0), soit (254, 0, 0).

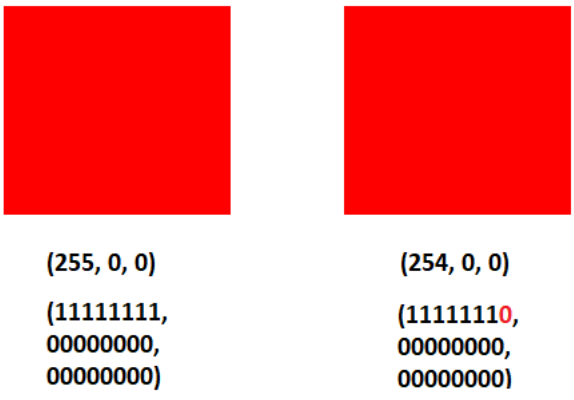

FIGURE 2.3 – Valeur réelle de ces deux nuances de rouge

2.2.2 Implémentation

Dissimuler une image dans une image

Afin de cacher une image dans une autre, il est possible de "combiner" chaque octet des deux fichiers. En effet, un octet est composé de 8 bits dont 4 bits de poids fort et 4 bits de poids faible. Il est possible d'utiliser tous les bits d'un octet correspondant à un canal d'un pixel, mais cela reste très bruyant : l'altération de la couleur sera évidente. Utiliser le LSB avec seulement le bit de poids le plus faible comme nous l'avons vu précédemment posera également des problèmes de volume pour dissimuler de plus gros fichiers ou même équivalents à celui d'origine.

Par conséquent, il est possible d'utiliser les 2 bits de poids le plus faible (2-LSB), ou encore les 3 (3-LSB), les 4 (4-LSB), etc... du moment que le bruit est maîtrisé. Cependant, il reste relativement rare d'aller au-delà du 2-LSB.

Pour réduire le bruit, dans le cas où l'objet à dissimuler est relativement peu volumineux, il est possible de ne pas utiliser tous les canaux d'un pixel : seulement 1 ou 2 peuvent suffire. Cependant, la qualité de l'image à l'extraction dépendra du nombres de bits et des canaux utilisés.

L'implémentation de ce procédé en Python sert de *Proof Of Concept* (POC). L'interface graphique a été conçue grâce au module Tkinter, et le traitement d'image à l'aide de la bibliothèque PIL. Les formats PNG, BMP, et JPG sont pris en charge. Il est possible de choisir le nombre de canaux, le type de LSB (2, 3, 4-LSB...) ainsi que le canal précis à utiliser. La dissimulation et l'extraction sont exécutés conjointement.

Complexité

En prenant deux images à taille identique, on se retrouve à faire des opérations sur N (largeur) x M (hauteur) pixels. Les opérations en question (récupération des couleurs de pixels, manipulation des canaux, conversion en binaire, concaténation des bits de poids fort et faible...) dépendent des boucles imbriquées qui permettent de parcourir les deux images. Elle sont effectuées en temps constant et ne contribuent donc pas à la complexité globale.

Les opérations étant proportionnelles au nombre de pixels, nous pouvons alors compter un round comme ayant un coût de 1. Comme il y a N x M rounds pour deux images, nous avons alors une complexité linéaire, en O(NxM), liée aux dimensions de nos deux images.

2.3 Autres cas d'utilisation

2.3.1 Stéganographie lors de la compression d'images

Les méthodes de stéganographie basées sur les transformées sont souvent liées au format JPEG. Les internautes s'échangent constamment des images en .JPEG.

Ce format est intéressant car lorsqu'on enregistre en .JPEG une image en .JPEG, on effectue de nouveau une compression. Ce qu'il se passe en réalité, c'est que durant le processus de compression, l'image doit se débarrasser des données superflues (des bits en excès) qui pourraient l'empêcher de mener à terme cette tâche. Durant la compression, une image JPEG effectue une approximation d'elle-même pour devenir plus petite. Ce changement transforme l'espace des données de l'image, d'où le terme "Transform Domain Based Techniques". Il est alors question de tirer profit de cette transformation pour cacher des informations grâce à une décomposition en ondelettes discrètes [2].

La *Discrete Wavelet Transform* (DWT) permet de décomposer l'image en différentes fréquences, créant ainsi des sous-bandes. Pendant le processus de compression, des données secrètes sont incorporées dans les coefficients de haute fréquence de la DWT, où les altérations sont moins visibles. Une fois l'image

compressée, les données cachées peuvent être extraites ultérieurement. Cette approche offre une méthode discrète et robuste pour la dissimulation d'informations, tout en préservant l'apparence visuelle de l'image (bruit minimisé).

2.3.2 Stéganographie sur les palettes

Lorsqu'on enregistre une image ayant un mode de couleur plus performant que le mode palette avec un format de fichier à palette (comme GIF et PNG dans certains cas), les 256 couleurs les plus récurrentes dans l'image sont retenues et référencées dans l'entête du fichier image sous forme de tableau. Ce tableau est appelé *palette de couleur*.

En résultat, nous obtenons des images très légères, de qualité moindre et ayant subit beaucoup de modifications. Il nous est alors possible de faire passer le message secret (dissimulé en LSB dans les pixels de l'image) comme étant du bruit.

Cette méthode est relativement efficace avec des images en niveau de gris, puisque ces images comportent déjà 256 couleurs.

2.4 Contraintes et limitations

De nos jours, de nombreux outils existent pour faciliter l'extraction d'informations issus de ces algorithmes de stéganographies extrêmement connus. La méthode classique LSB a pour avantage de fonctionner sur tous les types de fichiers, ne pas se limiter à un cas de figure précis comme c'est le cas avec la compression, et, selon le nombre de canaux et bits choisis, minimiser le bruit sur une image stéganographiée. Les dissimulation lors de la compression ou avec le mode palette ont pour plus grand inconvénient d'être vulnérable aux attaques bruteforce : il suffit simplement de tester successivement tous les algorithmes connus (dont DWT) pour arriver à l'information sensible.

Chapitre 3

La stéganalyse

Pour considérer un algorithme de stéganographie comme sécurisé, il se doit de produire des supports où la dissimulation apparaît comme imperceptible non seulement à l'oeil nu mais aussi après analyse informatique.

Définition La stéganalyse est l'art et la science de détecter si un fichier numérique contient des informations cachées. Une stéganalyse, soit l'attaque des processus de stéganographie, est considérée comme réussie si l'information dissimulée est détectée et extraite.

La stéganalyse peut être divisée en deux grandes catégories : la méthode ciblée (dite "Targeted Analysis" et la méthode aveugle (dite "Blind Steganalysis"). Lors d'une analyse ciblée, l'attaquant se sert des objets de connaissance dont il dispose (généralement il peut générer lui-même des supports avec l'algorithme de stéganographie - sans savoir comment il est implémenté), tandis que lors d'une analyse aveugle, l'attaquant ne dispose d'aucune information.

3.1 Analyse ciblée

Il existe 6 types d'attaques dont les principes sont très similaires à celles qu'on retrouve en cryptographie.

Stego-Only Attack L'attaquant dispose uniquement du support stéganographié sans connaissance préalable du contenu dissimulé. L'objectif de cette attaque est de détecter des signes révélateurs de la présence de données dissimulées en se basant uniquement sur le support lui-même, sans avoir accès au message d'origine.

Known-carrier attack L'attaquant a une connaissance préalable du support stéganographié, c'est-à-dire qu'il connaît le fichier original sans données dissimulées. L'attaquant compare le support d'origine avec le support suspecté de contenir des données dissimulées pour détecter des différences.

Known-message attack L'attaquant dispose du message clair qui a été dissimulé dans le support. L'attaque se concentre sur la recherche de signes révélateurs de la présence de ce message spécifique.

Chosen-stego attack L'attaquant dispose de la capacité de choisir lui-même les données à dissimuler dans un support stéganographié. Cette méthode lui permet de tester la résistance de la stéganographie en utilisant différents types de données.

Chosen-message attack Similaire à la méthode précédente, l'attaquant peut choisir le contenu du message à dissimuler plutôt que les données. Cela teste la capacité de la stéganographie à dissimuler différents types de messages.

Known-steganography attack L'attaquant connaît le message secret, l'image d'origine, l'image support et il peut utiliser l'algorithme de stéganographie à sa guise.

3.2 Analyse aveugle

Digital Forensics L'investigation numérique est une branche des sciences forensiques englobant la récupération et l'étude des matériaux trouvés dans les dispositifs numériques, souvent en relation avec la criminalité informatique.

L'analyste chargé d'enquêter "*à l'aveugle*" passe en revue l'image en utilisant son sens de la vue pour repérer des distorsions, des zones inhabituelles, des motifs répétés ou des anomalies qui pourraient indiquer la présence de données dissimulées. Par exemple, des variations de couleur ou de texture dans certaines parties de l'image peuvent éveiller les soupçons.

Il existe des outils logiciels spécialement conçus pour détecter des anomalies visuelles dans les images, tels que le logiciel de manipulation d'images forensiques. Un outil très sollicité et à la disposition de tous est le site internet *fotoforensics.com*, qui permet d'évaluer les photos que les internautes soumettent grâce à l'algorithme nommé *Error Level Analysis* (ELA : l'analyse du niveau d'erreur).

Fotoforensics définit l'ELA comme suit :

> *"L'analyse du niveau d'erreur (ELA) est un algorithme qui évalue le potentiel du niveau d'erreur d'une image JPEG. JPEG est un format d'image avec perte : chaque sauvegarde dégrade l'image. La quantité de dégradation varie en fonction du nombre de sauvegardes. La première sauvegarde perd beaucoup, la deuxième un peu plus, et à la 20ème sauvegarde, la qualité est probablement la plus basse possible.*
>
> *Lorsqu'une image est modifiée, les parties modifiées ont un potentiel d'erreur plus élevé que le reste de l'image. L'ELA fonctionne en sauvegardant l'image à un niveau de qualité connu (comme un JPEG à 95 %), puis détermine ce qui a été modifié. Les modifications et les raccords apparaissent comme des régions plus modifiées."*

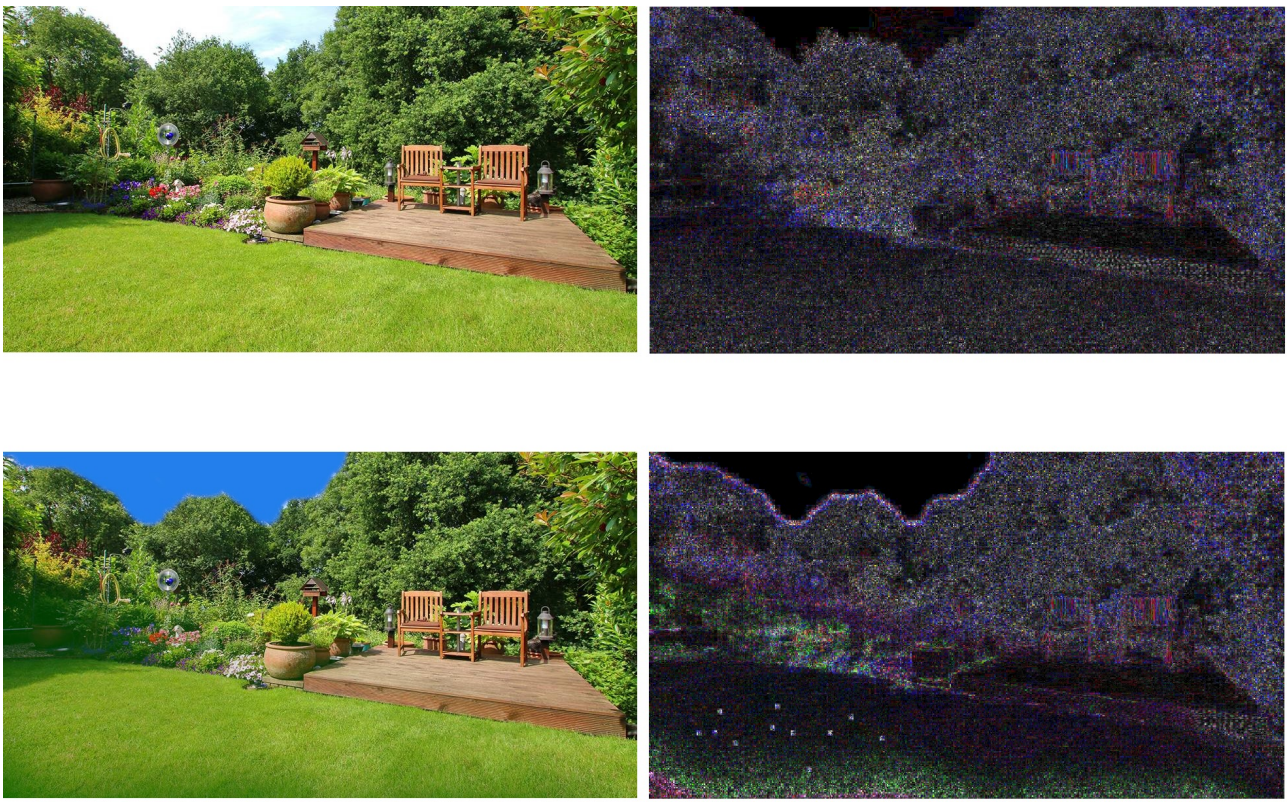

FIGURE 3.1 – ELA avec une image non modifiée et avec la même image modifiée

Dans l'image modifiée, nous avons rendu le ciel plus bleu à l'aide d'un détourage, rendu l'herbe plus verte et nous avons ajouté quelques points noirs sur l'herbe.
En soumettant l'image sur Fotoforensics, nous remarquons que l'image ELA met en surbrillance le détourage du ciel, la verdure de l'herbe et les points noirs. C'est avec la surbrillance que les parties modifiées d'une image de dessinent.

3.3 Attaques d'élimination

Il existe également d'autres types d'attaques, dont l'objectif n'est pas de récupérer l'information dissimulée mais plutôt de la rendre irrécupérable. Ces attaques sont intéressantes pour mesurer la robustesse de l'algorithme de stéganographie.

Destroy-Everything Attack Ce type d'attaque vise à détruire complètement le message et à la suite de ça l'attaquant n'essaiera peut-être même pas de récupérer le message.

Random-Tweaking attacks De légers changements dans les fichiers sont ajoutés pour que le message devienne illisible.

Add New Information Les attaquants utilisent la même technique de dissimulation de données pour intégrer un nouveau message dans le support de stéganographie. Le message original peut être écrasé.

Reformat Attack Un moyen courant pour les informations cachées dans un fichier est de changer le format du fichier. Par exemple, si notre image stéganographiée est en .JPEG et qu'elle est convertie en .PNG, la structure de données va énormément changer. Ce type d'attaque peut produire beaucoup de dommages au message caché.

Compression Attack L'attaquant peut compresser le fichier, ce qui pourrait entraîner la perte totale du secret incorporé dans le fichier, car les algorithmes de compression ont tendance à supprimer certaines informations pendant la compression.

Chapitre 4

Conclusion

La stéganographie, telle que présentée dans ce rapport, offre une méthode puissante pour la transmission discrète d'informations. Sa force réside dans deux concepts fondamentaux : la capacité de nos sens à ne pas détecter des altérations infimes au sein de médias stéganographiés, et le fait que, a priori, nous ignorons la présence d'informations cachées dans un fichier donné (même si des techniques d'analyse existent pour détecter de telles dissimulations).

Combinée à la cryptographie, elle peut présenter une réelle forme de danger, notamment dans le cadre d'ex-filtration illégale de données.

Toutefois, la stéganographie n'est pas limitée aux activités malveillantes. Au contraire, elle peut servir à les combattre : la traçabilité d'œuvres d'art numériques en est la preuve. De nos jours, de nombreux services de distribution de médias ont recours au tatouage numérique pour protéger le contenu qu'ils hébergent. La démocratisation de cette pratique pourrait constituer un réel tournant dans le combat contre la piraterie numérique.

Une autre utilisation possible pourrait être la stéganographie appliquée aux fichiers audio. Le chiffrement permettrait d'une part de rendre toute donnée sensible, comme des messages vocaux, intelligibles en cas de fuite, et la stéganographie de les rendre indétectables.

Il est donc essentiel de reconnaître que la stéganographie ne doit pas être condamnée d'emblée, mais plutôt étudiée et régulée pour garantir son utilisation éthique. Les futures recherches et développements dans le domaine sont prometteurs : nul doute qu'ils joueront un rôle clé dans la définition du paysage de la sécurité numérique et de la confidentialité dans les années à venir.

Bibliographie

[1] Rene Alt. "LA TRANSFORMATION EN ONDELETTES". In : (). URL : https://perso.telecom-paristech.fr/bloch/P6Image/ondelettestrsp.pdf.

[2] RANDRIANJOHANY Tolotra Andrianaina. *STEGANOGRAPHIE UTILSANT LA TRANSFORMEE EN ONDELETTE DISCRETE APPLIQUEE AUX IMAGES NUMERIQUES*. URL : http://biblio.univ-antananarivo.mg/pdfs/randrianjohanyTolotraA_ESPA_MASTPRO_16.pdf.

[3] Mitre | ATT&CK. "Obfuscated Files or Information : Steganography". In : (). URL : https://attack.mitre.org/techniques/T1027/003/.

[4] Frédéric Bayart. *Petite histoire de la stéganographie*. URL : http://mathweb.free.fr/crypto/stegano/histstegano.php3.

[5] Adrien C. "La stéganographie une technique souvent oublié". In : (). URL : https://theexpert.squad.fr/theexpert/digital/la-steganographie-une-technique-souvent-oubliee/.

[6] Jacques Cheminat. "Sécurité : Stegano se camoufle dans les pixels des bannières de pub". In : (). URL : https://www.silicon.fr/securite-stegano-se-camoufle-dans-pixels-bannieres-pub-164499.html#.

[7] ClashInfo.com. ".bmp, .tiff, .gif, . jpeg, .png, ... Tout sur les formats d'image". In : (). URL : http://www.clashinfo.com/aide-informatique/multimedia/art153-formats-image.html#:~:text=de%20plusieurs%20formats.-,Les%205%20formats%20que%20je%20vais%20d%C3%A9tailler%2C%20qui%20sont%20%C3%A9galement,JPEG%20%3A%20Joint%20Photographic%20Expert%20Group.

[8] Mr P. Prasanna Kumar Dr. KP. Ramesh Babu Mr. Taffera Mekonen. *Palette Embedded Images by Steganographic Technique*. URL : https://www.ijireeice.com/upload/2016/november-16/IJIREEICE%5C%203.pdf.

[9] Hacker Factor. *Fotoferensis*. URL : http://fotoforensics.com/faq.php?show=General&c=guidelines.

[10] FireEye. "Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims With SUNBURST Backdoor". In : (). URL : https://www.fireeye.com/blog/threat-research/2020/12/evasive-attacker-leverages-solarwinds-supply-chain-compromises-with-sunburst-backdoor.html?ref=hackernoon.com.

[11] O. Touraille & J. M. Mény N. Buyle-Bodin. *Module PIL - Stéganographie*. URL : http://math.univ-lyon1.fr/irem/Formation_ISN/formation_prog_images/module_PIL/stegano_PIL.html.

[12] Michael T. Raggo. *Steganography, Steganalysis, & Cryptanalysis*. URL : https://www.blackhat.com/presentations/bh-usa-04/bh-us-04-raggo/bh-us-04-raggo-up.pdf.

[13] ScienceDirect. "Transform Domain". In : (). URL : https://www.sciencedirect.com/topics/computer-science/transform-domain.

[14] *Steganalysis*. URL : http://io.acad.athabascau.ca/~grizzlie/Comp607/steganalysis.htm.

[15] Todd Veldhuizen. *Measures of image quality*. URL : https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/VELDHUIZEN/node18.html.

[16] WIKIPÉDIA. “Digital forensics”. In : (). URL : https://en.wikipedia.org/wiki/Digital_forensics.

[17] WIKIPÉDIA. “Ondelette”. In : (). URL : https://fr.wikipedia.org/wiki/Ondelette.

[18] WIKIPÉDIA. “Stéganographie”. In : (). URL : https://fr.wikipedia.org/wiki/St%C3%A9ganographie#:~:text=Pour%20prendre%20une%20m%C3%A9taphore%2C%20la,un%20coffre%20dans%20son%20jardin.