

# Stéganographie d'image

Pasquier Maxime  
Sciences Informatiques  
[maxime.pasquier@etu.unige.ch](mailto:maxime.pasquier@etu.unige.ch)

**Abstract**—La stéganographie d'image est une méthode visant à dissimuler des informations au sein d'images sans que cela soit détectable par l'œil humain. Ce travail explore les méthodes LSB (Least Significant Bit) et DCT (Transformée en Cosinus Discrète) utilisées pour réaliser la stéganographie d'images, mettant en lumière à la fois leurs avantages et leurs limites.

## I. INTRODUCTION

Hormis la cryptographie pour transmettre des messages, il existe des méthodes de dissimulation comme la stéganographie. Ces dernières ne transforment pas le message en un autre non lisible, mais se contentent de cacher le message dans un autre document. Le travail se concentre sur les méthodes de "data hiding" dans des images. En effet, une image est un très grand ensemble de données avec beaucoup de redondance. Par conséquent, il est possible d'y dissimuler des informations de manière secrète sans pour autant affecter la qualité de l'image.

La stéganographie doit être distinguée de la cryptographie. La table suivante présente une comparaison entre ces deux méthodes permettant de transmettre des messages entre deux parties tout en assurant divers aspects de la sécurité de l'information.

Cryptographie	Stéganographie
Connaissance qu'un message est transmis	Aucune connaissance qu'un message est transmis
L'encryption empêche des personnes non autorisées d'accéder au contenu du message transmis	La stéganographie empêche la découverte de l'existence d'un message transmis
Technologie commune	Technologie peu commune
La cryptographie altère la structure du message transmis	La stéganographie n'altère pas la structure du message transmis

Table I: Illustration des différences entre cryptographie et stéganographie (1).

Les techniques de compression sont largement utilisées en stéganographie d'image. Parmi ces méthodes, on distingue la compression avec pertes (lossy compression) de la compression sans pertes (lossless compression). Les compressions sans pertes sont beaucoup plus couramment utilisées en stéganographie, et parmi les formats couramment employés, on peut citer le GIF (Graphics Interchange Format) et le BMP (Bitmap) (2).

Les algorithmes du **Least Significant Bit (LSB)** et de la **Discrete Cosine Transform (DCT)** sont mis en œuvre et discutés dans ce document. Leur fonctionnement est exposé,

et une implémentation est proposée. Les vulnérabilités de ces méthodes sont également explorées, et des attaques y sont décrites.

## II. MÉTHODOLOGIE

Chaque algorithme développé suit une méthodologie similaire. Tout d'abord, nous expliquons le fonctionnement de la méthode. Ensuite, nous proposons une implémentation pour l'encodage et le décodage. Enfin, nous approfondissons les vulnérabilités et discutons des attaques potentielles.

## III. THÉORIE

### A. Least Significant Bit (LSB)

La méthode de stéganographie par Least Significant Bit exploite les bits les moins significatifs du codage des pixels. Généralement, lorsque l'on parle de méthodes LSB, on se réfère à la modification des derniers bits des séquences. En effet, chaque pixel est codé sur 3 canaux de couleurs, et chacune de ces valeurs peut dissimuler des bits d'information. Par exemple, dans la séquence 10001011, qui peut représenter l'intensité d'un canal pour un pixel, un ou plusieurs bits de faible poids peuvent être modifiés pour cacher de l'information.

La technique exploite le fait que les bits d'une séquence n'ont pas la même pondération. En effet, les bits de poids fort (MSB) ont une pondération élevée car ils représentent des valeurs de puissances de 2 plus grandes, tandis que les bits de poids faible (LSB) ont une pondération plus faible dans la séquence binaire. Par exemple, sur une séquence de 8 bits, le premier bit représente à lui seul une contribution de 128, tandis que le dernier bit ne représente qu'une contribution de 1. Par conséquent, si l'on utilise ce dernier bit pour dissimuler des informations, cela a peu d'incidence sur la valeur globale de la séquence.

Il existe plusieurs techniques pour le codage des LSB. Une méthode consiste à modifier les bits de poids faible de chaque composante de chaque pixel d'une image, puis à les extraire sous forme d'une chaîne de bits pour interprétation ultérieure. Une autre approche pourrait consister à ajuster les LSB de manière à ce que la somme des bits à 1 dans la séquence binaire soit rendue paire ou impaire. Lors du décodage, une somme paire pourrait être interprétée comme un 1, tandis qu'une somme impaire pourrait être interprétée comme un 0, par exemple. Toutes ces variantes de LSB reposent sur les mêmes principes, à savoir la modification des bits de poids

faible pour y dissimuler les informations souhaitées.

Les informations dissimulées peuvent revêtir diverses formes. Il est envisageable de cacher du texte aussi bien que des images par le biais de la stéganographie. Grâce à la méthode proposée, il est possible d'incorporer une image dans une autre sans altérer de manière significative la qualité visuelle des deux images.

La méthode la plus couramment employée de nos jours est l'insertion de bits de poids faible (*LSB Insertion*), qui implique la modification du bit de poids le plus faible de l'image de couverture. Bien que cette technique soit relativement simple à mettre en œuvre, elle présente aussi une vulnérabilité accrue aux attaques. De plus, la moindre altération de la palette de couleurs ou de l'image peut entraîner la perte totale du message dissimulé. (1).

#### B. Discrete Cosine Transform (DCT)

La Transformée en Cosinus Discrète (DCT) est une transformation appliquée à une image qui convertit les données spatiales initiales en coefficients de fréquence. Elle effectue une transition du domaine spatial vers le domaine fréquentiel et trouve une utilité dans de nombreuses applications. Par exemple, la compression JPEG fait usage de la DCT.

Les coefficients de fréquence fournis par la DCT permettent de reconstruire l'image initiale avec une qualité de restitution élevée. En appliquant d'abord la DCT à une image, puis son inverse, il est possible de retrouver l'image initiale. En conséquence, la DCT est une transformation réversible.

De manière similaire à la méthode LSB, il est possible de modifier les bits de poids faible des coefficients par le biais de la DCT pour dissimuler de l'information sans compromettre la qualité de l'image. Pour extraire ensuite les informations dissimulées, il suffit d'appliquer la DCT à l'image stéganographique et de récupérer les bits de poids faible des coefficients ainsi obtenus.

### IV. IMPLÉMENTATIONS

L'algorithme pour la méthode Least Significant Bit permettant de cacher un message dans une image de nuances de gris est (1) :

- 1) Lire l'image de couverture ainsi que le texte que l'on souhaite cacher.
- 2) Convertir le texte en un message binaire.
- 3) Calculer les LSB de chaque pixels de l'image de couverture.
- 4) Remplacer les LSB de l'image de couverture par chaque bit du message secret.
- 5) Écrire l'image stéganographique.

Afin de récupérer le message depuis une image stéganographique nous faisons les étapes (1) :

- 1) Lire l'image stéganographique.
- 2) Calculer les LSB de chaque pixels de l'image stéganographique.
- 3) Récupérer les bits et les convertir en des caractère de 8 bits.

Le papier (3) propose une implémentation pour effectuer la stéganographie avec DCT. Les procédures pour l'encryption et la decryption sont les suivantes.

- 1) Lire l'image de couverture.
- 2) Lire le message secret et le convertir en une forme binaire.
- 3) L'image de couverture est divisée en blocs de  $8 \times 8$  pixels.
- 4) En partant de gauche à droite et de haut en bas, nous soustrayons 128 à chaque bloc de pixels.
- 5) La DCT est appliquée à chaque bloc.
- 6) Chaque table est compressée par une table de quantification.
- 7) Calculer les LSB de chaque coefficient DC et les remplacer avec chaque bit du message secret.
- 8) Ecrire l'image stéganographique.

Afin de récupérer le message secret d'une image stéganographique par DCT nous faisons (3) :

- 1) Lire l'image stéganographique.
- 2) L'image stéganographique est découpée en blocs de  $8 \times 8$  pixels.
- 3) En partant de gauche à droite et de haut en bas, nous soustrayons 128 à chaque bloc de pixels.
- 4) La DCT est appliquée sur chacun des blocs.
- 5) Chaque table est compressée par une table de quantification.
- 6) Calculer les LSB de chaque coefficient DC.
- 7) Reconstruire le message secret et le convertir en caractère sur 8 bits.

Ce travail propose des implémentations visant à dissimuler du texte dans des images, ainsi que des images dans d'autres images. Dans un premier temps, pour valider la méthode, nous avons réalisé des implémentations LSB et DCT pour cacher du texte dans des images. La récupération du texte dissimulé dans l'image stéganographique atteste de la validité des méthodes employées.

Dans un second temps, nous avons adapté les scripts afin de pouvoir dissimuler des images au sein d'autres images. Nous avons également réalisé une série d'analyses quantitatives sur les résultats obtenus, lesquelles sont présentées dans notre étude.

### V. ATTAQUES & VULNÉRABILITÉS

Contrairement à la cryptographie, la stéganographie n'altère pas le message pour le transformer en une forme illisible.

Par conséquent, la sécurité de la stéganographie repose sur le fait qu'elle ne doit pas être détectée. Tout système stéganographique est considéré comme compromis dès lors qu'un attaquant en a connaissance.

Par conséquent, une méthode de stéganographie efficace doit être difficile à détecter. En stéganalyse, il existe de nombreuses méthodes pour détecter la présence ou l'absence de stéganographie. Ce document se concentre exclusivement sur LSB et DCT et ne développe donc pas les méthodes qui ne sont pas pertinentes pour les implémentations réalisées.

Comme mentionné précédemment, si la présence de la stéganographie est détectée, la sécurité est compromise. Pour éviter de révéler la présence d'un message caché, il est essentiel de ne pas altérer la qualité visuelle de l'image et de ne pas introduire d'anomalies dans ses données.

En ce qui concerne les méthodes LSB, l'extraction des informations dissimulées est relativement simple à réaliser. Un attaquant peut facilement extraire ces informations à partir des pixels de l'image. Afin de rester discret, il est crucial de ne pas modifier un nombre excessif de LSB, ce qui pourrait détériorer considérablement la qualité de l'image.

Un algorithme de Steganalysis sur une image stéganographique encodée par une technique de Moderate Significant Bit replacement (MSB) avec 4 bits est proposé par l'étude (2). L'image stéganographique contient une image secrète codée sur les 4 bits de poids les plus faibles :

- 1) L'image stéganographique est shiftée de 4 bits vers la gauche.
- 2) L'image est ensuite ANDED avec 255, c'est-à-dire 11111111 afin de récupérer l'image secrète initiale.
- 3) L'image est ensuite convertie en le type Uint8 afin que ses valeurs de pixels soient comprises dans l'intervalle [0, 255].

La méthode DCT présente des vulnérabilités similaires à celles des méthodes LSB. En altérant les coefficients de fréquence, nous impactons la qualité de l'image. Plus nous cachons d'informations, plus nous accroissons le risque d'être détectés. L'un des avantages majeurs de la DCT par rapport à la LSB est que nous pouvons dissimuler davantage d'informations sans être facilement repérés. Pour y parvenir, nous pouvons sélectionner des fréquences spécifiques et modifier leurs bits pour cacher notre message. En effet, les coefficients de différentes fréquences ont des effets différents sur la qualité visuelle de l'image résultante. Par conséquent, il est essentiel de choisir les coefficients qui minimisent l'impact sur l'apparence de l'image stéganographique.

## VI. RÉSULTATS

Les résultats obtenus sont structurés en deux parties : tout d'abord, les analyses d'images pour LSB sont présentées, puis

les analyses pour DCT sont exposées.

Dans le cadre de ce travail, seules deux images sont traitées : une image de couverture et une image secrète. L'image de couverture agit comme le contenant de l'image secrète. L'objectif est de dissimuler l'image secrète dans l'image de couverture sans altérer la qualité visuelle des images. Pour chaque scénario, une représentation visuelle ainsi que des mesures quantitatives sont fournies.



((a)) Image de couverture

((b)) Image secrète

Figure 1: Les résultats se concentrent sur ces deux images en RGB. L'image (a) est l'image de couverture et l'image (b) est l'image secrète.

### A. Least significant bit (LSB)

La première opération réalisée avec LSB consiste à dissimuler du texte dans une image. Dans les images suivantes, nous avons caché une phrase parmi les pixels de l'image de couverture. Le code mis en place par cette étude permet de récupérer correctement le message dissimulé dans l'image stéganographique. Étant donné la courte longueur du message caché, aucune autre analyse n'est présentée pour cette situation.



((a)) Image de couverture

((b)) Image stéganographique dissimulant un message texte secret.

Figure 2: Le message secret : "Ceci est le message secret caché dans l'image de couverture" est dissimulé dans l'image stéganographique (b).

Ensuite, dans une seconde phase, nous avons dissimulé l'image secrète au sein de l'image de couverture. Étant donné que l'image secrète a la même taille que l'image de couverture, il est nécessaire de déterminer quelle quantité d'information nous souhaitons dissimuler. En effet, chaque pixel de l'image stéganographique doit contenir les informations pour les deux images. Le travail propose trois configurations différentes pour cela.

- 1) Modification de 2 LSB.

- 2) Modification de 4 LSB.
- 3) Modification de 6 LSB.



Figure 3: Image stéganographique contenant l'image secrète sur 2 LSB.



Figure 4: Reconstruction de l'image secrète à partir de l'image stéganographique sur 2 bits.



Figure 5: Image stéganographique contenant l'image secrète sur 4 LSB.



Figure 6: Reconstruction de l'image secrète à partir de l'image stéganographique sur 4 bits.



Figure 7: Image stéganographique contenant l'image secrète sur 6 LSB.



Figure 8: Reconstruction de l'image secrète à partir de l'image stéganographique sur 6 bits.

Les tableaux ci-dessous affichent les mesures du Peak Signal to Noise Ratio (PSNR) ainsi que de l'erreur quadratique moyenne (MSE) pour les images stéganographiques et les images reconstruites à partir des images stéganographiques. Ces images sont comparées respectivement avec l'image de couverture et l'image secrète initiale.

<b>2 LSB</b>	Image stéganographique	Image secrète reconstruite
MSE	2.38	106.46
PSNR	44.39	27.89

Table II: Mesures quantitatives pour 2 LSB.

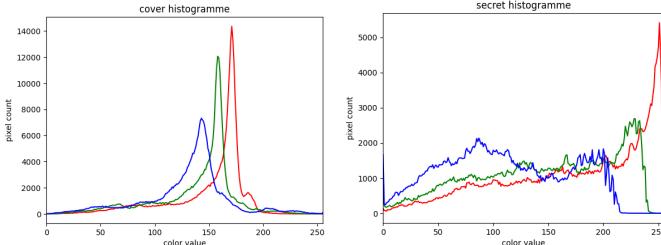
<b>4 LSB</b>	Image stéganographique	Image secrète reconstruite
MSE	39.27	77.57
PSNR	32.22	29.27

Table III: Mesures quantitatives pour 4 LSB.

<b>6 LSB</b>	Image stéganographique	Image secrète reconstruite
MSE	93.14	3.47
PSNR	28.47	42.76

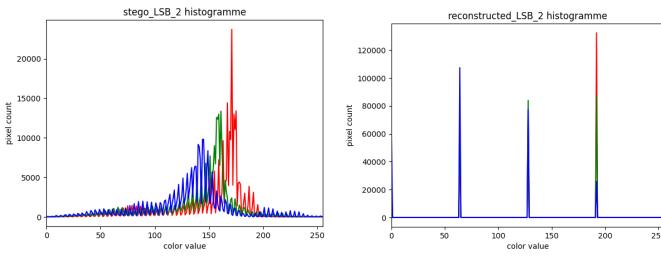
Table IV: Mesures quantitatives pour 6 LSB.

Chaque image est composée de 3 canaux de couleur en format **RGB**. La répartition des couleurs peut être visualisée à l'aide d'un histogramme. En stéganalyse, un histogramme peut révéler des anomalies statistiques, ce qui peut potentiellement trahir la présence d'un message secret. Ci-dessous, vous trouverez les histogrammes des images stéganographiques, ainsi que ceux des images reconstruites à partir des images stéganographiques.



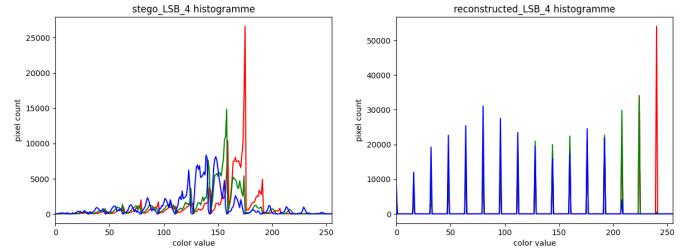
((a)) Histogramme de l'image de couverture ((b)) Histogramme de l'image secrète

Figure 9: Ces figures donnent la répartition des pixels de couleur pour les 3 canaux RGB sur l'image de couverture ainsi que l'image secrète. Aucun traitement n'a été effectué sur ces deux images.



((a)) Image stéganographique ((b)) Secret décodé

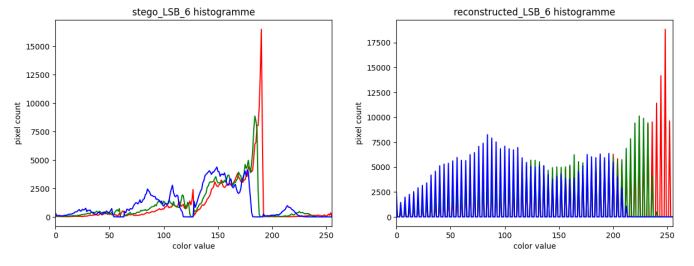
Figure 10: Histogrammes pour un encodage et décodage sur 2 LSB.



((a)) Image stéganographique

((b)) Secret décodé

Figure 11: Histogrammes pour un encodage et décodage sur 4 LSB.



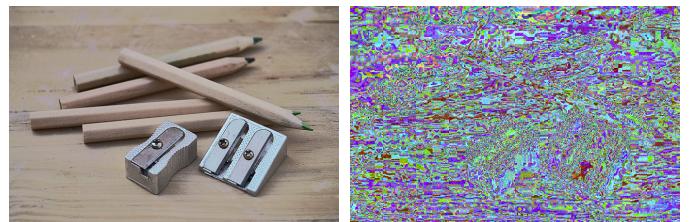
((a)) Image stéganographique

((b)) Secret décodé

Figure 12: Histogrammes pour un encodage et décodage sur 6 LSB.

### B. Discrete Cosine Transfert (DCT)

En raison de difficultés rencontrées lors de la mise en œuvre de la stéganographie avec DCT, nous avons subdivisé le processus en deux parties distinctes, chacune traitée individuellement. Tout d'abord, nous présentons le cas où l'image stéganographique est cohérente, mais où il est impossible de récupérer l'image secrète. Ensuite, nous examinons le cas où l'image stéganographique n'est pas cohérente, mais où il est possible de récupérer l'image secrète.



((a)) Image stéganographique

((b)) Secret décodé

Figure 13: Illustration du premier cas où l'image stéganographique est similaire à l'image de couverture mais dont il est impossible de récupérer l'image secrète dissimulée. La figure (b) est le résultat du décodage de l'image stéganographique.



((a)) Image stéganographique ((b)) Secret décodé

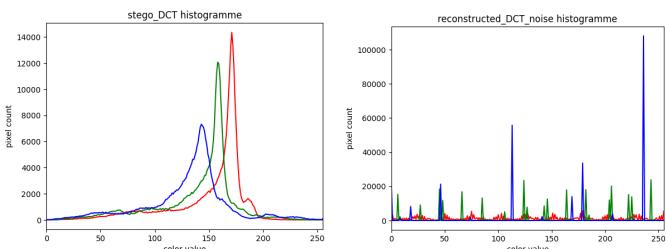
Figure 14: Illustration du second cas où l'image stéganographique est très différente de l'image de couverture mais dont nous pouvons récupérer l'image secrète dissimulée. La figure (b) est le résultat du décodage de l'image stéganographique.

DCT 1	Image stéganographique	Image secrète reconstruite
MSE	0	106.45
PSNR	$\infty$	27.89

Table V: Mesures quantitatives pour le premier cas de DCT.

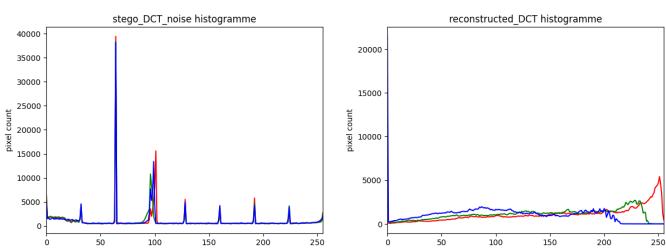
DCT 2	Image stéganographique	Image secrète reconstruite
MSE	106.44	2.74
PSNR	27.89	43.78

Table VI: Mesures quantitatives pour le second cas de DCT.



((a)) Image stéganographique co- ((b)) Secret décodé non cohérent hérante

Figure 15: Histogrammes pour un encodage et décodage sur le premier cas.



((a)) Image stéganographique non cohérente ((b)) Secret décodé cohérent

Figure 16: Histogrammes pour un encodage et décodage sur le second cas.

## VII. DISCUSSION

Dans un premier temps, nous avons mis en œuvre la stéganographie en dissimulant du texte dans l'image de couverture. Cette étape visait à valider les méthodes utilisées, plutôt qu'à obtenir des résultats significatifs.

L'encodage ainsi que le décodage sont réalisés sans erreur en utilisant les LSB dans le domaine spatial. Cependant, dans le domaine fréquentiel avec la DCT, le même processus produit des erreurs. À titre d'exemple, si nous encodons la phrase : "Ceci est du texte caché dans les coefficients du cosinus.", nous obtenons lors du décodage : "Ce#i est du texte caché dans les Cmeffici\$Jd2ducosinus.¶¶[mµ]". Ces erreurs s'expliquent par la représentation en virgule flottante finie. En effet, les coefficients de la DCT sont des nombres réels, et ils ne peuvent pas être représentés avec une précision parfaite en utilisant des nombres flottants de 64 bits. Les scripts produisant ces résultats sont disponibles sous les noms de *LSB\_text.py* et *DCT\_text.py*.

Ce travail propose trois configurations pour la méthode LSB. Les illustrations montrent des altérations nettement visibles dans les images, perceptibles à l'œil nu. Nous constatons que plus nous dissimulons de bits, plus nous altérons la qualité de l'image stéganographique, mais en même temps, nous obtenons une meilleure qualité pour l'image secrète. À l'inverse, lorsque nous dissimulons peu de bits, le phénomène s'inverse, altérant moins l'image stéganographique mais détériorant la qualité de l'image secrète.

La visualisation des histogrammes des images se révèle être un moyen efficace en stéganalyse pour la détection de messages secrets. En effet, en modifiant les LSB de l'image de couverture, nous assignons les pixels à des valeurs plus spécifiques. Bien entendu, cet effet dépend de l'image secrète que nous dissimulons, mais les résultats montrent clairement des irrégularités dans la répartition des valeurs. Au lieu d'observer des transitions de couleurs progressives, nous observons des trous. Ce phénomène est particulièrement notable dans les images stéganographiques des figures 10 et 11. Sans surprise, les images décodées ne présentent que peu de couleurs différentes, car leur codage offre seulement  $2^{LSB}$  couleurs possibles.

Le premier cas de DCT échoue à restituer l'image secrète en raison de la conversion des coefficients en pixels codés sur 8 bits. Lorsque nous appliquons la DCT à une image contenant des valeurs comprises entre 0 et 255 sur 8 bits, nous obtenons une matrice de même taille composée de coefficients en virgule flottante codés sur 64 bits. Par la suite, nous modifions les LSB de ces coefficients avant de les reconvertis dans le domaine spatial. La structure de données résultante de la transformation inverse est constituée de nombres à virgule flottante sur 64 bits. Pour créer une image à partir de ces données, nous arrondissons les valeurs au nombre

entier le plus proche et les encodons sur un seul bit (Uint8). Cependant, cet arrondi détruit l'image secrète, aboutissant ainsi à une image stéganographique identique à l'image de couverture. Par conséquent, le secret décodé ne contient aucune information relative à l'image secrète. Les méthodes *dct* et *idct* du package *scipy.fft* sont utilisées pour ces transformations.

Dans le deuxième cas de DCT, nous n'effectuons pas d'arrondi ni de mappage vers Uint8. Nous conservons les structures de données produites par les méthodes de *scipy.fft*. Par conséquent, l'image stéganographique est composée de nombres à virgule flottante codés sur 64 bits. La bibliothèque *matplotlib.pyplot* utilise des méthodes de conversion pour interpréter ces valeurs flottantes. Cependant, le secret n'est pas détruit et peut être récupéré presque entièrement. Le niveau de reconstruction est très bon, à l'exception de quelques anomalies. Les valeurs de MSE ainsi que de PSNR sont meilleures que celles obtenues avec la méthode à 6 LSB modifiés. Le problème réside dans le fait que les coefficients de DCT sont des nombres réels, et que les ordinateurs ont du mal à les représenter avec des puissances de 2. Par conséquent, des erreurs d'arrondi génèrent des anomalies lors de la reconstruction de l'image secrète. Dans notre exemple, ce sont des pixels rouges qui sont visibles à l'œil nu.

### VIII. CONCLUSION

L'objectif de ce travail est d'explorer les méthodes classiques de LSB (Least Significant Bit) et de DCT (Discrete Cosine Transform) pour la stéganographie. Nous proposons une implémentation des deux méthodes et décrivons leurs vulnérabilités aux attaques. La méthode LSB dans le domaine spatial est toutefois limitée par le nombre de bits que nous pouvons dissimuler sans éveiller les soupçons quant à la présence d'un message caché. De plus, les histogrammes des images stéganographiques peuvent révéler la présence du message, ce qui rend ces méthodes classiques de LSB peu discrètes.

La méthode DCT a montré des améliorations par rapport à la méthode LSB dans le domaine spatial. Cependant, nous n'avons pas pu mettre en œuvre correctement la stéganographie avec DCT, mais nous avons néanmoins obtenu des résultats intéressants. La DCT permet une reconstitution de meilleure qualité du point de vue quantitatif, bien que visuellement, cela puisse être moins satisfaisant. Elle offre la possibilité de dissimuler et de restaurer une grande quantité d'informations.

### REFERENCES

- [1] K. Thangadurai and G. Sudha Devi, "An analysis of lsb based image steganography techniques," in *2014 International Conference on Computer Communication and Informatics*, 2014, pp. 1–4.
- [2] D. Neeta, K. Snehal, and D. Jacobs, "Implementation of lsb steganography and its evaluation for various bits," in *2006 1st International Conference on Digital Information Management*, 2007, pp. 173–178.
- [3] E. Walia, P. Jain, and Navdeep, "An analysis of lsb & dct based steganography," *Global Journal of Computer Science and Technology*, vol. 10, 01 2010.