Attaque d'une méthode d'obscuration d'images par bit-flipping

N. Hutte[1] W. Puech[1]

[1] LIRMM, Univ. Montpellier, CNRS, Montpellier, France

Résumé

De nos jours, il est de plus en plus nécessaire de sécuriser les données multimédia. Le chiffrement intégral est efficace et sûr, mais il ne garantit pas un bon compromis entre la sécurité et d'autres exigences. L'obscuration d'images est une solution possible, préservant l'intégrité des données multimédia tout en obscurcissant leur contenu. En 2021, Aprilpyone et Kiya [1] ont proposé une méthode d'obscuration d'image par bit-flipping, utilisant une clé secrète permettant d'obscurcir et de garder secrète l'image originale, tout en étant réversible. Toutefois, cette méthode présente une faille exploitable sujette à une attaque. Dans cet article, nous présentons une méthode pour attaquer les images obscurcies par bit-flipping, afin de reconstruire l'image originale sans aucune connaissance de la clé secrète.

Mots clefs

Sécurisation des images, obscuration d'images, attaque.

1 Introduction

Depuis plus de dix ans, la transmission et l'archivage sur le cloud d'images et de vidéos ont augmenté de façon spectaculaire. Afin de garantir leur confidentialité, il est nécessaire de les sécuriser visuellement. De nombreuses études sur les communications sécurisées, efficaces et flexibles ont été rapportées [2]. Pour les données multimédia, le chiffrement intégral avec une sécurité éprouvée (tel que les chiffrements RSA ou AES) est l'option la plus sûre. Toutefois, de nombreuses applications multimédia nécessitent un compromis entre sécurité et d'autres exigences, telles qu'un traitement peu coûteux et la préservation du format. C'est pourquoi des méthodes de chiffrement sélectif ont été étudiées [3]. Parmi les approches visant à protéger la vie privée tout en maintenant la qualité et l'intégrité des données, l'obscuration d'images (comme le flou [4] ou le mélange des pixels [5]) est une solution possible. Ces techniques sont pertinentes pour des applications en anonymisation et en confidentialité des images, dans des domaines tels que la vidéo-surveillance, la télé-médecine ou les réseaux sociaux, par exemple. Les méthodes réversibles permettent de reconstruire l'image originale à l'aide d'une clé secrète. En 2021, Aprilpyone et Kiya ont proposé une méthode réversible d'obscuration d'image bloc par bloc par bit-flipping, basée sur l'inversion d'un sous-ensemble de pixels pour chaque bloc d'une image [1].

Dans cet article, nous proposons une attaque de la méthode d'obscuration par bit-flipping proposée par Aprilpyone et Kiya [1]. Sur la base d'une analyse des bits de poids fort de tous les pixels de l'image obscurcie, nous proposons de reconstruire le motif binaire qui a été généré à partir d'une clé secrète au cours de l'étape d'obscuration. Nous pouvons alors reconstruire deux versions possibles pour chaque composante de couleur de l'image et, en les combinant, obtenir la reconstruction de huit images couleur. Sur la base d'un classifieur binaire, nous pouvons alors déduire laquelle de ces huit images peut correspondre à l'image originale, le tout sans aucune connaissance de la clé secrète.

2 Attaque proposée

De nombreuses attaques contre les méthodes d'obscuration d'images ont été proposées, en particulier pour les systèmes de chiffrement d'images basés sur les mélanges de pixels [6]. À notre connaissance, aucune attaque n'a été proposée pour les méthodes d'obscuration par bit-flipping. Dans cette section, nous présentons une attaque de la méthode d'obscuration par bit-flipping [1].

2.1 Obscuration par bit-flipping [1]

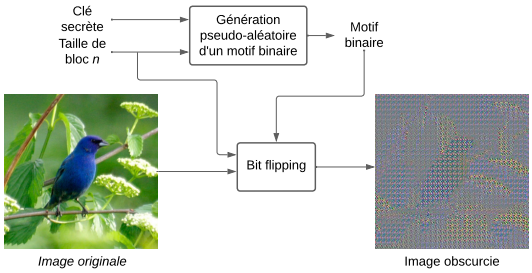


FIGURE 1 – *Obscuration d'images par bit-flipping [1].*

Dans la méthode d'obscuration d'images bloc par bloc proposée par Aprilpyone et Kiya [1], tous les bits d'un sous-ensemble de pixels de chaque bloc sont inversés. Cela est effectué séparément pour chaque composante couleur RGB, sur la base d'un motif binaire généré pseudo-aléatoirement par une clé secrète. Tous les blocs de $n \times n$ pixels de l'image subissent le même traitement, sur la base du même motif binaire. Comme l'illustre la figure 1, les principales étapes de cette méthode sont les suivantes :

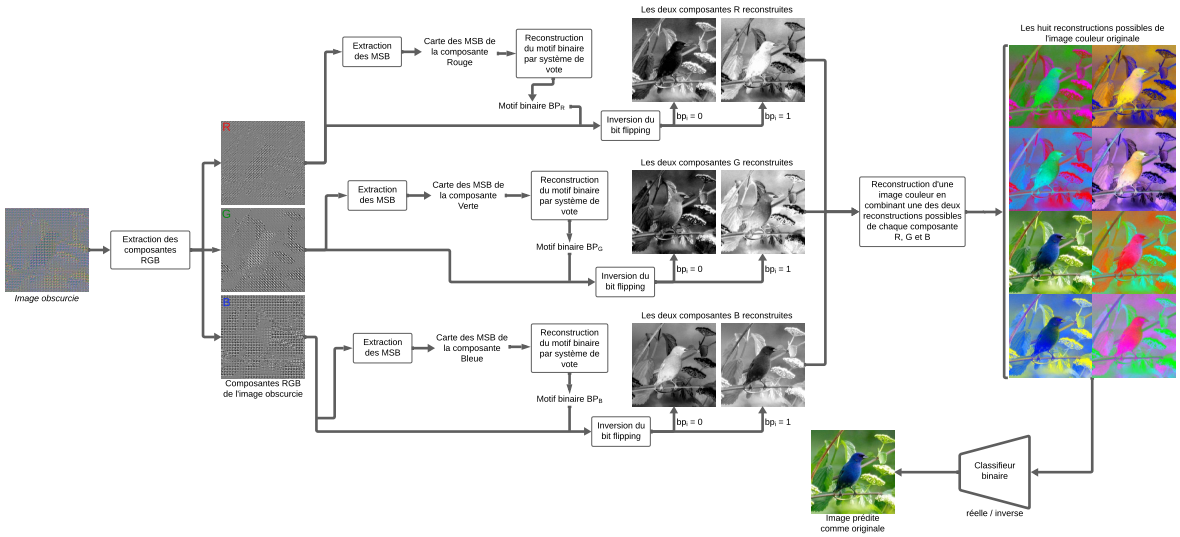


FIGURE 2 – *Attaque proposée pour la méthode d'obscuration par bit-flipping [1].*

1. Générer un motif binaire $\mathbf{BP}$ pseudo-aléatoire par composante couleur, de taille $n \times n$ bits, avec n le côté des blocs de pixels : $\mathbf{BP} = \{bp_0, \ldots, bp_i, \ldots, bp_{n^2-1}\}$.
2. Diviser chaque composante couleur de l'image en blocs de taille $n \times n$ pixels.
3. Lire les n^2 pixels p_i de chaque bloc de pixels et appliquer à chacun leur nouvelle valeur, telle que :

$$p'_i = \begin{cases} p_i, & \text{si } bp_i = 0, \\ p_i \oplus (2^L - 1), & \text{si } bp_i = 1, \end{cases} \quad (1)$$

avec L le nombre de bits par pixel pour composante couleur ($L = 8$ bits dans cet article).

Un point important dans cette méthode est qu'un seul motif binaire par composante de couleur est généré de manière pseudo-aléatoire. Il est ensuite utilisé de la même manière pour tous les blocs de pixels de l'image.

2.2 L'attaque proposée

La faille que nous exploitons dans la méthode par bit-flipping est qu'un motif binaire unique est utilisé pour l'ensemble des blocs de chaque composante de couleur d'une image. Comme ce motif binaire $\mathbf{BP}$ est appliqué de la même manière à tous les blocs, nous pouvons les analyser afin de le reconstruire, puis l'utiliser pour attaquer la méthode et tenter de reconstruire l'image originale sans la clé secrète. Cependant, pour cette attaque, nous supposons connaître le taille des blocs de pixels de l'image obscurcie. Pour chaque composante d'une image obscurcie, comme illustré en figure 2, nous appliquons d'abord les trois étapes suivantes de manière indépendante :

1. Diviser chaque composante couleur en blocs de taille $n \times n$ pixels et extraire le MSB (bit de poids fort) de chaque pixel.
2. Reconstruire le motif binaire de la composante couleur en votant sur les MSB extraits de chaque bloc et décider s'il s'agit d'un 0 ou d'un 1.
3. A partir du motif binaire reconstruit, pour chaque composante, appliquer l'opération inverse du bit-flipping à la composante couleur de l'image obscurcie. Toutefois, comme nous ne savons pas quel sous-ensemble de pixels a été inversé, nous pouvons reconstruire deux composantes différentes, inversées l'une par rapport à l'autre.

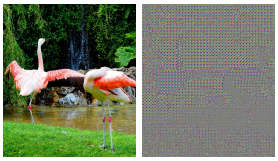
Ces trois premières étapes nous permettent d'obtenir 6 composantes différentes, avec deux versions reconstruites pour chaque composante couleur. À partir de ces 6 composantes, comme illustré en figure 2, nous pouvons reconstruire 8 images couleur différentes, dont une seule d'entre elles devrait correspondre à l'image originale. Pour déterminer laquelle de ces 8 images est prédite comme étant une image originale, comme illustré en figure 2, nous utilisons un modèle de classification binaire avec deux classes : "réelle" et "inverse".

3 Résultats expérimentaux

3.1 Exemple détaillé de l'attaque proposée

Nous appliquons d'abord notre attaque proposée à l'image Flamingo de 500×437 pixels de la base de données ILSVRC2012 (ImageNet Large Scale Visual Recognition Challenge 2012) [7], illustrée dans la figure 3.a. La figure 3.b montre les résultats obtenus avec la méthode d'obscuration par bit-flipping [1] avec des blocs de 10×10 pixels. A l'issue de cette obscuration, nous obtenons un PSNR [8] de 6,05 dB, un SSIM [9, 8] de 0,02, un UACI [10] de 26,95 %, un NPCR [10] de 10,98 % et un EDR [11] de 0,45.

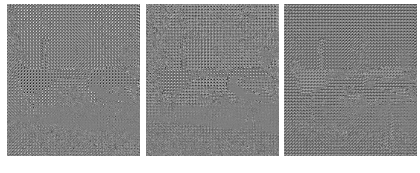
Pour l'attaque, la première étape consiste à décomposer l'image obscurcie, figure 3.b., en trois composantes cou-



(a) (b)

FIGURE 3 – *Obscuration par bit-flipping : a) Image originale issue de la base ILSVRC2012 [7], b) L'image obscurcie par bit-flipping [1] correspondante.*

leur, comme illustré dans la première ligne de figure 4. Sur la base du système de vote, nous pouvons alors reconstruire un motif binaire pour chaque composante couleur, comme illustré en seconde ligne de la figure 4.



(a) (b) (c)

(d) (e) (f)

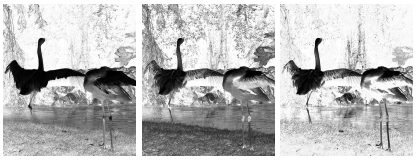
FIGURE 4 – *Les premières étapes de la méthode d'attaque proposée appliquée à l'image obscurcie Flamingo, figure 3.b : a), b) et c) Composantes RGB de l'image obscurcie, d), e) et f) Les motifs binaires déduits.*

À partir des trois composantes RGB de l'image obscurcie et de leurs motifs binaires reconstruits correspondants, illustrés en figure 4, nous pouvons alors reconstruire, pour chaque composante, deux images possibles en fonction du sous-ensemble de pixels que nous décidons d'inverser. Ainsi, à partir de l'image présentée dans la figure 3.b., nous reconstruisons six composantes couleur, deux par composante (figure 5.a à figure 5.f), et finalement nous avons huit combinaisons possibles pour reconstruire l'image originale comme illustré de la figure 5.g à la figure 5.n.
Pour déterminer laquelle des huit images reconstruites est la bonne, nous utilisons un classifieur binaire [1] qui a été entraîné pour deux classes : les images "réelles" (les images originales que nous recherchons) et les images "inversées" (les images dont au moins un des plans RGB est inversé, ce qui implique des couleurs erronées). Pour entraîner ce classifieur binaire, nous avons utliser les 50 000 images de validation de la base ILSVRC2012 [7] et appliqué une inversion aléatoire des composantes (inversion d'au moins une des trois composantes RGB), de manière équiprobable.

1. https://github.com/A-Jatin/CNN-implementation-for-binary-image-classification

Pour chaque image, les motifs binaires sont différents. 80 % de la base de données générée est utilisée pour l'entraînement, et 20 % pour tester l'attaque proposée.

(a) (b) (c)



(d) (e) (f)

(g) (h) (i) (j)

(k) (l) (m) (n)

FIGURE 5 – *Étapes suivantes de l'attaque proposée figure 4 : a) à f) Les possibles composantes RGB reconstruites, deux par composante, g) à n) Les huit images reconstruites possibles.*

Nous obtenons ainsi 80 000 images (40 000 "réelles" et 40 000 "inversées") dédiées à l'entraînement. Le classifieur binaire, conçu pour des images de taille (3, 128, 128) et composé de 4 couches (2 pour la convolution et 2 pour la partie fully connected) est entraîné pendant 50 epochs, avec une batch size de 32, en utilisant l'optimiseur Adam, et ReLU et Sigmoïde comme fonctions d'activation. Nous obtenons une précision de validation de 0,9714 %, sur les 20 000 images restantes (10 000 "réelles" et 10 000 "inverses").
En utilisant ce modèle entraîné, nous pouvons alors prédire si une image reconstruite par notre attaque correspond à une image "réelle" ou "inverse". Des huit images reconstruites présentées en figure 5, comme indiqué dans le tableau 1, nous constatons que seule la figure 5.g est prédite comme une image "réelle", tandis que les sept autres images reconstruites sont prédites comme des images "inverses". Et en effet, l'image reconstruite illustrée figure 5.g correspond bien à l'image originale. Nous avons donc réussi à reconstruire l'image originale sans la clé secrète, uniquement avec l'image obscurcie.

Image	Classe prédite	Probabilité
Originale	réelle	0.999
figure 5.g	réelle	0.999
figure 5.h	inverse	1.0
figure 5.i	inverse	1.0
figure 5.j	inverse	0.999
figure 5.k	inverse	1.0
figure 5.l	inverse	1.0
figure 5.m	inverse	1.0
figure 5.n	inverse	0.999

TABLEAU 1 – *Prédictions obtenues par le classifieur binaire pour les huit images reconstruites figure 5*

.

3.2 Analyse sur une plus large base d'images

Nous testons l'attaque proposée sur les 10 000 images restantes de la base de données présentée. Ces images ont toutes été obscurcies avec une clé différente. Après notre attaque, nous obtenons alors 80 000 images reconstruites.

Résultats		Nombre d'images	Total
Correctement prédites	Parfaitement reconstruites	8 063	9 201
	Partiellement reconstruites	1 138	
Incorrectement prédites		799	

TABLEAU 2 – *Prédiction pour les 10 000 images obscurcies attaquées par la méthode proposée.*

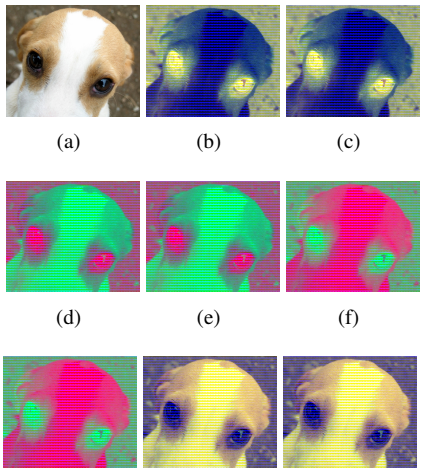
Le tableau 2 montre les résultats obtenus pour ces 10 000 images obscurcies. Parmi les 10 000 images obscurcies, le classifieur binaire a réussi à prédire 9 201 images comme étant la bonne image "réelle". Parmi ces images prédites comme "réelle", 8 063 sont parfaitement reconstruites (PSNR$= \infty$), les motifs binaires ont donc été parfaitement reconstruits. 1 138 images sont correctement prédites mais partiellement reconstruites, et conduisent à un PSNR moyen de 30 dB, comme illustré en figure 6. Cela est dû à un motif binaire mal reconstruit pendant la phase de vote, entraînant une inversion incomplète, et ainsi des différences imperceptibles ou des artefacts réguliers.

(a) (b)

FIGURE 6 – *Image partiellement reconstruite classifiée comme "réelle" : a) Image originale, b) Image reconstruite.*

Concernant les 799 images obscurcies que nous n'avons pas réussi à reconstruire comme images "réelles" avec notre attaque, plusieurs scénarios sont possibles :

— Les images sans aucune reconstruction prédite comme "réelle" (comme illustré sur la figure 7).
— Les images comportant au moins deux reconstructions prédites comme "réelles", dont la correcte, comme le montre l'illustration figure 8).
— Images avec au moins une reconstruction prédite comme "réelle", mais sans la correcte.

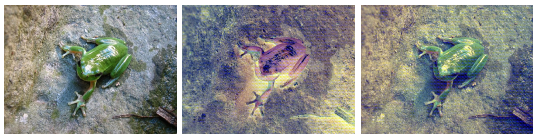


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 7 – *Exemple avec aucune des huit images prédite comme "réelle" : a) Image originale, b) à i) Les huit images reconstruites possibles, toutes prédites comme "inverses".*



(a) (b) (c)

FIGURE 8 – *Exemple avec deux images prédites comme "réelles" : a) Image originale, b) Image reconstruite incorrecte prédite comme "réelle" (PSNR = 9,17 dB), c) Image reconstruite correcte prédite comme "réelle" (PSNR = 21,07dB).*

4 Conclusion

Dans cet article, nous avons proposé une attaque de la méthode d'obscuration par bit-flipping qui ne nécessite que la taille des blocs comme paramètre pour reconstruire l'image originale. Cette attaque consiste en une analyse bloc par bloc de chaque composante couleur, afin de reconstruire le motif binaire original. Les résultats montrent qu'il est possible de reconstruire parfaitement l'image originale, même si parfois nous obtenons une image avec des artefacts. En perspective, nous envisageons d'appliquer notre système de vote que sur les blocs homogènes, puis, de développer une détection automatique de la taille des blocs.

Remerciements

Ce travail a bénéficié d'une aide de l'état gérée par l'Agence National de la Recherche au titre de France 2030 portant la référence ANR-22-PECY-0011.

Références

[1] Maungmaung Aprilpyone et Hitoshi Kiya. Block-Wise Image Transformation With Secret Key for Adversarially Robust Defense. *IEEE Transactions on Information Forensics and Security*, 16 :2709–2723, 2021.

[2] R.L. Lagendijk, Zekeriya Erkin, et Mauro Barni. Encrypted signal processing for privacy protection : Conveying the utility of homomorphic encryption and multiparty computation. *IEEE Signal Processing Magazine*, 30(1) :82–105, 2013.

[3] W. Puech, Z. Erkin, M. Barni, S. Rane, et R. L. Lagendijk. Emerging cryptographic challenges in image and video processing. Dans *2012 19th IEEE International Conference on Image Processing*, pages 2629–2632, 2012.

[4] Steven Hill, Zhimin Zhou, Lawrence K. Saul, et Hovav Shacham. On the (In)effectiveness of Mosaicing and Blurring as Tools for Document Redaction. *Proceedings on Privacy Enhancing Technologies*, 2016 :403 – 417, 2016.

[5] Kenta Kurihara, Sayaka Shiota, et Hitoshi Kiya. An encryption-then-compression system for JPEG standard. Dans *2015 Picture Coding Symposium (PCS)*, pages 119–123, 2015.

[6] Tatsuya Chuman, Kenta Kurihara, et Hitoshi Kiya. On the security of block scrambling-based ETC systems against jigsaw puzzle solver attacks. Dans *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2157–2161, 2017.

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, et Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3) :211–252, 2015.

[8] Alain Horé et Djemel Ziou. Image Quality Metrics : PSNR vs. SSIM. Dans *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[9] Zhou Wang, A.C. Bovik, H.R. Sheikh, et E.P. Simoncelli. Image quality assessment : from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4) :600–612, 2004.

[10] Guanrong Chen, Yaobin Mao, et Charles K Chui. A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons & Fractals*, 21(3) :749–761, 2004.

[11] Deok-Han Kim et Young-Gab Kim. A Method for De-Identification Analysis of Encrypted Video. Dans *2024 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON)*, pages 233–236, 2024.