Utilisation d'insertions adverses pour améliorer la stéganographie

Solène BERNARD[1], Tomáš PEVNÝ[2], Patrick BAS[1], John KLEIN[1]

[1]CRIStAL, Centre de Recherche en Informatique, Signal et Automatique de Lille
[2]CVUT, Czech Technical University in Prague
solene.bernard@univ-lille.fr, pevnak@protonmail.ch
patrick.bas@centralelille.fr, john.klein@univ-lille.fr

Résumé – Ce travail propose un protocole pour construire itérativement une fonction de distorsion qui permette d'accroître la sécurité du schéma d'insertion après chaque itération. Il allie l'attaque d'un schéma de stéganalyse ciblé avec une stratégie $\min\max$ qui sélectionne à chaque itération l'image *stego* la plus difficile associée au meilleur classifieur. Appliqué à J-Uniward, ce nouveau protocole accroît P_{err} estimée par Xu-Net de 7% à 20%, et de 10% à 23% pour la stéganalyse non ciblée par un classifieur linéaire avec des caractéristiques GFR.

Abstract – This work proposes a protocol to iteratively build a distortion function for adaptive steganography with increased practical security after each iteration. It combines a prior-art targeted attack technique on a given detector with a $\min\max$ strategy to dynamically select the most difficult stego content associated with the best classifier at each iteration. Applied on J-Uniward, this new protocol increases P_{err} from 7% to 20% when the attacked detector is Xu-Net, and from 10% to 23% for a non-targeted steganalysis by a linear classifier with GFR features.

1 Introduction

Une image $\mathbf{x} = (x_i) \in \mathcal{X}$ est *cover* quand elle ne contient pas de contenu caché. Une image $\mathbf{y} = (y_i) \in \mathcal{Y}$ désigne la version *stego* (*i.e.* avec contenu caché) de $\mathbf{x}$. Le but du stéganographe est d'insérer un message dans toute image *cover* tout en minimisant la probabilité de détection de ce dernier. Dans la problématique abordée par le présent travail, le stéganographe a accès à une base d'images de taille N et à un outil réputé être utilisé par l'adversaire (stéganalyste) f qui évalue la probabilité qu'une image soit *stego*.

Les schémas stéganographiques les plus performants sont basés sur le *principe de minimisation de la distorsion*, qui consiste à assigner des coûts de modification à chaque élément d'une image *cover*. Durant le processus d'insertion, l'objet *cover* est modifié afin de communiquer un message tout en minimisant la distorsion totale mesurée par la somme des coûts de modification, soit :

$$D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{H\times W} \rho_i^+ \delta_1(y_i - x_i) + \rho_i^- \delta_{-1}(y_i - x_i). \quad (1)$$

Il existe plusieurs manières pour calculer les coûts ρ_i, qui correspondent chacune à un schéma stéganographique différent. Ci-dessous, nous décrivons un schéma qui permet d'ajuster les coûts ρ_i afin de générer une image *stego* antagoniste (ou bien dite adverse) qui va contrer un adversaire particulier f.

La méthode ADV-EMB [1] consiste à rectifier les coûts ρ_i^+ et ρ_i^- de modification $+1$ ou -1 du $i-$ième élément, qui étaient préalablement calculés par un schéma stéganographique, afin de tromper la détection du classifieur f. Ainsi les coûts sont mis à jour comme suit :

$$q_i^+ = \begin{cases} \rho_i^+/\alpha & \text{si } -\frac{\partial f}{\partial x_i} > 0, \\ \rho_i^+ & \text{si } -\frac{\partial f}{\partial x_i} = 0, \\ \rho_i^+\alpha & \text{si } -\frac{\partial f}{\partial x_i} < 0, \end{cases} \quad (2)$$

où $\frac{\partial f}{\partial x_i}$ est la dérivée partielle de f par rapport à la valeur du $i-$ième pixel de valeur x_i. α est un paramètre que les auteurs fixent à 2. Les coûts q_i^- sont modifiés de manière analogue, en inversant le sens des inégalités. On note $\mathbf{z} = (z_i) \in \mathcal{Z}$ l'image *stego* obtenue sur la base des coûts $q_i^\pm$, *i.e.* par attaque du stéganalyste. Notons que $\mathcal{Z} \subset \mathcal{Y}$.

Cette attaque nécessite d'abord de calculer les coûts ρ_i^+ et ρ_i^- donnés par un schéma stéganographique arbitraire (ici J-Uniward). Ensuite l'insertion consiste à modifier le minimum de coefficients selon l'Equation (2) tel que l'image *stego* obtenue soit classifiée *cover* par le classifieur.

2 Itérations sur des attaques adverses

En faisant l'hypothèse d'équiprobabilité *stego*/*cover*, le taux d'erreur $P_{\text{err}}(f|P_{\mathcal{X}}, P_{\mathcal{Y}})$ d'un classifieur[1] $f : \mathbb{X} \mapsto \mathbb{R}$ sur les objets *cover* et *stego*, tirés selon les distributions $P_{\mathcal{X}}$ et $P_{\mathcal{Y}}$, est donné par :

$$P_{\text{err}}(f|P_{\mathcal{X}}, P_{\mathcal{Y}}) = \frac{1}{2}\mathbb{E}_{P_{\mathcal{X}}}[\mathbb{I}\{f > 0\}] + \frac{1}{2}\mathbb{E}_{P_{\mathcal{Y}}}[\mathbb{I}\{f < 0\}], \quad (3)$$

1. Sans perte de généralité, il est supposé que la sortie du classifieur f est positive (resp. négative) si l'objet $\mathbf{x}$ est classifié *stego* (resp. *cover*). Ainsi, pour simplifier la lecture, nous appelons f un classifieur, alors qu'il s'agit de la fonction discriminante dont nous devons regarder le signe pour obtenir la classe d'une image quelconque.

où $\mathbb{I}$ est la fonction indicatrice. Comme le stéganographe vise à maximiser son indétectabilité vis-à-vis d'une classe d'adversaires $\mathcal{F}$, le stéganographe choisit une fonction d'insertion h_{ins} qui maximise l'erreur du meilleur classifieur que l'adversaire détient, soit :

$$\max_{h_{\text{ins}}} \min_{f \in \mathcal{F}} P_{\text{err}}(f | P_{\mathcal{X}}, P_{\mathcal{Y}}(h_{\text{ins}})), \tag{4}$$

où $P_{\mathcal{Y}}(h_{\text{ins}})$ est la distribution des images *stego* générées par la fonction d'insertion h_{ins} à partir des images *cover* issues de la distribution $P_{\mathcal{X}}$. Ce problème est évidemment non trivial car $\mathcal{F}$ peut en théorie contenir une infinité de fonctions f. En pratique, $\mathcal{F}$ est réduit à un ensemble fixé de fonctions paramétrisées par $\theta \in \Theta$. Ces fonctions sont par exemple l'ensemble des réseaux de neurones convolutifs ayant une architecture fixée, où θ sont ses paramètres.

En supposant que l'ensemble de fonctions $\mathcal{F}$ est fini (dans le but de trouver une solution à l'équation (4)), nous pouvons utiliser les méthodes [1, 2, 3] afin d'insérer un message m dans la *cover* $\mathbf{x}$ et de créer la *stego* $\mathbf{y}$ en minimisant la performance du meilleur classifieur $f \in \mathcal{F}$ au lieu de maximiser P_{err}, *i.e.* :

$$\min_{\mathbf{y}} \max_{f \in \mathcal{F}} f(\mathbf{y}). \tag{5}$$

L'équation (5) transforme le problème de détermination de la fonction de distorsion en un problème de définition de l'ensemble de fonctions $\mathcal{F}$, qui doit être suffisamment riche pour détecter toutes sortes d'attaques stéganographiques, tout en étant assez petit afin de rendre la minimisation possible. Le reste de cette section vise à présenter un protocole itératif afin de construire cet ensemble $\mathcal{F}$ de classifieurs pour un nombre fixé d'architectures de réseaux de neurones.

2.1 Fonction de distorsion $\min\max$

Le protocole commence par la création d'un ensemble d'images *stego* $\mathcal{Y}^0$ en utilisant le schéma stéganographique initial h_{ins}. Ensuite, un classifieur f^0 est entraîné afin de classifier les images *cover* $\mathcal{X}$ et $\mathcal{Y}^0$. Il est ajouté à l'ensemble des classifieurs disponibles $\hat{\mathcal{F}}_0 = \{f^0\}$. Dans un souci de simplification des notations, nous notons $\mathcal{Z}^0 = \mathcal{Y}^0$.

À l'itération suivante, le stéganographe génère l'ensemble d'images *stego* $\mathcal{Z}^1$ en attaquant f^0. Ensuite, à partir de $\mathcal{Z}^0$ et $\mathcal{Z}^1$, le stéganographe crée le nouvel ensemble d'images *stego* $\mathcal{Y}^1$ en sélectionnant à chaque fois la version *la plus difficile* par rapport à l'ensemble des classifieurs disponibles $\hat{\mathcal{F}}^0$ (qui pour le moment ne contient que f^0), c'est-à-dire :

$$\mathcal{Y}^1 = \left\{ \mathbf{z} \,\middle|\, \mathbf{z} = \underset{\mathbf{z} \in \{\mathbf{z}_i^0, \mathbf{z}_i^1\}}{\arg\min} \max_{f \in \hat{\mathcal{F}}^0} f(\mathbf{z}) \, , 1 \leq i \leq N \right\}, \tag{6}$$

où $\mathbf{z}_i^{\cdot}$ représente l'image *stego* de l'ensemble $\mathcal{Z}^{\cdot}$ générée à partir de l'image *cover* $\mathbf{x}_i \in \mathcal{X}$. Le stéganographe continue ensuite en créant le nouveau classifieur f^1, qui classifie $\mathcal{X}$ et $\mathcal{Y}^1$, puis l'ajoute à $\hat{\mathcal{F}}^0$. Le nouvel ensemble de classifieurs est $\hat{\mathcal{F}}^1 = \hat{\mathcal{F}}^0 \cup \{f^1\}$.

À la $k^{\text{ième}}$ itération du protocole, le stéganographe crée l'ensemble d'images $\mathcal{Z}^k$ qui attaquent le dernier classifieur f^{k-1}. Les ensembles $\mathcal{Z}^0, \mathcal{Z}^1, \ldots, \mathcal{Z}^k$ sont ensuite utilisés afin de créer le nouvel ensemble d'image *stego* $\mathcal{Y}^k$ en sélectionnant les images qui maximisent leur indétectabilité par tous les classifieurs $f \in \hat{\mathcal{F}}^{k-1}$, *i.e.* :

$$\mathcal{Y}^k = \left\{ \mathbf{z} \,\middle|\, \mathbf{z} = \underset{\mathbf{z} \in \{\mathbf{z}_i^0, \mathbf{z}_i^1, \ldots, \mathbf{z}_i^k\}}{\arg\min} \max_{f \in \hat{\mathcal{F}}^{k-1}} f(\mathbf{z}) \, , 1 \leq i \leq N \right\}. \tag{7}$$

Les images $\mathcal{Y}^k$ sont ensuite utilisées pour entraîner un nouveau classifieur f^k pour classifier les *cover* $\mathcal{X}$ et *stego* $\mathcal{Y}^k$, qui est par la suite ajouté à l'ensemble des classifieur, *i.e.* $\hat{\mathcal{F}}^k = \hat{\mathcal{F}}^{k-1} \cup \{f^k\}$. La méthode qui permet de créer $\mathcal{Y}^k$ en utilisant le protocole $\min\max$ est illustrée dans la Figure 1.

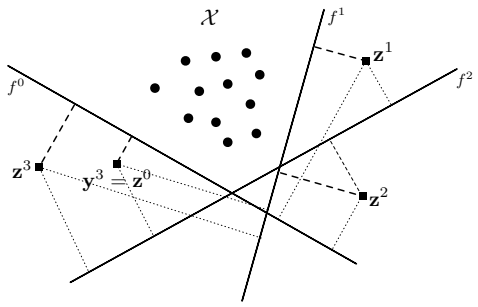

FIGURE 1 – Illustration de la création d'une image *stego* $\mathbf{y}^3$ à l'itération $k = 3$ par la stratégie $\min\max$ à partir de $\{\mathbf{z^0}, \mathbf{z^1}, \mathbf{z^2}, \mathbf{z^3}\}$. On suppose que les classifieurs sont linéaires et que la valeur $f^i(\mathbf{z}^j)$ est la distance algébrique par rapport au plan de séparation. Les lignes en traitillé correspondent aux distances positives, les lignes en pointillé à celles négatives, et les lignes en gras correspondent à $\max_i f^i(\mathbf{z}^j)$. Ici la stratégie $\min\max$ donne $\mathbf{y}^3 = \mathbf{z}^0$.

Notons que la sécurité de cet algorithme de stéganographie dépend de deux facteurs : (i) l'ensemble de tous les classifieurs possibles $\mathcal{F}$; (ii) la qualité de l'attaque d'un classifieur $f \in \mathcal{F}$. Ainsi, apporter une amélioration à l'un de ces deux points amène à l'amélioration global du schéma.

3 Expériences

3.1 Choix de l'ensemble des classifieurs $\mathcal{F}$

Dans ce papier, nous voyons les réseaux de neurones comme un outil permettant de choisir un classifieur f parmi la classe de fonctions $\mathcal{F}$, tel que f minimise le taux de mauvaise classification (3) :

$$\hat{P}_{\text{err}}(f; \mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{I}\{f(x) > 0\} + \frac{1}{|\mathcal{Y}|} \sum_{x \in \mathcal{Y}} \mathbb{I}\{f(x) < 0\}. \tag{8}$$

Les réseaux de neurones convolutifs (CNN) ont le bon goût d'être différentiables, c'est-à-dire qu'on peut calculer $\frac{\partial f}{\partial x}$ pour tout x et pour tout $f \in \mathcal{F}$.

C'est pourquoi nous décidons que l'ensemble des classifieurs $\mathcal{F}$ est égal à l'ensemble des CNNs qui ont pour archi-

tecture celle de Xu-Net [4]. Nous choisissons ce classifieur car il atteint de très bonnes performances sur la stégalanyse des images JPEG, et parce que son entraînement nécessite moins de ressources en temps et mémoire que SRNet [5] par exemple.

3.2 Détails des expériences

Insertion : L'algorithme d'insertion utilisé pour initialiser le protocole pour calculer les coûts de modifications des coefficients DCT est J-Uniward [6]. Nous utilisons la version JPEG de la base d'images BossBase [7] de taille 512×512 en niveau de gris et compressées avec un facteur de qualité de 75. Un taux d'insertion de 0.4 bits par coefficient AC DCT non nul (bpnzac) est choisi.

Classification/Stéganalyse : L'implémentation proposée de Xu-Net utilise la librairie TensorFlow. Les poids du classifieur sont initialisés aléatoirement (bruit Gaussien de moyenne nulle et d'écart-type 0.01), et utilise 2×4000 images *cover* et *stego* pour l'entraînement, et les 2×6000 restantes pour le test. 290 époques sont utilisées pour l'entraînement avec l'algorithme d'optimisation ADAM avec pour taux d'apprentissage initial de 0.001 qui est diminué toutes les 5000 itérations à 0.9 fois sa valeur. Les autres paramètres d'ADAM sont ceux par défaut. La taille du *mini-batch* est de 64 (32 paires *cover-stego*) et l'entraînement utilise les images entières de taille 512×512 pixels. La configuration qui donne la plus basse erreur de classification sur la base d'entraînement est utilisée comme le résultat de l'entraînement. Les expériences ont été réalisées avec la carte Nvidia GPU Quadro P6000 (24 GB de mémoire). Chaque itération dure approximativement 35 heures : 30h pour entraîner XU-Net et 5h pour générer la base de *stego* antagonistes (en parallèle sur 36 cœurs de CPU).

Attaque : L'attaque ADV-EMB décrite dans la Section 1 est implémentée afin de modifier les coefficients DCT selon (2). Pour calculer la dérivée partielle $\frac{\partial f}{\partial x_i}$ par rapport au $i-$ième coefficient DCT, et parce que Xu-Net admet des images spatiales non arrondies en entrée, l'IDCT est effectuée dans une couche supplémentaire ajoutée à l'entrée du réseau. La dérivée par rapport à l'image dans le domaine JPEG est alors calculée par différentiation automatique par la fonction `tf.gradient()` de la librairie TensorFlow.

Les autres stratégies d'insertion : Dans le papier [1] sont suggérées deux stratégies itératives, que l'on appelle dans la suite "Stratégie aléatoire" et "Stratégie de la dernière itération". La stratégie aléatoire diffère de la stratégie $\min\max$ à l'étape de l'attaque lors de la création de l'ensemble d'images $\mathcal{Y}^k$. En effet, le stéganographe choisit à la place au hasard à chaque fois une image issue de l'ensemble $\mathcal{Z}^0, \mathcal{Z}^1, \dots$ ou $\mathcal{Z}^k$. Ainsi le stéganalyste entraîne le classifieur f^k entre les *cover* $\mathcal{X}$ et les *stego* $\mathcal{Y}^k_{\text{random}} = \mathcal{Z}^0 \cup \mathcal{Z}^1 \cup \cdots \cup \mathcal{Z}^k$. La stratégie de la dernière itération s'obtient à partir de la stratégie aléatoire, seulement au lieu de choisir au hasard, le stéganographe choisit les *stego* issues de la dernières itération, *i.e.* il envoie l'attaque $\mathcal{Y}^k_{\text{last-it}} = \mathcal{Z}^k$. Ainsi cette fois $f^k_{\text{last-it}}$ est entraîné entre $\mathcal{X}$ et $\mathcal{Y}^k_{\text{last-it}}$. Dans la Table 1, les résultats de ces deux stratégies sont comparées à la stratégie $\min\max$ pour les cinq premières itérations.

Évaluation : Afin d'évaluer la sécurité de ce nouveau protocole vis-à-vis des méthodes de stéganalyse différentes de celui pris pour cible (ici XU-Net), nous calculons les caractéristiques DCTR [8] et GFR [9]. Les ensembles d'entraînement et de test sont constitués chacun de 5000 paires d'images *cover* et *stego*. Le classifieur linéaire régularisé [10] est utilisé pour calculer P_{err}. On entraîne également SRNet [5] sur 290 époques pour seulement deux itérations : la première et la dernière, à cause de son coût de calcul. Les images utilisées sont toujours de taille 512×512, la taille du *mini-batch* est de 16 (8 paires *cover-stego*).

3.3 Résultats expérimentaux

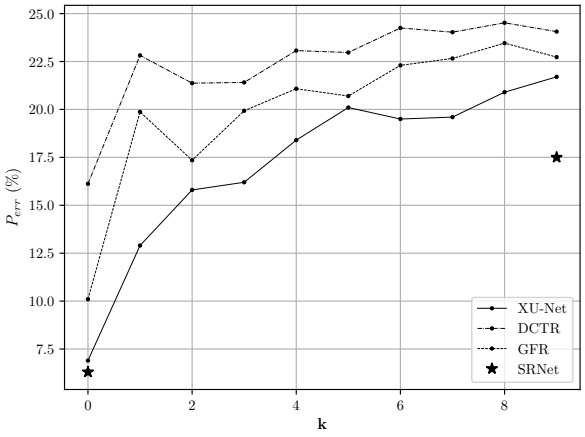

FIGURE 2 – P_{err} des classifieurs entraînés et évalués à chaque fois sur $\mathcal{X}$ et $\mathcal{Y}^k$) pour chaque itération k. L'algorithme est optimisé par rapport à Xu-Net. Les classifieurs DCTR et GFR sont basés sur l'association de caractéristiques stéganographiques (DCTR [8] et GFR [9]) et l'analyse discriminante linéaire régularisée de Fisher [10].

La Figure 2 montre l'évolution de l'erreur P_{err} en fonction du paramètre k pour différents classifieurs f^k qui ont été entraînés pour classifier les images *cover* $\mathcal{X}$ et *stego* $\mathcal{Y}^k$ envoyées par le stéganographe à la k^{ieme}-itération du protocole. Les images *stego* $\mathcal{Y}^k$ sont sélectionnées de manière à ce qu'elles soient le moins détectables par tous les classifieurs précédents $\{f^0, f^1, \dots, f^{k-1}\}$, ce qui signifie qu'elles ne sont pas optimisées par rapport au classifieur f^k utilisé pour évaluer la sécurité du schéma (et qui est donc le meilleur classifieur en possession du stéganalyste car il est optimisé pour détecter

la dernière attaque du stéganographe). Ainsi l'évaluation du scénario obéit au principe de Kerckhoffs : le stéganalyste connaît la stratégie du stéganographe.

Cette figure montre également que le taux d'erreur augmente pour les différents schémas de stéganalyse utilisés, ce qui signifie que le schéma stéganographique devient en moyenne plus sécurisé au fur et à mesure des itérations. Si nous évaluons la qualité de la procédure par l'erreur P_{err} du classifieur Xu-Net, la sécurité a augmenté de 6.9% à 21.7% au bout de neuf itérations, ce qui est très satisfaisant.

Si nous évaluons la sécurité de la stratégie obtenue à l'aide de Xu-Net avec d'autres schémas stéganographiques, l'évolution des autres schémas est similaire : il y a augmentation et convergence du taux d'erreur au fur et à mesure des itérations. L'amélioration apportée par le protocole proposé dans ce papier n'est ainsi pas propre au stéganalyste cible, mais généralisable à d'autres méthodes de stéganalyse.

Dans la Table 1, la stratégie d'insertion (7) est comparée aux stratégies "aléatoire" et "dernière itération". Nous remarquons que la stratégie $\min\max$ offre une meilleur sécurité jusqu'à l'itération $k = 5$, avec un gain de 4.7% par rapport à la stratégie aléatoire à l'itération 5, et 5% par rapport à la "dernière itération" à l'itération 2. [2]

k	0	1	2	3	4	5
$\min\max$	6.9	12.9	15.8	16.2	18.4	20.1
Aléatoire	6.9	12.1	12.4	16.2	15.3	15.4
Dernière itération	6.9	12.1	10.5	–	–	–

TABLE 1 – P_{err} (en %) donné par les différentes stratégies pour les cinq premières itérations : stratégie $\min\max$, stratégie aléatoire et stratégie de la dernière itération données dans [1].

4 Conclusions et perspectives

Ce papier propose un schéma stéganographique qui utilise une insertion antagoniste, basée sur une stratégie $\min\max$. Ce protocole peut être utilisé pour améliorer le schéma stéganographique classique basé sur le calcul des coûts comme J-Uniward, mais il peut en pratique être appliqué à n'importe quel schéma qui permet d'ajuster les coûts de modification. Nos test montrent l'avantage (i) du caractère itératif du protocole qui permet d'améliorer la sécurité des objets *stego* et (ii) de l'utilisation d'une stratégie $\min\max$ à la place d'une stratégie aléatoire (qui consiste à choisir au hasard une *stego* issue de l'une des itérations passées) ou d'une stratégie qui consiste à choisir les *stego* issues de l'itération passée.

On remarque que, contrairement aux autres insertions antagonistes effectuées par les GANs [11], ce schéma repose sur l'attaque d'un classifieur cible, ce qui permet d'accélérer la convergence et d'améliorer l'efficacité.

2. La version finale du papier présentera les résultats après des itérations supplémentaires, résultats non obtenus à temps pour la soumission de ce papier.

De prochains travaux seront dédiés à déterminer de meilleures stratégies pour ajuster les coûts, ainsi que de nouvelles manières d'attaquer un classifieur.

Références

[1] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "Cnn-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, 2019.

[2] T. Pevný and A. D. Ker, "Exploring non-additive distortion in steganography," in *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2018, pp. 109–114.

[3] T. Filler and J. Fridrich, "Gibbs construction in steganography," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 4, pp. 705–720, 2010.

[4] G. Xu, "Deep convolutional neural network to detect j-uniward," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*. ACM, 2017, pp. 67–73.

[5] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, 2019.

[6] V. Holub, J. Fridrich, and T. Denemark, "Universal distortion function for steganography in an arbitrary domain," *EURASIP Journal on Information Security*, vol. 2014, no. 1, p. 1, 2014.

[7] P. Bas, T. Filler, and T. Pevný, ""break our steganographic system" : The ins and outs of organizing boss," in *International Workshop on Information Hiding*, vol. 6958, LNCS. Springer Berlin Heidelberg, 2011, pp. 59–70.

[8] V. Holub and J. Fridrich, "Low-complexity features for jpeg steganalysis using undecimated dct," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 2, pp. 219–228, 2015.

[9] X. Song, F. Liu, C. Yang, X. Luo, and Y. Zhang, "Steganalysis of adaptive jpeg steganography using 2d gabor filters," in *Proceedings of the 3rd ACM workshop on information hiding and multimedia security*. ACM, 2015, pp. 15–23.

[10] R. Cogranne, V. Sedighi, J. Fridrich, and T. Pevný, "Is ensemble classifier needed for steganalysis in high-dimensional feature spaces ?" in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015, pp. 1–6.

[11] W. Tang, S. Tan, B. Li, and J. Huang, "Automatic steganographic distortion learning using a generative adversarial network," *IEEE Signal Processing Letters*, vol. 24, no. 10, pp. 1547–1551, 2017.