LEARNING SYMMETRIC AND ASYMMETRIC STEGANOGRAPHY VIA ADVERSARIAL TRAINING

Zheng Li, Ge Han, Yunqing Wei, Shanqing Guo

Shandong University
zheng.li@mail.sdu.edu.cn, hangehg@126.com, wei895177311@gmail.com, guoshanqing@sdu.edu.cn

ABSTRACT
Steganography refers to the art of concealing secret messages within multiple media carriers so that an eavesdropper is unable to detect the presence and content of the hidden messages. In this paper, we firstly propose a novel key-dependent steganographic scheme that achieves steganographic objectives with adversarial training. Symmetric (secret-key) and asymmetric (public-key) steganographic scheme are separately proposed and each scheme is successfully designed and implemented. We show that these encodings produced by our scheme improve the invisibility by 20% than previous deep-learning-based work, and further that perform competitively remarkable undetectability 25% better than classic steganographic algorithms. Finally, we simulated our scheme in a real situation where the decoder achieved an accuracy of more than 98% of the original message.

Index Terms— steganography, adversarial training, symmetric, asymmetric

1. INTRODUCTION
The security of communication is of increasing concern, as the ongoing development of computers and the Internet provides a cheap and fast way for data transmission. Steganography, as an approach to protect communication, plays an important role in hiding the presence of sensitive information within transmitted messages, which has much practical significance in military, e-commerce, government affairs, copyright protection, etc.

The formulation of steganography in the modern scientific literature may be traced back to the year 1983, in which Simmons modeled it as the prisoner problem [1]. There are two prisoners Alice and Bob, separately locked in jail cells and wishing to plot some secret plan of escaping. However, the prison warden, Eve, can monitor any communication between Alice and Bob, and if he detects any hint of “unusual” communications, Alice and Bob would be removed into solitary confinement as punishment. Steganography solves this problem by concealing secret digital data into an appropriated multimedia carrier, e.g. text, image, audio and video, so that the eavesdropper Eve would regard the non-secret transmission of carrier files between Alice and Bob as innocent communication.

Nowadays, classic steganographic algorithms, such as WOW [2], HUGO [3] and S-UNIWARD [4], conceal the secret information into cover images spatial domain or transform domains by hand-crafted rules. However, they can systematically alter the statistics of the image, leading to reliable detection [5]. With the rise of deep learning in recent years, deep learning has been applied to steganography [6, 7]. The goal of Volkonskiy et al. [6] is to generate images suitable for steganography, whereas we seek to train a model that learns a steganographic algorithms by itself. Hayes and Danezi [7] proposed a GAN-based model using a fully connected networks to hide information resulting in weak invisibility. Furthermore, their work did not take the usage of keys into consideration. Nevertheless, existing works in steganography [8, 9] have formally defined and discussed the great importance of keys for security.

The main contributions of this work include:

• Symmetric and asymmetric steganographic schemes are designed and successfully implemented without any prescribed rules or hand-crafted features, further that achieved an accuracy of more than 98% of original messages in a simulation of real situation,

• Compared to previous deep-learning-based work, we take usage of convolutional layers and additionally use the structure similarity index (SSIM) as metric, achieving better performance about 20% in invisibility,

• Defending against the existing steganalysis tool, resultant algorithms have outstanding performance 25% better than classic steganographic methods.

2. RELATED WORK

2.1. Steganography
A wide variety of steganography settings and methods have been proposed in the literature. One of the most popular and
easy-to-implement steganographic algorithms is the Least Significant Bit (LSB) algorithm [10]. Its main idea is to store the secret message in the least significant bits (last bits) of some color channel of each pixel in a given image container; Others change mid-frequency components in the frequency domain [11]. There are several popularly used and sophisticated steganographic algorithms developed from above, like WOW [2], HUGO [3] and S-UNIWARD [4]. The compared details is in section 5.

2.2. Adversarial training

Recently, Adversarial training have received more and more attention, achieved the state-of-art performance on tasks such as image generation, style transfer, speech synthesis and so on. It normally consists of generator network for generating data, a discriminator network for discriminating generated data from real data. Volkhonskiy et al. introduced a steganographic generative adversarial networks model (SGAN) to generate images as steganographic containers, which is in different goal and way to our scheme, Hayes and Danezis [7] also applied adversarial training to the design of steganographic algorithms, but the invisibility is weak. The compared detail is in section 5.

3. STEGANOGRAPHIC ADVERSARIAL TRAINING

3.1. Scheme overview

This section discusses our key-dependent steganographic scheme, the models we use and the information each party wishes to conceal or reveal. Each of Alice, Bob and Eve is designed as a convolutional networks with trainable parameters \( \theta_A, \theta_B \) or \( \theta_E \) and our training method is exactly a process to adjust these parameters. The detailed adversarial training schemes of symmetric and asymmetric steganography are separately described in the rest of this section.

3.2. Symmetric

Symmetric steganography, allows Alice and Bob to communicate with some shared secret. As depicted in Figure[1]

The encoder Alice receives a cover image \( I \) of shape \( C \times H \times W \), a binary secret message \( M \in \{0,1\}^{L_M} \) of length \( L_M \) and a binary random secret key \( sk \in \{0,1\}^{L_{sk}} \) of length \( L_{sk} \). The message \( M \) and key \( sk \) is firstly concatenated and resized to a tensor of size \( 1 \times h \times w \) where \( h \times w = (L_M + L_{sk}) \), then fed into several transposed convolutional layers generating a feature map of size \( C \times H \times W \), the same size of cover image \( I \). The feature map and \( I \) are concatenated to a tensor of size \( 2C \times H \times W \) which the feature map is dispersed throughout the bits in the image, then fed to several residual convolutional layers to reduce the number of channels from \( 2C \) to \( C \). Finally we get the output of Alice, the stego image \( I' \) of size \( C \times H \times W \).

The decoder Bob accepts as input the stego image and the secret key. The secret key \( sk \) firstly is resized and fed to several convolutional layers generating a feature map. It is concatenated with the stego image \( I' \), which is then fed into last several convolutional layers, outputting the reconstructed message \( M' \). The adversary Eve has a structure similar to the decoder, but it outputs a binary classification \( p(\tilde{I}) \) instead, given an image \( \tilde{I} \in \{I, I'\} \), i.e. either a cover image or stego image.

Unlike previous works [7], they just use the mean square error (MSE) or the the Euclidean distance, which only punishes the large errors of the corresponding pixels between two images, ignoring the underlying structure in the images. The Human Visual System (HVS) is more sensitive to brightness and color changes in non-textured areas. So we introduce the structure similarity index (SSIM). The SSIM index separates the task of similarity measurement into three comparisons: luminance, contrast and structure.

\[
MSE(I, I') = \frac{1}{n} \sum (x_i - y_i) \quad (1)
\]

\[
SSIM(I, I') = \frac{(2\mu_I \mu_{I'} + c_1)(2\sigma_{I \mid I'} + c_2)}{\mu_I^2 + \mu_{I'}^2 + c_1 \sigma_{I \mid I'}^2} \quad (2)
\]

Considering pixel value differences and structure differences simultaneously, we put MSE (1) and SSIM (2) together. The value range of the SSIM index is \([0,1]\). The higher the index is, the more similar the two images are. So the loss (metric) between cover and stego image is as below:
\(\text{Loss}_I(I, I') = \alpha \text{MSE}(I, I') + \beta (1 - \text{SSIM}(I, I'))\) (3)

where \(\alpha, \beta > 0\) are the hyperparameters directing the trade off among the importance of each individual metric term.

We set Bobs loss (the secret message reconstruction loss), to be the L1 distance between \(M\) and \(M'\):

\[\text{Loss}_M(M, M') = \sum_i |M_i - M'_i|\] (4)

And adversarial loss, the ability of the adversary Eve to detect a stego image \(I'\):

\[\text{Loss}_p(I') = \log (1 - p(I'))\] (5)

The secret key \(sk\) also plays an important role in the design of steganographic algorithms. It is a bridge to strengthen the link between Alice and Bob, directing them to learn how to encode and decode. We perform stochastic gradient descent (SGD) on updating \(\theta_A\) and \(\theta_B\), then obtain the optimal Alice and Bob by minimizing the following loss over the distribution of input images, messages and key:

\[
E_{I,M,sk} = \text{Loss}_I(I, I') + \gamma \text{Loss}_M(M, M') + \delta \text{Loss}_p(I')
\]

\[O_{AB}(\theta_A, \theta_B) = \arg \min_{\theta_A, \theta_B} E_{I,M,sk}\] (6)

where \(\gamma, \delta > 0\) are the hyperparameters to trade of the quality of stego images, revealed secret message and the ability of defending against Eve’s detecting.

The steganalyzer Eve incurs a classification loss function from its predictions:

\[\text{Loss}(\tilde{I}) = (1 - \log(p(I))) + \log(p(I'))\] (7)

To minimize the above loss means improving the ability of detecting whether a given image contains an encoded message. Simultaneously, this provides an adversarial property that improves the quality of the Alice’s output, stego image \(I'\). We also perform SGD on updating \(\theta_E\), and obtain the optimal Eve by minimizing the following loss over the same distribution:

\[
E_{I,M,sk} = \text{Loss}(\tilde{I})
\]

\[O_E(\theta_E) = \arg \min_{\theta_E} E_{I,M,sk}\] (8)

### 3.3. Asymmetric

As depicted in Figure 2, Receiver Bob in asymmetric steganography has an additional public-key generator and given a secret key \(sk\), outputs a public key \(pk\), then publishes it over the channel. Only Bob has access to the underlying secret key.

The public-key generator is itself a neural network, with its own trainable parameters \(\theta_K\). The secret key \(sk\) is fed to several fully connected layers generating a random public key \(pk\), a tensor of length \(L_{pk}\), note that \(L_{pk} = L_{sk}\). Then the public key \(pk\), cover image and secret message participate in the training process which is same as that of symmetric scheme. Loss function and optimal value of each agent in this asymmetric scheme is defined in a similar manner as those in the symmetric steganograph:

\[
E_{I,M,sk} = \text{Loss}_I(I, I') + \gamma \text{Loss}_M(M, M') + \delta \text{Loss}_p(I')
\]

\[\text{Loss}(\tilde{I}) = \arg \min_{\theta_E} E_{I,M,sk}\]

\[O_{ABK}(\theta_A, \theta_B, \theta_K) = \arg \min_{\theta_A, \theta_B, \theta_K} E_{I,M,sk}\]

\[O_E(\theta_E) = \arg \min_{\theta_E} E_{I,M,sk}\]

In asymmetric scheme, public-key generator serves for Alice and Bob and its objective is consistent with Alice and Bobs. Therefore, during the updating of parameters \(\theta_A, \theta_B\), for Alice and Bob, \(\theta_K\) is updated at the same time.

### 4. EXPERIMENT

As a proof-of-concept, we implement our steganographic training scheme for asymmetric as well as symmetric. In this section, we firstly explain the datasets and primary hyperparameters, then we display the training results.

---

1 Source code will be made available.
Both symmetric steganography and asymmetric steganography were successfully implemented in our experiment. Figure 3 shows the training loss of symmetric and asymmetric. In the first few rounds of training, the visual quality of Alice output is low in a way that allows Eve to improve its understanding as well so that the loss value of Eve is declining. Gradually, After approximately 10 epoches, Alice learns to successfully embed information in a cover image such that Eve is fooled whose loss value is increasing, and Bob learns to correctly decode the message. After 20 epoches, Bob’s loss is much low, while Eves loss remains the same. This demonstrates that Alice and Bob were trained from random initialization to appropriate performance and they have managed to effectively defeat against Eve.

The declining trend of Bob's loss value was echoed by its rising decoding success shown in Figure 4. It can be seen that the number of correctly decoded bits from either the symmetric or the asymmetric decoder Bob with bitrate of 0.06bpp, climbs faster than that from decoders with bitrate of 0.5bpp. At the end of our training epoches, both of the symmetric and the asymmetric decoders with bitrate of 0.06bpp achieved accuracy over 100%, while decoders with bitrate of 0.5bpp have been more than 99% correct.

## 5. EVALUATION

We analyze the performance of our method by measuring invisibility, the distortion made upon cover images; capacity, the size of the message we can hide; security, the degree to which encoded images can be detected by steganalyzers; and practicality, whether our scheme are practical for information hiding in real situation or not.

### Invisibility

For many scenarios, it can safely be assumed that access to the original cover image is impossible for an attacker. However, what if the cover image was discovered, even without access to the decoding network? Table 2 shows the residual image between the cover and the stego. The black pixel implies the corresponding position in the cover image and stego image are exactly the same in all RGB channels. Moreover, since the distortions are too subtle for human to observe distributions, we magnified all residual images by 8 times in Table 2. It can be found that our encoders introduce subtle distortions on cover images, and the longer message concealed in a cover image, the larger distortion occurs. It also shows the hidden message is dispersed throughout the bits in all the image, and generated stego images are quite similar with original images in appearance and can hardly be distinguished with naked eyes.

We also compare our work with Hayes and Danezi [7], which uses the fully connected layer in encoder. The alterations occur in cover images are quantitatively measured by MSE and SSIM in this paper. As displayed in Table 3 the

### 4.1. Implementation

**Datasets and Hyperparameters.** We started our experiment with “ImageNet” Dataset [12]. It is a large-scale, accurate and diverse image database built upon the hierarchical structure provided by WordNet, consisting of a total of 3.2 million cleanly annotated images. In the experiment, we make use of randomly selected pairs of images from the “ImageNet” training set. The testing was conducted on 8,000 image pairs taken from ImageNet images (not used in training).

The values of primary hyperparameters used for training are given in Table 1. Moreover, we introduce a parameter bitrate referring to the number of bits to be concealed within per pixel of containers (bpp). The experimental results and evaluation are as follows.

| Parameter | Dataset(ImageNet) |
|-----------|-------------------|
| ImageSize | $3 \times 128 \times 128$ |
| KeyLength | 1024bit |
| LearningRate | 0.001 |
| BatchSize | 64 |
| Epoches | 100 |
| Payload | 0.06bpp 0.5bpp |
| $\alpha$ | 0.5 0.5 |
| $\beta$ | 0.5 0.5 |
| $\gamma$ | 0.02 0.03 |
| $\delta$ | 0.01 0.02 |

### 4.2. Results

Both symmetric steganography and asymmetric steganography were successfully implemented in our experiment.

Figure 3 shows the training loss of symmetric and asymmetric. In the first few rounds of training, the visual quality of Alice output is low in a way that allows Eve to improve its understanding as well so that the loss value of Eve is declining. Gradually, After approximately 10 epoches, Alice learns to successfully embed information in a cover image such that Eve is fooled whose loss value is increasing, and Bob learns to correctly decode the message. After 20 epoches, Bob’s loss is much low, while Eves loss remains the same. This demonstrates that Alice and Bob were trained from random initialization to appropriate performance and they have managed to effectively defeat against Eve.

The declining trend of Bob's loss value was echoed by its rising decoding success shown in Figure 4. It can be seen that the number of correctly decoded bits from either the symmetric or the asymmetric decoder Bob with bitrate of 0.06bpp, climbs faster than that from decoders with bitrate of 0.5bpp. At the end of our training epoches, both of the symmetric and the asymmetric decoders with bitrate of 0.06bpp achieved accuracy over 100%, while decoders with bitrate of 0.5bpp have been more than 99% correct.

We analyze the performance of our method by measuring invisibility, the distortion made upon cover images; capacity, the size of the message we can hide; security, the degree to which encoded images can be detected by steganalyzers; and practicality, whether our scheme are practical for information hiding in real situation or not.

### Invisibility

For many scenarios, it can safely be assumed that access to the original cover image is impossible for an attacker. However, what if the cover image was discovered, even without access to the decoding network? Table 2 shows the residual image between the cover and the stego. The black pixel implies the corresponding position in the cover image and stego image are exactly the same in all RGB channels. Moreover, since the distortions are too subtle for human to observe distributions, we magnified all residual images by 8 times in Table 2. It can be found that our encoders introduce subtle distortions on cover images, and the longer message concealed in a cover image, the larger distortion occurs. It also shows the hidden message is dispersed throughout the bits in all the image, and generated stego images are quite similar with original images in appearance and can hardly be distinguished with naked eyes.

We also compare our work with Hayes and Danezi [7], which uses the fully connected layer in encoder. The alterations occur in cover images are quantitatively measured by MSE and SSIM in this paper. As displayed in Table 3 the
expected MSE of stego images by our training scheme is kept to a satisfactorily lower value than Hayes and Danezi [7]. Furthermore, the SSIM of that is also kept to a higher value than Hayes and Danezi [7]. On average, the results is 20% better.

The above convincingly demonstrates that the invisibility of our proposed steganographic scheme are superior to previous work, as well as visually indistinguishable from the cover image.

**Capacity and Security.** We compare our scheme against both state-of-the-art steganographic algorithms and steganalysis tools and show that it performs competitively.

Often, the amount of hidden message is set to 0.4bpp or lower. The longer the message, the larger the bpp, and therefore the more the carrier is altered [13]. The state-of-the-art steganographic algorithms, like WOW [2], HUGO [3] and S-UNIWARD [4], are commonly employed with relatively small messages (no more than 0.4bpp). So our work matches the baselines on BPP (0.4 vs 0.4). As displayed in Figure 4, both of symmetric and asymmetric decoders with bitrate of 0.5bpp achieved accuracy over 99%, so there is additionally the bitrate of 0.5bpp to compare our work with the state-of-the-art steganographic algorithms.

We compare our results against a state-of-the-art steganalyzer, ATS [14]. ATS uses labeled data to build artificial training sets of cover and stego images, and is trained using an SVM with a Gaussian kernel. Results are shown in Table 4. When we test ATS on 0.4bpp and 0.5bpp of our key-dependent steganographic scheme, it achieves a detection accuracy of around 60%, a little better than random guessing. In comparison, ATS achieves detection accuracies of 76% for HUGO [3] and more than 80% for WOW [2] and S-UNIWARD [4], even though there is only 0.4bpp of these algorithms. The detection rate of our scheme decreased by 25%, which indicates an advantage of learned steganographic scheme: unlike static classical algorithms, they can employ a different set of encoding strategies, achieving larger capacity and higher security.

**Practicality.** To validate the practicality of our idea, we simulated a real situation and implemented our asymmetric scheme with bitrate of 0.06bpp. We firstly implement our training scheme locally on machine A and then model information is transmitted to another machine B that reconstructed the learned models. The stego images embedded by the Alice

---

**Table 2. Visual effects of resultant steganographic encoders.**

| Cover bpp | Symmetric (stego / residual × 1 / residual × 8) | Asymmetric (stego / residual × 1 / residual × 8) |
|-----------|-----------------------------------------------|-----------------------------------------------|
| 0.06      | ![Symmetric 0.06](image) ![Asymmetric 0.06](image) | ![Symmetric 0.06](image) ![Asymmetric 0.06](image) |
| 0.5       | ![Symmetric 0.5](image) ![Asymmetric 0.5](image) | ![Symmetric 0.5](image) ![Asymmetric 0.5](image) |

**Table 3. Alteration in appearance measured by MSE and SSIM**

| Steganography       | Bitrate (bpp) | MSE   | SSIM  |
|---------------------|---------------|-------|-------|
| Hayes and Danezi[11]| 0.06          | 0.0894| 0.8910|
| Symmetric steganography | 0.06        | 0.0384| 0.9710|
| Asymmetric steganography | 0.5         | 0.0856| 0.9432|
| Asymmetric steganography | 0.06        | 0.0517| 0.9581|
| Asymmetric steganography | 0.5         | 0.0901| 0.9346|
Table 4. Accuracy of distinguishing between cover and steganographic images for the steganalyzers ATS

| Steganography | Bitrate (bpp) | Accuracy(%) |
|---------------|--------------|-------------|
| HUGO          | 0.4          | 76          |
| WOW           | 0.4          | 84          |
| S-UNIWARD     | 0.4          | 81          |
| Symmetric     | 0.4          | 55          |
| steganography | 0.5          | 61          |
| Asymmetric    | 0.4          | 56          |
| steganography | 0.5          | 59          |

is passed from Machine A, to Machine B, who used the Bob model to recover the secret messages. Table 5 shows Machine B was able to recover 98% of messages even though there is noise in public communication channel. The performance indicates that our steganographic decoder algorithm allows a rather effective communication between two real-world machines without obvious influence on the meaning of messages.

6. CONCLUSION

In this paper, we firstly proposed a novel key-dependent steganographic scheme with adversarial training. Compared to previous works, our key-dependent scheme improves the invisibility by 20%. In terms of capacity and security, the steganography algorithm obtained through this scheme has improved them by 25%, even if the secret message is larger. Furthermore, in the simulation of the real situation, the decoder’s accuracy rate exceeds 98%, which indicates that our scheme are effective and practical for information hiding in communication.

Furthermore, we expect this work to be expanded to other cover media such as audio and video, and expect it to be implemented on other forms of deep learning e.g. recurrent neural networks, etc.

7. REFERENCES

[1] Gustavus J. Simmons, “The prisoner’s problem and the subliminal channel,” in Advances in Cryptology. Springer, 1984, pp. 51–67.

[2] Vojtěch Holub and Jessica Fridrich, “Designing steganographic distortion using directional filters,” in 2012 IEEE International workshop on information forensics and security (WIFS). IEEE, 2012, pp. 234–239.

[3] Tomáš Pevný, Tomáš Filler, and Patrick Bas, “Using high-dimensional image models to perform highly undetectable steganography,” in International Workshop on Information Hiding. Springer, 2010, pp. 161–177.

[4] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark, “Universal distortion function for steganography in an arbitrary domain,” EURASIP Journal on Information Security, vol. 2014, no. 1, pp. 1, 2014.

[5] Jiaohua Qin, Xuyu Xiang, and Meng Xian Wang, “A review on detection of lsb matching steganography,” Information Technology Journal, vol. 9, no. 8, pp. 1725–1738, 2010.

[6] Denis Volkonskiy, Boris Borisenko, and Evgeny Burnaev, “Generative adversarial networks for image steganography,” 2016.

[7] Jamie Hayes and George Danezis, “Generating steganographic images via adversarial training,” in Advances in Neural Information Processing Systems, 2017, pp. 1954–1963.

[8] Joseph A O’Sullivan, Pierre Moulin, and J Mark Ettinger, “Information theoretic analysis of steganography,” in Information Theory, 1999. Proceedings. 1998 IEEE International Symposium on. IEEE, 1998, p. 297.

[9] Luis Von Ahn and Nicholas J Hopper, “Public-key steganography,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2004, pp. 323–341.

[10] Jarno Mielikainen, “Lsb matching revisited,” IEEE signal processing letters, vol. 13, no. 5, pp. 285–287, 2006.

[11] Ning Bi, Qiyu Sun, Daren Huang, Zhihua Yang, and Jiwu Huang, “Robust image watermarking based on multiband wavelets and empirical mode decomposition,” IEEE Transactions on Image Processing, vol. 16, no. 8, pp. 1956–1966, 2007.

[12] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaou Tang, “Deep learning face attributes in the wild,” in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3730–3738.

[13] Jessica Fridrich and Miroslav Goljan, “Practical steganalysis of digital images: state of the art,” in Security and Watermarking of Multimedia Contents IV. International Society for Optics and Photonics, 2002, vol. 4675, pp. 1–14.

[14] Daniel Lerch-Hostalot and David Megías, “Unsupervised steganalysis based on artificial training sets,” Engineering Applications of Artificial Intelligence, vol. 50, pp. 45–59, 2016.
Table 5. An example in simulated communication

| Cover Message                                                                 | Before encode | After decode | Reconstructed message |
|------------------------------------------------------------------------------|---------------|--------------|------------------------|
| Two months ago, across an assembly-room table in a factory in Jacksonville, Fla., President Barack Obama was talking to me about... | 01010100 01110111 01110100 01100001 ... | 01010100 01110111 01110100 01100 ... | Two months ago, across an assembly-room table in a factory in Jacksonville, Fla., President Barack Obama was talking to me about... |