A Robust Steganography-without-Embedding Approach Against Adversarial Attacks

Donghui Hu,[1] **Song Yan,** [1] **Wenjie Jiang,** [1] **Run Wang** ✉ [2]

[1] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China
[2] School of Cyber Science and Engineering, Wuhan University, Wuhan, China
hudh@hfut.edu.cn, songyan@mail.hfut.edu.cn, jwj743@163.com, wangrun@whu.edu.cn

Abstract

Generative adversarial networks (GAN) provide a promising idea in achieving steganography-without-embedding (SwE) where the carrier (cover) remains intact without any modification like traditional embedding based image steganography algorithms. In SwE with GAN, the noises are automatically generated and mapped to the recoverable secret information. Unfortunately, the state-of-the-art (SOTA) SwE techniques with GAN are still in their infancy and not prepared for the adversarial attack where the model could be easily stolen and the parameters could be inferred by attackers as well. In such adversarial circumstances, attackers could easily confuse or deceive the receivers intentionally, which cause serious security issues in transmitting information with steganography techniques. In this paper, we propose a GAN-based SwE by introducing asymmetric encoding and decoding key in the embedding and extracting stage, respectively, for protecting the safety of our embedded message, which totally satisfies the Kerckhoffs's principle of cryptography. Experimental results show that our proposed approach could effectively embed secret information and successfully evade the adversarial attacks. Moreover, our research findings also pose a new insight for developing secure steganography algorithms against adversarial attacks for both white-box and black-box settings.

Introduction

With the rapid development of deep learning (DL), information hiding steps into a new milestone where the automatic embedding requires less much human-efforts than ever before. In the traditional information hiding (Bamatraf, Ibrahim, and Salleh 2010; Holub and Fridrich 2012; Holub, Fridrich, and Denemark 2014; Guo, Ni, and Shi 2014), namely steganography for secret communication via multimedia and digital watermarking for protecting copyright, large domain knowledge is required for designing powerful steganography against being detected by steganalysis (You, Zhang, and Zhao 2020), or robust watermarking to defend the malicious watermark attacks (Hosam 2019; Geng et al. 2020). Studies have shown that the position for embedding could be learned and SwE could be also achieved with the advances of GANs (Yamamoto, Song, and Kim 2020; Yu and Pool 2020; Kong, Kim, and Bae 2020).

The SOTA DL-based steganography algorithms (Wang et al. 2018; Zhang, Dong, and Liu 2019; Ke et al. 2019; Tang et al. 2019; Zhang et al. 2019a; Yu 2020; Liu et al. 2020; Ghamizi et al. 2021) generally fall into two categories, *i.e.*, DL-based steganography with embedding and DL-based steganography without embedding. The first category combines DL with traditional embedding based steganography, aiming to find more suitable embedding carriers, more appropriate embedding locations, and more optimized distortion function. The second category directly uses the DL generated images as stego without any introduced, thus avoid being detected by steganalysis based on the embedding features. In generating the DL-based SwE, a sender Alice trains a generator to generate "natural" image (stego) that hides secret information, while a receiver Bob uses a trained extractor to recover the secret information from the received stego. During the training of the generator, as general GAN, a discriminator plays the role in making the generated image more "real". However, the SOTA DL-based SwE is vulnerable to the adversarial attacks (Auernhammer, Kolagari, and Zoppelt 2019; Wang et al. 2020a) in both white-box and black-box settings, which calls for effective techniques to address this issues.

In the white-box adversarial attack, we consider the following two threats, 1) the generator and its parameters are stolen by attackers; and 2) the extractor and its parameters are stolen by attackers. In the first case, the attacker could generate and transmit a confused stego by the obtained generator to the receiver. In the second case, the attackers could extract the secret message from the intercepted stego via their obtained extractor. In the black-box adversarial attack (Wang et al. 2020b), assuming that attackers have obtained large numbers of secret message and stego pairs by querying, then they can train their own extractor which has the same or similar capabilities as the original extractor. Unfortunately, both white-box and black-box cases pose a safety threat to the secret communication of GAN-based SwE.

A straightforward idea for addressing the two aforementioned attacks is using cryptography to protect the parameters of the models and then share the encrypted parameters between the sender and receiver. However, this is not practical due to the huge numbers of parameters in deep neural network and the limitations on the time-consuming and storage space as well. Furthermore, applying cryptog-

raphy directly cannot be employed in tackling the black-box schemes. Thus, new techniques need to be proposed for defending against the white-box and black-box adversarial attacks. Instead of applying cryptography straightforwardly on the model itself, a better choice is to affect the model during the training process by adding some factors on the inputs or outputs. The factors are only accessible to model users, therefore attackers cannot use the model or surrogate model without these factors. By introducing little overhead, the security of secret communications could be ensured. In the real scenario, the factors could be specific images, numbers, texts, *etc*.

In this paper, we propose a robust GAN-based SwE approach to defend against white-box and black-box adversarial attacks. In our proposed method, an encoding key is introduced in the generator, and combined with the secret message in two modes, *i.e*., concatenated mode and bitwise addition mode, then the encoding key and the secret message are used to feed the generator and produce stego. Correspondingly, a decoding key is introduced in the extractor to recover the secret message from the stego. The encoding key and the decoding key are different, that is, the proposed SwE is a asymmetric key based steganography. An encoding key strictly corresponds to a decoding key, and the generator and extractor are jointly trained, thus even if a bit of encoding or decoding key is incorrect, the secret message cannot be recovered in a correct manner. As for white-box attacks, even if the attacker obtains the structure and parameters of the generator and extractor, he/she cannot recover the secret message without a decoding key, meanwhile he/she cannot generate a confused stego without an encoding key. Then for black-box attacks, even if the attacker obtains a large number of secret message and stego pairs, he/she cannot train an available extractor that has the same behavior as original extractor due to the lacking of a decoding key.

Experimental results demonstrate that our method can protect the secret communication on the premise of ensuring the generated images' quality. By disturbing the introduced keys in different degrees, we show that our steganography model can resist both white-box and black-box attacks. All the experiments were conducted on an public dataset FFHQ.

Our main contribution are summarized as follows:

- To the best of our knowledge, we are the first to introduce the asymmetric keys for achieving DL-based SwE. Our proposed model can significantly improve the security of DL-based steganography, which satisfies the Kerckhoffs's principle of cryptography well. We hope this work could inspire future researchers to develop more robust and secure methods for advancing SwE.
- We designed a steganography model based on GAN by incorporating DL-based generator and extractor jointly trained with asymmetric keys. Our model ensures that the secret information could be recovered only when all the keys are 100% correct.
- Experimental results are conducted on a public dataset FFHQ, which demonstrated the effectiveness of our approach in fighting against the both white-box and black-box adversarial attacks.

Related Work

Steganography-without-embedding

SwE (Barni 2011) generates stego images driven by secret message or selects carriers mapping to secret message, instead of requiring a given carrier. Compared with embedding-based steganography, SwE, a new type of steganography with great development potential, is theoretically secure to resist machine learning (ML)-based steganslysis relying on embedding (pixel modification) features. According to different implementation methods, SwEs include carrier selection based approach and carrier generation based approach. The main idea of carrier selection based approach is to establish the relationship between the secret message and stego image according to the inhere characteristics of the image. Zheng et al. (2017) used robust hashing to map the stego images to secret message. By using texture synthesis, Wu and Wang (2014) generated synthesis images with complex texture to hide the secret message. Hu et al. (2018) used DCGAN (Radford, Metz, and Chintala 2015) to design the first GAN-based SwE. In this method, the stego image is generated by the generator according to the noise vector mapped from a secret message. Zhang et al. (2019b) also proposed a data-driven SwE scheme. Compared with the first DL-based SwE (Hu et al. 2018), the steganography capacity of this method has been improved, but the recovery accuracy has decreased to a certain degree. Zhang et al. (2020b) proposed SSS-GAN model that uses the category of image semantic information to generate stego. However, the SOTA DL-based SwEs have security issues in protecting the safety of models and parameters, what is we focus on.

Steganography and watermarking with deep learning

In 2017, Baluja (2017, 2019) proposed a deep steganography method that placing a full size color image within another image of the same size. Volkhonskiy, Nazarov, and Burnaev (2020) proposed a generative approach that produce realistic images that could serve as containers for secure message embedding. Tang et al. (2017) proposed ASDL-GAN, an automatic steganographic distortion learning framework with GAN, which simulates the rivalry between steganography with additive distortion and deep-learning based steganalysis. Yang et al. (2018) proposed a new secure GAN-based steganographic framework which outperformed the previous method ASDL-GAN and can resist current advanced steganalysis methods. Yang et al. (2019) proposed an enhanced GAN-based to learn the embedding cost for image steganography. In very recently, a new steganography and watermarking technique based on multi-label targeted evasion attacks, which simultaneously satisfies embedding effectiveness, elusiveness, confidentiality and robustness, was propsed by Ghamizi et al. (2021). However, the aforementioned methods are based on the embedding of natural images, which could be easily detected by the existing DL-based steganalysis techniques.

Zhu et al. (2018) developed an end-to-end neural networks with noise layer for image steganography and watermarking, which is robust against different types of noise at-

tacks. Liu et al. (2019) proposed a novel two-stage separable deep learning framework for practical blind watermarking. Zhang et al. (2020a) proposed the first deep watermarking framework for protecting deep learning based image processing models. Quan et al. (2020) protected the intellectual properties of trained DL models by modifying the host models to degrade its performance on a specific image that has a statistically significant difference from the training data. These watermarking methods modify the carriers by leveraging the neural network to find the suitable positions.

Our Method

Problem Definition and the Framework

Our proposed secure GAN-based SwE framework includes the secret message sender Alice, the receiver Bob, the discriminator Dev, and the attacker Eve. Among them, Alice and Bob are both communicating parties. Dev plays a role of the discriminator to help Alice and Bob to establish a secure communication channel (making the stego image generated by Alice more real and natural). Eve is the attacker who attempts to disturb or even steal the secret message.

A stego image c' is generated by the generator Alice with the secret message m and an encoding key en_key.

$$A(m, en\_key) = c'. \tag{1}$$

The secret massage m' is recovered by the extractor Bob with the stego image c' and a decoding key de_key.

$$B(c', de\_key) = m'. \tag{2}$$

The discriminator Dev inputs the stego image c' generated by Alice and a real image c from the real image data set, and determine whether the input image is real or fake according to its output:

$$D(c, c') = 0/1. \tag{3}$$

Eve is an attacker who has a lot of background knowledge under different attack modes. In this paper, we consider two types of white-box threats and one type of black-box threat as follows.

- **Generator Disclosure in White-box**. The attacker Eve has the network parameters of the generator, including the network structure published by the algorithm. Eve attempts to transmit message m'' to confuse the receiver and mislead the receiver to make wrong decisions. For example, the attacker uses the obtained generator to generate a misleading image c'' with the pre-designed misleading message m'' and then sends it to the receiver for misleading purpose:
$$A(m'') = c'' \tag{4}$$
- **Extractor Disclosure in White-box**. The attacker Eve obtains both the stego image c' and the network parameters of the extractor. Thereby, Eve can recover the secret message m' with the obtained extractor which may cause information disclosure:
$$B(c') = m' \tag{5}$$
where $m' = m$ if Bob can 100% recover the secret message m.
- **Surrogate Extractor in Black-box**. Supposing that the attacker Eve possesses a large number of secret message and stego image pairs (c', m). Noting that we do not entangle with the approach of attackers obtaining those pairs in real scenarios. Instead, a extreme case is considered in the black-box case so that the attacker is able to train the extraction network B' based on these. Afterwards, if Eve obtains a suspected image, the secret message $\widetilde{m}$ can be extracted via the network, which may result in information disclosure.
$$B'(c') = \widetilde{m} \tag{6}$$

Generator

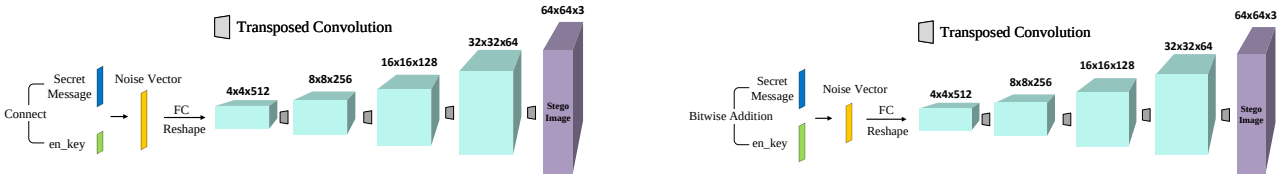
The generator Alice generates a stego image based on the input secret message. Recall the aforementioned first threat in white-box, it is necessary to design a scheme that even if the attacker knows the network parameters of the generator and generates a forged stego image based on the obtained generator and misleading message, the receiver cannot extract misleading message from the forged stego image.

Generally, the generation of stego image only needs secret message and a trained generator. If the input of the network contains not only the secret messge m, but also an encoding key en_key, then the parameters of trained generation network will also be determined by the encoding key. A stego image can be generated when the key is known merely. The sender generates the stego image c' with Eq. (1). The attacker does not know the correct encoding key, so he/she chooses a key x (x represents the randomized key), along with a forged secret message m'', feeds to the obtained generator and obtains a misleading image c'' by $A(m'', x) = c''$. Apparently, $c'' \neq c'$ due to an incorrect encoding key. Therefore, recipient could not extract the misleading secret message and the first threat in white-box can be solved.

According to different ways of combining encoding key and secret message, we design two specific steganography (generator) structures as shown in Fig. 1.

Concatenated Mode In this mode, the encoding key en_key is directly concatenated with the secret message m as the input noise vector n of generator, *i.e.*, $n = en\_key|m$, to incorporate in the network training. The illustration is shown in Fig. 1(a). Following the concatenated input n, there is a fully connected layer and several deconvolution layers, which learn and generate samples that conform the real image distribution. In training process, the value of secret message here is a random number sampled from $(-1, 1)$ which determines the characteristics of the generated image. And the dimension of secret message determines the scale of network parameters of generator and the complexity of generated image. The value of encoding key en_key is also a random number sampled from $(-1, 1)$. The concatenating operation increases the total dimension of the input vector in a disguised way. Note that the generated key is fixed, therefore the noise vector dynamically changes only with the message.

Bitwise Addition Mode In this mode, the encoding key en_key and secret message m are combined by bitwisely addition, *i.e.*, $n = en\_key \oplus m$, as the feed of the generator, which is shown in Fig. 1(b). In this structure, en_key shares



(a) (b)

Figure 1: The steganography (generator) network structure of (a) concatenated mode and (b) bitwise addition mode.

the same dimension with m. If the the dimension of en_key is different with that of the secret message, a liner operation is required to expand the dimension of the encoding key or the secret message. Same as before, m and en_key are random values sampled from $(-1, 1)$.

Extractor

The extractor plays the role in extracting the hidden secret message from stego image with decoding key. To well address the the second threat in white-box, we introduced a decoding key in the extractor side and jointly trained with the generator (fed with a corresponding encoding key), thus an attacker who does not have the correct decoding key cannot recover the secret message even if he/she has the knowledge of extractor.

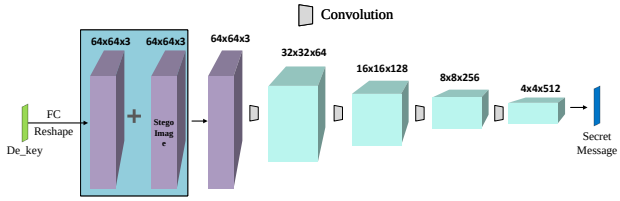


Figure 2: The structure of secret message extraction network.

Fig. 2 shows the network structure of our designed extractor. Decoding key de_key is transformed into a matrix which shares the same size and dimension with input stego image. Then this matrix bitwise adds with input image, and the result is the input of extractor for secret message extraction. Under this circumstance, the introduction of the decoding key affects the parameters of extraction network, so any change of the key will affect the final extracted message.

We add the part of secret key recovery in the loss function of extractor as:

$$\begin{aligned} L_B &= \lambda_1 d\left(m, m'\right) + \lambda_2 d\left(de\_key, m'\right) \\ &= \lambda_1 \sum_{i=1}^{N} \left(m, m'\right)^2 + \lambda_2 \sum_{i=1}^{N} \left(de\_key, m'\right)^2, \end{aligned} \tag{7}$$

where N is the dimension of input noise vector, m is the secret message hidden in stego image, m' is the message extracted by receiver and de_key is the decoding key. λ_1 and λ_2 represent the recovery weights of secret message and decoding key, respectively.

Training Process

The training process of the framework designed in this paper adopts a combination of end-to-end and training by stage. In image steganography algorithms, the generation of stego image should be guaranteed first. Therefore, the generator and the discriminator should be pre-trained so that the generator can stably generate stego image. Then in mid-training phase, the extractor participates in training for recovering secret message from stego images. We dynamically adjust the weights of networks of Alice, Bob and Dev in different processes of pre-training, mid-training and end-training.

In detail, in the mid-training process, when the generator and the discriminator iterate once, the extractor iterates 3 times, which helps normalize the convergence of generator to match the direction of secret message recovery while ensuring that the generator continues learning to generate images. After a certain number of iterations, only the extractor will be updated, and the generator only needs to generate stego images as the input of extractor, we call process of this phase as end-training. The aim of end-training phase is to further improve the accuracy of secret message recovery.

Experiments

In this section, we evaluate the proposed GAN-based SwE from the following aspects. Firstly, we evaluate the security of the proposed method in defending the white-box and black-box attacks.

Datasets

FFHQ[1](Karras, Laine, and Aila 2019) is a large human facial data set which consists of 70,000 high-quality PNG images at 1024×1024 resolution and contains considerable variation in terms of age, ethnicity and image background. It also has good coverage of accessories such as eyeglasses, sunglasses, hats, *etc*. In the following experiments, we choose the thumbnails in 128×128 resolution.

Implementation Details

In our experiments, we use Tensorflow for the implementation and test them on a NVIDIA 2080Ti GPU. We train the model for 100 epochs with stochastic gradient descent and the batch size is set to 64. The size of generated images is 64×64×3. The inputted noise vector is a random value sampled from $(-1, 1)$ and the dimension of the secret message are 100, 200 and 300 respectively. The dimensions of

[1]https://github.com/NVlabs/ffhq-dataset

encoding key and decoding key are both set to 10 in our experiments. As for the hyperparameters, learning rate is set to 0.0002 and one step is recorded every 1,000 batches.

Quality of Generated Image

Different from the original GAN, the input of our generator includes not only secret message, but also the encoding key. Observed images generated by the generator designed in this paper and the original generator (without keys) are shown in Fig. 3. We can see that the introduced encoding key does not affect the quality of the generated samples. Similar to original GAN's generator, our generator is able to learn to generate samples that obey the distribution of real dataset.

Furthermore, we evaluate the generated images with more rigorous quantitative indicators from the perspective of image quality and sample diversity, and measure the difference between the distribution of real images and stego images. FID (Frechet Inception Distance) (Dowson and Landau 1982) is a metric that is widely used to evaluate the quality of images. As for GAN, FID can measure the difference between the distribution of real and generated images.

$$FID(r,g) = ||\mu_r - \mu_g||^2 + Tr(\sum_r + \sum_g - 2(\sum_r \sum_g)^{\frac{1}{2}})), \tag{8}$$

where r and g are the distribution of real images and generated images, respectively. By calculating the mean and covariance of the two distributions, the similarity of the two sets of images is measured. The lower the FID score, the closer the generated images are to the real images. The results are shown in Table 1, from which we can see that the quality of generated images is not affected by the encoding key we introduced in the generator, and the fluctuation of FID is lower than 1, which demonstrates that the quality of generated images are almost the same as the original GAN's.

Style	Original GAN	Concatenated	Bitwise addition
FID	30.81	30.97	30.09

Table 1: FID scores of different generation structures.

Security Analysis

Recalling three potential threats we discussed above: generator disclosure (the first threat in white-box), extractor disclosure (the second threat in white-box) and train extractor (the threat of black-box), in this subsection, we conduct experiments to evaluate the security of our method in defending against these threats.

Generator Disclosure To estimate the impact of encoding key on the generator disclosure threat, we conduct experiments on generator with two modes of using encoding key and the results are shown in Table 2. The dimension of secret message is set to 100, 200 and 300. We compare message recovery accuracy of original GAN, our network with correct and randomized encoding key (representing the key randomly guessed by an attacker). As we can observe from Table 2, the recovery accuracy of our network with encoding key in concatenated mode is almost the same as the original GAN without keys. In addition, there is a slight improvement in the circumstance of secret message in 300 dimension when the given encoding key is correct. But if the key is wrong or randomized, the message recovery accuracy is only approximately 50%. For 0/1 bits, it is almost the result of randomization, which exactly demonstrates the security of our method in defending white-box attack at the generator's side. As for generator using encoding key with bitwise addition mode, the recovery accuracy is lower than 60%, which demonstrates that the security of generator in bitwise addition mode is slightly weak than that in concatenating mode in defending the white-box attack.

Extractor Disclosure For extractor disclosure threat, we introduce decoding key in the input (as well as considering the element of the decoding key in the loss function of extractor). We compare message recovery accuracy of original GAN, our network with correct decoding key and randomized decoding key (representing that used by an attacker). As shown in Table 3, the hidden message recovery accuracy of extractor without a decoding key is 65.3%, 59.7%, and 59.2%, respectively, and a relative high accuracy (above 94%) is guaranteed when the key is correct. It demonstrates that the extractor with a decoding key can well address the known extractor disclosure threat.

Surrogate Extractor To simulate the process of an attacker using collected stego image and secret message pairs to train the surrogate extractor, we use a trained generator with encoding key to generate secret message and stego image pairs. We assume that the attacker hold an extraction network without decoding key in the input (the input only includes stego image), which is slightly different from our designed model (shown in Fig. 2). The loss function of attacker is

$$L_{Eve} = d(m, E(c')) = \sum_{i=1}^{N'} (m, E(c'))^2, \tag{9}$$

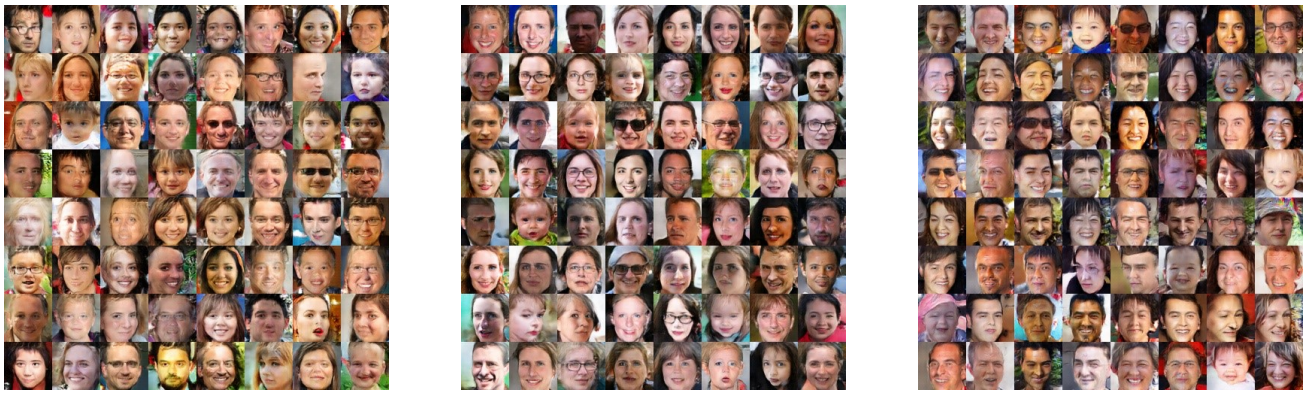
where m is secret message, c' is stego image, N' is the dimension of input noise vector, and E is the extractor trained by attacker Eve.

Due to the attacker's joining, the loss function of original extractor also need to be adjusted as follows:

$$L_{Bob} = \lambda_B L_B - \lambda_E L_{Eve}, \tag{10}$$

where λ_B and λ_E are the extractor weights of receiver Bob and attacker Eve, respectively, which guarantee that Bob's extractor converges while maximizing Eve's.

The experimental results are shown in Fig. 4. In the left side of dotted line, the extractor of receiver and the surrogate extractor of attacker conduct a confrontation training process. At this time, the message recovery accuracy is gradually approaching to 50%. Then in the right side, after training approximately 20 iterations, the training of receiver tends to be converged. As we can observe, the surrogate extractor of attacker gradually converges in the following several steps and keeps stable at approximately 0.85, 0.76 and 0.57. Therefore, even if the attacker hold a large number of stego image and secret message pairs, he/she still cannot train an effective surrogate extractor to recover secret



(a) Original GAN (b) Concatenated (c) Bitwise Addition

Figure 3: Images generated by generator with different structures: (a) Images generated by original GAN without keys, (b) Images generated by generator with encoding key in concatenated mode, and (c) Images generated by generator with encoding key in bitwise addition mode.

(a)

Dim \ En_key	None	Correct	Randomized
100	99.9%	99.9%	50.1%
200	99.4%	99.7%	50.2%
300	97.5%	98.7%	49.9%

(b)

Dim \ En_key	None	Correct	Randomized
100	99.9%	99.9%	59.8%
200	99.4%	99.3%	53.4%
300	97.5%	97.2%	52.2%

Table 2: Message recovery accuracy of original GAN without key, generator with correct encoding key and generator with randomized encoding key. (a) The encoding key used in concatenated mode. (b) The encoding key used in bitwise addition mode.

Dim \ De_key	None	Correct	Randomized
100	99.9%	99.6%	65.3%
200	99.4%	97.6%	59.7%
300	97.5%	94.8%	59.2%

Table 3: Message recovery accuracy of original GAN, extractor with correct and randomized decoding key.

message. We can also find from the figure that the larger the dimension of the input, the more secure of the performance in defending the black-box attack can be achieved.

Even More Background Knowledge?

In the last subsection, we have discussed how our designed method solves the known threats when the attacker does not even known anything about encoding and decoding key. But what if a part of the keys is intercepted or inferred by the attacker? Thus we conduct further experiments to evaluate the security of our method by observing the secret message recovery accuracy with different error bits of given encoding and decoding key. Both the encoding and decoding key use a uniform length of 10 bits.

As shown in Fig. 5, for the case of encoding key in concatenated mode, when the key is incorrect by only one bit,

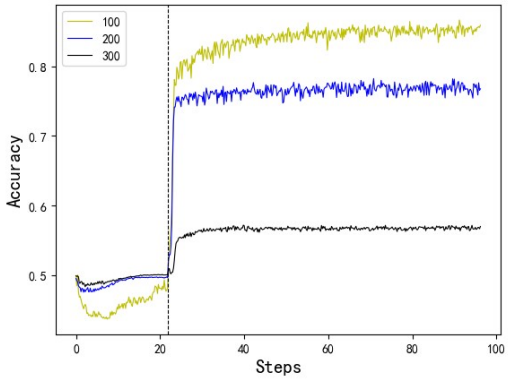
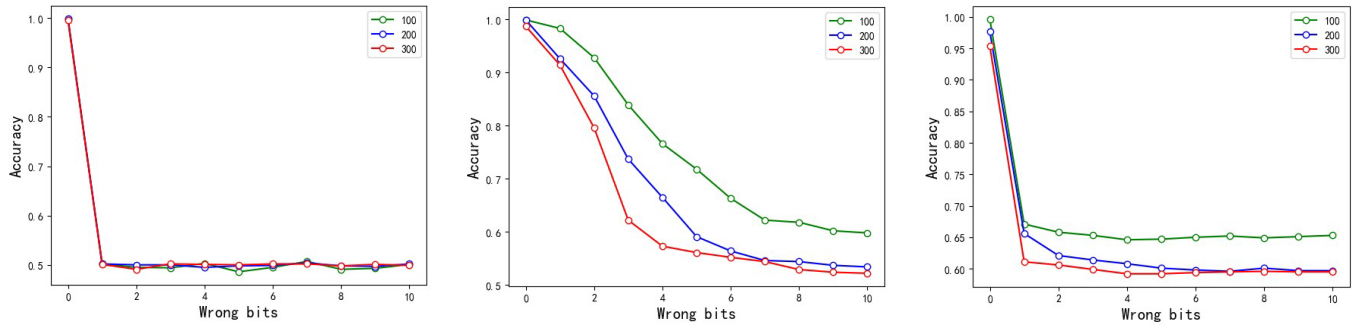


Figure 4: Message recovery accuracy of extractor training by attacker.

the extractor cannot extract the secret message successfully (the accuracy is approximately 50%). As the number of error bits increases, the randomness is still maintained. In the bitwise addition mode, when the encoding key is incorrect by one bit, the extractor maintains a high recovery accuracy (higher than 90%), and then as the number of error bits increases, the extraction accuracy constantly decline, and is



(a) (b) (c)

Figure 5: Secret message recovery accuracy with different error bits of given encoding or decoding keys: (a) encoding key in concatenated mode, (b) encoding key in bitwise addition mode, and (c) decoding key.

Steganography Methods	Steganography Capacity	Image Size	Recovery Accuracy
Abadi and Andersen (2016)	16	-	99.7%
Li and Zhang (2018)	152/334	300×300/500×500	100%
Zhang et al. (2019b)	146∼1083	64×64	70%
Zheng et al. (2019)	128×128	256×256	-
Ours	300	64×64	>94%

Table 4: In comparing with other steganography methods based on encryption

finally lower than 60%. Similar with the circumstance of encoding key, the recovery accuracy of the extractor drops to approximately 65% when the decoding key has one bit error, and it gradually decreases as the number of error bits increases. The recovery accuracy of secret message in 100 dimension keeps stable at approximately 65%, and the circumstance of secret message in 200 and 300 dimension keep stable at approximately 60%. Experimental results demonstrate that the attacker could not recover secret message accurately either decoding or encoding key was wrong, which further prove the security of our proposed method.

Steganography Performance Comparsion

In this section, we compare the designed secure SwE scheme based on GAN with other steganography methods based on encryption. Two indicators are used to evaluate steganography performance: steganography capacity and message recovery accuracy.

As shown in Table. 4, the steganography capacity of our method in a single image can reach 300 bits (300-dimensional). (Abadi and Andersen 2016) is the first work that encryption is introduced into the security model of deep learning. It generates a messy ciphertext instead of an image, and the steganography capacity is low while the recovery accuracy is relatively high. (Li and Zhang 2018) is a SwE method based on fingerprint construction. Due to the introduction of the RS error correction code, the recovery accuracy reaches 100%. (Zhang et al. 2019b) is an encryption embedded-based steganography method which has a great improvemet in steganography capacity compared with previous work (Hu et al. 2018). But the recovery accuracy has dropped severely, only reaching 70%. (Zheng et al. 2019) embed image in image based on GAN. Due to the large number of image pixels and huge steganography capacity, the recovery of information is not at pixel but the 0,1 bit level which relies more on vision. In comparison, the security model designed in this paper performs well in terms of steganography. The steganography capacity could reaches 300 bits while ensuring high information recovery accuracy.

Conclusion

We have proposed a secure SwE based on GAN. Starting from designing principles of modern cryptography, asymmetric encoding and decoding keys are introduced. Considering the security threats of generator disclosure, extractor disclosure, and the surrogate of extractor, the encoding key with two specific using modes is added to the generator, and the decoding key is added to the extractor. We have conducted experiments on a benchmark dataset to evaluate the performance of our proposed method. The results show that the introduced keys do not reduce the quality of generated stego images, and the security of defending white-box and block-box attacks is greatly enhanced. Even if one bit of the encoding key or decoding key is incorrect, the recovery accuracy is very close to the result of random guess.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under grant No. U1836102, the National Key R&D Program of China (Grant No. 2021YFB3100700), the fellowship of China National Postdoctoral Program for Innovative Talents (Grant No. BX2021229), the Fundamental Research Funds for the Central Universities (Grant No. 2042021kf1030), and the Natural Science Foundation of Hubei Province (Grant No. 2021CFB089).

References

Abadi, M.; and Andersen, D. G. 2016. Learning to protect communications with adversarial neural cryptography. *arXiv preprint arXiv:1610.06918*.

Auernhammer, K.; Kolagari, R. T.; and Zoppelt, M. 2019. Attacks on machine learning: Lurking danger for accountability. In *SafeAI@ AAAI*.

Baluja, S. 2017. Hiding images in plain sight: Deep steganography. In *Advances in Neural Information Processing Systems*, 2069–2079.

Baluja, S. 2019. Hiding images within images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Bamatraf, A.; Ibrahim, R.; and Salleh, M. N. B. M. 2010. Digital watermarking algorithm using LSB. In *2010 International Conference on Computer Applications and Industrial Electronics*, 155–159.

Barni, M. 2011. Steganography in Digital Media: Principles, Algorithms, and Applications (Fridrich, J. 2010)[Book Reviews]. *IEEE Signal Processing Magazine*, 28(5): 142–144.

Dowson, D.; and Landau, B. 1982. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3): 450–455.

Geng, L.; Zhang, W.; Chen, H.; Fang, H.; and Yu, N. 2020. Real-time attacks on robust watermarking tools in the wild by cnn. *Journal of Real-Time Image Processing*, 1–11.

Ghamizi, S.; Cordy, M.; Papadakis, M.; and Le Traon, Y. 2021. Evasion Attack STeganography: Turning Vulnerability Of Machine Learning To Adversarial Attacks Into A Real-world Application. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 31–40.

Guo, L.; Ni, J.; and Shi, Y. Q. 2014. Uniform embedding for efficient JPEG steganography. *IEEE transactions on Information Forensics and Security*, 9(5): 814–825.

Holub, V.; and Fridrich, J. 2012. Designing steganographic distortion using directional filters. In *2012 IEEE International workshop on information forensics and security (WIFS)*, 234–239. IEEE.

Holub, V.; Fridrich, J.; and Denemark, T. 2014. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1): 1.

Hosam, O. 2019. Attacking image watermarking and steganography-a survey. *International Journal of Information Technology and Computer Science*, 11(3): 23–37.

Hu, D.; Wang, L.; Jiang, W.; Zheng, S.; and Li, B. 2018. A novel image steganography method via deep convolutional generative adversarial networks. *IEEE Access*, 6: 38303–38314.

Karras, T.; Laine, S.; and Aila, T. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ke, Y.; Zhang, M.-q.; Liu, J.; Su, T.-t.; and Yang, X.-y. 2019. Generative steganography with Kerckhoffs' principle. *Multimedia Tools and Applications*, 78(10): 13805–13818.

Kong, J.; Kim, J.; and Bae, J. 2020. HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 17022–17033. Curran Associates, Inc.

Li, S.; and Zhang, X. 2018. Toward construction-based data hiding: From secrets to fingerprint images. *IEEE Transactions on Image Processing*, 28(3): 1482–1497.

Liu, J.; Ke, Y.; Zhang, Z.; Lei, Y.; Li, J.; Zhang, M.; and Yang, X. 2020. Recent advances of image steganography with generative adversarial networks. *IEEE Access*, 8: 60575–60597.

Liu, Y.; Guo, M.; Zhang, J.; Zhu, Y.; and Xie, X. 2019. A novel two-stage separable deep learning framework for practical blind watermarking. In *Proceedings of the 27th ACM International Conference on Multimedia*, 1509–1517.

Quan, Y.; Teng, H.; Chen, Y.; and Ji, H. 2020. Watermarking Deep Neural Networks in Image Processing. *IEEE Transactions on Neural Networks and Learning Systems*.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Tang, W.; Li, B.; Tan, S.; Barni, M.; and Huang, J. 2019. CNN-based adversarial embedding for image steganography. *IEEE Transactions on Information Forensics and Security*, 14(8): 2074–2087.

Tang, W.; Tan, S.; Li, B.; and Huang, J. 2017. Automatic steganographic distortion learning using a generative adversarial network. *IEEE Signal Processing Letters*, 24(10): 1547–1551.

Volkhonskiy, D.; Nazarov, I.; and Burnaev, E. 2020. Steganographic generative adversarial networks. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, 114333M. International Society for Optics and Photonics.

Wang, L.; Yang, K.; Wang, W.; Wang, R.; and Ye, A. 2020a. MGAAttack: Toward More Query-efficient Black-box Attack by Microbial Genetic Algorithm. In *Proceedings of the 28th ACM International Conference on Multimedia*, 2229–2236.

Wang, R.; Juefei-Xu, F.; Guo, Q.; Huang, Y.; Xie, X.; Ma, L.; and Liu, Y. 2020b. Amora: Black-box adversarial morphing attack. In *Proceedings of the 28th ACM International Conference on Multimedia*, 1376–1385.

Wang, Z.; Gao, N.; Wang, X.; Qu, X.; and Li, L. 2018. SSteGAN: self-learning steganography based on generative adversarial networks. In *International Conference on Neural Information Processing*, 253–264. Springer.

Wu, K.-C.; and Wang, C.-M. 2014. Steganography using reversible texture synthesis. *IEEE Transactions on Image Processing*, 24(1): 130–139.

Yamamoto, R.; Song, E.; and Kim, J.-M. 2020. Parallel WaveGAN: A fast waveform generation model based on

generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6199–6203. IEEE.

Yang, J.; Liu, K.; Kang, X.; Wong, E. K.; and Shi, Y.-Q. 2018. Spatial image steganography based on generative adversarial network. *arXiv preprint arXiv:1804.07939*.

Yang, J.; Ruan, D.; Huang, J.; Kang, X.; and Shi, Y.-Q. 2019. An embedding cost learning framework using GAN. *IEEE Transactions on Information Forensics and Security*, 15: 839–851.

You, W.; Zhang, H.; and Zhao, X. 2020. A Siamese CNN for Image Steganalysis. *IEEE Transactions on Information Forensics and Security*, 16: 291–306.

Yu, C. 2020. Attention based data hiding with generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 1120–1128.

Yu, C.; and Pool, J. 2020. Self-Supervised Generative Adversarial Compression. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M. F.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 8235–8246. Curran Associates, Inc.

Zhang, J.; Chen, D.; Liao, J.; Fang, H.; Zhang, W.; Zhou, W.; Cui, H.; and Yu, N. 2020a. Model Watermarking for Image Processing Networks. In *AAAI*, 12805–12812.

Zhang, K. A.; Cuesta-Infante, A.; Xu, L.; and Veeramachaneni, K. 2019a. SteganoGAN: High capacity image steganography with GANs. *arXiv preprint arXiv:1901.03892*.

Zhang, R.; Dong, S.; and Liu, J. 2019. Invisible steganography via generative adversarial networks. *Multimedia tools and applications*, 78(7): 8559–8575.

Zhang, Z.; Fu, G.; Ni, R.; Liu, J.; and Yang, X. 2020b. A generative method for steganography by cover synthesis with auxiliary semantics. *Tsinghua Science and Technology*, 25(4): 516–527.

Zhang, Z.; Liu, J.; Ke, Y.; Lei, Y.; Li, J.; Zhang, M.; and Yang, X. 2019b. Generative steganography by sampling. *IEEE Access*, 7: 118586–118597.

Zheng, S.; Wang, L.; Ling, B.; and Hu, D. 2017. Coverless information hiding based on robust image hashing. In *International Conference on Intelligent Computing*, 536–547. Springer.

Zheng, Z.; Liu, H.; Yu, Z.; Zheng, H.; Wu, Y.; Yang, Y.; and Shi, J. 2019. EncryptGAN: Image Steganography with Domain Transform. *arXiv preprint arXiv:1905.11582*.

Zhu, J.; Kaplan, R.; Johnson, J.; and Fei-Fei, L. 2018. Hidden: Hiding data with deep networks. In *Proceedings of the European conference on computer vision (ECCV)*, 657–672.