electronics

MDPI

Article

Deep Multi-Image Steganography with Private Keys

Hyeokjoon Kweon [1,†], Jinsun Park [2,†], Sanghyun Woo [2] and Donghyeon Cho [1,*]

[1] Department of Electronics Engineering, Chungnam National University, Daejeon 34134, Korea; jjun@o.cnu.ac.kr
[2] School of Electrical Engineering, KAIST, Daejeon 34141, Korea; zzangjinsun@kaist.ac.kr (J.P.); shwoo93@kaist.ac.kr (S.W.)
* Correspondence: cdh12242@cnu.ac.kr; Tel.: +82-42-821-5667
† These authors contributed equally to this work.

Abstract: In this paper, we propose deep multi-image steganography with private keys. Recently, several deep CNN-based algorithms have been proposed to hide multiple secret images in a single cover image. However, conventional methods are prone to the leakage of secret information because they do not provide access to an individual secret image and often decrypt the entire hidden information all at once. To tackle the problem, we introduce the concept of private keys for secret images. Our method conceals multiple secret images in a single cover image and generates a visually similar container image containing encrypted secret information inside. In addition, private keys corresponding to each secret image are generated simultaneously. Each private key provides access to only a single secret image while keeping the other hidden images and private keys unrevealed. In specific, our model consists of deep hiding and revealing networks. The hiding network takes a cover image and secret images as inputs and extracts high-level features of the cover image and generates private keys. After that, the extracted features and private keys are concatenated and used to generate a container image. On the other hand, the revealing network extracts high-level features of the container image and decrypts a secret image using the extracted feature and a corresponding private key. Experimental results demonstrate that the proposed algorithm effectively hides and reveals multiple secret images while achieving high security.

Keywords: steganography; steganalysis; private keys

Citation: Kweon, H.; Park, J.; Woo, S.; Cho, D. Deep Multi-Image Steganography with Private Keys. *Electronics* **2021**, *10*, 1906. https://doi.org/10.3390/electronics10161906

Academic Editor: Mazdak Zamani

Received: 11 July 2021
Accepted: 3 August 2021
Published: 9 August 2021

1. Introduction

Steganography is an algorithm to conceal information within an object while keeping the object containing the hidden information indistinguishable from the original one. The main purpose of steganography is to grant access to the hidden information only to the authorized clients while keeping its content and its presence unrevealed to the others. Various kinds of carriers such as physical objects, texts, sounds, and network packets have been utilized to safely conceal and deliver confidential data. Among them, a digital image is one of the widely used carriers in recent digital steganographic algorithms (i.e., image steganography).

Conventional image steganography methods usually aim at hiding secret messages within a cover image. To this end, various studies including spatial domain-based methods [1,2] and frequency domain-based methods [3–7] have been actively conducted, and remarkable results have been achieved. Although there has been tremendous progress in image steganography, there is still a limitation in hiding a large amount of data. Recently, several studies have tried to hide full-size secret images inside a cover image using deep CNN [8–10]. These methods are completely different from the conventional image steganography approaches. The deep learning-based steganography method usually consists of a hiding network and a revealing network. The hiding network takes a cover image and a secret image as inputs then creates a container image by hiding the secret

image into the cover image. The revealing network extracts a hidden secret image from the container image.

With deep learning-based works, it was demonstrated that full-size images can be concealed in a single cover image with minimal quality loss. However, most studies are designed to hide one secret image in the cover image. Hiding [10] extended the image steganography model to hide multiple image simultaneously as shown in Figure 1a. Unfortunately, it contains only a single reveal network, which may cause a critical security problem. In specific, the revealing network does not consider the extraction of a subset of secret images, and it just extracts the entire hidden images together at the decoding stage. Therefore, it is not possible to assign specific access permission for a subset of hidden images depending on the security level. One of the simple workarounds is preparing multiple reveal networks for each secret image as shown in Figure 1b, however, it leads to increased memory usage and can expose the number of hidden images that must be kept unrevealed.

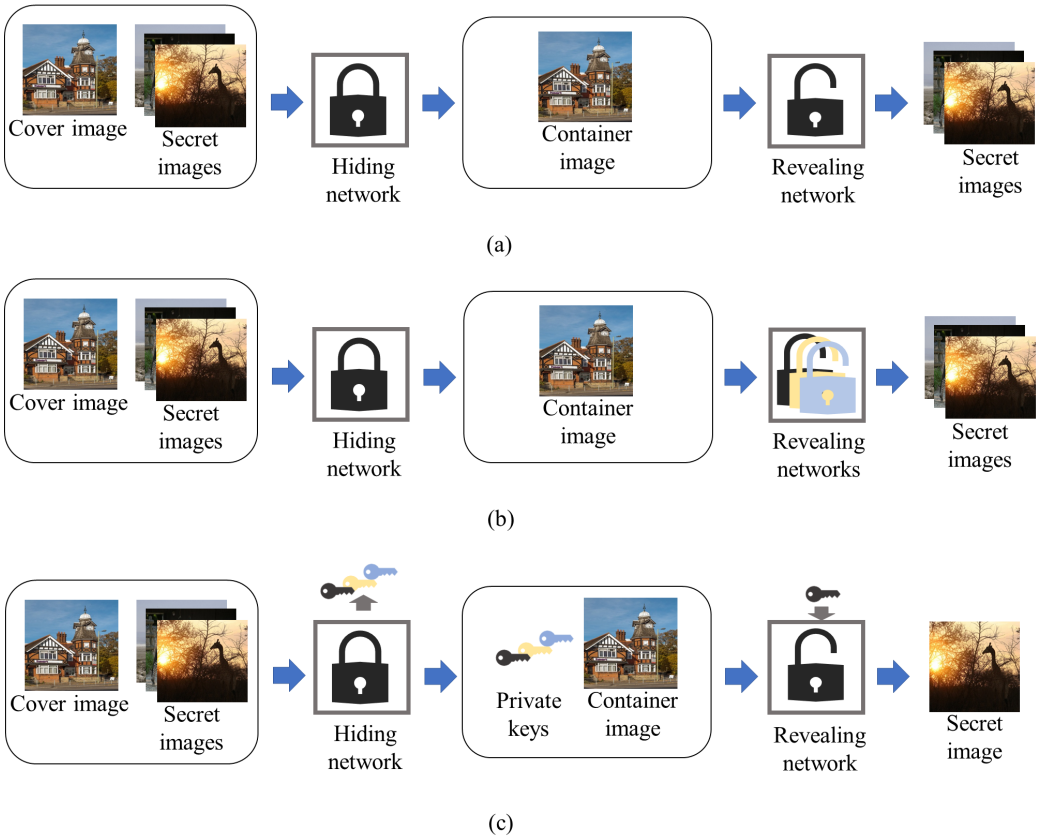

Figure 1. Comparison of image steganography models. (**a**) The conventional model does not provide individual access to each secret image. (**b**) The model has multiple revealing networks to extract hidden images, separately. (**c**) Our model supports individual access to secret images with private keys.

In [11], secret messages are hidden in a cover image by a private key, and those secret messages can be only accessed by using the correct private key. In this study, inspired by [11], we introduce the concept of the private key to multi-image steganography. As shown in Figure 1c, our steganography model consists of a hiding network and a reveal network. In the hiding step, the hiding network conceals multiple secret images into a cover image and produces private keys for each secret image. Each secret key is passed along with the container image to an account with access rights. In the revealing step, the target secret image can be obtained by feeding the container image and a private key to the revealing network. As a result, the proposed steganography model can achieve a high level

of security and efficiency while successfully hiding multiple secret images in a single cover image. Experimental results demonstrate that our method is very effective and versatile compared to previous image steganography models. To the best of our knowledge, this paper is the first attempt to utilize the concept of the private key in the field of image steganography that conveys multiple images, not messages.

Our main contributions can be summarized as follows: (i) We introduce a concept of the private key for the multi-image steganography problem. (ii) Unlike previous image steganography methods, our model can provide access right to the secert image for only authorized people. (iii) In addition, our model makes it possible to extract only quaried secret image without revealing other hidden images.

2. Related Works

In this section, we review conventional steganography methods based on both spatial domain and frequency domain, and recent deep learning-based methods.

As a pioneering work, Mielikainen [1] proposed an LSB (Least Significant Bit)-based method that adjusts the value of the least significant bit in the spatial domain of cover image to hide secret messages. Wu and Tsai [2] designed a PVD (Pixel Value Differencing)-based method that inserts secret data according to the difference in pixel values. However, these methods are vulnerable to well-designed steganalysis techniques [12–14]. Therefore, there have been various efforts including HUGO (Highly Undetectable steGO) [3], UNIWARD (UNIversal WAvelet Relative Distortion) [4], and WOW (Wavelet Obtained Weights) [5] to utilize the LSB of high-frequency components that are difficult to detect. Moreover, Chen and Lin [6] proposed a method that manipulates coefficients of the DWT (Discrete Wavelet Transform) and Kaur et al. [7] suggested a steganographic approach that embeds data in the mid-frequency band of the DCT blocks.

With recent progress of deep learning, there have been significant research achievements on various computer vision fields. Although it is not much compared to other fields, recent attempts to utilize deep learning technologies for steganography problems have increased. Baluja [8] introduced a deep steganography model that produces a container image by hiding a full-size image into a cover image. The hidden image can be extracted from the container image with a reveal network. Duan et al. [9] also proposed a new image steganography scheme based on a U-Net structure. Furthermore, Baluja [10] has shown that a deep CNN-based steganography model can hide multiple images into a single cover image as shown in Figure 1a. However, this method extracts all the hidden images at once. In other words, it cannot be utilized in a situation that we want to extract only one hidden query image and conceal the rest hidden images. To overcome above limiation, we propose a concept of private keys in the multi-image steganograpy task. With our method, it is possible to access only one hidden image without touching and revealing the other images.

3. Methods

In this section, we detail our steganography model consisting of a hiding network and a reveal network. We then describe the training processes of the proposed model. The overall pipeline of the proposed steganography model is illustrated in Figure 2.

3.1. Hiding Network

The hiding network takes a cover image $C \in \mathbb{R}^{256\times256\times3}$ and N secret images $\mathcal{S} = \{S_n\}_{n=1}^{N} \in \mathbb{R}^{256\times256\times3}$ as inputs, then produces a container image $C' \in \mathbb{R}^{256\times256\times3}$ and private keys $\mathcal{K} = \{K_n\}_{n=1}^{N}$ for each secret image S_n as follows:

$$\{C', \mathcal{K}\} = f\Big(C, \mathcal{S}; \phi_f\Big), \tag{1}$$

where ϕ_f denotes the trainable parameters of the hiding network $f(\cdot)$. The container image C' is visually indistinguishable to the input cover image C and holds the hidden information of the secret images $\mathcal{S}$. While not being sensitive to the architectural designs,

we adopt the U-Net structure [15] for the hiding network. We note that a single U-Net based hiding network is shared across the input images (i.e., cover image and multiple secret images). The encoder consists of a series of strided seven 4×4 convolution layers [16], LeakyReLU [17], and batch normalization [18] while the decoder is composed of a series of seven 4×4 deconvolution layers [19], ReLU [20], and batch normalization. Also, there are ReLU and Sigmoid layers respectively instead of LeakyReLU and ReLU layers at the last of the encoder and decoder. The visual features of each image are extracted by forwarding the cover image C and secret images $\mathcal{S}$ respectively. Then all extracted visual features from the encoder are concatenated along the channel dimension and passed into the decoder to produce a container image C'. Note that visual features of the secret image S_n from the encoder are used as its private key K_n. Architecture details of the hiding network are presented in Table 1.

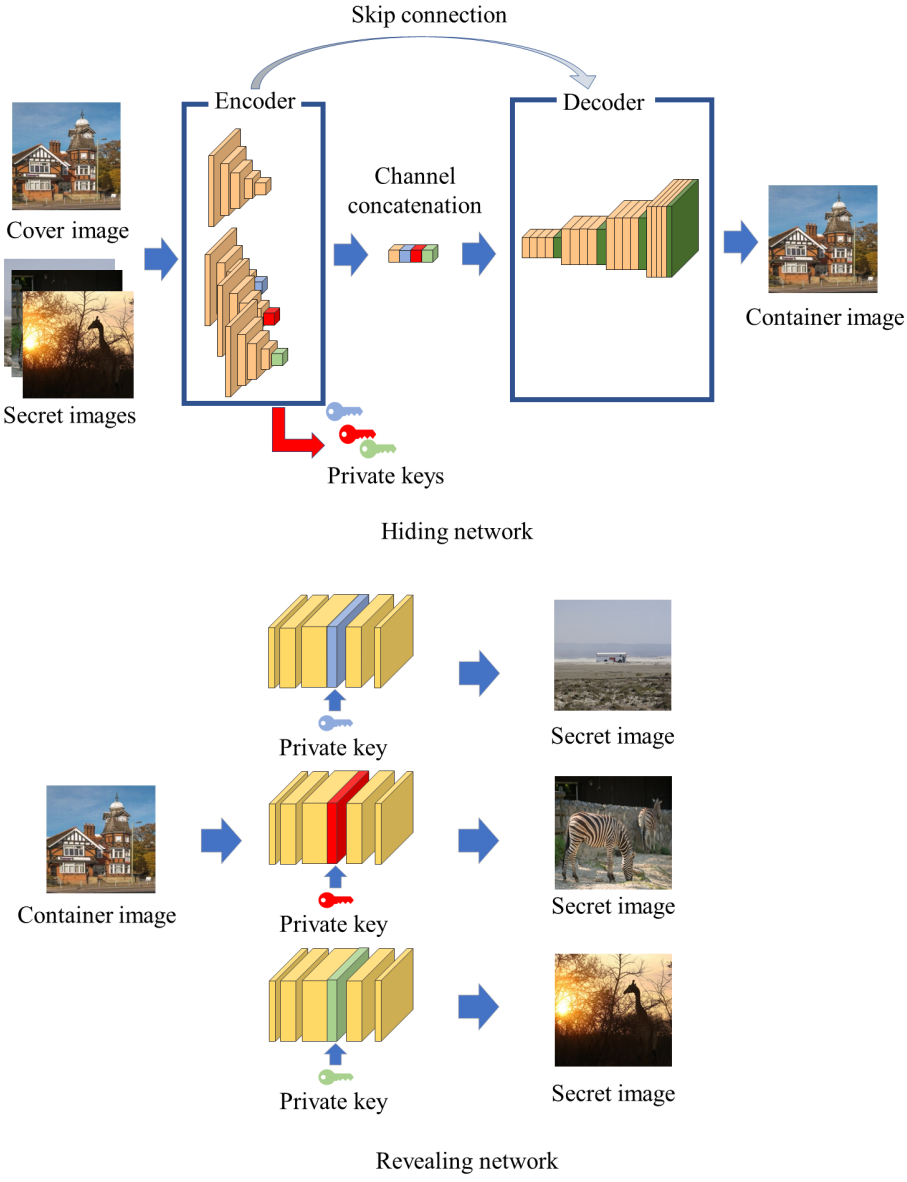

Figure 2. Overall pipeline of our steganography model. In the hiding network, the encoder takes a cover image and secret images as inputs and produces visual features for each image. Then decoder takes the concatenated visual features and produces a container image. The visual features of secret images are used as private keys. The revealing network takes a container image and a query private key as inputs then extracts a hidden secret image corresponding to the query private key.

Table 1. Hiding Network Architecture.

Step	Feature-Map (W × H× D)	Output
4 × 4 Conv+BN+LeakyReLU	128 × 128 × 64	2 × 2 × 512
4 × 4 Conv+BN+LeakyReLU	64 × 64 × 256	
4 × 4 Conv+BN+LeakyReLU	32 × 32 × 512	
4 × 4 Conv+BN+LeakyReLU	16 × 16 × 512	
4 × 4 Conv+BN+LeakyReLU	8 × 8 × 512	
4 × 4 Conv+BN+LeakyReLU	4 × 4 × 512	
4 × 4 Conv+ReLU	2 × 2 ×512	
4 × 4 TransConv+BN+LeakyReLU	4 × 4 × 512	256 × 256 × 3
4 × 4 TransConv+BN+LeakyReLU	8 × 8 × 512	
4 × 4 TransConv+BN+LeakyReLU	16 × 16× 512	
4 × 4 TransConv+BN+LeakyReLU	32 × 32 × 256	
4 × 4 TransConv+BN+LeakyReLU	64 × 64 × 128	
4 × 4 TransConv+BN+LeakyReLU	128 × 128 × 64	
TransConv+ReLU	256 × 256 × 3	

3.2. Revealing Network

In the revealing stage, the revealing network extracts a secret image corresponding to a query private key from the container image as follows:

$$S'_n = g(C', K_n; \phi_g), \tag{2}$$

where ϕ_g is the trainable parameters of the reveal network $g(\cdot)$. To be specific, the revealing network consists of six 3×3 convolutional blocks without downsampling. Therefore, the spatial dimension is maintained while passing through the revealing network. As shown in Figure 2, the query private key is resized to the same spatial size using nearest interpolation as the output of the third convolution layer, then concatenated with intermediate activations of the container image obtained from the revealing network. Finally, a hidden secret image $S'_n \in \mathbb{R}^{256\times256\times3}$ corresponding to the query private key K_n is reconstructed. Architecture details of the revealing network are presented in Table 2.

Table 2. Revealing Network Architecture.

Step	Feature-Map (W × H × D)	Output
3 × 3 Conv+BN+ReLU	256 × 256 × 64	256 × 256 × 3
3 × 3 Conv+BN+ReLU	256 × 256 × 128	
3 × 3 Conv+BN+ReLU	256 × 256 × 256	
3 × 3 Conv+BN+ReLU	256 × 256 × 128	
3 × 3Conv+BN+ReLU	256 × 256 × 64	
3 × 3 Conv+Sigmoid	256 × 256 × 3	

3.3. Training

Both hiding and reveal networks are trained in an end-to-end manner. The loss function of our steganography model is defined as follows:

$$\mathcal{L} = E\left[\left\|C' - C\right\| + \beta \sum_{n=1}^{N} \left\|S'_n - S_n\right\|\right], \tag{3}$$

where β is a weighting coefficient for balancing two terms in (3). We set $\beta = 0.75$ in our all experiments.

4. Experimental Results

For training, we have randomly collected 20,000 training images from MS-COCO train dataset, 5000 validation images from MS-COCO validation dataset, and 5000 test images from the MS-COCO test dataset [21]. All the images are resized to 256×256 using bicubic interpolation. Our method is implemented using PyTorch on Ubuntu 18.04 with a Titan RTX GPU. The proposed model is trained for 100 epochs with an Adam optimizer [22] with the learning rate 1×10^{-3}. The learning rate is multiplied by 0.2 when the loss has stopped decreasing.

4.1. Model Analysis

We first have investigated which feature layer is useful for the private key generation. For this purpose, we compare the performance with private keys selected from the output features of the penultimate ($K^p \in \mathbb{R}^{4\times4\times512}$) and the last ($K^l \in \mathbb{R}^{2\times2\times512}$) layers of the encoder. We have trained two networks with each private key to hide 1–5 images. Each network is evaluated by comparing PSNR [23] and SSIM [24] of the container and retrieved images with corresponding original images by using correct and random keys. As shown in Table 3, K^p and K^l show similar performance with $N = 1$ regardless of the key size. With the increasing number of hidden images, however, K^p still shows reliable performance while K^l suffers from severe performance degradation. This behavior is expected because the same amount of information should be reconstructed with the less amount of private key information for K^l compared with K^p. Also, as reported in the third column of Table 3, we obtain poor quantitative results when random keys are used. It shows that only the authorized private key can access the hidden image.

Table 3. Quantitative results for both K^p and K^l according to N. There are also results of revealed secret images with random keys.

N	C vs. C′ (PSNR/SSIM)		S vs. S′ (PSNR/SSIM)		S vs. S′ with Random Key (PSNR/SSIM)	
	K^p	K^l	K^p	K^l	K^p	K^l
1	34.54/0.9780	33.67/0.9707	36.55/0.9371	34.70/0.9269	13.79/0.5399	16.48/0.5012
2	33.05/0.9292	30.45/0.9103	28.31/0.8651	26.01/0.8749	9.03/0.2923	12.09/0.4264
3	30.58/0.9121	28.45/0.8748	27.70/0.8098	24.61/0.7317	10.74/0.2213	11.41/0.3838
5	28.25/0.8549	27.02/0.8434	23.25/0.6267	20.10/0.5159	11.10/0.1984	10.66/0.2641

Figure 3 shows encryption and decryption results of the proposed algorithm with $N = 3$. In most cases, we successfully generate container images with minimum distortion compared to cover images regardless of the private key size. For the extraction of secret images, each private key accurately reconstructs the corresponding original hidden image. In particular, K^p successfully extracts all of the secret images with high quality compared to those extracted with K^l due to the high capacity of private keys (i.e., $4\times$ capacity). Note that decrypted images with K^l often suffer from slightly blurry and noisy images. In addition, visual artifacts originated from the other secret images are often observed too. Furthermore, we verify the robustness of our algorithm by feeding random private keys to the revealing network together with the container image. Regardless of the capacity of private keys, secret images are not correctly extracted with random private keys as shown in Figure 3. In detail, the extracted images severely suffer from mixed textures and noises and it makes the hidden information indistinguishable. Therefore, these results prove that the proposed algorithm can provide permission for a certain hidden image to an authorized person.

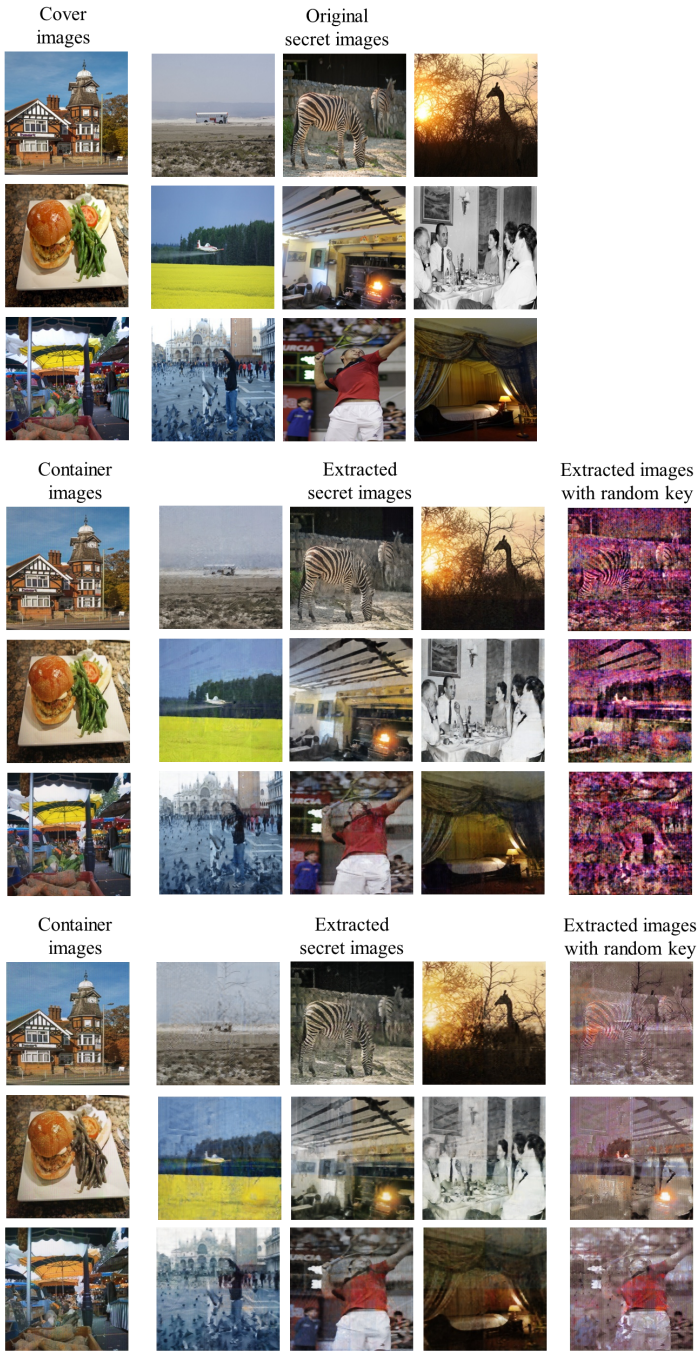

Figure 3. Qualitative Results of our steganography model when $N = 3$. Two types of key size were tested. The three examples above are the original cover and secret images. The three examples middle are the results of using K^p, and the three examples below are the results of using K^l.

4.2. Robustness to Steganalysis

As conducted in [8], we investigate the robustness of our method against StegExpose [25], which is a popular LSB-based steganalysis method. If the existing staganalysis technique does not distinguish the container image and the cover image well, it can be regarded that the proposed method hides the secret images well. For experiments, we utilize 200 pairs of cover and container images to obtain receiver operating characteristic (ROC) curves with varying threshold values. As shown in Figure 4a, ROC curves with both K^p and K^l are close to the case of a random guess (i.e., a straight diagonal line). It means that the proposed method hides secret images well because performance of the steganalysis is similar to random guessing. Also, area under the curve (AUC) values for

both K^p and K^l are very low along with N, as illustrated in Figure 4b. These experimental results show that the proposed method is quite robust to the common steganalysis method regardless of the size of the private key.

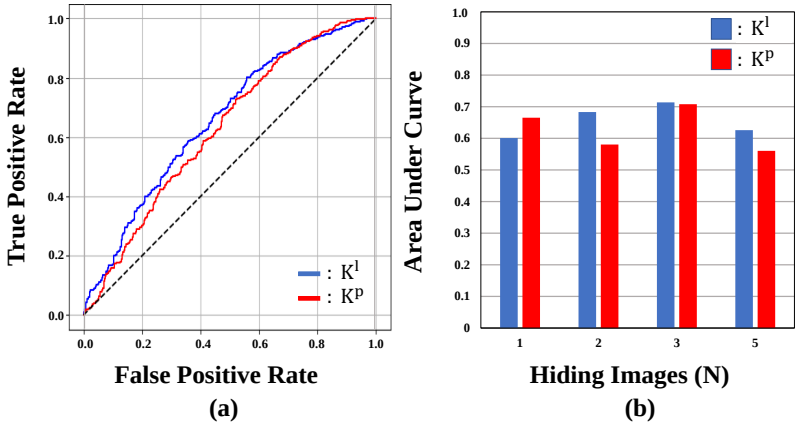

Figure 4. (**a**) ROC curves according to the size of the key. (**b**) Area under the curve (AUC) values according to N (Red: K^p, Blue: K^l).

4.3. Effects of Noise

To verify the susceptibility of our algorithm to private key contamination, we perform experiments to check how well the revealing network extracts hidden images when noise is added to the private key. To this end, we extracted secret images using private keys with noises. In particular, we added Gaussian or salt-and-pepper (S&P) noises to private keys with various noise levels, and extracted secret images with contaminated keys. Quantitatively, performances decrease as the noise level increases as shown in Figure 5. In particular, in the case of Gaussian noise, we can check that K^p is more susceptible to high noise levels compared to K^l. Meanwhile, there is not much visual distortion at low noise levels as shown in Figure 6c,f. In other words, the private key can withstand a low level of noise, but it is difficult to extract the original secret image when the noise is severe.

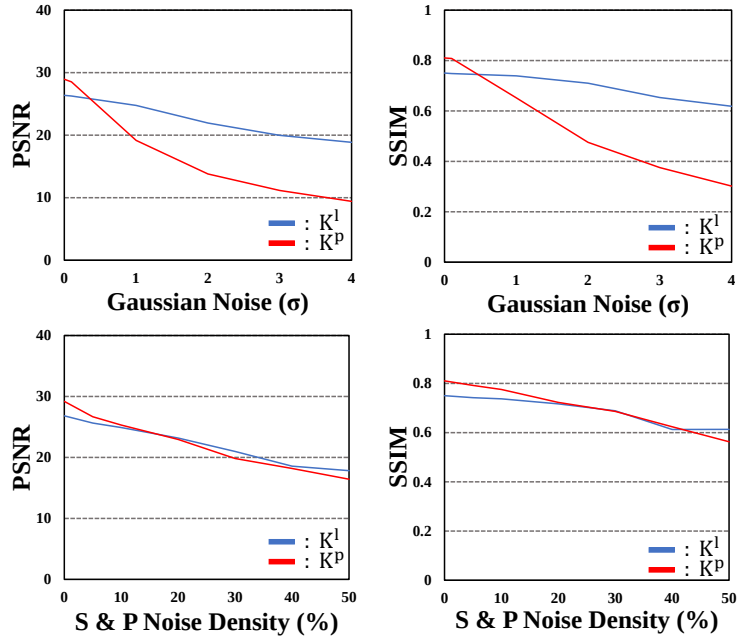

Figure 5. Average PSNR and SSIM measures on secret images reconstructed from private keys contaminated by various noises. (Red: K^p, Blue: K^l).

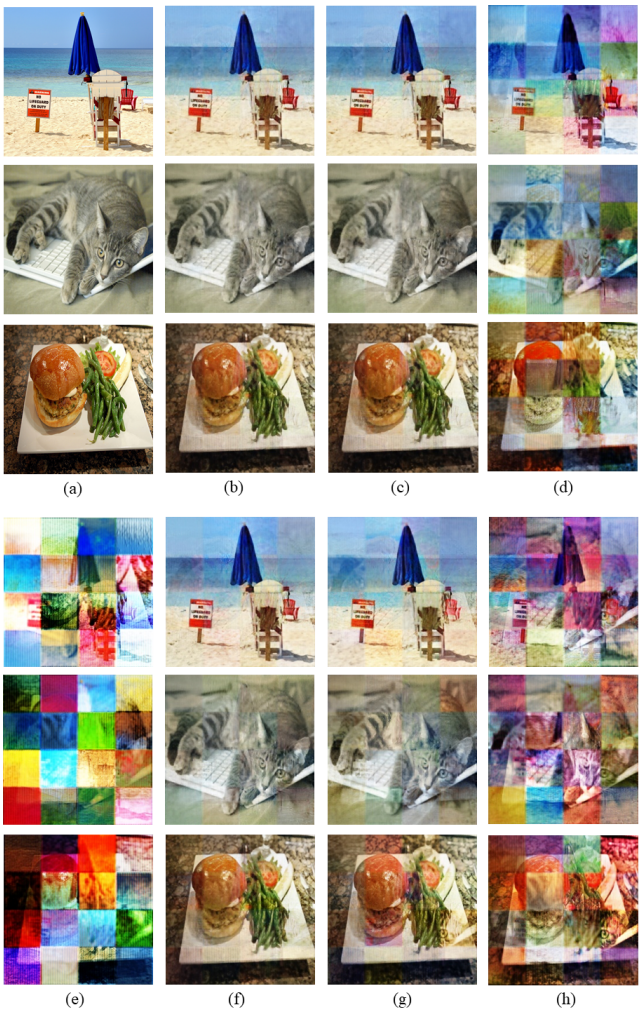

Figure 6. Results of inserting private key (K^p) with noise when $N = 3$. (**a**) Original secret images. (**b**) Results from noise-free private keys. (**c**–**e**) Results from private keys with Gaussian noise (σ = 0.1, 1, 3). (**f**–**h**) Results from private keys with S&P noises (density = 5%, 10%, 50%).

5. Conclusions

In this paper, we extended the concept of the private key to the multi-image steganography, which hides multiple secret images within a single cover image. Our steganography model takes stack of secret images and a cover image as inputs then produces a container image and private keys for each secret image. In order to extract a secret image from the container image, the corresponding private key is required. The proposed model provides a hidden image only when a proper private key is provided, and does not disclose information about other hidden images. Through extensive experiments, we verified the effectiveness of our method under various conditions (i.e., random key and noisy key).

Author Contributions: Conceptualization, D.C.; formal analysis, J.P.; investigation, S.W., and D.C.; data curation, H.K.; writing–original draft preparation, D.C.; writing—review and editing, S.W.; supervision, D.C. and J.P.; funding acquisition, D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government(MOE).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
2. Wu, D.C.; Tsai, W.H. A steganographic method for images by pixel-value differencing. *Pattern Recognit. Lett.* **2003**, *24*, 1613–1626. [CrossRef]
3. Pevnỳ, T.; Filler, T.; Bas, P. Using high-dimensional image models to perform highly undetectable steganography. In *International Workshop on Information Hiding*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 161–177.
4. Holub, V.; Fridrich, J.; Denemark, T. Universal distortion function for steganography in an arbitrary domain. *EURASIP J. Inf. Secur.* **2014**, *2014*, 1–13. [CrossRef]
5. Holub, V.; Fridrich, J. Designing steganographic distortion using directional filters. In Proceedings of the 2012 IEEE International Workshop on Information Forensics and Security (WIFS), Costa Adeje, Spain, 2–5 December 2012; pp. 234–239.
6. Chen, P.Y.; Lin, H.J. A DWT based approach for image steganography. *Int. J. Appl. Sci. Eng.* **2006**, *4*, 275–290.
7. Kaur, B.; Kaur, A.; Singh, J. Steganographic approach for hiding image in DCT domain. *Int. J. Adv. Eng. Technol.* **2011**, *1*, 72.
8. Baluja, S. Hiding images in plain sight: Deep steganography. In Proceedings of the Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; pp. 2066–2076.
9. Duan, X.; Jia, K.; Li, B.; Guo, D.; Zhang, E.; Qin, C. Reversible image steganography scheme based on a U-Net structure. *IEEE Access* **2019**, *7*, 9314–9323. [CrossRef]
10. Baluja, S. Hiding images within images. *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)* **2019**, *42*, 1685–1697. [CrossRef] [PubMed]
11. Channalli, S.; Jadhav, A. Steganography an art of hiding data. *arXiv* **2009**, arXiv:0912.2319.
12. Zhang, X.; Wang, S. Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognit. Lett.* **2004**, *25*, 331–339. [CrossRef]
13. Goljan, M.; Fridrich, J.; Holotyak, T. New blind steganalysis and its implications. In *Security, Steganography, and Watermarking of Multimedia Contents (VIII)*; International Society for Optics and Photonics: Washington, DC, USA, 2006; Volume 6072, p. 607201.
14. Pevny, T.; Bas, P.; Fridrich, J. Steganalysis by subtractive pixel adjacency matrix. *IEEE Trans. Inf. Forensics Secur. (TIFS)* **2010**, *5*, 215–224. [CrossRef]
15. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241
16. Springenberg, J.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. In Proceedings of the International Conference on Learning Representation Workshops (ICLRW), San Diego, CA, USA, 7–9 May 2015.
17. Xu, B.; Wang, N.; Chen, T.; Li, M. Empirical evaluation of rectified activations in convolutional network. *arXiv* **2015**, arXiv:1505.00853.
18. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the Int'l Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 448–456.
19. Noh, H.; Hong, S.; Han, B. Learning deconvolution network for semantic segmentation. In Proceedings of the Int'l Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1520–1528.
20. Agarap, A.F. Deep learning using rectified linear units (relu). *arXiv* **2018**, arXiv:1803.08375.
21. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2014; pp. 740–755.
22. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representation (ICLR), San Diego, CA, USA, 7–9 May 2015.
23. Hore, A.; Ziou, D. Image quality metrics: PSNR vs. SSIM. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; pp. 2366–2369.
24. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
25. Boehm, B. Stegexpose-A tool for detecting LSB steganography. *arXiv* **2014**, arXiv:1410.6656.