symmetry

MDPI

Article

Separable Reversible Data Hiding in Encrypted Signals with Public Key Cryptography

Wei-Liang Tai [1] and Ya-Fen Chang [2,*]

[1] Department of Information Communications, Chinese Culture University, Taipei 111, Taiwan; tai.wei.liang@gmail.com

[2] Department of Computer Science and Information Engineering, National Taichung University of Science and Technology, Taichung 404, Taiwan

* Correspondence: cyf@cs.ccu.edu.tw; Tel.: +886-4-2219-6320

Received: 5 December 2017; Accepted: 9 January 2018; Published: 10 January 2018

Abstract: We propose separable reversible data hiding in an encrypted signal with public key cryptography. In our separable framework, the image owner encrypts the original image by using a public key. On receipt of the encrypted signal, the data-hider embeds data in it by using a data-hiding key. The image decryption and data extraction are independent and separable at the receiver side. Even though the receiver, who has only the data-hiding key, does not learn about the decrypted content, he can extract data from the received marked encrypted signal. However, the receiver who has only the private key cannot extract the embedded data, but he can directly decrypt the received marked encrypted signal to obtain the original image without any error. Compared with other schemes using a cipher stream to encrypt the image, the proposed scheme is more appropriate for cloud services without degrading the security level.

Keywords: separable reversible data hiding; homomorphic cryptosystem; encrypted signal; public key cryptography

1. Introduction

With the ease of editing and the Internet distribution for digital media, copyright protection and the prevention of malicious tampering are important and challenging topics. Data hiding has been proposed to alleviate these concerns, which involves hiding data in media for various applications, such as copyright protection, image authentication, and access control. However, hiding data inevitably damages the content, though the distortion is imperceptible to human eyes. In some scenarios, such as encrypted data and medical images, any distortion is not tolerable. Hence, reversible data hiding (RDH) presents a solution to the problem of how to embed, in a lossless manner, data into the digital media such that the media can be completely recovered after data extraction.

The image owner may not believe the cloud service providers, which disposed the owner to encrypt the image before sending it to the cloud. However, traditional image processing is usually used before encryption or after decryption. Hence, the ability to directly process the encrypted signal while keeping the plain text unrevealed is desired. Therefore, reversible data hiding in encrypted signals provides privacy preserving services, where the data-hider can hide additional data in the encrypted signal for some applications such as authentication or annotation.

In 2011, Zhang [1] presented a reversible data hiding method in encrypted images. He encrypted the original image by using an exclusive or (XOR) operation with a cipher stream. He flipped least significant bits (LSBs) of selected encrypted pixels to hide the additional data. After data extraction, he used the spatial correlation to recover the original image. The resulting error rate of data extraction was decreased by Hong et al. [2]. They considered the pixel correlations in neighboring blocks and used the side-match scheme to improve the accuracy of data extraction. To improve the hiding capacity,

Zhang et al. [3] chose half of the fourth LSB as the space to carry the data to be embedded. Different from encrypted images, Qian et al. [4] embedded data into the Joint Photographic Experts Group (JPEG) encrypted stream. These schemes tried to directly vacate room after encryption (VRAE) in order to embed data, as shown in Figure 1. However, it is difficult to find extra hiding space after encryption when the encrypted image achieves maximum entropy. Thus, the VRAE schemes may result in small payloads and some error rates introduced by data extraction.

To overcome these drawbacks, Ma et al. [5] reserved room before encryption (RRBE) to hide data later, as illustrated in Figure 2. They emptied out LSBs of some pixels before encryption, and used an RDH scheme to embed these LSBs into other pixels. Zhang et al. [6] estimated some pixels before encryption to embed data in these estimating errors. To take advantage of the correlation between neighbor pixels, Cao et al. [7] considered the patch-level sparse representation and used sparse coding to hide the data. At the receiver side of above schemes, the image decryption and data extraction have to be processed together. The original image content needs to be revealed before data extraction in order to retrieve the embedded data, as shown in Figure 3. However, in some scenarios, the owner does not want the receiver who has no encryption key to know the original content.

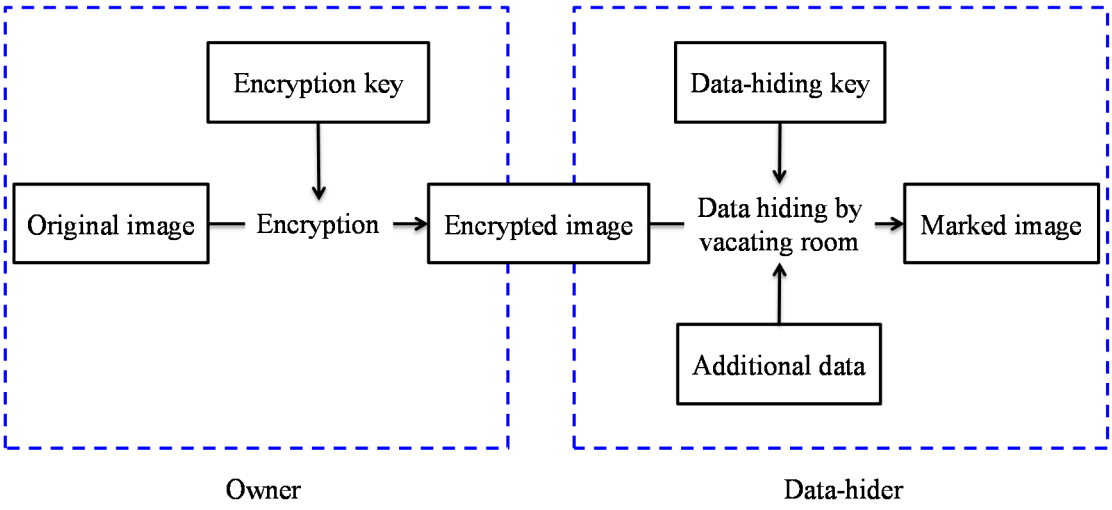

Figure 1. Framework VRAE [4].

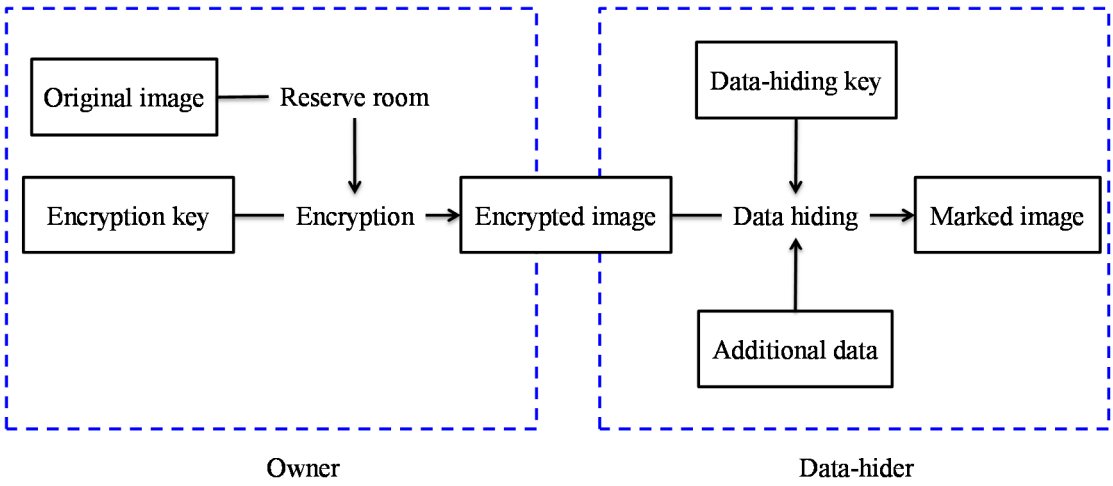

Figure 2. Framework RRBE [5].

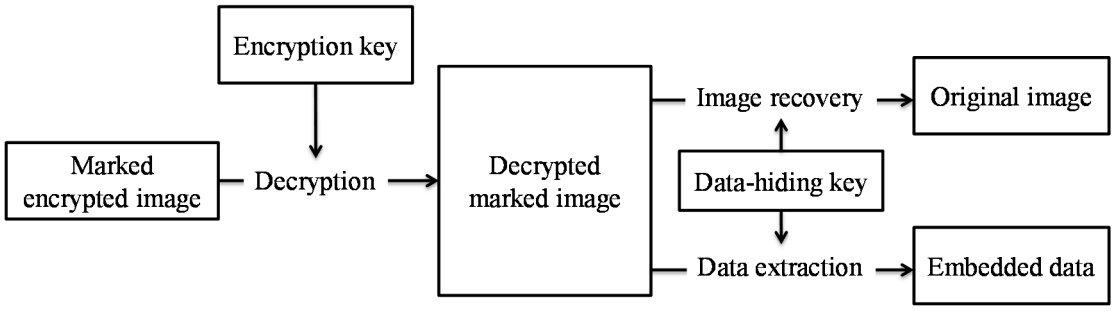

Figure 3. Non-separable reversible data hiding in encrypted image [1].

Thus, Zhang [8] presented a separable framework for RDH in encrypted image. Before transmitting the original image to the data-hider, the image owner firstly encrypted it with an encryption key. The data-hider embedded data in the received encrypted image by using the VRAE or RRBE framework with a data-hiding key. As shown in Figure 4, there are three cases occurring at the receiver side. In case 1, the receiver can use a data-hiding key to directly extract the data from the marked encrypted image without the knowledge of the original content. In case 2, only when the receiver has both a data-hiding key and an encryption key can he extract the embedded data and restore the image to its original form. In case 3, the receiver cannot extract the embedded data, but can decrypt the marked image and obtain an approximate image close to the original image. Wu and Sun [9] used the prediction error to propose a joint RDH and a separable RDH schemes in encrypted images. Xiao and Chen [10] left some space in the image for embedding data later to achieve separability. The quality of the decrypted image is improved by Liao et al. [11] using compressive sensing and discrete Fourier transform (DFT). Qian and Zhang [12] compressed some bits from the encrypted image to find room for hiding. In 2017, Wu et al. [13] proposed a separable RDH scheme for palette images. They used palette colors to build color triples, and embedded those indices into the encrypted image.

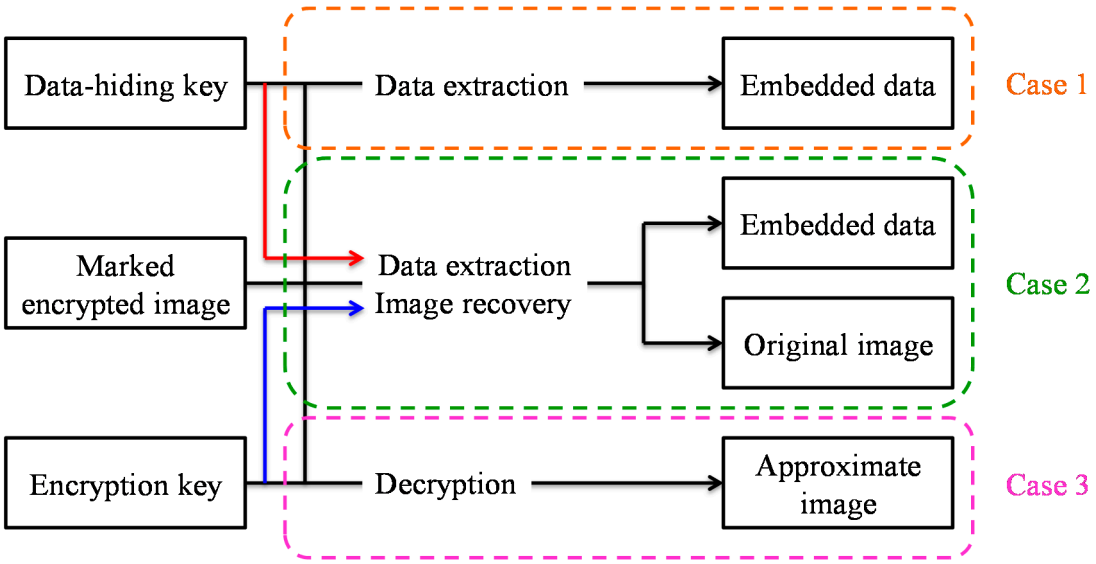

Figure 4. Three cases at the receiver side of separable reversible data hiding [8].

The above-mentioned methods embed the additional data into encrypted images by using symmetric cryptography. However, in some applications, the transmission of a secret key via a secure channel is impractical. Chen et al. [14] proposed a RDH method for encrypted signal with public key cryptosystem. They used Paillier encryption [15] to encrypt an image, and embedded data

into adjacent encrypted pixels. Zhang et al. [16] proposed a combined RDH for images encrypted with public-key cryptosystem according to the homomorphic property.

In this paper, we present a separable RDH in encrypted signals with public key cryptography. The original image is encrypted by Paillier encryption with a public key. After that, the data-hider directly hides the additional data in the encrypted signal with a data-hiding key. In our separable framework, there are two cases occurring at the receiver side. The receiver who has only the data-hiding key can directly extract the embedded data from the received marked encrypted signal without prior knowledge of the original content. However, the receiver who has only the private key can directly decrypt the received marked encrypted signal to obtain the original image without loss, but cannot retrieve the embedded data. The proposed method ensures that the image decryption and data extraction are separable at the receiver side. Moreover, compared with the image encrypted with a cipher stream, the proposed scheme is more applicable in the cloud without degrading the security level.

The paper is organized as follows: In Section 2, we describe the Paillier homomorphic cryptosystem. The proposed separable RDH scheme in encrypted signals is introduced in Section 3, where we also describe the procedures for extracting the embedded data and recovering the original image. The proposed method is experimentally validated in Section 4. Finally, the paper is concluded in Section 5.

2. Paillier Cryptosystem

The Paillier cryptosystem [15], whose security is under the RSA strong assumption, is a probabilistic asymmetric cryptography. Based on the additive homomorphic property, Paillier encryption is extensively used for privacy-preserving applications. The cryptosystem is depicted below. For key generation, the sender randomly chooses two large primes p and q, where gcd(pq, $(p–1)(q–1)$) = 1, and the sender calculates $n = pq$, $\lambda = \mathrm{l\,cm}\ (p-1, q-1)$, and selects a random integer $g \in \mathbb{Z}^*_{n^2}$ where gcd($L(g^\lambda \bmod n^2)$, n) = 1, and $L(x) = \frac{x-1}{n}$. Finally, (n, g) is the public key and (λ) is the private key.

Given a message $m \in \mathbb{Z}^*_n$, the sender randomly chooses integer $r \in \mathbb{Z}^*_n$, and then computes the ciphertext of m:

$$c = E_{pk}\ (m, r) = g^m\ r^n \bmod n^2, \tag{1}$$

where E is the encryption function and pk is the public key.

The receiver can decrypt the ciphertext with private key:

$$m = D_{sk}(c) = \frac{L\left(c^\lambda \bmod n^2\right)}{L\left(g^\lambda \bmod n^2\right)} \bmod n, \tag{2}$$

where D is the decryption function and sk is the private key.

The additive homomorphic properties are held by Paillier encryption. Given two encryption functions $E_{pk}\ (m_1, r_1)$ and $E_{pk}\ (m_2, r_2)$, the two functions are additively homomorphic on $\mathbb{Z}_n$:

$$E_{pk}(m_1,\ r_1)E_{pk}(m_2,\ r_2) = g^{m_1+m_2}(r_1r_2)^n \bmod n^2 = E_{pk}(m_1 + m_2, r_1r_2).$$

Thus:

$$D_{sk}\ (E_{pk}\ (m_1, r_1) \times E_{pk}\ (m_2, r_2)) = m_1 + m_2. \tag{3}$$

Additionally, this brings about the following properties:

$$D_{sk}\ (E_{pk}\ (m_1, r_1)^k) = km_1,$$

$$D_{sk}\ (E_{pk}\ (m_1, r_1) \times g^{m_2} \bmod n^2) = m_1 + m_2,$$

$$D_{sk}\left(E_{pk}(m_1, r_1)^{m_2} \bmod n^2\right) = m_1 m_2.$$

These additively homomorphic identities are known to be appropriate for designing proxy signatures, electronic voting, watermarking, and authentication.

3. Proposed Scheme

The proposed scheme consists of image encryption, data embedding, data extraction, and image recovery. Figure 5 shows the framework of the proposed scheme. The image owner uses a public key to encrypt the original image to generate an encrypted signal. After receiving the encrypted signal, the data-hider is able to directly hide data in it with a data-hiding key. At the receiver side, the image recovery and data extraction are independent. The receiver can easily retrieve the embedded data from the received marked encrypted signal with only the data-hiding key. When the receiver has only the private key, he can directly decrypt the received marked encrypted signal regardless of data extraction to perfectly recover the original image rather than an approximate image.

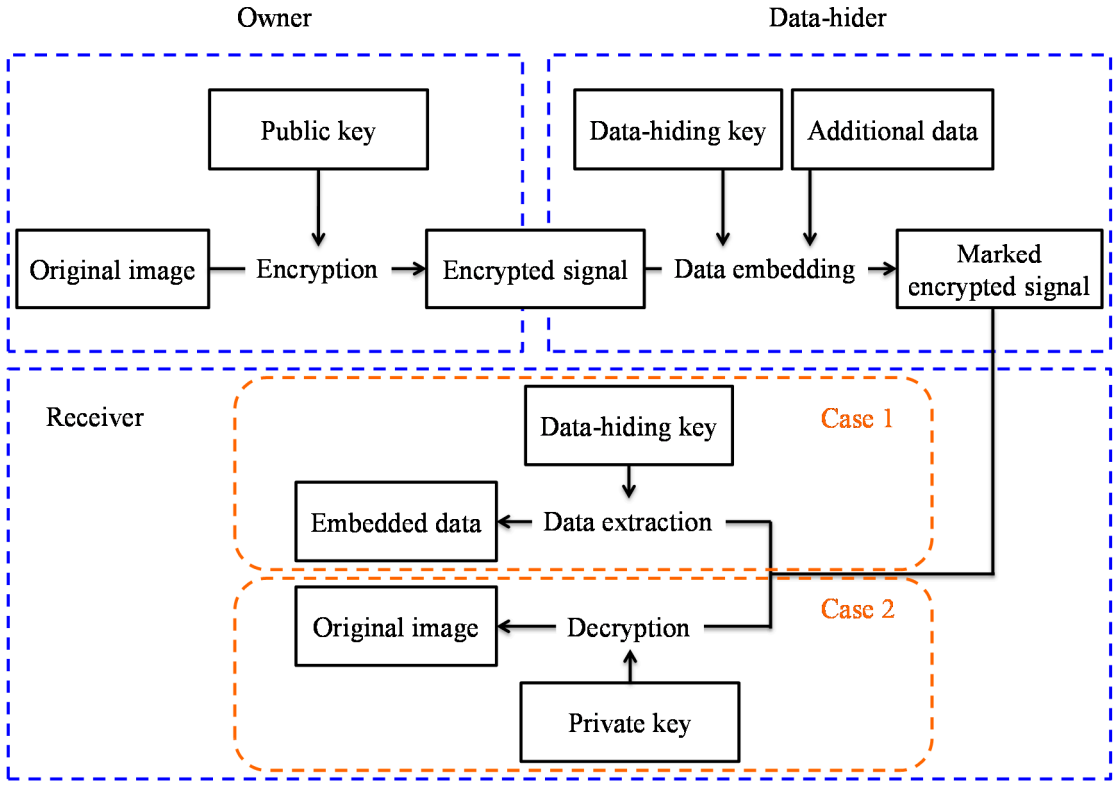

Figure 5. Framework of the proposed scheme.

3.1. Image Encryption

Suppose that the original image is an 8-bit grayscale image of $W \times H$ pixels, and the pixel value $x_{i,j}$ denotes the grayscale value at the coordinate of (i, j), where $1 \le i \le H$, $1 \le j \le W$, and $0 \le x_{i,j} \le 255$. The detailed procedure is listed below.

Step 1. For each pixel $x_{i,j}$, convert $x_{i,j}$ to ${x^1}_{i,j}$ and ${x^2}_{i,j}$, where $x_{i,j} = {x^1}_{i,j} + {x^2}_{i,j}$.

Step 2. Choose a random integer $r_1 \in \mathbb{Z}_n^*$, and then computes the encryption function E_{pk} $({x^1}_{i,j}, r_1)$ with a public key by Equation (1).

Step 3. Choose a random integer $r_2 \in \mathbb{Z}_n^*$, and then computes the encryption function E_{pk} $({x^2}_{i,j}, r_2)$ with a public key by Equation (1) so as to meet that E_{pk} $({x^1}_{i,j}, r_1) \neq E_{pk}$ $({x^2}_{i,j}, r_2)$.

Step 4. All the encrypted units comprise the encrypted signal.

3.2. Data Embedding

The following algorithm describes how to embed data into the received encrypted signal. Suppose that the received encrypted signal with size of $W \times H \times 2 \times (\lfloor \log_2^{n^2} \rfloor + 1)$ bits consists of encrypted units $EU_i = (EU^1{}_i, EU^2{}_i)$, where $1 \leq i \leq W \times H$.

Step 1. Construct a non-repeat random embedding sequence using the data-hiding key.
Step 2. Embed a secret bit into an encrypted unit EU_i according to the embedding sequence. If the secret bit is 1 and $EU^1{}_i < EU^2{}_i$, $EU^1{}_i$ swaps $EU^2{}_i$.
Step 3. If the secret bit is 0 and $EU^1{}_i > EU^2{}_i$, $EU^1{}_i$ swaps $EU^2{}_i$.
Step 4. Generate a marked encrypted signal when all bits are embedded.

3.3. Data Extraction

Suppose that the received marked encrypted signal with size of $W \times H \times 2 \times (\lfloor \log_2^{n^2} \rfloor + 1)$ bits consists of marked encrypted units $MEU_i = (MEU^1{}_i, MEU^2{}_i)$, where $1 \leq i \leq W \times H$. If the receiver has only a data-hiding key, the data extraction procedure is listed below.

Step 1. Step 1. Construct a non-repeat random embedding sequence using the data-hiding key.
Step 2. Extract a secret bit from a marked encrypted unit MEU_i according to the embedding sequence. If $MEU^1{}_i > MEU^2{}_i$, the extracted bit is 1.
Step 3. If $MEU^1{}_i < MEU^2{}_i$, the extracted bit is 0.
Step 4. Obtain the embedded data when all the bits are extracted.

3.4. Image Recovery

Suppose that the received marked encrypted signal with size of $W \times H \times 2 \times (\lfloor \log_2^{n^2} \rfloor + 1)$ bits consists of marked encrypted units $MEU_i = (MEU^1{}_i, MEU^2{}_i)$, where $1 \leq i \leq W \times H$. We assume that the receiver has only a private key. The image recovery procedure is listed below.

Step 1. Decrypt the marked encrypted unit MEU_i using the private key by:

$$D_{sk}\left((MEU^1{}_i \times MEU^2{}_i) \bmod n^2\right) = x^1{}_{i,j} + x^2{}_{i,j} = x_{i,j}. \quad (4)$$

Step 2. Recover the original image when all marked encrypted units are decrypted.

An example of the proposed method is given. Consider the original pixel with value $x_{i,j} = x^1{}_{i,j} + x^2{}_{i,j} = 100 + 68 = 168$, and set the secret bit as 0, two primes as $p = 17$ and $q = 19$. Thus, (323, 324) is the public key, and (144) is the private key. Compute the encrypted unit $EU_i = (EU^1{}_i, EU^2{}_i) = (E_{pk}(100, 7), E_{pk}(68, 11)) = (74{,}871, 34{,}549)$. In the data embedding phase, because the secret bit is 0 and $EU^1{}_i > EU^2{}_i$, $EU^1{}_i$ swaps $EU^2{}_i$. Therefore, the marked encrypted unit $MEU_i = (EU^2{}_i, EU^1{}_i) = (34{,}549, 74{,}871)$. At the receiver side, the receiver who has only data-hiding key can extract secret bit 0 because $MEU^1{}_i < MEU^2{}_i$. However, if the receiver has only a private key, he can decrypt the marked encrypted unit MEU_i to obtain the original pixel by computing $D_{sk}\left((MEU^1{}_i \times MEU^2{}_i) \bmod n^2\right) = D_{sk}\left((34{,}549 \times 74{,}871) \bmod 323^2\right) = D_{sk}(89{,}282) = 168$.

4. Experimental Results

We used the test image Lena sized 512 × 512 pixels as the original image in the experiments. Different images do not affect the performance of the proposed method. Table 1 lists the embedding rate and PSNR of the directly decrypted Lena image. In this experiment, a total of 512 × 512 = 262,144 bits are hidden in each encrypted unit. As a result, the corresponding embedding rate is 1 bit per pixel (bpp), whatever the kind of images are selected as the original image. Note that the $+\infty$ denotes that the original image is perfectly recovered without any loss.

Table 1. The embedding rate and PSNR of directly decrypted image for Lena.

Original Image	Embedding Rate (bpp)	PSNR (dB)
Lena	1	+∞

Figure 6 compares the embedding rate and image quality for Lena with other schemes [2,3,5–8]. In our scheme, the directly decrypted image without data extraction is the same as the original one. Hence, the associated PSNR of decrypted image is +∞ dB regardless of the embedding rate. Other schemes [2,3,5–8] degraded the visual image quality of the decrypted image due to the embedding distortion. They cannot restore the original image with only the encryption key. Clearly, the performance of the proposed method is significantly better than that of schemes [2,3,5–8]. Table 2 summarizes the comparison of characteristics of the considered schemes [2,3,5–8]. We note that in our scheme, asymmetric cryptography is adopted for image encryption. Therefore, data expansion exists in the encrypted signal. However, based on additive homomorphic properties, the proposed scheme is able to be further applied to multimedia without infringing the privacy.

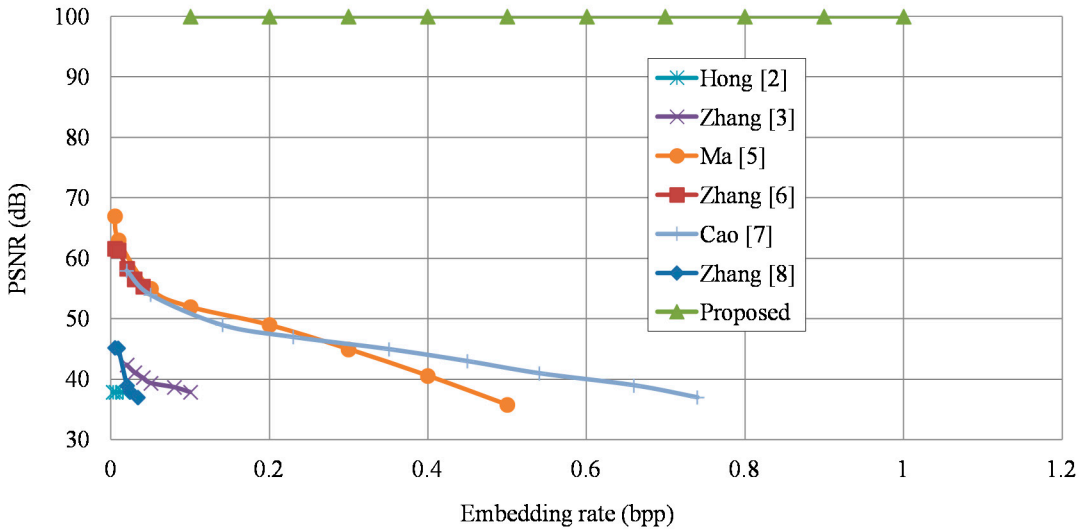

Figure 6. Performance comparison for Lena with existing schemes [2,3,5–8].

Table 2. Characteristics of various schemes [2,3,5–8].

Scheme	Encryption	Receiver	Embedding Rate	Expansion
Hong [2]	Stream cipher	Non-separable	0.015	No
Zhang [3]	Stream cipher	Separable	0.17	No
Ma [5]	Stream cipher	Separable	0.5	No
Zhang [6]	Symmetric key	Separable	0.04	No
Cao [7]	Stream cipher	Separable	0.74	No
Zhang [8]	Stream cipher	Separable	0.034	No
Proposed	Public key	Separable	1	Yes

5. Conclusions

This paper proposes a separate RDH method for images encrypted by public key cryptography. The two ciphertext values are exchanged with each other for embedding the additional data. Based on additive homomorphic properties, we can directly extract the embedded data from the encrypted domain without knowing the original content. Moreover, perfect image recovery can be directly

processed without prior data extraction. Since the content privacy can be securely preserved by Paillier encryption, the proposed scheme is appropriate for cloud services without degrading the security level.

Acknowledgments: This work was supported in part by Ministry of Science and Technology under the grants MOST 106-2221-E-034-006-, MOST 106-2622-E-034-002-CC3, and MOST 106-2410-H-025-006-.

Author Contributions: Wei-Liang Tai designed the algorithm, conducted all experiments, analyzed the results, wrote the manuscript, and conducted the literature review. Ya-Fen Chang conceived the algorithm, analyzed the results, and wrote the manuscript.

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

References

1. Zhang, X. Reversible data hiding in encrypted image. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
2. Hong, W.; Chen, T.S.; Wu, H. An improved reversible data hiding in encrypted images using side match. *IEEE Signal Process. Lett.* **2012**, *19*, 199–202. [CrossRef]
3. Zhang, X.; Qian, Z.; Feng, G.; Ren, Y. Efficient reversible data hiding in encrypted images. *J. Vis. Commun. Image Represent.* **2014**, *25*, 322–328. [CrossRef]
4. Qian, Z.; Zhang, X.; Wang, S. Reversible data hiding in encrypted JPEG bitstream. *IEEE Trans. Multimed.* **2014**, *16*, 1486–1491. [CrossRef]
5. Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F. Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
6. Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. *Signal Process.* **2014**, *94*, 118–127. [CrossRef]
7. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Trans. Cybern.* **2016**, *46*, 1132–1143. [CrossRef] [PubMed]
8. Zhang, X. Separable reversible data hiding in encrypted image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
9. Wu, X.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Process.* **2014**, *104*, 387–400. [CrossRef]
10. Xiao, D.; Chen, S. Separable data hiding in encrypted image based on compressive sensing. *Electron. Lett.* **2014**, *50*, 598–600. [CrossRef]
11. Liao, X.; Li, K.; Yin, J. Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform. *Multimed. Tools Appl.* **2017**, *76*, 20739–20753. [CrossRef]
12. Qian, Z.; Zhang, X. Reversible data hiding in encrypted image by distributed encoding. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 636–646. [CrossRef]
13. Wu, H.Z.; Shi, Y.Q.; Wang, H.X.; Zhou, L.N. Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1620–1631. [CrossRef]
14. Chen, Y.C.; Shiu, C.W.; Horng, G. Encry pted signal-based reversible data hiding with public key cryptosystem. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1164–1170. [CrossRef]
15. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. *Adv. Cryptol.* **1999**, *1592*, 223–238.
16. Zhang, X.; Long, J.; Wang, Z.; Cheng, H. Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 1622–1631. [CrossRef]