Copyright © 2018 Tech Science Press CMC, vol.55, no.3, pp.523-539, 2018

On the Privacy-Preserving Outsourcing Scheme of Reversible Data Hiding over Encrypted Image Data in Cloud Computing

Lizhi Xiong[1, *] and Yunqing Shi[2]

Abstract: Advanced cloud computing technology provides cost saving and flexibility of services for users. With the explosion of multimedia data, more and more data owners would outsource their personal multimedia data on the cloud. In the meantime, some computationally expensive tasks are also undertaken by cloud servers. However, the outsourced multimedia data and its applications may reveal the data owner's private information because the data owners lose the control of their data. Recently, this thought has aroused new research interest on privacy-preserving reversible data hiding over outsourced multimedia data. In this paper, two reversible data hiding schemes are proposed for encrypted image data in cloud computing: reversible data hiding by homomorphic encryption and reversible data hiding in encrypted domain. The former is that additional bits are extracted after decryption and the latter is that extracted before decryption. Meanwhile, a combined scheme is also designed. This paper proposes the privacy-preserving outsourcing scheme of reversible data hiding over encrypted image data in cloud computing, which not only ensures multimedia data security without relying on the trustworthiness of cloud servers, but also guarantees that reversible data hiding can be operated over encrypted images at the different stages. Theoretical analysis confirms the correctness of the proposed encryption model and justifies the security of the proposed scheme. The computation cost of the proposed scheme is acceptable and adjusts to different security levels.

Keywords: Cloud data security, re-encryption, reversible data hiding, cloud computing, privacy-preserving.

1 Introduction

Reversible data hiding (RDH) [Ni, Shi, Ansari et al. (2006)] as a lossless data hiding technique aims to exactly recover both the embedded secret information and the original cover multimedia data. It is quite desirable and helpful in some sensitive applications such as remote sensing, multimedia archive management and military communication. There are some requirements for RDH methods [Sachnev, Kim, Nam et al. (2009); Li, Yang and Zeng (2011); Xuan, Yao, Yang et al. (2006); Tian (2003); Khelifi (2018)]: 1)

[1] School of Computer and Software, Nanjing University of Information Science and Technology, No. 219, Ningliu Road, Nanjing 210044, China.

[2] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, University Heights Newark, New Jersey 07102, USA

[*] Corresponding author: Lizhi Xiong. Email: lzxiong16@163.com.

CMC. doi:10.3970/cmc.2018.01791 www.techscience.com/cmc

Embedding capacity and the quality of marked image may be high; 2) The security level of the method should be high. It is well expected that the distortion is reduced and the payload is improved, including coverless data hiding.

As we have entered the era of ‘big data’, the capability of multimedia data has dramatically increased and reached an unprecedented level [Zhu, Luo, Wang et al. (2011)]. Cloud computing as an advanced technique can provide huge storage space and on-demand access service, thus becoming a research platform for the multimedia big data. Therefore, users and enterprises with various capabilities as Data Owners (DOs), are highly motivated to store their huge amount of personal multimedia data files and computationally expensive tasks onto remote cloud servers [Sebe, Domingo-Ferrer, Martinez-Balleste et al. (2008)]. Reversible data hiding over outsourced multimedia data in cloud computing becomes a mainly research interest. However, the outsourcing of data storage and computation to the cloud raises great security and privacy concerns due to the different trust domains the data owner and the cloud. Obviously, the expose of original image data to the semi-trusted cloud service provider may inevitably reveal the data owner’s private information. To provide privacy guarantees for sensitive data, a straight-forward approach is to encrypt the sensitive multimedia data locally before outsourcing. It can be seen that the RDH over encrypted image data (RDHEI) is an intuitive and effective way to meet such requirement. RDH in encrypted images can transmit additional data in encryption domain to process authentication and content annotation.

To this end, many RDHEI schemes have been proposed in past years. One of the common techniques is based on manipulating the least-significant-bit (LSB) planes by directly replacing the three LSBs of the cover-image with the message bits, which is kind of the pixel-level compressive methods essentially. In Zhang [Zhang (2011)], the encrypted image is segmented into a number of non-overlapped blocks, while each block is divided into two sets. Each block carries one bit by flipping three LSBs of a set for predefined pixels. Hong et al. [Hong, Chen and Wu (2012)] gave an improved version based on [Zhang (2011)]. Specifically, they fully harness the pixels in calculating the smoothness of each block and consider the pixel correlations in the border of neighboring blocks. The resulting error rate of extracted-bits is thereby decreased.

Zhang et al. [Zhang, Qian, Feng et al. (2014)] use a half of the fourth LSB plane in an encrypted image to improve the payload of embedded bits. In JPEG domain, an RDH scheme in the encrypted JPEG images was proposed [Qian, Zhang and Wang (2014)]. The above schemes try to create room in the encrypted image, so called reserve room after image encryption. In fact, the scheme has a heavy cost of computation and a low capacity. To overcome the drawbacks, some schemes of reserve room before image encryption have been introduced in Ma et al. [Ma, Zhang, Zhao et al. (2013); Cao, Du, Wei et al. (2016); Zhang, Ma and Yu (2014)]. In Zhang et al. [Zhang, Ma and Yu (2014)], the secret data are embedded into an encrypted image by altering the prediction errors. A good predictor will enhance the performance of the method. Xiong et al. [Xiong, Xu and Shi (2018)] propose a lossless RDH in encrypted images scheme based on the Integer Wavelet Transform (IWT) and Orthogonal decomposition. The independence of orthogonal coefficients keeps the reversibility. Zhang et al. [Zhang, Long and Wang (2016)] proposed a RDH scheme in encrypted images with public public-key cryptography.

The above methods rapidly promote the development of RDH in encrypted image. But most of them work in traditional data transmission and a very few are suitable for cloud computing. As well known, cloud computing is a multi-user platform. Conventional two-party encryption is not suited for three-party data security in cloud computing. Re-encryption as a cryptographic algorithm involving three parties thus becomes a promising approach to maintain data confidentiality in cloud data services [Xiong, Xu and Xu (2015)]. Re-encryption permits a proxy server to transfer a ciphertext designated for one user to another ciphertext designated for another user without the need to have knowledge of the plaintext. Ateniese et al. [Ateniese, Fu, Green et al. (2006)] improved the concept of proxy re-encryption and applied it to data storage. In this scheme, the owner encrypts his/her files and outsourced them to a proxy server. The proxy server can transfer a ciphertext for the owner to a ciphertext for the requester if and only if he has obtained a re-encryption key. These methods have been widely adopted to secure data storage on trusted servers. However, with an untrusted or semi-trusted CSP, these methods are not applicable. Vimercati et al. [Vimercati, Foresti, Jajodia et al. (2007)] proposed a solution for securing data storage on untrusted servers based on key derivation methods. In this scheme, each file is encrypted with a symmetric key, and each user is assigned a secret key. To grant the access privilege for a user, the DO creates corresponding public tokens together with his secret key, from which the user is able to derive decryption keys of desired files. Then DO transmits these public tokens to the semi-trusted server (STS) and delegates the task of token distribution to STS. However, transferring these secret keys inherently requires additional secure channels, and these keys require rather expensive secure space to store them. The cost and complexities involved generally increase with the number of data users. Additionally, this method introduces symmetric encryption to encrypt DO data. In this case, the secret key is exposed easily in the re-encryption key generation phase and ciphertext decryption phase so that the data user can acquire the DO's private key. Therefore, this paper introduces a secure re-encryption model for Multimedia Data (MD) sharing services in cloud computing.

Therefore, re-encryption technique ensures multimedia data security in cloud computing. Motivated by the above observations, RDH over encrypted images based on re-encryption model is desired. This paper therefore proposes a privacy-preserving outsourcing scheme for reversible data hiding over encrypted image data that can not only ensure image data security without relying on the trustworthiness of cloud servers, but also transmits additional data in encryption domain. The data owner can full recover the image and extracted the embedded bits.

Our main contributions can be summarized as follows.

1) We design two novel secure RDHEI methods: reversible data hiding by homomorphic encryption and reversible data hiding in encrypted domain.

2) The two methods can be combined as a comprehensive scheme to protect the confidentiality of image data and transmit the additional data.

3) We carefully analyze our scheme to show that it can preserve much well the performance in terms of capacity and robustness as compared to the existing solutions.

The rest of this paper is organized as follows. The basic security primitives of our proposed scheme are described in Section 2. The reversible data hiding by homomorphic encryption and reversible data hiding in encrypted domain are introduced in Section 3 and 4, respectively. The combined data hiding scheme is shown in Section 5. An analysis of the proposed scheme is presented in Section 6. Experimental results are provided in Section 7. We conclude this paper with a discussion of the results in Section 8.

2 Our proposed basic security primitives

In this section, we introduce a re-encryption model for cloud computing and encryption method using EIGamal cryptosystem. The re-encryption model describes the procedures for all parties, including Data Owner (DO), Cloud Service Provider (CSP), and Data User (DU). The property of encryption method causes a special feature on data operation.

2.1 Re-encryption model in cloud computing

Since cloud computing has low hardware and software capital costs, low management and maintenance overhead, and universal on-demand data access, Cloud storage service become an attractive option for data owners. They would outsource their data to cloud servers for storage and assign some complex calculation for computation. However, data owners lose the physical control of data in the cloud, and therefore the trustworthiness of a cloud service provider become a factor and influences the development of the service. This paper introduces a re-encryption method without relying on this trustworthiness factor. A re-encryption model in cloud computing is illustrated in Fig. 1. The framework can be described as follows.

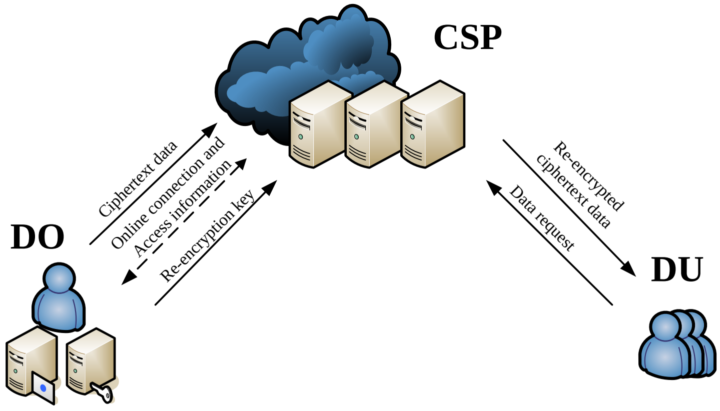

Figure 1: The framework of re-encryption model in cloud computing

First, a secure pair of private-public keys is generated by public key cryptography in data owner’s side. The DO encrypts his/her own data with the public key and obtains a ciphertext data. Then, the ciphertext data is outsourced to cloud servers.

Second, a secure pair of private-public keys is generated by public key cryptography in data user’s side. a DU sends a request for the desired data derived in several ways such as by searching the encrypted data, to the CSP. The CSP then transmits the request to the DO.

Third, after receiving the request, a re-encryption key, *rekey*, is generated with the DO's private key and access information on the DO's side, then the DO sends the *rekey* to the CSP.

Fourth, after receiving the *rekey*, the ciphertext would be re-encrypted with the *rekey*. The CSP will obtain the re-encrypted ciphertext.

Fifth, the DU downloads the re-encrypted ciphertext from the cloud servers. The re-encrypted ciphertext is decrypted with DU's private key on the DO's side.

The above procedures describe a privacy-preserving service in cloud computing. In the model, the confidentiality of the DO data is achieved using an asymmetric cryptographic algorithm in the cloud. This is because only the DO has a decryption key for the first layer ciphertext. The CSP can only access the encrypted cloud data. The syntactic definitions of re-encryption are shown in Tab. 1.

Table 1: Syntactic definition of re-encryption

Notations	Description
par	a public parameters
(sku,pku)	DU's secret/public key pair
(sko, pko)	DO's secret/public key pair
rekeyou	a re-encryption key
M	a plaintext
C	a first level ciphertext
C'	a re-encrypted ciphertext
Rekeygen (par, sko, pku)	a re-encryption key generator
E1 (M, pko)	an encryption function
D1 (C, sko)	a decryption function for C
E2 (C, rekeyou)	a re-encryption function
D (C', sku)	a decryption function for re-encrypted ciphertext

2.2 Encryption method using EIGamal cryptosystem

The EIGamal cryptosystem [EIGamal (1985)] is a public-key cryptosystem, which is based on discrete logarithms problem. The details are shown in Fig. 2. After encrypting a plaintext digital image, a string of big integers is generated and each of them represents an encrypted pixel value.

1) Assume that p is a large prime number and that (p-1) has a large prime factor; also, that g (g <p) is a primitive element in $GF(p) = \mathbb{Z}_p^*$.

2) Choose a random number x ($x \in \mathbb{Z}_p^*$) such that $Gcd(x, p-1) = 1$ and compute $y = g^x \bmod p$. Then, y and x is the public/private key pair. g and p are shared to the group users.

3) Encryption

Choose a random number k such that $Gcd(k, p-1) = 1$. The ciphertext is

$E(M) = (a,b) = (g^k \bmod p, y^k M \bmod p)$

4) Decryption

The plaintext is $M = b \cdot (a^x)^{-1} \bmod p$.

Figure 2: The procedure of EIGamal algorithm

As a public-key cryptography, the encryption keys pk$_o$, pk$_u$ are to be public known in re-encryption model. To add m_1 to another plain-text value m_2 in the encryption domain, the following operation can be performed on their encrypted values, E (m_1, pk$_o$) and E (m_2, pk$_o$) by

$$\mathrm{E}(m_1, \mathrm{pk}_o) = y^k \cdot m_1 \bmod p \tag{1}$$

$$\mathrm{E}(m_2, \mathrm{pk}_o) = y^k \cdot m_2 \bmod p \tag{2}$$

$$\mathrm{Opt}(m_1, m_2) = \mathrm{E}(m_1, \mathrm{pk}_o) + \mathrm{E}(m_2, \mathrm{pk}_o) = y^k \cdot (m_1 + m_2) \bmod p \tag{3}$$

where Opt($\cdot$) is an operation between the two encrypted values.

The corresponding plain-text value m' can be obtained by decrypting the cipher value $\mathrm{Opt}(m', \mathrm{pk}_o)$.

$$m' = \mathrm{D}(\mathrm{Opt}(m_1, m_2), \mathrm{sk}_o) = y^k \cdot (m_1 + m_2) \cdot (a^{\mathrm{sk}_o})^{-1} \bmod p = m_1 + m_2 \tag{4}$$

where m' is a decrypted data.

Therefore, in the re-encryption model, the plaintext M is encrypted and the corresponding ciphertext is obtained, E$_1$ (M, pk$_o$).

3 Reversible data hiding by homomorphic encryption

In this section, we introduce a reversible data hiding scheme by homomorphic encryption. First, data embedding method is shown based on homomorphic property. Second, the re-encryption operation is complete. Then Data Extraction and Image Recovery are presented.

3.1 Data embedding

Based on the above mechanism in Section 2.2, a data embedding method is developed as follows.

Given an image encrypted in EIGamal cryptosystem, a portion of the encrypted pixels are reserved to hide the side information such as the amount of bit values to be hidden and the number of encrypted pixel values used for data hiding.

To embed a bit value $h_i \in \{0,1\}$ into $\mathrm{E}_1(\mathrm{M}_i, \mathrm{pk}_o)$, which is the encrypted value of a pixel value M_i, an embedded value $\mathrm{Opt}(\mathrm{M}'_i, \mathrm{pk}_o)$ is generated by

$$\mathrm{Opt}(\mathrm{M}_i, \mathrm{h}_i) = 2 \cdot \mathrm{E}_1(\mathrm{M}_i, \mathrm{pk}_o) + \mathrm{E}_1(\mathrm{h}_i, \mathrm{pk}_o) = y^k \cdot (2\mathrm{M}_i + \mathrm{h}_i) \bmod p \tag{5}$$

where $\{\mathrm{M}_i\}_{i=1,2,\cdots,l} = M$, l denotes the length of embedded bits.

According to Eqs. (1)-(5), we can obtain the following one operation.

$$\mathrm{Opt}(\mathrm{M}_i, \mathrm{h}_i) = \begin{cases} \mathrm{E}_1(2\mathrm{M}_i + 1, \mathrm{pk}_o) & \text{if } \mathrm{h}_i = 1 \\ \mathrm{E}_1(2\mathrm{M}_i, \mathrm{pk}_o) & \text{if } \mathrm{h}_i = 0 \end{cases}, \tag{6}$$

For every encrypted pixel value, Eq. (6) are sequentially executed so that the same number of bit values as the pixels can be hidden into an encrypted image.

In this way, the bits to be embedded $\{0,1\}^l$ is carried out by the above method. The encrypted and embedded pixel C_h is obtained by the following operation.

$$\mathrm{C}_h = \mathrm{Opt}(\mathrm{M}, \mathrm{h}) = \mathrm{E}_1(2\mathrm{M{+}h}, \mathrm{pk}_o) \tag{7}$$

where $\mathrm{h} = \{0,1\}^l$.

3.2 Re-encryption

After a DU sends a request about the desired MD to the CSP, the CSP forwards this message to the DO. And then the DO computes a re-encryption key, $rekey_{ou}$, by re-encryption key generator, $\mathrm{ReKeygen}(\cdot,\cdot,\cdot)$, as follows.

$$rekey_{ou} = \mathrm{ReKeygen}(sk_o, pk_u) \tag{8}$$

Then, the DO sends the rekey, $rekey_{ou}$, to the CSP.

After receives the re-encryption key, the CSP re-encrypts the C_h by the re-encryption operation, $RE(\cdot)$, as following. The CSP obtains the re-encrypted C'_h.

$$C'_h = E_2(C_h, rekey_{ou}) \tag{9}$$

where $E_2(\cdot,\cdot)$ is the above-mentioned re-encryption function.

Therefore, the CSP obtains the re-encryption result, C'_h and stores it in the cloud servers.

3.3 Data extraction and image recovery

To recover the original pixel value, C'_h needs to be firstly decrypted. According to the re-encrypted model, the re-encrypted and embedded pixel can be one-time decrypted with DU's secret key as following.

$$\mathrm{D}(C'_h, \mathrm{sk}_u) = 2M_i + h_i \tag{10}$$

As $\mathrm{M}_i \in [0,255]$ for a grey-level pixel value, we know that $2\mathrm{M}_i + \mathrm{h}_i \in [0,511]$. So, the original values of M_i and h_i can be obtained by

$$\mathrm{M}_i = \left\lfloor \frac{\mathrm{D}(C_h', \mathrm{sk}_u)}{2} \right\rfloor = \left\lfloor \frac{2\mathrm{M}_i + \mathrm{h}_i}{2} \right\rfloor = \mathrm{M}_i + \left\lfloor \frac{\mathrm{h}_i}{2} \right\rfloor \tag{11}$$

where $\mathrm{h}_i \in \{0,1\}$, $\lfloor \ \rfloor$ is the floor function.

$$\mathrm{h}_i = \mathrm{D}(C_h', \mathrm{sk}_u) - 2\mathrm{M}_i \tag{12}$$

Therefore, the original pixel is obtained by Eq. (11). At the same time, the embedded data is extracted from $\mathrm{D}(C_h', \mathrm{sk}_u)$.

Based on the above analysis, a lossless reversible data hiding operation is carried on the re-encrypted data. The embedded data can be accurately extracted after decryption and the original image can be fully recovered. In the method, data embedding is also taken by CSP. In some scenarios, data embedding would be taken by special data hider. In this way, the method ensures multimedia data security without relying on the trustworthiness of cloud servers, but also transmits additional data in encryption domain for authentication and content annotation.

4 Reversible data hiding in encrypted domain

Different from the RDH for Homomorphic encryption proposed in Section 3, the algorithm to be presented in this section is applicable to the scenario where data extraction is required before image decryption. As an encrypted image data, the value is close to the randomness. Based on the principle, The LSB of every encrypted data is ‘0’ or ‘1’. Clearly, the probability of ‘0’ or ‘1’ is 1/2. Therefore, a RDH in encrypted domain is introduced based on the theory. By exploiting the randomness in EIGamal cryptosystem, both data embedding and extraction can be conducted in the encryption domain.

Firstly, the ciphertext is as follows.

$$C = \cdot E(m_1, pk_o) = y^{k_1} \cdot m_1 \bmod p \tag{13}$$

where Eq. (13) satisfies the ciphertext format in EIGamal cryptography.

4.1 Data embedding and data extraction

For each encrypted pixel, the data-hider selects a random integer k_1' and N in $\mathbb{Z}_p^*$, and calculates the value

$$C_h = y^{k_1'} \cdot E(m_1, pk_o) \cdot (1 + N \cdot p) = y^{k_1' + k_1} \cdot m_1 \cdot (1 + N \cdot p) \bmod p \tag{14}$$

Denote $\tilde{k}_1 = k_1' + k_1$, $C_h = y^{\tilde{k}_1} m_1 \cdot (1 + N \cdot p) \bmod p$.

To embed a bit value $h_i \in \{0,1\}$ into an encrypted data C_h, the LSB of the data, S_i , is obtained by the following equation.

$$\mathrm{S}_i = C_h \bmod 2 \tag{15}$$

If $\mathrm{S}_i \neq h_i$, C_h would be recomputed according to Eq. (13) until $\mathrm{S}_i = h_i$. When meet the requirement, an integer k_1' with N would be chosen. In the implementation, the encrypted value is multiplied by $(1 + N \cdot p)$ because

$$C_h = y^{k_1'} \cdot E(m_1, pk_o) \cdot (1 + N \cdot p) = y^{k_1'+k_1} \cdot m_1 \cdot (1 + N \cdot p) \bmod p = y^{k_1'+k_1} \cdot m_1 \bmod p \tag{16}$$

After receiving an encrypted image containing the additional data, if the receiver knows the data-hiding key, he/she may extract the embedded data from the following equation.

$$b' = C_h \bmod 2 \tag{17}$$

where b' denotes the extracted bit value.

In this case, the embedded bits are extracted from the ciphertext.

4.2 Re-encryption

After a DU sends a request about the desired MD to the CSP, the CSP forwards this message to the DO. And then the DO computes a re-encryption key, $rekey_{ou}$, by re-encryption key generator, $\mathrm{ReKeygen}(\cdot,\cdot,\cdot)$. The procedure is similar as Section 3.2. Then, the DO sends the rekey, $rekey_{ou}$, to the CSP.

After receives the re-encryption key, the CSP re-encrypts the C_h by the re-encryption operation, $RE(\cdot)$, as following. The CSP obtains the re-encrypted C_h' .

$$C_h' = E_2(C_h, rekey_{ou})$$

where $E_2(\cdot,\cdot)$ is the above-mentioned re-encryption function.

Therefore, the CSP obtains the re-encryption result, C_h' and stores it in the cloud servers.

4.3 Data decryption

If the receiver knows the private key of the used cryptosystem, he/she performs decryption to obtain the original plaintext image. Therefore, the decryption operation is as follows.

$$m_1 = D(C_h', sk_u)$$

In this case, the plaintext is obtained by decrypt the re-encrypted ciphertext. The embedded data can be accurately extracted before decryption and the original image can be fully recovered.

5 Combined data hiding scheme

As described in Section 3 and 4, reversible data hiding by homomorphic encryption and encryption domain are proposed. In both of the two schemes, the data embedding operations are performed in encrypted domain by the homomorphism. On the other hand, the data extraction procedures of the two schemes are very different. As with the former scheme, data extraction and image recovery must be performed in the plaintext domain. As with the latter scheme, data embedding does not affect the plaintext content and data extraction is performed in the encrypted domain. Therefore, we combine the two schemes to construct a new scheme in this section.

In the combined scheme, the image provider performs image encryption. When having the encrypted image, the cloud service provider (CSP) may embed the first part of additional data using the method described in Section 3. The method described in Section

4 is used to embed the second part of additional data. On the receiver side, the receiver firstly extracts the second part of additional data from the encrypted data. Then, after decryption with the data user's secret key, the DU extracts the first part of additional data and recovers the original plaintext image from the directly decrypted image as described in Section 3. The framework of the combined scheme is shown in Fig. 3.

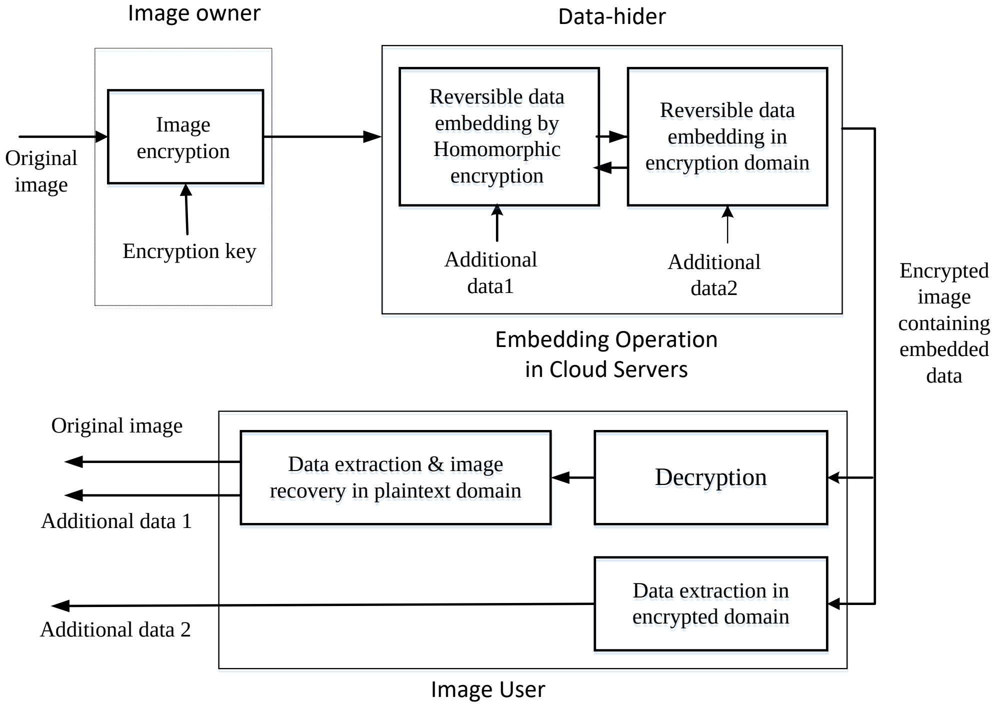
Image owner
Data-hider
Original image
Image encryption
Encryption key
Reversible data embedding by Homomorphic encryption
Reversible data embedding in encryption domain
Additional data1
Additional data2
Embedding Operation in Cloud Servers
Encrypted image containing embedded data
Original image
Additional data 1
Data extraction & image recovery in plaintext domain
Decryption
Data extraction in encrypted domain
Additional data 2
Image User

Figure 3: The framework of combined scheme

6 Analysis of proposed scheme

In this part, we give an example of a re-encryption algorithm based on EIGamal [EIGamal (1985)] in Correctness analysis. Meanwhile, the security analysis of the proposed scheme is also presented.

6.1 Correctness analysis

In our combined scheme, we first carry out the encryption phase. Given that the parameters (*y*, *g*, *p*) are the public key and $y = g^x \bmod p$, then the private key is *x*, and the plaintext is $\mathrm{M} = \{\mathrm{M}_i\}_n$. The DO chooses a random number x_o ($x_o \in \mathbb{Z}_p^*$) and computes $y_o = g^{x_o} \bmod p$. Therefore, $sk_o = x_o$, and $pk_o = (g, p, y_o)$. Similarly, the DU chooses a random number x_u ($x_u \in \mathbb{Z}_p^*$) and computes $y_u = g^{x_u} \bmod p$; therefore, $sk_u = x_u$,and $pk_u = (g, p, y_u)$.

The encryption phase. The DO chooses a random number k_1, k_2 ($k_1, k_2 \in \mathbb{Z}_p^*$) such that $Gcd(k_1, p-1) = 1$, $Gcd(k_2, p-1) = 1$.

Then, the DO will own the ciphertext C and computes

$$C = (a,b) = (g^{k_1} \bmod p, y_o^{k_1} \cdot M_i \bmod p)$$

where $a = g^{k_1} \bmod p$ and $b = y_o^{k_1} \cdot M_i \bmod p$.

The first data embedding phase.

Eq. (5) is redefined as follows.

$$\mathrm{E}_1(\mathrm{M}_i, pk_o) = y_o^{k_1} \cdot \mathrm{M}_i \bmod p \text{ and } \mathrm{E}_1(\mathrm{h}, pk_o) = y_o^{k_1} \cdot \mathrm{h} \bmod p$$

$$\mathrm{Opt}(\mathrm{M}_i, \mathrm{h}_i) = 2 \cdot \mathrm{E}_1(\mathrm{M}_i, \mathrm{pk}_o) + \mathrm{E}_1(\mathrm{h}_i, \mathrm{pk}_o) = y_o^{k_1} \cdot (2\mathrm{M}_i + \mathrm{h}_i) \bmod p \tag{18}$$

where the encrypted pixel is $C = (a, \mathrm{E}_1(\mathrm{M}_i, pk_o))$, the encrypted bit to be embedded is $C_b = (a, \mathrm{E}_1(\mathrm{h}_i, pk_o))$.

The re-encryption phase. The DO computes re-encryption key $rekey_{oi}$ by re-encryption key generator, $\mathrm{ReKeygen}(sk_o, pk_u) \to rekey_{ou}$.

Given that $\mathrm{rekey}_{ou} = \mathrm{ReKeygen}(sk_o, pk_u) = (pk_u)^{k_1} / (g^{k_1})^{sk_o} = (y_u)^{k_1} / g^{k_1 x_o} = g^{k_1 x_u} / g^{k_1 x_o} \bmod p$.

It can be established that the security of the re-encryption key generator depends upon the difficulty of a specific problem in a cyclic group related to computing discrete logarithms, which as shown in EIGamal [EIGamal (1985)].

The CSP re-encrypts the ciphertext C with the re-encryption key, $rekey_{ou}$. The CSP obtains re-encryption ciphertext by the function $E_2(\cdot,\cdot,\cdot)$.

$$\begin{aligned} C' = E_2(C, rekey_{ou}) &= (C \cdot \mathrm{g}^{k_1 x_u} \cdot \mathrm{g}^{-k_1 x_o}) \bmod p \\ &= (y_o^{k_1} \cdot (2\mathrm{M}_i + \mathrm{h}_i) \cdot \mathrm{g}^{k_1 x_u} \cdot \mathrm{g}^{-k_1 x_o}) \bmod p \\ &= (\mathrm{g}^{k_1 x_u} \cdot (2\mathrm{M}_i + \mathrm{h}_i)) \bmod p \end{aligned}$$

where $y_o = g^{x_o} \bmod p$.

The second data embedding phase. For each encrypted pixel, the data-hider selects a random integer k_1' in $\mathbb{Z}_p^*$, and calculates the value

$$C_h' = y_u^{k_1' + k_1} \cdot \mathrm{E}(2\mathrm{M}_i + h_i, pk_o) \cdot (1 + N \cdot p) = y_u^{k_1' + k_1} \cdot (2M_i + h_i) \cdot (1 + N \cdot p) \bmod p$$

$$= y_u^{\tilde{k}_1} \cdot (2M_i + h_i) \bmod p$$

Denote $\tilde{k}_1 = k_1' + k_1$, $(1 + N \cdot p)$ is a random code.

To embed a bit value $h_i \in \{0,1\}$ into an encrypted data c_h, the LSB of the data, S_i, is obtained by the following equation.

$$\mathrm{S}_i = C_h' \bmod 2 = h_i$$

The first data extraction phase. Each bit can be extracted from the encrypted and embedded data by the following equation.

$$b_i = C_h' \bmod 2$$

where b_i is the extracted bit.

The first part of embedding bits $\{b_i\}_i$ are extracted from the ciphertext.

The data decryption and extraction phase. The DU decrypts C'_h with his or her private key sk_u. Therefore, the decryption operation is shown as follows according to Eq. (10).

$$D(C'_h, sk_u) = C'_h \cdot (a^{x_u})^{-1} \bmod p = (2M_i + h_i) \cdot g^{\tilde{k}_i x_u} \cdot (g^{\tilde{k}_i x_u})^{-1} \bmod p = 2M_i + h_i$$

where $y_u = g^{x_u} \bmod p$.

Thus, the DU obtains $2M_i + h_i$ and correctness of re-encryption model is demonstrated.

As $\mathrm{M}_i \in [0,255]$ for a grey-level pixel value, we know that $2\mathrm{M}_i + \mathrm{h}_i \in [0,511]$. So, the original values of M_i and h_i can be obtained by

$$\mathrm{M}_i = \left\lfloor \frac{\mathrm{D}(C'_h, \mathrm{sk}_u)}{2} \right\rfloor = \left\lfloor \frac{2\mathrm{M}_i + \mathrm{h}_i}{2} \right\rfloor = \mathrm{M}_i + \left\lfloor \frac{\mathrm{h}_i}{2} \right\rfloor \tag{19}$$

where $\mathrm{h}_i \in \{0,1\}$, $\lfloor\ \rfloor$ is the floor function.

$$\mathrm{h}_i = \mathrm{D}(C'_h, \mathrm{sk}_u) - 2\mathrm{M}_i \tag{20}$$

Therefore, the original pixel is obtained by Eq. (19). At the same time, the second part of embedded data $\{h_i\}_i$ are extracted from $\mathrm{D}(C'_h, \mathrm{sk}_u)$.

Thus, the DU obtains the embedded plain-text. Therefore, the correctness of re-encryption model is demonstrated.

6.2 Security analysis

The security of proposed scheme relies critically on the security of fundamental algorithm (such as re-encryption) used in our construction process, and on the security of the proposed structure itself. In our proposed scheme, the fundamental algorithms are not restricted to specific algorithms.

The fundamental algorithm used in our scheme are mature and well-studied techniques and believed to be secure, if properly used. Therefore, the security of the proposed scheme is valid. The analysis supports the proposed scheme as a promising scheme for building the MD security.

***(1) Confidentiality of Pixel Values*:**

In the image encryption step, every pixel is encrypted. The interaction between DO and CSP are built on secure public cryptograph system and all the intermediate results are generated in the encrypted form. The CSP cannot learn anything but the final encrypted results. Therefore, we claim that the confidential of pixel values are well protected against the servers.

(2) Privacy of Image Content

In this aspect, we need to prove that the CSP are unable to deduce the original image by what it has obtained. As illustrated above, the CSP can get the following values: Encrypted pixel, re-encrypted key, DO's public key and DU's public key. Based on the secure of re-encrypted key, the above four values cannot deduce the DO's private key.

Although the CSP conclude with the DU, the DO's private key cannot be deduced. Therefore, the privacy of image content will be preserved against attacks of the theory.

7 Experimental results

Four gray images sized 512×512, Lena, Man, Plane and Crowd, were used as the original plaintext covers in the experiment. First, all pixels in the cover images were firstly encrypted using EIGamal cryptosystem, and then the first-part additional data were embedded into the ciphertext pixel-value using Homomorphic encryption as in Section 3.1.

All ciphertext pixels can be modified to carry the second-part additional data as in Section 4.1. On receiver side, the second-part embedded data can be extracted from the encrypted domain as in Section 4.2. After decryption, the original plaintext images can be full recovered and the first-part embedded data is also full extracted as in Section 3.3. In other word, when decryption was performed on the encrypted images containing additional data, the original plaintext images can be obtained.

A standard test image: Lena, with the size of 512×512, is shown in Fig. 4(a), and was used to demonstrate the feasibility of our proposed scheme. The following simulation experiments were performed in MATLAB. When the embedding rate is 1 bpp, we obtained a decrypted and embedded image as displayed in Fig. 4(b). The PSNR was employed to evaluate the quality of the marked decrypted image quantitatively. The PSNR of the marked decrypted image was 56.93 dB.

We compared our proposed scheme with other RM schemes in the encryption domain [Zhang (2011); Qian, Zhang and Wang (2014); Zhang, Ma and Yu (2014)]. For the sake of convenience, our simulation was based on 10 512×512 images and the results represent an average. Fig. 5 shows the average value of embedding rates compared with Zhang's scheme [Zhang (2011)], Qian et al. 's scheme [Qian, Zhang and Wang (2014)] and Zhang et al.'s scheme [Zhang, Ma and Yu (2014)]. In fact, the theory of embedding rate is up to very high. In the proposed scheme, the payload can also get larger if the memory space of encrypted data is enough.

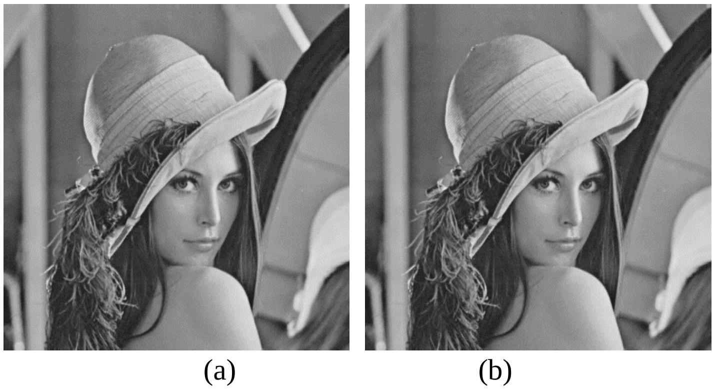
(a) (b)

Figure 4: (a) Original image, (b) Marked and decrypted image PSNR=56.93 dB with embedding rate 1 bpp

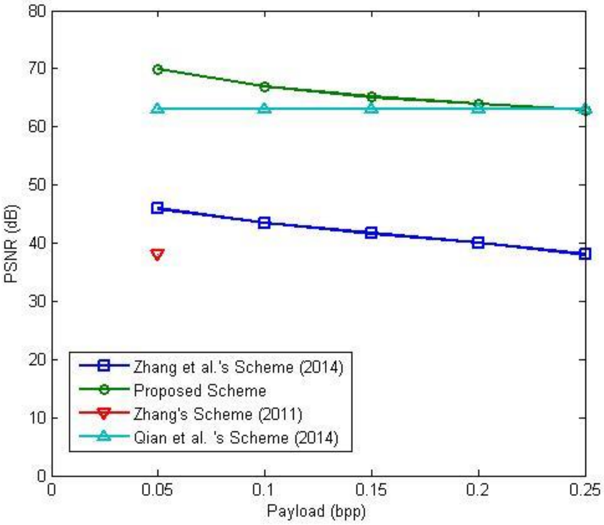

Figure 5: Comparison of rate-PSNR performance between the proposed reversible scheme and previous methods

Table 2: Feature comparison of related methods

Methods	Feature					
	Separable	Error in data extraction	Error in image recovery	Reserving room for data embedding	Encryption method	Re-encryption property
Proposed scheme	Yes	No	No	No	Public cryptographic	Yes
Zhang et al.'s method (2016)	Yes	No	Yes	Yes	Public cryptographic	No
Zhang's method (2011)	Yes	No	Yes	Yes	exclusive-or	No
Qian et al.'s method (2014)	No	No	No	No	exclusive-or	No
Zhang et al.'s method (2014)	Yes	No	No	Yes	AES	No

Feature comparison of related methods

The performance of the proposed algorithm was compared with that of Zhang [Zhang (2011); Qian, Zhang and Wang (2014); Zhang, Ma and Yu (2014); Zhang, Long and Wang (2016)], as shown in Tab. 2. It can be seen that the 'Error in data extraction' of the proposed algorithm is different from that of Zhang et al. [Zhang (2011); Zhang, Long and Wang (2016)]. At the same time, “Re-encryption property” of the proposed scheme is different from that of Zhang et al. [Zhang (2011); Qian, Zhang and Wang (2014); Zhang, Ma and Yu (2014); Zhang, Long and Wang (2016)], which is essential factor for data security in cloud computing. “Separable” indicates that data extraction can be arbitrarily performed before or after image decryption. As for the proposed two algorithms, the data extraction domain is fixed (one is in plaintext domain and the other is in Homomorphic encryption domain) so that the suitable one should be chosen according to the specific applications.

8 Discussion and conclusion

Advanced cloud computing technology provides cost saving and flexibility of services for users. With the explosion of multimedia data, more and more data owners would outsource their personal multimedia data on the cloud. Privacy-preserving is a hot topic in cloud computing recent years. Once users’ data are leak, the negative impact can be huge. Therefore, the development of privacy-preserving technology is the key factor for cloud computing. Reversible data hiding as a lossless data hiding technique aims to exactly recover both the embedded secret information and the original cover multimedia data. It is quite desirable and helpful in some sensitive applications such as remote sensing, multimedia archive management and military communication.

Therefore, the privacy-preserving outsourcing schemes of Reversible Data Hiding over Encrypted Image (RDHEI) in cloud computing are desired. In this paper, we design two novel secure RDHEI methods: Reversible data hiding by homomorphic encryption and reversible data hiding in encrypted domain. The embedded bits can be extracted before decryption or after decryption. The two methods can be also combined as a comprehensive scheme to protect the confidentiality of image data and transmit the additional data in cloud data services. The experimental results demonstrate that it can preserve much well the performance in terms of capacity and security as compared to the existing solutions. However, based on public key cryptosystem, some additional ciphertext data would be stored in cloud server, which take up extra storage. In future work, the scheme with secure, less storage space, and better performance will be studied.

Acknowledgement: This work was supported by the National Natural Science Foundation of China (No. 61702276), the Startup Foundation for Introducing Talent of Nanjing University of Information Science and Technology under Grant 2016r055 and the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions. The authors are grateful for the anonymous reviewers who made constructive comments and improvements.

References

Ateniese, G.; Fu, K.; Green, M.; Hohenberger, S. (2006): Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1-30.

Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. (2016): High capacity reversible data hiding in encrypted images by patch-level sparse representation. *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1132-1143.

EIGamal, T. (1985): A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans Information Theory*, no. 31, pp. 469-472.

Hong, W.; Chen, T.; Wu, H. (2012): An improved reversible data hiding in encrypted images using side match. *IEEE Signal Processing Letters*, vol. 19, no. 4, pp. 199-202.

Khelifi, F. (2018): On the security of a stream cipher in reversible data hiding schemes operating in the encrypted domain. *Signal Processing*, vol. 143, pp. 336-345.

Li, X.; Yang, B.; Zeng, T. (2011): Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3524-3533.

Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F. (2013): Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 553-562.

Ni, Z.; Shi, Y.; Ansari, N.; Su, W. (2006): Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 354-362.

Qian, Z.; Zhang, X.; Wang, S. (2014): Reversible data hiding in encrypted JPEG bitstream. *IEEE Transactions on Multimedia*, vol. 16, no. 5, pp. 1486-1491.

Sachnev, V.; Kim, H. J.; Nam, J.; Suresh, S.; Shi, Y. Q. (2009): Reversible watermarking algorithm using sorting and prediction. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 989-999.

Sebe, F.; Domingo-Ferrer, J.; Martinez-Balleste, A.; Deswarte, Y.; Quisquater, J. J. (2008): Efficient remote data possession checking in critical information infrastructures. *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1034-1038.

Tian, J. (2003): Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896.

Vimercati, S. D. C. D.; Foresti, S.; Jajodia, S.; Paraboschi, S.; Samarati, P. (2007): Over-encryption: Management of access control evolution on outsourced data. *Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 123-134.

Xuan, G.; Yao, Q.; Yang, C.; Gao, J.; Chai, P. et al. (2006): Lossless data hiding using histogram shifting method based on integer wavelets. *Digital Watermarking*, pp. 323-332.

Xiong, L.; Xu, Z.; Shi, Y. (2018): An integer wavelet transform based scheme for reversible data hiding in encrypted images. *Multidimensional Systems and Signal Processing*, vol. 29, no. 3, pp. 1191-1202.

Xiong, L.; Xu, Z.; Xu, Y. (2015): A secure re-encryption scheme for data services in a cloud computing environment. *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 4573-4585.

Zhang, W.; Ma, K.; Yu, N. (2014): Reversibility improved data hiding in encrypted images. *Signal Processing*, vol. 94, pp. 118-127.

Zhang, X. (2011): Reversible data hiding in encrypted image. *IEEE Signal Processing*

Letters, vol. 18, no. 4, pp. 255-258.

Zhang, X.; Long, J.; Wang, Z. (2016): Lossless and reversible data hiding in encrypted images with public-key cryptography. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1622-1631.

Zhang, X.; Qian, Z.; Feng, G.; Ren, Y. (2014): Efficient reversible data hiding in encrypted images. *Journal of Visual Communication and Image Representation*, vol. 25, no. 2, pp. 322-328.

Zhu, W.; Luo, C.; Wang, J.; Li, S. (2011): Multimedia cloud computing. *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 59-69.