High-capacity and Lossless Reversible Data Hiding for Encrypted Multimedia Data in Cloud Computing

Lizhi Xiong[1,2], Zhengquan Xu[3]

[1] School of Computer and Software, Nanjing University of Information Science and Technology, China
[2] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, China
[3] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China
lzxiong16@163.com, xuzq@whu.edu.cn

Abstract

Multimedia data sharing services are attracting more and more attention. Advanced cloud computing technology provides huge storage and efficient support for multimedia big data services. Multimedia cloud services become a mainly development trend. In this paper, a high-capacity and lossless reversible data hiding scheme is proposed for encrypted multimedia data in cloud computing. The proposed scheme not only ensures multimedia data security without relying on the trustworthiness of cloud servers, but also transmits additional data in encryption domain by combining re-encryption and reversible data hiding techniques. 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. Experimental results demonstrate that the proposed scheme can perform better than existing methods.

Keywords: Multimedia data security, Cloud computing, Re-encryption, Reversible data hiding

1 Introduction

With the development of multimedia data processing and network technologies, the use of multimedia data is becoming more and more widespread. Many diffusion and distribution tools are opening the way towards new services and applications for multimedia data, including mobile devices. As we have entered the era of "big data", the capability of multimedia data has dramatically increased and reached an unprecedented level [1]. Currently, few enterprises or individuals can afford to store the increasing mass of multimedia data. Cloud computing as an advanced technique can provide huge storage space and on-demand access service, thus becoming a research platform for those multimedia big data. Therefore, users and enterprises with various capabilities as Data Owners (DOs), can store and process their multimedia data in third-party data centers and then authorize Data Users (DUs) can acquire desired data from the cloud [2]. However, this new paradigm introduces new security challenges.

Multimedia data storage security is a critical issue when DOs outsource their data to a third party such as cloud computing servers. In this situation, the cloud servers possess the outsourced data; it is possible that a cloud service provider (CSP) can duplicate DO data while claiming that the data are still stored confidentially in the cloud. Thus, DOs need to be convinced that their data is secure and has not been exposed or stolen from the cloud. A feasible solution is data encryption using certain cryptographic primitives, with disclosure of the decryption keys only to authorized users. However, 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 [3].

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. [4] 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. [5] 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

*Corresponding Author: Lizhi Xiong; E-mail: lzxiong16@163.com
DOI: 10.3966/160792642018081904021

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.

However, in some scenarios, a data owner does not trust the cloud service provider but would like to make use of the cloud computing capability to process encrypted data in cloud servers for authentication and content annotation, such as medical image. Therefore, the ability to manipulate the encrypted data when keeping the plain content unrevealed is desired. Reversible data Hiding (RDH) has been proposed and a lot RDH methods have been developed to transmit additional data, such as [6-10], including coverless information hiding method [18-19]. Recently, RDH in encrypted images provide a solution to solve the above issue, such as [11-16]. In those schemes, additional data can be hidden into encrypted images and correctly extracted when need, or the original plaintext image can be obtained after decryption and data extraction.

Therefore, re-encryption technique ensures MD security in cloud computing. RDH in encrypted images transmits additional data in encryption domain to process authentication and content annotation. In our case, the combination of the two techniques provides the comprehensive protection for multimedia data storage and transmission in cloud computing. There are plenty of literature delineating RDH schemes for an encrypted domain [11-16]. But these methods are not applicable to images where an insufficient number of pixels can be selected to form vacating room, such as a smoothness image. What's more, these methods cannot be competent for cloud computing environment. This paper therefore proposes a combining re-encryption and reversible data hiding method for multimedia data in cloud computing that can not only ensure multimedia data security without relying on the trustworthiness of cloud servers, but also transmits additional data in encryption domain.

The rest of this paper is organized as follows. The scheme of the proposed method is described in Section 2. An analysis of the proposed scheme is presented in Section 3. Experimental results are provided in Section 4. We conclude this paper with a discussion of the results in Section 5.

2 Proposed Scheme

In this section, we introduce an encryption method for cloud computing and present data pre-processing and encryption operations. First, we describe how additional data is embedded into the encrypted carrier and how the encrypted and watermarked carrier is re-encrypted when needs. We discuss the means by which the encrypted and watermarked carrier is decrypted. Finally, the embedded information is extracted from the watermarked carrier, and the watermarked carrier is recovered to the original cover.

2.1 Encryption Model in Cloud Computing

A cloud storage service allows data owners to outsource their data to cloud servers for storage and maintenance an attractive option since the service has low hardware and software capital costs, low management and maintenance overhead, and universal on-demand data access. Data owners however, 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 Figure 1. The framework can be described as follows.

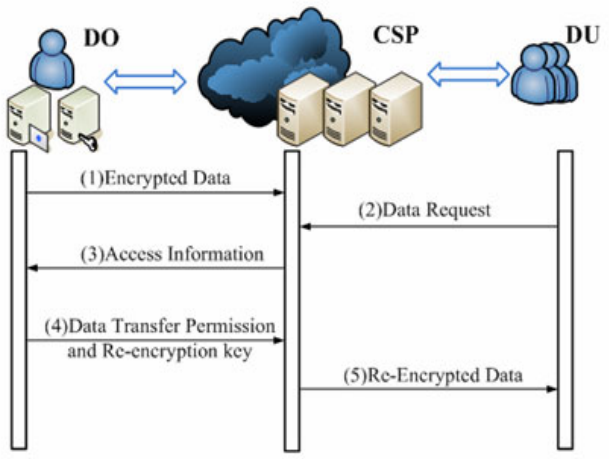


Figure 1. The framework of re-encryption model in cloud computing

First, a DO generates secure private-public key parings using the asymmetric cryptographic algorithm. The DO encrypts his/her own data with the public key and obtains a ciphertext data. Then, the ciphertext data is uploaded to cloud servers.

Second, a DU sends a request for the desired MD 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, the DO generates a re-encryption key, rekey, with the DO's private key and access information, then the DO sends the rekey to the CSP.

Fourth, after receiving the rekey, the CSP obtains the re-encrypted ciphertext by re-encrypting the ciphertext data with the rekey.

Fifth, the DU downloads the re-encrypted ciphertext from the cloud servers. The DU decrypts the re-encrypted ciphertext with DU's private key.

In the proposed model, the confidentiality of the DO's 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 Table 1.

Table 1. The syntactic definitions of re-encryption

Notations	Description
par	a public parameters
(sk_u, pk_u)	DU's secret/public key pair
(sk_o, pk_o)	DO's secret/public key pair
$rekey_{ou}$	a re-encryption key
M	a plaintext
C	a first level ciphertext
C'	a re-encrypted ciphertext
Rekeygen(par, sk_o, pk_u)	a re-encryption key generator
E_1(M, pk_o)	an encryption function
D_1(C, sk_o)	a decryption function for C
E_2(C, $rekey_{ou}$)	a re-encryption function
D(C', sk_u)	a decryption function for re-encrypted ciphertext

2.2 Encryption Method Based on EIGamal Cryptosystem

The EIGamal cryptosystem is a public-key cryptosystem, which is based on discrete logarithms problem [17]. The details are shown in Figure 2. After encrypting a plain-text digital image, a string of big integers are 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, 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

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

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

$$\begin{aligned} \mathrm{Opt}(m_1, m_2) &= \mathrm{E}(m_1, \mathrm{pk}_o) + \mathrm{E}(m_2, \mathrm{pk}_o) \\ &= y^k \cdot (m_1 + m_2) \bmod p \end{aligned} \quad \textbf{(3)}$$

where Opt is an operation between the two encrypted values.

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

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

where m' is a decrypted data.

We can see that the encryption operation satisfies the property of the homomorphism.

In this case, the plain-text M is firstly encrypted in the re-encryption model. The ciphertext is obtained, E_1(M, pk_o).

2.3 Data Embedding Method

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

$$\begin{aligned} \mathrm{Opt}(\mathrm{M}_i, \mathrm{h}_i) &= 2 \cdot \mathrm{E}_1(\mathrm{M}_i, \mathrm{pk}_o) + \mathrm{E}_1(\mathrm{h}_i, \mathrm{pk}_o) \\ &= y^k \cdot (2\,\mathrm{M}_i + \mathrm{h}_i) \bmod p \end{aligned} \quad \textbf{(5)}$$

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.

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

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.

$$C_h = \mathrm{Opt}(M, h) = E_1(2M+h, pk_o) \quad (7)$$

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

2.4 Re-encryption Method

After a DU sends a request about the desired multimedia data 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.

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

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'.

$$C_h' = E_2(C_h, rekey_{ou}) \quad (9)$$

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.

2.5 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.

$$D(C_h', sk_u) = \{2M_i + h_i\}_{i=1,2,\cdots,l} \quad (10)$$

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

$$M_i = \left\lfloor \frac{D(C_h', sk_u)}{2} \right\rfloor = \left\lfloor \frac{2M_i + h_i}{2} \right\rfloor = M_i + \left\lfloor \frac{h_i}{2} \right\rfloor \quad (11)$$

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

$$h_i = D(C_h', sk_u) - 2M_i \quad (12)$$

Therefore, the original pixel is obtain by Eq. (11). At the same time, the embedded data is extracted from $D(C_h', sk_u)$.

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 and the original image can be fully recovered. The proposed scheme provides a RDH for encrypted multimedia data in cloud computing, which 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.

3 Analysis of Proposed Scheme

In Section 2.2, we construct the proposed scheme where we assumed the existence of appropriate re-encryption techniques. In this part, we give an example of a re-encryption algorithm based on EIGamal [17] in Correctness analysis. Meanwhile, the security analysis of the proposed scheme, the computational analysis of re-encryption model, and embedding capacity analysis of the proposed scheme are presented.

3.1 Correctness Analysis

In our proposed 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 *M*. 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 $pk_u = (g, p, y_u)$.

In Section 2.2, 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* by computing the following operation.

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

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

Therefore, Eq. (5) is redefined as follows.

$$\begin{aligned} &E_1(M, pk_o) = y_o^{k_1} \cdot M \bmod p \text{ and } E_1(h, pk_o) = y_o^{k_1} \cdot h \bmod p \\ &\mathrm{Opt}(M_i, h_i) = 2 \cdot E_1(M_i, pk_o) + E_1(h_i, pk_o) \\ &\qquad = y_o^{k_1} \cdot (2M_i + h_i) \bmod p \end{aligned} \quad (13)$$

where the encrypted pixel is $C = (a, E_1(M, pk_o))$, the encrypted bit to be embedded is $C_b = (a, E_1(h, pk_o))$.

In Section 2.4, the DO computes re-encryption key $rekey_{ou}$ by re-encryption key generator, ReKeygen $(sk_o, pk_u) \to rekey_{ou}$.

Since $rekey_{ou} = \mathrm{ReKeygen}(sk_o, pk_u) = pk_u^{1/sk_o} = (g, p, y_u^{1/sk_o}) = (g, p, (g^{x_u})^{1/x_o} \bmod p)$, then it can be established that the security of the re-encryption key generator depends upon the difficulty of a given problem in a cyclic group related to computing discrete logarithms, which was solved in [16]. This equation is an instance of Eq. (8).

Thus, the CSP re-encrypts the ciphertext *C* with the

re-encryption key, $rekey_{ou}$. The CSP obtains re-encryption ciphertext by the function $\mathrm{E}_2(\cdot,\cdot)$.

$$C' = (a,c,d) = (g^{k_1} \bmod p, g^{k_2} \bmod p, \mathrm{E}_2(b, rekey_{oi}))$$
$$d = \mathrm{E}_2(b, rekey_{ou}) = e(y_o^{k_1}, rekey_{ou}) \cdot e(g, g^{k_2}) \cdot (2\mathrm{M}+\mathrm{h}) \bmod p$$
$$= e(g^{k_1 x_o}, g^{x_u / x_o}) \cdot e(g,g)^{k_2} \cdot (2\mathrm{M}+\mathrm{h}) \bmod p$$
$$= e(g,g)^{k_1 x_u} \cdot e(g,g)^{k_2} \cdot (2\mathrm{M}+\mathrm{h}) \bmod p$$

where $e(\cdot,\cdot)$ is a bilinear map function.

Thus, $C' = (a,c,d)$

$$= (g^{k_1} \bmod p, g^{k_2} \bmod p, e(g,g)^{k_1 x_u} \cdot e(g,g)^{k_2} \cdot (2\mathrm{M}+\mathrm{h}) \bmod p)$$

Therefore, Eq. (9) is redefined as follows.

$$\mathrm{C}'_h = \mathrm{E}_2(\mathrm{E}_1(2\mathrm{M}+\mathrm{h}), rekey_{ou}) = \mathrm{d} \qquad \textbf{(14)}$$

In Section 2.5, the DU decrypts C' with his private key sk_u. Therefore, the decryption operations are shown as follows.

$$\mathrm{D}(C', sk_u) = \mathrm{d} \cdot (e(g, \mathrm{a}^{x_u}) \cdot e(g,\mathrm{c}))^{-1}$$
$$= (e(g,g)^{k_1 x_u} \cdot e(g,g)^{k_2})^{-1} \cdot d \bmod p$$
$$= (2\mathrm{M}+\mathrm{h}) \cdot e(g,g)^{k_1 x_u} \cdot e(g,g)^{k_2} \cdot \frac{1}{e(g,g)^{k_1 x_u} \cdot e(g,g)^{k_2}} \bmod p$$
$$= (2\mathrm{M}+\mathrm{h})$$

Thus, the DU obtains the embedded plaintext $(2\mathrm{M}+\mathrm{h})$. Therefore, the correctness of re-encryption model is demonstrated.

3.2 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.

3.3 Computation Analysis

In the proposed scheme, re-encryption model consists of the following phases: encryption, re-encryption key generation, re-encryption and decryption. The computational cost of the encryption and decryption operations is the same as in the encryption and decryption of an applied asymmetric cryptographic algorithm, such as EIGamal. Only the re-encryption key generation and re-encryption operation will cause additional computation cost.

In the cryptology, there are following definitions about computation analysis. Denote $\mathrm{t_p}$ as modular exponentiation running time, and $\mathrm{t_e}$ as bilinear pairing running time.

According to correctness analysis, the computation cost of re-encryption key generation is $\mathrm{t_p}$, which is half of that of the encryption of EIGamal. The additional computation is acceptable for the DO. As for the re-encryption operation, the computation cost is $3 \cdot \mathrm{t_p} + 2 \cdot \mathrm{t_e}$. However, in our proposed scheme, the re-encryption task is delegated to the CSP with the powerful processing capability. The computation can be achieved by cloud servers.

In traditional data services without cloud computing, if there are n-time requests to a DO, the computation cost of the DO is $2n \cdot \mathrm{t_p}$. However, in the proposed scheme, the cost is $(2+\mathrm{n}) \cdot \mathrm{t_p}$. Therefore, our proposed scheme is more effective than traditional methods.

3.4 Embedding Capacity Analysis

Based on the embedding method in Section 2.3, we know that every grey-level value from 0 to 255 is used to hide bit values, respectively. Therefore, the capacity is up to 1 bpp. It is not difficult to find that every pixel can be embedded multiple times. In the Eq. (5), the encrypted pixel is multiplied by 2. If multiplied by 2 again, the multiple embedding operations are as follows.

$$\mathrm{Opt}'(\mathrm{M}_i, \mathrm{h}_i) = 2 \cdot (2 \cdot \mathrm{E}_1(\mathrm{M}_i, \mathrm{pk}_o) + \mathrm{E}_1(\mathrm{h}_i, \mathrm{pk}_o)) + \mathrm{E}_1(\mathrm{h}_j, \mathrm{pk}_o)$$
$$= \mathrm{E}_1(4\mathrm{M}_i + 2\mathrm{h}_i + \mathrm{h}_j, \mathrm{pk}_o)$$

where h_j is another bit to be embedded.

According to Eqs.(11) and (12), the extraction and recovery operation are follows.

$$\mathrm{M}_i = \left\lfloor \frac{4M_i + 2\mathrm{h}_i + \mathrm{h}_j}{4} \right\rfloor = \mathrm{M}_i + \left\lfloor \frac{2\mathrm{h}_i + \mathrm{h}_j}{4} \right\rfloor \qquad \textbf{(15)}$$

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

$$\mathrm{h}_i = \left\lfloor \frac{\mathrm{D}(C'_h, \mathrm{sk}_u) - 4\mathrm{M}_i}{2} \right\rfloor = \mathrm{h}_i + \left\lfloor \frac{\mathrm{h}_j}{2} \right\rfloor \qquad \textbf{(16)}$$

$$\mathrm{h}_j = \mathrm{D}(C'_h, \mathrm{sk}_u) - 4\mathrm{M}_i - 2\mathrm{h}_i \qquad \textbf{(17)}$$

where $\mathrm{D}(C'_h, \mathrm{sk}_u)$ is a decrypted and embedded pixel.

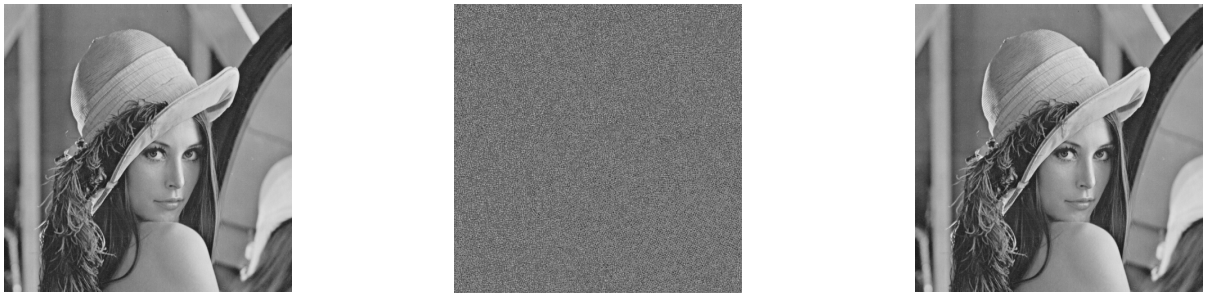
In this way, the capacity is up to 2 bpp. But a larger memory space is need to store a decrypted value. Therefore, the capacity of the proposed scheme follows "pay as you go". Compared with the existing RDH methods for the encrypted images, the proposed scheme has a higher capacity.

4 Experimental Results

We compared our proposed scheme with other RDH schemes in the encryption domain [14-15]. For the sake of convenience, our simulation was based on 10 512 × 512 images and the results represent an average.

A standard test image: Lena, with the size of 512× 512, is shown in Figure 3(a), and was used to demonstrate the feasibility of our proposed scheme. The following simulation experiments were performed in MATLAB. In the data encryption phase, it is known that a larger size private key can resist more crack attacks. To facilitate the encryption computation, we choose the following values for related parameters: the size of large prime p was 1024 bits, the size of prime order g is 1024 bits, the size of private key x was 1024 bits, and the size of the random number k was 1024 bits. In this way, the storage cost was 1024bits × 256=262144bits.

When the embedding rate is 1 bpp and 2 bytes is used to store a decrypted value, we obtained an encrypted and embedded image as displayed in Figure 3(b). Figure 3(c) is the decrypted image containing hidden data (watermarked and decrypted). The objective criteria: PSNR, was employed to evaluate the quality of the watermarked and decrypted image quantitatively. The PSNR of the watermarked and decrypted image was 56.93 dB, which is bigger than [14-15].



(a) Original image (b) Encrypted image (c) Watermarked and decrypted image

Figure 3.

In order to better compare the proposed method with other work [14-15], we tested ten images. The comparative results of the proposed method with that in [14] and that in [15] are shown in Table 2. The gain in term of PSNR is significantly higher at the embedding rate range that the methods in [14-15] could achieve.

Table 2. Average performance comparison of the proposed method, [14]'s and [15]'s on 10 images of database

ER (bpp)	PSNR of Proposed	PSNR of Zhang *et al.*'s [15]	PSNR of Zhang's [14]
0.01	75.63	61.35	41.04
0.02	72.93	58.31	38.94
0.03	70.07	56.54	38.49
0.04	68.82	55.34	38.05

5 Discussion and Conclusion

Multimedia Data (MD) is classical example of big data, its services and applications are becoming more and more widely valued and used in a variety of domains. Cloud computing technology provides a platform for multimedia big data services which can support the storage of huge volume of MD and efficient on-demand access. However, to the best of our knowledge, there has been no report about MD's storage and usage security in cloud computing. The re-encryption method ensures MD security in cloud computing and reversible data hiding technique protect MD's copyright. This paper proposes a new method combined re-encryption and reversible data hiding techniques to protect the security of storage and usage in the cloud computing environment.

Our experiments demonstrate that the proposed scheme can perform better than the existing methods in the capacity. Theoretical analysis have been presented to verify the correctness of encryption model and justify the security of the proposed scheme. The computation cost of the proposed scheme is acceptable and adjustable according the different security level.

Acknowledgements

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

[1] W. Zhu, C. Luo, J. Wang, S. Li, Multimedia Cloud Computing, *IEEE Signal Processing Magazine*, Vol. 28, No. 3, pp. 59-69, May, 2011.

[2] F. Sebe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, J. J. Quisquater, Efficient Remote Data Possession Checking in Critical Information Infrastructures, *IEEE Transactions on Knowledge and Data Engineering,* Vol. 20, No. 8, pp. 1034-1038, August, 2008.

[3] L. Xiong, Z. Xu, Y. Xu, 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, December, 2015.

[4] G. Ateniese, K. Fu, M. Green, S. Hohenberger, Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage, *ACM Transactions on Information and System Security (TISSEC),* Vol. 9, No. 1, pp. 1-30, February, 2006.

[5] S. D. C. D. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, P. Samarati, Over-encryption: Management of Access Control Evolution on Outsourced Data, *VLDB*' 07 *Proceedings of the 33rd International Conference on Very Large Data Bases*, Vienna, Austria, 2007, pp. 123-134.

[6] Z. C. Ni, Y. Q. Shi, N. Ansari, W. Su, Reversible Data Hiding, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 16, No. 3, pp. 354-362, March, 2006.

[7] V. Sachnev, H. J. Kim, J. Nam, S. Suresh, Y. Q. Shi, Reversible Watermarking Algorithm Using Sorting and Prediction, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 19, No. 7, pp. 989-999, July, 2009.

[8] X. L. Li, B. Yang, T. Y. Zeng, Efficient Reversible Watermarking Based on Adaptive Prediction-Error Expansion and Pixel Selection, *IEEE Transactions on Image Processing*, Vol. 20, No. 12, pp. 3524-3533, December, 2011.

[9] G. Xuan, Q. Yao, C. Yang, J. Gao, P. Chai, Y. Q. Shi, Z. Ni, Lossless Data Hiding Using Histogram Shifting Method Based on Integer Wavelets, *IWDW' 06 Proceedings of the 5th International Workshop on Digital Watermarking,* Jeju Island, Korea, 2006, pp. 323-332.

[10] J. Tian, Reversible Data Embedding Using a Difference Expansion, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 13, No. 8, pp. 890-896, August, 2003.

[11] L. Xiong, Z. Xu, Y.-Q. Shi, 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, July, 2018.

[12] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. Au, Y. Y. Tang, Secure Reversible Image Data Hiding over Encrypted Domain via Key Modulation, *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 26, No. 3, pp. 441-452, March, 2016.

[13] X. Zhang, J. Long, Z. Wang, H. Cheng, 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, September, 2016.

[14] X. P. Zhang, Separable Reversible Data Hiding in Encrypted Image, *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 2, pp. 826-832, April, 2012.

[15] W. M. Zhang, K. D. Ma, N. H. Yu, Reversibility Improved Data Hiding in Encrypted Images, *Signal Processing*, Vol. 94, pp. 118-127, January, 2014.

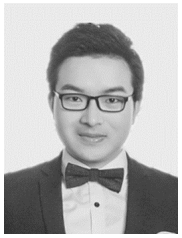
[16] X. P. Zhang, Reversible Data Hiding in Encrypted Image, *IEEE Signal Processing Letters*, Vol. 18, No. 4, pp. 255-258, April, 2011.

[17] T. Elgamal, A Public-key Cryptosystem and A Signature Scheme based on Discrete Logarithms, *IEEE Transactions on Information Theory*, Vol. 31, No. 4, pp. 469-472, July, 1985.

[18] X. Chen, S. Chen, Y. Wu, Coverless Information Hiding Method Based on the Chinese Character Encoding, *Journal of Internet Technology*, Vol. 18, No. 2, pp. 313-320, March, 2017.

[19] J. Zhang, L. Wang, H. Lin, Coverless Text Information Hiding Method Based on the Rank Map, *Journal of Internet Technology*, Vol. 18, No. 2, pp. 427-434, March, 2017.

Biographies



Lizhi Xiong is currently a Assistant Professor in the School of Computer and Software at Nanjing University of Information Science and Technology, Nanjing, China. He received his Ph.D. degree in Communication and Information System from Wuhan University, Wuhan, China, in 2016. His research interests include digital multimedia protection and cloud security.

Zhengquan Xu is a Professor with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China. He received his Ph.D. degree in Biomedicine Engineering from Hong Kong Polytechnic University, Kowloon, Hong Kong, in 1998. His research interests include digital watermarking.