International Journal of Digital Crime and Forensics
Volume 13 • Issue 4 • July-August 2021

Lossless Data Hiding in LWE-Encrypted Domains Based on Key-Switching

Ting-ting Su, College of Cryptography Engineering, Engineering University of PAP, Shanxi, China
Yan Ke, Engineering University of PAP, Shanxi, China
https://orcid.org/0000-0002-6229-9998
Yi Ding, College of Cryptography Engineering, Engineering University of PAP, Shanxi, China
Jia Liu, College of Cryptography Engineering, Engineering University of PAP, Shanxi, China

ABSTRACT

This paper proposes a lossless data hiding scheme in learning with errors (LWE)-encrypted domain based on key-switching technique. Lossless data hiding and extraction could be realized by a third party without knowing the private key for decryption. Key-switching-based least-significant-bit (KSLSB) data hiding method has been designed during the lossless data hiding process. The owner of the plaintext first encrypts the plaintext by using LWE encryption and uploads ciphertext to a (trusted or untrusted) third server. Then the server performs KSLSB to obtain a marked ciphertext. To enable the third party to manage ciphertext flexibly and keep the plaintext secret, the embedded data can be extracted from the marked ciphertext without using the private key of LWE encryption in the proposed scheme. Experimental results demonstrate that data hiding would not compromise the security of LWE encryption, and the embedding rate is 1 bit per bit of plaintext without introducing any loss into the directly decrypted result.

KEYWORDS

Information Security, Key Switching, LWE, Reversible Data Hiding in Encrypted Domain

INTRODUCTION

Reversible data hiding in encrypted domain (RDH-ED) is an information hiding technique that aims to not only accurately embed and extract the additional messages in the ciphertext, but also restore the original plaintext losslessly (Ma et al., 2013)(Shi et al., 2016). RDH-ED is useful in some distortion intolerable applications, such as ciphertext management or retrieval in the cloud, ciphertext annotation for medical or military use. With the increasing demand for information security and the development of the encrypted signal processing techniques, RDH-ED has been an issue of great attention in the field of privacy protection and ciphertext processing.

From the viewpoint of the cryptosystem that RDH-ED methods are based on, existing RDH-ED methods could be classified into two categories: Symmetric encryption based RDH-ED (Ma et al., 2013), (Zhang 2011; Zhou 2016; Wu & Sun 2014; Qian et al., 2014; Puech et al., 2008; Zhang et al., 2014; Li et al., 2015; Cao et al., 2016; Zhang 2012; Wu et al., 2017;Puteaux & Puech 2018; Huang et al., 2016), and public key encryption based RDH-ED. Symmetric cryptography that has been introduced into RDH-ED includes stream encryption (Ma et al., 2013), (Qian et al., 2014; Wu &

DOI: 10.4018/IJDCF.20210701.oa6

Sun, 2014; Zhang, 2011; Zhou et al., 2016), (Puteaux & Puech, 2018), advanced encryption standard (AES) (Puech et al., 2008), (Zhang et al., 2014), and RC4 encryption (Li et al., 2015).

According to the methods of utilizing the redundancy in the cover for data hiding, symmetric encryption based RDH-ED methods were classified into two categories (Ma et al., 2013)(Shi et al., 2016): "vacating room before encryption (VRBE)" (Ma et al., 2013)(Puech et al., 2008)(Zhang et al., 2014)(Cao et al., 2016)(Puteaux & Puech, 2018) and "vacating room after encryption (VRAE)"(Qian et al., 2014; Wu & Sun, 2014; Zhang, 2011; Zhou et al., 2016). The room, namely the redundancy in the cover, is vacated for reversible data hiding. The first RDH-ED method was proposed by Zhang for encrypted images (Zhang, 2011), and then (Wu & Sun, 2014; Zhou et al., 2016) enhanced its capacity. Qian *et al.* proposed a similar method to embed data in an encrypted JPEG bit stream (Qian et al., 2014). AES was introduced in (Puech et al., 2008) to encrypt the cover image. Each block containing n pixels could carry one bit data. The embedding rate (ER) is $1/n$ bits per pixel (bpp). Then difference prediction was introduced before encryption in (Zhang et al., 2014), and AES was used to encrypt pixels except the embedding ones, thus resulting in a better embedding capacity (EC) and reversibility. However, it needed decryption first before data extraction in the above RDH-ED methods, which restricted the practicability in practical applications. The separable RDH-ED was proposed in (Zhang, 2012)(Wu et al., 2017). Separability has been so far an important attribute of practicality for current RDH-ED.

The redundancy introduced by VABE or VARE is independent from the encryption, resulting in the mutual restriction between decryption distortion and the embedding capacity, which is a major obstacle to the realization of separability and a high EC. There existed two main solutions proposed: one is to improve the quality of redundancy introduced before encryption. For example in (Puteaux & Puech, 2018), a separable high embedding algorithm was proposed by making full use of prediction error introduced before encryption. Second, the correlation of the plaintext is preserved in the ciphertext, so that RDH in spatial domain, such as difference expansion technique (DE) (Tian, 2003), histogram shifting technique (HS) (Li et al., 2011; Ni et al., 2006; Ou et al., 2013), could be implemented in the encrypted domain. For example in (Huang et al., 2016), a new framework of RDH-ED was proposed, in which a specific stream cipher was used to preserve the correlation between the neighboring pixels. The above mentioned symmetric encryption based algorithms are fast and efficient in practice, which has significant research value and technological potential in the future.

However, there are also technical defects in symmetric encryption based RDH-ED. The correlation of plaintext would be destroyed because of the *confusion* and *diffusion* principles of symmetric encryption. To achieve reversible data hiding, it usually needs to introduce embedding redundancy. While it is difficult to vacate room after encryption, the current attention focuses more on the VEBE methods (Puteaux & Puech, 2018), by which more computational expense is introduced into the client end for data hiding. The preprocessing in the plaintext is similar to data compression, and the compression capability determines the performance of RDH-ED. As for the methods of preserving plaintext correlation by a specific encryption (Huang et al., 2016), it currently mainly relies on reusing the same random sequence to encrypt a specific pixel block. It could provide certain security guarantees, but key reusing would weaken the encryption intensity of the symmetric encryption in theory. The more correlation among ciphertext is remained, the more the encryption intensity is reduced. The RC4 encryption was declared breached in 2013 (AlFardan et al., 2013), RDH-ED based on early RC4 has certain limitations in future security applications. In addition, symmetric encryption requires a geometrically increasing amount of encryption keys with the number of communication participants. The local key storage cost is high for each user.

Compared with symmetric encryption, public key encryption has some advantages for RDH-ED, which is worthy of our attention: first, public key encryption requires a linear increasing amount of key usage in the communication network. The local key storage cost is only the private key of the user's own, while all the public keys are publicly released. It has been widely used in electronic finance and network communication protocols, which provides application prospects for RDH-ED. Second,

public key encryption introduces ciphertext extension, namely, the redundancy from the ciphertext itself. Through a certain embedding strategy (Ke et al., 2018), we could select embedding positions and improve EC effectively. Third, flexible cryptosystems of the public key encryption, especially the homomorphic encryption, provide reliable technical supports for RDH-ED. However, there are still technical limitations and application dilemmas in public key based RDH-ED. We shall discuss those in Section II. This paper focuses on the current state of public key based RDH-ED, aiming at making full use of LWE-based fully homomorphic encryption (FHE) technique to implement DE encapsulation. A novel RDH-ED method is proposed, which is superior to the current public key based RDH-ED in practicality, security and reversibility.

The rest of this paper is organized as follows. The following section introduces the art of state about public encryption based RDH-ED and analyzes the potential of DE for RDH-ED. Section III introduces the techniques of FHE, key-switching, and bootstrapping. Section IV describes the detailed processes of the proposed full homomorphic encryption encapsulated difference expansion. In Section V, the three judging standards of RDH-ED, including *correctness*, *security* and *efficiency*, are discussed theoretically and verified with experimental results. Finally, Section VI summarizes the paper and discusses future investigations.

RELATED WORK

Currently, researches of public key encryption based RDH-ED are mainly based on Paillier encryption (Chen et al., 2014; Shiu et al., 2015; Wu, Chen & Weng 2016; Zhang et al., 2016; Wu, Cheung & Huang 2016; Li & Li 2017; Xiang & Luo 2018) and learning with Error (LWE) encryption (Ke et al., 2016; Ke et al., 2018; Li et al., 2018). Probabilistic and homomorphic properties of the above cryptography allow the third party, *i.e.*, the cloud servers, to conduct operations directly on ciphertext without knowing the private key, which shows potential for more flexible realizations of RDH-ED.

The first Paillier encryption based RDH-ED was proposed by Chen *et al.* (2014). Shiu *et al.* (2015) and Wu *et al.* (2016) improved the EC of (Chen et al., 2014) by solving the pixel overflow problem. Those algorithms were VRBE methods. Li *et al.* (2017) proposed a VRAE method with a considerable EC by utilizing the homomorphic addition property of Paillier encryption and HS technique. The above algorithms were all inseparable. Data extraction was implemented only in the plaintext domain. It was a crucial bottleneck of public key encryption based RDH-ED to realize data extraction directly from the encrypted domain. Wu *et al.* proposed two RDH-ED algorithms for the encrypted images in (Wu, Cheung & Huang 2016): a high-capacity algorithm based on Paillier cryptosystem was presented for data extraction after image decryption. The other one could operate data extraction in the encryption domain. Zhang *et al.* (2016) proposed a combined scheme consisting of a lossless scheme and a reversible scheme to realize separability. Data was extracted from the encrypted domain in the lossless scheme and from the plaintext domain in the reversible scheme. In Xiang & Luo, (2018), Xiang embedded the ciphertext of additional data into the LSBs of the encrypted pixels by employing homomorphic multiplication. Only the ciphertext of additional data could be obtained during extraction directly from ciphertext. To distinguish the corresponding plaintext of the ciphertext of additional data without the private key, a one-to-one mapping table from ciphertext to plaintext was introduced while the ciphertext of additional data for embedding was not from encryption but from the mapping table. However, the exposure and accumulation of a large number of the mapping tables to an untrusted third party might increase the risk of cryptanalysis in theory, while the Paillier algorithms cannot resist *adaptive chosen ciphertext attack* (ACCA or CCA2) (Paillier & Pointcheval, 1999).

LWE based RDH-ED was first proposed in Xiang & Luo, (2018) by quantifying the LWE encrypted domain and recoding the redundancy from ciphertext. Ke *et al.* fixed the parameters for LWE encryption and proposed a multilevel RDH-ED with a flexible applicability and high EC in Ke et al., (2016). However, the data-hiding key used for extraction overlapped partly with the private

key for decryption, thus resulting in limitation for embedding by a third party. In Li et al., (2018), separability could be achieved by preserving correlation from the plaintext in the ciphertext through a modified somewhat LWE encryption. However, the correlation among ciphertext was strong, and it was theoretically vulnerable to cryptanalysis attacks. This paper proposes a lossless data hiding in encrypted domain (RDH-ED) scheme. To realize the data extraction directly from the encrypted domain without the private key, a key-switching based least-significant-bit (LSB) data hiding method has been designed. In application, the user first encrypts the plaintext and uploads ciphertext to the server. Then the server performs key-switching based LSB to obtain the marked ciphertext. Additional data can be extracted directly from the marked ciphertext by the server without the private key, which enables the (trusted or untrusted) third party to manage ciphertext flexibly under the premise of keeping the plaintext secret. The Experimental results demonstrate that the embedding capacity is 1bit per bit of plaintext. Data hiding would not affect the accuracy and the security of encryption.

PRELIMINARIES

LWE Encryption

The private key is denoted as $\boldsymbol{s}$, and the public key $\boldsymbol{A}$ is generated by $\boldsymbol{s}$ and $\boldsymbol{e}$ satisfying Eq. (1), where $\boldsymbol{e}$ is sampled randomly:

$$\boldsymbol{A} \cdot \boldsymbol{s} = 2\boldsymbol{e} \tag{1}$$

Encryption:

The plaintext is $m \in \{0, 1\}$. Set $\boldsymbol{m} = (m, 0, 0, \ldots, 0)$. Generate a 0-1 sequence $\boldsymbol{a}_r$ uniformly and output the ciphertext:

$$\boldsymbol{c} = \boldsymbol{m} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{a}_r \tag{2}$$

Decryption:

$$\left[\left[\langle \boldsymbol{c}, \boldsymbol{s}\rangle\right]_q\right]_2 = \left[\left[\left\langle \boldsymbol{m} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{a}_r, \boldsymbol{s}\right\rangle\right]_q\right]_2 = \left[\left[\boldsymbol{m}^{\mathrm{T}}\boldsymbol{s} + \left(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{a}_r\right)^{\mathrm{T}}\boldsymbol{s}\right]_q\right]_2$$

$$= \left[\left[m + \boldsymbol{a}_r{}^{\mathrm{T}}\boldsymbol{A}\boldsymbol{s}\right]_q\right]_2 = \left[\left[m + \boldsymbol{a}_r{}^{\mathrm{T}}2\boldsymbol{e}\right]_q\right]_2 = m \tag{3}$$

where $[.]_q$ means to perform modulo q. The correctness lies in that the total introduced noise could be restrained to meet:

$$\boldsymbol{a}_r{}^{\mathrm{T}}\boldsymbol{e} < q/4 \tag{4}$$

Key–Switching (Brakerski et al., 2014)

There is data expansion in LWE encrypted ciphertext (Ke et al., 2018). In fully homomorphic encryption based on LWE, a secondary expansion would occur when ciphertext got multiplied.

Therefore, the amount of data will again expand geometrically. If the secondary expansion cannot be eliminated or controlled, the amount of ciphertext data can produce an excessively extension that is unacceptable in practice. Key-switching can effectively eliminate the extension by replacing the extended ciphertext with new ciphertext of any shorter length without decrypting it, and ensure the new ciphertext corresponds to the same decryption as the extended ciphertext.

We use the key-switching technique in the proposed scheme not to eliminate the ciphertext secondary expansion, but to operate a key-switching based LSB data hiding by randomly changing the LSB of specific ciphertext using key switching until the LSB is the same as the to-be-embedded bit..

THE PROPOSED SCHEME

Parameters Setting and Function Definition

The cryptosystem is parameterized by the integers: n (the length of the private key), $q \in (n^2, 2n^2)$ (the modulus), $d \geq (1+\varepsilon)(1+n)\log_2 q$ (the dimension of the public key space), $\varepsilon > 0$. If q is a prime, all the operations in the cryptosystem are performed modulo q in $\mathbb{Z}_q$, $\beta = \lfloor \log_2 q \rfloor$. We denote the noise probability distribution on $\mathbb{Z}_q$ as χ, $\chi = \bar{\Psi}_{aq}$, where the discrete Gaussian distribution $\bar{\Psi}_{aq} = \{ \lceil qx \rfloor \bmod q \mid x \sim N(0, \alpha^2)\}$, and $\lceil qx \rfloor$ denotes rounding qx to the nearest integer (Ke et al., 2018).

Definition 1: The private key generating function:

$$\boldsymbol{s} = SKGen_{n,q}(.) \tag{5}$$

which returns the private key $\boldsymbol{s} \in \mathrm{Z}_q^n : \boldsymbol{s} = (1, \boldsymbol{t})$, where $\boldsymbol{t} \in \mathrm{Z}_q^{n-1}$ is sampled from the distribution χ.

Definition 2: The public key generating function:

$$\boldsymbol{A} = PKGen_{(d,n),q}(\boldsymbol{s}) \tag{6}$$

in which a matrix $\boldsymbol{W} \in \mathrm{Z}_q^{d \times (n-1)}$ is first generated uniformly and a vector $\boldsymbol{e} \in \mathrm{Z}_q^d$ is sampled from the distribution χ, then the vector $\boldsymbol{b} \in \mathrm{Z}_q^d$ is obtained:

$$\boldsymbol{b} = \boldsymbol{W}\boldsymbol{t} + 2\boldsymbol{e} \tag{7}$$

the n-column matrix $\boldsymbol{A} \in \mathrm{Z}^{d \times n}$ is consisting of $\boldsymbol{b}$ followed by $-\boldsymbol{W}$, $\boldsymbol{A} = \left(\boldsymbol{b}, -\boldsymbol{W}\right)$. $\boldsymbol{A}$ is returned as the public key.

Remark: Observe that $\boldsymbol{A} \cdot \boldsymbol{s} = 2\boldsymbol{e}$ for Eq. (8).
Definition 3: The encrypting function:

$$\boldsymbol{c} = Enc_{\boldsymbol{A}}(m) \tag{8}$$

which returns a vector $\boldsymbol{c}$ as the ciphertext of one bit plaintext $m \in \{0,1\}$ with the public key $\boldsymbol{A}$: Set $\boldsymbol{m} = (m, 0, 0, \ldots, 0) \in \mathrm{Z}_2^n$. Generate a random vector $\boldsymbol{a}_r \in \mathrm{Z}_2^d$ uniformly and output $\boldsymbol{c}$:

$$\boldsymbol{c} = \boldsymbol{m} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{a}_r \tag{9}$$

Definition 4(Brakerski et al., 2014): The function $BitDe(\boldsymbol{x})$, $\boldsymbol{x} \in \mathrm{Z}_q^n$, decomposes $\boldsymbol{x}$ into its bit representation. Namely, it outputs $(\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \ldots, \boldsymbol{u}_\beta) \in \mathrm{Z}_q^{n\beta}$, $\boldsymbol{x} = \sum_{j=0}^{\beta-1} 2^j \cdot \boldsymbol{u}_j$, $\boldsymbol{u}_j \in \mathrm{Z}_2^n$.

Definition 5: The decrypting function:

$$m = Dec_{\boldsymbol{s}}(\boldsymbol{c}) = \left[\left[\langle \boldsymbol{c}, \boldsymbol{s} \rangle\right]_q\right]_2 \tag{10}$$

which returns the plaintext bit $m \in \{0,1\}$ with the private key $\boldsymbol{s}$. If the inputs of the decryption function are in binary form, we could regard such a function as a decryption circuit, denoted as $Dec^*_{\boldsymbol{S}}(\boldsymbol{C})$, $\boldsymbol{C} = BitDe(\boldsymbol{c})$, $\boldsymbol{S} = BitDe(\boldsymbol{s})$.

Definition 6(Brakerski et al., 2014): The function $Powersof(\boldsymbol{x})$, $\boldsymbol{x} \in \mathrm{Z}_q^n$, outputs the vector $(\boldsymbol{x}, 2\boldsymbol{x}, 2^2\boldsymbol{x}, \ldots, 2^{\beta-1}\boldsymbol{x},) \in \mathrm{Z}_q^{n\cdot\beta}$.

Next, we will give the procedure of key-switching, which takes a ciphertext $\boldsymbol{c}_1$ under $\boldsymbol{s}_1$ and outputs new ciphertext $\boldsymbol{c}_2$ that encrypts the same plaintext under the private key $\boldsymbol{s}_2$.

Definition 7(Brakerski et al., 2014): The switching key generating function:

$$\boldsymbol{B} = SwitchKGen(\boldsymbol{s}_1, \boldsymbol{s}_2) \tag{11}$$

where $\boldsymbol{s}_1 \in \mathrm{Z}_q^{n1}$, $\boldsymbol{s}_2 \in \mathrm{Z}_q^{n2}$. $\boldsymbol{A}_{\text{temp}} = PKGen_{(n1\bullet\beta,\, n2),\, q}(\boldsymbol{s}_2)$. The matrix $\boldsymbol{B} \in \mathrm{Z}_q^{(n1\cdot\beta)\times n2}$ can be obtained by adding $Powersof(\boldsymbol{s}_1)$ to $\boldsymbol{A}_{\text{temp}}$'s first column.

Ciphertext $\boldsymbol{c}_2$ can be obtained by using the switching key:

$$\boldsymbol{c}_2 = BitDe(\boldsymbol{c}_1)^{\mathrm{T}} \cdot \boldsymbol{B} \tag{12}$$

In our scheme, there is a *key-switching based LSB data hiding* method proposed to ensure that the servers could directly extract additional data from ciphertext without using the private key. We generate a pseudo-random binary sequence $\boldsymbol{k}$ for the servers to randomly scramble the additional data before key-switching based LSB data hiding. The switching key for key-switching based LSB data hiding is:

$$\boldsymbol{B}_{\text{LSB}} = SwitchKGen(\boldsymbol{s}, \boldsymbol{s}) \tag{13}$$

where $\boldsymbol{s} \in Z_q^n$ .All different keys are distributed as shown in Table 1:

Encryption

For the pixel pair (X, Y), whose iLSBs are denoted by b_X^i , b_Y^i (i=1, 2, …, 8).

Each bit is encrypted by LWE encryption with a new public key. We omit the symbol "$_A$" in Eq. (8) for short in this paper: $\boldsymbol{c}_X^i = Enc\,(b_X^i)$, $\boldsymbol{c}_Y^i = Enc\,(b_Y^i)$, i=1, 2,…, 8.

Key-Switching Based LSB Data Hiding

Step 1: Randomly scramble the additional data sequence $\boldsymbol{b}_{\mathrm{s}}$ by using data hiding key $\boldsymbol{k}$ to obtain the to-be-embedded data $\boldsymbol{b}_{\mathrm{r}}$:

$$\boldsymbol{b}_{\mathrm{r}} = \boldsymbol{k} \oplus \boldsymbol{b}_{\mathrm{s}} \tag{14}$$

where $b_{\mathrm{r}} \in \boldsymbol{b}_{\mathrm{r}}$.

Denote the last element of $\boldsymbol{c}_{X'}^1$ as c_{LX1}, whose LSB would be replaced by b_{r} (X is the "1" signed pixel by $\boldsymbol{M}_{\mathrm{ava}}$).

Step 2: If $b_{\mathrm{r}} = LSB(c_{LX1})$, $\boldsymbol{c}_{X'}^1$ maintains the same, or if $b_{\mathrm{r}} \neq LSB(c_{LX1})$, $\boldsymbol{c}_{X'}^1$ is refreshed by:

$$\boldsymbol{c}_{X'}^1 = BitDe\,(\boldsymbol{c}_{X'}^1)^{\mathrm{T}} \cdot \boldsymbol{B}_{\mathrm{LSB}}.$$

Step 3: Repeat Step 2 until $LSB(c_{LX1}) = b_{\mathrm{r}}$.

The marked ciphertext is obtained: $\boldsymbol{c}_{X'}^i$ and $\boldsymbol{c}_{Y'}^i$ (i=1, 2,…, 8).

After receiving the marked ciphertext, the client user could implement the decryption on the marked ciphertext to obtain X and Y by using $\boldsymbol{s}$: $b_{X'}^i = Dec_{\boldsymbol{s}}\,(\boldsymbol{c}_{X'}^i)$, $b_{Y'}^i = Dec_{\boldsymbol{s}}\,(\boldsymbol{c}_{Y'}^i)$, ($i$=1, 2, …, 8).

LSB Extraction from the Marked Ciphertext

Additional data could be directly extracted from ciphertext without the private key $\boldsymbol{s}$ (X is the "1" signed pixel by $\boldsymbol{M}_{\mathrm{ava}}$):

$$b_{\mathrm{r}} = LSB(c_{LX1}) \tag{15}$$

$$\boldsymbol{b}_{\mathrm{s}} = \boldsymbol{k} \oplus \boldsymbol{b}_{\mathrm{r}} \tag{16}$$

THEORETICAL ANALYSIS AND EXPERIMENTAL RESULTS

Correctness

The correctness of the proposed scheme includes the lossless restoration of plaintext and the accurate extraction of the embedded data. The test images, 512×512 8-bit grayscale images, are from image libraries, USC-SIPI (http://sipi.usc.edu/database/database.php?volume= misc) and Kodak (http://r0k.us/graphics/kodak/index.html).

The experimental results of six test images were selected in this section to demonstrate the correctness. The six test images are as shown in Figure 1. The preprocessing LWE encryption & decryption, key switching, and key-switching based LSB were all implemented on MATLAB2010b with a 64-bit single core (i7-6800K) @ 3.40GHz.

Parameters setting: Solving the LWE problem with given parameters is equivalent to solving Shortest Vector Problem (SVP) in a lattice with a dimension $\sqrt{n\log_2(q)/\log_2(\delta)}$.

Considering the efficiencies of the best known lattice reduction algorithms, the secure dimension of the lattice must reach 500 ($\delta = 1.01$) (Gama & Nguyen, 2010), (Ruckert & Schneider, 2010). An increase in n will result in a high encryption blowup. To balance security and the efficiency of practical use, we set n =240, q=57601, d =4573. To ensure the fidelity of the marked plaintext, we set h_{fid}=10.

Accuracy of Plaintext Recovery

In the proposed scheme, the user directly decrypts the marked ciphertext to get the plaintext. We calculated the PSNR of the plaintext

The values of PSNR with the maximum EC are listed in Table 1. From the results of PSNR, it could be seen that there is no embedding distortion in the plaintext.

Accuracy of Data Extraction

There are three cases of data extraction in this paper. The realization of the three cases is the embodiment of the separability of the proposed scheme:

a) The third-party server directly extracts the embedded data from the marked ciphertext by using key-switching based LSB extraction. Figure 2 shows the comparison result bit by bit between the extracted data and the additional data with an EC of 100000 bits in the experiment. It demonstrates that the extraction accuracy was 100%.

Security

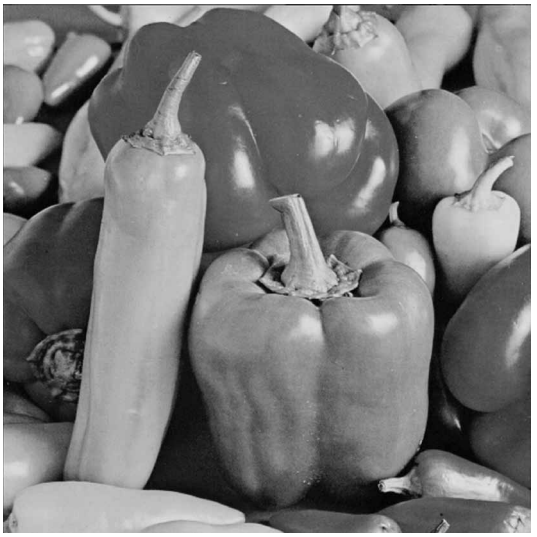
Security of RDH-ED mainly includes two aspects: *a)* Data hiding should not weaken the security of the original encryption or leave any hidden danger of security cracking. *b)* The embedded information cannot be obtained by an attacker without the extraction key or the private key.

In (Ke et al., 2016; Ke et al., 2018), through the derivation of the probability distribution function (PDF) on the marked ciphertext and the experimental analysis of the statistical features, it was proved that the ciphertext distribution before and after data hiding did not change, so that the security of the RDH-ED method was proved by certain reasoning.

In this paper, all the operations of the proposed scheme are equivalent to the operations of re-encryption (Gentry, 2009), and the encryption security can be directly guaranteed by the re-encryption principles. The histograms of ciphertext before and after lossless data hiding are demonstrated in Figure 3, in which the statistical characteristics of ciphertext before and after embedding remain stable, thus ensuring the secrecy of lossless data hiding method.

The processes of implementing key-switching based LSB on the ciphertext are equivalent to the processes of re-encrypting the ciphertext, which would not reveal anything about the private key or reduce the encryption security. The additional data is scrambled using sequence encryption by the third party before key-switching based LSB data hiding, which ensure the confidentiality of the additional data. During the transmission or processing by third-party servers, the third party does not obtain any information related to the client user's private key, nor did it expose any relationship between plaintext and its corresponding ciphertext.

Figure 1a. The test images: Peppers



In summary, the security of the proposed scheme can realize the security that LWE encryption has achieved. What is more, the security of LWE encryption reaches anti-quantum algorithm analysis, while Paillier algorithms cannot resist quantum algorithm analysis.

Efficiency

Public Key Consumption

A new ciphertext can be obtained by performing only once matrix multiplication between a switching key and the old ciphertext, which is fast and can ensure the confidentiality of plaintext and the private key.

Key-switching based LSB data hiding is to randomly change the LSB of specific ciphertext by key switching until the LSB is the same as the to-be-embedded bit. Therefore, the ciphertext from one bit of plaintext could carry one bit of additional data after key-switching based LSB method.

Let the number of times of key switching performed for one bit embedding be λ, that is, the public key consumption of key-switching based LSB for one bit embedding is λ. Since the LSB of the ciphertext is 0 or 1 randomly appeared with a probability of 0.5, $\lambda+1$ obeys the geometric distribution as shown in Table 2. It demonstrates that it would be a small probability event with a probability less than 3% to operate more than 4 times key switching to realize one bit embedding. The theoretical value of λ is 0.8906. In the experiment, we performed 1000 key-switching based LSB data hiding tests. The actual λ was 0.995 on average, indicating a high embedding accuracy and efficiency.

Elapsed Time

The public key encryption algorithms, including the Paillier algorithm and the LWE algorithm, have ciphertext extension. In Ke et al., (2018), the ciphertext extension of Paillier and LWE encryption was discussed in detail. Due to the application of the separability of RDH-ED, ciphertext is usually stored in the server or the cloud, the local storage cost of users is not too much. However, the elapsed time of encryption, decryption, data hiding, and data extraction is related to the efficiency in practice. In this section, we mainly demonstrate the elapsed time of each operation. Table 3 lists the elapsed time of the four main operations.

Figure 1b. The test images: Lena

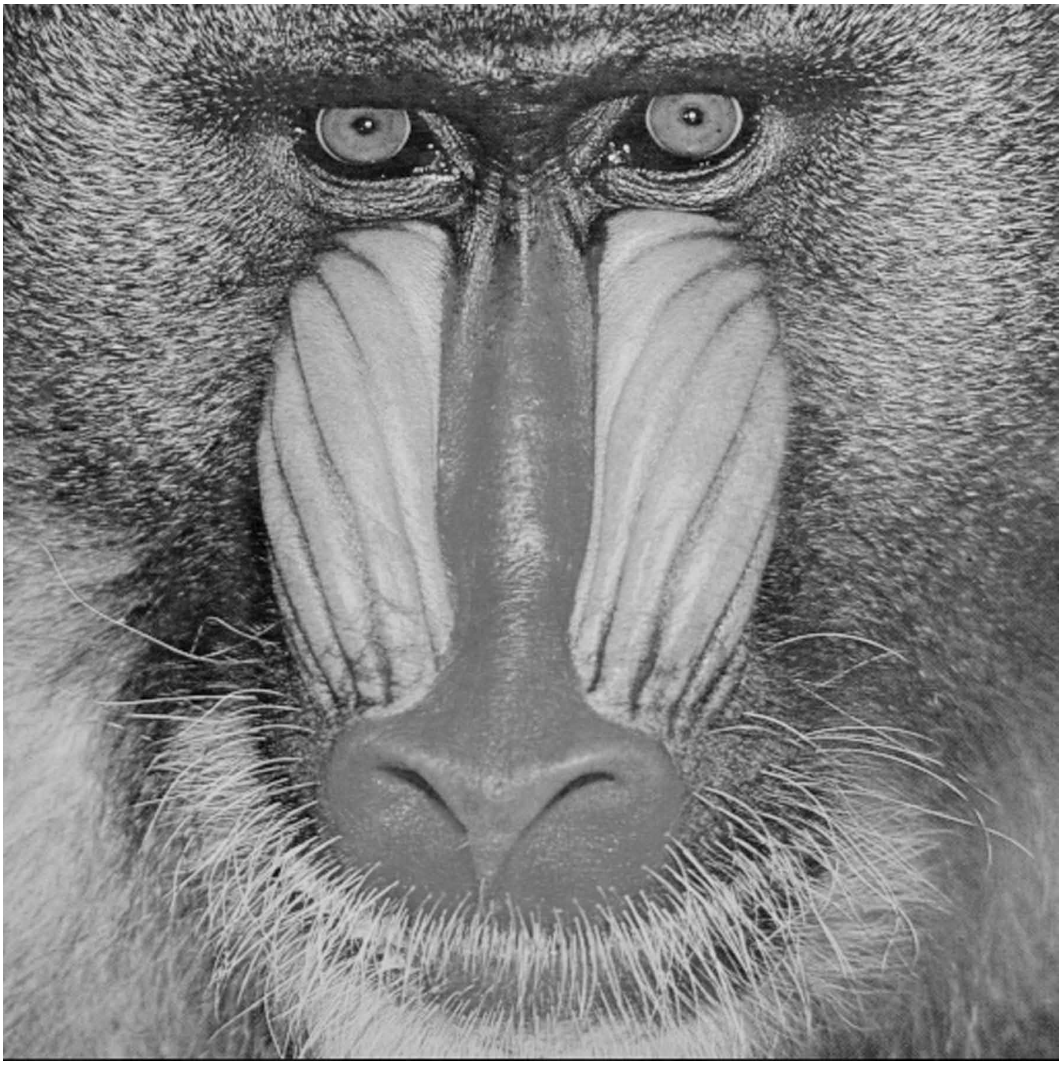
The elapsed time is specifically the time (milliseconds) when one bit plaintext gets decrypted, or one public key is generated and consumed by the operation, e.g., one bit of plaintext gets encrypted after each elapsed time of encryption.

The brief structure and linear operations of LWE provide low time consumption, which are significant in practice. The results in Table 2 indicate that the elapsed time of the proposed method is acceptable for practical use.

CONCLUSION

This paper proposes a lossless data hiding in encrypted domain (RDH-ED) scheme. To realize the data extraction directly from the encrypted domain without the private key, a key-switching based least-significant-bit data hiding method has been designed. The Experimental results demonstrate

Figure 1c. The test images: Baboon



that the embedding capacity is 1bit per bit of plaintext. Data hiding would not affect the accuracy and the security of encryption.

ACKNOWLEDGMENT

This work was supported by National Key Research & Development Program of China under Grant No. 2017YFB0802000, and the National Natural Science Foundation of China under Grant No.61379152, Grant No. 61872384 and Grant No.61403417.

Figure 1d. The test images: Crowd

Figure 1e. The test images: Tank

Figure 1f. The test images: Plane

Table 1. The PSNR (db) with the maximum EC(bits)

Image	EC	PSNR
Lena	110195	∞
Baboon	69286	∞
Crowd	104882	∞
Tank	108963	∞
Peppers	110558	∞
Plane	114834	∞
Average	103120	∞

Figure 2. Errors of the extracted data from LSB extraction on the marked ciphertext

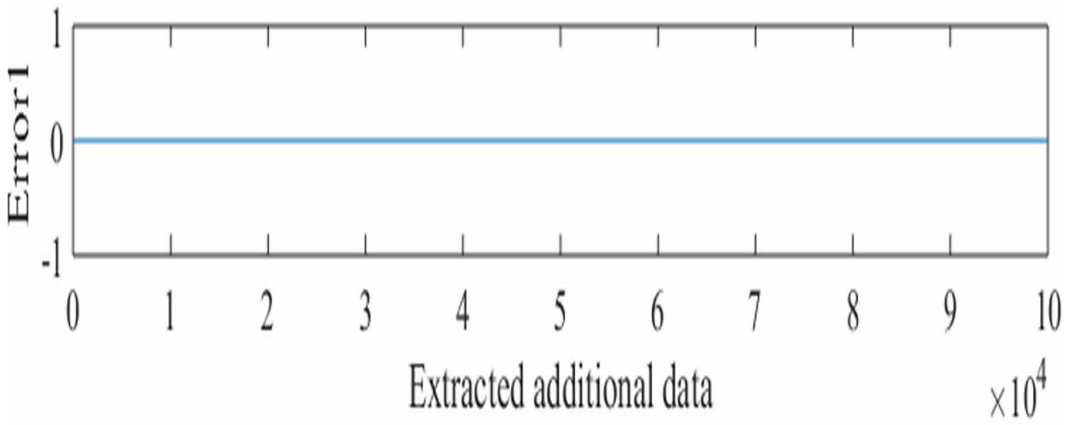


Figure 3a. Histogram of ciphertext before and after lossless data hiding

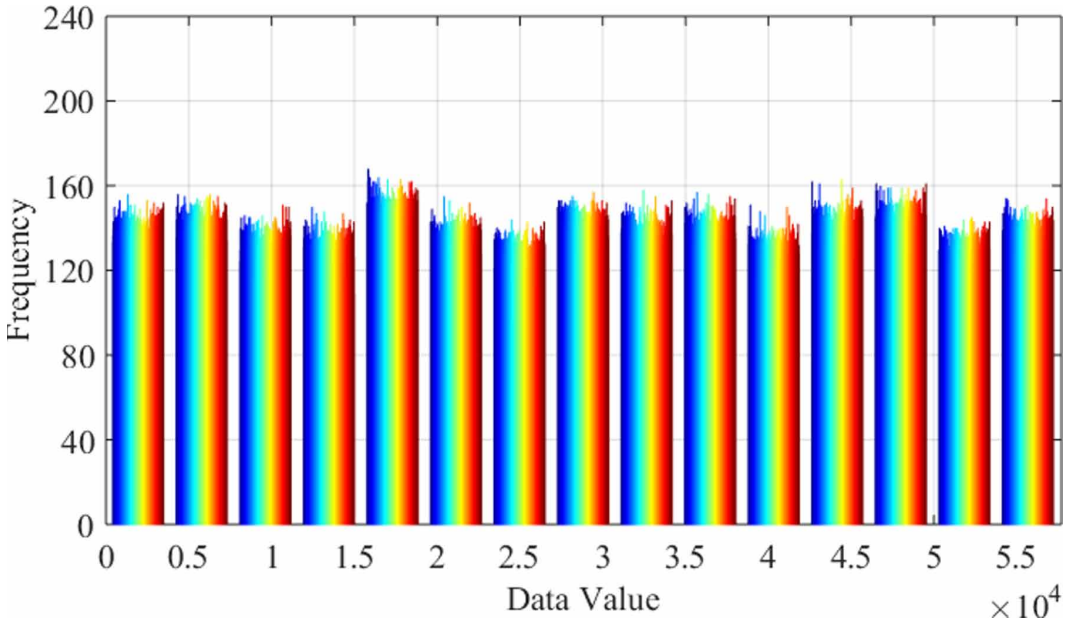


Figure 3b. Histogram of ciphertext before and after lossless data hiding

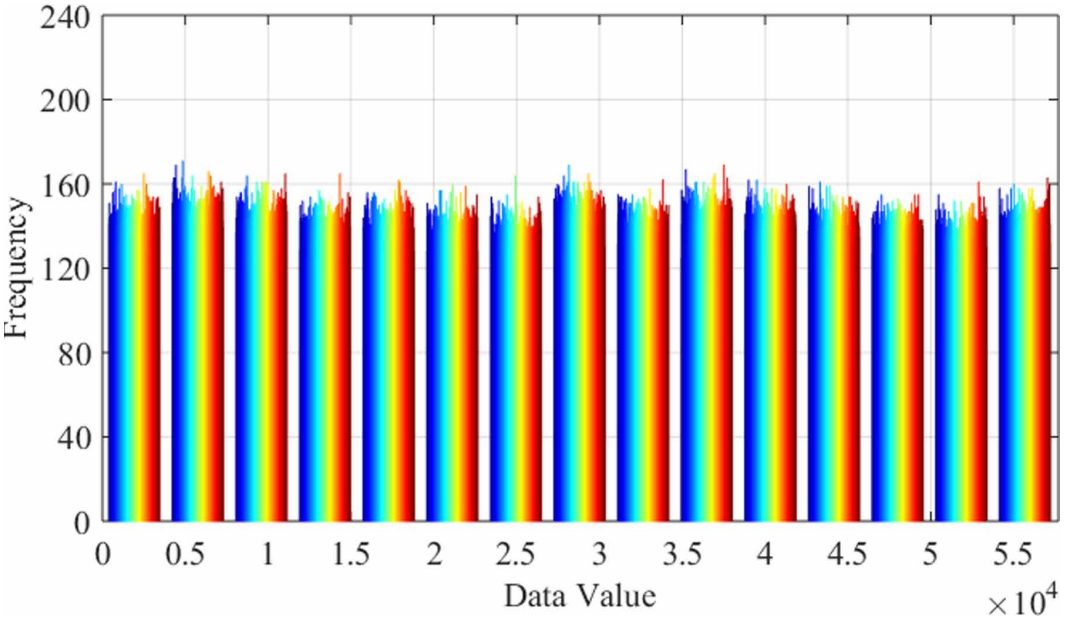


Table 2. The probability distribution of β

λ	0	1	2	3	4	5
P	0.5	0.25	0.125	0.0625	0.0313	0.0156

Table 3. Elapsed time (ms) of once operation

Operation	Encryption	Decryption	Key switching
Elapsed time	20.5971	0.0067	0.1054

REFERENCES

AlFardan, Bernstein, & Paterson. (2013). *On the security of RC4 in TLS and WPA*. Available: http://cr.yp.to/streamciphers/rc4biases- 20130708. pdf

Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2014). (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory*, *6*(3), 1–36. doi:10.1145/2633600

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*, *46*(5), 1132–1143. doi:10.1109/TCYB.2015.2423678 PMID:25955861

Chen, Y.-C., Shiu, C.-W., & Horng, G. (2014). Encrypted signal-based reversible data hiding with public key cryptosystem. *Journal of Visual Communication and Image Representation*, *25*(5), 1164–1170. doi:10.1016/j.jvcir.2014.04.003

Gama, N., & Nguyen, P. Q. (2010). Predicting lattice reduction. *Advances in cryptology-Eurocrypt: 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 31-51.

Gentry, Halevi, & Smart. (2012). Better bootstrapping in fully homomorphic encryption. *International Conference on Practice and Theory in Public Key Cryptography, PKC2012, LNCS, 7293*, 1-16.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. *41st ACM Symposium on Theory of Computing*, 169-178. doi:10.1145/1536414.1536440

Huang, F. J., Huang, J. W., & Shi, Y. Q. (2016, December). New Framework for Reversible Data Hiding in Encrypted Domain. *IEEE Transactions on Information Forensics and Security*, *11*(12), 2777–2789. doi:10.1109/TIFS.2016.2598528

Ke, Y., Zhang, M., & Liu, J. (2016). Separable multiple bits reversible data hiding in encrypted domain. In *Digital Forensics and Watermarking - 15th International Workshop, IWDW 2016* (pp. 470-484). Springer.

Ke, Y., Zhang, M., Liu, J., Su, T., & Yang, X. (2018). A multilevel reversible data hiding scheme in encrypted domain based on LWE. *Journal of Visual Communication and Image Representation*, *54*(7), 133–144. doi:10.1016/j.jvcir.2018.05.002

Li, , MLi, , Y. (2017). Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding. *Signal Processing*, *130*(1), 190–196.

Li, Z. X., Dong, D. P., & Xia, Z. H. (2018). High-capacity reversible data hiding for encrypted multimedia data with somewhat homomorphic encryption. IEEE Access, 6(10), 60635-60644.

Li, M., Di Xiao, Y. Z., & Nan, H. (2015). Reversible data hiding in encrypted images using cross division and additive homomorphism. *Signal Processing Image Communication*, *39PA*(11), 234–248. doi:10.1016/j.image.2015.10.001

Li, X., Yang, B., & Zeng, T. (2011). Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Transactions on Image Processing*, *20*(12), 3524–3533. doi:10.1109/TIP.2011.2150233 PMID:21550888

Ma, K., Zhang, W., Zhao, X., Yu, N., & Li, F. (2013, March). Reversible data hiding in encrypted images by reserving room before encryption. *IEEE Transactions on Information Forensics and Security*, *8*(3), 553–562. doi:10.1109/TIFS.2013.2248725

Ni, Z., Shi, Y.-Q., Ansari, N., & Su, W. (2006, March). Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *16*(3), 354–362. doi:10.1109/TCSVT.2006.869964

Ou, B., Li, X., Zhao, Y., Ni, R., & Shi, Y.-Q. (2013). Pairwise Prediction-Error Expansion for Efficient Reversible Data Hiding. *IEEE Transactions on Image Processing*, *22*(12), 5010–5021. doi:10.1109/TIP.2013.2281422 PMID:24043388

Paillier, P., & Pointcheval, D. (1999). Efficient Public-Key Cryptosystems Provably Secure Against Active Adversaries. Lecture Notes in Computer Science, 1716, 165-179. S doi:10.1007/978-3-540-48000-6_14

Puech, W., Chaumont, M., & Strauss, O. (2008). A reversible data hiding method for encrypted images. Proc. SPIE 6819, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, 68, 191E–68 191E–9.

Puteaux, P., & Puech, W. (2018). An efficient msb prediction-based method for high-capacity reversible data hiding in encrypted images. *IEEE Transactions on Information Forensics and Security*, *13*(7), 1670–1681. doi:10.1109/TIFS.2018.2799381

Qian, Z., Zhang, X., & Wang, S. (2014, May). Reversible data hiding in encrypted JPEG bitstream. *IEEE Transactions on Multimedia*, *16*(5), 1486–1491. doi:10.1109/TMM.2014.2316154

Ruckert, M., & Schneider, M. (2010). *Estimating the security of latticed-based cryptosystems*. Available: http://eprint.icur.org/ 2010/ 137.pdf

Shi, Y.-Q., Li, X., Zhang, X., Wu, H., & Ma, B. (2016, May). Reversible Data Hiding: Advances in the Past Two Decades. *IEEE Access: Practical Innovations, Open Solutions*, *4*(5), 3210–3237. doi:10.1109/ACCESS.2016.2573308

Shiu, C.-W., Chen, Y.-C., & Hong, W. (2015). Encrypted image-based reversible data hiding with public key cryptography from difference expansion. *Signal Processing Image Communication*, *39*, 226–233. doi:10.1016/j.image.2015.09.014

Tian, J. (2003, August). Reversible data embedding using a difference expansion. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(8), 890–896. doi:10.1109/TCSVT.2003.815962

Wu, H.-T., Cheung, Y.-M., & Huang, J.-W. (2016). Reversible data hiding in paillier cryptosystem. *Journal of Visual Communication and Image Representation*, *40*(10), 765–771. doi:10.1016/j.jvcir.2016.08.021

Wu, H.-Z., Shi, Y.-Q., Wang, H.-X., & Zhou, L.-N. (2017, August). Separable reversible data hiding for encrypted palette images with color partitioning and flipping verification. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(8), 1620–1631. doi:10.1109/TCSVT.2016.2556585

Wu, X., Chen, B., & Weng, J. (2016). Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer. *Journal of Visual Communication and Image Representation*, *41*(11), 58–64. doi:10.1016/j.jvcir.2016.09.005

Wu, X., & Sun, W. (2014, November). High-capacity reversible data hiding in encrypted images by prediction error. *Signal Processing*, *104*(11), 387–400. doi:10.1016/j.sigpro.2014.04.032

Xiang, S.-J., & Luo, X. (2018). Reversible Data Hiding in Homomorphic Encrypted Domain By Mirroring Ciphertext Group. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(11), 3099–3110. doi:10.1109/TCSVT.2017.2742023

Zhang, W., Ma, K., & Yu, N. (2014). Reversibility improved data hiding in encrypted images. *Signal Processing*, *94*(1), 118–127. doi:10.1016/j.sigpro.2013.06.023

Zhang, X. (2011, April). Reversible data hiding in encrypted image. *IEEE Signal Processing Letters*, *18*(4), 255–258. doi:10.1109/LSP.2011.2114651

Zhang, X. (2012). Separable reversible data hiding in encrypted image. *IEEE Transactions on Information Forensics and Security*, *7*(2), 826–832. doi:10.1109/TIFS.2011.2176120

Zhang, X.-P., Loong, 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*, *26*(9), 1622–1631. doi:10.1109/TCSVT.2015.2433194

Zhou, J., Sun, W., Dong, L., Liu, X., Au, O. C., & Tang, Y. Y. (2016, March). Secure reversible image data hiding over encrypted domain via key modulation. *IEEE Transactions on Circuits and Systems for Video Technology*, *26*(3), 441–452. doi:10.1109/TCSVT.2015.2416591

Tingting Su received the B.S. degree in information research & security from Engineering University of PAP, Xi'an, China in 2010 and the M.S. degree in cryptography from Engineering University of PAP, Xi'an, China in 2013. Her research interest includes mathematics statistics and cryptography.

Yan Ke received the M.S. degree in cryptography from Engineering University of PAP, Xi'an, China in 2016. He is currently received the Ph.D. degree in cryptography at Engineering University of PAP. His research interest includes reversible data hiding, lattice cryptography.

Yi Ding (PhD) received the B.S. degree and M.S. degree in cryptography from Engineering University of PAP, Xi'an, China. Her research interest includes information security, mathematics statistics and cryptography.

Jia Liu received the M.S. degree in cryptography from Engineering University of PAP, Xi'an, China, in 2007 and Ph.D. degree in neural network and machine learning from Shanghai Jiao Tong University, Shanghai, China, in 2012. Currently, he has been with the Key Laboratory of Network and Information Security under PAP as an associate professor. His research interests include pattern recognition and image processing.