entropy

MDPI

Article

Reversible Data Hiding Algorithm in Fully Homomorphic Encrypted Domain

Jingxuan Li, Xingyuan Liang, Ceyu Dai and Shijun Xiang *

College of Information Science and Technology, Jinan University, Guangzhou 510632, China; nchujnu@gmail.com (J.L.); lahm201311@gmail.com (X.L.); 201822010625@std.uestc.edu.cn (C.D.)
* Correspondence: xiangshijun@gmail.com; Tel.: +86-020-8522-0658

Received: 25 April 2019; Accepted: 17 June 2019; Published: 26 June 2019

Abstract: This paper proposes a reversible data hiding scheme by exploiting the DGHV fully homomorphic encryption, and analyzes the feasibility of the scheme for data hiding from the perspective of information entropy. In the proposed algorithm, additional data can be embedded directly into a DGHV fully homomorphic encrypted image without any preprocessing. On the sending side, by using two encrypted pixels as a group, a data hider can get the difference of two pixels in a group. Additional data can be embedded into the encrypted image by shifting the histogram of the differences with the fully homomorphic property. On the receiver side, a legal user can extract the additional data by getting the difference histogram, and the original image can be restored by using modular arithmetic. Besides, the additional data can be extracted after decryption while the original image can be restored. Compared with the previous two typical algorithms, the proposed scheme can effectively avoid preprocessing operations before encryption and can successfully embed and extract additional data in the encrypted domain. The extensive testing results on the standard images have certified the effectiveness of the proposed scheme.

Keywords: reversible data hiding; DGHV; public key cryptosystem; information entropy; cloud computing

1. Introduction

Reversible data hiding [1–3] is an efficient technology that combines the robustness and provability of digital information, and embeds information for authentication. The hidden data can be extracted completely and the original carrier can be restored completely after data extraction. Because of the existence of these characteristics, reversible data hiding has been applied in many areas such as business and military. The most basic reversible data hiding technology dealt with the redundancy of digital information and then embedded the additional data. There are several reversible data hiding algorithms using difference expansion [4–6], histogram shifting [7–10] and the new prediction error algorithms have higher payload and better image quality [11–15].

For the sake of information security and privacy protection, data are usually encrypted before uploading and transmission. The sender encrypts the plaintext by using the keys and sends the encrypted ciphertext to the receiver. Since the ciphertext is sent, the security of the information is ensured. The receiver can decrypt the ciphertext into plaintext based on the obtained keys. The secret homomorphism was proposed by Rivest in 1978 [16]. First, several plaintexts were encrypted, then the encrypted ciphertexts could be multiplied or added, and decrypted finally. After experimental verification, the results of the operations performed under the idea has been consistent with the results of performing the same operations directly on the same plaintext. There have been more developments in the design of homomorphic encryption schemes. Because of the particularity of the encryption scheme, the homomorphic encryption technology can perform data operations in the encrypted domain

without affecting the final decrypted data. Therefore, homomorphic encryption technologies have been widely used in the operation of secure data. The common homomorphic encryption technology had Pallier encryption system [17], RSA encryption system [18] and ElGmal encryption system [19]. Reversible data hiding technique in encrypted domain is based on this property. Reversible data hiding technique in encrypted domain can implement data hiding procedure in encrypted domain and can recover the original plaintext without error after decryption and data extraction. Owing to this merit, reversible data hiding in encrypted images has been a research hotspot in information security community recently.

There are several categories of reversible data hiding techniques in the literature. The first category is difference expansion-based algorithm, originally proposed by Tian [4]. The reversible data hiding algorithms based on the difference expansion algorithm utilize the correlation of the pixel values of adjacent pixels, and replace the original pixels with the difference of adjacent pixels. After that, the other category of reversible data hiding algorithms was proposed by Thodi [20] and then developed by Li et al. [12]. There are several methods for reversible data hiding in encrypted domain. In [21], by using absolute mean difference of multiple neighboring pixels, the authors proposed a reversible data hiding algorithm in encrypted domain. In [22], the authors proposed another algorithm based on discrete fourier transform and compressive sensing in encrypted domain. In [23,24], the authors proposed two methods for reversible data hiding by using Paillier cryptosystem. In [23], two adjacent pixels as a group are encrypted and then the differences of the two adjacent pixels in each group are computed to generate a difference histogram. The information can be hiding into the histogram by using histogram shifting technique and the homomorphic properties of Paillier system.

In recent years, image encryption technology has developed rapidly. Li [25] briefly summarized the design of image encryption schemes and made an analysis of the challenge of the image encryption faced in the future. In this paper, we propose a new reversible data hiding scheme in the fully homomorphic encrypted domain with the DGHV public key cryptosystem which was proposed by Dijk, Gentry, Halevi and Vaikuntanathan in [26]. By analyzing the insufficiency of existing homomorphic encryption systems, the homomorphic encryption scheme has been improved for multiple bits data encryption in [27]. With the improved encryption scheme, in an image, the use of two adjacent pixels as a group is encrypted by using the same random number. After that, the difference of the two adjacent pixels in each group was computed for computation of histogram. In the encrypted domain, additional data can be embedded and extracted by shifting the difference histogram. Without the data hiding key, the embedded data cannot be extracted. In addition, the additional data can be extracted from the directly decrypted image and the original image can be restored. Compared with the the work in [23,24], the proposed scheme has no preprocessing operations and has lower computational cost.

The remainer of this paper is organized as follows. Shannon's information entropy theory in combination with the encrypted domain is introduced in Section 2. A brief introduction of fully homomorphic encryption over the integers is given in Section 3. The details of the proposed reversible data hiding scheme is introduced in Section 4. Experiment results are given in Section 5. Finally, we have a conclusion in Section 6.

2. Shannon Information Theory

In 1948, Shannon put forward the mathematical theory of communication, which initiated the study of modern information theory [28]. He considered that information entropy can be used to measure the probability distribution of the pixels of a grayscale image. The larger is the information entropy, the more uniform is the probability distribution of the gray image pixels. In the Shannon's information theory, the set of different states of the message samples in the information source is called the probability space, which can be represented by X. In the probability space, the probability of occurrence of the samples is different, and their uncertainty is different. The greater is the probability of a sample appearing, the smaller is its uncertainty; conversely, the smaller is the probability of a

sample appearing, the greater is its uncertainty. If the probability of sample x_i is $p(x_i)$, the information entropy is defined as

$$H(X) = -\sum_i p(x_i) log p(x_i) \tag{1}$$

where $H(X)$ is called information entropy. As is known, the encrypted image loses the correlation between pixels and increases the information entropy of the image. The increase of the entropy makes the histogram distribution of image more uniform. Li [29] pointed out that the information entropy of encrypted image tends to the maximum, thus extra information is difficult to be embedded in the encrypted domain.

In the proposed algorithm, we used two adjacent pixels as a group and encrypted the two pixels in a group with the same parameter. In such a way, the correlation between adjacent pixels can be transmitted to the encrypted domain. As a result, the difference histogram distribution of the encrypted image is not uniform. Thus, there is a residual entropy space for additional information embedding in the encrypted domain.

3. Fully Homomorphic Encryption over the Integers

DGHV fully homomorphic encryption over the integers, which has the characteristics of additive homomorphism and multiplicative homomorphism, is widely used in the field of security. In this cryptosystem, a plaintext is encrypted with public keys. The plaintext can be retrieved after decrypting corresponding ciphertexts with private keys. To ensure the security of the encryption system, the proposed algorithm introduces the greatest common divisor (GCD) problem. Since additive homomorphism and multiplicative homomorphism are permitted in this cryptosystem, it provides an efficient approach to process the original data in encrypted domain.

3.1. Key Generation

Select two integers p and q. p is an odd number and is used as the private key, and q is the large integer. For the security of the ciphertexts, the two integers p and q must satisfy $q >> p$. The greatest common divisor problem is introduced by adding a number of ciphertexts $x_i (0 \leq i \leq l)$ with plaintexts of 0, so that the value of ciphertext is large, to ensure it is not easy to decrypt the ciphertext. At the same time, we must ensure that x_0 is the largest. The public key is $pk, pk =< x_0, x_1, ..., x_l >$.

3.2. Encryption

For each original data m, select an integer r randomly. r is an integer that is generated randomly during the encryption process and n is the number of bits encrypted at one time. Private key p must satisfy

$$m + 2^n r < \frac{p}{2} \tag{2}$$

Denote the encryption function as $E[\cdot]$. The corresponding ciphertext c can be obtained by

$$c = E[m] = m + 2^n r + pq \tag{3}$$

The greatest common divisor problem is introduced by adding a number of ciphertexts with a plaintext of 0, so that the value of ciphertext is large, and the ciphertext is not easy to be decrypted. Equation (3) can be formulated as Equation (4) by introducing the greatest common divisor problem

$$c = E[m] = (m + 2^n r + \sum_{i \in S} x_i) \ mod \ x_0 \tag{4}$$

where c represents the ciphertext of m after adding the greatest common divisors $\sum_{i \in S} x_i$, and S is a subset of $\{0, 1, ..., l\}$.

3.3. Decryption

With corresponding private key p, the original plaintext m can be derived by

$$m = D[c] = (c \ mod \ p) \ mod \ 2^n \tag{5}$$

where the decryption function is denoted as $D[\cdot]$.

3.4. Homomorphic Addition

According to Equations (3) and (5), we have the following derivations for the ciphertexts of m_1 and m_2:

$$E[m_1] = (m_1 + 2^n r_1 + pq_1) \tag{6}$$

$$E[m_2] = (m_2 + 2^n r_2 + pq_2) \tag{7}$$

$$E[m_1] + E[m_2] = (m_1 + m_2) + 2^n(r_1 + r_2) + p(q_1 + q_2) \tag{8}$$

$$\begin{aligned} &D[E[m_1] + E[m_2]] \\ &= [((m_1 + m_2) + 2^n(r_1 + r_2) + p(q_1 + q_2)) \ mod \ p] \ mod \ 2^n \\ &= ((m_1 + m_2) + 2^n(r_1 + r_2)) \ mod \ 2^n \\ &= m_1 + m_2 \end{aligned} \tag{9}$$

where the result of encryption after addition of plaintexts m_1 and m_2 is the same as the result of encrypting the plaintexts m_1 and m_2 and then adding in the ciphertext domain.

3.5. Homomorphic Multiplication

According to Equations (3) and (5), we have the following derivations for the ciphertexts of m_1 and m_2:

$$E[m_1] \times E[m_2] = (m_1 + 2^n r_1)(m_2 + 2^n r_2) + p((m_1 + 2^n r_1)q_2 + (m_2 + 2^n r_2)q_1 + pq_1q_2) \tag{10}$$

$$\begin{aligned} &D[E[m_1] \times E[m_2]] \\ &= [((m_1 + 2^n r_1)(m_2 + 2^n r_2) + p((m_1 + 2^n r_1)q_2 + (m_2 + 2^n r_2)q_1 + pq_1q_2)) \ mod \ p] \ mod \ 2^n \\ &= ((m_1 + 2^n r_1)(m_2 + 2^n r_2)) \ mod \ 2^n \\ &= (m_1m_2 + 2^n(m_1r_2 + m_2r_1 + 2^n r_1r_2)) \ mod \ 2^n \\ &= m_1m_2 \end{aligned} \tag{11}$$

where the result of encryption after multiplication of plaintexts m_1 and m_2 is the same as that of multiplication of ciphertext after encryption of plaintext m_1 and m_2. In conclusion, DGHV homomorphic encryption satisfies both additive homomorphism and multiplicative homomorphism, which means that DGHV is a fully homomorphic encryption scheme.

4. Reversible Data Hiding Scheme with Public Key Cryptosystem

Figure 1 plots the sketch of the proposed scheme, which is composed of four main phases: image encryption, data hiding, data extraction and image restoration. For the data hiding algorithm, refer to [23]. Reversible data hiding is an effective authentication or content integrity verification technique in which hidden data can be completely extracted and the original carrier can be recovered non-destructively after data extraction [30]. After the image owner encrypts the image, the data hider will embed the hidden data in the encrypted domain. With the private key, the receiver can use the corresponding decryption method to extract the encrypted embedded data. Then, by using the steps of extracting algorithm, the embedded data in the encrypted domain is extracted. Only when the private

key is possessed, the receiver can obtain an image containing the embedded data similar to the original image. After decrypting, the embedded data can be extracted and the original image can be restored.

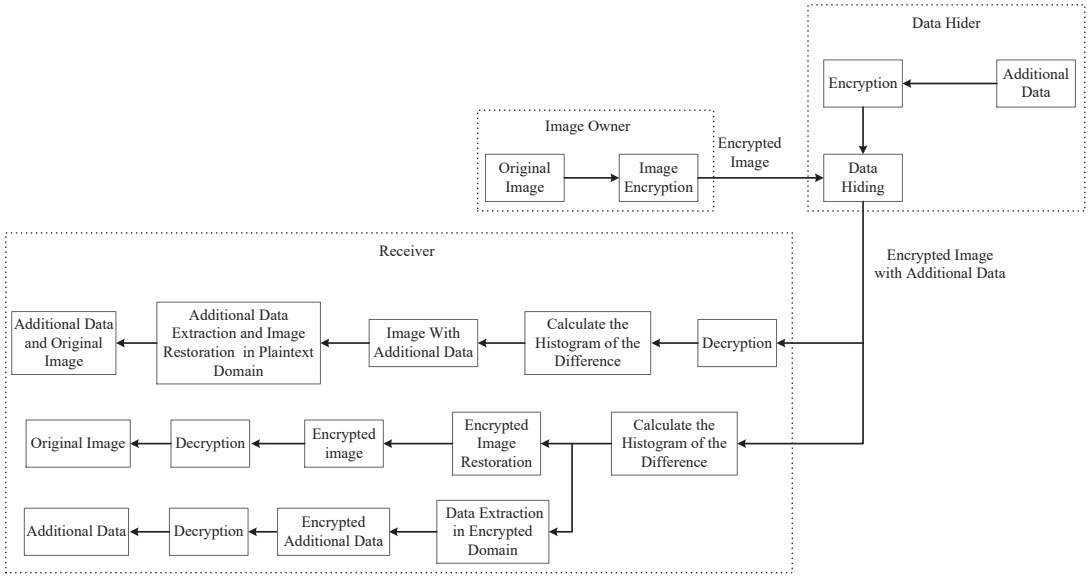

Figure 1. Sketch of the proposed reversible data hiding scheme with public key cryptography.

4.1. Image Encryption

According to the property of DGHV homomorphic cryptosystem, the parameter $r(k)$ is selected randomly to ensure security. It is difficult to embed additional data directly into an encrypted image, because magnitude relationships among plaintexts cannot be kept to the corresponding ciphertexts. For this reason, we designed a corresponding data hiding strategy to embed additional data in the encrypted domain. With this strategy, we can shift the difference histogram for hiding data in encrypted domain.

Firstly, groups of two selected pixels are chosen from original image. Denote the two pixels in kth group as $P_1(k)$ and $P_2(k)$. A data owner selects an integer $r(k)$ randomly, and encrypts $P_1(k)$ and $P_2(k)$ with the public key $\sum_{i \in S(k)} x_i$,

$$C_1(k) = E(m_1) = (m_1 + 2^n r_k + \sum_{i \in S(k)} x_i) \ mod \ x_0 \tag{12}$$

$$C_2(k) = E(m_2) = (m_2 + 2^n r_k + \sum_{i \in S(k)} x_i) \ mod \ x_0 \tag{13}$$

where $C_1(k)$ and $C_2(k)$ are the corresponding ciphertexts of $P_1(k)$ and $P_2(k)$, respectively, and $S(k)$ is a subset of $\{0, 1, ..., l\}$ in the kth group. We let $n = 9$ in this paper to ensure correct decryption. To ensure the security of the ciphertext, we select an integer r_2 in each group, which satisfies $r_2 = i$ and encrypt 0 to generate $C_2(k)'$,

$$C_2(k)' = C_2(k) + E[0, r_2(k)] = C_2(k) + C(0) \tag{14}$$

Let $D[C(k)]$ be the decrypted version of $C(k)$ and $D[C(k)]$ satisfies

$$D[C(k)] = (C(k) \ mod \ p) \ mod \ 2^n \tag{15}$$

A data owner encrypts the original image, and a data hider can obtain the encrypted image and r_2. The embedded data are also encrypted in this method.

4.2. Data Hiding

With the same method of the data encryption, the additional data can be encrypted. Data hider calculates the difference of the two pixels of a group, and then calculates the positive peak point and the negative peak point of the histogram. The pixels corresponding to the two peak points are used to embed additional data.

4.3. Data Embedding

After the image owner encrypts the original image by using the public key pk, the encrypted image I_m and r_2 is transmitted to the data hider. When the data hider receives the encrypted image, the data hider obtains $C(0)$ by receiving r_2, and makes $C_2(k)'$ subtract $C(0)$ to recover $C_2(k)$. The $C_2(k)$ can be calculated by

$$C_2(k) = C_2(k)' - C(0) \tag{16}$$

Then, the data hider embeds the additional data in the encrypted image and obtains a new image I_w. Finally, the data hider sends I_w and r_2, the position of the positive peak point of the histogram and the position of the negative peak point of the histogram to the receiver, and the receiver can extract the embedded data and restore the original image with private keys.

In the proposed algorithm, the adjacent two encrypted pixels $C_1(k)$ and $C_2(k)$ are subtracted to obtain $C_d(k)$. Then, the position of the positive peak point of the histogram is recorded as EC_{max}, and the position of the negative peak point of the histogram is recorded as EC_{min}. When $C_d(k) = EC_{max}$, one bit of encrypted additional data is embedded in the adjacent encrypted pixel $C_1(k)$, and, when $C_d(k) = EC_{min}$, one bit of encrypted additional data is embedded in the adjacent encrypted pixel $C_2(k)$. In this paper, only one round of extra information is embedded.

The specific additional data embedding steps are shown as follows. Firstly, the difference between adjacent pixels in the encrypted domain is calculated as

$$C_d(k) = C_1(k) - C_2(k). \tag{17}$$

Secondly, the position of the positive peak point of the histogram and the position of the negative peak point of the histogram are selected, respectively. When $C_1(k) \geqslant C_2(k)$, the additional data embeds in the right half of the histogram. When $C_1(k) < C_2(k)$, the additional data embeds in the left half of the histogram. The histogram shifting process is shown in Figure 2.

Thirdly, the additional information is embedded in the carrier image in the encrypted domain by shifting the difference histogram. When the additional data is encrypted, we denote it as $E_w(w)$. In this paper, the same public key $\sum_{i \in S(k)} x_i$ and random number $r(k)$ are used to encrypt the additional information. Bit 1 and bit 0 are denoted as $E(0)$ and $E(1)$, respectively, in the encrypted domain and encrypted by the public key, which is the same as the public key used to encrypt the additional information. The embedded image pixels are $C_{w1}(k)$ and $C_{w2}(k)$. $C_{w1}(k)$ and $C_{w2}(k)$ are calculated as follows:

If $C_1(k) \geqslant C_2(k)$,

$$C_{w1}(k) = \begin{cases} (C_1(k) + E_w(w)) \, mod \, x_0, & if \; C_d(k) = EC_{max} \\ (C_1(k) + E(1)) \, mod \, x_0, & if \; C_d(k) \geqslant EC_{max} + 1 \\ (C_1(k) + E(0)) \, mod \, x_0, & else \end{cases} \tag{18}$$

$$C_{w2}(k) = (C_2(k) + E(0)) \, mod \, x_0 \tag{19}$$

else

$$C_{w2}(k) = \begin{cases} (C_2(k) + E_w(w)) \bmod x_0, & if\ C_d(k) = EC_{min} \\ (C_2(k) + E(1)) \bmod x_0, & if\ C_d(k) \leqslant EC_{min} - 1 \\ (C_2(k) + E(0)) \bmod x_0, & else \end{cases} \tag{20}$$

$$C_{w1}(k) = (C_1(k) + E(0)) \bmod x_0 \tag{21}$$

Denote $P_{w1}(k)$ and $P_{w2}(k)$ as the plaintext versions of $C_{w1}(k)$ and $C_{w2}(k)$, respectively. The effect of data hiding on plaintexts is to change $P_1(k)$ and $P_2(k)$ to $P_{w1}(k)$ and $P_{w2}(k)$:

If $P_1(k) \geqslant P_2(k)$,

$$P_{w1}(k) = \begin{cases} P_1(k) + w, & if\ C_d(k) = EC_{max} \\ P_1(k) + 1, & if\ C_d(k) \geqslant EC_{max} + 1 \\ P_1(k), & else \end{cases} \tag{22}$$

$$P_{w2}(k) = P_2(k) \tag{23}$$

else

$$P_{w2}(k) = \begin{cases} P_2(k) + w, & if\ C_d(k) = EC_{min} \\ P_2(k) + 1, & if\ C_d(k) \leqslant EC_{min} - 1 \\ P_2(k), & else \end{cases} \tag{24}$$

$$P_{w1}(k) = P_1(k) \tag{25}$$

To ensure the security of the ciphertext, according to Equations (14), we generate $C_{w2}(k)'$ with r_2 in each group,

$$C_{w2}(k)' = C_{w2}(k) + E[0, r_2(k)] = C_{w2}(k) + C(0) \tag{26}$$

It can be seen that the embedding algorithm can be used not only in the plaintext domain, but also in the ciphertext domain. In other words, the data owner sends the encrypted image to the data hider, and then the data hider embeds the encrypted additional data into the encrypted image directly.

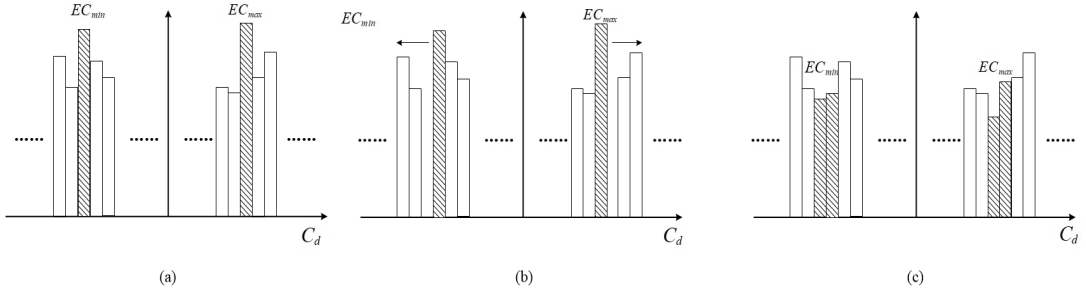

Figure 2. Embedding additional information by shifting the histogram: (**a**) histogram of the differences before embedding; (**b**) shifting the difference histogram to free up embedding space; and (**c**) histogram after embedding the additional information.

4.4. Data Extraction and Image Restoration

In the proposed scheme, data extraction and image restoration can be completed together. There are two ways to extract the hidden data and restore the original image.

4.4.1. Extract the Hidden Data and Restore Original Ciphertext Image in Encrypted Domain

When the receiver receives the encrypted embedded image, r_2, the position of the positive and the negative peak point of the histogram, which has 18 bits of side information, the receiver can find the embedded position and use the following method to extract the embedded additional data and restore original pixel:

Firstly, the receiver obtains $C(0)$ by receiving r_2, and makes $C_{w2}(k)'$ subtract $C(0)$ to recover $C_{w2}(k)$. The $C_{w2}(k)$ can be calculated by

$$C_{w2}(k) = C_{w2}(k)' - C(0) \tag{27}$$

Secondly, calculate the difference between adjacent pixels:

$$C_{wd}(k) = C_{w1}(k) - C_{w2}(k) \tag{28}$$

where $C_{wd}(k)$ is denoted as the difference between adjacent pixels, which contain the additional data in the encrypted domain.

Thirdly, extract embedded data and restore original pixels:

$$E_w(w) = \begin{cases} E_w(0), & if\ C_{wd}(k) = EC_{max}\ or\ C_{wd}(k) = EC_{min} \\ E_w(1), & if\ C_{wd}(k) = EC_{max} + 1\ or\ C_{wd}(k) = EC_{min} - 1 \end{cases} \tag{29}$$

If $C_{w1}(k) \geq C_{w2}(k)$,

$$C_1(k) = \begin{cases} (C_{w1}(k) - E(1))\ mod\ x_0, & if\ C_{wd}(k) \geqslant EC_{max} + 1 \\ (C_{w1}(k) - E(0))\ mod\ x_0, & else \end{cases} \tag{30}$$

$$C_2(k) = (C_{w2}(k) - E(0))\ mod\ x_0 \tag{31}$$

else

$$C_2(k) = \begin{cases} (C_{w2}(k) - E(1))\ mod\ x_0, & if\ C_{wd}(k) \leqslant EC_{max} - 1 \\ (C_{w2}(k) - E(0))\ mod\ x_0, & else \end{cases} \tag{32}$$

$$C_1(k) = (C_{w1}(k) - E(0))\ mod\ x_0 \tag{33}$$

$$W(w) = (E_w(w)\ mod\ p)\ mod\ 2^n \tag{34}$$

$$P(k) = (C_I(k)\ mod\ p)\ mod\ 2^n \tag{35}$$

where $W(w)$ represents the extracted additional data after decryption, p is the private key and $P(k)$ is the restored image after decryption. It can be seen that the receiver can extract the additional data and restore original image without any losses.

4.4.2. Extract the Hidden Data and Restore the Original Image after Decryption

When the receiver receives the encrypted embedded image, the position of the positive peak point of the histogram and the position of the negative peak point of the histogram, the receiver can find the embedding position and use the following method to extract the embedded additional data and restore the original image:

Firstly, decrypt the encrypted embedded image by:

$$P_w(k) = (C_w(k)\ mod\ p)\ mod\ 2^n \tag{36}$$

where $P_w(k)$ represents the decrypted image including the additional data in the plaintext domain.

Secondly, calculate the differences between adjacent pixels:

$$P_{wd}(k) = P_{w1}(k) - P_{w2}(k) \tag{37}$$

where $P_{wd}(k)$ is denoted as the difference between the adjacent pixels in the plaintext domain.

Finally, extract embedded data with:

$$W(k) = \begin{cases} 1, & if\ P_{wd}(k) = EP_{max} + 1\ or\ P_{wd}(k) = EP_{min} - 1 \\ 0, & if\ P_{wd}(k) = EP_{max}\ or\ P_{wd}(k) = EP_{min} \end{cases} \tag{38}$$

If $P_{w1}(k) \geq P_{w2}(k)$.

$$P_1(k) = \begin{cases} P_{w1}(k) - 1, & if\ P_{wd}(k) \geqslant EP_{max} + 1 \\ P_{w1}(k), & else \end{cases} \tag{39}$$

$$P_2(k) = P_{w2}(k) \tag{40}$$

else

$$P_2(k) = \begin{cases} P_{w2}(k) - 1, & if\ P_{wd}(k) \leqslant EP_{min} - 1 \\ P_{w2}(k), & else \end{cases} \tag{41}$$

$$P_1(k) = P_{w1}(k) \tag{42}$$

However, since the value range of the grayscale image pixel in the plaintext domain is [0,255], the pixel value of some images restored to the plaintext domain may be 256 after decryption, which is the overflowing problem. At this point, we change the pixel value 256 to 255, and restore their position information. Then, we can embed these positions into the watermarked image through a reversible algorithm. On the receiver side, the receiver can recover the image by extracting the position information with the reversible algorithm. After that, the legal receiver can extract the additional data and restore the original image without any loss.

5. Experimental Results

In the experiment, we objectively evaluated the experimental results from the aspects of computational cost, peak signal-to-noise ratio (PSNR), embedded rate and imperceptibility. At the same time, by comparing with Xiang [23] and Xiang [24], it showed that the DGHV fully homomorphic encryption watermarking algorithm proposed in this paper is greatly improved in terms of computational cost.

5.1. Computational Cost

A series of experiments confirmed the performance of proposal algorithm. Different standard test images were used to verify the results of the experiment. One of the important performance indicators for measuring the homomorphic encryption algorithms is computational cost. The lower is the computational cost, the more useful the program is in the application. Table 1 presents four grayscale images of size 512×512 for testing and shows the average computational cost of ten times under different capacity.

On the one hand, it can be seen from the experimental data in Table 1 that the encryption and decryption times of different images are slightly different while the embedding capacity and the carrier image size are the same. Besides, the embedding and extraction times are also slightly different, respectively, which are mainly due to the different images in the text. On the other hand, at the stage of embedding additional data and extracting data, the computational cost increases as the number of embedded data increases.

By comparing a plaintext and the resulting version from the corresponding ciphertext, the results show that the pixel values of the original image are the same as the pixel values of the image pixels after decryption. As for the extraction of the additional data, there is no data loss or disorder of the sequence in the extraction of the embedded data.

Table 1. Computational cost for different embedded bits.

Picture	Embedded Data Bits	Time (s)					
		Encryption	Embed	Extraction (Encrypted Domain)	Extraction (Plaintext Domain)	Decryption (Embedded Image)	Decryption (Original Image)
Lena	1024	1.5094	0.0309	0.0270	0.0264	0.0019	0.0019
Airplane		1.5095	0.0460	0.0388	0.0395	0.0021	0.0020
Lake		1.5102	0.0737	0.0638	0.0639	0.0019	0.0020
Man		1.5065	0.0455	0.0387	0.0398	0.0019	0.0019
Lena	2048	1.5103	0.0443	0.0395	0.0399	0.0021	0.0020
Airplane		1.5078	0.0584	0.0505	0.0520	0.0019	0.0019
Lake		1.4905	0.0875	0.0724	0.0753	0.0019	0.0019
Man		1.5257	0.0636	0.0552	0.0566	0.0018	0.0018
Lena	4096	1.5106	0.0707	0.0648	0.0667	0.0018	0.0019
Airplane		1.5125	0.0795	0.0730	0.0740	0.0019	0.0020
Lake		1.5231	0.1244	0.1078	0.1128	0.0019	0.0019
Man		1.5261	0.0960	0.0879	0.0907	0.0021	0.0019

Table 2 uses three same images, respectively, and selects Lena diagrams but different size as a group to compare the average computational cost of ten times of embedded additional data with different number of bits. In Table 2, it can be seen that the computational cost of the proposed algorithm corresponds to the size of the embedded data and the original image. For the same embedding rate, the smaller is the carrier image size, the lower is the computational cost. Furthermore, the computational cost is positively related to the size of the original image. The larger is the image, the higher is the computational cost needed for the encryption. The size of the embedded data has less influence on the computational cost in comparison with the image size.

Table 2. Computational cost with the Lena image with different sizes and embedded rates.

Size	Embedded Data Bits	Time (s)					
		Encryption	Embed	Extraction (Encrypted Domain)	Extraction (Plaintext Domain)	Decryption (Embedded Image)	Decryption (Original Image)
256 × 256	1024	0.3784	0.0125	0.0106	0.0107	0.0016	0.0017
512 × 512		1.5094	0.0309	0.0270	0.0264	0.0019	0.0019
1024 × 1024		6.0922	0.4456	0.4242	0.4194	0.0069	0.0071
256 × 256	2048	0.3745	0.0230	0.0210	0.0213	0.0016	0.0016
512 × 512		1.5103	0.0443	0.0395	0.0399	0.0021	0.0020
1024 × 1024		6.0482	0.4462	0.4262	0.4243	0.0066	0.0076
256 × 256	4096	0.3772	0.0473	0.0456	0.0469	0.0016	0.0016
512 × 512		1.5106	0.0707	0.0648	0.0667	0.0018	0.0019
1024 × 1024		6.0841	0.4533	0.3805	0.4347	0.0067	0.0068

5.2. Security

The security of hiding information in the encrypted domain is one of the important issues. The algorithm proposed in this paper is an algorithm for hiding information in the DGHV full homomorphic encrypted domain, which has both the property of addition homomorphism and multiplication homomorphism. To ensure the security of additional information, the greatest common divisor problem is introduced in the proposed algorithm. That is to say, some ciphertext sets x_i with zero in the plaintext are added. Let x_i be the public key set and randomly select a subset of x_i as the public key for the encryption. Because the additive subsets are ciphertext with 0 in the plaintext, these subsets have no effect on the decryption. If an attacker does not know the private key, he cannot decrypt the encrypted image and cannot extract the additional information since the plaintexts were encrypted with the random integer $r(k)$. Therefore, the security of the embedded information is ensured.

5.3. Imperceptibility

PSNR is one of the important indicators to measure the imperceptibility of the watermarking algorithm in the spatial domain. Generally speaking, the larger is the PSNR, the better is the imperceptibility achieved. Table 3 lists the PSNR values with four different images sized by 512 × 512.

We can see that all of the PSNR values are over 50 dB, showing the embedding distortion is smaller. In Table 3, it can be seen that, when the embedding capacity is larger, the PSNR decreases accordingly.

Table 3. PSNR values with the different embedding capacity.

Picture	Embedded Data Bits	PSNR (dB)	Picture	Embedded Data Bits	PSNR (dB)
Lena	2048	65.8630	Lake	2048	59.5992
	4096	61.4712		4096	57.5459
	8192	57.2428		8192	54.9346
	16384	53.6700			
Airplane	2048	62.8160	Man	2048	62.1950
	4096	60.4387		4096	59.2608
	8192	57.9015		8192	56.2677
	16384	55.3511		16384	53.6685

5.4. Original Images and Their Processed Versions

Figure 3 plots the four original images (Lena, Airplane, Lake and Man), the encrypted images after watermarking, the directly decrypted images, and the restored images. In total, 4096 bits of data are embedded in the encrypted images, respectively. We have the following observations from this figure:

(1) The encrypted image after embedding data has no correlation with the original image. The experimental results show that the additional information in the DGHV fully homomorphic encrypted domain has achieved good results, which depends on the security of DGHV fully homomorphic encryption. That is, without the private key, the encrypted image cannot be decrypted and the additional information cannot be extracted in the encrypted domain.

(2) Comparing the original image with the decrypted image containing the additional information, the visual distortion due to the watermark is small. After the original images are encrypted and embedded with 4096 bits of data, the PSNR values of the watermarked images are greater than 57 dB in Figure 3, which has satisfactory imperceptibility.

(3) The information embedded in the fully homomorphic encrypted domain in the proposed algorithm is reversible, and the original image can be successfully recovered without distortion after decrypting and extracting the embedded information.

5.5. Performance Comparison

The proposed algorithm in this paper uses the histogram shifting technique to embed additional information in the DGHV encryption domain. Compared with Xiang [23], the algorithm proposed in this paper has lower computational cost.

To compare the proposed method with two existing methods [23,24], Figure 4 plots the PSNR values at different embedding rates by using the Lena image and the Airplane image, respectively. It can be seen in Figure 4 that, at the low embedding rate, the PSNR is slightly different from that of the method in [23]. At the high embedding rate, the PSNR of the proposed algorithm is higher than that of the method in [23]. In addition, the PSNR value of the proposed method is much higher than that of the method in [24]. Compared with Xiang [24], the proposed algorithm has greatly improved the imperceptibility of the image after embedding for the same embedding rate.

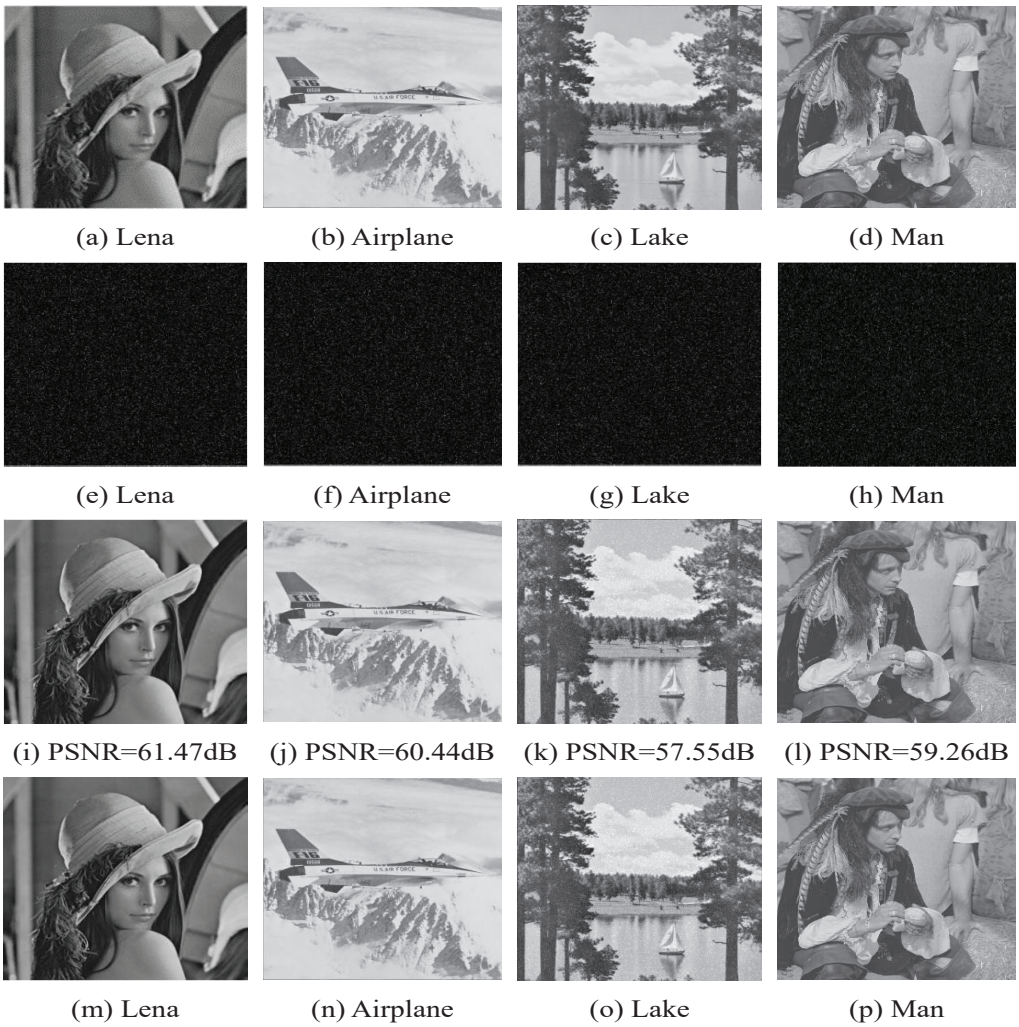

Figure 3. Four original test images Lena, Airplane, Lake and Man (**a**–**d**); the four encrypted images with embedded additional data (**e**–**h**); the four decrypted images (**i**–**l**); and the four restored images (**m**–**p**).

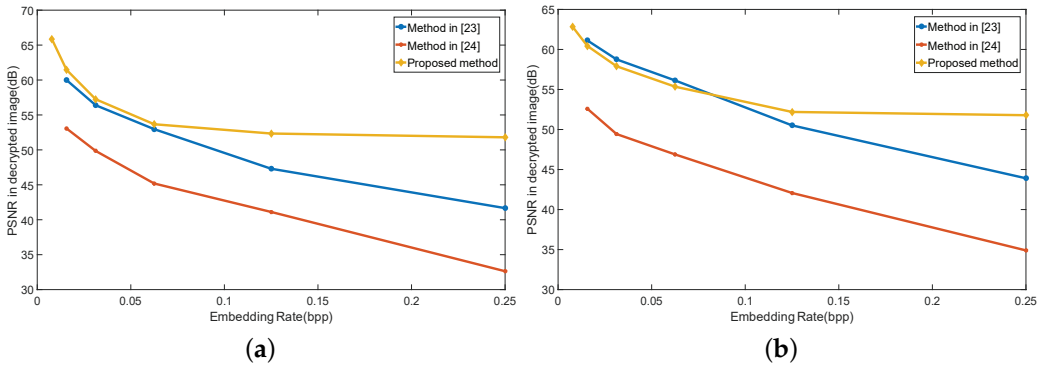

Figure 4. Comparison of embedding capacity versus embedding distortion in different images: (**a**) lena; and (**b**) airplane.

6. Conclusions

In this paper, we propose a reversible data hiding algorithm with DGHV fully homomorphic encryption and analyzed the feasibility of the scheme from the perspective of information entropy. In the proposed algorithm, we used two adjacent pixels as a group and encrypted the two pixels in a group with the same parameter. In such a way, the correlation between adjacent pixels can

be transmitted to the encrypted domain. As a result, the difference histogram distribution of the encrypted image is not uniform. Thus, there is a residual entropy space for additional information embedding in the encrypted domain.

This method has better solved the problems of quickly encrypting multiple bits of data and embedding additional data in an encrypted domain. This algorithm has lower computational cost, higher security, and better imperceptibility. In future research, how to embed a robust and reversible watermark into this encrypted domain will be considered.

Author Contributions: Methodology, S.X.; Writing—original draft, J.L.; and Data curation, J.L., X.L. and C.D..

Funding: This research was funded by the National Nature Science Foundation of China (NSFC) projects (No. 61772234).

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

Abbreviations

The following abbreviations are used in this manuscript:

DGHV	Dijk, Gentry, Halevi and Vaikuntanathan
GCD	Greatest Common Divisor
PSNR	Peak Signal-to-noise Ratio

References

1. Xiang, S.; Li, Z. Reversible audio data hiding algorithm using noncausal prediction of alterable orders. *EURASIP J. Audio Speech Music Process.* **2017**, *2017*, 4. [CrossRef]
2. Xiang, S.; Yang, L.; Wang, Y. Robust and reversible audio watermarking by modifying statistical features in time domain. *Adv. Multimed.* **2017**. [CrossRef]
3. Chen, B.; Wu, X.; Wei, Y.S. Reversible data hiding in encrypted images with private-key homomorphism and public-key homomorphism. *J.* Vis. Commun. Image Represent. **2018**, *57*, 272–282. [CrossRef]
4. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circ. Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
5. Alattar, A.M. Reversible watermarking using the difference expansion of a generalized integer transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef] [PubMed]
6. Hu, Y.; Lee, H.K.; Chen, K.; Li, J. Difference expansion based reversible data hiding using two embedding directions. *IEEE Trans. Multimed.* **2010**, *10*, 1500–1512. [CrossRef]
7. Tai, W.L.; Yeh, C.M.; Chang, C.C. Reversible data hiding based on histogram modification of pixel differences. *IEEE Trans. Circ. Syst. Video Technol.* **2009**, *19*, 906–910.
8. Wu, H.T.; Dugelay, J.L.; Shi, Y.Q. Reversible image data hiding with contrast enhancement. *IEEE Signal Process. Lett.* **2014**, *22*, 81–85. [CrossRef]
9. Chen, Y.H.; Huang, H.C.; Lin, C.C. Block-based reversible data hiding with multi-round estimation and difference alteration. *Multimed. Tools Appl.* **2016**, *75*, 13679–13704. [CrossRef]
10. Tsai, P.; Hu, Y.C.; Yeh, H.L. Reversible image hiding scheme using predictive coding and histogram shifting. *Signal Process.* **2009**, *89*, 1129–1143. [CrossRef]
11. Hong, W.; Chen, T.S.; Shiu, C.W. Reversible data hiding for high quality images using modification of prediction errors. *J. Syst. Softw.* **2009**, *82*, 1833–1842. [CrossRef]
12. Li, X.; Yang, B.; Zeng, T. Efficient reversible watermarking based on adaptive prediction-error expansion and pixel selection. *IEEE Trans. Image Process.* **2011**, *20*, 3524–3533. [PubMed]
13. Kumar, M.; Agrawal, S. Reversible data hiding based on prediction error expansion using adjacent pixels. *Secur. Commun. Netw.* **2016**, *9*, 3703–3712. [CrossRef]
14. Yi, S.; Zhou, Y.; Hua, Z. Reversible data hiding in encrypted images using adaptive block-level prediction-error expansion. *Signal Processing: Image Communication*, **2018**, *64*, 78–88. [CrossRef]
15. Xiang, Y.; Wu, G. Pixel prediction based reversible data hiding scheme for image. *Comput. Sci.* **2018**, *45*, 189–196.

16. Rivest, R.L.; Adleman, L.; Dertouzos, M.L. On data banks and privacy homomorphisms. *Found. Secur. Comput.* **1978**, *4*, 169–180.
17. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; pp. 223–238.
18. Rivest, R.L.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. Assoc. Comput. Mach.* **1978**, *21*, 120–126.
19. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
20. Thodi, D.M.; Rodríguez, J.J. Expansion embedding techniques for reversible watermarking. *IEEE Trans. Image Process.* **2007**, *16*, 721–730. [CrossRef]
21. Liao, X.; Shu, C. Reverisible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels. *J. Vis. Commun. Image Represent.* **2015**, *28*, 21–27. [CrossRef]
22. Liao, X.; Li, K.; Yin, J. Separable data hiding in encrypted image based on compressive sensing and discrete fourier transform. *Multimed. Tools Appl.* **2016**, *76*, 20739–20753. [CrossRef]
23. Xiang, S.; Luo, X. Efficient reversible data hiding in encrypted image with public key cryptosystem. *EURASIP J. Adv. Signal Process.* **2017**, *1*, 59. [CrossRef]
24. Xiang, S.; Luo, X. Reversible data hiding in homomorphic encrypted domain by mirroring ciphertext group. *IEEE Trans. Circ. Syst. Video Technol.* **2018**, *28*, 3099–3110. [CrossRef]
25. Li, C.; Zhang, Y.; Xie, E.Y. When an attacker meets a cipher-image in 2018: A Year in Review. *arXiv* **2019**, arxiv:1903. 11764.
26. Van Dijk, M.; Gentry, C.; Halevi, S.; Vaikuntanathan, V. Fully homomorphic encryption over the integers. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco and Nice, France, 30 May–3 June 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 24–43.
27. Sun, N. Fully homomorphic encryption scheme applied to n bit. *Comput. Appl. Res.* **2018**, *35*, 1179–1181.
28. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
29. Li, M.; Xiao, D.; Zhang, Y.; Nan, H. Reversible data hiding in encrypted images using cross division and additive homomorphism. *Signal Process. Image Commun.* **2015**, *39*, 234–248. [CrossRef]
30. Fridrich, J.; Goljan, M.; Du, R. Lossless data embedding for all image formats. In *Security and Watermarking of Multimedia Contents IV*; International Society for Optics and Photonics: Bellingham, WA USA, 2002; Volume 4675, pp. 572–584.