Non-local Denoising in Encrypted Images*

Xianjun Hu, Weiming Zhang, Honggang Hu, and Nenghai Yu

Key Laboratory of Electromagnetic Space Information, Chinese Academy of Sciences, University of Science and Technology of China, Hefei, China, 230027
hxj2012@mail.ustc.edu.cn, {zhangwm,hghu2005,ynh}@ustc.edu.cn

Abstract. Signal processing in the encrypted domain becomes a desired technique to protect privacy of outsourced data in cloud. In this paper we propose a double-cipher scheme to implement non-local means denoising in encrypted images. In this scheme, one ciphertext is generated by Paillier scheme which enables the mean-filter, and the other is obtained by a privacy-preserving transform which enables the non-local searching. By the privacy-preserving transform, the cloud can search the similar pixel blocks in the ciphertexts with the same speed as in the plaintexts, so the proposed method can be fast executed. The experimental results show that the quality of denoised images in the encrypted domain is comparable to that obtained in plain domain.

Keywords: Paillier homomorphic encryption, Image denoising, Non-Local Means, Johnson-Lindenstrauss Transform.

1 Introduction

Computable cloud is now prevalent in our daily life, by which customers can remotely store their data so as to enjoy the convenient and effective services [9]. More and more sensitive information such as e-mails and finance data are professionally maintained in data centers. Although outsourcing data storage and processing are quite promising, it still faces a large number of basic challenges, for which the first we need to consider about is security [11]. In fact, many corporations and companies are still afraid of outsourcing their data to the cloud server for the reason that their data may be leaked and cloud sever could abuse their data. So it comes that sensitive data has to be encrypted prior for data privacy and combating unsolicited access.

This leads to a need for techniques of signal processing on encrypted data, which obviously is a difficult problem, because we must have a secure encryption

* This work was supported in part by the Natural Science Foundation of China under Grant 61170234 and Grant 60803155, by the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant XDA06030600, by the Funding of Science and Technology on Information Assurance Laboratory under Grant KJ-13-02, by the National Natural Science Foundation of China (61271271), 100 Talents Program of Chinese Academy of Sciences, and the Fundamental Research Funds for the Central Universities in China (WK2101020005).

R.C.-H. Hsu and W. Shangguang (Eds.): IOV 2014, LNCS 8662, pp. 386–395, 2014.

scheme that allows computations in the encrypted domain. In 1978, Rivest et al. [12] proposed to solve this problem by a scheme called homomorphic encryption that keeps the algebraic relations between plaintexts and ciphertexts. After that, several homomorphic encryption schemes [5], [10], [4] were presented, which process encrypted data with only one homomorphic property, such as addition or multiplication. For instance, the Paillier scheme [10] has additive homomorphism that means one can realize the addition of two plaintext signals by some operations on the two corresponding encrypted signals. A scheme is called fully homomorphic encryption (FHE) if it enables additive and multiplicative homomorphisms at the same time. The first secure FHE scheme is proposed by Gentry [6] in 2009, which, from a theoretical perspective, can solves any privacy-preserving computation problem. However, due to the huge computational complexity and cipher context expansion, FHE scheme is too inefficient to be applied in practice. So far, additive homomorphic encryption is the most popular scheme used by privacy protection community. Based on additive homomorphic encryption, some linear computations have been realized in the encrypted domain, such as discrete fourier transform [2], discrete cosine transform[1], discrete wavelet transform [13] [14] and Walsh-Hadamard transform [15].

An interesting and challenge problem is how to do nonlinear computations in encrypted domain without FHE. In the present paper, we will present a framework to solve a problem of encrypted image denoising that involves some nonlinear operations. Image denoising is one of the most popular image processings, and there exists many classical image denoising algorithms, such as Gauss filter, neighborhood filter, and non-local means (NLM) [3]. Among them NLM and its extensions can reach better performance by exploiting the similarity between the non-local pixel blocks with the current block. However, the Computational complexity of NLM algorithm is very high because it needs to search for the similar pixel blocks. Such hardness of computation is suitable for outsourcing to cloud, but the user may hope to prevent the cloud server from getting the content of the images. Therefore, the cloud should implement denoising in the encrypted images.

In this paper, we try to implement the NLM in the encrypted domain, which consists of two operations, i.e., non-local searching and mean-filter. Mean-filter in encrypted domain can be realized based on additive homomorphic cryptosystem such as Paillier scheme [10], while non-local searching is a nonlinear operation that needs FHE. To avoid FHE, we proposed a double-cipher denoising scheme in which we encrypt the image with two cryptosystems and thus outsource two ciphertexts to the cloud. One ciphertext is generated by Paillier scheme [10] which enables the mean-filter, the other is obtained by a privacy-preserving transform which enables the non-local searching. By the privacy-preserving transform, the cloud can search the similar pixel blocks in the ciphertexts with the same speed as in the plaintexts, so the proposed method can be fast executed.

The rest of the paper is organized as follows. In Section 2, we will give some preliminaries about NLM, Paillier cryptosystem and Johnson-Lindenstrauss (JL) transform. In Section 3, we will describe how to perform image denoising in the encrypted domain in detail. Complexity analysis of our proposed scheme will

be given in section 4. The experimental results are shown in Section 5 and the paper is concluded with discussion in Section 6.

2 Preliminary

2.1 Non-local Means

The NLM method proposed by Buades et al. [3], was widely used in image denoising. Unlike local smoothing methods which only use the local relativity within pixels, NLM tries to exploit the relativity between the pixels that are not close to each other. We briefly introduce NLM below.

We assume the noise is the additive Gaussian noise with mean zero and variance σ^2. So we can describe a discrete noisy image as follows:

$$v(i) = u(i) + n(i) \tag{1}$$

where i is the pixel index in the set I, $v(i)$ is the observed value, $u(i)$ is the original value, and $n(i)$ is the i.i.d.Gaussian noise.

The denoised pixel value at position i is obtained by

$$NL(i) = \sum_{j \in \Omega} w(i,j)v(j) \tag{2}$$

where Ω is the searching window for the similar pixel. The weights $\{w(i,j)\}_j$ are determined by the similarity between the pixel i and j and satisfy $0 \leq w \leq 1$ and $\sum_j w(i,j) = 1$, and the similarity usually can be calculated by the Euclidean distance between the two blocks centered at the i-th and the j-th pixels respectively, such that

$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_2^2}{h^2}}. \tag{3}$$

Herein, N_i denotes the pixel patch centered at the i-th pixel, $\|\cdot\|_2$ is the Euclidean norm, and h is used to control the decay of the weights. $Z(i)$ is normalizing parameter which is defined as

$$Z(i) = \sum_{j \in \Omega} e^{-\frac{\|v(N_i) - v(N_j)\|_2^2}{h^2}}, \tag{4}$$

2.2 Paillier Cryptosystem

Paillier homomorphic cryptosystem [10], is one of the well-known probabilistic and homomorphic schemes with an additive homomorphic property, which is realized as follows.

Initialization. Compute $N = pq$, which p, q was selected as two large prime numbers. Let $\lambda = lcm(p-1, q-1)$ and $g \in Z_{N^2}^*$ where the order of g is a multiple of N. The public key is (N, g) and the private key is λ.

Encryption. Take a plaintext $m \in Z_N$, and a random number (blinding factor) $r \in Z_N^*$. The corresponding ciphertext is

$$c = E_P(m, r) = g^m r^N \ mod \ N^2 \tag{5}$$

Decryption. Let the ciphertext $c \in Z_{N^2}^*$, so the plaintext m is

$$m = D_P(c) = \frac{L(c^\lambda \ mod \ N^2)}{L(g^\lambda \ mod \ N^2)} \ mod \ N \tag{6}$$

where $L(\phi) = (\phi - 1)/N$.

Homomorphism. The additive homomorphism means that the sum of two plaintexts m_1 and m_2 can be obtained by decrypting the product of corresponding ciphertexts.

$$D(E(m_1, r_1) \cdot E(m_2, r_2)) = D(E(m_1 + m_2, r_1 r_2)) = m_1 + m_2 \tag{7}$$

Moreover, let α be a constant and m be a plaintext, then αm can be calculated by decrypting the power of the ciphertext.

$$D(E(m, r)^\alpha) = D((g^m r^N)^\alpha) = D(E(\alpha m, r^\alpha)) = \alpha m \tag{8}$$

2.3 Johnson-Lindenstrauss Transform

Johnson-Lindenstrauss Transform (JL Transform) [7] is a dimension reduction method preserving Euclidean distance, which comes from the following lemma.

Lemma 1. *Given $0 < \varepsilon < 1$, a set Q of n points in R^d, and $k = \Omega(\frac{\log n}{\varepsilon^2})$, there exits a linear map $f : R^d \to R^k$, for any two vectors $\alpha, \beta \in Q$, there exits an inequality below*

$$(1 - \varepsilon)\|\alpha - \beta\|_2^2 \leq \|f(\alpha) - f(\beta)\|_2^2 \leq (1 + \varepsilon)\|\alpha - \beta\|_2^2 \tag{9}$$

Lemma 1 means that we can map some d-dimensional vectors to k-dimensional vectors, and the Euclidean distance between these d-dimensional vectors can be estimated by corresponding k-dimensional vectors. Usually f is defined by $f(\alpha) = P\alpha$ where $P \in R^{k \times d}$ is a random matrix. When $k < d$, one cannot recover α from $P\alpha$, so JL Transform can be used to conceal the elements in vector α and thus used for privacy protection. Kenthapadi et al. [8] analyzed the security of JL Transform and proposed the following private projection (Algorithm 1) and (Algorithm 2).

3 Image Denoising in the Encrypted Domain

In this section, we describe the double-cipher denoising method. Assume that the owner of one image I wants to denoise I with the NLM method. The owner hopes to outsource this work to a cloud server without leaking the content of I.

Algorithm 1. JL Transform-based Private Projection

Input: d-dimensional vector X; $k \times d$ random matrix P; noise parameter ζ.
Output: The projected k-dimensional vector Z.

1. $Y := PX$
2. Construct a random k-dimensional noise vector Δ based on the noise parameter ζ.
3. $Z := Y + \Delta$

Algorithm 2. JL Transform-based distance recover

Input: two k-dimensional vector α and β published in a privacy-preserving manner; Noise parameter ζ;
Output: Estimated squared distance between α and β in the original space.

1. Output $dist^2_{\alpha,\beta} = \|\alpha - \beta\|_2^2 - 2k\zeta^2$.

To do that, the owner encrypts I by JL Transform (Algorithm 1) and Paillier cryptosystem, and gets two encrypted images denoted by $E_J(I)$ and $E_P(I)$ respectively. Then the owner sends $E_J(I)$ and $E_P(I)$ to the cloud. With the help of $E_J(I)$, the cloud executes a non-local filter on $E_P(I)$ and yields a denoised cipher-image $E'_P(I)$ that is sent back to the owner. The owner decrypts $E'_P(I)$ and gets a plain denoised image I'. Next we elaborate the details of each step.

Encryption with JL Transform. When encrypting I with Algorithm 1, the owner takes the random matrix P and noise parameter ζ as the key. For each pixel $v(i)$ of I, take a $s \times s$ block centered at $v(i)$ and permute the block as a s^2-dimensional vector, denoted by N_i. With Algorithm 1, the owner projects N_i into a k-dimensional vector $E_J(N_i)$ which is just the ciphertext of $v(i)$. In other words, by JL Transform, each pixel is encrypted into a k-dimensional vector. For marginal pixels, some elements of the block matrix is blank, so we fill them with the surrounding pixels.

Encryption with Paillier Cryptosystem. For each pixel $v(i)$, take a random number $r_i \in Z_N^*$ and encrypt $v(i)$ by Eq. (5) as

$$E_P(v(i)) = E_P(v(i), r_i) = g^{v(i)} r_i^N \ mod\ N^2. \tag{10}$$

Denosing in Encrypted Images. As shown in Eq. (2) and Eq. (3), to do non-local filter, the cloud should first calculate the weights $\{w(i,j)\}_j$ that are determined by the Euclidean distance between N_i and N_j. Note that JL Transform preserves Euclidean distance, so the cloud can estimate $\|v(N_i) - v(N_j)\|_2$ by the ciphertexts of $v(i)$ and $v(j)$, i.e., $E_J(v(N_i))$ and $E_J(v(N_j))$. Therefore the weights are estimated by

$$w'(i,j) = \frac{1}{Z(i)} e^{-\frac{\|E_J(v(N_i)) - E_J(v(N_j))\|_2^2}{h^2}}, \tag{11}$$

where h is the decaying parameter and the normalizing parameter $Z(i)$ is obtained by

$$Z(i) = \sum_{j \in \Omega} e^{-\frac{\|E_J(v(N_i)) - E_J(v(N_j))\|_2^2}{h^2}}. \tag{12}$$

With weights $\{w'(i,j)\}_j$, the cloud filters the other encrypted image $E_P(I)$ as follows. For each ciphertext $E_P(v(i))$, the filtered value $E'_P(v(i))$ is

$$E'_P[v(i)] = \prod_{j \in \Omega} E_P[v(j)]^{w'(i,j)} \tag{13}$$

Collecting all $E'_P(v(i))$, the cloud yields a denoised encrypted image $E'_P(I)$ that is sent back to the owner of I.

Decryption. After receiving $E'_P(I)$, the image owner decrypts it pixel by pixel. According to the homomorphism Eq. (7) and Eq. (8), the pixel $E'_P(v(i))$ is decrypted as

$$NL'(i) = D_P[\prod_{j \in \Omega} E_P[v(j)]^{w'(i,j)}] = \sum_{j \in \Omega} w'(i,j)v(j) \tag{14}$$

Compare Eq. (2) with Eq. (14), we conclude that the denoising result obtained in the encrypted image is similar with that obtained in the plain image because JL Transform can preserve Euclidean distance and thus $w'(i,j)$ is a good estimation of $w(i,j)$.

Note that the wights $\{w'(i,j)\}_j$ are real numbers, but to calculate Eq. (13) and Eq. (14) according to Paillier cryptosystem, the values $\{w'(i,j)\}_j$ have to be quantified as integer numbers. The quantization process can be expressed as

$$W(i,j) = \lfloor Aw'(i,j) \rceil \tag{15}$$

where $\lfloor \cdot \rceil$ is the rounding function and A is the scaling factor. For simplicity, we rewrite Eq. (15) as follow:

$$W(i,j) = Aw'(i,j) + \varepsilon_{w_j} \tag{16}$$

where $|w'(i,j)| \leq 1$ and ε_{w_j} is the error caused by quantization with $|\varepsilon_{w_j}| \leq 1/2$.

Replacing $w'(i,j)$ by $W(i,j)$, the decrypted result Eq. (14) will be changed to

$$NL''(i) = \sum_{j \in \Omega} W(i,j)v(j) = \sum_{j \in \Omega} (Aw'(i,j) + \varepsilon_{w_j})v(j). \tag{17}$$

From $NL''(i)$, the owner of I can estimate $NL'(i)$ by

$$\overline{NL'(i)} = \frac{NL''(i)}{A} = NL'(i) + \frac{\sum_{j \in \Omega} \varepsilon_{w_j} v(j)}{A} \tag{18}$$

The last item $\frac{\sum_{j \in \Omega} \varepsilon_{w_j} v(j)}{A}$ is the error caused by quantization, which can be controlled by selecting a large enough parameter A. Note that, to get the accurate result of $NL''(i)$ by decryption in the Paillier cryptosystem, we have to limit $NL''(i) < N$. In other words, the value of A cannot be as large as possible, and must be chosen properly according to the settings of Paillier cryptosystem.

Table 1. Result of simulation

house					parrot				
Index	d	k	PSNR before	PSNR after	Index	d	k	PSNR before	PSNR after
(c)	25	-	22.09	32.39	(h)	25	-	22.09	29.03
-	25	20	22.09	30.93	-	25	20	22.09	27.93
-	25	18	22.09	30.54	-	25	18	22.09	27.76
(d)	25	16	22.09	30.03	(i)	25	16	22.09	27.51
-	25	12	22.09	28.83	-	25	12	22.09	26.86
(e)	25	9	22.09	28.08	(j)	25	9	22.09	26.36
peppers					cameraman				
Index	d	k	PSNR before	PSNR after	Index	d	k	PSNR before	PSNR after
(m)	25	-	22.09	30.15	(r)	25	-	22.09	29.48
-	25	20	22.09	28.61	-	25	20	22.09	28.45
-	25	18	22.09	28.30	-	25	18	22.09	28.24
(n)	25	16	22.09	27.95	(s)	25	16	22.09	27.94
-	25	12	22.09	27.03	-	25	12	22.09	27.29
(o)	25	9	22.09	26.40	(t)	25	9	22.09	26.82

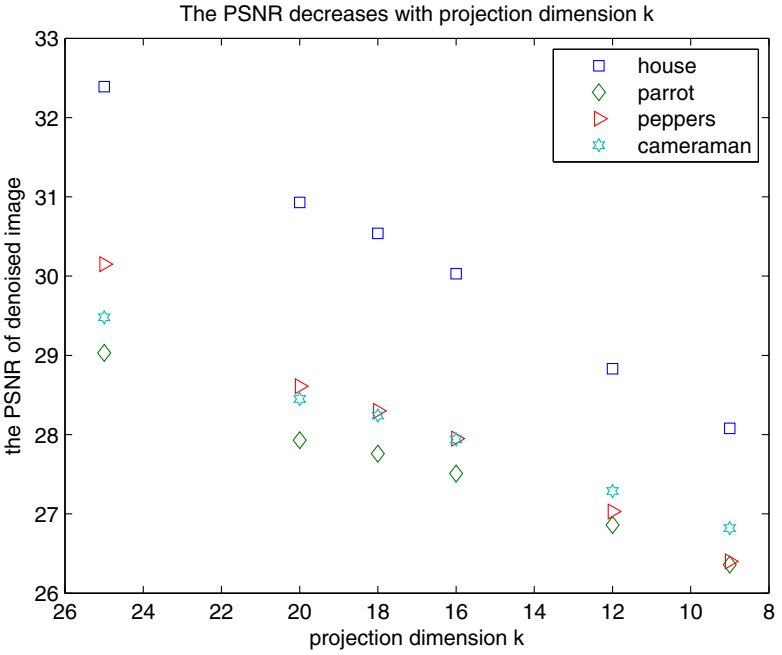

Fig. 1. The PSNR of denoised image decreases with the projection dimension k

4 Complexity Analysis

Now we estimate the computational complexity of our proposed scheme. Our scheme has two parts, first part is non-local searching, and second part is mean-filter in the encrypted domain.

So, we analysis the computational complexity from the first part. Here we refer to the literature [3]. If we use a similarity window of size $s \times s$, so for a image of size $n \times n$, the computational complexity is $n^2 \times s^2 \times n^2 = s^2 \times n^4$. So the computational complexity of the original algorithm is $O(n^4)$. If we also restrict the search of the similarity windows in the size of $L \times L$. The entire computational complexity of the algorithm is $n^2 \times s^2 \times L^2$.

Now we discuss computational complexity of the second part. In the second part we only consider about the Eq. (14), there are only modular exponentiation and modular multiplication operation in the encrypted domain. Hence, we can use the number of modular exponentiation (m_{exp}) and modular multiplication (m_{mul}) to evaluate the computational complexity. As we defined before, for a $n \times n$ image, the similarity window size is $s \times s$, and the size of the search window is $L \times L$. Hence for a single pixel, the number of modular exponentiation is $L^2 \times W(i,j) = L^2 \times Aw'(i,j)$, and the number of modular multiplication is $L^2 - 1$. For the whole image, the computation of modular exponentiation and modular multiplication are $n^2 \times L^2 \times Aw'(i,j)$ and $n^2 \times (L^2 - 1)$, respectively. When the search window size is $n \times n$, the total number of modular exponentiation is $n^2 \times n^2 \times Aw'(i,j) = n^4 \times Aw'(i,j)$ and the total number of modular multiplication is $n^2 \times (n^2 - 1)$ in encrypted domain.

5 Experiments

In our experiments, for security reason, the product of two primes ($N = p \times q$) for the Paillier cryptosystem is longer than 1024 bits. For $A \times NL'(i) < N$, we can get $A < 2^{1024}/255 = 2^{1016}$, For $A = 2^{30}$, the error $\frac{\sum_{j \in \Omega} \varepsilon_{w_j} v(j)}{A}$ can be less than $L^2/2^{23}$, so we set the size of searching window $L \times L = 21 \times 21$, the size of similar block $d = s \times s = 5 \times 5$, Gaussian noise $\sigma = 20$, the decaying parameter $h = 0.75\sigma$, the noise parameter $\zeta = 1$, and the images "house, parrot, peppers, cameraman" are used as an example. In Fig. 2, the first column is the original image sized of 256×256, the second column is the images after adding $(0, 20^2)$ Gaussian noise, the third column is the images denoised in the plaintext, the fourth and fifth column are the images denoised in the encrypted domain with projection dimension $k = 16, 9$, respectively. The peak signal-to-noise ratios (PSNR) of the noisy images and the denoised images are listed in Table 1. It can be seen from Fig. 2 and Table 1, the denoised images in encrypted domain can achieve similar quality as done in plain domain.

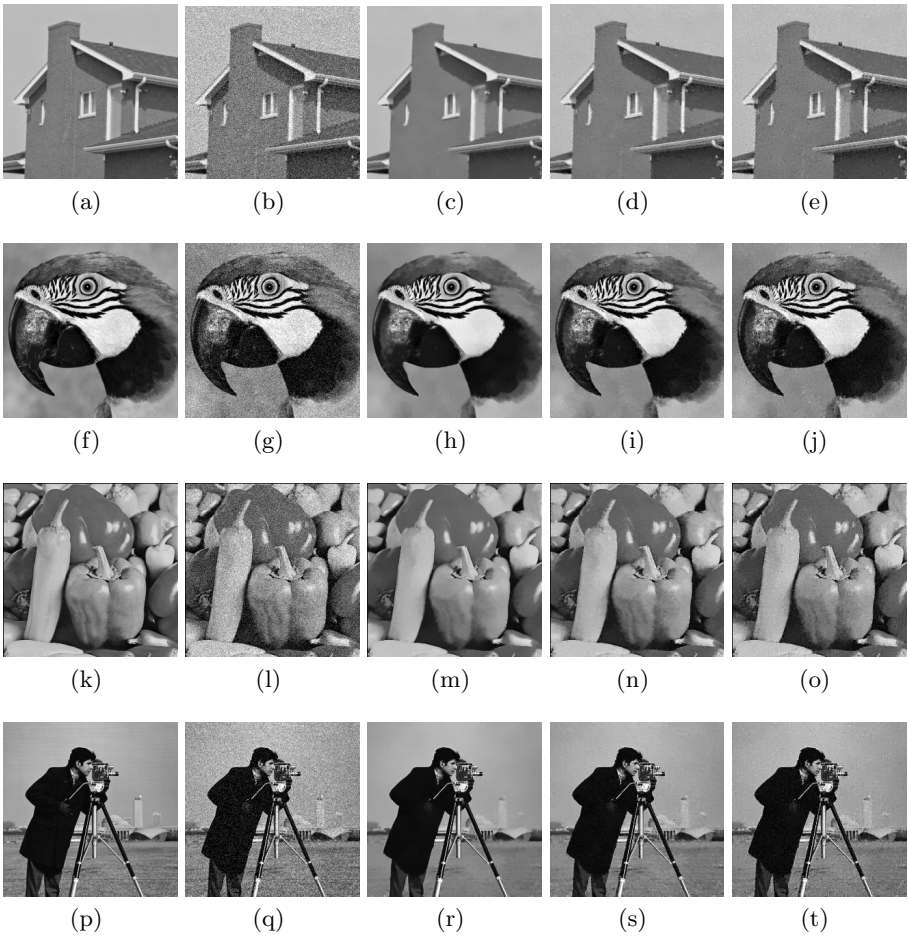

Fig. 2. (1) the first column is the original images; (2) the second column is the images after adding $(0, 20^2)$ Gaussian noise; (3) the third column is the images denoised in the plaintext; (4) the fourth column is the images denoised in the encrypted domain with projection dimension $k = 16$; (5) the fifth column is the images denoised in the encrypted domain with projection dimension $k = 9$.

6 Conclusion and Future Work

In this paper, we propose a double-cipher denosing method in encrypted images, which enables the cloud to implement NLM with the ability of preserving privacy of the image contents. This double-cipher scheme is a novel framework to deal with nonlinear operations in the encrypted domain avoiding FHE. In the present scheme, the nonlinear operation is searching for similar blocks which is realized based on the privacy projection. As shown in Table 1 and Figure 1, the PSNR of denosed image decreases with the projection dimension k. In fact, the secure

level of the privacy projection increases with decreasing k. However, because of the limits of pages, the security of the proposed scheme is not analyzed, which will be studied in our further work.

References

1. Bianchi, T., Piva, A., Barni, M.: Encrypted domain dct based on homomorphic cryptosystems. EURASIP Journal on Information Security 2009, 1 (2009)
2. Bianchi, T., Piva, A., Barni, M.: On the implementation of the discrete fourier transform in the encrypted domain. IEEE Transactions on Information Forensics and Security 4(1), 86–97 (2009)
3. Buades, A., Coll, B., Morel, J.: A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation 4(2), 490–530 (2005)
4. Damgård, I., Jurik, M.: A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 119–136. Springer, Heidelberg (2001)
5. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory 31(4), 469–472 (1985)
6. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
7. Johnson, W., Lindenstrauss, J.: Extensions of lipschitz mappings into a hilbert space. Contemporary Mathematics 26(189-206), 1 (1984)
8. Kenthapadi, K., Korolova, A., Mironov, I., Mishra, N.: Privacy via the johnson-lindenstrauss transform. arXiv preprint arXiv:1204.2606 (2012)
9. Mell, P., Grance, T.: Draft nist working definition of cloud computing. Referenced on June 3, vol. 15 (2009)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
11. Ren, K., Wang, C., Wang, Q.: Security challenges for the public cloud. IEEE Internet Computing 16(1), 69–73 (2012)
12. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. Foundations of Secure Computation 32(4), 169–178 (1978)
13. Zheng, P., Huang, J.: Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. In: Proceedings of the 19th ACM International Conference on Multimedia, pp. 413–422. ACM (2011)
14. Zheng, P., Huang, J.: Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain (2013)
15. Zheng, P., Huang, J.: Walsh-hadamard transform in the homomorphic encrypted domain and its application in image watermarking. In: Kirchner, M., Ghosal, D. (eds.) IH 2012. LNCS, vol. 7692, pp. 240–254. Springer, Heidelberg (2013)