Walsh-Hadamard Transform in the Homomorphic Encrypted Domain and Its Application in Image Watermarking

Peijia Zheng and Jiwu Huang

School of Information Science and Technology
Sun Yat-Sen University
Guangzhou, 510006, China
zhengpj@mail2.sysu.edu.cn, isshjw@mail.sysu.edu.cn

Abstract. How to embed and/or extract watermarks on encrypted images without being able to decrypt is a challenging problem. In this paper, we firstly discuss the implementation of Walsh-Hadamard transform (WHT) and its fast algorithm in the encrypted domain, which is particularly suitable for the applications in the encrypted domain for its transform matrix consists of only integers. Then by modifying the relations among the adjacent transform coefficients, we propose an WHT-based image watermarking algorithm in the encrypted domain. Due to the constrains of the encryption, extracting a watermark blindly from an encrypted image is not a easy task. However, our proposed algorithm possesses the characteristics of blind watermark extraction both in the decrypted domain and the encrypted domain. This means neither the plain image nor its encrypted version is required for the extraction. The experiments demonstrate the validity and the advantages of our proposed method.

Keywords: Secure signal processing, watermark, homomorphic encryption, signal processing in the encrypted domain, Walsh Hadamard transform.

1 Introduction

Watermarking is an method to protect the copyright of digital media by hiding proprietary information in media. The security of watermarking is a challenging problem in the watermarking community. Many efforts focusing on watermark security have been reported in literature [1] [2]. In fact, there are at least two problems on the security. The first one is the security of the original media under being watermarked. Almost all the existing watermark schemes accomplish the watermark embedding and extraction on the plain media. Hence, the watermark embedder must be the owner of the plain media or the trusted third party, in order to make sure the original media is not exposed to the untrusted party. The second one is the security of the watermark scheme itself. For example, how to prevent illegal watermark embedding, extracting, and removal.

M. Kirchner and D. Ghosal (Eds.): IH 2012, LNCS 7692, pp. 240–254, 2013.

Though there are some reports on integrating watermark embedding and encrypting [3] [4], it causes additional constraints to the watermarking algorithm, meanwhile. Some works [5] have been proposed to solve the first problem, however, the visual quality of the watermarked images are not so good as expected. Single processing in the encrypted domain, also referred to as secure signal processing (SSP), provides another way to solve the first problem. This new technology allows one to manipulate the encryption data by means of signal processing without decrypting.

There have been some related works on secure signal processing over the past few years. An interactive buyer-seller watermarking protocol for invisible watermarking was proposed in [6], where the seller does not get to know the exact watermarked copy that the buyer receives. Bianchi *et al.* [7] conducted an investigation on the implementation of the discrete Fourier transform (DFT) as well as the fast Fourier transform (FFT) on encrypted signals. A data encrypting method, which packs several samples as a single one, was proposed by Troncoso-Pastoriza *et al.* [8], and later generalized by Bianchi *et al.* [9]. In [10] [11], the authors proposed schemes for privacy-preserving face recognition by using the Paillier cryptosystem. Zheng *et al.* [12] presented a new technique to implement the discrete wavelet transform (DWT) and Multiresolution Analysis (MRA) in the encrypted domain. They also provided a new method to handle the data expansion without decrypting. Barni *et al.* gave a privacy-preserving fingercode authentication in [13]. In [14], they proposed a system for the secure classification of ECG (electrocardiogram) signals with branching programs and neural networks.

Due to the limitation of the encryption, it is very difficult, sometimes impossible, to transplant the existing mature watermark scheme to the encrypted domain. Thus it is meaningful to design a new image watermark scheme under the constraints of the homomorphic encrypted domain. Generally, the watermark algorithms based on transform domain are more robust than the others. Owing to the quantization error, DFT [7] and DCT [15] in the encrypted domain will bring a noise to the plain reconstructed image, which may decrease the visual effect of the watermarked image. Since the transform matrix of the Walsh-Hadamard transform (WHT) contains only $+1$ and -1, one can avoid the quantization error of its implementation in the encrypted domain. Therefore WHT is particularly suitable to be used as a transform method for image watermarking in the encrypted domain.

This paper addresses the issue of image watermarking in the encrypted domain. Firstly, we describe a framework for performing WHT in a homomorphic encrypted domain. Secondly, we develop a WHT-based image watermarking scheme and transplant it to the encrypted domain. The proposed scheme possesses the characteristics of blind watermark extraction both in the decrypted domain and the encrypted domain. Finally, we conduct several experiments to substantiate the proposed scheme. Our technique can be applied to other applications where a secure watermarking algorithm is required.

The remainder of this paper is organized as follows. In Section 2, we discuss the implementation of WHT in the encrypted domain. In Section 3, we propose the blind-extraction image watermarking algorithm in the encrypted domain. Section 4 gives some experiments on the image watermarking algorithm. We conclude the paper and provide suggestions for future work in Section 5.

2 Walsh-Hadamard Transform in the Encrypted Domain

WHT is used widely in the field of signal processing. The transform matrix of WHT contains only ± 1, and no multiplications are required in the computation. Thus WHT is more efficient than other orthogonal transformations, such as DFT or DCT. Another advantage of WHT has is that WHT will not bring the quantization error in the encrypted domain. WHT can therefore be perfectly reconstructed in the encrypted domain, which is shown in Section 2.3. Hence, in contrast to DFT and DCT, WHT is particularly suitable for image watermarking in the encrypted domain. Since the implementation of WHT in the encrypted domain has not been reported yet, we present the implementation first.

2.1 Homomorphic Cryptosystem

The homomorphic cryptosystem [16] is an encryption function which allows one to operate the ciphertexts without decrypting. Specifically, suppose $\mathcal{D}[\cdot]$ and $[\![\cdot]\!]$ are the decrypting operator and encrypting operator, respectively. If m_1 and m_2 are any two plaintexts, we have

$$\mathcal{D}\left[[\![m_1]\!] \star [\![m_2]\!]\right] = m_1 * m_2 \tag{1}$$

where operator '$\star$' and '$*$' are the algebraic operations performed in the ciphertext space and the plaintext space, respectively.

For convenience, we use the Paillier cryptosystem as data encryption method in this paper. We refer to [17] for the detailed definition of the Paillier cryptosystem. Based on the definition, we have the additive homomorphic properties as

$$\mathcal{D}\left[[\![m_1]\!]\,[\![m_2]\!] \bmod N^2\right] = m_1 + m_2 \bmod N, \tag{2}$$

$$\mathcal{D}\left[[\![m_1]\!]^{m_2} \bmod N^2\right] = m_1 m_2 \bmod N. \tag{3}$$

The Paillier cryptosystem also has the self-blinding property, i.e.,

$$\mathcal{D}\left[[\![m_1]\!]\, r^N \bmod N^2\right] = m_1 \bmod N \tag{4}$$

where r is a random element in $\mathbb{Z}_N^*$. $\mathbb{Z}_N^*$ consists of all the integers in $\mathbb{Z}$ which are relative prime with N. The self-blinding property means that every ciphertext can be publicly changed into another ciphertext which has the same plaintext.

These properties will be applied in the following sections to perform the implementation of WHT and image watermarking in the encrypted domain.

2.2 Integer Approximation and Evaluation

Let us consider the image $I(x, y)$, with the size of $M \times M$, where M is assumed to be the power of two. The 2D WHT of natural ordering is defined as

$$X(k,l) = \frac{1}{M} \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \mathbf{H}_\mu(k,x)\, I(x,y)\, \mathbf{H}_\mu(y,l), \quad k,l = 0,1,\ldots,M-1 \tag{5}$$

where $\mu = \log_2 M$ and $\mathbf{H}_\mu$ denotes the Hadamard transform matrices. $\mathbf{H}_\mu$ can be generated by the core matrix

$$\mathbf{H}_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{6}$$

and the Kronecker product recursion

$$\mathbf{H}_\mu = \mathbf{H}_1 \otimes \mathbf{H}_{\mu-1} = \begin{pmatrix} \mathbf{H}_{\mu-1} & \mathbf{H}_{\mu-1} \\ \mathbf{H}_{\mu-1} & -\mathbf{H}_{\mu-1} \end{pmatrix} \tag{7}$$

where $\otimes$ is the Kronecker product operator. According to the method in [18], one can easily obtain WHT of sequency ordering and other orderings by rearranging the outputs (5). Therefore we will focus on WHT of natural ordering in the following.

Since all the plaintexts and the ciphertexts are represented by integers in the cryptosystem, the signal must also be represented by integers too. Obviously, all the elements of $I(x, y)$ are integers between 0 and 255, i.e., $I(x, y) \in \mathbb{Z}_{256}$. However, the transform coefficients of an image may be negative, and we still need to consider the problem of representing the negative integers in the cryptosystem. Suppose N is the modulus of the cryptosystem. We let $N \geq 2 \sup\{|S(k)|\} + 1$, where $\sup\{\cdot\}$ denotes the least upper bound operator performed on a sequence, and $S(k)$ is the plain value of the processed result in the encrypted domain.

According to the above discussion, we give the definition of the integer approximation of the 2D WHT as

$$V(k,l) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \mathbf{H}_\mu(k,x)\, I(x,y)\, \mathbf{H}_\mu(y,l), \quad k,l = 0,1,\ldots,M-1. \tag{8}$$

Since all the operations are either integer additions or integer subtractions, (8) can be implemented in the encrypted domain by using the homomorphic properties. In the case that the input signal is encrypted with the Paillier cryptosystem, by means of the equations (2) and (3), the implementation of the 2D WHT in the encrypted domain is given as

$$[\![V(k,l)]\!] = \prod_{x=0}^{M-1} \prod_{y=0}^{M-1} [\![I(x,y)]\!]^{\mathbf{H}_\mu(k,x)\mathbf{H}_\mu(y,l)} \triangleq \tilde{V}(k,l), \quad k,l = 0,1,\ldots,M-1 \tag{9}$$

where all the multiplications and exponentiations are carried out under N^2.

The definition of the inverse WHT (IWHT) is identical to the forward WHT. If $X'(k,l)$ is the input transform coefficients, which may not be identical to $X(k,l)$, then the reconstructed image is given as

$$\hat{I}(x,y) = \frac{1}{M} \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} \mathbf{H}_\mu(x,k) X'(k,l) \mathbf{H}_\mu(l,y), \quad x,y = 0,1,\ldots,M-1. \tag{10}$$

A similar approach leads to the definition of the integer IWHT. Assuming we have already obtained the integer 2D WHT coefficients $V'(k,l)$, the integer approximation of the 2D IWHT is defined as

$$I'(x,y) = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \mathbf{H}_\mu(x,k) V'(k,l) \mathbf{H}_\mu(l,y), \quad x,y = 0,1,\ldots,M-1, \tag{11}$$

where $V'(k,l)$ is corresponding to $X'(k,l)$. Since all the input arguments are integers, (11) can be computed in the encrypted domain as

$$[\![I'(x,y)]\!] = \prod_{k=0}^{M-1} \prod_{l=0}^{M-1} \tilde{V}'(k,l)^{\mathbf{H}_\mu(x,k)\mathbf{H}_\mu(l,y)} \triangleq \tilde{I}'(x,y), \quad x,y = 0,1,\ldots,M-1. \tag{12}$$

For the sake of simplicity, we use WHT-ed and IWHT-ed to denote the implementation of WHT and IWHT in the encrypted domain, respectively.

2.3 Data Recovery and Upper Bound

In order to implement WHT and IWHT in the encrypted domain by using (9) and (12), we need to consider some issues. Since all the calculations of (9) and (12) are in the finite ring $\mathbb{Z}_N$, the plain value of the processed result S must not be larger than N. Thus we should find a upper bound on S. Let us consider the implementation of WHT in the encrypted domain first. It is obvious that

$$\begin{aligned} \mathcal{D}\left[\tilde{V}(k,l)\right] &= V(k,l) \mod N = MX(k,l) \mod N \\ &\triangleq Z(k,l). \end{aligned} \tag{13}$$

However, $Z(k,l)$ may sometimes be negative. Taking the negative coefficients into account, the recovery condition is given as

$$2M \sup_{k,l} \{|X(k,l)|\} + 1 < N. \tag{14}$$

Moreover, we must find a method to recover every value from the decryption of the output. Actually, under the condition (14), $X(k,l)$ can be obtained directly from $\tilde{V}(k,l)$ as

$$X(k,l) = \begin{cases} \dfrac{\mathcal{D}\left[\tilde{V}(k,l)\right]}{M}, & \text{for } Z(k,l) < N/2 \\ \dfrac{\mathcal{D}\left[\tilde{V}(k,l)\right] - N}{M}. & \text{for } Z(k,l) > N/2 \end{cases} \tag{15}$$

As for the inverse WHT in the encrypted domain, a similar approach leads to the upper bound of the reconstructed image. By using the homomorphic property, we have

$$\begin{aligned}\mathcal{D}\left[\tilde{I}'(x,y)\right] &= \sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\mathbf{H}_\mu(x,k)\,\mathcal{D}\left[\tilde{V}'(k,l)\right]\mathbf{H}_\mu(l,y)\\ &= M\sum_{k=0}^{M-1}\sum_{l=0}^{M-1}\mathbf{H}_\mu(x,k)\,X'(k,l)\,\mathbf{H}_\mu(l,y)\ \bmod\ N\\ &= M^2\hat{I}(x,y)\ \bmod\ N \triangleq Y(x,y).\end{aligned} \tag{16}$$

Specifically, if $V'(k,l) = V(k,l)$, then $Y(x,y) = M^2 I(x,y)$. It implies that any image can be completely reconstructed in the encrypted domain, i.e. perfect reconstruction. The recovery condition of the reconstructed image is given as

$$2M^2 \sup_{x,y}\left\{\hat{I}(x,y)\right\} + 1 < N. \tag{17}$$

When condition (17) is satisfied, we can obtain $\hat{I}(x,y)$ from the $\tilde{I}'(x,y)$ as

$$\hat{I}(x,y) = \begin{cases} \dfrac{\mathcal{D}\left[\tilde{I}'(x,y)\right]}{M^2}, & \text{for } Y(x,y) < N/2 \\ \dfrac{\mathcal{D}\left[\tilde{I}'(x,y)\right] - N}{M^2}. & \text{for } Y(x,y) > N/2 \end{cases} \tag{18}$$

Obviously, $\sup_{x,y}\{I(x,y)\} = 255$. The first element of matrix $X(k,l)$ is the sum of all the pixels in $I(x,y)$. Thus we have $\sup_{k,l}\{|X(k,l)|\} = 255M^2$. In the case of $V'(k,l) = V(k,l)$, by combining (14) and (17), the final recovery condition can be given as

$$N > \max\{510M^3, 510M^2\} = 510M^3. \tag{19}$$

According the above analysis, an interesting phenomenon may be obtained. In contrast to the implementation of WHT in the plain domain, the implementation in the encrypted domain will expand the plain value of the expected value. The expanding factor depends on two parameters, the dimension and the length of the input signal. More specifically, each implementation of 2D WHT-ed and 2D IWHT-ed will expand the plain value by a fixed factor M. Generally, the image size M is only tens of bits for real images, while N should be 1024 bits according to [17]. Therefore the expanding factor M is negligible compared with N, and the WHT-based applications can be well transplanted to the encrypted domain, without considering the data overflow.

2.4 Fast WHT in the Encrypted Domain

2D WHT is a separable transform, i.e., a 2D transform which can be decomposed into two 1D transforms. Specifically, performing 2D WHT on $I(x,y)$ is equivalent

to performing 1D WHT on the each column of $I(x, y)$ first and then performing 1D WHT on the each row to the former result. Hence, we focus on the fast algorithm of 1D WHT-ed in this paper.

In fact, the computational complexity of WHT can be reduced from M^2 to $M \log M$ by a fast algorithm [18]. The fast algorithm follows the recursive definition of the Hadamard matrix (7). Similar to FFT, the fast WHT recursively breaks down a WHT of size M into two smaller WHTs of size $M/2$. Therefore there are totally $\log_2 M$ stages of breaking down by means of the fast algorithm. Since there are only M additions/subtractions at each stage, there are totally $M \log_2 M$ additions/subtractions for the fast WHT. More specifically, every two coefficients are obtained at one stage from another two coefficients at the previous stage by using only addition or substraction. That is, by omitting the scaling factor, the fast WHT at i-th stage can described as

$$v^i(k_0) = v^{i-1}(k_0) + v^{i-1}(k_1) \tag{20}$$

$$v^i(k_1) = v^{i-1}(k_0) - v^{i-1}(k_1) \tag{21}$$

where $v^i(k_0)$ and $v^i(k_1)$ are the two coefficients obtained at i-th stage, $i = 1, 2, \ldots, \log_2 M$. The indices k_0, k_1 are integers which vary between 0 and $M-1$.

By using the homomorphic properties, we implement the fast WHT at i stage in the encrypted domain as

$$[\![v^i(k_0)]\!] = [\![v^{i-1}(k_0)]\!]\,[\![v^{i-1}(k_1)]\!], \tag{22}$$

$$[\![v^i(k_1)]\!] = [\![v^{i-1}(k_0)]\!]\,[\![v^{i-1}(k_1)]\!]^{-1}. \tag{23}$$

Suppose $\{[\![v^{\log_2 M}(k)]\!]\}$ are the encrypted coefficients obtained at the final stage. After a simple deduction, we get the relationship between the direct WHT-ed and the fast WHT-ed as

$$[\![v^{\log_2 M}(k)]\!] = \tilde{v}(k) \tag{24}$$

where $\tilde{v}(k)$ is the coefficient obtained by the direct WHT-ed. Since the definition of IWHT is identical to that of WHT, the method described above can also be used as a fast algorithm to implement IWHT in the encrypted domain.

3 Blind Image Watermarking in the Encrypted Domain

In order to embed a watermark on an encrypted image, we should tackle two challenging issues. The first one is how to achieve the goal of blind watermark exaction. Since the original image is protected by the encryption, it is not practical to involve the plain original image into the extraction. The second one is how to evaluate the visual quality of the watermarked image. Since the input image is in the encrypted form and the embedder don't have the decrypting key, it is difficult for him/her to determine whether the visual effect of the watermarked images is good or bad.

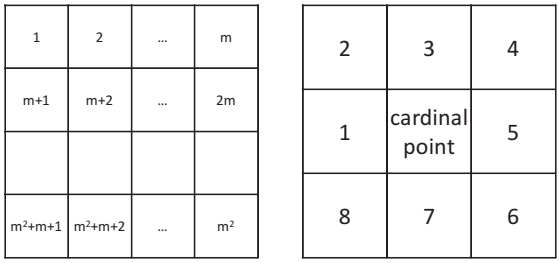

Fig. 1. The relationship between the reference positions and the values e_j and d_j

3.1 Watermark Embedding

The embedding domain, e.g. the spatial domain or the transform domain, plays a crucial role in robust performance and the visual quality of the watermarked image. In order to make the watermark scheme more robust, we choose to embed the watermark in the transform domain rather than the spatial domain. We describe the algorithms in the plain domain first and then give its implementation in the encrypted domain.

Watermarking in the Plain Domain. Suppose the embedding message is a binary signal $\mathbf{w} = \{w_1, w_2, \ldots, w_n\}$, where $w_j \in \{0, 1\}$. Our watermarking algorithm in the plain domain can be described as follows.

(1) To segment the original image $I(x, y)$ into non-overlapping blocks of $m \times m$. m is assumed to be an integral power of two. Thus there are totally $M_b = (M/m)^2$ blocks after the segmentation.
(2) To perform WHT of sequency ordering on each segmented block and obtain the transform coefficient blocks, denoted by $\{V_j\}_1^{M_b}$. In order to protect the watermarked images from illegal extraction, a random number sequence is introduced to control the embedding. Denote the random number sequence by $\mathbf{a} = \{a_1, a_2, \ldots, a_n\} \in \mathscr{P}(\{1, 2, \ldots, M_b\})$, where $\mathscr{P}(\cdot)$ denotes the power set of a set. Select n coefficient blocks from $\{X_j\}_1^{M_b}$ according to $\mathbf{a}$ in sequential scan order. The selected blocks are denoted by $\{X_1, X_2, \ldots, X_n\}$.
(3) To choose two random sequences $\mathbf{e} = \{e_1, e_2, \ldots, e_n\}$ and $\mathbf{d} = \{d_1, d_2, \ldots, d_n\}$, where $e_j \in \{2, 3, \ldots, m^2\}$ and $d_j \in \{1, 2, \ldots, 8\}$. e_j denotes one special point in block X_j, called the cardinal point of X_j. The value of e_j corresponds to the position in X_j in sequential scan order. Whereas d_j stands for the orientation which surrounds the cardinal point. The value of d_j increases as we revolve clockwise around the cardinal point. We show the corresponding relation between the values of e_j and d_j and the positions in block X_j in Fig. 1.
(4) In the selected block X_j, we choose the cardinal point according to the value of e_j. The cardinal point of X_j is $X_j(k_0, l_0) = V_j(\lfloor e_j/m \rfloor, e_j \bmod m)$. We use $X_j(k_1, l_1)$ to denote the adjacent point surrounding $X_j(k_0, l_0)$, with respect

to d_j. The watermark is embedded by modifying the transform coefficient $X_j(k_1, l_1)$. The detailed modification of the coefficient $X_j(k_1, l_1)$ is given as

$$X_j(k_1, l_1) = \begin{cases} X_j(k_0, l_0), & \text{if } w_j = 0 \\ X_j(k_0, l_0) + \alpha_j. & \text{if } w_j = 1 \end{cases} \tag{25}$$

where $\alpha_j \in \mathbb{N}^*$ is a locally adjustable amplitude factor. Since the other coefficients is quite small compared with the $V_j(1, 1)$, this modification is actually very slight. We use X_j^* to denote the coefficient block which has been modified.

(5) To perform IWHT on all the coefficient blocks, including the modified blocks and the unmodified ones, in order to output the watermarked image, denoted by $I_w(x, y)$. In order to keep format compliance, $I_w(x, y)$ will undergo the quantization process. The quantized watermarked image is denoted by $I_{w,256}(x, y)$.

The triple $(\mathbf{a}, \mathbf{e}, \mathbf{d})$ is the secret key of the watermark algorithm. It determines the positions where the watermark is embedded. It will be sent to the watermark extractor and take part in the process of watermark extraction.

Watermarking in the Encrypted Domain. By using the homomorphic properties of the cryptosystem, the watermark embedding algorithm can also be implemented in the encrypted domain. Suppose the input to the watermark embedder is an encrypted image $[\![I(x, y)]\!]$. The embedder knows nothing about the plain image while still try to embed $\mathbf{w}$ in the plain image. Actually the watermark embedding can be carried out in the encrypted domain without an interactive protocol. The detail of the implementation is given as follows.

We segment the encrypted image $[\![I(x, y)]\!]$ into $(M/m)^2$ blocks of $m \times m$. Then we apply WHT-ed to each block. According to the random integer sequence $\mathbf{a}$, n blocks are selected for watermark insertion. We denote those selected blocks by $\{\tilde{V}_1, \tilde{V}_2, \ldots, \tilde{V}_n\}$. In the block $\tilde{V}_j$, the cardinal point $\tilde{V}_j(k_0, l_0)$ is chosen according to the value of e_j, i.e., $\tilde{V}_j = \tilde{V}_j(\lfloor e_j/m \rfloor, e_j \bmod m)$. With respect to the value of d_j, we choose the adjacent point of $\tilde{V}_j(k_0, l_0)$, denoted by $\tilde{V}_j(k_1, l_1)$. Then the watermark embedding in the encrypted domain can be accomplished by modifying the encrypted coefficients. Specifically, the coefficient modification of j-th selected block can be given as

$$\tilde{V}_j(k_1, l_1) = \begin{cases} \tilde{V}_j(k_0, l_0)\, r^N \mod N^2, & \text{if } w_j = 0 \\ \tilde{V}_j(k_0, l_0)\, [\![\alpha_j m]\!] \mod N^2. & \text{if } w_j = 1 \end{cases} \tag{26}$$

where r is a random number chosen in $\mathbb{Z}_N$. We use $\tilde{V}_j^*$ to denote the encrypted coefficient block which has been modified. After modifying the coefficients, we perform IWHT-ed on all the coefficient blocks, including both the modified blocks and the unmodified ones. The processed encrypted image, i.e. the encrypted version of the watermarked image, is denoted by $\tilde{I}_w(x, y)$. The above manipulations only use the homomorphic properties of the encryption, and rely on no interactive protocol.

We now explain why we call $\tilde{I}_w(x,y)$ the encrypted version of $I_w(x,y)$. Since the homomorphic cryptosystem possesses the self-blinding property (4), by using the equation (13), we have

$$\mathcal{D}\left[\tilde{V}_j\left(k_0,l_0\right)r^N \mod N^2\right] = mX_j^*\left(k_0,l_0\right). \tag{27}$$

Similarly, by using the homomorphic properties (2) and equation (13) , we have

$$\mathcal{D}\left[\tilde{V}_j\left(k_0,l_0\right)[\![\alpha_j m]\!] \mod N^2\right] = mX_j^*\left(k_0,l_0\right) + m\alpha_j. \tag{28}$$

Hence, by combining the two equations (27) and (28), we obtain

$$\mathcal{D}\left[\tilde{V}_j^*\left(k_1,l_1\right)\right] = mX_j^*\left(k_1,l_1\right). \tag{29}$$

Since $\tilde{I}_w(x,y)$ is obtained by performing 2D IWHT-ed on all the encrypted coefficient blocks, we can get the relationship between $\tilde{I}_w(x,y)$ and $I_w(x,y)$ by using (16). Specifically, the relationship can be obtained as

$$\mathcal{D}\left[\tilde{I}_w\left(x,y\right)\right] = m^2 I_w\left(x,y\right) \mod N. \tag{30}$$

This means that the image $\mathcal{D}[\tilde{I}_w(x,y)]$ is the same as the image $I_w\left(x,y\right)$ in the finite ring $\mathbb{Z}_N$ if the scale factor m^2 is not considered. By using a method similar to (18), we are able to recover the desired watermarked image from the encrypted image $\tilde{I}_w(x,y)$.

3.2 Watermark Extraction

For our watermark scheme, the watermark extraction can be accomplished in either the plain domain or the encrypted domain. That is, we can extract the watermark either from the image $I_{w,256}\left(x,y\right)$ or from the encrypted image $\tilde{I}_w\left(x,y\right)$.

After the watermark has been extracted, it will be compared to the original watermark with some metrics. We use the bit error rate (BER) to measure the difference between the extracted watermark and the original one. If we denote the extracted watermark by w_j', then the BER of w_j' and w_j is given as

$$\mathrm{BER}\left(\mathbf{w}',\mathbf{w}\right) = \frac{1}{n}\sum_{j=0}^{n-1} w_j' \,\mathrm{XOR}\, w_j \tag{31}$$

where XOR is the *exclusive or* operator. If the BER is less than or equal to some threshold τ, it indicates the presence of watermark, otherwise it indicates the absence of watermark.

We shall show that our watermark scheme possesses the characteristics of blind extraction in two domains, i.e., the decrypted domain and the encrypted domain. More specifically, in the plain domain, the watermark can be extracted from the watermarked image I_w or $I_{w,256}$ without requiring the original image I. While in the encrypted domain, the watermark can be extracted from the encrypted data $\tilde{I}_w$ without requiring either $[\![I]\!]$ or I. We describe the extracting algorithm of our watermark scheme below.

Extraction in the Encrypted Domain. In order to extract the watermark from the encrypted image, we segment $\tilde{I}_w(x, y)$ into non-overlapping blocks of $m \times m$. According to the sequence $\mathbf{a}$, we select n blocks from total $(M/m)^2$ blocks. We then apply WHT-ed of size $m \times m$ to all the selected blocks to output the encrypted coefficients. Let us denote those encrypted coefficient blocks by $\{\tilde{V}_1^{\varepsilon}, \tilde{V}_2^{\varepsilon}, \ldots, \tilde{V}_n^{\varepsilon}\}$. According to the values of e_j and d_j, we choose the cardinal point $\tilde{V}_j^{\varepsilon}(k_0, l_0)$ and the adjacent point $\tilde{V}_j^{\varepsilon}(k_1, l_1)$ in the block $\tilde{V}_j^{\varepsilon}(k, l)$. If we use $\tilde{w}'_j$ to denote the extracted information from j-th selected block, then the watermark extraction in the encrypted domain can be given as

$$\tilde{w}'_j = \tilde{V}_j^{\varepsilon}(k_1, l_1) \left[\tilde{V}_j^{\varepsilon}(k_0, l_0)\right]^{-1}. \tag{32}$$

By using the homomorphic properties of the cryptosystem and (30), we have

$$\mathcal{D}\left[\tilde{w}'_j\right] = \begin{cases} 0, & \text{if } w_j = 0 \\ m^3 \alpha_j. & \text{if } w_j = 1 \end{cases} \tag{33}$$

The scaling factor m^3 can be easily removed after decryption, or directly removed from $m^3\alpha_j$ in the encrypted domain by using the multiplicative inverse method [12]. If the scaling factor is not considered, there is no difference between $\mathcal{D}[\tilde{w}'_j]$ and w_j. Assuming that $\frac{\mathcal{D}[\tilde{w}'_j]}{m^3\alpha_j}$ is denoted by ϖ_j, then we have

$$\varpi_j = w_j. \tag{34}$$

Therefore we have proved the extracted encrypted watermark $\tilde{w}'_j$ is the encrypted version of the original watermark w_j. We also show an interesting property of the watermark extraction in the encrypted domain by using equation (34). It means that after performing a simple scaling, the extracted watermark is identical to the original watermark without any distortion.

Extraction in the Decrypted Domain. Let us consider the case of extracting the watermark from the decrypted watermarked image. Based on the analysis in Section 3.1, the implementation of watermarking in the encrypted domain will enlarge the plain value of the watermarked image. And small modification of the transform coefficients may result in large variation in the spatial domain. Thus the decrypted values are very likely to be greater than 255 or less than 0. Moreover, all the elements of I_w may not be integers. In order to keep the format compliance, the decrypted values should be mapped to the integers between 0 and 255. Suppose we have already recovered the correct value $m^2 I_w$ from the decryption of $\tilde{I}_w$. Generally, the process of mapping can be given as

$$I_{w,256} = \left\lfloor 255 \cdot \frac{m^2 I_w - \min\left\{m^2 I_w\right\}}{\max\left\{m^2 I_w\right\} - \min\left\{m^2 I_w\right\}} \right\rfloor \tag{35}$$

where $\lfloor \cdot \rfloor$ is the flooring function, while $\min\{\cdot\}$ and $\max\{\cdot\}$ are the minimum and maximum operators, respectively.

Both the quantized watermarked image $I_{w,256}$ and the watermarking key $(\mathbf{a}, \mathbf{e}, \mathbf{d})$ are sent to the extraction device for further processing. Specifically, we segment $I_{w,256}(x, y)$ into non-overlapping blocks of $m \times m$ first, then select n blocks among all the blocks according to the random integer sequence $\mathbf{a}$. WHT of size $m \times m$ is applied to all the selected blocks to output the encrypted coefficients. Let us denote the encrypted coefficients by $\{V_1^{\varepsilon}, V_2^{\varepsilon}, \ldots, V_n^{\varepsilon}\}$. According to the values of e_j and d_j, we choose the cardinal point $V_j^{\varepsilon}(k_0, l_0)$ and the adjacent point $V_j^{\varepsilon}(k_1, l_1)$ in the block V_j^{ε}. If we use w'_j to denote the extracted bit in the V_j^{ε}, then the process of watermark extraction can be given as

$$w'_j = V_j^{\varepsilon}(k_1, l_1) - V_j^{\varepsilon}(k_0, l_0). \tag{36}$$

$\{w'_j\}$ will be compared with the embedding message $\{w_j\}$ by using the BER metric to output the result that whether there is a watermark in $I_{w,256}(x, y)$ or not.

4 Experimental Results

We test the proposed algorithm on a few images. Due to the limitation of paper length, we only show the results on 'Lena' image of $512 \times 512 \times 8$ bits. The original watermark message is chosen as a binary image of $64 \times 64 \times 1$ bits. The original image and the watermark are shown in Fig. 2(d)-2(g). We exploit the 2D WHT in the experiments and choose two large prime numbers p and q for the cryptosystem. The product of p and q is longer than 1024 bits, so the encryption is secure in practice. We show the encrypted image in Fig. 2(c), which is sufficiently scrambled and secure enough to protect the image.

Firstly, we perform WHT-ed of size 512×512 to the whole image. The decryption of the result looks the same as the WHT of the plain image. We then perform IWHT-ed to reconstruct the image in the encrypted domain. After decrypting, we obtain an image which looks the same as the original one. The experimental result is shown in Fig. 2(h)-2(i)

Secondly, the encrypted image is segmented into non-overlapping blocks of 8×8. We perform WHT-ed of size 8×8 on each block. Since there are totally $4096(=64 \times 64)$ bits in $\mathbf{w}$, we choose all the blocks for the watermark insertion. We adopt $e_j = 64$, $d_j = 1$ and $\alpha_j = 8$ for $j = 1, 2, \ldots, 4096$. According to the value of e_j and d_j, the cardinal point and its adjacent point are selected in $\tilde{V}_j$. By means of (26), we modify coefficients in all the selected blocks for watermark embedding. We then perform IWHT-ed to output the encrypted watermarked image. We show the encryption data and its decryption in Fig. 2(h)-2(i).

Thirdly, by using (32) we extract the encrypted watermark, which is embedded in the encrypted image. The extracted encrypted data and its decryption are shown in Fig. 2(j)-2(k). It can be seen that the decryption looks the same as the original watermark. Actually it is identical to the original watermark in Fig. 2(b) after removing the scaling factor.

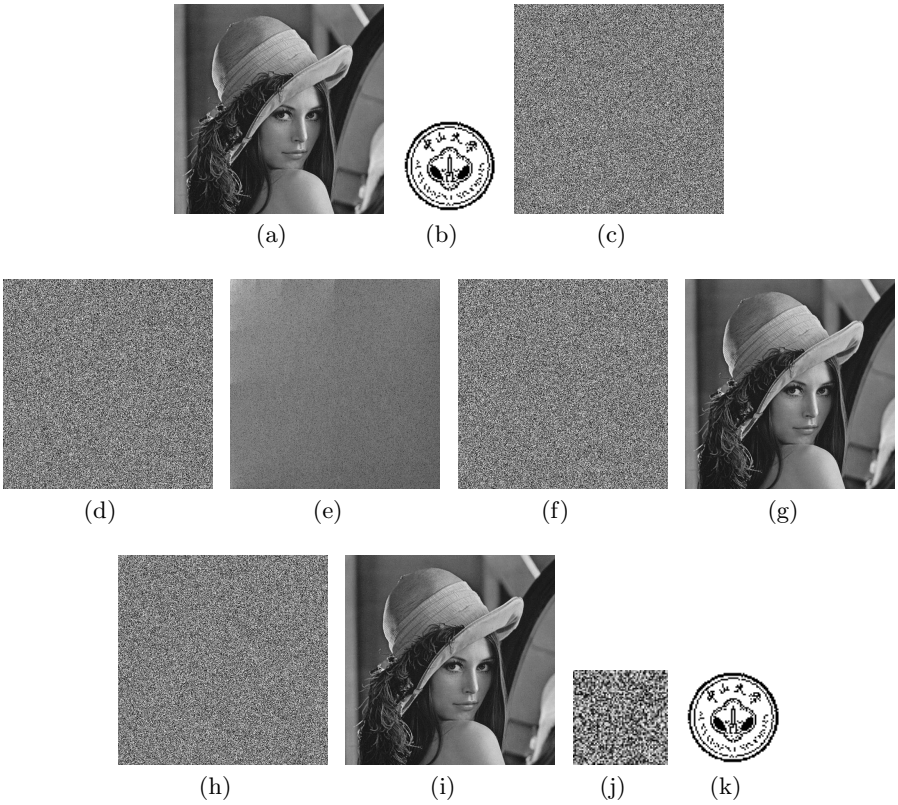

Fig. 2. Experimental Results: (a) The Original "Lena" image; (b) The watermark; (c) The encrypted "Lena" image; (d) The WHT-ed coefficients; (e) Decryption of WHT-ed coefficients; (f) The encrypted reconstruction; (g) Decryption of reconstruction; (h) The encrypted watermarked image; (i) Decryption of the encrypted watermarked image; (j) The extracted watermark from the encrypted data; (k) Decryption of the extracted watermark

In order to evaluate the visual effect of the watermarked image, we compute the peak signal-to-noise ratio (PSNR) between the original and the watermarked images. The PSNR of the watermarked image in our experiment is 43.31 dB. We also apply our watermark algorithm to 100 grayscale images, each of which is of $512 \times 512 \times 8$ bits. The watermark we use is the one shown in Fig. 2(b). The average PSNRs of the rounding watermarked images $I_{w,256}$ and the no-rounding watermarked images I_w are 43.18 dB and 43.92 dB, respectively. However, all the BERs (error in detection) are 0 under these two situations. This means our algorithm can keep the watermarked image in a good visual quality.

The attackers may perform the attacks on the decrypted image or the encrypted image. Since the attack on the encrypted image may result in a random decrypted image, the attacker is more likely to attack the decrypted image.

Thus we consider the watermark detection performance against Gaussian noise. The Gaussian noise is added in the decrypted watermarked image $I_{w,256}$. For WNR (watermark to noise ratio) > -2 dB, the BER < 0.032 by using (36) in watermark retrieval. In practical applications, our watermark algorithm can be extended to the case of spread spectrum scheme, which will greatly improve the robust performance of our watermark.

5 Conclusions

This paper has investigated the implementation of WHT and its applications in image watermarking in a homomorphic encrypted domain. The main contributions are listed as follows:

1) We have described a method to perform WHT and the fast WHT in the encrypted domain, which is based on the homomorphic properties. By using our method, WHT can be implemented in the encrypted domain without any quantization error. We also deduce some elegant equations to show the relationship between WHT(IWHT) in the encrypted domain and WHT(IWHT) in the plain domain.
2) We have proposed an image watermarking scheme based on block WHT-ed. The watermark embedding is carried out in the encrypted domain. However, we can extract the watermark both in the plain domain and the encrypted domain. Both the extractions are blind processing, without involving either the plain original image or the encrypted one.

Our algorithm gives a possible solution to the security problem in the watermarking community. It is possible to use our watermarking scheme to design a secure media distribution system. However, due to the constraints of the homomorphic cryptosystems, the encryption of the original image results in a high store and computation overhead. It is our future work to address the issues regarding the limitation, and to extend our watermarking algorithms to other transforms, e.g., DWT in the encrypted domain.

Acknowledgements. This work was supported by 973 Program (2011CB302200) and NSFC (U1135001).

References

1. Cayre, F., Fontaine, C., Furon, T.: Watermarking security: Theory and practice. IEEE Transactions on Signal Processing 53(10), 3976–3987 (2005)
2. Kalker, T.: Considerations on watermarking security. In: 4th IEEE Workshop on Multimedia Signal Processing–MMSP 2001, pp. 201–206. IEEE (2001)
3. Adelsbach, A., Huber, U., Sadeghi, A.-R.: Fingercasting—Joint Fingerprinting and Decryption of Broadcast Messages. In: Batten, L.M., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 136–147. Springer, Heidelberg (2006)

4. Celik, M., Lemma, A., Katzenbeisser, S., van der Veen, M.: Lookup-table-based secure client-side embedding for spread-spectrum watermarks. IEEE Transactions on Information Forensics and Security 3(3), 475–487 (2008)
5. Venkata, S., Emmanuel, S.K.M.: Robust watermarking of compressed and encrypted jpeg 2000 images. IEEE Transactions on Multimedia 99, 1 (2011)
6. Memon, N., Wong, P.: A buyer-seller watermarking protocol. IEEE Transactions on Image Processing 10(4), 643–649 (2001)
7. 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)
8. Troncoso-Pastoriza, J., Katzenbeisser, S., Celik, M., Lemma, A.: A secure multidimensional point inclusion protocol. In: 9th ACM Workshop on Multimedia and security–MM&Sec 2007, pp. 109–120. ACM (2007)
9. Bianchi, T., Piva, A., Barni, M.: Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Transactions on Information Forensics and Security 5(1), 180–187 (2010)
10. Erkin, Z., Franz, M., Guajardo, J., Katzenbeisser, S., Lagendijk, I., Toft, T.: Privacy-Preserving Face Recognition. In: Goldberg, I., Atallah, M.J. (eds.) PETS 2009. LNCS, vol. 5672, pp. 235–253. Springer, Heidelberg (2009)
11. Sadeghi, A.-R., Schneider, T., Wehrenberg, I.: Efficient Privacy-Preserving Face Recognition. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 229–244. Springer, Heidelberg (2010)
12. Zheng, P., Huang, J.: Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. In: 19th ACM International Conference on Multimedia–MM 2011, pp. 413–422. ACM (2011)
13. Barni, M., Bianchi, T., Catalano, D., Di Raimondo, M., Donida Labati, R., Failla, P., Fiore, D., Lazzeretti, R., Piuri, V., Scotti, F., et al.: Privacy-preserving fingercode authentication. In: 12th ACM Workshop on Multimedia and Security–MM&Sec 2010, pp. 231–240. ACM (2010)
14. Barni, M., Failla, P., Lazzeretti, R., Sadeghi, A., Schneider, T.: Privacy-preserving ecg classification with branching programs and neural networks. IEEE Transactions on Information Forensics and Security, 452–468 (2011)
15. Bianchi, T., Piva, A., Barni, M.: Encrypted domain dct based on homomorphic cryptosystems. EURASIP Journal on Information Security 2009 1 (2009)
16. Rivest, R., Adleman, L., Dertouzos, M.: On data banks and privacy homomorphisms. Foundations of Secure Computation, 169–178 (1978)
17. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–254. Springer, Heidelberg (1999)
18. Fino, B., Algazi, V.: Unified matrix treatment of the fast walsh-hadamard transform. IEEE Transactions on Computers 100(11), 1142–1146 (1976)