IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2022, accepted February 6, 2022, date of publication February 11, 2022, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151115

Privacy-Preserving Image Watermark Embedding Method Based on Edge Computing

HANG CHENG[1,2], QINJIAN HUANG[3], FEI CHEN[3], MEIQING WANG[1], AND WANXI YAN[1]

[1]School of Mathematics and Statistics, Fuzhou University, Fuzhou, Fujian 350108, China
[2]Key Laboratory of Information Security of Network Systems, Fuzhou University, Fuzhou, Fujian 350108, China
[3]College of Computer Science and Big Data, Fuzhou University, Fuzhou, Fujian 350108, China

Corresponding author: Fei Chen (chenfei314@fzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62172098, Grant 62072109, and Grant 61702105; in part by the Natural Science Foundation of Fujian Province under Grant 2020J01497; and in part by the Education Research Project for Young and Middle-Aged Teachers of the Education Department of Fujian Province under Grant JAT200064 and Grant JAT190020.

ABSTRACT To solve the problem of privacy leakage and response latency in outsourced image watermark embedding in cloud computing, an efficient and privacy-preserving watermark embedding method for outsourced digital images was proposed by introducing edge computing technology. We had proposed a perturbing encryption method with homomorphism to ensure the information security and the correctness of discrete wavelet transformation in the encrypted domain. In addition, the framework was designed to guarantee the safety of singular value decomposition that edge server could not recover the original image matrix. The experimental results show that the proposed method is superior to similar secure watermarking schemes in terms of encryption/decryption time and ciphertext expansion. The proposed method enables the watermarking operation to be performed in an unsafe outsourced environment while achieving a watermarking effect similar to the plaintext equivalent.

INDEX TERMS Cloud computing, digital images, discrete wavelet transformation, edge computing, encryption, information security, privacy, safety, technology, watermarking.

I. INTRODUCTION

The development of artificial intelligence and big data has prompted an explosion of information. Every minute, 1.1 million tweets are sent, 684,478 contents are shared on Facebook, 3.2 million queries are searched on Google, and 48h of videos are uploaded to YouTube [1]. The creation and the dissemination of massive multimedia information leads to disputes over digital copyrights. It poses challenges to users with limited storage/computing resources in watermarking processing. Combining digital watermarking technology [2] and cloud computing technology is effective to alleviate these problems.

In recent years, the development of cloud servers with massive storage and powerful computation has made it possible to outsource large-scale data storage and processing. The task of watermark embedding massive images can now be transferred to the cloud, ensuring the user's copyright ownership and decreasing the amount of local computing/storage resources required [3]. However, outsourcing data often involves trade secrets and user sensitive data [4]. Therefore, it is necessary to construct a privacy-preserving outsourcing watermarking scheme that can protect the privacy and security of data while implementing watermark embedding in the cloud.

There are a lot of digital watermarking related work in recent years [5]–[8], [9]. In the plaintext domain, digital watermarking is a technology that embeds additional information into the host carrier to prove ownership. This process mainly includes spatial domain watermarking [10], [11] and transformation domain watermarking [12]. To date, many approaches have been developed for secure image watermarking, for example, the method of reversible data hiding (RDH) in the encrypted domain [13]. For example, Zhang [14] segmented image pixels into groups by blocks in the spatial domain, then encrypted them by the bit flip, and finally embedded secret bits into the least significant bits of the host image. To reduce the error rate of information extraction, Hong *et al.* [15] improved the algorithm in [14] by employing smoothness between each block and

The associate editor coordinating the review of this manuscript and approving it for publication was Gerard-Andre Capolino.

implementing a correlation of pixels at the block boundary. However, this improvement had little effect on performance. To facilitate information embedding, a novel embedding framework was proposed by reserving the space before encryption [16], [17]. However, this was not suitable for practical applications because content owners need to perform extra work, except for image encryption. For improved security and embedding rate, Zhang *et al.* [18] utilized a public key mechanism to encrypt the carrier image and used the homomorphism of encryption technology to embed secret information. Similar methods were found in the literature [19]–[22]. These methods were based on a public key mechanism, but suffered from data inflation and high computational complexity. Considering the potential value of different image file formats, some secure RDH schemes have been proposed for JPEG images [23] and 2-dim vector graphics [24], respectively. It is widely accepted that the ciphertext RDH technology can achieve secure watermark embedding. However, robustness has consistently been a major problem for schemes based on secure RDH. In [25], Peng *et al.* proposed a separable watermarking scheme to improve the robustness of the method. Based on chaotic encryption, Gao and Gao [26] proposed a novel verifiable image encryption, which not only protects the image information but also allows watermark data to be hidden and extracted. However, redundancy loss leads to difficulty in embedding watermarks, as well as the reduction of the capacity to embed watermarks. Yao *et al.* [27] proposed a scheme for embedding a visible watermark in the bit plane of the encryption domain. However, the visual impact of this watermark on the decrypted image was apparent and caused the use-value of the image to be highly susceptible to destruction. Literature [28] studied the watermark embedding of medical images in the encryption domain based on JPEG-LS, concluding that the watermark could be extracted in both the encryption and plaintext domains.

By contrast, much less research has focused on secure watermarking in the transform domain. However, this method has a better robustness and higher embedding capacity [29]. This is especially true for mixing different transformations, such as the combination of discrete wavelet transformation (DWT) and singular value decomposition (SVD). Presently, the watermarking method that combines DWT and SVD can achieve an acceptable balance between robustness and invisibility under appropriate scaling factors, and it has become a common method of watermarking in the plaintext domain [30]. In summary, securely implementing DWT and SVD is imperative in an untrusted environment. The main challenge of the transform domain watermarking scheme in ciphertext is how to ensure a secure frequency transformation operation while obtaining the same effect as plaintext. Zheng and Huang [31] combined the Paillier cryptosystem [32] to propose a DWT transformation in the encrypted domain. However, their data reduction method presented some consistent pixel ciphertext that resulted in a risk of information leakage. Additionally, the discrete Fourier transform [33] and discrete cosine transform [34] of encrypted signals were studied. Inspired by Zheng and Huang's research [31], Xiang *et al.* [35] proposed a reversible watermark embedding scheme based on the Paillier cryptosystem and multi-level DWT decomposition in the encrypted domain. However, it was challenging to perform multiplication for the Paillier cryptosystem.

Compared to the ciphertext wavelet transformation, there were few studies on singular value decomposition in the encrypted domain. To design a secure outsourcing watermarking framework based on DWT-SVD that was more robust [29] , we advanced the encryption method from the literature [36] to encrypt image data and securely perform DWT. Additionally, a secure singular value decomposition framework was designed to compute outsourced SVD. The cloud server can process a large amount of data, but the transmission time is proportional to the amount of data. This implies that data processing outsourced to the cloud may cause a response delay [37]. To solve this problem, we introduced edge computing technology to propose a lightweight privacy-preserving digital watermarking method for outsourced host images.

The proposed scheme requires that the content owner (CO) encrypts the host and watermark images and uploads the encrypted data to the edge computing server. After obtaining the ciphertext, the server performs the watermark embedding method by combining the Haar DWT (HDWT) and SVD. The server then returns the encrypted host image containing the watermark to the authorized user who can then decrypt and obtain the corresponding plaintext of the image containing the watermark. This paper is summarized as follows:

1) The first contribution: A privacy-preserving color image watermarking framework was designed using edge computing technology. With this framework, the CO only needs to encrypt and upload the ciphertext. The edge computing server embeds the watermark into the host image without any interaction with the CO. In addition, a secure quadratic SVD framework was proposed to improve the watermark embedding effect.

2) The second contribution: An encryption algorithm that is able to perform the HDWT in an encrypted domain was designed. Compared with existing encryption algorithms commonly used for DWT in the encryption domain (such as the Paillier homomorphic encryption), the encryption and decryption times and storage costs were significantly improved. The SHA256 hash algorithm and logistic mapping (LM) fusion method were proposed to ensure the uniqueness of the watermark image key and protect the privacy information of the watermark image.

3) Experimental results showed that the proposed method was superior to other encrypted watermarking schemes in terms of encryption and decryption times and the extension rate of ciphertext. When encrypting the same 8-bit data using the proposed method, the ciphertext extension was approximately 50% of the Paillier cryptosystem. Compared with other methods that use the RSA algorithm, the

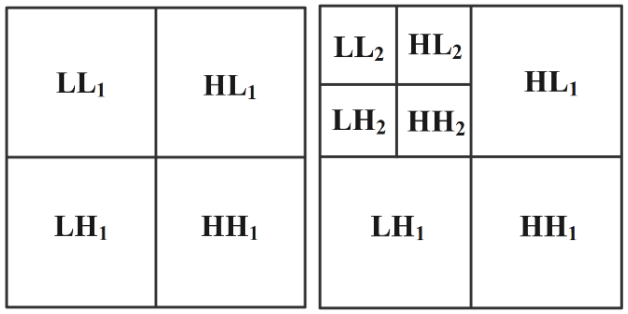


FIGURE 1. Flow of discrete wavelet transformation. ((a) is one-level DWT, (b) is two-level DWT).

encryption/decryption speed improved by 3.46 s/259.57 s under the same image condition. Compared with the same watermarking scheme in plaintext, the PSNR of the proposed scheme reduced slightly by approximately 0.05 dB on average, when the watermark embedding factor was 0.25. Additionally, less data were lost.

II. PRELIMINARY

A. LOGISTIC MAPPING

Logistic mapping (LM) is a type of chaotic mapping [38] widely used in digital communication security, multimedia data security, and other domains. It is defined as follows:

$$X(k+1) = u * X(k) * (1 - X(k)). \tag{1}$$

Specially, $k = 0, 1, \ldots, n, X(k) \in (-1, 1), u \in (0, 4)$. $X(k)$ is the mapping variable, and u is the system parameter. When $0 < X(0) < 1$ and $3.5699456 < u < 4$, the mapping function is in a chaotic state, which is an unordered and unpredictable state.

$\boldsymbol{M} = \{m(i)\}_{i=1}^{n}$ represents an image matrix with n pixels. When encrypting $\boldsymbol{M}$, the LM function requires n iterations to obtain a sequence of one-dimensional vectors of length n. The sequence is then normalized (such as Liu *et al.* [39]), and the result is mapped to the interval $[0, 255]$ to obtain an encryption vector $\boldsymbol{L} = \{l(i)\}_{i=1}^{n}$. $\boldsymbol{E} = \{e(i)\}_{i=1}^{n}$ is the ciphertext of $\boldsymbol{M}$, and is generated using pseudo random xor encryption as follows:

$$e(i) = m(i) \oplus l(i). \tag{2}$$

In decryption processing, $\boldsymbol{L}$ is restored through the private key $\{X(0), u\}$, and the plaintext of $\boldsymbol{M}$ is obtained by the xor operation between $\boldsymbol{L}$ and $\boldsymbol{E}$.

B. DISCRETE WAVELET TRANSFORM

DWT is a process of multiscale and frequency-domain decomposition of images [40]. In the watermarking scheme based on DWT, DWT decomposes the target image matrix into four sub-band matrices, which are often used for embedding watermarks. As shown in Fig.1, the low-frequency (LL) component represents the approximate image information. The other three components are horizontal high-frequency (HL), vertical high-frequency (LH), and high-frequency (HH).

Haar discrete wavelet transform (HDWT) is a type of DWT [41] and is the key technology used for the frequency domain of the host image in this study. As the calculation procedure of HDWT is simple, the image matrix encrypted by the proposed encryption method does not cause an excessive extension of the ciphertext during the calculation of HDWT. We briefly introduce the calculation process for HDWT and inverse-HDWT(IHDWT). Assuming that an image of size $m \times n$ is divided into non-overlapping 2×2 blocks $\{B_i\}_{i=1}^{(m \times n)/4}$, the elements of any block B_i will be denoted from left to right and from top to bottom as a_1, a_2, a_3 and a_4, and the coefficients of B_i after HDWT will be calculated as $(a_1 + a_2 + a_3 + a_4)/2$, $(a_1 - a_2 + a_3 - a_4)/2$, $(a_1 + a_2 - a_3 - a_4)/2$, and $(a_1 - a_2 - a_3 + a_4)/2$, respectively. Based on these transformation coefficients, the IHDWT can be realized by performing the same calculation process. If the data are not changed, the inverse transformation can restore the original image data.

C. SINGULAR VALUE DECOMPOSITION

SVD [42] is a mathematical tool commonly used in signal and image processing. The calculation formula can be expressed as follows:

$$\boldsymbol{I} = \boldsymbol{U} \times \boldsymbol{\Sigma} \times \boldsymbol{V}^{\mathrm{T}}. \tag{3}$$

Here, $\boldsymbol{I}$ is a matrix of size $m \times n$. The matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ have magnitudes of $m \times m$ and $n \times n$, respectively. $\boldsymbol{\Sigma}$ is a diagonal matrix of size $m \times n$, and T is the transpose of the matrix. SVD has two commonly used properties.

$$\begin{cases} \boldsymbol{I} \times \boldsymbol{I}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}} \\ \boldsymbol{I}^{\mathrm{T}} \times \boldsymbol{I} = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \boldsymbol{V}\boldsymbol{\Sigma}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}. \end{cases} \tag{4}$$

According to (4), it is straightforward to get $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ or $\boldsymbol{V}$, $\boldsymbol{\Sigma}$ by means of eigenvalue decomposition of $\boldsymbol{I} \times \boldsymbol{I}^{\mathrm{T}}$ or $\boldsymbol{I}^{\mathrm{T}} \times \boldsymbol{I}$.

III. PRIVACY-PRESERVING WATERMARKING SCHEME

The privacy-preserving watermark outsourcing scheme proposed in this study is outlined in Fig.2, including content owners(COs), edge computing servers(S_1, S_2, S_3), a trusted third party(TTP), and an authorized-user(AU). TTP distributes key parameters, CO encrypts and uploads data to the edge server, the edge computing server performs secure HDWT and SVD, and AU decrypts the ciphertext to obtain the plaintext of the image with the watermark.

A. SYSTEM INITIALIZATION

The TTP generates the integer γ_1 and produces two large primes D and F, where $F \gg D$. Specifically, $|D| = \gamma_2$, $|F| = \gamma_3$, γ_2 and γ_3 are the system parameters, and $|\cdot|$ indicates the number of binary bits of data. TTP then sends γ_1, D, and F to the CO, AU, S_1, and S_3 through the secure channel.

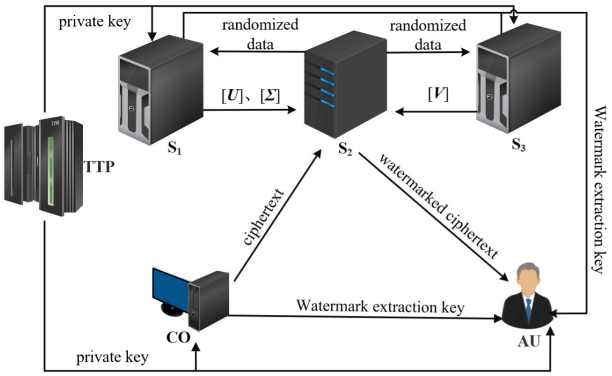


FIGURE 2. Framework of watermarking scheme based on edge computing.

The image encryption in our framework consists of host image encryption and watermark image encryption.

1) HOST IMAGE ENCRYPTION

The CO encrypts the host images as follows [36]:

$$[p] = p + c \cdot D + y \cdot F. \tag{5}$$

where p represents plaintext data, $[\cdot]$ indicates ciphertext encrypted by (5), c and y are random positive integers, and $c, y \in (1, t)$, $t = 2^{\gamma_1}$. For enhanced security, c and y can be selected as different positive integers for each pixel. The plaintext can be obtained by successively modulating D and F with ciphertext, as follows:

$$p = [p] \bmod F \bmod D. \tag{6}$$

To ensure that (5) can derive decimal and negative values, the following improvement treatments are proposed:

when p is a decimal number, p expands into an integer with a certain proportion factor Q. Subsequently, an encryption operation is performed. When decrypting, Q is divided to obtain the original plaintext.

When the plaintext is negative, (5) can be rewritten as:

$$[p] = (p + D) + (c - 1) \cdot D + y \cdot F. \tag{7}$$

Using (6) to decrypt (7), results in $(p + D) < D$ as p is negative. Then, we can use $(p+D)-D$ to restore the plaintext of p because $(p + D) \gg p$.

The encryption method in this study includes addition and multiplication homomorphisms. For example,

$$\begin{cases} [p_1] + [p_2] = [p_1 + p_2] \\ [p_1] \cdot [p_2] = [p_1 \cdot p_2]. \end{cases} \tag{8}$$

To ensure accuracy of the subsequent wavelet decomposition process and the inverse decomposition process, the ciphertext of the host image I_h is enlarged with $[\boldsymbol{I}] = \left[\boldsymbol{I}_h Q^2\right] \cdot \beta$, and β becomes a multiple of two to counteract the two in the denominator in HDWT.

2) WATERMARK IMAGE ENCRYPTION

When the precondition for a chaotic state of the LM is satisfied, $0 < X(0) < 1$, $3.5699456 < u < 4$, and CO randomly selects the initial value for the LM, namely, the private key $\{X(0), u\}$. To increase the sensitivity and uniqueness of the private key, this study uses SHA256 to generate the hash value of the watermark image $\boldsymbol{W}$ and then divides the hash value into several non-overlapping bit vectors with a length of 8 bits. This method then generates an eight-dimensional vector using a vector-by-vector xor operation and finally normalizes it to the interval (0, 1), denoted by $X_W(0)$. By making $X'(0) = (X_W(0) + X(0))/2$, the final private key becomes $\{X'(0), u\}$, which generates the final encryption vector $[\boldsymbol{L}]$ that encrypts the watermark image to obtain the encrypted watermark image $[\boldsymbol{W}]_l$. $[\cdot]_l$ represents ciphertext encrypted by (2).

B. PRIVACY-PRESERVING HAAR DISCRETE WAVELET TRANSFORMATION

After receiving the encrypted data $[I]$, the server S_2 performs HDWT in the encryption domain.

As shown in II-B, HDWT involves subtraction. To ensure that the ciphertext is a positive integer for correct subsequent decryption, the generation mode of random numbers c and y in (5) is stipulated during encryption. Initially, the CO divides the host image into several blocks of size 2×2, that is, $\{B_i\}_{i=1}^{(m\times n)/4}$. The random number c is then marked in the four position pixels of each B_i, denoted by c_1, c_2, c_3 and c_4, which correspond to a_1, a_2, a_3 and a_4 in section II-B. Thereafter, random numbers are generated within the interval $(1, t)$, which satisfy $c_2 > c_4 > r_1 > r_2 > 0$, $c_1 = c_2 + r_1$ and $c_3 = c_4 + r_2$. Thus, the ciphertext of the frequency coefficient, after the HDWT in the encryption domain, is guaranteed to be a positive integer. The random number y is handled in a similar manner.

In this study, the watermark is embedded in the low-frequency region coefficient $\boldsymbol{LL}$ after HDWT. Therefore, random numbers c and y in the subsequent re-encrypted $\boldsymbol{LL}$ ciphertext change, causing the result of the calculation to be negative and inadequate to carry out the inverse wavelet calculation directly. Consequently, the random number c_g used for re-encryption of the inverse process in the low-frequency region can be specified as $c_g > c_2 > c_4 > r_1 > r_2$ and $c_g \in (t + 1, 2t)$. y_g is handled in the same manner. Thus, it can be observed that the four ciphertext elements of any block, B_i^{re}, after recovery are all positive during IHDWT to ensure the correctness of the decryption.

Based on the above provisions, after completing the HDWT under the ciphertext, S_2 extracts the low-frequency coefficient $\boldsymbol{LL}$ of the ciphertext, calculates $\boldsymbol{A}_1 = [\boldsymbol{LL}] \times [\boldsymbol{LL}]^{\mathrm{T}}$ and $\boldsymbol{A}_2 = [\boldsymbol{LL}]^{\mathrm{T}} \times [\boldsymbol{LL}]$, and sends $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ to the server S_1 and S_3.

The flow of privacy-preserving Haar discrete wavelet transformation is shown in Fig.3.

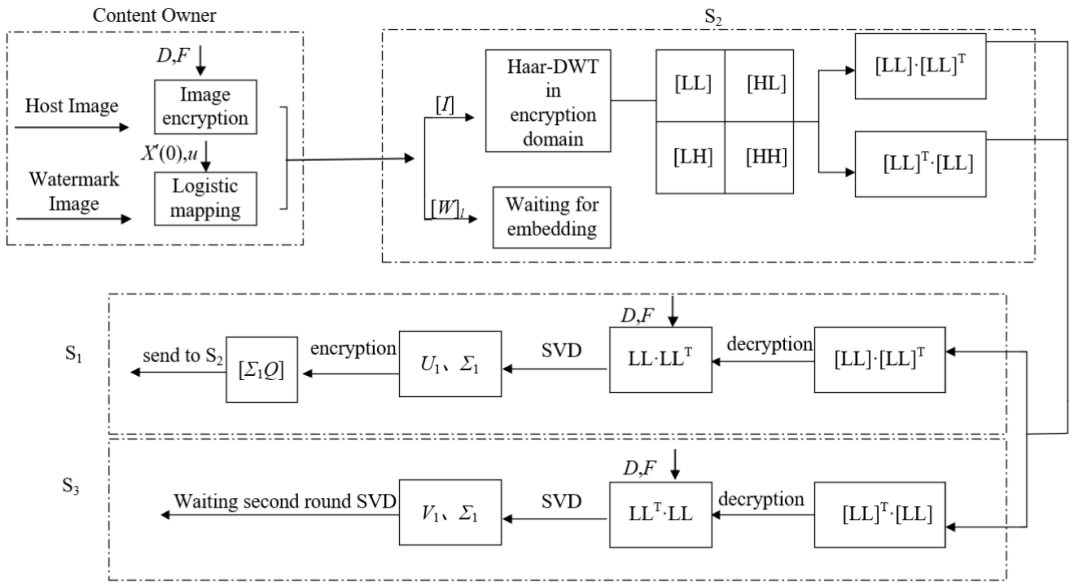


FIGURE 3. Process of privacy-preserving Haar discrete wavelet transformation and secure singular value decomposition in the proposed framework.

C. SECURE WATERMARK EMBEDDING

The final effect of watermark embedding can be improved using a second SVD on the watermarked diagonal matrix $\mathbf{\Sigma}$, as confirmed by Liu *et al.* [39]. Therefore, in this study, a quadratic SVD mechanism is adopted to implement the watermarking operation in the encryption domain. The specific operations are as follows:

Step1. When $\boldsymbol{A}_1$ is received, S_1 decrypts $\boldsymbol{A}_1$ with the key $\{D, F\}$, and obtains the plaintext of $\boldsymbol{U}_1$ and $\mathbf{\Sigma}_1$ using (4) decomposition. Finally, S_1 sends the encrypted magnified ciphertext $\mathbf{\Sigma_1}Q$ to S_2. S_3 simulates S_1. S_3 decomposes A_2. However, it does not encrypt $\mathbf{\Sigma}_1$, and only preserves $\boldsymbol{V}_1$.

Step2. S_2 performs watermarking operations on $[\mathbf{\Sigma_1}]$ in the encryption domain: $[\mathbf{\Sigma_n}] = [\mathbf{\Sigma_1}Q] + [\alpha Q] \cdot [\boldsymbol{W}]_l$. Then, $[\mathbf{\Sigma}_n] \cdot [\mathbf{\Sigma}_n]^{\mathrm{T}}$ and $[\mathbf{\Sigma}_n]^{\mathrm{T}} \cdot [\mathbf{\Sigma}_n]$ are computed and sent to S_1 and S_3, respectively. α is the scaling factor that controls the watermark intensity embedded in the host image, and $[\alpha Q]$ is obtained using (5) in CO.

Step3. S_1 decrypts $[\mathbf{\Sigma}_n][\mathbf{\Sigma}_n]^{\mathrm{T}}$, and obtains $\boldsymbol{U}_2$ and $\mathbf{\Sigma}_2$ using eigenvalue decomposition. S_1 encrypts $\boldsymbol{U}_1 \cdot \mathbf{\Sigma}_2 \cdot Q$ and sends it to S_2. S_3 maintains $\boldsymbol{V}_2$ after the eigenvalue decomposition of $[\mathbf{\Sigma}_n]^{\mathrm{T}}[\mathbf{\Sigma}_n]$, and sends the encrypted $\boldsymbol{V}_1 \cdot Q$ to S_2.

The flow of secure watermark embedding is given in Fig.4.

D. IMAGE DECRYPTION AND WATERMARK EXTRACTION

In the encryption domain, inverse singular value decomposition is calculated by using S_2 to obtain the watermarked ciphertext in the low-frequency region: $[\boldsymbol{LL}_{\mathrm{new}}] = [\boldsymbol{U}_1\mathbf{\Sigma}_2 Q] \cdot [\boldsymbol{V}_1 Q]^T \cdot \beta$. The watermarked encryption image $[\boldsymbol{I}_W]$ is generated using the IHDWT in S_2. $[\boldsymbol{I}_W]$ is returned to AU, and S_1 and S_3 send $[\mathbf{\Sigma}_1]$, $[\boldsymbol{U}_2]$ and $[\boldsymbol{V}_2]$ to the AU. The following steps show how the AU performs image decryption and watermark extraction.

Step1. $\boldsymbol{I}_W$ is obtained by decrypting $[\boldsymbol{I}_W]$ with the private key $\{D, F\}$.

Step2. HDWT is performed for $\boldsymbol{I}_W$, and singular value decomposition is performed on the low-frequency coefficient $\boldsymbol{LL}_{\mathrm{ext}} : \boldsymbol{U}_{e1}\mathbf{\Sigma}_{e2}\boldsymbol{V}_{e1}^T = \mathrm{SVD}\,(\boldsymbol{LL}_{\mathrm{ext}})$.

Step3. The ciphertext of the watermark is extracted as $\boldsymbol{W}_e = \left(\boldsymbol{U}_2\mathbf{\Sigma}_{e2}\boldsymbol{V}_2^{\mathrm{T}} - \mathbf{\Sigma}_1\right)/\alpha$.

Step4. The watermark image is restored using the initial condition of logistic mapping $\{X'(0), u\}$ to generate vector $\boldsymbol{L} = \{l(i)\}_{i=1}^{n}$ and xor operations in the elements of W_e in sequence.

Fig.5 illustrates the flow of image decryption and watermark extraction.

IV. CORRECTNESS AND SECURITY ANALYSIS

A. CORRECTNESS ANALYSIS

The decryption of the encryption system in this study uses modular operation, and the key bit length is affected by matrix multiplication. As a result, the bit length of the key $\{D, F\}$ is required to meet certain conditions to decrypt correctly; that is, P must become a matrix, and o_{ij}, p_{ij} are the elements in the (i, j)-th position of $[P] \cdot [P]^{\mathrm{T}}$ and P, respectively. o_{ij} can be calculated using a matrix multiplication algorithm.

$$\begin{aligned} o_{ij} &= \sum_{k=1}^{N} [p_{ik}] \cdot [p_{jk}] \\ &= \sum_{k=1}^{N} (p_{ik} + c_{ik}D + y_{ik}F) \cdot (p_{jk} + c_{jk}D + y_{jk}F) \end{aligned}$$

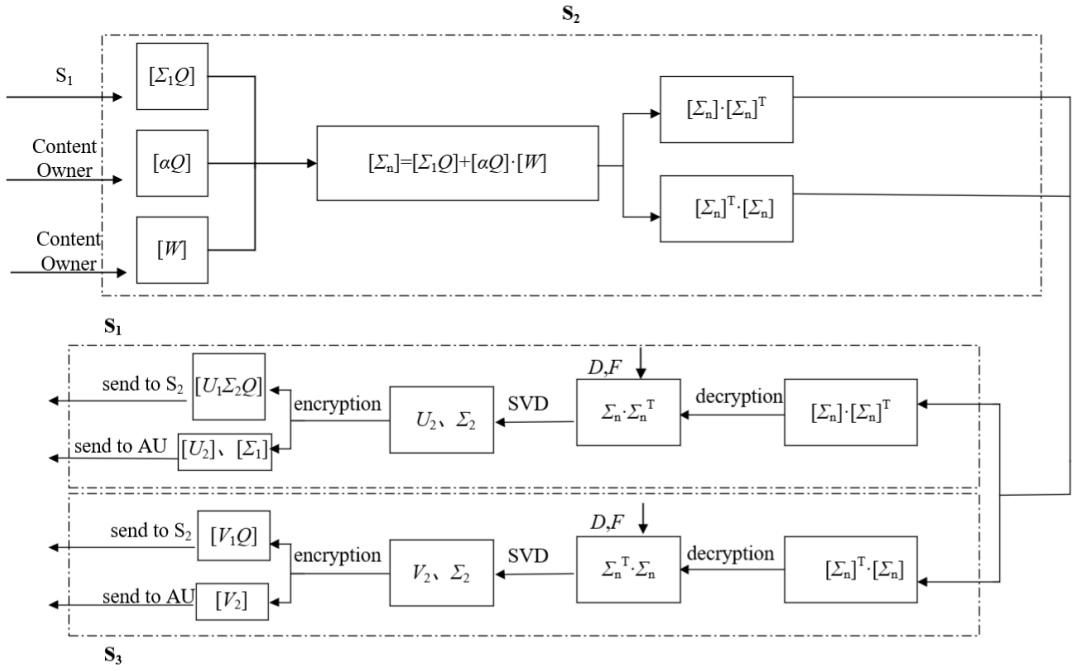


FIGURE 4. Secure watermark embedding process of the proposed privacy-preserving image watermark embedding scheme based on edge computing.

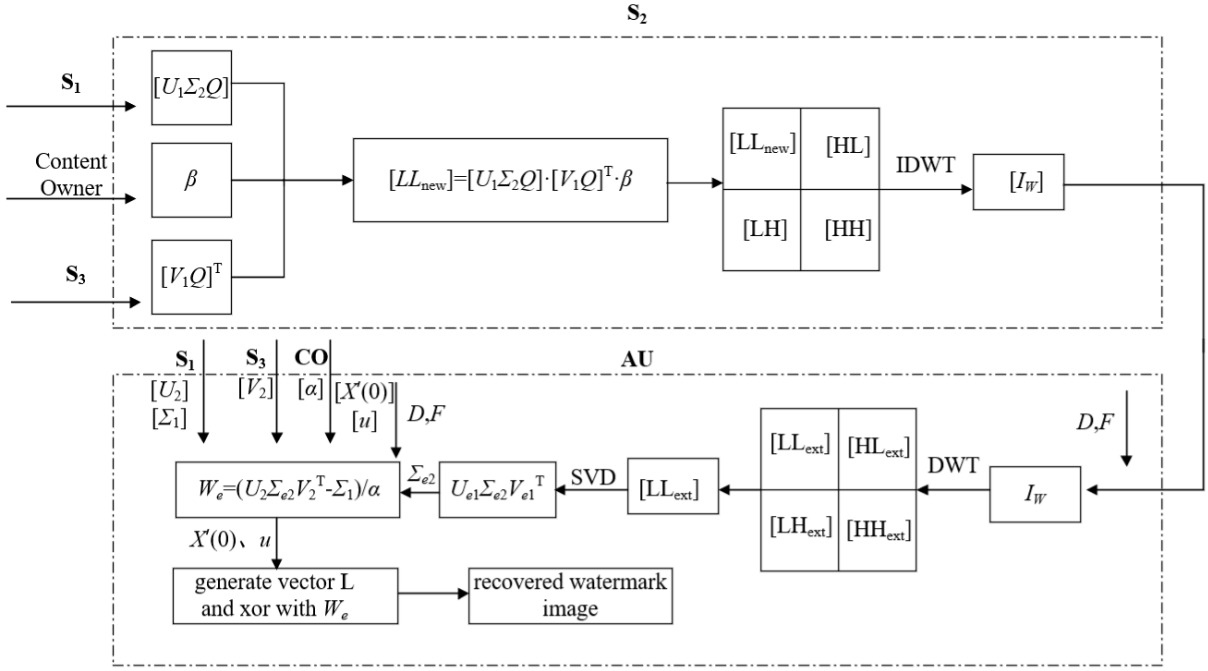


FIGURE 5. Decrypted process of the watermarked image in the edge server and the watermark extraction of decrypted watermarked image in AU (authored user).

$$= \sum_{k=1}^{N} p_{ik}p_{jk} + \sum_{k=1}^{N}((c_{ik}p_{jk} + c_{jk}p_{ik})D + c_{ik}c_{jk}D^2) + F\sum_{k=1}^{N}(y_{ik}p_{jk} + y_{jk}p_{ik} + (c_{ik}y_{jk} + c_{jk}y_{ik})D + y_{ik}y_{jk}F). \quad (9)$$

$$\sum_{k=1}^{N} p_{ik}p_{jk} + \sum_{k=1}^{N}\left(\left(c_{ik}p_{jk} + c_{jk}p_{ik}\right)D + c_{ik}c_{jk}D^2\right) < N\left(p^2 + 2tpD + t^2D^2\right) < F. \quad (10)$$

To ensure accurate execution of the modular operation, we need to ensure that the key F satisfies (10).

In (10), $\left|N\left(p^2+2tpD+t^2D^2\right)\right| < |F| = \gamma_3$, p is the product of 8-bit pixel plaintext and extension coefficient Q^2. We assume that p is the product of 255 and Q^2, and let $|\cdot|$ represent the binary bit length of the data. Thus, the modular operation with F can be performed correctly.

$$o_{ij} \bmod F = \sum_{k=1}^{N} p_{ik}p_{jk} + \sum_{k=1}^{N}\left(\left(c_{ik}p_{jk} + c_{jk}p_{ik}\right)D + c_{ik}c_{jk}D^2\right). \quad (11)$$

In addition, the key D satisfies $\sum_{k=1}^{N} p_{ik}p_{jk} < Np^2 < D$, that is, $\left|Np^2\right| < |D| = \gamma_2$. Thus, a modular operation with D can be performed correctly.

$$\left(o_{ij} \bmod F\right) \bmod D = \sum_{k=1}^{N} p_{ik}p_{jk}. \quad (12)$$

B. SECURITY ANALYSIS

This scheme assumes that edge computing servers are honest, curious, and abide by predetermined protocols. However, edge servers may use obtained intermediate data to infer information about private data. This hypothesis model has been widely used in outsourcing data security processing. This study also assumes that participants do not collude with each other [43]. Based on the information obtained by the edge server, two threat models are considered: **known ciphertext model** and **known background knowledge model** [44]. The following analyses show that the edge server cannot infer the original data from the obtained data under these two threat models.

1) KNOWN CIPHERTEXT MODEL

In this model, the edge server S_2 knows the ciphertext of the host and watermark images, as well as the ciphertext data associated with the calculation. However, for ciphertext of the form $[p] = p + c \cdot D + y \cdot F$, it is impossible to obtain any knowledge of the original plaintext data without knowing the random positive integer c, y and the private key $\{D, F\}$. Moreover, the watermark image is encrypted by Logistic Mapping, and its decryption requires the key $\left\{X'(0), u\right\}$. Otherwise, the decryption vector $\boldsymbol{L} = \{l(i)\}_{i=1}^{n}$ cannot be generated and plaintext $\boldsymbol{W}$ cannot be recovered.

2) KNOWN BACKGROUND KNOWLEDGE MODEL

Under this stronger threat model, the server obtains additional statistical information. Assuming $\boldsymbol{I} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$, the server S_1 obtains the plaintext $\boldsymbol{I}\cdot\boldsymbol{I}^{\mathrm{T}}$ by decrypting $[\boldsymbol{I}]\cdot[\boldsymbol{I}]^{\mathrm{T}}$, and obtains $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$ through singular value decomposition: Although S_1 can determine whether $\boldsymbol{I}$ has occurred through decryption, S_1 only has $\boldsymbol{U}$ and $\boldsymbol{\Sigma}$, which cannot recover the original I without the help of additional information. This is because the possible value of the unitary matrix $\boldsymbol{V}$ is close to infinity. Similarly, the server S_3 cannot recover the original matrix I. In addition, this scheme allows the random positive integer c, y chosen to encrypt each plaintext to be different to ensure the server S_2 is unable to determine the correlation between plaintext and ciphertext. That is, encrypting the same plaintext may generate different ciphertexts. For the ciphertext of watermark images, S_2 cannot infer plaintext content from the ciphertext data by statistical means. This is mainly because the scheme adopts a mechanism that binds the logistic mapping key and the watermark image. SHA256 hash technology is used to generate a logistic mapping key for each image, which makes the plaintext independent of the ciphertext of the watermark image. Therefore, the watermark image encryption mechanism in this study can resist known—ciphertext and chosen—plaintext attack.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

In this study, color images of size 512 × 512 are used as host images, such as Lena and Airplane. We used the Fuzhou University gray logo of size 256 × 256 as a watermark image. The B channel of the host image is chosen to perform the ciphertext watermark embedding operation because the B channel has little effect on the color image. The R and G channels are encrypted and stored in the edge server. The experimental parameters are set as follows: $Q = 10^3$, $\beta = 10$, $t = 2^{21}$.

In Fig.6, (a) is the host image and (d) is the watermark image. Fig.6(b)–(c) and (e)–(g) show sample images of the running results of the proposed framework at each stage. To facilitate the effect display, the display method shown in Fig.6(b) molds the ciphertext data of the encrypted Lena image with 256 to ensure that the data can be mapped to the range [0, 255] for display. Fig.6(e) is the ciphertext of the watermark image after logistic chaos mapping, where the initial key is $X(0) = 0.5$, $u = 3.9654$. Fig.6(c) is the watermarked host diagram obtained by decrypting the encrypted host image after embedding the encrypted watermark in Fig.6(e) with the embedding factor $\alpha = 0.01$. Fig.6(g) is the watermark image obtained by decrypting the ciphertext watermark. From the experimental results, the encryption method used in this study does not possess the risk of visual information leakage.

Fig.7(a)–(d) are the histograms corresponding to the B channel of the original host, watermark, encrypted watermark, and embedded watermark images, respectively. The horizontal axis represents the value of the B channel in the image, and the vertical axis represents the number of pixels. In general, both the image and the distribution of the number of pixels changes. Fig.7(a) and (d) show that the embedded watermark has an insignificant influence on the host image, indicating that the watermark embedding scheme in this study is imperceptible. As shown in Fig.7(c), the watermark image presents an approximately uniform distribution after logical scrambling encryption, and the specific content of the encrypted image cannot be analyzed using histogram statistics, meaning that it can resist a histogram statistics attack.

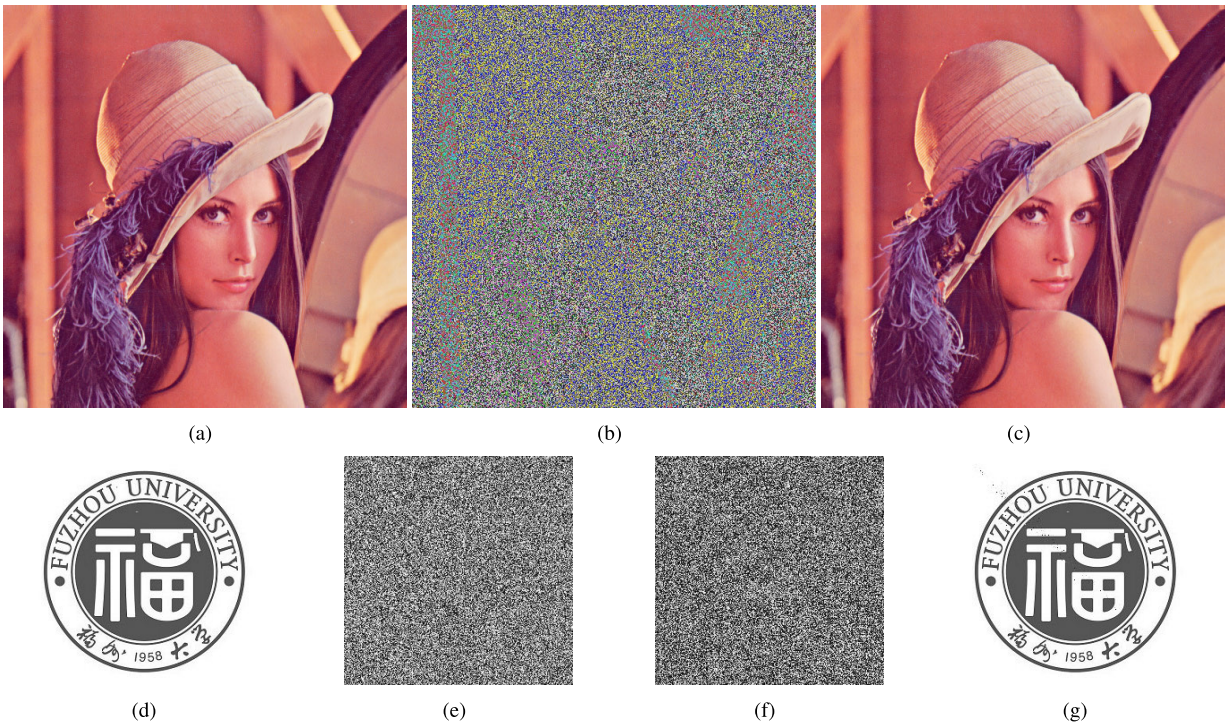


FIGURE 6. Experimental results of host image and watermark image at each stage in the proposed framework. ((a) is host image, (b) is encrypted host image, (c) is decrypted host image, (d) is watermark, (e) is encrypted watermark, (f) is extracted watermark, (g) is decrypted watermark).

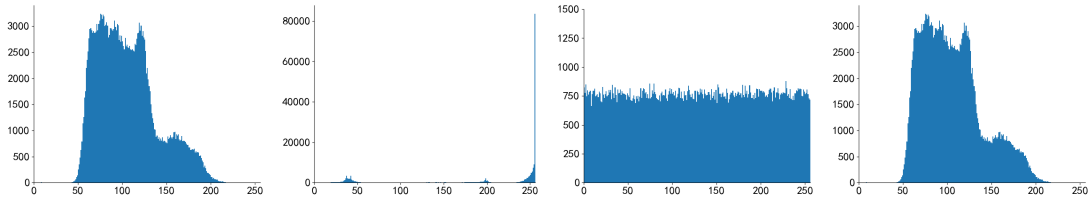


FIGURE 7. Histogram of experimental results of each stage. ((a) is the B channel histogram of the host image, (b) is the histogram of the watermark, (c) is the histogram of (b) after Logistic Mapping, (d) is the histogram of the watermarked host image).

A. CONTRAST EXPERIMENT OF ENCRYPTION PERFORMANCE

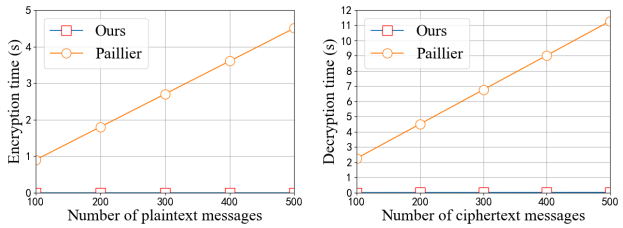
Paillier and RSA have a superior effect on watermarking in the encryption domain. To reflect the advantages of the scheme in this paper, the Paillier algorithm commonly employed in the DWT and RSA algorithms used by Liu *et al.* [39] are compared in terms of time cost of encryption, decryption, and ciphertext expansion. To ensure the security of the encryption algorithm in this study, to fulfil the requirement of key bit length in Section IV-A is fulfilled, we set $|F| = \gamma_3 = 1024$ to meet the requirement of [45] for symmetric encryption security key length. The RSA encryption algorithm requires a random selection of prime numbers p and q to compute $n = pq$. For comparison purposes, n is assumed to be 1024-bit, and the two prime numbers p and q are 512-bit. The same is performed for n, p, and q in Paillier's encryption system. The key F in the proposed scheme is set to 1024-bit.

TABLE 1. Comparison of encryption performance in different schemes.

Scheme	Encryption time	Decryption time	Cipher bit-length	Encrypted HDWT
Prosposed framework	0.0139s	0.0355s	1045-bit	✓
[39]	3.4727s	259.61s	1024-bit	✗
[35]	594.89s	1469.86s	2048-bit	✓

The experimental data are listed in Table 1. The image size is 256 × 256. It can be observed that the algorithm in this study has shorter encryption and decryption times compared to the other two schemes. Compared with that of Liu *et al.* [39], the decryption time of the algorithm in this study is decreased by approximately 259.57 s. However, Paillier presents longer encryption and decryption times, which

are proportional to the amount of encrypted/decrypted data. As shown in Fig.8, the Paillier cryptosystem used in [35] increases with an increase in the encrypted/decrypted data. In the proposed scheme, the encryption/decryption times do not increase with an increase in the amount of information data, and the growth is close to 0. For a 256 × 256 image, the encryption/decryption time in the proposed framework is approximately 0.0139 s/0.0355 s, respectively. However, Paillier encryption takes more than 4.505 s when the number of encryptions is 500 pixels, and the decryption time is close to 11.284 s.



(a) (b)

FIGURE 8. Performance comparison between the proposed encryption scheme and Paillier cryptosystem in the terms of encryption and decryption time. ((a) is encryption time, (b) is decryption time).

Given the same 8-bit data to be encrypted, it can be observed from Table 1 that the data extension caused by the proposed scheme is significantly smaller than the Paillier cryptosystem and slightly worse than the RSA algorithm. This is, for the most part, because a random positive integer, y, in the encryption of (5) is introduced. Meanwhile, RSA cannot support HDWT calculations in the encrypted domain. As a result, the proposed scheme has an excellent time consumption performance for encryption and decryption and achieves secure HDWT operation with low data extension.

B. LOSS OF WATERMARKED IMAGE IN PLAINTEXT AND ENCRYPTION DOMAIN

The peak signal-to-noise ratio (PSNR) is commonly used to measure the peak error between the original image and the image embedded with additional information. In this study, the PSNR value of the Y channel is calculated after converting the RGB image into a YUV image, which is general practice for calculating the PSNR of the RGB image.

$$PSNR = 10\log_{10}\frac{255^2}{MSE}. \tag{13}$$

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(Z_{ij} - \tilde{Z}_{ij}\right)^2. \tag{14}$$

(13) represents the PSNR formula, where MSE is the mean square error Z_{ij} is the value of channel Y derived from the original RGB host image, and $\tilde{Z}_{ij}$ is the value of channel Y derived from decrypted and watermarked RGB image. PSNR is measured in dB. The larger the PSNR value, the smaller the distortion, and the better the image quality.

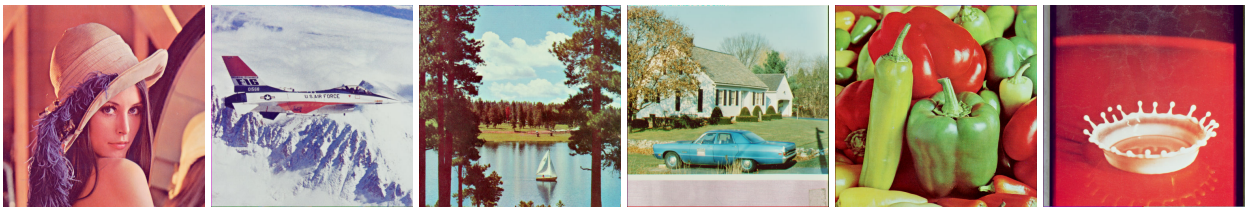
TABLE 2. PSNR of color images in different scaling factors.

Scaling factor α	0.01	0.05	0.15	0.25
Lena	52.4009	49.4972	38.3465	32.4423
Airplane	49.6764	48.0034	38.6107	31.6891
Sailboat	51.5307	49.6877	38.8704	32.0045
House	50.2554	48.3430	37.7887	31.7434
Pepper	52.3064	49.5882	38.4795	32.2094
Splash	53.0949	48.2984	36.2413	30.2435
Average	51.5441	48.9029	38.0561	31.7220

Table 2 uses a different embedding factor, α, to embed watermarks in six color images of 512 × 512 sizes, such as Lena in Fig.9. The watermark image to be embedded is shown in Fig.6(d) with a size of 256 × 256. The PSNR value between the watermarked image and the original image is shown in Table 2. When α is 0.01, 0.05, 0.15, and 0.25, the corresponding average PSNR is 52, 49, 38, and 32 dB, respectively. When the scaling factor α is 0.01, only the PSNR of the Airplane in the six images is below 50 dB. Overall, the PSNR is generally stable.

The PSNR value of each host image gradually decreases with an increase in α. The PSNR value of the color image Splash reaches the maximum value among the six images when α is 0.01. When α increases to 0.25, the PSNR becomes the lowest among all images. The main reason for this is that the Splash image is mostly smooth, and the watermark embeds the low-frequency region of the DWT. According to the principle that human vision is more sensitive to low-frequency than high-frequency information, the low-frequency information distortion of the host image is minimized, and the PSNR of the image is relatively high when α is small. However, when α is amplified, the low-frequency region of the image changes significantly, resulting in visual differences and a significant decrease in the PSNR. To demonstrate that the proposed privacy-preserving watermarking scheme can maintain a PSNR value similar to that of the same scheme in plaintext and has little influence on the decrypted watermarked image, the PSNR values of the six color images in Fig.9 are compared under different values of α.

As shown in Fig.10, an increase of the watermark scaling factor α affects the PSNR value of the host image to varying degrees. However, the PSNR value of the image embedded by the proposed privacy-preserving watermark embedding scheme is similar to the plaintext PSNR value of the same embedding method, with a maximum difference of approximately 1 dB. Therefore, the proposed scheme has an insignificant influence on plaintext data. Fig.10 also shows that the higher the value of α, the smaller the PSNR. This denotes



(a) (b) (c) (d) (e) (f)

FIGURE 9. Example of different typical color images for testing the proposed framework. ((a) is Lena, (b) is Airplane, (c) is Sailboat, (d) is House, (e) is Pepper, (f) is Splash).

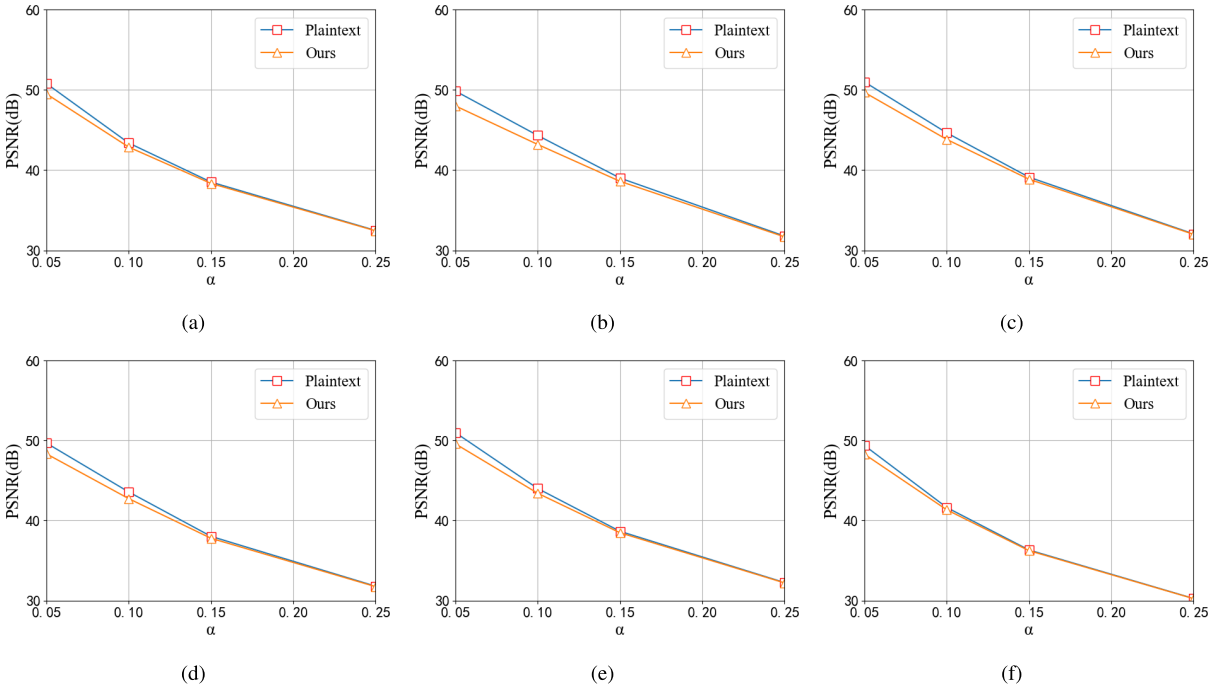


FIGURE 10. PSNR comparison between the proposed secure DWT-SVD-based image watermark embedding framework and the plaintext version. ((a) is Lena, (b) is Airplane, (c) is Sailboat, (d) is House, (e) is Pepper, (f) is Splash).

that the imperceptibility of the watermark is weakened, and the robustness of the watermark is improved simultaneously. The user can control the balance between robustness and imperceptibility by selecting the appropriate α according to the needs of a specific practical application.

C. IMAGE ATTACK EXPERIMENT

In this study, Lena and Splash are used to test the performance of the scheme-based watermark embedding method under different image attacks; Lena has abundant details, while Splash does not. The scaling factor α is 0.01. As shown in Fig.11, the image quality is clearly affected by an increase in α. The scaling factor of 0.01 indicates high imperceptibility and thus is suitable for practical applications. High levels of imperceptibility is helpful in evaluating the ability of the scheme to resist attack. Various common attacks are employed: Mean filtering, Median filtering, Gaussian noise, Salt & Pepper noise, Rotation, Crop, Mean Blur, Gaussian

(a) (b) (c)

FIGURE 11. Embedding the watermark into the color Lena image by different α. ((a) is 0.01, (b) is 0.55, (c) is 1).

Blur, JPEG Compress. The Mean filtering and Median filtering windows are 3×3 and 5×5. The variance coefficient of Gaussian noise is separately set to 0.01 and 0.02, and the mean coefficient is 0.001. Salt & Pepper noise coefficient is 0.02 and 0.1. Rotation attacks use rotation angles of 15°, 30°, 45°, 90°, 135° and 180°. The shear range of Crop is (100,100) and (150,150). The kernel size of Mean Blur and Gaussian

TABLE 3. NCC value of extracted watermark under different attacks.

Attack types	Splash	Lena
Mean filtering (3×3)	0.6031	0.6121
Mean filtering (5×5)	0.5682	0.5144
Median filtering (3×3)	0.7417	0.7421
Median filtering (5×5)	0.6191	0.6299
Gaussian noise(0.01)	0.5914	0.6003
Gaussian noise(0.02)	0.5791	0.5876
Salt & Pepper noise(0.02)	0.5677	0.6071
Salt & Pepper noise(0.1)	0.5052	0.5446
Rotation (15°)	0.5417	0.5312
Rotation (30°)	0.5541	0.5381
Rotation (45°)	0.5346	0.5480
Rotation (90°)	0.9341	0.9074
Rotation (135°)	0.5307	0.5303
Rotation (180°)	0.9746	0.9166
Crop(100,100)	0.5194	0.5134
Mean Blur(3,3)	0.5123	0.5021
Gaussian Blur(3,3)	0.5135	0.5109
JPEG Compression(0.6)	0.5227	0.5321
JPEG Compression(0.7)	0.5691	0.5709

Blur is (3,3). The JPEG Compression attack employed different compression ratio with 0.6 and 0.7. In general, normalized cross-correlation (NCC) can intuitively evaluate the quality of extracted data, and can be used to measure the similarity between the original watermark and the watermark extracted from the image. This calculation is given in (15).

$$\mathrm{NCC} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} \boldsymbol{W}(x,h)\cdot\tilde{\boldsymbol{W}}(x,h)}{\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\boldsymbol{W}^2(x,h)}\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{\boldsymbol{W}}^2(x,h)}}. \tag{15}$$

where $\boldsymbol{W}(x,h)$ is the original watermark, and $\tilde{\boldsymbol{W}}(x,h)$ is the extracted watermark. Additionally, $\boldsymbol{W}(x,h)$ and $\tilde{\boldsymbol{W}}(x,h)$ are normalized into the range of $\{-1, 1\}$. The NCC value ranges from 0 to 1; the larger the value, the better the performance of the watermarking scheme. Table 3 shows the NCC value of the extracted watermark from the host image after different attacks. Table 3 shows that decrypted images of the privacy-preserving watermarking scheme proposed in this study can more effectively resist a median filtering attack after image decryption than a rotation attack (except in cases of 90° and 180° rotation as there is no information loss), and its NCC value after being subjected to a median filtering attack can be maintained at approximately 0.65. Fig.12 further verifies that the proposed scheme can effectively resist a median filtering attack, and the extracted watermark image remains visible. In other attacks, the extracted watermark can also display a rough image and determine ownership disputes.

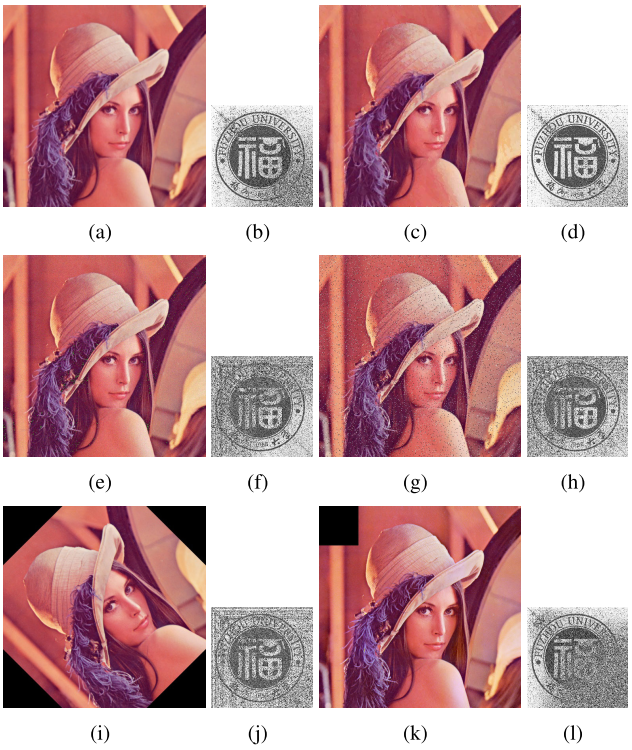


FIGURE 12. Watermarked host image and watermark image extracted after attack. ((a) is mean filtering (3 × 3), (c) is median filtering (3 × 3), (e) is Gaussian noise (0.01), (g) is Salt & Pepper (0.02), (i) is rotation (45°), (k) is Crop (100,100), (b), (d), (f), (h), (j) and (l) are extracted watermarks).

TABLE 4. Average NCC value comparison of extracted watermark image after different attacks.

Attack Types	Proposed framework	[46]	[47]
Mean filtering (3×3)	0.6331	0.6033	0.7041
Median filtering (3×3)	0.7417	0.7174	0.6411
Gaussian noise(0.01)	0.6144	0.6348	0.5163
Rotation (90°)	0.9541	0.9287	0.9139
Salt & Pepper noise (0.02)	0.5519	0.5268	0.4894
Crop (100,100)	0.5142	0.4854	0.4947

We use three images (Lena, Airplane, Splash) to test the robustness, and average the NCC values of watermarks extracted from host images after different attacks and compare NCC with the other two schemes. Table 4 shows the robustness comparisons of the proposed watermarking scheme with the schemes of [46] and [47] with embedding

intensity of 0.01; the NCC is used for comparison. Our proposed scheme is more robust than other two schemes in Median filtering, Rotation (90°), Salt & Peeper noise and Crop (100,100). In [47], the watermark was embedded into the low-frequency sub-band of the host image after DWT. Additionally, [46] hides the watermark in the middle frequency band of the host image. In the proposed algorithm, the sub-band was further decomposed by SVD and watermark was embedded into the singular value of the sub-bands. As singular value has the property of geometric invariance, the low frequency sub-bands of DWT are not sensitive to various noises. This indicates that the proposed algorithm performs better when subjected to various types of attacks.

VI. CONCLUSION

In this paper, a Haar discrete wavelet transform scheme in the encryption domain was initially proposed. A watermark embedding framework based on edge computing for privacy protection was subsequently proposed, which successfully implements the watermark embedding process of privacy preservation in an insecure outsourcing environment, to achieve an embedding effect similar to that of plaintext domain. Compared with the Paillier cryptosystem commonly used in DWT in the encryption domain, the proposed scheme presents a significant improvement in encryption and decryption rates. Compared with Paillier, the data extension is reduced by about 50% in the ciphertext, which makes it feasible for practical applications.

REFERENCES

[1] S. Rewaria, "Data privacy in social media platform: Issues and challenges," SSRN, Tech. Rep. 3793386, 2021.

[2] M. Begum and M. S. Uddin, "Digital image watermarking techniques: A review," *Information*, vol. 11, no. 2, p. 110, Feb. 2020.

[3] R. Maher and O. A. Nasr, "DropStore: A secure backup system using multi-cloud and fog computing," *IEEE Access*, vol. 9, pp. 71318–71327, 2021.

[4] K. M. Hosny, M. M. Darwish, and M. M. Fouda, "New color image zero-watermarking using orthogonal multi-channel fractional-order legendre-Fourier moments," *IEEE Access*, vol. 9, pp. 91209–91219, 2021.

[5] L.-Y. Hsu and H.-T. Hu, "QDCT-based blind color image watermarking with aid of GWO and DnCNN for performance improvement," *IEEE Access*, vol. 9, pp. 155138–155152, 2021.

[6] W. Huan, S. Li, Z. Qian, and X. Zhang, "Exploring stable coefficients on joint sub-bands for robust video watermarking in DT CWT domain," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Jun. 24, 2021, doi: 10.1109/TCSVT.2021.3092004.

[7] L. Zhu, X. Luo, Y. Zhang, C. Yang, and F. Liu, "Inverse interpolation and its application in robust image steganography," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 24, 2021, doi: 10.1109/TCSVT.2021.3107342.

[8] P. Yang, Y. Lao, and P. Li, "Robust watermarking for deep neural networks via bi-level optimization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 14841–14850.

[9] J. Zhang, D. Chen, J. Liao, W. Zhang, H. Feng, G. Hua, and N. Yu, "Deep model intellectual property protection via deep watermarking," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Mar. 9, 2021, doi: 10.1109/TPAMI.2021.3064850.

[10] M. Xiao, X. Li, Y. Wang, Y. Zhao, and R. Ni, "Reversible data hiding based on pairwise embedding and optimal expansion path," *Signal Process.*, vol. 158, pp. 210–218, May 2019.

[11] M. Ishtiaq, W. Ali, W. Shahzadm, M. A. Jaffar, and Y. Nam, "Hybrid predictor based four-phase adaptive reversible watermarking," *IEEE Access*, vol. 6, pp. 13213–13230, 2018.

[12] T.-S. Nguyen, C.-C. Chang, and X.-Q. Yang, "A reversible image authentication scheme based on fragile watermarking in discrete wavelet transform domain," *AEU-Int. J. Electron. Commun.*, vol. 70, no. 8, pp. 1055–1061, Aug. 2016.

[13] Y.-Q. Shi, X. Li, X. Zhang, H.-T. Wu, and B. Ma, "Reversible data hiding: Advances in the past two decades," *IEEE Access*, vol. 4, pp. 3210–3237, 2016.

[14] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Process. Lett.*, vol. 18, no. 4, pp. 255–258, Apr. 2011.

[15] W. Hong, T.-S. Chen, and H.-Y. Wu, "An improved reversible data hiding in encrypted images using side match," *IEEE Signal Process. Lett.*, vol. 19, no. 4, pp. 199–202, Apr. 2012.

[16] K. Ma, W. Zhang, X. Zhao, N. Yu, and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 3, pp. 553–562, Mar. 2013.

[17] W. Zhang, K. Ma, and N. Yu, "Reversibility improved data hiding in encrypted images," *Signal Process.*, vol. 94, pp. 118–127, Jan. 2014.

[18] X. Zhang, J. Long, Z. Wang, and H. Cheng, "Lossless and reversible data hiding in encrypted images with public-key cryptography," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1622–1631, Sep. 2016.

[19] D. Bouslimi, R. Bellafqira, and G. Coatrieux, "Data hiding in homomorphically encrypted medical images for verifying their reliability in both encrypted and spatial domains," in *Proc. 38th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2016, pp. 2496–2499.

[20] H.-T. Wu, Y.-M. Cheung, and J. Huang, "Reversible data hiding in Paillier cryptosystem," *J. Vis. Commun. Image Represent.*, vol. 40, pp. 765–771, Oct. 2016.

[21] S. Xiang and X. Luo, "Efficient reversible data hiding in encrypted image with public key cryptosystem," *EURASIP J. Adv. Signal Process.*, vol. 2017, no. 1, pp. 1–13, Dec. 2017.

[22] X. Liang, S. Xiang, L. Yang, and J. Li, "Robust and reversible image watermarking in homomorphic encrypted domain," *Signal Process., Image Commun.*, vol. 99, Nov. 2021, Art. no. 116462.

[23] Z. Qian, X. Zhang, and S. Wang, "Reversible data hiding in encrypted JPEG bitstream," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1486–1491, Aug. 2014.

[24] F. Peng, Z.-X. Lin, X. Zhang, and M. Long, "Reversible data hiding in encrypted 2D vector graphics based on reversible mapping model for real numbers," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 9, pp. 2400–2411, Sep. 2019.

[25] F. Peng, W.-Y. Jiang, Y. Qi, Z.-X. Lin, and M. Long, "Separable robust reversible watermarking in encrypted 2D vector graphics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2391–2405, Aug. 2020.

[26] H. Gao and T. Gao, "Double verifiable image encryption based on chaos and reversible watermarking algorithm," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 7267–7288, Mar. 2019.

[27] Y. Yao, W. Zhang, H. Wang, H. Zhou, and N. Yu, "Content-adaptive reversible visible watermarking in encrypted images," *Signal Process.*, vol. 164, pp. 386–401, Nov. 2019.

[28] S. Haddad, G. Coatrieux, A. Moreau-Gaudry, and M. Cozic, "Joint watermarking-encryption-JPEG-LS for medical image reliability control in encrypted and compressed domains," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2556–2569, 2020.

[29] T. K. Araghi and A. A. Manaf, "An enhanced hybrid image watermarking scheme for security of medical and non-medical images based on DWT and 2-D SVD," *Future Gener. Comput. Syst.*, vol. 101, pp. 1223–1246, Dec. 2019.

[30] N. Bisla and P. Chaudhary, "Comparative study of DWT and DWT-SVD image watermarking techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 6, pp. 821–825, 2013.

[31] P. Zheng and J. Huang, "Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2455–2468, Jun. 2013.

[32] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1999, pp. 223–238.

[33] T. Bianchi, A. Piva, and M. Barni, "On the implementation of the discrete Fourier transform in the encrypted domain," *IEEE Trans. Inf. Forensics Security*, vol. 4, no. 1, pp. 86–97, Mar. 2009.

[34] T. Bianchi, A. Piva, and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems," *EURASIP J. Inf. Secur.*, vol. 2009, Dec. 2009, Art. no. 716357.

[35] S.-J. Xiang, X.-R. Luo, and S.-X. Shi, "A novel reversible image watermarking algorithm in homomorphic encrypted domain," *Chin. J. Comput.*, vol. 39, no. 3, pp. 571–581, 2016.
[36] S. Chen, R. Lu, and J. Zhang, "A flexible privacy-preserving framework for singular value decomposition under Internet of Things environment," in *Proc. IFIP Int. Conf. Trust Manage.* Cham, Switzerland: Springer, 2017, pp. 21–37.
[37] M. Satyanarayanan, "The emergence of edge computing," *Comput.*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
[38] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image Vis. Comput.*, vol. 24, no. 9, pp. 926–934, 2006.
[39] Y. Liu, S. Tang, R. Liu, L. Zhang, and Z. Ma, "Secure and robust digital image watermarking scheme using logistic and RSA encryption," *Expert Syst. Appl.*, vol. 97, pp. 95–105, May 2018.
[40] M. Shensa, "The discrete wavelet transform: Wedding the a trous and Mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.
[41] R. S. Stanković and B. J. Falkowski, "The Haar wavelet transform: Its status and achievements," *Comput. Electr. Eng.*, vol. 29, no. 1, pp. 25–44, Jan. 2003.
[42] V. S. Verma and R. K. Jha, "An overview of robust digital image watermarking," *IETE Tech. Rev.*, vol. 32, no. 6, pp. 479–496, Nov. 2015.
[43] H. Cheng, X. Liu, H. Wang, Y. Fang, M. Wang, and X. Zhao, "SecureAD: A secure video anomaly detection framework on convolutional neural network in edge computing environment," *IEEE Trans. Cloud Comput.*, early access, Apr. 27, 2020, doi: 10.1109/TCC.2020.2990946.
[44] H. Cheng, H. Wang, X. Liu, Y. Fang, M. Wang, and X. Zhang, "Person re-identification over encrypted outsourced surveillance videos," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1456–1473, Jun. 2021.
[45] D. Giry. (2017). *Keylength-Nist Report on Cryptographic Key Length and Cryptoperiod (2016)*. Accessed: Aug. 1, 2017. [Online]. Available: https://www. keylength.com/en/4/
[46] B. J. Saha, C. Pradhan, K. K. Kabi, and A. K. Bisoi, "Robust watermarking technique using Arnold's transformation and RSA in discrete wavelets," in *Proc. Int. Conf. Inf. Syst. Comput. Netw. (ISCON)*, Mar. 2014, pp. 83–87.
[47] P. V. V. Kishore, N. Venkatram, C. Sarvya, and L. S. S. Reddy, "Medical image watermarking using RSA encryption in wavelet domain," in *Proc. 1st Int. Conf. Netw. Soft Comput. (ICNSC)*, Aug. 2014, pp. 258–262.

HANG CHENG received the B.S. and M.S. degrees in applied mathematics from Fuzhou University, Fuzhou, China, in 2002 and 2005, respectively, and the Ph.D. degree in signal and information processing with Shanghai University, Shanghai, China, in 2016. From August 2018 to August 2019, he held the position of a Research Scholar at the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. He is currently an Associate Professor with the Department of Information and Computational Science, School of Mathematics and Statistics, Fuzhou University. He has published more than 30 articles on the topics of privacy protection and cloud security, including, but not limited to, articles in the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, the IEEE SIGNAL PROCESSING LETTERS, and the *Information Sciences* (Elsevier). His current research interests include multimedia security, image processing, cryptography, and information hiding.

QINJIAN HUANG is currently pursuing the master's degree with the College of Computer Science and Big Data, Fuzhou University, Fuzhou, China. His current research interest includes privacy preserving computation.

FEI CHEN received the Ph.D. degree in signal and information processing from Zhejiang University, Hangzhou, China, in 2013. He is currently an Associate Professor with the College of Computer and Data Science, Fuzhou University. His current research interests include machine learning, computer vision, and deep learning techniques in image processing.

MEIQING WANG received the B.S. and M.S. degrees in applied mathematics from Tsinghua University, Beijing, China, in 1987 and 1989, respectively, and the Ph.D. degree from the Department of Computing, Xi'an Jiaotong University, China, in 2002. She is currently a Professor with the Department of Information and Computational Science, School of Mathematics and Statistics, Fuzhou University, Fuzhou, China. Her current research interests include computing science, image processing, and computational finance.

WANXI YAN is currently pursuing the master's degree with the School of Mathematics and Statistics, Fuzhou University, Fuzhou, China. Her current research interest includes privacy-preserving image retrieval.