Research Article

Achieving Lightweight Privacy-Preserving Image Sharing and Illegal Distributor Detection in Social IoT

Tianpeng Deng,¹ Xuan Li ,^{1,2} Biao Jin,¹ Lei Chen,³ and Jie Lin¹

¹College of Mathematics and Informatics, Fujian Normal University, Fuzhou 350117, China

²Fujian Provincial Key Laboratory of Network Security and Cryptology, Fuzhou 350117, China

³College of Engineering and Computing, Georgia Southern University, Statesboro, GA 30458, USA

Correspondence should be addressed to Xuan Li; jessieli24@163.com

Received 21 February 2021; Revised 31 March 2021; Accepted 20 May 2021; Published 14 June 2021

Academic Editor: David Megías

Copyright © 2021 Tianpeng Deng et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The applications of social Internet of Things (SIoT) with large numbers of intelligent devices provide a novel way for social behaviors. Intelligent devices share images according to the groups of their specified owners. However, sharing images may cause privacy disclosure when the images are illegally distributed without owners' permission. To tackle this issue, combining blind watermark with additive secret sharing technique, we propose a lightweight and privacy-preserving image sharing (LPIS) scheme with illegal distributor detection in SIoT. Specifically, the query user's authentication information is embedded in two shares of the transformed encrypted image by using discrete cosine transform (DCT) and additive secret sharing technique. The robustness against attacks, such as JPEG attack and the least significant bit planes (LSBs) replacement attacks, are improved by modifying 1/8 of coefficients of the transformed image. Moreover, we adopt two edge servers to provide image storage and authentication information embedding services for reducing the operational burden of clients. As a result, the identity of the illegal distributor can be confirmed by the watermark extraction of the suspicious image. Finally, we conduct security analysis and ample experiments. The results show that LPIS is secure and robust to prevent illegal distributors from modifying images and manipulating the embedded information before unlawful sharing.

1. Introduction

The continuous and rapid development of the Internet promotes the interconnections of things. However, the Internet is a double-edged sword. On the one hand, it brings about the rise of emerging fields and the integration of multiple fields, such as social networks [1–4], Internet of Things (IoT) [5–9], and Industrial IoT [10, 11]. On the other hand, it also brings a lot of problems, such as malicious information dissemination [12] and privacy disclosure [13, 14]. Known as the new generation of Internet, IoT is becoming more prevalent with the development of 5G [15] and expands the forms of communications among people to people and things [16]. Moreover, IoT allows various types of devices to perform local tasks with the help of the cloud [17]. However, the response speed from the cloud is generally slow to support the demand of IoT devices. In order to

improve the response speed to IoT devices, edge computing adopts devices and platforms with computing, storage, and application to provide nearby services for IoT devices [18–20]. After assigning social behaviors to the communication processes, IoT devices can communicate with each other for the purpose of social activities [21], which contribute to the development of social Internet of Things (SIoT) [22–24]. SIoT is rich in applications and has many data sources and types, such as images, text, and sound, which greatly facilitates people's daily life and work.

As a broadly used carrier of information, images are widely used in SIoT. They can not only record wonderful moments, such as scenery and ethnic activities, but also hold a lot of personal information which may be sensitive to users' privacy, such as facial information. When an image is captured by a mobile camera, the sensor itself cannot determine whether the image contains private information. To

share the information in SIoT, the sensor will share the image according to the group set by its owner. After the image is shared, other users' operations on the image will not be controlled by the image owner, and others may modify or illegally distribute the image. In this case, the owner's privacy will be threatened. Access control is a widely used security mechanism that allows only selected groups of authorized users to access images through the designed protocols [25–28]. When an authorized user obtains an image, the image can be fully manipulated because the user's behavior is often autonomous, unpredictable, and not controlled by the image owner. Therefore, access control is restricted in the case of illegal distribution of images by authorized users. Data hiding techniques are suitable for authentication by embedding hidden data in an image [29]. Embedding data in the least significant bit planes (LSBs) is a common way for hiding. Zhang et al. [30] used stream cipher for the encryption and divided the encrypted image into different blocks. Then, each bit of information is embedded in a block by changing half of pixels' LSBs of the block. Based on [30], Xia et al. [29] proposed a privacy-preserving image retrieval scheme with copy detection. However, these methods are pregnable to attacks but maintain a good visual quality, because the embedded data can be erased by modifying the values of LSBs. To improve the robustness against attacks, researchers have turned their eyes on frequency domain embedding [31]. Singh et al. [32] transformed the cover image through the combination of discrete wavelet transformation (DWT), discrete cosine transform (DCT), and singular value decomposition (SVD) and then embedded image watermark and text watermark in the different components of these coefficients. Zear et al. [33] enriched the diversities of watermarks for adapting the medical needs. Although these methods have good performance in high embedding capacity, they do not support embedding in encrypted image, because values in transformed images are always floating point numbers. Moreover, these methods are not as lightweight as we desired for meeting SIoT applications.

Considering the uncontrollability and unpredictability of authorized users' behaviors in SIoT, we propose a lightweight and privacy-preserving image sharing (LPIS) scheme with the capability of illegal distributor detection, aiming at dealing with the potential illegal distribution problem of authorized users. In this paper, illegal distributors may attempt to remove the embedded information for the purpose of illegally sharing images. Direct embedding in the spatial domain is usually vulnerable to LSB attacks [30]. Moreover, JPEG compression technique regards the information embedded in the spatial domain as redundant information and may cause the loss of information upon JPEG attack [34]. We embed the authentication information in the DCT domain by modifying a small number of coefficients of the transformed image for improving the robustness against these two attacks for the reason that the IDCT maps the embedded information to the overall brightness information of the block. Furthermore, the embedding should be operated on the encrypted image to further protect the privacy of the owner. Since the coefficients of the transformed image

are floating point numbers, and homomorphic encryption can greatly increase the burden of IoT devices, we adopt the additive secret sharing technique to provide lightweight encryption for preventing image content from being captured by semihonest edge servers. To support additive secret sharing, two edge servers are needed as operating platforms to provide storage and embedding operations. With the help of two edge servers, the image owner can accurately and conveniently identify illegal distributor's identity when there occurs a suspicious image. The main contributions are summarized as follows:

- (1) We propose the LPIS scheme with illegal distributor detection in SIoT. By adopting the additive secret sharing technique, the shared images are encrypted, and the embedding phase is supported to be operated in the encrypted domain. Moreover, the embedded information can be extracted without the original image.
- (2) We use DCT transformation to improve the robustness of resisting against JPEG attack. The visual quality of the embedded image is maintained since only a few coefficients in frequency domain of each block are modified. Meanwhile, we adopt additive secret sharing technique for lightweight encryption to protect the content of the outsourced image and reduce the operational burdens of the owner.
- (3) Security analysis and ample experiments show that LPIS is secure and robust to different attacks.

The rest of this paper is organized as follows: Section 2 details the problem of illegal distribution, outlines the system model and security model considered in this paper and then highlights the system goals. Section 3 describes the construction of LPIS. Section 4 shows the security analysis and experimental results. The conclusion is given in Section 5.

2. Problem Description

In this section, we present the system model of the LPIS scheme, followed by the security model and system goals.

2.1. System Model. In this paper, we aim at addressing the issue of illegal distribution in SIoT. When IoT devices communicate for social activities, device owners always protect their privacy by allowing information to be shared only within their designated groups. Members of a group tend to share the same interests, but may not necessarily protect the privacy of the data shared by the sharers. For example, when the sensor camera automatically shares the image which contains owner's private information according to the group specified by the owner, other users in the group (image users) may choose to ignore the owner's privacy and may share the image, which will cause the privacy disclosure of the image's owner. We consider such a situation and limitations and propose the LPIS scheme to quickly confirm the identity of the illegal distributor when there occurs a suspicious image. Moreover, to meet the high

response speed for IoT devices, we adopt edge servers to provide computing power and nearby services for IoT devices.

As illustrated in Figure 1, the system model in this paper mainly involves four entities: image owner (IO), image users (IUs), and two edge servers (S_1 and S_2). The IO collects images with sensors and outsources images to edge servers for relieving their local storage and for the convenience of image sharing. We denote the original outsourced image as I and transform it into frequency domain by DCT transformation. DCT is adopted to improve the robustness against JPEG attack. JPEG is a widely used compression technique which can remove the redundant information of the image. The information embedded in the spatial domain is always considered as redundant information, so direct embedding in the original image may cause information losses after JPEG compression. Embedding information in DCT domain may mitigate this effect, because IDCT will map the embedded information to the overall brightness information of the block. To protect the privacy of I_{DCT} , the IO first encrypts I_{DCT} by splitting it into two random shares, I' and I'' , where $I_{DCT} = I' + I''$. Then, I' and I'' are distributed to S_1 and S_2 , respectively. To obtain the shared image, the IU sends his authentication information AI to the servers, where AI is a binary image of his signature image and the values of AI belong to $\{0, 1\}$. The servers find the retrieved image and check if IU is on the list of authorized users specified by IO. Once the IU is an authorized user, S_1 and S_2 will generate two different random vectors representing the values of AI. S_1 and S_2 , respectively, embed their random vectors in the corresponding blocks of the image as required (see Section 3.2 for details) and send the embedded image to the IU. The IU just needs to add up two shares and use IDCT; then, the embedded image with authentication information I_{AI} will be obtained. For the entire process, we assume that all communication channels among entities are secure.

2.2. Security Model. Similar to [35], we set edge servers as semihonest entities, which can perform the protocols as defined but may try to learn private information as much as possible when they are interested in the stored data. Both edge servers S_1 and S_2 are out of the trusted domain of IO. Moreover, we assumed that two servers are noncolluding, which means that neither of servers will reveal more information than protocol messages [35]. This assumption is reasonable and practical because S_1 and S_2 may be managed by two different service providers. We also consider the external attackers $\mathcal{A}^*$ who may eavesdrop on communication channels to obtain encrypted data. In addition, we define the illegal distributors $\mathcal{A}^{**}$ as those who are authorized IUs but illegally distribute shared images for their personal benefits. We have defined illegal distributors $\mathcal{A}^{**}$ separately from external attackers $\mathcal{A}^*$, meaning that they will have different behaviors and no conversion between $\mathcal{A}^{**}$ and $\mathcal{A}^*$ is allowed. Meanwhile, we enhance the attack capability of $\mathcal{A}^{**}$ to show that LPIS have good robustness

against $\mathcal{A}^{**}$, while allowing illegal distributors to collude with either S_1 or S_2 . The specific attack capabilities of the two are shown as follows:

- (i) $\mathcal{A}^*$ may reach a *compromise* with S_1 or S_2 to guess plaintexts of all ciphertexts outsourced by IO or the ciphertexts sent from S_1 or S_2 through an interactive protocol.
- (ii) $\mathcal{A}^{**}$ may *illegally distribute* the embedded image for their own benefits after destroying the embedded information. This means that $\mathcal{A}^{**}$ are authorized users but do not obey the rules of image sharing as IO expects. At the same time, $\mathcal{A}^{**}$ may attempt to modify the obtained image to remove the authentication information embedded in the image.

Note that $\mathcal{A}^*$ and $\mathcal{A}^{**}$ are forbidden to compromise both S_1 and S_2 simultaneously. $\mathcal{A}^*$ are not allowed to become authorized users in our system either.

2.3. System Goals. The system is expected to achieve the goals of both security and lightweight with high visual performance. In this paper, we assume that edge servers may attempt to learn some private information about IO from the encrypted image while following protocol steps to complete the assigned storage or processing tasks. Therefore, the system treats them as semitrusted service providers. The system goals of the scheme are described as follows:

- (1) Security: security is the primary goal in our scheme. Our system will suffer threats from two kinds of entities: edge servers and illegal distributors.
 - (a) Security for edge servers: the edge servers are assumed to be honest-but-curious, so the content of the image should be invisible to edge servers during the image sharing process. When being operated in the encrypted form, the image data will never be leaked to any of the edge servers.
 - (b) Robustness against illegal distributors' attacks: it is difficult to prevent the occurrence of illegal distribution, but we can confirm illegal distributor's identity as soon as a suspicious image is found. Therefore, the scheme should be robust against various attacks.
- (2) Lightweight and efficiency: computational processes should be executed on the server side, and the costs at both owners and users should be as low as possible. Because the edge servers are always service providers and the clients are service users, the computational operations should be executed by servers to lower the burden of clients. In addition, IoT devices are always resource-constraint and require timely service responses, so the transmission cost should be as low as possible, and the computational time cost on the server side should be kept as low as possible. The encryption method and secure

FIGURE 1: System architecture.

computation techniques we used are lightweight and suitable for IoT scenarios, and edge servers are able to provide high speed responses for IoT devices.

In the image sharing system, the detection of illegal distributors should not affect the performance of image sharing, which means that the embedding of authentication data should not affect the visual quality of the image. This is the basic requirement of the normal use of shared images.

3. Construction of the LPIS Scheme

In this section, we introduce the implementation of LPIS scheme with illegal distributor detection in SIoT, aiming at achieving the goals as described in Section 2.3. The whole process of LPIS is also shown in Figure 2.

3.1. Image Sharing. Before sending the encrypted image to edge servers, the IO uses DCT transformation to improve the robustness of the scheme, such as hiding the original pixel value of the image and resisting JPEG attack:

Step 1. Image transformation: since two edge servers are curious-but-honest entities, IO needs to convert the image from spatial domain to transformed domain by using DCT to protect the original pixel value and the image content. The IO divides I into 8×8 blocks of pixels in spatial domain and applies blockwise DCT transformation for all blocks. We denote the transformed image as I_{DCT} .

$$I_{\text{DCT}}(u, v) = \frac{2}{\sqrt{mn}} c(u)c(v) \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(i, j) \cos\left[\frac{(i+0.5)\pi}{m}u\right] \cos\left[\frac{(j+0.5)\pi}{n}v\right], c(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0, \\ 1, & x \neq 0. \end{cases} \quad (1)$$

Here, $I(i, j)$ is the image's original pixel value, $I_{\text{DCT}}(u, v)$ is the transformed coefficient, and $M \times N$ is the size of the image. Each DCT raw coefficient matrix's first entry is called Direct Current component and the rest are Alternating Current components. Our scheme is to alter a few AC components to achieve the purpose of embedding authentication information.

Step 2. Image encryption: to protect the privacy of I_{DCT} , the IO encrypts it based on the additive secret sharing technique. For the transformed image I_{DCT} of size $M \times N$, the IO first generates a matrix R of the same size as I_{DCT} randomly. For protecting the content of I_{DCT} , the elements of R are randomly and uniformly distributed over a large interval $[-10^{n-1}, 10^{n-1} - 1]$, where $10^{n-1} - 1$ should be larger than the largest element of I_{DCT} , and n acts as the security parameter. Here, we set $n = 8$. The IO then encrypts I_{DCT} by

splitting it into two shares of elements, such as $I' = R$ and $I'' = I_{\text{DCT}} - R$. Finally, I' and I'' are, respectively, sent to two edge servers S_1 and S_2 .

3.2. Query and Information Embedding. When IUs from the same social circle request a shared image they need, they should send their identity information to the edge servers. Here, we use the signature image signed by IU as the identity information and transform it into **binary image** of size $W \times H$ which is denoted as AI. Note that the edge servers will receive the same AI, and only when the AI passes the validation on both of edge servers, will any further action be taken. Otherwise, the request will be rejected.

Step 1. Since the numbers the IO encrypted are decimals, S_1 and S_2 should first convert the encrypted numbers into integers by multiplying the numbers by

FIGURE 2: Flowchart of LPIS.

10^p , where p is the precision of the number they desire. It is worth noting that the operation of multiplying 10^p is in the encrypted domain, and no privacy will be leaked. The correctness can be demonstrated as follows:

For a shared number $w = w_1 + w_2$, two servers S_i ($i = 1, 2$) compute $\bar{w}_i = \lfloor w_i \times 10^p \rfloor$, respectively, by using the same p (e.g., the hash value of identity information can be used as p). Then,

$$\bar{w} = \bar{w}_1 + \bar{w}_2 = \lfloor w_1 \times 10^p \rfloor + \lfloor w_2 \times 10^p \rfloor \approx \lfloor (w_1 + w_2) \times 10^p \rfloor = \lfloor w_1 + w_2 \rfloor \times 10^p, \quad (2)$$

where $\lfloor \cdot \rfloor$ denotes the round-down operation. After $\bar{w}$ is divided by 10^p , the result is similar to w when ignoring the accuracy of rounded decimals.

Step 2. Both I' and I'' , sized $M \times N$, are divided into blocks of size $l \times l$ (usually $l = 8$ to match the block size of DCT transformation). In order to guarantee that the following steps are well processed, the AI needs to be resized to $w \times h$, which satisfies $w \leq \lfloor M/l \rfloor$ and $h \leq \lfloor N/l \rfloor$ at the same time. By doing so, each pixel of resized AI corresponds to a block of I' and I'' .

Step 3. S_1 and S_2 , respectively, generate two different pseudorandom vectors, k_1 and k_2 , of length l . The value of k_i ($i = 1, 2$) belongs to $[0, 1]$. S_1 holds k_1 and S_2 holds k_2 . Each vector represents a value of AI, in which the value belongs to $\{0, 1\}$. For example, vector k_1 represents the value 0 and k_2 indicates 1. We also introduce the intensity indicator α to control the strength of k_i ($i = 1, 2$) to be added (3). Note that α is fixed before the embedding phase and is a constant value which will not be changed across different images. In particular, different α values have different effects on image quality as shown in Section 4.2.

$$K_i = \alpha \cdot k_i, \quad i = 1, 2. \quad (3)$$

Step 4. For each side of edge servers, S_1 (S_2) only embeds K_1 (K_2) in the diagonal elements of the blocks

of I' (I'') where the corresponding pixel value of AI is 0 (1). Note that both K_1 and K_2 are multiplied by 10^p to be consistent with I' and I'' . We denote the embedded shares of I as I'_{AI} and I''_{AI} .

Step 5. S_1 and S_2 send I'_{AI} and I''_{AI} to IU after they are divided into 10^p , respectively.

The detailed algorithm is given in Algorithm 1.

After the embedding phase, we obtain an encrypted DCT image with authentication information in it. Note that the information embedding phase is performed on both of the edge servers, respectively, and each server embeds its own part of AI. When IU receives the two shares of the image, he only needs to add up these two shares and use IDCT to convert the image into spatial domain, and then the embedded image with authentication information will be obtained. There are two main purposes for this design. First, the robustness of LPIS against attackers is improved. In this scheme, the authentication information is not embedded in a fixed pixel with original AI value, but with two random vectors instead, so IU cannot easily extract the embedded data while knowing the embedding locations. Second, the quality of the image is maintained. After IDCT transformation, the embedded information will be mapped to the overall brightness information of the block. Because the same intensity indicator α is used and only a few

Input: Encrypted shares of the image: I' and I'' of size $M \times N$; pseudorandom vectors k_1 and k_2 of length l ; authentication information AI of size $w \times h$; embedding intensity indicator α .

Output: Encrypted shares of the image with authentication information I_{AI}' and I_{AI}'' .

```

(1) for  $i \leftarrow 1: w$  do
(2) for  $j \leftarrow 1: h$  do
(3) @  $S_1$  side:
(4)  $x = i \times l, y = j \times l$ 
(5) if  $AI(i, j) = 0$  then
(6) for  $t \leftarrow 1: l$  do
(7)  $I_{AI}'[x+t, y+l-t] = I'[x+t, y+l-t] + K_1[t]$ 
(8) end
(9) end
(10) @  $S_2$  side:
(11)  $x = i \times l, y = j \times l$ 
(12) if  $AI(i, j) = 1$  then
(13) for  $t \leftarrow 1: l$  do
(14)  $I_{AI}''[x+t, y+l-t] = I''[x+t, y+l-t] + K_2[t]$ 
(15) end
(16) end
(17) end
(18) end

```

ALGORITHM 1: Authentication information embedding.

coefficients are modified, the brightness information changes in the same extent and the visual quality of the image is maintained.

3.3. Illegal Distributor Detection. When there occurs a suspicious image I_S , IO will identify the illegal distributor with the help of S_1 and S_2 . The following are detailed steps:

Step 1. IO transforms I_S from spatial domain into frequency domain by using DCT transformation and sends it to S_1 and S_2 .

Step 2. S_1 and S_2 share their private random vectors k_1 and k_2 with each other and calculate the correlation coefficient between each block's diagonal elements and two pseudorandom vectors, respectively. Subsequently, they can extract the authentication information AI' . As the block has the size $l \times l$, we first extract l diagonal elements from each block, denoted as $\vec{p}$, and then use the following formulation to calculate Pearson correlation between $\vec{p}$ and $k_i, i = 1, 2$.

$$r_i(\vec{p}, k_i) = \sum_{j=1}^l (\vec{p}_j - \bar{p}) \left(\frac{k_{i,j} - \bar{k}_i}{\sqrt{\sum_{j=1}^l (\vec{p}_j - \bar{p})^2 \sum_{j=1}^l (k_{i,j} - \bar{k}_i)^2}} \right), \quad i = 1, 2, \quad (4)$$

where $\bar{p} = \sum_{i=1}^l (\vec{p}_i)/n$ and $\bar{k}_i = \sum_{j=1}^l (k_{i,j})/n$.

Step 3. IO identifies illegal distributor's identity according to AI' .

Through the three aforementioned processes, LPIS can be established securely under the defined security model (Section 4.1). The embedding location of each block is set as intermediate frequency coefficient. Moreover, the embedding phase is operated in the encrypted domain by additive secret sharing technique, which proved to be lightweight and efficient in SIoT.

4. System Evaluation

4.1. Security Analysis. In LPIS, we set S_1 and S_2 to be honest-but-curious and noncollusive entities. Based on this

assumption, we need to prove that LPIS is secure under the threats defined in Section 2.2. Lemma 1 shows that the additive secret sharing technique is secure.

Lemma 1 (see [35]). If a random element r is uniformly distributed on $\mathbb{Z}_n$ and independent from any variable $x \in \mathbb{Z}_n$, then $r \pm x$ is also uniformly random and independent from x .

Proof of Lemma 1 is shown in [36]. Because the embedding of authentication information is independent of each edge server, the attacker cannot obtain the embedding information of the other edge server under the assumption that he can only collude with one edge server. To prove that LPIS is secure under outside attacks and the threat of illegal distribution, we use the definition in [37].

Definition 1 (see [37]). We say that a protocol is secure if there exists a probabilistic polynomial-time simulator S that can generate a view for the adversary A in the real world and the view is computationally indistinguishable from its real view.

As shown in Section 2.2, we defined two kinds of attackers: external attackers $\mathcal{A}^*$ who can collude with only one of the edge servers; illegal distributors $\mathcal{A}^{**}$ who are authorized users but illegally distribute the obtained image.

Theorem 1. *LPIS is secure under threats of external attackers $\mathcal{A}^*$.*

Proof. Since the embedding is operated by S_1 and S_2 locally without any interactions, they can be perfectly simulated. If $\mathcal{A}^*$ compromise S_1 or S_2 , they can pass only one edge server's verification and obtain one share of the image and one of the random vectors. In LPIS, only those who can pass the verifications of both of S_1 and S_2 can obtain the shares of the image. Additionally, the one who compromises only one edge server can do nothing with the information he obtained. $\square$

Theorem 2. *LPIS is secure under threats of illegal distributors $\mathcal{A}^{**}$ who are authorized users.*

Proof. $\mathcal{A}^{**}$ are special attackers who are authorized users but destroy the privacy of IO for their own benefits. They will try to destroy the embedded information and distribute the image mainly by two ways. Firstly, they may try various attacks to destroy the embedded authentication information without severely damage the visual quality of embedded image. LPIS is robust to LSB replacement attack, JPEG compression attack, and other attacks which will be shown in Section 4.2.3. Secondly, they may collude with either S_1 or S_2 to eliminate the embedded information and obtain the plaintext of the image. Because of the defined large interval, the pixel values are uniformly split into that interval, so the image content is well protected from statistical analysis. The plaintext of the image is not obtainable by only colluding with one server. In addition, when $\mathcal{A}^{**}$ obtain the share of S_i , they will know the secret embedded vector k_i and the plaintext of embedded information from S_i , while they cannot obtain the other part because it is not allowed to collude with the other server S_{1-i} . As a result, it is a hard problem to guess k_{1-i} that is the same as k_{1-i} to destroy the whole embedded information. Therefore, LPIS is secure under $\mathcal{A}^{**}$.

The above proofs show that our system is secure when there exist external attackers and semihonest entities. Especially for illegal distributors, our solution has countermeasures against them. Therefore, our scheme is secure. Specifically, when $\mathcal{A}^{**}$ are authorized users yet illegally distributing the image, IO can easily identify them by extracting their authentication information with the help of S_1 and S_2 . The authentication information of IUs is always unique and has one-to-one correlation with authorized users. $\square$

4.2. System Performance. In this section, we evaluate the performance of LPIS and the performance of resistance against attacks.

To evaluate the visual quality of the embedded image, we use the peak signal to noise ratio (PSNR) and the structural similarity index (SSIM) as the evaluation indicators. Usually, the higher the PSNR value, the better the image quality, the closer the SSIM to 1, and the better the image structure.

For an image I and a compared noisy image K , the definitions of PSNR and SSIM are as follows:

$$\text{PSNR} = 10 \cdot \log_{10} \left(\frac{\max^2(I)}{\text{MSE}} \right), \quad (5a)$$

$$\text{MSE} = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - K(i, j)]^2, \quad (5b)$$

where $\max(I)$ is the max gray value of I .

$$\text{SSIM}(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)}, \quad (6)$$

where μ_I and μ_K are the means of I and K , σ_I and σ_K are the variances, σ_{IK} is the covariance between I and K , $c_1 = (k_1 L)^2$, and $c_2 = (k_2 L)^2$. L is the dynamic range of pixel values, and k_1 and k_2 are always set to 0.01 and 0.03, respectively.

In experiments, we take "Lena," which is of size 512×512 , as the cover image I (Figure 3(a)) and denote the signature image "Alice," which is resized to 64×64 , as IU's authentication information (AI) (Figure 3(d)). When outsourcing I into S_1 and S_2 , the IO first transforms it into DCT domain (denoted as I_{DCT}) and randomly splits it into two shares, where $I_{\text{DCT}} = I' + I''$. Then, IO sends I' and I'' to S_1 and S_2 , respectively (Figures 3(b) and 3(c)).

4.2.1. Impact on Intensity Indicator α . As addressed earlier, the intensity indicator α will affect the intensity of the embedded vector, which in turn affects the visual quality of the embedded image. Here, we discuss the impact of α , where $\alpha = \{5, 10, 15, 20, 25, 30\}$. The random vectors to be embedded by S_1 and S_2 stay the same.

Figure 4 shows the embedded images and the extracted signature images under different α values. Associated with PSNR and SSIM shown in Table 1, with the increase of α , the embedded image's quality is affected more intensively, but the quality of extracted signature images becomes better. The greater the embedded value is, the greater the proportion in this DCT block will be; thus, the more the brightness of this block will then be affected after IDCT. For this reason, we make a trade-off between the quality of the image with embedded information and the quality of the extracted signature image, where $\alpha = 25$ is used as the embedding intensity in our following experiments. When using $\alpha = 25$, PSNR is greater than 30, which means that the image quality is not seriously affected (Figure 4(i)), but the quality of the extracted AI is maintained at a high level, and the SSIM value

FIGURE 3: (a) Original “Lena” image; (b) I' of S_1 ; (c) I'' of S_2 ; (d) original authentication information AI.

between the original AI and extracted AI reaches 1 (Figure 4(j)).

4.2.2. Computational Time Cost of LPIS. Since IoT devices are usually resource constrained, we try to make LPIS as lightweight as possible. One manifestation of lightweight is that the computational time cost of all operations should be low. We analyze the time cost of each operation step. As shown in Table 2, the total computational cost of LPIS is less than 0.2 s for an image of size 512×512 without considering the communication delay. The DCT transformation and encryption are operated by IO, and the time cost of these two operations is less than 0.1 s. The embedding operation is carried out in the cloud, and the average time cost of embedding 1 bit for each server is $11 \mu\text{s}$.

Moreover, we compare LPIS with other algorithms [30, 38–43] in terms of the embedding time cost. As the embedding capacities of each algorithm are different, we compare the average computational time cost of embedding 1 bit as criteria. Table 3 shows the comparison results among these algorithms, in which LPIS reaches a low computational time cost compared to current methods. Spatial domain embedding is the most direct way for hiding secret data, such as LSB flipping [30, 38] and LSB replacement [40]. Compared to existing spatial embedding methods in encryption domain, LPIS reaches high embedding efficiency. The efficiency of LPIS is similar to [41], but LPIS is more robust against JPEG attack because of the transformed domain embedding. Meanwhile, additive secret sharing technique is more efficient than the homomorphic encryption scheme [42], which meets the goal of lightweight for LPIS. Although [39] also used additive embedding as the embedding operation, it first calculated and retained the mean value of pixels in each set before embedding, which increased the computational embedding cost.

4.2.3. Performance under Attack. LPIS mainly aims at the scenario where there exist illegal distributors who try to remove the embedded data and distribute the image without

IO’s permission. Therefore, LPIS is robust against various attacks such as LSBs replacement attacks and JPEG attack. In many existing approaches, data is embedded in the LSB or the three LSBs [29, 30], which are not appropriate for illegal distributor detection in the cloud environment. Besides, IU can easily erase the embedded information (Figure 5(a)) while the visual quality of the image is not severely affected. However, setting more LSBs to zero will indeed damage the visual quality (Figures 5(b) and 5(c)). Note that the higher the bit plane is, the higher the proportion of the image brightness will be. After the LSBs of the block are deleted and the block is transformed into frequency domain, the changes of the block will also be transformed into the frequency domain corresponding to the original DCT coefficient, and the influence of each pixel on the DCT coefficients will be weakened. Therefore, LPIS has superior robustness against deleting LSBs attacks; even high bit planes are removed and the visual quality is severely affected (Figures 6(a)–6(c) and Figures 7(a)–7(c)).

JPEG is the most common format of images used in networks, which can greatly reduce the image size and maintain the visual quality by removing redundant data of the image. Here, we use MATLAB to implement JPEG compression with its default compress rate (Figure 6(d)). The extracted authentication information under JPEG attack is shown in Figure 7(d). The SSIM between extracted AI under JPEG attack and original AI is 1, which shows that LPIS can resist JPEG attack.

Furthermore, the image may probably be polluted by the white noise through the communication. The amplitude of white noise roughly follows Gaussian distribution. Here, we set $\sigma = 0.02, 0.1, 0.2, 0.5$ of Gaussian distribution to simulate different amplitudes of the white noise. As we can see in Figures 7(e)–7(h), the higher the value of σ is, the more severely the visual quality will be affected and the more severely the extracted AI is polluted. However, with the improvement of the quality of communication channels, a low value of σ is more likely to appear. We can therefore draw a conclusion that LPIS is robust enough to resist white noise attacks.

FIGURE 4: Visual performances of embedded images and extracted AIs under different intensity indicator α values. The first and third columns are embedded images; the second and fourth columns are extracted AIs. (a) $\alpha = 5$; (b) $\alpha = 5$; (c) $\alpha = 10$; (d) $\alpha = 10$; (e) $\alpha = 15$; (f) $\alpha = 15$; (g) $\alpha = 20$; (h) $\alpha = 20$; (i) $\alpha = 25$; (j) $\alpha = 25$; (k) $\alpha = 30$; (l) $\alpha = 30$.

TABLE 1: Image quality under different intensity indicator α values.

α	PSNR	SSIM
5	45.5	0.99
10	39.5	0.95
15	35.9	0.89
20	33.5	0.82
25	31.5	0.75
30	30.0	0.68

TABLE 2: Computational time cost of each operation in LPIS.

Operations	DCT transform	Image encryption	Embedding 1 bit for each server	Image decryption
Time cost	89.9 ms	9 ms	11 μ s	3.1 ms

TABLE 3: Comparison with other algorithms.

Algorithm	[30]	[38]	[39]	[40]
Embedding operation ¹ and domain ²	LSB Flp. (S)	LSB Flp. (S)	Add. Emb. (S)	LSB Rep. (S)
Embedding time per bit (μ s)	7×10^3	3.3×10^3	36	109
Algorithm	[41]	[42]	[43]	Our scheme
Embedding operation ¹ and domain ²	Loc. Map (S)	Hist. Shft. + Homo. Add. Emb. (S)	GWT + Alpha Ble. (T)	Add. Emb. (T)
Embedding time per bit (μ s)	9	1.0×10^5	20	11

¹LSB Flp.: LSB flipping; Add. Emb.: additive embedding; LSB Rep.: LSB replacement; Loc. Map: location map; Hist. Shft.: histogram shifting; Homo. Add. Emb.: homomorphic additive embedding; GWT: graph wavelets transform; Alpha Ble.: alpha blending. ²“S” represents spatial domain embedding; “T” represents transformed domain embedding.

FIGURE 5: Visual performances of the original image under LSBs replacement attacks: (a) setting 3 LSBs to zero; (b) setting 4 LSBs to zero; (c) setting 5 LSBs to zero.

FIGURE 6: Visual performances of the embedded image “Lena” under LSBs replacement attacks, JPEG attack, and white noise attacks: (a) deleting 3 LSBs; (b) deleting 4 LSBs; (c) deleting 5 LSBs; (d) JPEG attack; (e) white noise attack under $\sigma = 0.02$; (f) white noise attack under $\sigma = 0.1$; (g) white noise attack under $\sigma = 0.2$; (h) white noise attack under $\sigma = 0.5$.

FIGURE 7: Visual performance of extracted AIs under LSBs replacement attacks, JPEG attack, and white noise attacks corresponding to Figure 6: (a) deleting 3 LSBs; (b) deleting 4 LSBs; (c) deleting 5 LSBs; (d) JPEG attack; (e) white noise attack under $\sigma = 0.02$; (f) white noise attack under $\sigma = 0.1$; (g) white noise attack under $\sigma = 0.2$; (h) white noise attack under $\sigma = 0.5$.

5. Conclusion

In this paper, we proposed LPIS scheme to solve the problem of detecting illegal distributor in SIoT. In order to adapt to the IoT environment, we adopted the additive secret sharing technique to encrypt the shared image. LPIS can help the image owner identify illegal distributor's identity by the embedded authentication information, and no plaintext information of the image was leaked to any of edge servers. Moreover, the design and implementation of privacy-preserving embedding improved the robustness of detection. The embedded information cannot be removed easily by illegal users through traditional attacks or JPEG compression attack. In conclusion, LPIS can effectively support privacy-preserving image sharing in SIoT with illegal distributor detection.

Data Availability

The cover image data used to support the findings of this study are included within the article.

Disclosure

An earlier version of this paper has been presented as a conference paper in SpaCCS: International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61872088, 61872090, and 61872086); the Natural Science Foundation of Fujian Province (2019J01276); and the Guizhou Provincial Key Laboratory of Public Big Data Research Fund (2019BDKFJJ004).

References

- [1] D. Wu, B. Liu, Q. Yang, and R. Wang, "Social-aware cooperative caching mechanism in mobile social networks," *Journal of Network and Computer Applications*, vol. 149, Article ID 102457, 2020.
- [2] F. Ali, S. El-Sappagh, S. M. R. Islam et al., "An intelligent healthcare monitoring framework using wearable sensors and social networking data," *Future Generation Computer Systems*, vol. 114, pp. 23–43, 2021.
- [3] J. A. Benítez-Andrades, I. García-Rodríguez, C. Benavides, H. Alaiz-Moretón, and J. E. Labra Gayo, "An ontology-based multi-domain model in social network analysis: experimental validation and case study," *Information Sciences*, vol. 540, pp. 390–413, 2020.
- [4] Y. Pu, T. Xiang, C. Hu, A. Alrawais, and H. Yan, "An efficient blockchain-based privacy preserving scheme for vehicular social networks," *Information Sciences*, vol. 540, pp. 308–324, 2020.
- [5] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.
- [6] J. Xiong, L. Chen, M. Z. A. Bhuiyan et al., "A secure data deletion scheme for iot devices through key derivation encryption and data analysis," *Future Generation Computer Systems*, vol. 111, pp. 741–753, 2020.

- [7] X. Li, J. Li, S. Yiu, C. Gao, and J. Xiong, "Privacy-preserving edge-assisted image retrieval and classification in iot," *Frontiers of Computer Science*, vol. 13, no. 5, pp. 1136–1147, 2019.
- [8] Q. Jiang, N. Zhang, J. Ni, J. Ma, X. Ma, and K.-K. R. Choo, "Unified biometric privacy preserving three-factor authentication and key agreement for cloud-assisted autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9390–9401, 2020.
- [9] Q. Jiang, X. Zhang, N. Zhang, Y. Tian, X. Ma, and J. Ma, "Three-factor authentication protocol using physical unclonable function for IoV," *Computer Communications*, vol. 173, pp. 45–55, 2021.
- [10] J. Xiong, R. Ma, L. Chen et al., "A personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.
- [11] T. Cheng, G. Liu, Q. Yang, and J. Sun, "Trust assessment in vehicular social network based on three-valued subjective logic," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 652–663, 2019.
- [12] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, "A queuing delay utilization scheme for on-path service aggregation in services-oriented computing networks," *IEEE Access*, vol. 7, pp. 23816–23833, 2019.
- [13] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.
- [14] R. Wang, H. Liu, H. Wang, Q. Yang, and D. Wu, "Distributed security architecture based on blockchain for connected health: architecture, challenges, and approaches," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 30–36, 2019.
- [15] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.
- [16] L. Tan and N. Wang, "Future internet: The Internet of Things," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, China, September 2010.
- [17] T. Wang, H. Ke, X. Zheng, K. Wang, A. K. Sangaiah, and A. Liu, "Big data cleaning based on mobile edge computing in industrial sensor-cloud," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1321–1329, 2020.
- [18] C. Mateos, K. R. Choo, and A. Zunino, "Sharpening the edge: towards improved edge computing environment for mobile and iot applications," *Future Generation Computer Systems*, vol. 107, pp. 1130–1133, 2020.
- [19] F. Li, H. Li, B. Niu, and J. Chen, "Privacy computing: concept, computing framework and future development trends," *IACR Cryptology ePrint Archive*, vol. 2018, p. 1145, 2018.
- [20] Z. Liu, F. Huang, J. Weng et al., "BTMPP: balancing trust management and privacy preservation for emergency message dissemination in vehicular networks," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5386–5407, 2021.
- [21] L. Atzori, D. Carboni, and A. Iera, "Smart things in the social loop: paradigms, technologies, and potentials," *Ad Hoc Networks*, vol. 18, pp. 121–132, 2014.
- [22] R. Bi, Q. Chen, L. Chen, J. Xiong, and D. Wu, "A privacy-preserving personalized service framework through bayesian game in social IoT," *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–13, 2020.
- [23] L. Atzori, A. Iera, and G. Morabito, "IIoT: giving a social structure to the internet of things," *IEEE Communications Letters*, vol. 15, no. 11, pp. 1193–1195, 2011.
- [24] B. Wang, Y. Sun, S. Li, and Q. Cao, "Hierarchical matching with peer effect for low-latency and high-reliable caching in social iot," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1193–1209, 2019.
- [25] A. Narasimman, Q. Wang, F. Li, D. Lee, and B. Luo, "Arcana: enabling private posts on public microblog platforms," in *Proceedings of the 34th IFIP TC 11 International Conference ICT Systems Security and Privacy Protection*, pp. 271–285, Lisbon, Portugal, June 2019.
- [26] M. Mondal, G. S. Yilmaz, N. Hirsch et al., "Moving beyond set-it-and-forget-it privacy settings on social media," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 991–1008, London, UK, November 2019.
- [27] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.
- [28] P. Ilia, I. Polakis, E. Athanasopoulos, F. Maggi, and S. Ioannidis, "Face/off: preventing privacy leakage from photos in social networks," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 781–792, Denver, CO, USA, October 2015.
- [29] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2017.
- [30] X. Zhang, "Reversible data hiding in encrypted image," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 255–258, 2011.
- [31] D. Singh and S. K. Singh, "DWT-SVD and DCT based robust and blind watermarking scheme for copyright protection," *Multimedia Tools and Applications*, vol. 76, no. 11, pp. 13001–13024, 2017.
- [32] A. K. Singh, "Improved hybrid algorithm for robust and imperceptible multiple watermarking using digital images," *Multimedia Tools and Applications*, vol. 76, no. 6, pp. 8881–8900, 2017.
- [33] A. Zear, A. K. Singh, and P. Kumar, "A proposed secure multiple watermarking technique based on DWT, DCT and SVD for application in medicine," *Multimedia Tools and Applications*, vol. 77, no. 4, pp. 4863–4882, 2018.
- [34] T. Deng, X. Li, B. Jin, and J. Xiong, "Who distributes it? privacy-preserving image sharing scheme with illegal distribution detection," in *Proceedings of the SpaCCS 2020 International Workshops Security, Privacy, and Anonymity in Computation, Communication, and Storage*, pp. 257–268, Nanjing, China, December 2020.
- [35] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving CNN feature extraction framework for mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 3, pp. 1441–1455, 2019.
- [36] D. Bogdanov, M. Niitsoo, T. Toft, and J. Willemson, "High-performance secure multi-party computation for data mining applications," *International Journal of Information Security*, vol. 11, no. 6, pp. 403–418, 2012.
- [37] D. Bogdanov, S. Laur, J. Willemson, and Sharemind, "A framework for fast privacy-preserving computations," in *Proceedings of the 13th European Symposium on Research in Computer Security Computer Security-ESORICS 2008*, pp. 192–206, Málaga, Spain, October 2008.

- [38] X. Liao and C. Shu, "Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels," *Journal of Visual Communication and Image Representation*, vol. 28, pp. 21–27, 2015.
- [39] S. Agrawal and M. Kumar, "Mean value based reversible data hiding in encrypted images," *Optik-International Journal for Light and Electron Optics*, vol. 130, pp. 922–934, 2016.
- [40] S. Zheng, D. Li, D. Hu, D. Ye, L. Wang, and J. Wang, "Lossless data hiding algorithm for encrypted images with high capacity," *Multimedia Tools and Applications*, vol. 75, no. 21, pp. 13765–13778, 2016.
- [41] Z. Tang, S. Xu, H. Yao, C. Qin, and X. Zhang, "Reversible data hiding with differential compression in encrypted image," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 9691–9715, 2019.
- [42] M. Li and Y. Li, "Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding," *Signal Processing*, vol. 130, pp. 190–196, 2017.
- [43] V. K. Sharma, D. K. Srivastava, and P. Mathur, "Efficient image steganography using graph signal processing," *IET Image Processing*, vol. 12, no. 6, pp. 1065–1071, 2018.