HF-TPE: High-Fidelity Thumbnail-Preserving Encryption

Yushu Zhang, *Member, IEEE*, Ruoyu Zhao, Xiangli Xiao, Rushi Lan, *Member, IEEE*, Zhe Liu, *Senior Member, IEEE*, and Xinpeng Zhang, *Member, IEEE*

***Abstract*— With the popularity of cloud storage services, people are increasingly accustomed to storing images in the cloud. However, cloud storage services raise privacy concerns, e.g., leakage of images to unauthorized third parties and service providers may exploit image detection technologies to portrait users without permission. Although privacy concerns can be solved by encrypting images before they are uploaded to the cloud, traditional encryption methods significantly affect the usability and user experience, for example, users cannot preview images in the cloud. Recently, Marohn *et al.* proposed two approximate thumbnail-preserving encryption schemes, called DRPE and TPE-LSB, to balance the privacy and usability of images in the cloud. However, both schemes have defects that either the decryption may fail or the ciphertext images have poor performance in perceived quality and too many noise points after decryption. To this end, we pertinently propose a high-fidelity thumbnail-preserving encryption scheme (HF-TPE). Compared with the previous works, on the one hand, the HF-TPE scheme not only ensures the correct decryption of ciphertext images, but also makes the ciphertext thumbnails more close to the plaintext images perceptually. On the other hand, the decrypted thumbnails have lower noise intensity and upper limit of the number of noises. In addition, simulation experiments further show that the HF-TPE scheme can guarantee users' usability.**

***Index Terms*— Balancing privacy and usability, image encryption, thumbnail, cloud storage.**

I. Introduction

NOWADAYS, everyone with a phone is almost equivalent to having a camera to carry around as the proliferation of smartphones equipped with multi-camera. Besides, people have plenty of other image-capturing devices such as drones, tablets, pinhole cameras, and webcams. People can take high-resolution images with these devices anytime and anywhere. It is estimated that about 1.4 trillion images were taken in 2020 and the growth is quite rapidly each successive year [1]. These images often record people's daily life with family and friends. Therefore, images tend to contain a great number of sensitive information like identity, location, and health [2], [3]. A survey reveals that all of the 112 interviews believed that their images contain various types of personal privacy [4]. As a result, people tend to prevent others from viewing their personal albums, which is easy to achieve on the local devices.

However, cloud storage services are becoming increasingly popular with the rapid development of mobile network. There is an increasing number of people preferring to store their personal images in the cloud for convenience, e.g., iCloud, Google Drive, OneDrive, and Dropbox. Meanwhile, these cloud services automatically sync images from local devices. Storing images in the cloud allows users to preview, organize, and manage them more easily. For instance, users no longer worry about accidental damage to their images and can browse and download images through some devices connected to the Internet. However, images are beyond the control of users. A little thought shows that it is undoubted to increase the difficulty of users to protect image privacy. Data breaches in the cloud are becoming more and more frequent and the risks are also multifaceted. Hundreds of Hollywood stars had their private images on the iCloud leaked by hackers in 2014 [5], which is one of the biggest cloud privacy breach even to date. Meanwhile, cloud service providers themselves have many reasons to steal users' images. Facebook was exposed to exploit users' images to train its own AI models for automatic image recognition in 2019 [6], not the first time it has been exposed stealing users' images [7].

Manuscript received January 21, 2021; revised March 3, 2021; accepted March 29, 2021. Date of publication April 1, 2021; date of current version March 9, 2022. This work was supported in part by the Natural Science Foundation of China under Grant 62072237 and Grant U1936214, in part by the Guangxi Key Laboratory of Multi-Source Information Mining and Security through the Research Fund under Grant MIMS20-02, in part by the Guangxi Key Laboratory of Trusted Software under Grant KX202027, in part by the Foundation Research Project of Jiangsu Province through the Natural Science Fund under Grant BK20201290, and in part by the Graduate Innovation Base (Laboratory) of the Nanjing University of Aeronautics and Astronautics through the Open Fund Project under Grant kfjj20201601. This article was recommended by Associate Editor J.-M. Guo. *(Corresponding author: Zhe Liu.)*

Yushu Zhang is with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, also with the Guangxi Key Laboratory of Multi-Source Information Mining and Security, Guangxi Normal University, Guilin 541004, China, and also with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: yushu@nuaa.edu.cn).

Ruoyu Zhao, Xiangli Xiao, and Zhe Liu are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China (e-mail: zhaoruoyu@nuaa.edu.cn; xianglisch@163.com; zhe.liu@nuaa.edu.cn).

Rushi Lan is with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: rslan2016@163.com).

Xinpeng Zhang is with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: zhangxinpeng@fudan.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2021.3070348.

Digital Object Identifier 10.1109/TCSVT.2021.3070348

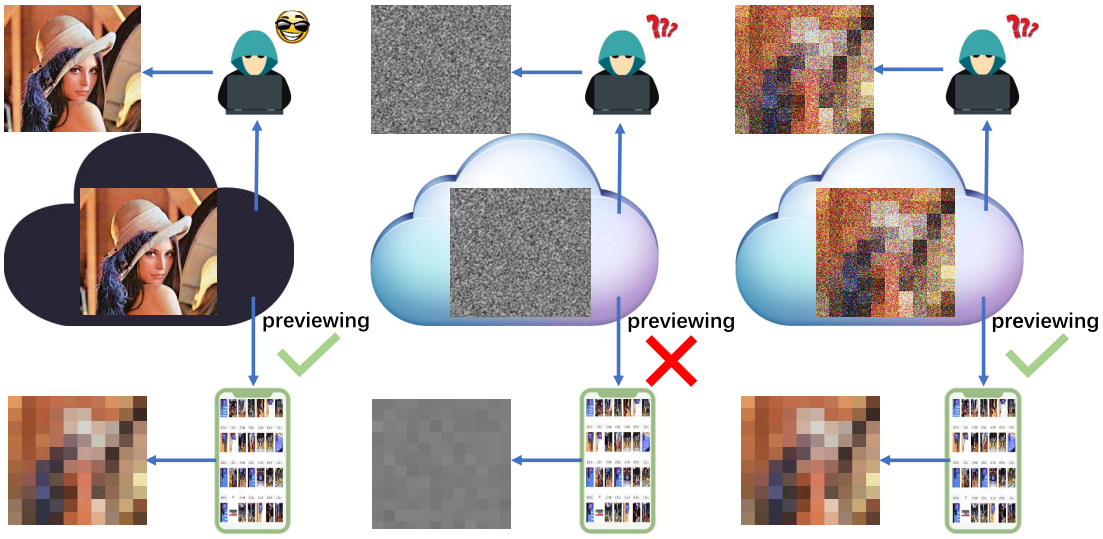


Fig. 1. Left: When original images are uploaded to the cloud, legitimate users can preview them and illegal third parties can learn the image privacy. Middle: When images are uploaded to the cloud after traditional encryption, legitimate users cannot preview them and illegal third parties cannot learn the image privacy. Right: When images are uploaded to the cloud after thumbnail-preserving encryption, legitimate users can preview them, but illegal third parties cannot learn the image privacy.

Traditional image encryption algorithm with the architecture of confusion-diffusion [8]–[10] is a common approach to protect the image privacy. While this approach is effective, there are still many problems after uploading encrypted images into the cloud. For example, users are not able to directly browse these ciphertexts over the cloud and thus they are forbidden to organize and manage images through visual content. Meanwhile, users can only understand the content of the images after downloading and decrypting the encrypted images. In other words, the traditional encryption disables the content-based image usability in the cloud. In order to have the success with users, the protection mechanism of image privacy cannot be at the expense of compromising the usability of cloud services.

Some research results on image degradation (i.e., the low resolution version or the poor visual effect version of the original image) and visual memory in visual psychology make it possible for ciphertext images to simultaneously preserve privacy and usability. Gregory's research [11] shows that subjects have the ability to identify degraded versions of previously browsed images, which is particularly prominent in portrait images. Moreover, Snodgrass's research [12] further shows that the above ability is stronger if images are made by the subjects themselves (i.e., shooting or drawing). The information that subjects learn by browsing original images is called **prior knowledge**. Research conducted by Denning *et al.* [13] indicates that the subjects are able to recognize the degraded versions of images based on their prior knowledge. In conclusion, the above researches suggest that image recognition is not only determined by people's visual input, but also strongly affected by the prior knowledge. In other words, image recognition is a kind of reasoning process based on the mixture of human visual input and prior knowledge, which has been proved by Peter *et al.* in neurology [14]. In addition, experiment performed by Rousselet *et al.* [15] shows that the human visual system can process four images in parallel and can remember the visual information (i.e., prior knowledge) of images only by browsing them briefly.

Recently, a novel image encryption concept based on the prior knowledge is proposed by Wright *et al.* [16], i.e., thumbnail-preserving encryption (TPE). The idea is that ciphertext images preserve degraded versions of original images, i.e., thumbnails. The images encrypted by TPE preserve the features larger than the thumbnail block, but erase the features smaller than the thumbnail block, which usually contain many sensitive details. Specifically, legitimate users with prior knowledge can combine the visual information of the encrypted image with prior knowledge to infer the specific content of the image, while illegal parties without prior knowledge cannot infer from the rough visual information in the encrypted image. Therefore, as shown in Fig. 1, legitimate users can acquire usability from the TPE-encrypted images, while illegal third parties cannot learn vital personal privacy information.

The first TPE scheme was designed by Wright *et al*. [16]. In the ciphertext images generated by this scheme, the exact thumbnails are preserved based on a permutation-only operation. Therefore, ciphertext images not only display the thumbnails, but also disclose every pixel value of original images. Many studies [17]–[20] have shown that permutation-only encryption cannot resist statistical attacks. Subsequently, Marohn *et al.* [21] proposed two approximate TPE schemes, namely DRPE and TPE-LSB. The two schemes are claimed to be secure by the authors and the information leaked by the encrypted image is described accurately. Nevertheless, they also have disadvantages. First, the thumbnails of ciphertext images generated by these two schemes are of poor quality, which is quite different from the original images' thumbnails. Second, ciphertext images reveal more than just thumbnails.

Third, the ciphertext images generated by the DRPE scheme have the probability of decryption failure and the decrypted images generated by the TPE-LSB scheme have too many noise points (the pixel value in decrypted images is different from corresponding pixel value in original images).

In this paper, we propose a high-fidelity TPE scheme (HF-TPE), in which ciphertext images have more similar thumbnails to the original ones and decrypted images have less noise. This scheme is divided into two basic operations: substitution and permutation. We exploit rank-then-encipher (RtE) [22] to construct a sum-preserving data embedding (SPDE) scheme. In this embedding scheme, each vector has a unique serial number (SN). The SN and the vector form a bijective relationship. After encrypting the original SN, the SPDE exploits the bijective relationship to convert the encrypted SN into a corresponding vector (i.e, data embedding), which is the substitution encryption part of the HF-TPE. Then the pixels are permutated in each thumbnail block. Note that noise may occur during the embedding process as the length of the SN may has one bit more than the embedded capacity in the vector.

In theory, the upper limit of pixel noise points in the HF-TPE scheme is 50% of the total pixel points in the image, but in practice it will be much lower than the upper limit. According to the experiments, there are only about 10% to 20% noise points. Moreover, the pixel value of each noise is only increased or decreased by one pixel value compared with the original value. Similarly, the pixel in the ciphertext thumbnail is increased or decreased by at most one pixel value compared with the pixel in the original thumbnail. This means that compared with the previous works, ciphertext thumbnails in the HF-TPE scheme not only have higher perception quality, but decrypted images also have less noise. Meanwhile, the images encrypted by the scheme can be successfully decrypted. In addition, user evaluations demonstrate that users are still able to browse and distinguish ciphertext images even when detection algorithms cannot detect the faces in ciphertext images.

The major contributions of this work can be highlighted as follows:

- We propose a new TPE scheme, i,e., HF-TPE, which has the advantages of no failure in decryption, less noise in the decrypted image, and well perceptual quality in the ciphertext compared with Marohn *et al.*' TPE schemes.
- We propose a sum-preserving data embedding scheme (SPDE), which can embed data without changing the sum of the vector and is used to construct the HF-TPE scheme. On the one hand, in order to solve the problem of possible overflow in pixel group embedding, the pixel group is divided into two parts. One part exploits the SPDE scheme to embed the pixel group data under the condition of preserving the sum unchanged and other part is applied to save the overflow data. On the other hand, the embedding state is divided into three parts to reduce the noise introduced in the embedding.
- We have done many experiment evaluations to show the superiority of the HF-TPE scheme. Objective experiments present that the ciphertext image has more similar thumbnail, higher perceptual quality, and less noise compared with the scheme proposed by Marohn *et al.*. Subjective user experiment displays that users can still distinguish and recognize ciphertext images through rough visual information.

The rest of this paper is organized in the following manner. Section II briefly introduces potential solutions. Section III presents threat model, goals, and assumptions. Section IV is preliminaries. Section V provides the concrete construction of the SPDE. Section VI exploits the SPDE to construct the HF-TPE scheme. Experiments with evaluation and comparison are given in Section VII and followed by conclusion in Section VIII.

II. Potential Solutions

Many scholars have been working on how to balance the privacy and usability of images and have put forward many potential solutions. These potential solutions fall into two main categories. a) Privacy is protected based on traditional encryption schemes and some auxiliary information is used to improve the usability of the ciphertext image. b) The image degradation version is combined with prior knowledge to ease the contradiction between privacy and usability.

A. Potential Solutions to Improve the Usability of Traditional Encryption

In recent years, searchable encryption as a potential solution to ease the privacy and usability of data in the cloud has received widespread research in various application scenarios (e.g., [23]–[25]). This kind of solution constructs the index based on the keywords of the data file and then encrypts the original data and its index. Legitimate users are able to submit the keywords based search requests to the cloud to obtain relevant data. Searchable encryption has been demonstrated to be useful in the text file, but its application in image files is facing great challenges. Specifically, the keywords needed for searchable encryption directly exist in the text file, but not in the image file.

The technology of extracting keywords from images to annotate them is also considered as a potential solution to lighten the contradiction between privacy and usability [26]. The idea of this technology is to summarize the significant information of the original image into keywords and then the image is encrypted. On the one hand, image privacy can be well protected as the characteristics of traditional image encryption. On the other hand, legitimate users are able to learn the content of the image through the keywords on annotations. Nevertheless, there exist serious deficiencies in this technology. Specifically, the image is a carrier containing rich information, but the annotation is only a few text words, which leads to the neglect of much image information. With the rise of artificial intelligence, a large number of excellent image annotation techniques have been proposed [27]–[29] to extract keywords as accurately as possible. However, the keywords that can distinguish different images tend to be subjective. Therefore, the keywords extracted by objective technologies may not be useful for users to recognize images. In addition,

it is an unrealistic option to require users to manually annotate from the perspective of the user experience.

Image retrieval is also a potential solution compatible with image privacy and usability [30], [31]. The idea of this solution is to exploit image feature extraction algorithm or annotation algorithm to extract image features or keywords, which together with images are encrypted and uploaded to cloud services. Legitimate users can retrieve according to image features or keywords to obtain desired images and thus acquire certain the usability when the privacy is ensured. However, there exist some limitations. First, legitimate users require to input keywords or features of the intended images during the retrieval. In many cases, they do not know which images are desired until having browsed the visual content in their albums. Second, the current mainstream cloud services do not support the storage of additional auxiliary information, which means that the image retrieval cannot be directly deployed to the existing cloud. Third, legitimate users are still browsing snowflake ciphertext images, which may be unnatural from the perspective of visual perception.

B. Potential Solutions Based on Visual Information and Prior Knowledge

Image filtering is a common solution to balance the privacy and usability of images [32]–[34]. The solution makes the images distorted by image filters like blurring or pixelation and then make them unrecognizable. In the blur filter, the Gaussian function is exploited to modify the adjacent pixels to remove the details in original images. Pixelation is to divide the image into the same dimension blocks, calculate the average value of each pixel block, and assign the average to all pixels in this block. The solution is widely used in daily life, especially in social networks, benefit by low complexity and well edge preservation properties. However, the solution is not suitable for cloud storage environment since the generated images are irreversible. Recently, Sun *et al*. [35] proposed a completely reversible Mosaic encryption scheme that the image can be protected by Mosaic and restored without loss. However, this solution is not secure against statistical attacks since it only permutates the pixel in the image [17]–[20].

Another solution to balance image privacy protection and usability is region of interest (ROI) encryption (e.g., [36]–[39]). This solution divides the image into the secret part and the public part, where the secret part containing sensitive information is encrypted and the public part only contains insensitive information. Anyone can get access to the public part, but only the authorized person with the key can acquire to the secret one. People have the ability to obtain certain usability by learning the insensitive information of the public part. However, the determination of image privacy region is very subjective. Different people browsing the same image may determine completely divergent privacy regions and the determination of privacy region for the same object in different scenarios is also dissimilar. Therefore, although the technology of exploiting computer vision to delimit the privacy area becomes sophisticated [40], [41], the use of objective technology can not accurately delineate the subjective sensitive area. In addition, it is an unrealistic choice for users to perform manual operations [42].

III. Threat Model and Goals

In this paper, we focus on two specific privacy threats to images resulting from cloud services. The first threat comes from hackers (e.g., [5]), who manage to illegally acquire users' images by attacking the cloud services or by other means. The second threat is from cloud service providers. On the one hand, the malicious insiders of the cloud services usually steal users' image privacy information for financial gain. On the other hand, the providers may illegally exploit users' images for the purpose of AI model training (e.g., [6], [7]).

The goal of this work is to design an image encryption scheme to make the images in the cloud possess both usability and privacy. Usability means that legitimate users can preview images in the cloud through visual information of ciphertext images and then recognize, organize, and manage images [43] combined with prior knowledge. Privacy refers to the fact that the exact information in the image cannot be learned by illegal third parties through ciphertext images. For example, one can get knowledge of the rough information from the ciphertext image such as facial contour, posture, and background environment after encrypting a portrait, while more information, e.g., which can be used to directly determine somebody, should not be exposed. Illegal parties without prior knowledge cannot know who is in the image from the ciphertext image, while the legitimate users with the prior knowledge can do.

The above statements imply several constraints of this work, which make it considerably challenging. First, the proposed scheme must be compatible with existing cloud services without causing any changes in the aspects of software, backend, framework, and usage pattern, which is crucial to the success deployment of this scheme. Second, the usability of this scheme should come from the content information of ciphertext images themselves rather than the additional information (i.e., keywords). Finally, the information exposed in ciphertext images should not be endanger privacy. For the sake of illustration, all local components on end user devices (e.g., phones, computers, and tablets) are assumed to be completely trusted, including the system, software, and hardware.

IV. Preliminaries

In this section, some definitions that help to construct the scheme are introduced.

Definition 1: Let Enc_K be an encryption algorithm with key K, nonce T, plaintext M as input, and ciphertext C as output. Let Dec_K be a decryption algorithm with key K, nonce T, ciphertext C as input, and plaintext M as output. $\mathcal{M}$ represents the message space and Φ represents a function that outputs the property we wish to preserve in the $\mathcal{M}$. An encryption scheme is **Φ-preserving** (over $\mathcal{M}$) if it satisfies the following properties for any K, T, and M:

$$\begin{aligned} \mathrm{Enc}_K(T, M) &= C \\ \mathrm{Dec}_K(T, C) &= M \\ C &\in \mathcal{M} \\ \Phi(C) &= \Phi(M), \end{aligned} \tag{1}$$

where Φ can get the same results from ciphertext and plaintext, which means that the ciphertext contains properties we want to preserve in the encryption.

The above definition is equivalent to the format-preserving encryption (FPE) [22] by considering $\{C \mid \Phi(C) = \Phi(M)\}$ as the "slices" of the message space.

The notion of security for FPE has been defined by Bellare *et al*. [22].

Definition 2: Let $\mathcal{F}_\Phi$ be the set of functions F: $\{0,1\}^* \times \mathcal{M} \rightarrow \mathcal{M}$, which are "$\Phi$-preserving" by considering $\Phi(\mathrm{F}(T, M)) = \Phi(M)$, for all T and M. The FPE scheme is **Pseudo Random Permutation (PRP) security** if, for all **Probabilistic Polynomial Time (PPT)** oracle machines $\mathcal{A}$ [22],

$$\left| \Pr_{K \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}^{\mathrm{Enc}_K(\cdot,\cdot)}(\lambda) = 1 \right] - \Pr_{\mathrm{F} \leftarrow \mathcal{F}_\Phi} \left[\mathcal{A}^{\mathrm{F}(\cdot,\cdot)}(\lambda) = 1 \right] \right|, \quad (2)$$

is negligible in λ.

Definition 3: $\mathcal{A}$ is **Nonce-Respecting (NR)** if it never makes two oracle calls with the same first argument (i.e., nonce T) [43].

The scheme is called **NR security** if it satisfies *Definition 2* but only with respect to the NR distinguisher. Tajik *et al*. proved that **NR security** is sufficient in the case of TPE, which gives the same guarantee as the PRP security when the same T is not called a second time [43]. In other words, images encrypted by the TPE scheme are indistinguishable from randomly chosen images with the same thumbnail. In fact, each image has the unique identifier (e.g., name) that can be as a nonce T.

An image thumbnail is generated by first selecting the block size B, dividing the original image into $B \times B$ blocks, and then calculating the average value of pixels in each block. The goal of TPE is that the information revealed by the ciphertext image is only the average value of the corresponding pixels in each $B \times B$ block.

Definition 4: Let $\mathcal{M} = (\mathbb{Z}_{d+1})^n$ be the message space. Every message has n individual elements, which are non-negative and have values of no more than d. Let $\Phi_{sum}(\vec{v})$ be the sum of $\vec{v}$. An encryption scheme is called sum-preserving encryption (SPE), if it satisfies the following properties:

$$\begin{aligned} \vec{v} &\in \mathcal{M} \\ \vec{v} &= (v_1, \ldots, v_n) \\ s &= \sum_{i=1}^{n} v_i \\ \mathrm{Enc}_K(T, \vec{v}) &= C \\ C &\in \left\{ \vec{x} \mid \vec{x} = (x_1, \ldots, x_n) \in \mathcal{M},\ s = \sum_{i=1}^{n} x_i \right\} \\ \Phi_{sum}(C) &= \Phi_{sum}(\vec{v}) = s. \end{aligned} \quad (3)$$

Definition 5: Let $\mathcal{M} = (\mathbb{Z}_{d+1})^n$ be the message space, $\vec{x} = (x_1, \ldots, x_n) \in \mathcal{M}$ be the vector, and $s = \sum_{i=1}^{n} x_i$. Let $\Phi_{sum}^{-1}(s)$ be a set of vectors that sum to s in the $\mathcal{M}$, which can be represented as follows:

$$\Phi_{sum}^{-1}(s) = \left\{ \vec{x} \mid \vec{x} = (x_1, \ldots, x_n) \in \mathcal{M}, s = \sum_{i=1}^{n} x_i \right\}. \quad (4)$$

Although the construction of the SPE is a considerately challenging work, the RtE [22] is a good solution. The basic idea of encrypting a vector $\vec{v}$ is in the following. First, the sum s of the original vector $\vec{v}$ is calculated and the set of vectors under the same sum s is determined, i.e., $\Phi_{sum}^{-1}(s)$. Second, the number of vectors in $\Phi_{sum}^{-1}(s)$ is set to $\mathrm{t}(s)$ in which each vector is numbered. Each number is unique and within the range of $\mathbb{Z}_{\mathrm{t}(s)}$, called SN. Third, the SN of the $\vec{v}$ is obtained through the $\mathrm{rank}_s(\vec{v})$ function and then is encrypted within the range of $\mathbb{Z}_{\mathrm{t}(s)}$. Finally, the encrypted SN can be converted into a vector (belonging to $\Phi_{sum}^{-1}(s)$) via $\mathrm{rank}_s^{-1}(\cdot)$. Details in the encryption of the RtE can be found in **Algorithm 1** and the decryption algorithm is reverse.

Algorithm 1 Rank-Then-Encipher

Input: T, K, and $\vec{v} = (v_1, \ldots, v_n) \in \mathcal{M}$
Output: $C = (y_1, \ldots, y_n) \in \mathcal{M}$ /* $\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} v_i$ */
$s = \sum_{i=1}^{n} v_i$
$\Phi_{sum}^{-1}(s) = \left\{ \vec{x} \mid \vec{x} = (x_1, \ldots, x_n) \in \mathcal{M}, s = \sum_{i=1}^{n} x_i \right\}$
$\mathrm{t}(s)$ = the number of the vector of $\Phi_{sum}^{-1}(s)$
$\mathcal{N} = \emptyset$
for $i = 0 : t(s) - 1$ **do**
    $\vec{z} \in (\Phi_{sum}^{-1}(s) - \mathcal{N})$
    Set the SN of $\vec{z}$ to be i
    $\mathcal{N} = \mathcal{N} \cup \vec{z}$
end for
$num_i = \mathrm{rank}_s(\vec{v})$ /* the SN of $\vec{v}$ */
num_j = encrypt num_i by a pseudorandom pad, $num_j \in \{0, \ldots, t(s) - 1\}$
$C = \mathrm{rank}_s^{-1}(num_j)$ /* the enciphered SN num_j corresponds to vector in set $\Phi_{sum}^{-1}(s)$ */
return C

However, the computational complexity of $\mathrm{rank}_s/\mathrm{rank}_s^{-1}$ is up to $\mathcal{O}(d^3n^3)$ [22], which makes the scheme unrealistic for real applications. This problem can be simply and practicably solved by setting $n = 2$ [43]. In addition, pseudo-random functions (PRFs) are exploited to generate keys. The standard definition of RPF is given in [44]. For a function $F : \mathcal{K} \times \mathcal{X} \rightarrow \mathcal{Y}$, F is a secure PRF if no efficient adversary can distinguish the output of $F(k, .)$ (k is randomly chosen from $\mathcal{K}$) from that of a truly random function $f(.)$ from $\mathcal{X} \rightarrow \mathcal{Y}$.

V. Sum-Preserving Data Embedding

In this section, the RtE algorithm is exploited to design a data embedding scheme under the premise of preserving the vector's sum invariance, which is called sum-preserving data embedding (SPDE).

A. Overview

From the previous section, all vectors with the same sum can be enumerated when the sum s of the vector is known,

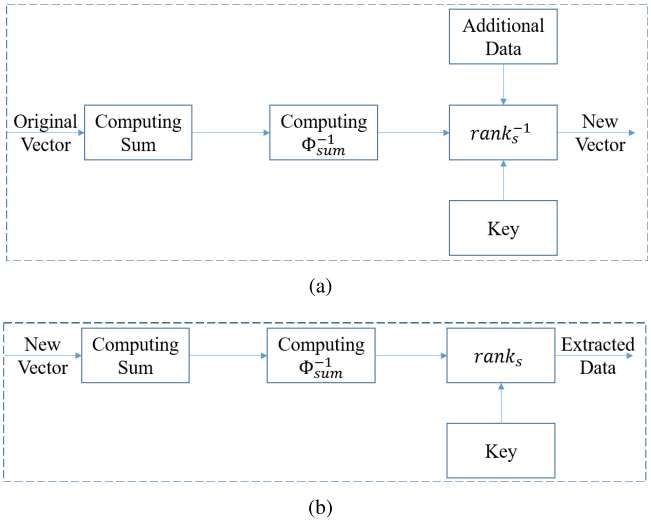


Fig. 2. The framework of SPDE. (a) The framework of data embedding. (b) The framework of data extraction.

i.e., $\Phi_{sum}^{-1}(s)$. The number of vectors in $\Phi_{sum}^{-1}(s)$ can be calculated by *Lemma 1*.

Lemma 1: Let $t(s)$ be the number of vectors in $\Phi_{sum}^{-1}(s)$, which can be calculated as follows:

$$\mathrm{t}(s) = \begin{cases} s+1 & \text{if } s \le d \\ 2 \times d - s + 1 & \text{otherwise.} \end{cases} \tag{5}$$

Proof 1: For convenience, we assume that a is an independent variable and b is a dependent variable (i.e., $b = s - a$). a and b can only be integers between 0 and d. When $s \le d$, a can be any integer between 0 and s and thus the total number of possible values is $s + 1$; Otherwise, a can be any integer between $s - d$ and d and thus the total number of possible values is $2 \times d - s + 1$. ■

The SNs of vectors in $\Phi_{sum}^{-1}(s)$ need to be determined before using the rank_s or rank_s^{-1} function. There are **bijective relationships** between the SNs and vectors, where $SN \in \{0, \ldots, \mathrm{t}(s) - 1\}$. The SN order of vectors in $\Phi_{sum}^{-1}(s)$ is based on the predetermined order, which is serves as a security key.

rank_s marked as a function that converts the vector in $\Phi_{sum}^{-1}(s)$ to corresponding SN and rank_s^{-1} is a function that converts the SN to corresponding vector. The functions rank_s and rank_s^{-1} are:

$$\begin{aligned} \mathrm{rank}_s(a, b) &= \text{the SN of vector } (a, b) \\ \mathrm{rank}_s^{-1}(SN) &= \text{the SN corresponding vector,} \end{aligned} \tag{6}$$

where $SN \in \{0, \ldots, \mathrm{t}(s) - 1\}$. On the one hand, rank_s^{-1} can convert any integer belonging to $\{0, \ldots, \mathrm{t}(s) - 1\}$ into a vector with the same sum s (i.e., in $\Phi_{sum}^{-1}(s)$), which is called **data embedding**. On the other hand, $\mathrm{rank}_s(a, b)$ can convert any vector into a corresponding integer (i.e., the SN), which is called **data extraction**. The frameworks of data embedding and data extraction are shown in Fig. 2.

Both the embedded data and the extracted data should be binary. However, the data extracted by rank_s are decimal, which need to be converted to binary. Similarly, the data embedded by rank_s^{-1} are decimal, i.e., the binary data to be embedded data need to be converted to decimal. Therefore, for the convenience of expression, this paper sets up two functions, $\mathrm{bin2dec}(bin)$ and $\mathrm{dec2bin}(dec)$. The function $\mathrm{bin2dec}(bin)$ represents converting a bitstream bin to corresponding decimal number, e.g., $\mathrm{bin2dec}(11_2) = 3$, $\mathrm{bin2dec}(100_2) = 4$. Similarly, the function $\mathrm{dec2bin}(dec)$ represents converting a decimal number dec to a corresponding bitstream, e.g., $\mathrm{dec2bin}(3) = 11_2$, $\mathrm{dec2bin}(4) = 100_2$. In addition, the SN is represented in binary:

$$\begin{aligned} r &= \mathrm{rank}_s(a, b) \\ SN &= \mathrm{dec2bin}(r). \end{aligned} \tag{7}$$

B. Data Embedding

Before embedding data into the original vector, the data capacity of each vector needs be determined. First, the sum s of the original vector is calculated. According to s, the number of the vector in $\Phi_{sum}^{-1}(s)$ is obtained, i.e., $\mathrm{t}(s)$. Then the data capacity of a vector determined by sum can be calculated.

Lemma 2: Let $\mathrm{Cap}_s(s)$ denote the data capacity of a vector determined by sum s calculated as:

$$\mathrm{Cap}_s(s) = \lfloor \log_2(\mathrm{t}(s)) \rfloor, \tag{8}$$

where $\lfloor . \rfloor$ represents round down.

Proof 2: Let $\mathbf{X}$ be an n-bit bitstream, where $X_i \in \{0, 1\}$, $n \in \mathbb{N}^+$, where $\mathbb{N}^+$ is the set of positive integers. When $1 \le n \le \lfloor \log_2(\mathrm{t}(s)) \rfloor$, $\mathrm{bin2dec}(\mathbf{X}) \in \{0, \ldots, \mathrm{t}(s) - 1\}$ for any n-bit bitstream $\mathbf{X}$; Otherwise, there must be at least an n-bit bitsream $\mathbf{X}$ making $\mathrm{bin2dec}(\mathbf{X}) \notin \{0, \ldots, \mathrm{t}(s) - 1\}$, which causes rank_s^{-1} to fail. ■

Therefore, after acquiring the original vector and computing the sum s, there must be $\mathrm{Cap}_s(s)$-bit data capacity in the vector. However, in some cases, the $\mathrm{Cap}_s(s)$-bit bitstream does not fully utilize the bijective relationships between SN and vectors in $\Phi_{sum}^{-1}(s)$.

Lemma 3: Let $\mathrm{Dom}(s)$ be the length of the SN of vectors in the $\Phi_{sum}^{-1}(s)$, it can be calculated as follows:

$$\mathrm{Dom}(s) = \lceil \log_2(t(s)) \rceil, \tag{9}$$

where $\lceil . \rceil$ represents round up.

Proof 3: Let $\mathbf{X}$ be an n-bit bitstream, where $X_i \in \{0, 1\}$, $n \in \mathbb{N}^+$. When $1 \le n \le \lceil \log_2(\mathrm{t}(s)) \rceil - 1$, there is at least one SN of the vector in $\Phi_{sum}^{-1}(s)$ that cannot be represented by $\mathbf{X}$; Otherwise, the SNs of all vectors in $\Phi_{sum}^{-1}(s)$ are represented by $\mathbf{X}$. ■

Lemma 4: Let $\mathrm{Dom}(s) - \mathrm{Cap}_s(s)$ be called $\mathrm{red}(s)$. It can be expressed as:

$$\mathrm{red}(s) = \begin{cases} 0 & \mathrm{t}(s) \text{ is the } n\text{-th power of } 2, n \in \mathbb{N} \\ 1 & \text{otherwise.} \end{cases} \tag{10}$$

Proof 4: When $\mathrm{t}(s) = 2^n$, $\mathrm{Cap}_s(s) = n$, $\mathrm{Dom}(s) = n$, as a result, $\mathrm{Dom}(s) - \mathrm{Cap}_s(s) = 0$; Otherwise, when $2^{n-1} < \mathrm{t}(s) < 2^n$, $\mathrm{Cap}_s(s) = n - 1$, $\mathrm{Dom}(s) = n$, and thus $\mathrm{Dom}(s) - \mathrm{Cap}_s(s) = 1$. ■

As mentioned above, the SPDE leverages the bijective relationships between SN and vectors in $\Phi_{sum}^{-1}(s)$. When $\mathrm{t}(s)$ is the n-th power of 2, all bijections between them are

Algorithm 2 Data-Embedding Algorithm

Input: Original vector: $\vec{v} = (a, b)$
Additional data (binary): $\beta = b_1 b_2 b_3 \ldots$,
Output: New vector: $C = (a', b')$ /* $a' + b' = a + b$ */
$s = a + b$
$n = \text{Cap}_s(s)$
$\mathbf{Y} = b_1 \ldots b_n$ /* $embed_1$ */
if $embedded\ state \notin S_1$ **then**
  $dom = \text{Dom}(s)$
  $\mathbf{X} = \text{dec2bin}(\text{t}(s) - 1) = X_{dom} \ldots X_1$
  $\mathbf{X'} = X_n \ldots X_1$ /* $n = dom - 1$ */
  if $\text{bin2dec}(\mathbf{Y}) \leq \text{bin2dec}(\mathbf{X'})$ **then**
    $R = b_{dom}$
    $\mathbf{Y} = Rb_1 \ldots b_n$ /* $embed_2$ */
  end if
end if
$r = \text{bin2dec}(\mathbf{Y})$
$C = \text{rank}_s^{-1}(r) = (a', b')$
return C

Algorithm 3 Data-Extraction Algorithm

Input: New vector: $C = (a', b')$
Output: Extracted data (binary): β
$s = a' + b'$
$dom = \text{Dom}(s)$
$r = \text{rank}_s(a', b')$
$SN = \text{dec2bin}(r) = Y_{dom} \ldots Y_1$ /* $Y_i \in \{0, 1\}$ */
if $embedded\ state \in S_1$ **then**
  $\beta = SN$
else
  $n = \text{Cap}_s(s)$
  $\mathbf{Y} = Y_n \ldots Y_1$ /* $embed_1$ */
  $\mathbf{X} = \text{dec2bin}(\text{t}(s) - 1) = X_{dom} \ldots X_1$ /*$X_i \in \{0, 1\}$*/
  $\mathbf{X'} = X_n \ldots X_1$
  if $\text{bin2dec}(\mathbf{Y}) \leq \text{bin2dec}(\mathbf{X'})$ **then**
    $R = Y_{dom}$
    $\beta = Y_n \ldots Y_1 R$ /* $embed_2$ */
  else
    $\beta = Y_n \ldots Y_1$
  end if
end if
return β

exploited. Otherwise, some bijections between them will never be exploited, which makes $\text{red}(s) = 1$, meaning that SN requires one more bit of binary than the embedding capacity determined by s. As shown in *proof 2*, adding one bit may cause rank_s^{-1} to fail. For example, let $\mathbf{X} = X_3 X_2 X_1$ be an *3*-bit bitstream, $s = 5$, $d = 255$. When $\mathbf{X} = 111_2$, i.e., $r = \text{bin2dec}(\mathbf{X}) = 7$, $\text{rank}_s^{-1}(r)$ cannot correctly calculate the vector.

However, assuming that the maximum value of SN in $\Phi_{sum}^{-1}(s)$, i.e., $\text{t}(s) - 1$, is represented by $\mathbf{X} = X_{dom} \ldots X_1 = \text{dec2bin}(\text{t}(s) - 1)$, where $dom = \text{Dom}(s)$. Under the condition of $\text{red}(s) = 1$, the data capacity of vectors determined by s is $cap = dom - 1$, where $cap = \text{Cap}_s(s)$. If $cap > 0$, one can take out cap bits from the bitstream to be embedded, which are marked as $\mathbf{Y} = Y_{cap} \ldots Y_1$. Meanwhile, the cap-bit LSB of $\mathbf{X}$ is intercepted, called $\mathbf{X'} = X_{cap} \ldots X_1$. If $\text{bin2dec}(\mathbf{Y}) \leq \text{bin2dec}(\mathbf{X'})$, then one bit R is taken from the bitstream to be embedded and meanwhile R is appended to the highest bit of $\mathbf{Y}$, i.e., $\mathbf{Y} = RY_{cap} \ldots Y_1$. $\text{rank}_s^{-1}(r)$ is still in $\Phi_{sum}^{-1}(s)$ for any $R \in \{0, 1\}$, where $r = \text{bin2dec}(\mathbf{Y})$.

The embedding directly determined by sum s is called $embed_1$. In addition, if one can continue to embed one bit, the embedding is called $embed_2$. For ease of exposition, we list three disjoint sets of the $embedded\ state$, S_1, S_2, and S_3:

$$\begin{aligned} S_1 &: t(s) = 2^n, \quad n \in \mathbb{N} \\ S_2 &: t(s) \neq 2^n, \quad \text{can do } embed_1 \text{ and } embed_2 \\ S_3 &: t(s) \neq 2^n, \quad \text{only do } embed_1. \end{aligned} \tag{11}$$

The data capacity of S_1 and S_2 is $\lceil log_2(t(s)) \rceil$ and S_3 is $\lfloor log_2(t(s)) \rfloor$, while every vector must belong to and only belong to one of three sets. The data-embedding algorithm is shown in **Algorithm 2**.

C. Data-Extraction

In this section, we want to extract the data $\beta = b_1 b_2 b_3 \ldots$ embedded in the vector $C = (a', b')$. The SN of the vector can be obtained by $\text{rank}_s(a', b')$. If the vector belongs to set $S1$, the SN is the embedded data; Otherwise, the low $\lfloor \log_2(\text{t}(s)) \rfloor$ bit of SN is extracted, i.e., $\mathbf{Y}$, and it is judged whether $\mathbf{Y}$ meets the conditions of S_2. If the conditions hold, the highest bit of the SN is appended to the end of $\mathbf{Y}$, which is the embedded data; Otherwise, only $\mathbf{Y}$ is the embedded data. The details of data-extraction are shown in **Algorithm 3**.

VI. HF-TPE Construction

A. Overview

We consider the following types of images:

- Each image has a unique set of public metadata, e.g., filename and timestamp, which can be regarded as a unique *nonce*, let it be T.
- Each image consists of one or more $channel(s)$ (e.g., grayscale, RGB, YUV, and HSV).
- Each channel is a two-dimensional array of pixels, whose values/intensities are taken from $\{0, \ldots, d\}$ (often, $d = 255$).

The following notations are used to represent the image M:

- Assuming that the dimension of images and thumbnail blocks is $H \times W$ and $B \times B$, respectively.
- $M_B(k, i, j)$ denotes the entire (i, j)-th $B \times B$ thumbnail block of the k-th channel.
- $M_{GB}(c, k, i, j)$ denotes the c-th pixel group (two pixels) in the $M_B(k, i, j)$.
- $p_{msb} \ldots p_1$ denotes the pixel is composed of msb-bit plane, where msb is the number of bits required to represent d in binary, p_1 is **least significant bit (LSB)**, and p_{msb} is **most significant bit (MSB)** (often, $msb = 8$).

Let F be a secure PRF and $\lambda \in N$ be a security parameter. The user is supposed to have a sufficiently complex passphrase

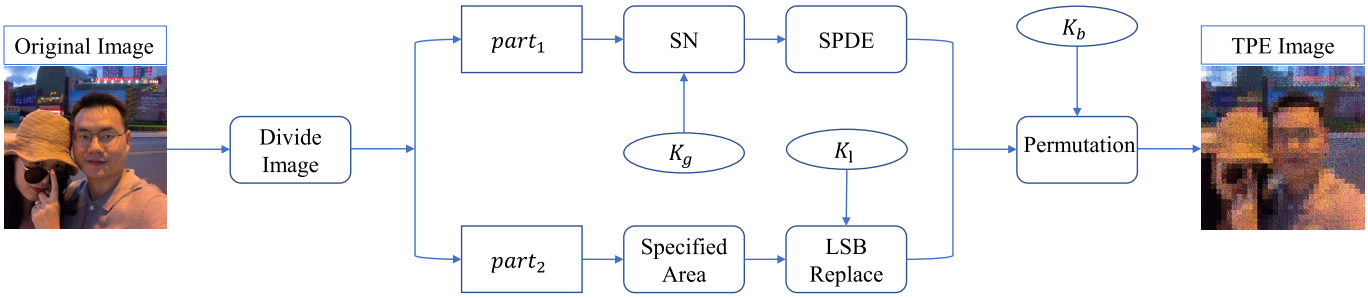


Fig. 3. The overall flow of the encryption scheme.

which can be applied to derive a cryptographic symmetric key K for F, as shown below:

$$K \xleftarrow{\$} \{0,1\}^{\lambda}, \tag{12}$$

where \$ represents a uniform random selector. The details of key generation in the construction are as follows:

- Each block in the M (i.e., $M_B(k,i,j)$) drives unique keys from each other, namely $K_b = F(K, T||k||i||j||per)$ and $K_l = F(K, T||k||i||j||lsb)$, which denote the permutation key and the LSB key in the block, respectively, where $||$ represents the string concatenation, per and lsb are leveraged to ensure the difference between K_b and K_l.
- Each pixel group in the $M_B(k,i,j)$ (i.e., $M_{GB}(c,k,i,j)$) has a unique key $K_g = F(K, T||c||k||i||j)$.
- The security of the F is that each subkey is indistinguishable from an independently uniformly chosen key. F ensures that each subkey is different and utilized only once.

B. Description of the HF-TPE Scheme

1) HF-TPE Encryption: The HF-TPE scheme is independently performed for each block in the image and thus the block $M_B(k,i,j)$ is taken as an example to illustrate the encryption process.

All the pixels in $M_B(k,i,j)$ are divided into groups (i.e., pairs) containing two pixels in a predetermined manner (the easiest way is to divide adjacent pixels into groups) and each pixel group can be regarded as a vector $\vec{v} = (a,b)$.

As shown in the previous section, each vector can be represented by a n_d-bit bitstream, where $s = a+b$, $n_d = \mathrm{Dom}(s)$, namely SN. The bitstream is then encrypted by standard one-time-pad-like password (i.e., K_g). Theoretically, the SPDE scheme is exploited to embed the encrypted bitstream into the vector while guaranteeing the sum of the vector to be constant, which is formally equivalent to **substitution encryption**.

However, if $embedded\ state \in S_3$, only $(n_d - 1)$-bit bitstream can be embedded in $\vec{v}$ at most. It means that extra space is needed to hold the remaining one bit. Otherwise, the original vector will not be restored, i.e., the encrypted image will not be decrypted correctly.

To solve this problem, the pixel group is divided into two parts. One part is the high $(msb-1)$-bit of the pixel, i.e., $p_{msb} \ldots p_2$, marked as $part_1$. The other part is the LSB of the pixel, denoted as $part_2$. For $part_1$, this SN is acquired by rank_s and then encrypted it by XOR operation with the key K_g. Then the encrypted data are embedded by exploiting the SPDE scheme. If there is a remaining bit, which is directly put into $part_2$, i.e., the original LSB is replaced by remaining bit. For $part_2$, a specific area (i.e., the LSB of some pixels in the block) is divided in advance to store these remaining bits. For simplicity, the first bit in each $part_2$ is taken as the area. The above operation is equivalent to the substitution encryption of $part_1$, as shown in **Algorithm 4**. In addition, the XOR operation with the key K_l is applied to encrypt the LSB in each block to prevent the LSB leakage.

Algorithm 4 Pixel Group Encryption in the HF-TPE

Input: Pixel group: $M_{GB}(c,k,i,j)$, key: K_g
Output: Encrypted pixel group: $M_{GE}(c,k,i,j)$

$group_1 = (p_a, p_b) = \lfloor M_{GB}(c,k,i,j)/2 \rfloor$ /* $part_1$ area of $M_{GB}(c,k,i,j)$ */
$part_2 = M_{GB}(c,k,i,j) \mod 2$
$s = p_a + p_b$
$r_g = \mathrm{rank}_s(group_1)$
$SN_g = \mathrm{dec2bin}(r_g)$
$\beta = SN_g \oplus K_g$
$C_g = \mathrm{SPDE}_{\mathrm{emb}}(\mathrm{group}_1, \beta)$ /* **Algorithm 2** */
if $embedded\ state \in S_3$ **then**
    The remaining bit are saved to the $part_2$ specified in $M_{GB}(c,k,i,j)$
end if
$M_{GE}(c,k,i,j) = C_g \times 2 + part_2$
return $M_{GE}(c,k,i,j)$

Finally, the pixels in each block are permuted, i.e., to exploit **pemutation encryption**. More precisely, a permutation operation is sampled π over $\{1, \ldots, B \times B\}$ and the pixels in the block are shuffled according to π. The overall flow of the HF-TPE encryption scheme is shown in Fig. 3.

2) HF-TPE Decryption: First, the permutation decryption is performed in $M_B(k,i,j)$ of the ciphertext image and XOR the LSB with the key K_l for decryption. Second, the pixels in $M_B(k,i,j)$ are divided into pixel groups $M_{GE}(c,k,i,j)$ and let $part_1$ of $M_{GE}(c,k,i,j)$ be $g_E = (pe_a, pe_b)$. Therefore, one can calculate $s = pe_a + pe_b$, $n_d = Dom(s)$.

Algorithm 3 is exploited to extract bitstream from g_E and the key K_g is applied to XOR the bitstream. If the bitstream has less than n_d bits, the remaining bit is taken from the

Algorithm 5 Pixel Group Decryption in the HF-TPE

Input: Pixel group: $M_{GE}(c,k,i,j)$, key: K_g /* The LSB of $M_{GE}(c,k,i,j)$ has been decrypted. */

Output: Encrypted pixel group: $M_{GB}(c,k,i,j)$

1: $g_E = (pe_a, pe_b) = \lfloor M_{GE}(c,k,i,j)/2 \rfloor$ /* $part_1$ area of $M_{GE}(c,k,i,j)$ */
2: $part_2 = M_{GE}(c,k,i,j) \mod 2$
3: $\beta_1 = SPDE_{ext}(g_E)$ /* **Algorithm 3** */l;
4: $\beta_s = \beta_1$
5: **if** *embedded state* $\in S_3$ **then**
6: $\beta_2 =$ The remaining bit is extracted from the $part_2$ of $M_{GE}(c,k,i,j)$
7: $\beta_s = \beta_s \times 2 + \beta_2$
8: **end if**
9: $\beta = \beta_s \oplus K_g$
10: $r_g = \text{bin2dec}(\beta)$
11: $s = pe_a + pe_b$
12: $group_1 = \text{rank}_s^{-1}(r_g)$
13: $M_{GB}(c,k,i,j) = group_1 \times 2 + part_2$
14: **return** $M_{GB}(c,k,i,j)$

specific area of $part_2$. The result is the SN of the original vector. The original $part_1$ can be losslessly restored by exploiting the rank_s^{-1} and then combined with the decrypted $part_2$ to fulfill the decryption, as shown in **Algorithm 5**.

C. Security Analysis

First of all, the subkeys (i.e., K_g, K_l, and K_b) are derived from PRF. The security of PRF is that its output bitstream is indistinguishable from the randomly selected bitstream. Meanwhile, the parameters are different each time PRF is called to generate the subkey, which ensures that each subkey is unique. Therefore, the XOR encrypted ciphertext in the scheme is a uniformly random bitstream and the permutation encryption arrangement is the uniformly random.

Furthermore, in order to analyze the security of substitution encryption, we model the SN of vectors in $\Phi_{sum}^{-1}(s)$ as a Markov chain, where s is the sum of the pixels in the block.

Definition 6: The finite Markov chain refers to the process of moving the elements of a finite set Ω based on a transition matrix called P (the sum of each row of P is 1 and all elements are non-negative, i.e., P is stochastic). For example, the current state $x \in \Omega$ and then the next state to be chosen according to a fixed probability distribution $P(x,.)$ [45].

Definition 7: The stationary distribution π of a Markov chain refers to the existence of a distribution π over Ω satisfying $\pi = \pi P$ [45].

It is known that the distribution of the Markov chain states will tend to a stationary distribution after enough rounds. The numbering of vectors in $\Phi_{sum}^{-1}(s)$ are treated as a Markov chain. For simplicity, the order of the vectors in $\Phi_{sum}^{-1}(s)$ is the corresponding SN. The SNs of the vectors in $\Phi_{sum}^{-1}(s)$ correspond to the states of the Markov chain. The transition probability matrix P of the Markov chain represents the probability of one state transitioning to another state by permuting the order of vectors (i.e., one round). In the HF-TPE scheme, the transition probabilities are determined by PRF. The permutation of vectors in $\Phi_{sum}^{-1}(s)$ is assumed to be a uniform manner. When the distribution state of the Markov chain is stationary distribution, the sampling is indistinguishable from the uniform sampling in the state space of the Markov chain.

The mixing time of the Markov chain is the minimum number of rounds required to arrive at a distribution on Markov chain states that is ϵ-close to the stationary distribution [43].

Let ϵ be a negligible function of the security parameter. Tajik *et al.* proved that the vectors permutation $t_{mix}(\epsilon)$ rounds can satisfy NR-security, which can be directly applied to the HF-TPE scheme. Meanwhile, they gave and proved the formula for calculating the upper limit of the mixing time [43]:

$$t_{mix}(\epsilon) = \lceil \frac{2(log\epsilon - log(|\Omega| - 1))}{log\lambda_*} \rceil, \tag{13}$$

where $|\Omega|$ is the number of states in the Markov chain and λ_* is the second-largest eigenvalue of $P\overline{P}$, which $\overline{P}$ is the conjugate matrix of P.

D. Discussion

The plaintext information leaked in the ciphertext image does not exceed the approximate thumbnail of the original image. It can be seen from the HF-TPE scheme that the information leaked by each encrypted block is the approximate sum of the pixels in the block.

The SN of each vector may have a bit that needs to be stored in the special area in $part_2$ (i.e., step 9 of **Algorithm 4**), which destroys the original bit. Therefore, the noise appears in the decrypted image. In theory, at most 50% of the LSB in each image are destroyed, as the SN of a vector (i.e., two elements) only needs to save one bit in $part_2$ at most. In reality, there may not be extra bits in the embedding process, and even if the extra bits appear, they may be the same as the replaced bits of the LSB. Therefore, the number of destroyed bits is far less than 50% of the pixels in the image. In the measurement of multiple decrypted images, the noise accounts for about 10% to 20% of the total pixels in images.

The effect diagrams of the HF-TPE scheme and TPE-LSB scheme on 1024×1024 RGB images are shown in the Figs. 5, 6, 7, 8, and 9, in which Fig. 5 presents original images and their thumbnails, Figs. 6, 7, 8, and 9 present the encrypted images and their decrypted images of HF-TPE scheme and TPE-LSB scheme, where LSB (i bit) indicates that the maximum flipped pixel bit in the TPE-LSB encryption process is i-bit. It can be seen that the ciphertext image of HF-TPE is closest to its corresponding thumbnail in perception and the decrypted image is almost the same as the original image.

Image privacy is highly subjective, in other words, everyone has a different understanding of the balance between image privacy and usability. The HF-TPE scheme allows users to achieve a good balance between privacy and usability by controlling the resolution of thumbnails in ciphertext. For instance, it can be observed from Fig. 4, which shows the TPE-encrypted images with different block sizes, that the more leaked image information, the better the usability and the

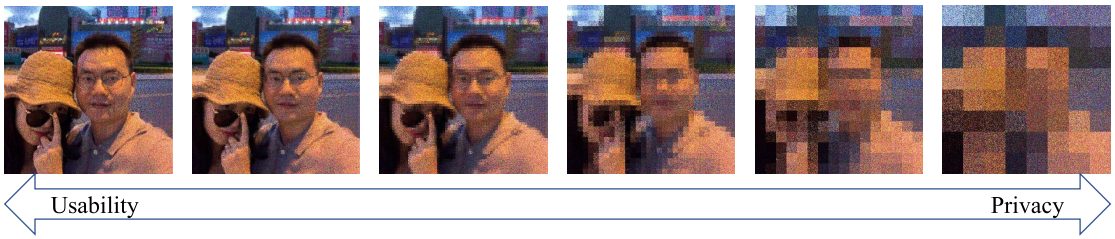


Fig. 4. TPE-encrypted images with different block sizes (4×4, 8×8, 16×16, 32×32, 64×64, 128×128).

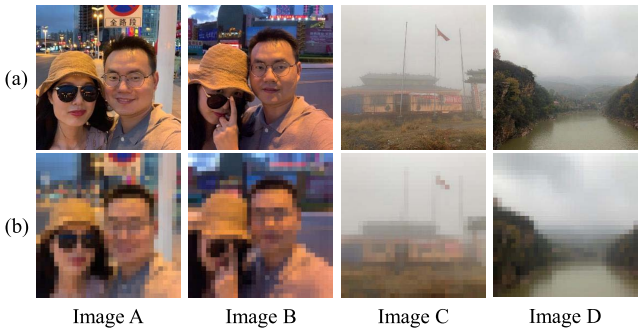


Fig. 5. (a) Original images. (b) Thumbnails (enlarged to 1024×1024) with 32×32 blocks.

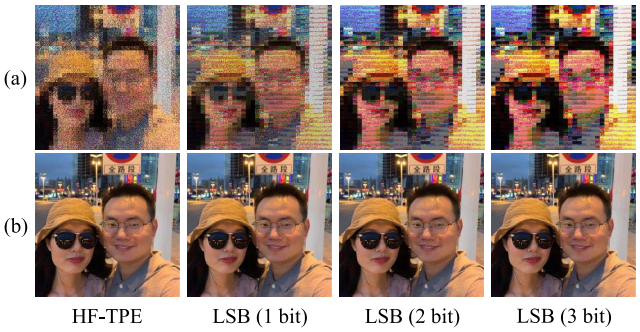


Fig. 6. Encryption and decryption effect of image A. (a) Encryption. (b) Decryption.

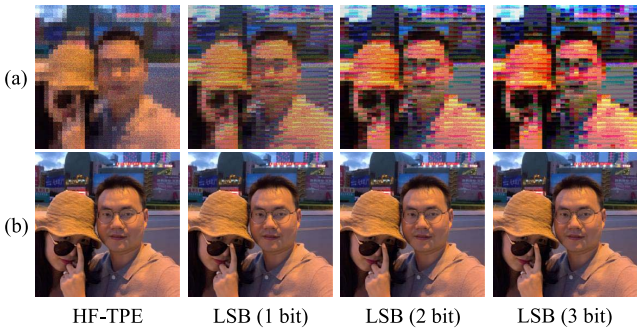


Fig. 7. Encryption and decryption effect of image B. (a) Encryption. (b) Decryption.

worse the privacy. Conversely, the better the privacy, the worse the usability.

Note that we do not claim that the HF-TPE scheme makes it completely impossible for illegal third parties to learn any information from the ciphertext images. In fact, the HF-TPE scheme is to acquire usability by exposing some insensitive information (i.e., image degradation version) in the ciphertext images. Nevertheless, images encrypted by the HF-TPE scheme are difficult for illegal third parties to learn more information from low-resolution images' thumbnails and the exposure of information be able to limited by adjusting the dimension of thumbnail block.

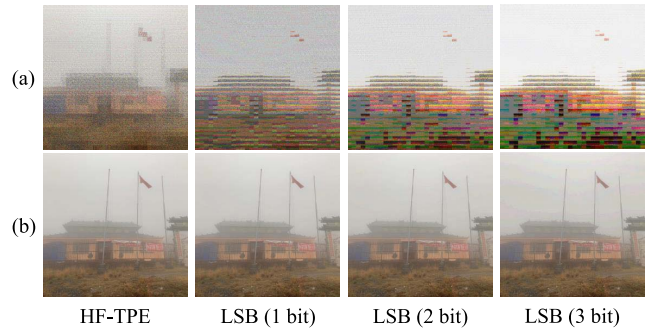


Fig. 8. Encryption and decryption effect of image C. (a) Encryption. (b) Decryption.

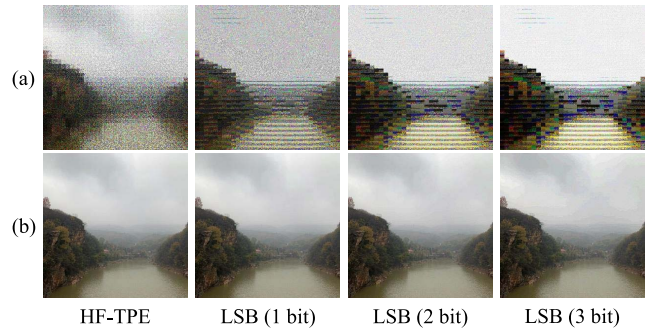


Fig. 9. Encryption and decryption effect of image D. (a) Encryption. (b) Decryption.

VII. Empirical Evaluation

A. Overview

Data: In this section, we exploit two datasets in evaluation. The first dataset is the first part of the Helen dataset[1] [46] with 500 images for performance evaluations. The second dataset is composed of the 46 portraits and 4 Jay Chou portraits for user evaluation.

[1] http://www.ifp.illinois.edu/ vuongle2/helen/

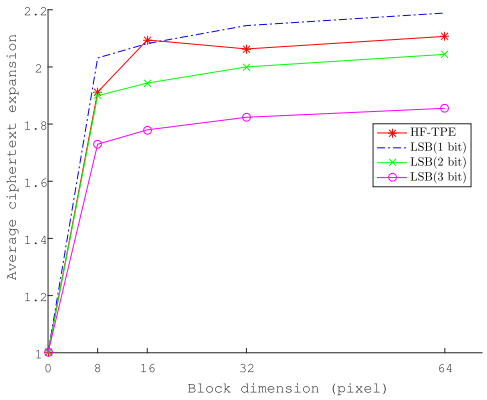


Fig. 10. Average expansion rate of encrypted images.

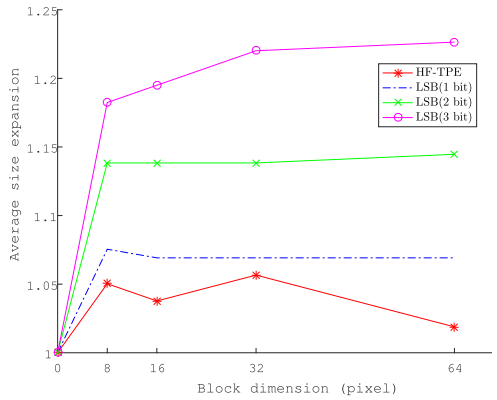


Fig. 11. Average expansion rate of decrypted images.

Pre-Processing: For the convenience of image processing, the images in these two datasets are adjusted to PNG images with bit depth of 24 bits. Then the images are resized to the resolution of 512×512.

So far, only two approximate TPE schemes have been proposed in work [21], i.e., DRPE and TPE-LSB. The former scheme may fail in decryption correctly, which is usually unacceptable for an encryption scheme. The latter scheme does not have this problem and the result of decryption is noisy, which is similar to the HF-TPE scheme. Therefore, the performance evaluation experiments are carried out by comparing HF-TPE with TPE-LSB. The TPE-LSB scheme approximates the thumbnail by flipping the LSB of the pixel in the image encrypted by traditional encryption. According to experiments, the quality of the decrypted image is poor after flipping more than 3 bits. As a result, it can only flip 3 bits at most.

B. Performance Evaluations

1) Size of Encrypted Images and Decrypted Images: Figure 10 shows the average size expansion rate of ciphertext images, i.e., $\frac{x}{y}$, where x and y represent the size of encrypted and original images, respectively. Although TPE does not increase the size of ciphertext images in theory, the size of ciphertexts still have a certain degree of expansion compared with the original images. The main reason is that the formats like PNG exploit some lossless compression algorithms for images. It is difficult to compress ciphertext images due to the poor spatial locality. With the increase of the block dimension, the permutation of pixels in the block is increasingly likely to destroy the spatial locality between pixels. In addition, the TPE-LSB scheme only flips 0 to 1 or 1 to 0 in a block. Therefore, the more bits are flipped, the better the spatial locality is and the smaller the ciphertext expansion is in the ciphertext images. However, this kind of flip is irreversible. Moreover, an interesting phenomenon is found, i.e., the richer the details of the image, the lower the expansion rate of the ciphertext image.

Figure 11 shows the average size expansion rate of decrypted images, demonstrating that the expansion rate of the HF-TPE scheme is the lowest and even almost no size expansion since this scheme introduces the least noise in the encryption process. In the TPE-LSB scheme, the more bits are flipped, the worse the spatial locality is and thus the higher the expansion rate is, since the flipping is irreversible. In general, the low expansion rate of decrypted images is more important than the expansion rate of encrypted images, because the decrypted images in the user devices with limited resources while the encrypted ones in the cloud.

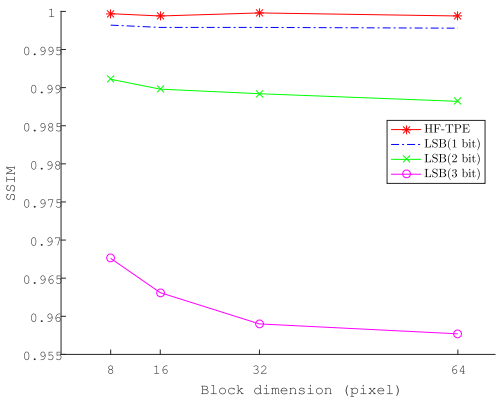


Fig. 12. Average SSIM of decrypted images.

2) Perceptual Quality of Decrypted Images: Two indicators commonly used are exploited to measure image perceptual quality, i.e., structural similarity (SSIM) [47] and peak signal to noise radio (PSNR). The closer SSIM is to 1 or the higher the PSNR is, the better the quality is.

Figure 12 presents the average SSIM of decrypted images, demonstrating that the HF-TPE scheme has the highest quality of decrypted images and the SSIM is basically equal to 1 (not less than 0.9995). Figure 13 shows the average PSNR of decrypted images, which manifests that the HF-TPE scheme also achieves the best result on the PSNR index. Therefore, it is demonstrated that the decrypted image in the HF-TPE scheme is closer to the original image.

3) Perceptual Quality of Ciphertext Thumbnails: In this section, the thumbnail of the original image is generated according to the method described above, in which the thumbnail dimensions are set to 100 and 150. Meanwhile, ciphertext images are scaled to the same dimensions.

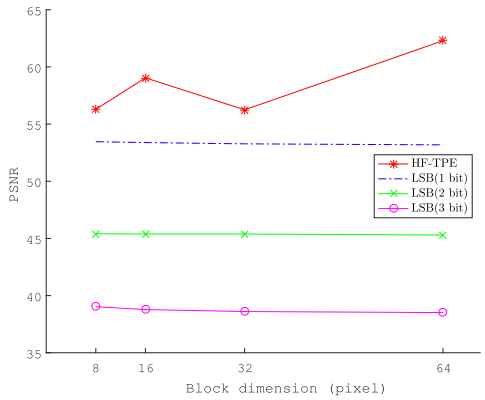


Fig. 13. Average PSNR of decrypted images.

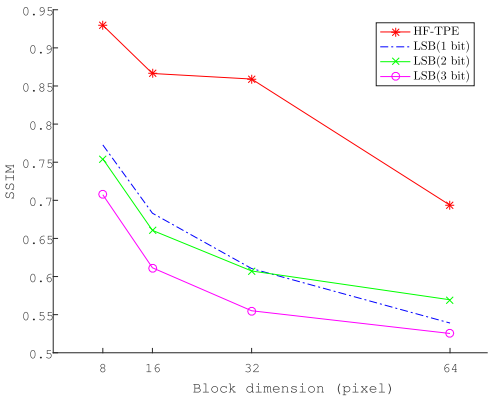


Fig. 16. Average SSIM of encrypted images when the dimension is 150.

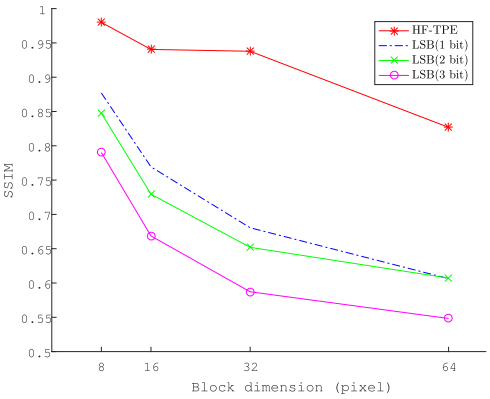


Fig. 14. Average SSIM of encrypted images when the dimension is 100.

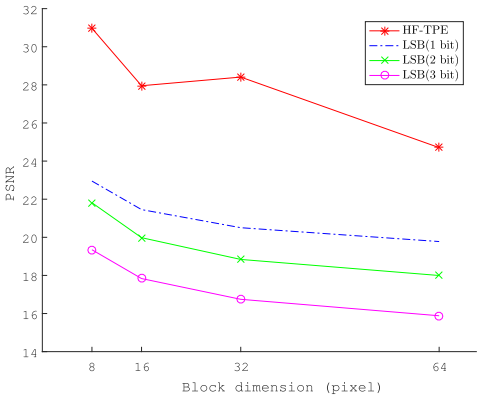


Fig. 17. Average PSNR of encrypted images when the dimension is 150.

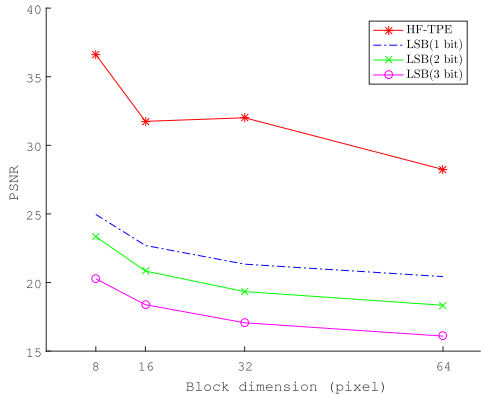


Fig. 15. Average PSNR of encrypted images when the dimension is 100.

Figures 14 and 15 show the average SSIM and PSNR of encrypted images when the thumbnail dimension is 100. They indicate that the HF-TPE scheme far exceeds the TPE-LSB scheme on both indicators. Similarly, Figures 16 and 17 display the case of the dimension 100. They also demonstrate that the HF-TPE scheme works best on these indicators. Therefore, the ciphertext perceptual quality in the HF-TPE scheme is the better, i.e., the ciphertext thumbnail is closer to the original one.

4) Security Against Facial Detection: The face detection technology is one of the most advanced detection technology so far and it greatly threatens personal privacy in images. Therefore, it is exploited to verify the effectiveness of the HF-TPE scheme against face detection.

In this evaluation, we exploit face++, which is one of China's most advanced computer vision platforms.[2] Initially, the original images are detected and a total of 593 faces are found in the first dataset. Then the HF-TPE and TPE-LSB scheme are utilized to encrypt images with different block dimensions, respectively. Figure 18 shows the success rate of face detection for TPE-encrypted images on the face++ platform. The results indicates that the probability that the face in the TPE-encrypted images can be detected is non-negligible when the dimension of thumbnail blocks is less than 32. Meanwhile, the probability is almost zero when the dimension is 32. Therefore, the default block dimension applied in the user evaluation can be 32.

C. User Evaluation

In this section, the usability of the images encrypted by the HF-TPE scheme to users are evaluated. Specifically, users' ability to recognize encrypted images when images are uploaded to the cloud. To ensure the objectivity of the evaluation, the images of Jay Chou are selected as the evaluative material since almost all Chinese young people have seen Jay Chou's videos and images in their daily lives and

[2]https://www.megvii.com/

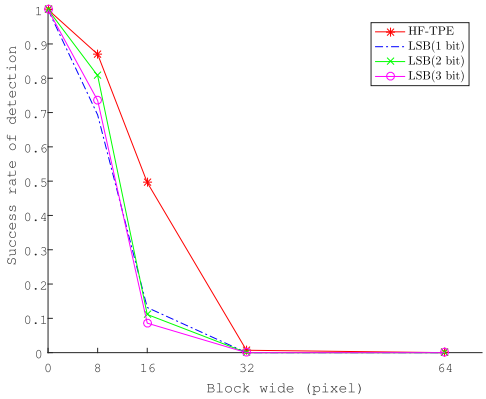


Fig. 18. Average face detection success rate on the face++ platform.

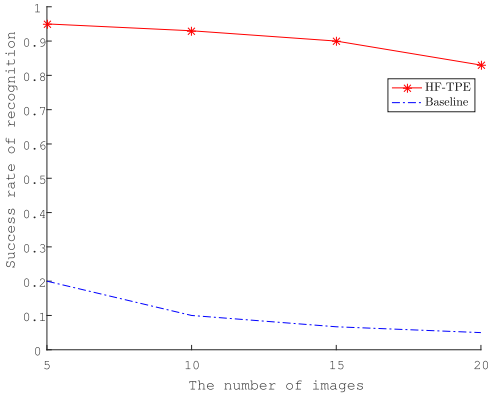


Fig. 20. The success rate of the subjects in identifying Jay Chou.

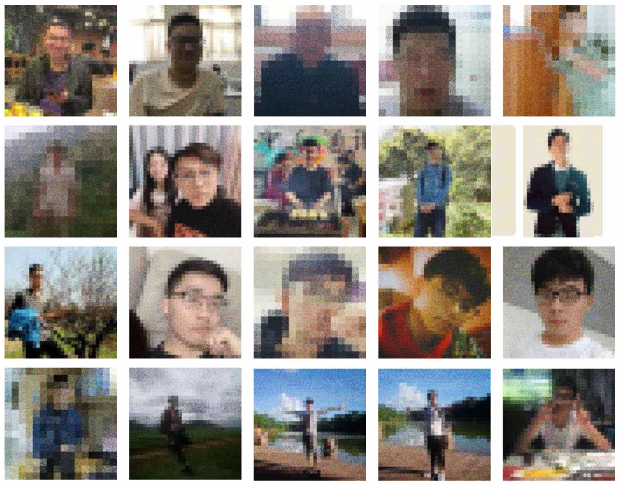


Fig. 19. 20 images encrypted by the HF-TPE scheme with the block dimension 32×32, one of which is Jay Chou.

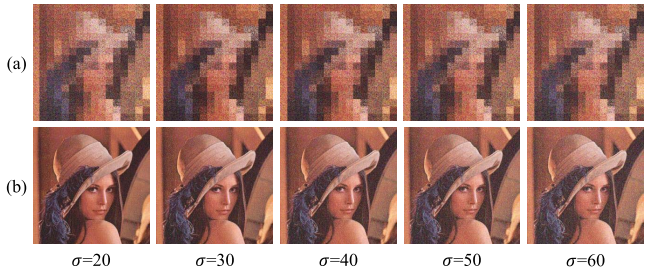


Fig. 21. Ciphertext images with noise and their decrypted images, the block dimension is 32×32. (a) Ciphertext images with different coefficient σ. (b) The corresponding decrypted images.

therefore have prior knowledge of Jay Chou. Notice that the prior knowledge not only refers to the Jay Chou's image in the evaluation that the users may have seen before, but also refers to the users' familiarity with some of the features used for identification such as Jay Chou's body shape and posture.

The images in the second dataset are encrypted by the HF-TPE scheme, where the parameter of the block dimension is 32. Meanwhile, these images are randomly divided into four sets with 5, 10, 15, and 20 images to simulate users browsing the album (as shown in Fig. 19) and each set has and only has one different Jay Chou's image. Then 100 subjects were asked to point out which image is Jay Chou from each set.

Figure 20 reflects the success rate of the subjects in pointing out Jay Chou successfully, in which the baseline is the probability of success under random guesses. It shows that the subjects can still distinguish images through visual content in ciphertext images combined with prior knowledge when the face detection algorithm is invalid. In addition, the subjects who cloud not correctly point out the Jay Chou's image are interviewed and most of them indicated that they were not familiar with Jay Chou. Meanwhile, some of the subjects who cloud correctly point out Jay Chou in all four sets are also interviewed and most of them expressed that the encrypted images preserve the key features such as character's posture. One familiar with Jay Chou can easily identify Jay Chou according to these key features.

D. Security Evaluations

1) Noise Attack: In this section, to evaluate the robustness of the HF-TPE scheme against additive noise, a white Gaussian random noise G with a mean of zero is added to the encrypted images:

$$I' = I(1 + \sigma G), \tag{14}$$

where I and I' present the ciphertext image under the ideal condition and the noise-affected condition, respectively, and the parameter σ represents the noise intensity.

Figure 21 shows the encrypted and decrypted images when σ is set to 20, 30, 40, 50, and 60, respectively, presenting that even if the noise is added to the image, the visual effect of the ciphertext image and the decrypted image preserve essentially unchanged. Therefore, it is concluded that the HF-TPE scheme can resist noise attack.

2) Data Loss Attack: The encrypted image may loss some data when transmitted over the insecure channel. For an excellent encryption algorithm, the image should still be able to correctly decrypt even if few data are lost. We evaluate the random loss of 1%, 5%, 10%, 15%, and 20% data in the ciphertext image to detect the tolerance of the HF-TPE scheme to data loss attack. Figure 22 shows that the encrypted images and the corresponding decrypted images still preserve high

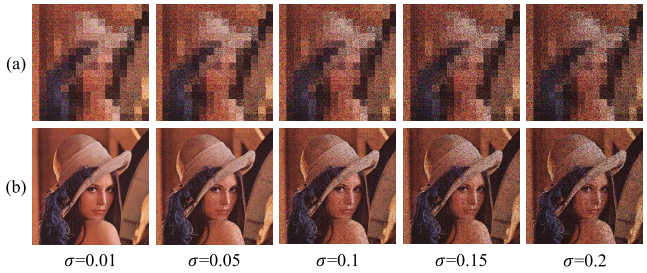


Fig. 22. Ciphertext images with data loss and their decrypted images, the block dimension is 32 × 32. (a) Ciphertext images with different coefficient σ. (b) The corresponding decrypted images.

visual quality despite data loss. Therefore, it is demonstrated that the HF-TPE scheme can resist the loss of encrypted data.

VIII. Conclusion

When images are stored in the cloud, they are no longer directly under the physical control of users, which inevitably brings privacy risks. The application of off-the-shelf image encryption methods in the cloud raises various usability problems, i.e., forcing users to choose between image privacy and usability. This flies in the face of the original intention of the cloud service. The proposed HF-TPE scheme can well balance the privacy and usability of images. Meanwhile, the trade-off between privacy and usability is tunable by controlling the size of the block. We exploit psychological research findings to enable users to mentally recover ciphertext images in real time. In addition, the proposed scheme can be used without any modification to existing cloud services. Experiments reveal that this scheme outperforms previous works. The ciphertext images have better perceptual quality and the decrypted ones have less noise and size expansion. Meanwhile, the usability of users is still guaranteed under the premise of failure of the face detection algorithm.

References

[1] M. Llc. (2020). *How Many Photos Will be Taken in 2020?*. [Online]. Available: https://focus.mylio.com/tech-today/how-many-photos-will-be-taken-in-2020

[2] M. Milian. (2010). *Digital Photos can Reveal Your Location, Raise Privacy Fears*. [Online]. Available: http://edition.cnn.com/2010/TECH/web/10/15/photo.gps.privacy/index.html/

[3] L. Fan, "A demonstration of image obfuscation with provable privacy," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2019, p. 608.

[4] A. Li, D. Darling, and Q. Li, "PhotoSafer: Content-based and context-aware private photo protection for smartphones," in *Proc. IEEE Symp. Privacy-Aware Comput. (PAC)*, Sep. 2018, pp. 10–18.

[5] R. Fallon, "Celebgate: Two methodological approaches to the 2014 celebrity photo hacks," in *Proc. Int. Conf. Internet Sci.* Cham, Switzerland: Springer, 2015. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-18609-2_4

[6] Z. Schiffer. (2019). *The big Facebook Outage Offers a Behind-the-Scenes Look at how the Social Network's ai 'Sees' Your Photos and Interprets Them for Blind Users*. [Online]. Available: https://www.businessinsider.com/facebooks-photo-outage-reveals-how-ai-sees-your-photos-2019-7

[7] G. J. Dance, M. LaForgia, and N. Confessore. *As Facebook Raised a Privacy Wall, it Carved an Opening for Tech Giants*. 2018. [Online]. Available: https://www.nytimes.com/2018/12/18/technology/facebook-privacy.html

[8] X. Kang and R. Tao, "Color image encryption using pixel scrambling operator and reality-preserving MPFRHT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 7, pp. 1919–1932, Jul. 2019.

[9] C. Qin, Q. Zhou, F. Cao, J. Dong, and X. Zhang, "Flexible lossy compression for selective encrypted image with image inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 11, pp. 3341–3355, Nov. 2019.

[10] P. Puteaux and W. Puech, "CFB-then-ECB mode-based image encryption for an efficient correction of noisy encrypted images," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Nov. 18, 2020, doi: 10.1109/TCSVT.2020.3039112.

[11] R. L. Gregory, "Knowledge in perception and illusion," *Philos. Trans. Roy. Soc. London. B, Biol. Sci.*, vol. 352, no. 1358, pp. 1121–1127, 1997.

[12] H. Kinjo and J. G. Snodgrass, "Does the generation effect occur for pictures?" *Amer. J. Psychol.*, vol. 113, no. 1, pp. 95–121, Apr. 2000.

[13] T. Denning, K. Bowers, M. van Dijk, and A. Juels, "Exploring implicit memory for painless password recovery," in *Proc. Conf. Hum. Factors Comput. Syst.*, May 2011, pp. 2615–2618.

[14] P. Kok, G. J. Brouwer, M. A. J. van Gerven, and F. P. de Lange, "Prior expectations bias sensory representations in visual cortex," *J. Neurosci.*, vol. 33, no. 41, pp. 16275–16284, Oct. 2013.

[15] G. A. Rousselet, S. J. Thorpe, and M. Fabre-Thorpe, "Processing of one, two or four natural scenes in humans: The limits of parallelism," *Vis. Res.*, vol. 44, no. 9, pp. 877–894, Apr. 2004.

[16] C. V. Wright, W.-C. Feng, and F. Liu, "Thumbnail-preserving encryption for JPEG," in *Proc. 3rd ACM Workshop Inf. Hiding Multimedia Secur.*, Jun. 2015, pp. 141–146.

[17] J. Chen, L. Chen, and Y. Zhou, "Universal chosen-ciphertext attack for a family of image encryption schemes," *IEEE Trans. Multimedia*, early access, Jul. 24, 2020, doi: 10.1109/TMM.2020.3011315.

[18] L. Y. Zhang, Y. Liu, C. Wang, J. Zhou, Y. Zhang, and G. Chen, "Improved known-plaintext attack to permutation-only multimedia ciphers," *Inf. Sci.*, vols. 430–431, pp. 228–239, Mar. 2018.

[19] J. Chen, L. Chen, and Y. Zhou, "Cryptanalysis of image ciphers with permutation-substitution network and chaos," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Sep. 4, 2020, doi: 10.1109/TCSVT.2020.3021908.

[20] A. Jolfaei, X.-W. Wu, and V. Muthukkumarasamy, "On the security of permutation-only image encryption schemes," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 2, pp. 235–246, Feb. 2016.

[21] B. Marohn, C. V. Wright, W.-C. Feng, M. Rosulek, and R. B. Bobba, "Approximate thumbnail preserving encryption," in *Proc. Multimedia Privacy Secur.*, Oct. 2017, pp. 33–43.

[22] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers, "Format-preserving encryption," in *Proc. Int. Workshop Sel. Areas Cryptogr.* Berlin, Germany: Springer, 2009, pp. 295–312. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-05445-7_19

[23] J. Yu, P. Lu, Y. Zhu, G. Xue, and M. Li, "Toward secure multikeyword top-K retrieval over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 4, pp. 239–250, Jul. 2013.

[24] H. Li, Y. Yang, Y. Dai, S. Yu, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Trans. Cloud Comput.*, vol. 8, no. 2, pp. 484–494, Apr. 2020.

[25] B. Chen, L. Wu, H. Wang, L. Zhou, and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5813–5825, Jun. 2020.

[26] Y. Tian, Y. Hou, and J. Yuan, "CAPIA: Cloud assisted privacy-preserving image annotation," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, Oct. 2017, pp. 1–9.

[27] F. Markatopoulou, V. Mezaris, and I. Patras, "Implicit and explicit concept relations in deep neural networks for multi-label video/image annotation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 6, pp. 1631–1644, Jun. 2019.

[28] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang, and S.-F. Chang, "Multi-modal multi-scale deep learning for large-scale image annotation," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1720–1731, Apr. 2019.

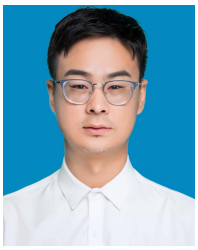
[29] X. Ke, J. Zou, and Y. Niu, "End-to-end automatic image annotation based on deep CNN and multi-label data augmentation," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2093–2106, Aug. 2019.

[30] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 784–798, Jul. 2019.

[31] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "BOEW: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," *IEEE Trans. Services Comput.*, early access, Jul. 10, 2019, doi: 10.1109/TSC.2019.2927215.

[32] E. von Zezschwitz, S. Ebbinghaus, H. Hussmann, and A. De Luca, "You Can't watch this!: Privacy-respectful photo browsing on smartphones," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, May 2016, pp. 4320–4324.
[33] M. Khamis, T. Seitz, L. Mertl, A. Nguyen, M. Schneller, and Z. Li, "Passquerade: Improving error correction of text passwords on mobile devices by using graphic filters for password masking," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, May 2019, pp. 1–8.
[34] C. N. Ochotorena and Y. Yamashita, "Anisotropic guided filtering," *IEEE Trans. Image Process.*, vol. 29, pp. 1397–1412, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8844987
[35] E. Y.-N. Sun, H.-C. Wu, C. Busch, S. C.-H. Huang, Y.-C. Kuan, and S. Y. Chang, "Efficient recoverable cryptographic mosaic technique by permutations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 112–125, Jan. 2021.
[36] J. He *et al.*, "Puppies: Transformation-supported personalized privacy preserving partial image sharing," in *Proc. 46th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2016, pp. 359–370.
[37] L. Zhang, T. Jung, C. Liu, X. Ding, X.-Y. Li, and Y. Liu, "POP: Privacy-preserving outsourced photo sharing and searching for mobile devices," in *Proc. IEEE 35th Int. Conf. Distrib. Comput. Syst.*, Jun. 2015, pp. 308–317.
[38] F. Peng, X.-W. Zhu, and M. Long, "An ROI privacy protection scheme for H.264 video based on FMO and chaos," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1688–1699, Oct. 2013.
[39] S. Beugnon, P. Puteaux, and W. Puech, "Privacy protection for social media based on a hierarchical secret image sharing scheme," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 679–683.
[40] A. Tonge and C. Caragea, "Image privacy prediction using deep neural networks," *ACM Trans. Web*, vol. 14, no. 2, pp. 1–32, Apr. 2020.
[41] J. Yu, B. Zhang, Z. Kuang, D. Lin, and J. Fan, "IPrivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 5, pp. 1005–1016, May 2017.
[42] J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "Visual privacy protection methods: A survey," *Expert Syst. Appl.*, vol. 42, no. 9, pp. 4177–4195, Jun. 2015.
[43] K. Tajik *et al.*, "Balancing image privacy and usability with thumbnail-preserving encryption," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, p. 295.
[44] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *J. ACM*, vol. 33, no. 4, pp. 792–807, Aug. 1986.
[45] D. A. Levin and Y. Peres, *Markov Chains Mixing Times*. Providence, RI, USA: American Mathematical Society, 2006.
[46] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2012, pp. 679–692.
[47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

Yushu Zhang (Member, IEEE) received the B.S. degree from the School of Science, North University of China, Taiyuan, China, in 2010, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, China, in 2014. He held various research positions at the City University of Hong Kong, Southwest University, the University of Macau, and Deakin University. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. He has produced more than 100 research peer-reviewed journals and conference papers. His research interests include multimedia security, artificial intelligence, and blockchain. He is also an Associate Editor of *Information Sciences* and *Signal Processing*.



Ruoyu Zhao received the B.S. degree in computer science and technology from the School of Software, Zhengzhou University, Zhengzhou, China, in June 2019. He is currently pursuing the M.S. degree in cyberspace security with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include multimedia security and cloud computing security.

Xiangli Xiao received the B.S. degree from the College of Electronic and Information Engineering, Southwest University, Chongqing, China, in June 2020. He is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His current research interests include multimedia security, blockchain, and cloud computing security.

Rushi Lan (Member, IEEE) received the B.S. degree in information and computing science and the M.S. degree in applied mathematics from the Nanjing University of Information Science and Technology, Nanjing, China, and the Ph.D. degree in software engineering from the University of Macau, Macau, China. He is currently an Associate Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. His research interests include image classification, image denoising, and metric learning.

Zhe Liu (Senior Member, IEEE) received the B.S. and M.S. degrees from Shandong University, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Luxembourg, Luxembourg, in 2015. He is currently a Professor with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. His research interests include security, privacy, and cryptography solutions for the Internet of Things. He has coauthored more than 100 research peer-reviewed journals and conference papers. He was a recipient of the prestigious FNR Awards–the Outstanding Ph.D. Thesis Award in 2016, the ACM CHINA SIGSAC Rising Star Award in 2017, and the DAMO Academy Young Fellow in 2019. He serves as a program committee member in more than 60 international conferences, including the Program Chair for INSCRYPT 2019, CRYPTOIC 2019, and ACM CHINA SIGSAC 2020.

Xinpeng Zhang (Member, IEEE) received the B.S. degree in computational mathematics from Jilin University, Changchun, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, Shanghai, China, in 2001 and 2004, respectively. Since 2004, he has been with the faculty of the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He is also with the faculty of the School of Computer Science, Fudan University, Shanghai. He was a Visiting Scholar with The State University of New York at Binghamton from 2010 to 2011 and also an experienced Researcher with Konstanz University, sponsored by the Alexander von Humboldt Foundation from 2011 to 2012. His research interests include multimedia security, image processing, and digital forensics. He has authored or coauthored more than 200 articles in his research interests. He was an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2014 to 2017.