Research Article

A VQ-Based Joint Fingerprinting and Decryption Scheme for Secure and Efficient Image Distribution

Ming Li ¹, Hua Ren ¹, En Zhang ¹, Wei Wang ¹, Lin Sun ¹ and Di Xiao ²

¹College of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China

²College of Computer Science, Chongqing University, Chongqing 400030, China

Correspondence should be addressed to Ming Li; liming@htu.edu.cn

Received 19 February 2018; Revised 26 April 2018; Accepted 13 June 2018; Published 6 August 2018

Academic Editor: Emanuele Maiorana

Copyright © 2018 Ming Li 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 first joint fingerprinting and decryption (JFD) for vector quantization (VQ) images addressed the problem that the decrypted multimedia data may be redistributed from authorized customers to unauthorized customers. The scheme also caused conventional JFD methods to be equipped with a special ability to resist noise interference. Till now, some existing schemes related have been proposed to protect the multimedia content and distribution, but these schemes failed to tackle several problems existing in the original JFD scheme based on VQ image, including high transmission cost and severe fingerprinted image distortion. In this paper, we propose a novel JFD method by combining a weight-sum function with fingerprinting embedding and extraction for VQ images. Under the combination, the visual quality of the fingerprinted image is further improved; also the fingerprint extraction implements a blind extraction process. Experiments and analyses demonstrate the feasibility of the proposed method.

1. Introduction

The digitized form of numerous multimedia contents not only facilitates various operations but also allows fast distribution from one physical location to another via the Internet. As any advancement in technique can always be used in good and bad ways, the convenience of technique simultaneously incurs the potential security hazard when illegally redistributing the protected content via the Internet. To improve the security of multimedia contents in transmission and distribution, digital fingerprint has been widely investigated over the past few years [1–4]. Fingerprint is the technique that is utilized to trace the traitors to ensure the reasonable distribution of the protected content; in this technique a fingerprint sequence only represents a unique user's identity and is usually embedded in such a way that remains invisible to the human eye. Conventional fingerprint embedding can be achieved in the spatial domain [5, 6] or in the transform domain [2, 7]. Once the fingerprinted content is illegally redistributed, the fingerprint sequence is required for extracting and checking the user's identity.

Specifically, fingerprint embedded into host image is especially designed in various application scenarios. In the

transmitter-side fingerprinting application [8, 9], such fingerprint is embedded on the transmitted side and then the fingerprinted contents are encrypted and sent to users via multiple unicast transmission. For the network-based fingerprinting application [10], this type of fingerprint is embedded as the data travels through the network with special network devices. The last type of fingerprint, i.e., the receiver-side fingerprint, which is the most promising method in the fingerprint research field, has been widely applied in some scenarios where the pirate possibly redistributes his or her copies over the Internet. Combining receiver-side fingerprint and image decryption operation can effectively trace the owners of illegal redistribution contents, which is the well-known traitor tracing problem.

The first receiver-side fingerprinting method [11] was introduced by processing decryption and fingerprinting independently. The respective operations of decryption and fingerprinting embedding made it possible for the attackers to intercept the decrypted content and illegally redistribute it via the Internet. To decrease the possibility of content leakage, some novel joint fingerprint and decryption (JFD) schemes were produced in [7, 12, 13], which ensure the fingerprint performed during decryption process. Although

these schemes effectively resolve the problem of illegal distribution at the receiver end, they cannot be applied to the VQ (vector quantization) domain and the encrypted image cannot effectively resist noise interference. To solve this problem, a new JFD scheme for VQ images was proposed by Lin et al. in 2012 [14]. In their method, the encryption process could be classified into two types: one was that the most well-known chaotic function and static-tree were employed in the permutation and codeword substitution processes, respectively. The other was to utilize the dynamic-tree and session key into the substitution process. Recently, a larger number of JFD schemes on the basis of Lin et al. method have been proposed [15–18]; in these schemes, the security of JFD methods was further enhanced and applied into different types of practical scenarios. Unfortunately, there still exist some problems to be solved for the original scheme [14].

In this paper, we improve the method of decryption and fingerprinting process of [14] through using index-block-based decryption method and a weight-sum function of [19]; namely, the weight-sum function is employed to embed $(n+1)$ fingerprint bits into a block with 2^n codeword indices by modifying only one codeword index. More specifically, the key factor for JFD method based on VQ image is to achieve fingerprint embedding during decryption. A shortcoming of the original scheme [14] is that it only easily embeds each fingerprint bit into each codeword index but has limitations not only in terms of the flexibility of maximum fingerprint bits but also in terms of the quality of the fingerprinted image. Conversely, in the proposed scheme, the major works concentrate on the JFD process. We select a block-based fingerprint embedding to implement this process. The major difference between the proposed method and scheme [14] is the way of how to efficiently embed fingerprint into codeword index and ensure the changeable maximum fingerprint capacity. The major merits are summarized as follows:

- (1) The encrypted codeword indices are capable of being efficiently transmitted with much smaller bandwidth and cost.
- (2) The flexibility of fingerprint embedding can be ensured, and the changeable maximum fingerprint capacity can be achieved.
- (3) The fingerprint extraction operation in the traitor tracing process is a blind extraction process.
- (4) Comparing with the original scheme in [14], the number of modifiable blocks is obviously lowered when introducing certain user's fingerprinting sequence. That is, given the same embedding rate, the PSNR of our method is higher than that in the original scheme.

2. Preliminary

A brief overview of the concepts used in the proposed scheme is discussed as follows.

2.1. Vector Quantization. Vector quantization (VQ) is the technique that converts original image blocks into specific

codeword indices relying on the known codebook to implement the goal of compression, in which the design of codebook directly affects the compression efficiency and image restoration quality. It has a similar functionality with palette technique, the latter transforms color value into index value which can be found by color look-up table, and the maximum of indispensable color of an image directly determines the ultimate compression result. Generally, vector quantization (VQ) designed for image compression is composed of two parts: encoding and decoding processes. Encoding process is to divide original image into nonoverlapping blocks with size of $w \times h$ and map the divided image blocks into a finite subset of codebooks (CB), where $CB = \{Y_1, Y_2, \dots, Y_L\}$ denotes the codebook and Y_i is the i th codeword that comprises $Y_i = \{y_1, y_2, \dots, y_{w \times h}\}$. For each vector X of original image, its nearest codeword in the codebook can be found by using Euclidean distance.

$$d(X, Y_i) = \|X - Y_i\| = \sqrt{\sum_{j=1}^{w \times h} (x_j - y_{i,j})^2}, \quad (1)$$

where x_j and $y_{i,j}$ denote the j th elements of vectors X and Y_i , respectively. The generated nearest codeword is then exploited to decrypt the vector X , i.e., one block of divided image. Transforming the encrypted indices into VQ image is the decoding process which can be easily realized by table look-up.

2.2. Principal Components Analysis. As [14] depicted, principal components analysis (PCA) algorithm has attracted considerable attention from various research fields, especially used in data and signal analysis [20, 21]. Here, the functionality of PCA is mainly used for sorting codewords of codebook before encryption, which is essential in the encryption and JFD phases. Given the unsorted codewords C_1, C_2 , and C_L , after applying PCA algorithm, the sorted codewords are denoted as Y_1, Y_2 , and Y_L , where L represents the size of codebook. The detailed process of PCA method can be found in [14].

2.3. Weight-Sum Function. An essential starting point of designing weight-sum function is to conceal as much additional data as possible into cover image with modifying as little data as possible, and its fundamental objective is consistent with that of exploiting modification direction (EMD) [22]. Concretely, the methods of both are to conceal given bits into one block with modifying only a pixel; the former transforms $(n+1)$ bits into decimal number under the size 2^n of one block to implement data hiding, but the latter requires converting additional data into $(2n+1)$ -ary notational system according to the size n of a block to attain data embedding. A weight-sum function [19] is also a typical application for watermarking embedding, which is primarily used for the detection of the tempered areas. The main contributions of the weight-sum function in [19] can be summarized as follows. In the watermark embedding, the host image is first divided into blocks and each block contains 2^n pixels, and then $(n+1)$ authentication bits need

FIGURE 1: The workflow of the weight-sum function.

to be embedded into one block with modifying only a pixel. However, how to find the position of the modifiable pixel is a pressing problem to be solved, which is the functionality of the weight-sum function. Figure 1 shows the workflow of the weight-sum function, and one can easily know that the weight-sum function in Figure 1 directly implements the watermark embedding process.

Note that embedding $(n+1)$ authentication bits into each block denotes that the whole embeddable authentication bits are $N_p(n+1)/2^n$ bits where N_p denotes the number of pixels of the host image, whereas the maximum number of the modifiable pixels is $N_p/2^n$. In other words, the quality of the watermarked image can be effectively ensured and the tampered areas can also be detected with the weight-sum function. For simplicity, let $n=2$ be an example to illustrate the detailed modification process. The host image is divided into some blocks with 2×2 pixels and suppose that one of the blocks contains four pixels 87, 212, 114, and 186. The binary value of a watermark subsequence is $(110)_2=6$. The value of one block can be easily computed by the weight-sum function as $f = (1 \times 87 + 2 \times 212 + 3 \times 114 + 4 \times 186) \bmod 2^3 = 5$. Since the value of weight-sum is not equal to the binary value of fingerprint subsequence, i.e., $5 \neq 6$, one pixel of the block should be found and modified so that the value of the block computed by the weight-sum f equals the binary value of the fingerprint subsequence. The difference between them is $d = 6 - 5 = 1$ and thus the modifiable pixel is the first pixel. The four pixels in the modifiable block are 88, 212, 114, and 186. The detailed modification process can be found in [14].

3. The Proposed Method

3.1. The Framework. The framework of the proposed JFD scheme based on VQ is shown in Figure 2. For the sender side, a given codebook is first sorted by PCA algorithm, and then the original image is transformed into index matrix combining the sorted codebook with VQ process. Under dynamic-tree keys, the indices substitution process is implemented by replacing the current index with any index depending on a given session key, which can be found in [14]. Besides, to increase the capability for resisting brute-force attacks, the permutation is still utilized before the codeword indices substitution process. For the receiver side, fingerprint sequence is embedded with ± 1 codeword indices replacement during the decryption process. For instance, when the modifiable codeword index in one block is found,

the ± 1 codeword indices replacement can be achieved by displacing the modifiable codeword index with its adjacent codeword indices in the sorted codebook. The detailed JFD method is depicted in the next subsection.

3.2. Block-Wise Joint Fingerprint and Decryption. In this subsection, the proposed JFD scheme in the receiver side is presented in detail. We replace the codeword-index-based JFD process with the proposed block-based JFD method. That is, one codeword index is only regarded as a part of the block. The process of the proposed JFD scheme is illustrated in Figure 3. The encrypted image is first divided into N blocks, and any block i containing 2^n codeword indices is decrypted by dynamic-key trees and simultaneously embedded fingerprint subsequence with the weight-sum function.

Exploiting the block-based idea is mainly attributed to two advantages: one is that only modifying one codeword index with ± 1 can embed $n+1$ bits' fingerprint into a block of 2^n indices, which is an extended application of the block-based method proposed by the authors in [19], and the other is that the block-based JFD method can ensure a blind fingerprint extraction operation; i.e., the extracted fingerprint sequence can directly generate users' identities without the content of the original VQ image.

Note that the proposed image encryption process based on dynamic-key trees depending on session key is different from the original encryption scheme. Consider the same situation where both the original scheme and the proposed JFD scheme accomplish encryption process with dynamic-key trees. The original scheme in [14] is to replace the current codeword content with any codeword content based on session key to obtain the encrypted image, whereas the proposed method is to substitute the current codeword index with any codeword index based on the sorted codebook and session key to produce the encrypted indices, which makes full use of the characteristic of VQ image to lower the bandwidth and cost in the transmitter side efficiently. Then, in the receiver side, the encrypted indices can be divided into nonoverlapping blocks of 2^n codeword indices and the binary value of a fingerprint sequence $(F_1, F_2, \dots, F_{n+1})_2$ is in the range of $[0, 2^{n+1}-1]$. Assume that the codeword indices in arbitrary one block are $c_1, c_2, \dots, c_{2^n}$, and the weight-sum can be defined as follows:

$$f = \left(\sum_{i=1}^{2^n} i \times c_i \right) \bmod 2^{n+1}, \quad (2)$$

where c_i denotes the i th codeword index in one block. As mentioned above, the reason why the weight-sum function is required is that it keeps the capability to find the position of modifiable codeword index in one block, and the value of the modified block computed by the weight-sum function is completely equal to the binary value of a fingerprint subsequence with $n+1$ bits, which is vital for the fingerprint extraction process since the generated value by weight-sum function is the binary value of the required fingerprint subsequence.

FIGURE 2: The framework of the proposed JFD scheme.

FIGURE 3: The whole process of block-based JFD scheme.

Similarly, suppose both encrypted VQ indices and $n=2$ are known, and the size of codebook is set as 512. The encrypted VQ indices are divided into nonoverlapping blocks with 2×2 codeword indices, and then one of the nonoverlapping blocks is decrypted to embed a fingerprint subsequence. Assume that the indices of codeword in one block are 245, 342, 311, and 187. Here, notice that the value of codeword indices depends on the size of the given codebook, which is dissimilar to the gray value of digital images. Accordingly, let the fingerprint sequence be $B = [101]_2$, and then the binary value of fingerprint B is 5. The value of weight-sum function can be easily computed as $f = (1 \times 245 + 2 \times 342 + 3 \times 311 + 4 \times 187) \bmod 2^3 = 2$. Clearly, the value of fingerprint is unequal to that of the weight-sum function; i.e., $5 \neq 2$. Thus, we need to find the changeable codeword index in the block by computing the difference $d=5-2=3$ between B and f . Since the value of d

is in the range of $[0, 2^3]$, the codeword index to be modified can be found and adding 1 to the index can accomplish the modification operation; namely, the index value of the fingerprinted codeword is $311+1=312$. The codeword indices in the fingerprinted decryption block are correspondingly 245, 342, 312, and 187. The detailed process of the block-wise JFD scheme is presented in the following.

Step 1. Obtain the encrypted indices together with the codebook size N_c and divide the encrypted indices into nonoverlapping blocks with the size of 2^n codeword indices, and let $c_1, c_2, \dots, c_{2^n}$ be the codeword indices in one block and c_i be in the range of $[0, N_c]$.

Step 2. Set all divided blocks as unprocessed encrypted index-based blocks.

FIGURE 4: The flow diagram of fingerprint extraction process.

Step 3. Generate the fingerprint subsequence $B = (b_1, b_2, \dots, b_{n+1})_2$ to be embedded into one block, and compute the value of block with the weight-sum function $f = (\sum_{i=1}^{2^n} i \times c_i) \bmod 2^{n+1}$.

Step 4. If $(b_1, b_2, \dots, b_{n+1})_2$ equals $f = (\sum_{i=1}^{2^n} i \times c_i) \bmod 2^{n+1}$, do nothing and go to Step 6. Else, compute $d = (b_1, b_2, \dots, b_{n+1})_2 - f$ and go to Step 5.

Step 5. If $0 < |d| < 2^n$, positive d implies the modifiable codeword index as $c_d + 1$ and negative d implies the modifiable codeword index as $c_{|d|} - 1$. Else, positive d implies the modifiable codeword index as $c_{|d-2^{n+1}|} - 1$ and negative d implies the modifiable codeword index as $c_{d+2^{n+1}} + 1$.

Step 6. Check whether there exist unprocessed encrypted index-based blocks. If they exist, set the current block as processed block, and go to Step 4.

3.3. Block-Wise Fingerprint Extraction Process. When illegally redistributing the fingerprinted decryption content via the Internet, the effective way to find the traitor is to examine the digital content with proper fingerprint extraction algorithm. It should be interpreted that since the inverse permutation process mentioned in Figure 2 is not the core in this section, we directly illustrate the fingerprint extraction process instead. Figure 4 illustrates the flow diagram of fingerprint extraction process. From Figure 4 one can know that the fingerprinted decryption image is first divided into N blocks with the help of the known codebook and each block contains 2^n codeword indices, and a user's fingerprint sequence will be extracted from 1 to N blocks in turn by exploiting the weight-sum function. In each block, the value computed by the weight-sum is the binary value of fingerprint subsequence, and thus the fingerprint extraction algorithm is a blind extraction algorithm and not required for the original VQ image. After extracting the fingerprinting subsequence in all blocks, a user's fingerprinting sequence can be easily reconstructed.

The simple extraction steps are depicted as follows.

Step 1. Divide the fingerprinted and decrypted image into N blocks containing 2^n codeword indices.

Step 2. Compute the binary value of fingerprinting subsequence from each block with weight-sum function.

Step 3. Reconstruct the extracted fingerprinting subsequence in turn to generate a user's fingerprint sequence.

Since the fingerprinting extraction method in the proposed JFD scheme is implemented in the receiver side, i.e., one piece of multimedia data may have many various versions to stand for different users' identities, the correlation between the extracted fingerprint F_k and someone's fingerprint F_l is required to confirm the traitor's identity, which can be denoted as $\langle F_k, F_l \rangle$ shown in

$$\langle F_k, F_l \rangle = \frac{\sum_{j=0}^{m-1} f_{j,k} f_{j,l}}{\sum_{j=0}^{m-1} (f_{j,l})^2}, \quad (3)$$

where m denotes the length of fingerprint sequence. The relationship between the generated $\langle F_k, F_l \rangle$ and a threshold T can be used to judge whether someone is the traitor or not, which is shown in

$$\begin{aligned} \langle F_k, F_l \rangle &\geq T, & \text{the owner of } F_l \text{ is a traitor.} \\ \langle F_k, F_l \rangle &< T, & \text{the owner of } F_l \text{ is not a traitor.} \end{aligned} \quad (4)$$

4. Experiment Results and Analyses

In this section, we offer experimental results to confirm the effectiveness of the improved JFD scheme.

Similarly, six standard gray-scale images sized 512×512 pixels, including Lena, Plane, Baboon, Boat, Pepper, and Barbara images, are tested with a series of experiments. The VQ images trained by standard gray-scale images are treated as the original images shown in Figure 5. During the training process, the original image is divided into 4×4 blocks and the size of codebook is selected as 256, 512, and 1024, respectively. The training algorithm of [23] is also used for the proposed VQ process

FIGURE 5: Six 512×512 -pixel standard gray-scale VQ images using L Codebooks. (a) Original Lena image (L=256, PSNR=31.36dB), (b) original Lena image (L=512, PSNR=32.24dB), (c) original Lena image (L=1024, PSNR=33.20dB), (d) original Plane image (L=256, PSNR=30.57dB), (e) original Plane image (L=512, PSNR=31.57dB), (f) original Plane image (L=1024, PSNR=32.23dB), (g) Original Baboon image (L=256, PSNR=24.37dB), (h) original Baboon image (L=512, PSNR=24.70dB), (i) original Baboon image (L=1024, PSNR=25.03dB), (j) Original Boat image (L=256, PSNR=29.38dB), (k) original Boat image (L=512, PSNR=30.15dB), (l) original Boat image (L=1024, PSNR=30.88dB), (m) original Pepper image (L=256, PSNR=30.72dB), (n) original Pepper image (L=512, PSNR=31.40dB), (o) original Pepper image (L=1024, PSNR=32.13dB), (p) original Barbara image (L=256, PSNR=25.80dB), (q) original Barbara image (L=512, PSNR=26.38dB), and (r) original Barbara image (L=1024, PSNR=27.05dB).

4.1. The Encryption Results. As previously mentioned, our proposed encryption method is to encrypt image contents into codeword indices to improve transmission efficiency in the transmitter side. When the encrypted indices are intercepted, even if the codebook opened can decode the encrypted indices into image contents, the hacker still cannot know any useful information related to the original image, the property of which implies the security of encryption is desirable. In this article, the size of codebook is set as 256 and the decoded encrypted indices are shown in Figure 6. Clearly, the PSNRs of directly decoded encrypted

indices demonstrate that the proposed encryption method is acceptable. For the customers in the receiver side, they can easily transform the encrypted indices into decrypted VQ image contents with the help of the known codebook and dynamic-key trees together with permutation key.

4.2. Imperceptibility of the Fingerprint. In a joint fingerprinting and decryption process, a receiver is supposed to require both dynamic-key and fingerprint sequence and then divide the obtained indices into blocks and embed fingerprint subsequence into each block during the decryption process.

FIGURE 6: Encrypted results under both permutation and dynamic-key trees. (a) Encrypted Lena image (PSNR= 9.57 dB), (b) encrypted Plane image (PSNR=8.68 dB), (c) encrypted Baboon image (PSNR= 9.85 dB), (d) encrypted Boat image (PSNR= 8.97 dB), (e) encrypted Pepper image (PSNR= 8.90 dB), and (f) encrypted Barbara image (PSNR=8.83 dB).

In other words, when completing one block decryption, a given fingerprint subsequence is simultaneously embedded into the decrypted block. Figure 7 shows the visual quality of fingerprinted VQ image. The fingerprint bits embedded under various codebook size for various images are invisible to the naked eye. In addition, with the increase of the codebook size, the quality of the fingerprinted image (PSNR) correspondingly increases. Thus, a larger codebook size will generate a higher quality of fingerprinted image. When the size of the codebook is 1024, the fingerprinted image will have the best image quality.

4.3. Different Sizes of Block. In this section, the relationship between the number of modification blocks and the embedding performance is further analyzed to present that the size of block has great influence on the embedding performance and the modifiable block number. It should be emphasized that, in this article, two dividing processes are mentioned, namely, pixel-based blocks for VQ process and index-based blocks for JFD process. For VQ process, as mentioned before, the original image is divided into 4×4 -dimension vector, and thus a 512×512 -pixel image contains 16384 pixel-based blocks. As for JFD process, the divided pixel-based blocks in VQ process had been substituted with code indices, and the code indices needed to further divide into index-based blocks to be embedded fingerprint. For instance, when containing one index in an index-based block, the maximum embeddable capacity of fingerprint sequence is 16384 bits for a 512×512 -pixel image. Here, the block mentioned in this section refers to the index-based blocks, and the relationship of the modified block number under different block sizes and

embedding rate is presented in Figure 8. When one block to be modified contains a codeword index, our proposed method is completely equal to the embedding scheme in [14], the maximum modified block number under the worst case is 16384, and the PSNRs of fingerprinted image under this situation are less desirable. For one block including 2 codeword indices, obviously, the modified block number is decreased by half; in other words, the PSNRs of fingerprinted image dramatically increases while the maximum embedding rate is equal to that in [14]. Besides, with the increase of the size of block, the modified block number apparently lowers and thus the PSNRs correspondingly enlarges. To realize a tradeoff between the embedding rate and the modified blocks, we suggest the block size is set as 4.

4.4. The Security. In terms of the security of encryption, it can be classified into two processes: one is that a given original image is first converted into index matrix under the VQ process and then a permutation process is performed on the generated index matrix to enhance the security during transmission, and the other is that the dynamic-key trees based on session key are exploited in the code substitution process, since the session key only exists in a limited time and the security performance is further enhanced. Therefore, an attacker that intercepts the encrypted indices from the Internet cannot obtain any useful information related to the original encrypted indices.

4.5. Comparison. In this section, the comparison results with the proposed method and the scheme [14] are further discussed.

FIGURE 7: Fingerprinted decryption images using different L Codebooks. (a) Fingerprinted Lena image (L=256, PSNR=30.06dB), (b) fingerprinted Lena image (L=512, PSNR=31.23dB), (c) fingerprinted Lena image (L=1024, PSNR=32.17dB), (d) fingerprinted Plane image (L=256, PSNR=29.69dB), (e) fingerprinted Plane image (L=512, PSNR=30.43dB), (f) fingerprinted Plane image (L=1024, PSNR=31.16dB), (g) fingerprinted Baboon image (L=256, PSNR=23.91dB), (h) fingerprinted Baboon image (L=512, PSNR=24.26dB), (i) fingerprinted Baboon image (L=1024, PSNR=24.57dB), (j) fingerprinted Boat image (L=256, PSNR=28.41dB), (k) fingerprinted Boat image (L=512, PSNR=39.13dB), (l) fingerprinted Boat image (L=1024, PSNR=30.05dB), (m) fingerprinted Pepper image (L=256, PSNR=29.63dB), (n) fingerprinted Pepper image (L=512, PSNR=30.24dB), (o) fingerprinted Pepper image (L=1024, PSNR=31.15dB), (p) fingerprinted Barbara image (L=256, PSNR=25.12dB), (q) fingerprinted Barbara image (L=512, PSNR=25.67dB), and (r) fingerprinted Barbara image (L=1024, PSNR=26.21dB).

4.5.1. Compressing Encryption. The main task in the sender side is to accomplish the encryption for VQ images. In the original scheme, the VQ images are encrypted with codeword substitution, which is based on codeword contents and then transmitted into the receiver side. On the contrary, for the proposed encryption scheme, a given codebook is first sorted and then the codeword indices in the sorted codebook are employed in the encrypted process; i.e., the proposed encryption process is to substitute the current block in the original image with any index depending on dynamic-key trees in the sorted codebook. In this way, the encryption result is to generate encrypted indices which is different from that in

the original scheme. Obviously, with the proposed encryption method, the bandwidth and cost for transmission process are decreased significantly compared with the original scheme.

4.5.2. Blind Extraction of Fingerprint. Once illegal multimedia contents, such as illegal image-based contents, occurred via the Internet, one can easily know the traitor by extracting the embedded fingerprint sequence. For the original scheme, the extraction of a fingerprint sequence is accomplished with the guidance of the original VQ image. Conversely, the extraction of fingerprint sequence in our method is a blind extraction process. Specifically, the fingerprinted image

TABLE 1: PSNRs under various L with method in [14] and the proposed method.

Size	images	PSNRs (dB)					
		Lena	Plane	Baboon	Boat	Pepper	Barbara
L=256	Fingerprinted [14]	29.96	29.23	23.73	27.95	29.17	24.82
	Fingerprinted (Proposed)	30.60	29.69	23.91	28.41	29.63	25.12
	Original VQ	31.36	30.57	24.37	29.38	30.72	25.80
L=512	Fingerprinted [14]	30.81	30.11	23.99	28.78	29.93	25.31
	Fingerprinted (Proposed)	31.23	30.43	24.26	29.13	30.24	25.67
	Original VQ	32.24	31.57	24.70	30.15	31.40	26.38
L=1024	Fingerprinted [14]	31.67	30.72	24.32	29.55	30.83	25.97
	Fingerprinted (Proposed)	32.17	31.16	24.57	30.05	31.15	26.21
	Original VQ	33.20	32.23	25.03	30.88	32.13	27.05

FIGURE 8: Modified block number versus embedding rate under different block size.

is first divided into blocks with 2^n codeword indices under the known codebook and the value of each block can be computed by the weight-sum function. As a result, the generated values can reconstruct the fingerprint sequence. Obviously, the fingerprint extraction is a blind extraction and not required for the original VQ image.

4.5.3. Changeable Maximum Capacity of Fingerprint. Although the original JFD method based on VQ can embed fingerprint into each block during the decryption process, there exists a major concern; i.e., the embedding capacity is constant. For instance, when the block size is set as 4×4 , a 512×512 -pixel image contains 16384 blocks, and thus the embedding capacity of fingerprint is 16384 bits.

This is quite inflexible and may limit the number of users due to the fact that various users have different fingerprint lengths. However, this concern can be effectively resolved by index-based fingerprint embedding because it offers a variety of options for fingerprint embedding. Thus, the proposed method is more applicable for practical scenarios.

4.5.4. PSNR versus Embedding Rate. In this module, PSNRs of fingerprinted decryption images under the embedding rate 0.75 bpp (bits per pixel) will be further compared with the original JFD scheme [14]. Note that since the original VQ images themselves are lossless compressed images, the performance improved is seriously limited by the original VQ image. To clearly present the improved performance, the PSNRs in the original VQ images are also presented in the comparison as shown in Table 1. From Table 1, we can know that the VQ images are inevitably distorted compared with the original standard images, and thus the PSNRs of fingerprinted decryption images based on VQ images must be lower than that in the original VQ images. In other words, it is impossible for the PSNR improvement of fingerprinted VQ image to realize the maximum results of original VQ images in Table 1. Clearly, the PSNR comparison results between method in [14] and the proposed method can easily be obtained to show that, under the same embedding rate, the PSNR of our method is higher than that in the original scheme [14]. Here, it should be emphasized that, in our method, the embedding rate 0.75 is the result of one block with 2×2 codeword indices, but the PSNRs of fingerprinted images under the embedding rate are not optimal. The relationship between the modified blocks and embedding rate can be formulized as shown in

$$y = \frac{Np}{(w \times h)(n+1)}x, \quad 0 < x \leq \frac{n+1}{2^n} \quad (5)$$

TABLE 2: Comparisons of the functionalities of our method and related mechanisms.

Functionality	Blind extraction	Transmission cost	The flexibility of maximum capacity of fingerprint
[15]	Yes	Higher	Yes
[16]	No	Higher	Yes
[17]	No	Higher	Yes
[18]	Yes	Higher	No
Our scheme	Yes	lower	Yes

where x and y denote the embedding rate and the modified blocks, respectively. $(n+1)/2^n$ denotes the maximum embedding rate. Clearly, N_p and $w \times h$ are constant values; under a given embedding rate x , the modified block number decreases with the increase of n . This also accounts for the reason why the block size with 2×2 codeword indices, i.e., $n=2$, is not an optimal case for the PSNRs of fingerprinted image. Besides, take an obvious comparison as an example. With the method in [14], embedding $(n+1)$ fingerprint bits requires modifying $(n+1)/2$ codeword indices averagely. For our method, however, only one codeword index is modified when the block size is set as 2^n . To sum up, our method can effectively improve the PSNRs of fingerprinted image

4.5.5. Comparisons with Related Mechanisms. The functionalities of the proposed method are further discussed and compared with mechanisms [15–18]. Table 2 illustrates the comparisons of our method and related mechanisms. Although the improved functionality of [18] achieves a blind fingerprint extraction, the computation and communication overheads are still higher, and the total amount of embedded fingerprint is still constant and inflexible due to the fact that the total amount of the fingerprint can only be equal to the block size of the specified VQ image. The performances of [15–17] are less satisfactory in terms of transmission cost. In addition, schemes [16, 17] cannot achieve a blind fingerprint extraction. The reason why our scheme has a lower transmission cost is that the original content under VQ encoding is encrypted into codeword indices and the encrypted indices are transmitted in the receiver side, but all of the others encrypt it into noise-like image content. After considering the global functionality of various mechanisms, the improved JFD based on VQ method is superior.

5. Conclusion

This paper proposes a block-based improved JFD scheme with the help of the method in [14]. On the sender side, we encrypt the original image into codeword indices to effectively decrease the transmission bandwidth and cost. For the JFD process in the receiver side, the encrypted indices are divided into blocks with 2^n codeword indices and a weight-sum function is utilized to embed $(n+1)$ bits fingerprint subsequence into each block, and the PSNRs of

fingerprinted images are improved compared with original scheme. As for the fingerprint extraction process, the values of the divided blocks computed by the weight-sum function are the binary value of fingerprint sequence, and thus the fingerprint extraction is a blind process without the original VQ image. The experiment results demonstrate that the proposed method can effectively improve the method in [14], especially when the block size is smaller.

Data Availability

All data generated or analyzed during this study are included in this article.

Disclosure

An earlier version of this paper was presented at the National Information Hiding Workshop (CIHW2018, <http://www.cihw.org.cn/index.php/4/>).

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 (Grant nos. 61602158, 61572089, U1604156, and 61772176), the Science Foundation for the Excellent Youth Scholars of Henan Normal University (Grant no. YQ201607), the Production and Learning Cooperation and Cooperative Education Project of Ministry of Education of China (Grant no. 201702115008), the Fundamental Research Funds for the Central Universities (Grant nos. 106112017CDJQJ188830, 106112017CDJXY180005), the Chongqing Research Program of Basic Research and Frontier Technology (Grant nos. cstc2017jcyjBX0008, cstc2014jcyjA40030), Science and Technology Research Project of Henan Province (Grant nos. 182102210374, 172102210045, and 182102210362), the Natural Science Foundation of Henan province (Grant no. 182300410368), the Plan for Scientific Innovation Talent of Henan Province (Grant no. 184100510003), and the Young Scholar Program of Henan Province (Grant no. 2017GGJS041).

References

- [1] J. A. Bloom, I. J. Cox, T. Kalker, Linnartz M. G., M. L. Miller, and C. Brendan Traw, "Copy protection for DVD video," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1267–1276, 1999.
- [2] D. Kundur and K. Karthik, "Video fingerprinting and encryption principles for digital rights management," *Proceedings of the IEEE*, vol. 92, no. 6, pp. 918–932, 2004.
- [3] L. Wang, H. Ling, H. Zou, P. Li, and S. Wang, "Robust embedding and detection for multimedia fingerprinting," in *Proceedings of the IEEE, International Conference on Signal Processing*, pp. 1857–1860, 2010.
- [4] K. H. Rhee, "Detection on the falsifier of multimedia fingerprinting code using PCA method," in *Proceedings of the 7th International Conference on Digital Content, Multimedia Technology and Its Applications, IDCTA 2011*, vol. 3, pp. 173–176, August 2011.
- [5] Y. Wang and A. Pearmain, "Blind image data hiding based on self reference," *Pattern Recognition Letters*, vol. 25, no. 15, pp. 1681–1689, 2004.
- [6] C.-C. Chang, Y.-S. Hu, and T.-C. Lu, "A watermarking-based image ownership and tampering authentication scheme," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 439–446, 2006.
- [7] V. Mehan, R. Dhira, and Y. S. Brar, "Joint watermarking and fingerprinting approach for colored digital images in double DCT domain," in *Proceedings of the 2013 IEEE International Conference on Signal Processing, Computing and Control, ISPC 2013*, pp. 1–6, September 2013.
- [8] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1673–1687, 1997.
- [9] A. Mishra, A. Goel, R. Singh, G. Chetty, and L. Singh, "A novel image watermarking scheme using Extreme Learning Machine," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 20, pp. 1–6, 2012.
- [10] P. Judge and M. Ammar, *WHIM: watermarking multicast video with a hierarchy of intermediaries*, Elsevier North-Holland, Inc, 2002.
- [11] B. M. Macq and J.-J. Quisquater, "Cryptology for Digital TV Broadcasting," *Proceedings of the IEEE*, vol. 83, no. 6, pp. 944–957, 1995.
- [12] B. Czapslewski and R. Rykaczewski, *Improvement of Fingerprinting Method based on Hill Cipher by using Frequency Domain*, ICT Young, 2012.
- [13] B. Czapslewski and R. Rykaczewski, Immunizing the hillcast method against the known-plaintext attack using periodic key exchange, *Telecommunication Review + Telecommunication News // Przegląd Telekomunikacyjny + Wiadomości Telekomunikacyjne*, no. 8-9, pp. 830–835, 2013.
- [14] C.-Y. Lin, P. Prangjarote, L.-W. Kang, W.-L. Huang, and T.-H. Chen, "Joint fingerprinting and decryption with noise-resistant for vector quantization images," *Signal Processing*, vol. 92, no. 9, pp. 2159–2171, 2012.
- [15] C. Ye, Z. Xiong, Y. Ding, G. Wang, J. Li, and K. Zhang, "Joint fingerprinting and encryption in hybrid domains for multimedia sharing in social networks," *Journal of Visual Languages and Computing*, vol. 25, no. 6, pp. 658–666, 2014.
- [16] B. Czapslewski, "Joint fingerprinting and decryption method for color images based on quaternion rotation with cipher quaternion chaining," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 1–13, 2016.
- [17] B. Czapslewski and R. Rykaczewski, "Matrix-based robust joint fingerprinting and decryption method for multicast distribution of multimedia," *Signal Processing*, vol. 111, pp. 150–164, 2015.
- [18] M. Li, D. Xiao, Y. Zhang, and H. Liu, "Attack and improvement of the joint fingerprinting and decryption method for vector quantization images," *Signal Processing*, vol. 99, no. 6, pp. 17–28, 2014.
- [19] P. Y. Lin, J. S. Lee, and C. C. Chang, "Protecting the content integrity of digital imagery with fidelity preservation," *Acm Transactions on Multimedia Computing Communications & Applications*, vol. 7, no. 3, pp. 1–20, 2011.
- [20] A. Hyvärinen, J. Karhunen, and E. Oja, "Chapter 4. Estimation Theory," in *Independent Component Analysis*, John Wiley & Sons, Inc., 2002.
- [21] R. C. T. Lee, Y. H. Chin, and S. C. Chang, "Application of principal component analysis to multikey searching," *IEEE Transactions on Software Engineering*, vol. SE-2, no. 3, pp. 185–193, 1976.
- [22] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781–783, 2006.
- [23] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.