Joint fingerprinting and decryption with noise-resistant for vector quantization images

Chih-Yang Lin, Panyaporn Prangjarote, Li-Wei Kang, Wei-Lun Huang, Tzung-Her Chen

Department of Computer Science & Information Engineering
Asia University, Taichung, Taiwan, ROC
Institute of Information Science
Academia Sinica, Taipei, Taiwan, ROC
Department of Computer Science & Information Engineering
National Chiayi University, Chiayi, Taiwan, ROC

Article info

Article history:
Received 27 September 2011
Received in revised form 21 December 2011
Accepted 6 February 2012
Available online 15 February 2012

Keywords:
Vector quantization
Joint fingerprinting and decryption
Traitor tracing

Abstract

With the popularity of the Internet and development of multimedia technology, media distribution and traitor tracing issues have become critical and urgent. In this paper, a joint fingerprinting and decryption (JFD) scheme based on vector quantization is proposed with the purpose of protecting media distribution. The proposed JFD scheme is equipped with two encryption techniques, which are performed on the server side. The first technique encrypts plain-images using static key-trees, but the second approach uses dynamic key-trees to further simplify the first method. When the subscriber receives the encrypted images, these images are jointly decrypted and fingerprinted and are slightly different from the original images. The proposed method, to the best of our knowledge, is the first JFD method for vector quantization (VQ) compressed images. The experimental results show that the encrypted image is unintelligible and that the recovered image has desirable image quality resistant to noise interference.

1. Introduction

As a consequence of ongoing developments in multimedia techniques and communication technology, multimedia data is prevalent in our daily lives. Thus, protecting such data from various attacks has become increasingly pressing and necessary. The properties of multimedia data that need to be protected are those confidentiality, integrity, and ownership. Confidentiality can be achieved through multimedia encryption methods [1–4], which transform intelligible multimedia data into unintelligible content. Only the user who possesses the secret key can decrypt it. Integrity and ownership protection can be preserved through digital watermarking methods [5–8], which embed authentication codes or watermarks in the original data. Only the rightful owner can successfully extract the embedded authentication code or watermark to verify the integrity or copyright authenticity.

However, protecting confidentiality, integrity, and ownership is insufficient for multimedia distribution. This is because the decrypted multimedia data may be redistributed from the authorized customer to unauthorized users. For example, a user may legally subscribe to a video but illegally distribute it to other users without the owner’s permission. The problem of finding the “traitor” is referred to as the traitor tracing problem. To solve this problem, digital fingerprinting methods, in which all the fingerprinted copies of the multimedia data are different but appear to have the same visual perception, have been intensively investigated [9–13]. A fingerprinted copy embeds a fingerprint in the original data to verify the user’s identity. Therefore, if a fingerprinted copy is illegally
redistributed, the fingerprint can be extracted and then be used to trace the traitor. Although the approach of embedding and extracting fingerprints is similar to that of watermarking, the goals of each method are quite different. Fingerprinting is aimed at traitor tracing, but watermarking is directed towards copyright protection. Generally, watermarking is a sender-side system, which embeds a unique watermark in all kinds of multimedia data. Therefore, for a specific piece of multimedia data, only one watermarked version is produced. On the contrary, fingerprinting is a receiver-side system since one piece of multimedia data will have many different versions to represent different users’ identities. If fingerprinting were implemented on the sender side, the cost of computations and bandwidth on the server would be unmanageable when the number of receivers greatly increases.

The first receiver-side fingerprinting [12] to perform decryption and fingerprinting independently, is shown in Fig. 1. The decrypted media content, however, could be intercepted in the gap between the decryption and fingerprinting operations. Therefore, a joint fingerprinting and decryption scheme (JFD) [11,14,15] was proposed to solve the content leakage problem. In the JFD scheme shown in Fig. 2, the decryption and fingerprinting processes work simultaneously and eliminate the loophole in order to prevent an attacker from obtaining the fingerprint-free decrypted version. Following this idea, Anderson and Manifavas [14] proposed the Chameleon method based on table lookup operations, which provided good imperceptibility for the fingerprinted data. The media data is decrypted using different secure tables for different users; since each user needs a different secure table, this method may consume greater bandwidth [16]. Kundur and Karthik [11] then proposed a JFD scheme based on partial decryption, which encrypts signs of discrete cosine transform (DCT) coefficients of the media on the sender side and then decrypts them partially according to the user’s fingerprint on the receiver side. The scheme provided a good framework for JFD, but the encrypted media data is not secure in visual perception since the encryption of signs of DCT coefficients cannot fully scramble the encrypted media.

Lemma et al. [15] also proposed a JFD method based on homomorphic operations. The media data achieves encryption and decryption by means of different key streams. However, each key stream is the same size as the media, leading to prohibitive transmission costs. A generalized version of Lemma et al.’s method has been proposed by Celik et al. [17], which not only reduces transmission costs but also provides an efficient fingerprint detection method and rigorous security proofs. Although the above JFD methods meet the requirements of protecting media distribution, they are performed on either the transform domain or pixel domain, and none of them can be applied to the vector quantization (VQ) domain.

In this paper, a JFD scheme to protect image distribution with traitor tracing based on vector quantization is proposed. Vector quantization (VQ) was originally used for image compression, but it has become widely used for many applications with similar concepts, such as in the “bag of word” (BoG) in computer vision or sound recognition. The main reason why we propose the JFD based on VQ is that VQ is not only suitable for a variety of applications, but is also particularly strong in fault tolerance, such as noise interference. This distinguishes it from the ideas used in the previous JFD methods. The proposed first JFD method applied to VQ compressed images encrypts images based on permutation and codeword substitution using static key-trees on the sender side.
while the second proposed JFD method uses a dynamic key-trees approach to replace permutation, which is more flexible but requires more information (a session key). The operations of decryption and fingerprinting on the receiver side recover the encrypted image according to the set of pre-designed key-trees. The set of elements of key-trees is unique (i.e., codebook) and can save significant bandwidth and conveniently update key-trees. The experimental results show that the encrypted images are unintelligible in visual perception and the fingerprinted images possess desirable visual quality.

The rest of this paper is organized as follows. In Section 2, techniques used in this paper will be introduced. Section 3 details the proposed JFD scheme based on vector quantization with two encryption methods. Then, the experimental results will be given in Section 4. Finally, conclusions are drawn in Section 5.

2. Review of techniques used

2.1. Vector quantization

Vector quantization (VQ) [18,19] is a classical lossy compression technique with simple and efficient encoding and decoding procedures, which is especially designed for digital images. Fig. 3 shows the flow chart of the VQ encoding and decoding process.

In the encoding process, the original image is first divided into several non-overlapping blocks of \( w \times h \) pixels. That is, each block can be denoted by a \( w \times h \)-dimensional vector. The main concept of VQ is to map each image block using a mapping function \( G \) from \( w \times h \)-dimensional Euclidean space \( R^w \times h \) to a finite subset \( CB \) of \( R^w \times h \), i.e., \( G: R^w \times h \rightarrow CB \). Where \( CB = \{ Y_1, Y_2, \ldots, Y_L \} \) is called the codebook and \( Y_i \) is the \( i \)-th codeword in \( CB \) and \( L \) refers to the size of the codebook. Each codeword \( Y_i \) corresponds to \( w \times h \) elements; that is, \( Y_i = \{y_{1,j}, y_{2,j}, \ldots, y_{w \times h,j} \} \).

For each vector \( X \in R^w \times h \) of the original image, the nearest codeword in the codebook can be found according to the distance between \( X \) and a codeword \( Y_i, i=1, 2, \ldots, L \). The distance is determined by the following Euclidean distance, \( d(X, Y_i) \):

\[
d(X, Y_i) = \| X - Y_i \| = \left( \sum_{j=1}^{w \times h} (x_{j} - y_{ij})^2 \right)^{1/2}.
\]

where \( x_{j} \) and \( y_{ij} \) represents the \( j \)-th elements of vectors \( X \) and \( Y_i \), respectively. When the nearest codeword \( Y_i \) of \( X \) is found, index \( i \) is used to encode vector \( X \). After all \( X \)'s are encoded, the original image can be eventually represented by indices of these nearest codewords.

In the decoding process, the image is reconstructed by table lookup operations. The same codebook that is used by the VQ encoder is also required by the VQ decoder. According to the indices represented by the VQ encoder, the VQ decoder uses the corresponding codewords to reconstruct the image. Therefore, the quality of the VQ-compressed image is significantly influenced by the quality of the codebook, which can be well-designed by [19,20].

2.2. The strength secure session key for media distribution

In a symmetric encryption scenario as shown in Fig. 4, the sender and receiver must share the same key for communication or transmission, and that key must be protected from access by others. A permanent key, which may be used without alterations for years, is convenient for users but is easily exposed to attackers, who can learn the key. Session keys can solve this problem.

Unlike typical keys or master keys, a session key is created to exist for a limited period of time. As shown in Fig. 5, the sender encrypts the original data with the session key and then encrypts the session key with the
master key. The sender sends the encrypted data and the encrypted session key to the recipient.

The decryption process works in reverse. The receiver uses the master key to recover the temporary session key, and then uses it to decrypt the encrypted data. Therefore, the session key distribution technique is suitable for image protection in a multimedia distribution system.

3. The proposed method

The first proposed JFD scheme based on VQ for image distribution is shown in Fig. 6. On the sender end, key-trees, also called static key-trees, should be first built for encryption and decryption. After the key-trees are constructed, these trees are also stored on the receiver side. When a VQ image $P$ is subscribed to, the image goes through a permutation process and is then encrypted by codeword substitution according to the set of key-trees. The dotted rectangle on the permutation in Fig. 6 means that the permutation process only enhances the unintelligibility of the encrypted image and is optional for the proposed method. In other words, without permutation, the attacker still cannot break the whole system. Such a case will be discussed in detail in Sections 3.4 and 4. In addition, in Section 3.4, an alternative JFD method will be proposed, in which a dynamic key-tree approach will be applied to replace the permutation process. In the decryption process, when the subscriber receives the encrypted image $C$, the image is recovered through the proposed joint decryption and fingerprinting method according to the static key-tree or dynamic key-tree approach to obtain the fingerprinted copy. The details are presented in the following subsections.
3.1. Sorting and declustering the codebook

The codebook $CB$ should be sorted prior to encryption. Assume that the codewords of the unsorted $CB$ are $W_1, W_2, \ldots, W_L$, and after sorting, the codewords of the sorted codebook $CB_0$ will become $Y_1, Y_2, \ldots, Y_L$. Then, $CB_0$ with size $L$ is partitioned into two groups $G_0$ and $G_1$ according to $CB_0 = G_0 \subseteq G_1$, $G_0 \cap G_1 = \emptyset$, and $|G_0| = |G_1|$, where the relationship between $G_0$ and $G_1$ is a one-to-one correspondence. In other words, any codeword $Y_i$ in $G_0$ must have one and only one corresponding codeword $Y_j$ in $G_1$ and vice versa. The Euclidean distance between each pair of $Y_i$ and $Y_j$ should be as large as possible. This can be achieved by applying principal components analysis (PCA) to the codebook. PCA is a widely used method in data and signal analysis [21–23], such as in multimedia coding and recognition. The power of PCA is in projecting a high-dimensional input vector onto a lower-dimensional space while still preserving the maximal variances of the input vectors on the new coordinate axes. In our scheme, each codeword with 16 dimensions is projected onto a 1-dimensional space to find its first principal component value, and these values are used to sort the codewords.

**PCA Algorithm.**

**Input:** The unsorted codebook $CB$ with codewords, $W_1, W_2, \ldots, W_L$.

**Output:** The sorted codebook $CB_0$ with codewords $Y_1, Y_2, \ldots, Y_L$.

**Step 1:** Compute the mean vector $m$ of the input vectors and normalize the vectors to be zero mean:

$$W_i' = W_i - m.$$

**Step 2:** Compute the covariance matrix $Cov$ of the normalized vectors.

**Step 3:** Find all the eigenvalues and eigenvectors of the covariance matrix $Cov$. Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues and $v_1, v_2, \ldots, v_n$ be the corresponding eigenvectors. The eigenvectors are sorted by the non-decreasing order of the corresponding eigenvalues.

**Step 4:** For each input vector $W_i$, its first principal component $c_{i1}$ can be computed by the following inner product:

$$c_{i1} = v_1^T W_i.$$

**Step 5:** The codewords $W_1, W_2, \ldots, W_L$ are sorted according to their $c_{i1}$’s in the nondecreasing order and then form the sorted codebook $CB_0$ with codewords $Y_1, Y_2, \ldots, Y_L$.

After applying PCA, since the neighboring codewords in $CB$ are similar and the difference between two codewords becomes more significant as the difference between their corresponding indices becomes larger. In such a way, $G_0$ and $G_1$ can be designed as:

$$G_0 = \{Y_0, Y_1, \ldots, Y_{L/2}\},$$

$$G_1 = \{Y_{L/2}, Y_{L/2+1}, \ldots, Y_{L-1}\}.$$

![Fig. 7. Decryption tree.](image)

![Fig. 8. Alternative JFD strategy.](image)
Therefore, \((Y_0, Y_{L/2}), (Y_1, Y_{L/2+1}), \ldots, (Y_{(L-2)/2}, Y_{L-1})\) constitute the \(L/2\) dissimilar pairs. This completes the declustering process.

3.2. Static key-trees construction

The structure of a static key-tree is shown in Fig. 7, where each node is regarded as a codeword. The parent node \(Y^c\) indicates the dissimilar codeword of the left-child node \(Y_i\), while the right-child node \(Y^0\) is the most similar codeword to the node \(Y_i\). For a given codebook containing \(L\) codewords, the index of the dissimilar codeword of \(Y_i\) in the codebook can be found according to the corresponding dissimilar pair \((Y_i, Y^c)\).

In the case of finding a similar pair of \((Y_i, Y^0)\), the most similar codeword to the node \(Y_i\) can be found through an exhaustive search of the codebook according to Eq. (1).

Note that for different codewords \(Y_i\) and \(Y_j\), the condition of \(Y^0_i \neq Y^0_j\) need not be satisfied since the fingerprint identification requires the original VQ image be involved. The dissimilar pairs are responsible for encryption to generate unintelligible images, but the similar pairs account for preserving the decrypted visual quality. Other efficient algorithms for similar and dissimilar pairing can be found in [24–26], respectively.

3.3. The encryption procedure on the sender side

The following details the steps of the encryption algorithm performed on the server end.

Step 1: Permute the VQ image \(P\) block by block using a chaotic map [3,27,28]. In this paper, Cat map [3] is

\[\text{Fig. 9. Six standard 512 \times 512-pixel grayscale VQ images using the codebook with L Codewords. (a) Original Lena image (L=256, PSNR=31.36 dB), (b) original Lena image (L=512, PSNR=32.24 dB), (c) original Lena image (L=1024, PSNR=33.20 dB), (d) original F16 image (L=256,PSNR=30.57 dB), (e) original F16 image (L=512,PSNR=31.57 dB), (f) original F16 image (L=1024, PSNR=30.88 dB), (g) original Baboon image (L=256, PSNR=24.37 dB), (h) original Baboon image (L=512, PSNR=24.70 dB), (i) original Baboon image (L=1024, PSNR=25.03 dB), (j) original Boat image (L=256, PSNR=29.38 dB), (k) original Boat image (L=512, PSNR=30.15 dB), (l) original Boat image (L=1024, PSNR=30.68 dB), (m) original Pepper image (L=256, PSNR=30.72 dB), (n) original Pepper image (L=512, PSNR=31.40 dB), (o) original Pepper image (L=1024, PSNR=32.13 dB), (p) original Barb image (L=256, PSNR=25.80 dB), (q) original Barb image (L=512, PSNR=26.38 dB), and (r) original Barb image (L=1024, PSNR=27.05 dB).}\]
employed as shown in Eq. (3).

\[
\begin{bmatrix}
  x_{i+1} \\
  y_{i+1}
\end{bmatrix} = \begin{bmatrix}
  1 & p \\
  q & p \times q + 1
\end{bmatrix} \begin{bmatrix}
  x_i \\
  y_i
\end{bmatrix} \left( \mod \frac{SP}{SB} \right).
\]

(3)

where \((x_i, y_i)\) is the block position in \(P\), \(p\) and \(q\) are control parameters for permutation, which are regarded as secrets, \(\mod\) stands for modulo operator, \(SP\) represents the size of images, and \(SB\) denotes the size of a block and is set to 4 in the proposed method, which means that the image is divided into blocks of sized \(4 \times 4\) pixels.

For each block, use Cat map to calculate the new position. After each block is relocated to a new position, the permuted image \(P_0\) is generated. However, the position of the \((0, 0)\) block is unchanged during the Cat map permutation and may lead to a security loophole. Therefore, it is necessary to exchange the \((0, 0)\) block with another one at the position \((r, s)\), where \(r\) and \(s\) are also considered as secrets.

Step 2: Replace each codeword \(Y_i\) with \(Y_i^c\) according to the set of static key-trees and produce the encrypted image \(C\) from \(P_0\).

After encryption, the attacker who intercepts the encrypted image from the Internet cannot decrypt the image or modify the embedded fingerprint since the secrets of the chaotic map and the static key-trees cannot be obtained.

3.4. Alternative strategy for JFD

In Fig. 6, the permutation process is used to enhance the security of the encrypted image. This process can be eliminated as shown in Fig. 8 when the session key and dynamic key-trees are applied. In the previous method, each codeword is statically associated with a dissimilar codeword and a most similar codeword. This approach will cause the encryption problem in which each identical image block will have the same encrypted block. Since natural images usually contain lots of smooth regions, such a problem will expose the rough sketch of an image (as shown in Fig. 13) and damage the confidentiality.

In view of this, a permutation process is required as shown in Fig. 6 in our previous method. However, we can eliminate this step using the proposed dynamic dissimilar pairs. In this approach, the codebook does not need to be partitioned into \(G_0\) and \(G_1\), and each codeword \(Y_i\) will not be statically associated with \(Y_i^c\). Instead, it is dynamically associated with a dissimilar codeword \(Y_{i(2^k+b)}\) where \(k\) is
designed as:

\[ k = H_{sk}(i) \mod (L/4). \]  

(4)

In Eq. (4), the session key \( sk \) will randomly generate an index \( k \), where \( 0 \leq k \leq [L/4] - 1 \). Therefore, the dissimilar codeword for \( Y \) would be \( Y_{[(L/2)+k+i]} \mod l \).

During the encryption process, when an image block \( Y \) requires encryption, the system dynamically generates \( Y_{[(L/2)+k+i]} \mod l \) based on \( sk \) to replace \( Y \). In this approach, since \( k \) is not constant and key-trees are not static, the encryption results for identical input blocks would not be the same. Note that, the generation of the most similar codeword of \( Y_i \) is the same as that described in 3.2. In other words, the similar pairs are all statically generated.

3.5. Joint decryption and fingerprinting process

We have proposed two methods for JFD, the former called the static key-trees approach and the latter called the dynamic key-trees approach. The following describes their two decryption processes.

Case 1. (Static key-trees approach):

In a joint fingerprinting and decryption process, a receiver requires two components, which are a set of key-trees and fingerprint to decrypt the encrypted image \( C \). Key trees can be pre-installed on the device; the fingerprint represented as \( F_k = f_{0,k}, f_{1,k}, \ldots, f_{m-1,k} \) where \( f_{i,k} = 0 \) or \( 1 \) and \( m = (SP/SB)^2 \) refers to the number of blocks of a subscribed image.

For each block of image \( C \), if \( f_{i,k} = 0 \), the left child \( Y_i \) as shown in Fig. 7 is used to decrypt the image block; otherwise, the right child \( Y_i \) is used.

Case 2. (Dynamic key-trees approach):

When a receiver gets the encrypted image \( C \) and the encrypted session key, the receiver decrypts the session key using the master key as shown in Fig. 5. In turn, for each index of \( Y_{[(L/2)+k+i]} \mod l \) (i.e., the codeword index of the encrypted image block), the receiver uses the session key to generate an index \( k \), and then the original index of the image block can be recovered as \( Y_i \). Therefore, the decryption process of the incoming block is to replace \( Y_{[(L/2)+k+i]} \mod l \) with \( Y_i \) if \( f_{i,k} = 0 \); otherwise, \( Y_i \) will be used.
3.6. Traitor tracing

Once the image owner detects that $P_k$ is received by an illegal user, the traitor will be revealed by the following steps.

**Step 1:** Perform the fingerprint extraction process. The fingerprint sequence $F_k$ can be extracted by the image owner according to the comparisons between $P_k$ and the original VQ image $P$. If a block of $P_k$ is equal to the corresponding block of $P$, a fingerprint bit 0 is extracted; otherwise, a bit 1 can be obtained.

**Step 2:** Evaluate the correlation between the extracted fingerprint $F_k$ and someone’s fingerprint $F_l$ denoted as $\langle F_k, F_l \rangle$ by:

$$\langle F_k, F_l \rangle = \left( \sum_{i=0}^{m-1} f_{ki} f_{li} \right) \bigg/ \left( \sum_{i=0}^{m-1} (f_{ki})^2 \right).$$  \hfill (5)

![Fig. 12.](image) Fingerprinted images using different sizes of codebooks. (a) Fingerprinted image ($L=256$, PSNR = 29.80 dB), (b) fingerprinted image ($L=512$, PSNR = 30.68 dB), (c) fingerprinted image ($L=1024$, PSNR = 31.59 dB), (d) fingerprinted image ($L=256$, PSNR = 29.06 dB), (e) fingerprinted image ($L=512$, PSNR = 29.98 dB), (f) fingerprinted image ($L=1024$, PSNR = 30.56 dB), (g) fingerprinted image ($L=256$, PSNR = 23.60 dB), (h) fingerprinted image ($L=512$, PSNR = 23.93 dB), (i) fingerprinted image ($L=1024$, PSNR = 24.24 dB), (j) fingerprinted image ($L=256$, PSNR = 27.89 dB), (k) fingerprinted image ($L=512$, PSNR = 28.69 dB), (l) fingerprinted image ($L=1024$, PSNR = 29.42 dB), (m) fingerprinted image ($L=256$, PSNR = 29.02 dB), (n) fingerprinted image ($L=512$, PSNR = 29.81 dB), (o) fingerprinted image ($L=1024$, PSNR = 30.76 dB), (p) fingerprinted image ($L=256$, PSNR = 24.77 dB), (q) fingerprinted image ($L=512$, PSNR = 25.23 dB), and (r) fingerprinted image ($L=1024$, PSNR = 25.84 dB).
Step 3: Discriminate whether \( F_j \) is the traitor according to \( \langle F_k, F_i \rangle \) compared with a threshold \( T \) by:

\[
\begin{align*}
&\langle F_k, F_i \rangle \geq T, \quad F_j \text{ is a traitor of } P_k, \\
&\langle F_k, F_i \rangle < T, \quad F_j \text{ is not a traitor of } P_k.
\end{align*}
\]

4. Experiments

In this section, we show some experimental results from our two proposed methods to demonstrate the effectiveness of our JFD scheme.

Six standard 512 × 512-pixel grayscale VQ images as shown in Fig. 9 (“Lena”, “F16”, “Baboon”, “Boat”, “Pepper”, and “Barb”) are used as the original images, which were divided into 16384 blocks of sized 4 × 4 pixels. The codebooks comprising 256, 512 and 1024 codewords of dimension 16 are, respectively, used in the experiments, which are trained by the LBG (Linde–Buzo–Gray) algorithm [19].

4.1. Perceptual security

Generally, the encrypted image should be unintelligible for confidentiality [29]. In the first proposed scheme, the image is encrypted by permutation via the chaotic map and the codeword substitution process. Fig. 10 shows the encryption results of six images using the codebook of sized 256. Furthermore, the encryption results using dynamic key-trees are shown in Fig. 11. It is clear that all the encrypted images become noise-like images and are all actually unintelligible.
The quality of an image is measured by the peak-signal-to-noise ratio (PSNR), which is defined as:

\[
\text{PSNR} = 10 \times \log_{10}(255^2 / \text{MSE}),
\]

where MSE denotes the mean square error between the original and the encrypted images. According to PSNR and visual quality, the proposed scheme indeed possesses high perceptual security.

4.2. Imperceptibility of the fingerprint

The fingerprint is embedded in the image during the joint decryption and fingerprinting process. In order to preserve visual quality, the fingerprint in the fingerprinted copy should be imperceptible and perceptually undetectable. Fig. 12 shows some experimental results of fingerprinted images. It can be observed that the quality of the fingerprinted image depends on the size of the codebook. In other words, the larger the size of the codebook, the higher will be the image quality of the fingerprinted image. When the size of the codebook is 1024, the fingerprinted images will have the best image quality.

4.3. Discussion of the permutation process

In Section 3 we mentioned that the permutation process in Fig. 6 only enhances the unintelligibility of the encrypted image and is optional for the proposed method. Therefore, even if the chaotic map used in permutation is cracked, the hacker still cannot decrypt the image since the key-tree remains secret. Fig. 13 shows the comparison of when a permutation process is and is not applied. It is clear that the permutation process in the proposed scheme can enhance perceptual security. Therefore, if confidentiality is in high demand, the proposed first method with permutation or the second approach can be applied. Otherwise, the first method without permutation can be performed since only a rough sketch without details would be revealed, making the perceptual quality unacceptable.

4.4. Analysis of noise resistance

The conventional JFD schemes have no capacity to resist noise interference that results from transmission error or man-made modifications. In other words, if
noise is added to an existing JFD method, the noise will still exist after decryption. The noise may damage some fingerprint bits and cause ambiguities in distinguishing different user identities. In other words, if the fingerprint has lost its fidelity, the identification clue may be gone and the real traitor may not be caught.

On the contrary, in the proposed method, the noise can be easily removed after joint fingerprinting and decryption because each codeword is decrypted and recovered by finding the nearest codeword in the key-trees. Fig. 14 shows the performance of noise resistance of the proposed method. The noise is in the range of [1,16] and evenly distributed over each element of a codeword. Three kinds of codebooks sized 256, 512, and 1024 are tested, respectively. The figure shows that the polluted image can be completely recovered when the distortion is not too large. More specifically, when the distortion is smaller than 5 (meaning that two polluted bits are allowed) in each codeword, almost all the code-words can be totally recovered. In addition, a larger codebook lowers the capability of noise resistance. Note that the recovery rate in Fig. 14 represents the percentage of the image blocks in a fingerprinted VQ image that can completely remove the noise.

5. Conclusions

The traditional JFD methods are all proposed for pixel domain images or DCT domain images. None of them can be applied to VQ images directly. In addition, these conventional methods cannot resist noise interference, which may damage the fidelity of the fingerprint after decryption. In this paper, the first JFD method for VQ images to deal with the issues of image distribution and traitor tracing is proposed. The proposed JFD method based on VQ is not only that VQ is quite popular for a variety of applications, but also that VQ helps the proposed JFD scheme equipped with the important property of fault tolerance.

In the JFD encryption process, we proposed two different methods: one based on static key-trees and the permutation technique, the other based on the session key and the dynamic key-trees technique. No matter which method is applied, an attacker that intercepts the encrypted image from the Internet cannot decrypt the image or successfully modify the embedded fingerprint. This is because the decryption involves various secrets (i.e., the chaotic map and the key-trees in the first method, and the session key and codebook in the second method). The experimental results show that the encrypted images are perceptually unintelligible and the PSNR value of the decrypted image increases positively with the size of the codebook. Finally, the issue of robustness against noise interference is also fully discussed and evaluated. The proposed JFD scheme with fault tolerance would be more suitable for real applications than conventional ones.

Acknowledgments

This work was supported by National Science Council, Taiwan, under Grant NSC 100-2221-E-468-021.

References