Secure multicasting of images via joint privacy-preserving fingerprinting, decryption, and authentication

Chih-Yang Lin, Kahlil Muchtar, Chia-Hung Yeh, Chun-Shien Lu

Article info
Article history:
Received 21 September 2015
Accepted 5 February 2016
Available online 11 February 2016

Keywords:
Authentication
Joint fingerprinting and decryption (JFD)
Privacy-preserving fingerprinting
Secret sharing
Multimedia distribution
Content integrity
Data encryption
Image authentication

Abstract
Joint fingerprinting and decryption (JFD) is useful in securing media transmission and distribution in a multicasting environment. Common drawbacks of the existing JFD methods are the transmitted data may leak the content of data, and a subscriber cannot determine if a received image is modified such that tampering attack can be mounted successfully. Here we focus on security and privacy of image multicasting and introduce a new framework called JFDA (joint privacy-preserving fingerprinting, decryption, and authentication). It has several main characteristics, JFDA: (1) accomplishes fingerprinting in the encryption domain to preserve privacy and prevent encrypted data from being tampered without additional hash code/digest, (2) prevents tampering attack on the decrypted data to ensure the fidelity of the fingerprinted data, (3) makes user subscribing to a visual media be an examiner to authenticate the same visual media over the Internet. The effectiveness of the proposed method is confirmed by experimental results.

1. Introduction

In the scenario of a sender–receiver–user (w.r.t. source-proxy-destination) multimedia multicasting system, traditional encryption is a common way used to secure media transmission, but the protection capability is nullified after decryption. In order to solve this problem, fingerprint-based methods, in which all the fingerprinted copies of a medium are different but appear to have the same visual perception, have been extensively investigated [3,5,8,9,11,14,16–18,20]. If a fingerprinted copy is illegally distributed, the fingerprint extracted from it can be used to identify the corresponding user. Despite the fact that multicasting of multimedia data has been broadly studied [5,10,12,31], we focus in this paper on several aspects of security and privacy.

In general, there are two types of fingerprinting approaches: server-side fingerprinting and receiver-side fingerprinting. The latter is more practical since only one media stream is required to be disseminated to users in a multicast manner [16,30]. On the contrary, if server-side fingerprinting is used [5,11], the sender needs to consume a lot of bandwidth to send different copies to different users. Therefore, this paper focuses solely on receiver-side fingerprinting.

Here, a receiver can be the set top box or another device to perform the joint fingerprinting and decryption task on behalf of a specific user who subscribes to the content from the server and receives a fingerprinted copy decrypted by the receiver. In the threat model considered in this paper, a receiver is not ideally assumed to be reliable, an attacker may intrude to obtain decrypted data from a receiver or modify it, and data transmitted from the sender to a receiver may be intercepted or modified. We follow the general rules of the media fingerprinting framework under the application scenarios of media fingerprinting [12,28,31], where robustness (aiming to remove fingerprints via any possible attacks excluding the collusion) and blind detection of embedded fingerprints are not the major concerns. Moreover, the collusion attack pertaining to media fingerprinting at the receiver side has also been thoroughly investigated in [28,29,32]. Therefore, we focus on the threat model presented in this paper in order to distinguish our paper from the existing works.

1.1. Conventional receiver-side fingerprinting architectures

The main concerns with receiver-side fingerprinting are content leakage and tampering attacks. If the fingerprint is embedded after
the media stream is decrypted [20], as shown in Fig. 1, attackers gain the opportunity to obtain the decrypted data before being fingerprinted. Therefore, a conventional conception is that security entirely depends on the hardware design and the assumption that the sealed box cannot be cracked to avoid content leakage.

Recently, a new receiver-side fingerprinting framework, called joint fingerprint embedding and decryption (JFD) [1,2,6,13–15], as shown in Fig. 2, has received considerable attention as a possible solution to the secure transmission and traceable distribution problem. The JFD approach encrypts the media on the server side to secure media transmission, and decrypts and fingerprints the media simultaneously on the receiver side by way of using multiple decryption keys on behalf of users. Note that decrypting the same encrypted data with different decryption keys yields different fingerprinted data for traceable distribution. Anderson and Manifavas proposed the Chameleon method [2] following this concept by using a lookup table. The lookup table functions as the key generator from which the key can be used to encrypt the media data using XOR operations. The decryption process is identical to the encryption process except that the keys for the two processes involved are slightly different. More specifically, the corresponding least-significant bits of entries in lookup tables, which belong to encoding and decoding phases, respectively, are different. Therefore, the difference between the encryption and decryption keys draws the fingerprint of a user. Although Chameleon is easy to implement, it is vulnerable to image processing and collusion attack [6,16].

Later, Kundur and Karthik [14], inspired by the concept of Chameleon [2], proposed an alternative JFD method. In their method, the signs of significant DCT coefficients of an image are encrypted on the sender side, and are partially decrypted by the receiver to perform fingerprinting. The positions of the undeciphered coefficients in an image form the fingerprint of a user. Thus, different arrangements of these undeciphered coefficients determine different fingerprints for users. Kundur et al.’s method provides a good framework for JFD, but it is found to have difficulty in achieving the tradeoff between unintelligible encryption and transparent fingerprinted copies. An improved version of Kundur et al.’s method using perceptual models is proposed by Karthik and Hatzinakos [13] to achieve imperceptible fingerprinting by the receiver. However, in this sign bit plane encryption method, the choices of suitable AC coefficients without causing ringing artifacts are content dependent. That is, the selective bit positions in each block for partial decryption, which is related to the user’s fingerprint, should be additionally transmitted to the receiver. These operations would increase computation and bandwidth consumption. Adelsbach et al. [1] extended Chameleon cipher to spread-spectrum watermarking. They also complemented the lack of security proof of Chameleon. Similarly, Lemma et al. [15] proposed a Chameleon-based JFD method using algebraic operations to achieve additive or multiplicative spread-spectrum watermarking. The processes of encryption and decryption are performed by means of different key streams, and each key stream has the same size as the media, leading to prohibitive transmission cost. A generalized version of Lemma et al.’s method has been proposed by Celik et al. [6] based on lookup table operations. In Celik et al.’s method, transmission cost is reduced, and an efficient fingerprint detection method and rigorous security proofs are provided. Lin et al. [17,18] proposed a JFD method based on vector quantization that performs two server-side encryption techniques. The aim of their method is to preserve image quality and resist to noise interference. Nevertheless, the drawback of this method is if the original image is lost or severely damaged, the fingerprint cannot be extracted and the tampered area cannot be identified. This led to the extension work as discussed in [19]. Recently, Czaplewski and Rykaczewski [9] presented a JFD method that utilizes a simple block cipher based on matrix multiplication to encrypt original images. In addition, an individual decryption keys approach is introduced to increase the robustness of proposed system. The above methods, however, still encounter some problems, which are discussed as follows.

1.2. Potential problems of conventional JFD methods and possible solutions

1.2.1. Problems

As shown in Fig. 3, there are two types of attackers over a JFD system: one can eavesdrop to obtain the transmitted encrypted data or tamper it, and the other is an intruder obtaining essential information from a receiver and can further modify not only the encrypted data but also the decrypted data. Since an attacker of the second type can obtain more information than the first type, we mainly focus on the unreliable receiver in the proposed method.

A successful tampering attack on the encrypted data means that malicious manipulation of the encrypted data would not cause significant damage to the decrypted data, as shown in Fig. 4. In Fig. 4, the sender sends the encrypted image to the receiver, but the
encrypted image has been tampered with by an attacker before decryption. When the receiver decrypts the image, the decrypted image is still recognizable with quality similar to the genuine one such that the receiver is unaware that the tampering attack has been mounted. Another type of tampering attack is mounted on the fingerprinted data, which has been a decrypted plaintext format, i.e., tampering attack on the decrypted data. If the decrypted data is modified without giving rise to perceptual artifacts, the forged decrypted data may have destroyed the embedded fingerprint of a user. Such an attack is unavoidable since an intruder can manipulate the data arbitrarily.

Because it is difficult to resist tampering attack on the decrypted data in the existing JFD methods, faithful receivers or some strict conditions are usually assumed in the previous literature. Although it cannot prevent an attacker from manipulating the received data, some strategies can still be investigated to deter a dishonest user from illegal distribution. A deterrent strategy will be discussed in this paper.

1.2.2. Possible solutions?

One may argue that certain solutions, discussed below, can be adopted for deterring tampering attacks from being conducted on the encrypted data (and the embedded fingerprints). Nevertheless, these solutions are found to be infeasible.

(1) Why traditional cryptographic methods are not suitable for the encrypted data?

An intuitive way to resist tampering attacks on the encrypted data may be to apply traditional cryptographic algorithms, such as DES or AES [26]. Once the encrypted data protected by DES/AES are tampered with, the decrypted data become unintelligible. The protection, however, is only valid during transmission from the sender to the receiver. When a receiver gets the encrypted data and lifts DES/AES protection, an intruder can still tamper the decrypted data prior to fingerprinting. In addition, using two layers of encryption operations for transmission is time-consuming.

(2) Why classic hashing cannot be adopted for the encrypted data?

One may consider that a classic hashing algorithm can be simply used to verify the authenticity of encrypted data on the receiver side. That is, any tampering attacks on the encrypted data can be detected by comparing the received hash (or called digest) and the hash generated from the received data. This seems to be rather intuitive. Nevertheless, such a loosely coupled authentication scheme encounters several problems. First, the digest of the encrypted data becomes useless after decryption, so tampering attacks on the decrypted data become un-detectable. Second, since the digest is irrelevant to the decryption process, the integrity check can be eluded, which is first considered in our paper. In other words, the digest is only a reference to check the content integrity, and the intruder can ignore the digest because it is independent of the decryption process. This consequence also leads to an unreasonable case; that is, even if the digest is stained, the recognizable image can still be generated.

(3) Why robust watermarking is insufficient for JFD?

Robust watermarking is widely used to resist distortion of watermarks that are usually in the form of images. Accordingly, some bits of a watermark that may have been modified will not deeply affect the robustness of the watermark. On the contrary, the fingerprint in JFD is in the form of a bit sequence designed delicately in order to resist collusion attacks, so the change of a few fingerprint bits is linked with the correctness of distinguishing different user identities. Thus, the modifications of some fingerprint bits cannot be avoided by solely depending on a robust watermarking
scheme since it focuses on robustness preservation instead of integrity guarantee of a fingerprint. That is why a JFD method with integrity authentication of a fingerprint is highly required.

1.3. Our contributions

As discussed above, a more practical and secure JFD system should consider and possess the following characteristics: (1) the JFD framework should possess the functionality of integrity detection for the encrypted data, called pre-authentication, without relying on sending additional message digest; (2) the JFD framework should be equipped with authentication of the decrypted fingerprinted data, referred to as post-authentication to deter a dishonest user from illegal distribution; (3) the digest used in post-authentication should be correlated with the decryption process to verify the genuineness of the digest; in other words, modifications to digest should affect the decryption result. Therefore, a tightly coupled authentication strategy associated with JFD should be investigated.

To design a JFD method satisfying the above requirements, in this paper, a new and extended JFD framework, called JFDA (joint privacy-preserving fingerprinting, decryption, and authentication) based on the principle of polynomial secret sharing, is proposed for secure image multicasting. To the best of our knowledge, JFDA is the first JFD framework that accomplishes privacy-preserving fingerprinting, decryption, and authentication at one heat. Privacy-preserving in this paper indicates that the user's fingerprint is directly processed and embedded in the encryption domain. Thus, privacy-preserving of fingerprinting, as mentioned in [23], aims at hiding the fingerprint from the receiver in order to prevent tampering with the fingerprint in a straightforward way.

In addition, the role of an attacker who intrudes in a receiver, which is often ignored in the previous JFD frameworks, is first considered as a threat in JFD and will be fully addressed in this paper. If tampering attacks conducted on the encrypted data occur, only a shuffled unintelligible image, as shown in Fig. 5 instead of Fig. 4, will be generated on the receiver side, so the tampering attack can be detected promptly. In this paper, we not only prevent tampering while performing JFD to preserve the losslessness of the fingerprint, but also provide a deterrent scheme to prevent dishonest users from illegal distribution.

1.4. Outline of this paper

The rest of this paper is organized as follows. Related works are reviewed in Section 2 to make this paper self-contained. In Section 3, we present the proposed JFDA method. The security analysis is provided in Section 4. Experimental results are given in Section 5, and conclusions are presented in Section 6.

2. Background

To make this paper self-contained, two prior works relevant to our method are briefly discussed.


Assume that the number of the coefficients to be encrypted is \( t \), i.e., \( c_0, c_1, \ldots, c_{t-1} \), and the size of the encryption table is \( L \), where \( E[0], E[1], \ldots, E[L-1] \) denote the entries of the table. The size of the decryption table for user \( u \) is also \( L \), whose entries are denoted as \( D_u[0], D_u[1], \ldots, D_u[L-1] \), where \( D_u[i] = -E[i] + W_u[i] \), for \( 0 \leq i < L \), and the vector \( W_u \) is the fingerprint of user \( u \). The encryption table \( E \) and the fingerprint \( W_u \) are usually designed as Gaussian distributions. In the encryption phase, if \( c_i \) will be encrypted, a pseudorandom index generator selects \( S \) indices from \( E \), and each index of \( S \) is denoted by \( e_i \) for \( c_i \), where \( 0 \leq e_i < L, 0 \leq j < S \), and \( 1 < S \leq L \). The resultant encrypted coefficient \( e_i \) is obtained by:

\[
e_i = c_i + \sum_{j=0}^{S-1} E[e_j], \quad (1)
\]

In contrast, the corresponding fingerprinted coefficient \( c_i' \) for a user \( u \) can be obtained via the decryption table using the same index generator as:

\[
c_i' = e_i + \sum_{j=0}^{S-1} D_u[e_j] = c_i + \sum_{j=0}^{S-1} W_u[e_j]. \quad (2)
\]

Basically, Celik et al.'s method [6] performs fast encryption/decryption with easy fingerprint design and fidelity control, and reasonable key size. Nevertheless, Celik et al.'s method is vulnerable to both the tampering attack and content leakage. When an attacker intercepts the encrypted data of user \( i \), the data can be slightly disturbed by \( e_i \), as indicated in Eq. (3). Consequently, the decrypted data shown in Eq. (4) would contain \( e_i \) with no apparent artifacts. The introduced noise may be difficult to detect in JFD and may destroy the authenticity of the fingerprint of a user.

\[
e_i = c_i + e_i + \sum_{j=0}^{S-1} E[e_j], \quad (3)
\]

\[
c_i' = e_i + \sum_{j=0}^{S-1} D_u[e_j] = c_i + e_i + \sum_{j=0}^{S-1} W_u[e_j]. \quad (4)
\]

Similarly, tampering the decryption table by \( e_i \) leads to the same effects. Why the conventional JFD methods (e.g., Celik et al.'s method) clearly suffer from tampering attacks is that they all encrypt each coefficient individually without considering the association with other coefficients or factors, so each individual coefficient can be easily manipulated by varying its value within an undetectable level.
2.2. Review of Shamir’s (t, n)-threshold for secret sharing

The first (t, n)-threshold scheme proposed by Shamir in 1979 [24] is discussed here because our method partially follows his concept. The notation t stands for the threshold value, and n is the number of participants (called shadow holders) involved. In the (t, n)-threshold scheme, the secret owner shares out the secret y among n different shadows, \( f(x_i) \)'s, and sets a threshold t for recovering the secrets by the following \((t - 1)\)-degree polynomial over the finite field \( GF(p) \):

\[
f(x) = y + c_1x + c_2x^2 + \ldots + c_{t-1}x^{t-1} \mod p,
\]

(5)

where \( c_1, c_2, \ldots, c_{t-1} \) are random coefficients.

Then, the owner computes the shadow \( f(x_i) \) for each participant \( U_i \) with key \( x_i \). If the number of shadow holders, n, is greater than or equal to t, the secret y can be recovered from t out of n shadows, i.e., \((x_0, f(x_0)), (x_1, f(x_1)), \ldots, (x_{t-1}, f(x_{t-1}))\). Based on the available t shadows, \( f(x) \) can be reconstructed as:

\[
f(x) = \sum_{i=0}^{t-1} f(x_i) \prod_{j=0, j \neq i}^{t-1} \frac{x - x_j}{x_i - x_j} \mod p.
\]

(6)

Otherwise, if the number of shadow holders is less than t, the secret cannot be recovered via their cooperation.

Shamir’s secret sharing is perfect, which indicates that any illegitimate subset of participants has absolutely no information of the secret. Formally speaking, let \( \Gamma \) be an access structure, which contains all the qualified subsets to access the secret y. If any subset \( U \) of participants satisfying \( U \subseteq \Gamma \), \( H(y - U) = 0 \); otherwise, \( H(y)(U) = H(y) \), where \( H(A) \) means the entropy or uncertainty about element A.

3. The proposed method

In this section, the new system architecture for joint privacy-preserving fingerprinting, decryption, and authentication (JFDA) is first presented in Section 3.1. Then, our polynomial-based JFDA method equipped with privacy-preserving fingerprinting is described in Section 3.2. In Section 3.3, a tampering-resilient and cost-efficient JFDA method is discussed. Finally, the proposed JFDA method equipped with both pre-authentication and post-authentication mechanisms is proposed in Section 3.4.

3.1. System architecture

The typical JFD framework [14] is shown in Fig. 2, in which fingerprinting and decryption are jointly performed on the receiver side. Hence, JFD of this model is like two sides of one coin, i.e., the decryption process results in fingerprinting and vice versa. The strong correlation between fingerprinting and decryption, however, limits the design strategies for a JFD method and exposes weaknesses in resistance against tampering attacks. In view of this, we propose a new framework, as shown in Fig. 6, i.e., fingerprinting is directly performed on the encrypted media data, followed by decryption on the receiver side. Since fingerprinting is conducted in the encryption domain without decrypting the media data beforehand, we call the proposed fingerprinting strategy, privacy-preserving fingerprinting.

In Fig. 6, labels 1, 2, and 3 denote three possible weak points at which an attacker may obtain content. At Point 1, the attacker can intercept the encrypted media through the Internet while a fingerprinted stream can be obtained at Points 2 and 3. In this framework, we should satisfy two requirements: (1) the valid decryption process must be performed after the fingerprinting process; (2) all tampering attacks should be detected.

3.2. Fundamental polynomial-based JFDA: Basic framework for privacy-preserving fingerprinting

Assume that VM denotes a visual media and that the features of VM, such as pixel values or transformed (e.g., DCT or wavelet) coefficients to be encrypted, are represented by \( c_1, c_2, \ldots, c_t \), where \( 0 \leq c < p \), and \( p \) is a prime number. Since our proposed JFDA method exploits the concept of Shamir’s secret sharing technology, these pixels/coefficients can constitute a polynomial \( g(x) \) as:

\[
g(x) = c_1x + c_2x^2 + \ldots + c_tx^t \mod p
\]

(7)

In contrast to Eq. (5), all \( c_i \)'s in Eq. (7) instead of merely \( y \) in Eq. (5) are regarded as secrets. Here, the reason that, \( g(x) \) of Eq. (7) excludes the constant term, is to prevent tampering attacks, which will be discussed at the end of this section.

The sender also randomly selects a \( t' \)-degree perturbation polynomial, \( \phi(x) \), as:

\[
\phi(x) = \phi_0 + x_1x + \ldots + x_ix^i \mod p.
\]

(8)

where \( \phi_0, x_1, \ldots, x_i, 0 \leq i < p \), are coefficients of \( \phi(x) \) and \( t' \geq t \). Then, the two polynomials defined in Eqs. (7) and (8) are combined to form \( G(x) = g(x) + \phi(x) \) for encrypting \( g(x) \) as:

\[
G(x) = g_0 + (c_1 + x_1)x + \ldots + (c_t + x_t)x^t \mod p.
\]

(9)

The function, \( \phi(x) \), is used to enhance the security of the proposed JFDA. The polynomial \( G(x) \) uses \( t' \) 's to produce \( t \) shadows (i.e., the values of \( G(x_i) \), \( 0 \leq i < t \)), and these shadows, called shuffled media (SM), are sent to the receiver side via the Internet. For attackers eavesdropping on the Internet, the number of shadows they can obtain is at most \( t \) but the degree of \( G(x) \) dominated by \( \phi(x) \) is \( t' \), which is greater than or equal to \( t \). Under this circumstance, \( G(x) \) cannot be reconstructed according to Shamir’s perfect secret sharing mechanism, to say nothing of getting \( c_i \)'s of \( g(x) \) since each \( c_i \) has been blended with \( x_i \) in \( G(x) \). Thus, the goal of secure transmission without content leakage is achieved (weak point 1 in Fig. 6).

Similarly, in order to avoid exposing the fingerprint to the receiver who performs joint fingerprinting and decryption on behalf of the user, the sender prepares a polynomial (i.e., a perturbed fingerprint), \( F_u(x) = f_u(x) - \phi(x) \), for user \( u \), where \( f_u(x) \) is the fingerprint of user \( u \) and \( \phi(x) \) is the same as the one used in \( G(x) \) on the sender side. The format of \( f_u(x) \) is also the same as that of \( g(x) \), but the selected coefficients \( w_i \)'s of \( f_u(x) \) relate to the robustness and perceptual quality of the fingerprinted media. According to Eqs. (7) and (9), \( f_u(x) \) and \( F_u(x) \) can be, respectively, designed as shown in Eqs. (10) and (11):

![Fig. 6. The proposed JFDA framework.](image-url)
\[ f_s(x) = w_1x + w_2x^2 + \ldots + w_nx^n \mod p; \quad (10) \]

\[ f_u(x) = -x_0 + (w_1 - x_1)x + \ldots + (w_{n-1} - x_{n-1})x^{n-1} + x_nx^n \mod p. \quad (11) \]

Note that, in general cases, the size of a fingerprint is much smaller than that of VM, so the repetition of a fingerprint can be applied to Eq. (10) to make the embedded fingerprint size be equal to \( t \).

Then, the sender produces a decryption key for the user \( u \); namely a lookup table \( C_u \) of size \( |C_u| = t \). Each entry of \( C_u \) denoted by \( C_u(i) \), is created by \( F_s(x) \) using the same \( x_i \)’s used in \( G(x) \). There are two main reasons why the values of \( F_s(x) \) (i.e., \( C_u \)) instead of the polynomial \( F_s(x) \) are placed on the receiver side: one is to speed up the decryption process; the other is that if the polynomial \( F_s(x) \) is given to the receiver, he/she would need all \( x_i \)’s to calculate the values of \( F_s(x) \), thus revealing all \( x_i \)’s to the receiver and validating tampering attacks. To decrypt the media correctly, the receiver also needs a set of Lagrange basis polynomials \( LP = \{ lp_j(x) | 0 \leq j < t \} \) with \( lp_j(x) \) defined as:

\[ lp_j(x) = \frac{1}{\sum_{j=0, j\neq j}^{t-1} x - x_j} \mod p. \quad (12) \]

Once SM is received, the receiver combines SM, \( C_u \), and \( LP \) by performing \( \sum_{j=1}^{t-1} G(x) + C_u(i) \times lp_j(x) \) to decrypt the media for user \( u \), and then \( t \) fingerprinting coefficients \( c_j \)’s can be generated, where \( c_j = c_i + w \). The complexity of getting \( c_j \)’s on the receiver side is \( O(t) \) (attributed to the help of Lagrange basis polynomials; otherwise, if all \( x_i \)’s instead of \( LP \) are given, the solution, \( c_j \)’s, can be obtained by Gaussian elimination, of which the complexity is \( O(t^3) \)). In addition, the given \( c_j \)’s would cause tampering attacks that will be discussed in Section 3.3. In the proposed method, if a tampering attack occurs, the whole image will remain unintelligible, which will be discussed in Section 4.

3.2.1. An example of privacy-preserving fingerprinting

Assume the sender wants to send three features (pixel values or transformed coefficients), \( 7, 6, 5 \), to the receiver who performs joint fingerprinting and decryption on behalf of user \( u \) and chooses a prime \( p = 11 \). In the beginning, the sender should perform the following (1) The three features constitute \( g(x) = 7x + 6x^2 + 5x^3 \mod 11 \); (2) Construct \( \phi(x) = 9 + 8x + 7x^2 + x^3 \mod 11 \); (3) Choose the fingerprint \( \{3, 2, 2\} \) for the user \( u \) expressed in the form of \( f_s(x) = 3x + 2x^2 + 2x^3 \mod 11 \). Therefore, \( f_u(x) = f_s(x) - \phi(x) = 2x + 6x + 6x^2 + x^3 \mod 11 \); (4) Select \( x_i \) as 1, 2, and 3. Thus, \( LP = \{3 - 8x + 6x^2 - 7 + 2x + 5x^2 + 4 + 5x + 2x^3\} \) can be obtained by Eq. (12); \( C_u = \{4, 2, 2\} \) is generated via \( f_u(x) \), and the encrypted media \( \{10, 7, 3\} \) is constructed by \( G(x) \); (5) Finally, the sender sends \( LP \), \( C_u \), and the shuffled media \( \{10, 7, 3\} \) to the receiver. Note that, the data of \( LP \) and \( C_u \) only need to be sent to the receiver once in the initial stage and can be reused for the later encryption process.

When the receiver receives \( LP \), \( C_u \), and the shuffled media \( \{10, 7, 3\} \), it can get the fingerprinting media by carrying out the task: \( \sum_{j=1}^{t-1} [G(x) + C_u(i) \times lp_j(x)] / \{ \text{that is } (10 + 4) \times (3 - 8x + 6x^2) + (7 - 2) \times (-7 + 2x + 5x^2) + (3 + 2) \times (4 + 5x + 2x^3) \} \mod 11 = 10 + 8x + 7x^2 \}, \) in which the coefficients \( 10, 8, 7 \) is the final result of the fingerprinting media. In this scheme, only the shuffled media and \( LP \) can be public, and the remaining should keep secret.

\[ \begin{bmatrix}
1 & x_0 & \cdots & x_{10}^3 \\
1 & x_1 & \cdots & x_{10}^3 \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{t-1} & \cdots & x_{10}^3 \\
\end{bmatrix} = \begin{bmatrix}
c_1 \\
c_1 \\
\vdots \\
\vdots \\
\end{bmatrix} = f_s(x) \mod p. \quad (13) \]

\[ g(x) = c_0 + c_1x + \ldots + c_tx^t \mod p. \quad (14) \]

3.3. Tampering-resilient and cost-effective JFDA

Section 3.2 presents a basic framework of privacy-preserving JFDA. There are, however, still two problems involved in the proposed JFDA method. The first one is relevant to the security breach of JFDA due to the unreliable receiver and the second one is the space requirements of each user for decryption.

3.4. Pre-authentication for tampering-resilience to unreliable receiver

For the first problem, an intruder can obtain more information, namely \( C_u \) and \( LP \), from the receiver than the eavesdropper. If the receiver is unreliable and an intruder cracks \( LP \) to get all \( x_i \)’s, the fingerprinted data can be successfully tampered with (weak point 2 in Fig. 6) without causing visual damages. For example, a fingerprinted pixel/coefficient can be successfully modified from \( c_j \) to \( c_j + \varepsilon \) by changing each \( G(x) + F_u(x) \) to \( G(x) + F_u(x) + \varepsilon \times x_i \) if all \( x_i \)’s can be known. How complex would it be for an intruder to successfully crack \( LP \) and obtain all \( x_i \)’s? In Shamir’s secret sharing method, the polynomial representation of Eq. (5) can be expressed as a matrix form for \( x_i \)’s, where \( 0 \leq i < t \), as:

\[ \begin{bmatrix}
1 & x_0 & \cdots & x_{10}^3 \\
1 & x_1 & \cdots & x_{10}^3 \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{t-1} & \cdots & x_{10}^3 \\
\end{bmatrix} = \begin{bmatrix}
y \\
c_1 \\
\vdots \\
c_{t-1} \\
\end{bmatrix} = AC. \quad (15) \]

Each row of \( A \) in Eq. (15) represents a different \( x_i \) used in Eq. (5) and \( t \) \( x_i \)’s are required because \( t \) variables in \( C \) constitute \( f(x) \) in Eq. (5). Since \( A \) is square and nonsingular, \( A^{-1} \) exists and is unique [27]. Assume that \( F \) is the set of shadows, composed of \( f(x) \) by Eq. (5).

\[ AC = FA \quad \text{and} \quad C = A^{-1}F. \]

According to the uniqueness of \( A^{-1} \), the structure of \( A^{-1} \) is the matrix representation of Eq. (6), where the coefficients of each Lagrange basis polynomial comprise one column of \( A^{-1} \). Once the receiver owns \( A^{-1} \), the time complexity of getting all \( x_i \)’s is \( O(t^3) \), which is equivalent to performing the inverse operation from \( A^{-1} \) to obtain \( A \).

Now, consider the case of the tampering attack on the method proposed in Section 3.2. The encrypted and fingerprinted copy obtained from Eq. (9) plus Eq. (11) can be expressed as:

\[ \begin{bmatrix}
x_0 & 0 & \cdots & 0 \\
0 & x_1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & x_{t-1} \\
\end{bmatrix} = \begin{bmatrix}
y \\
c_1 + W_1 \\
\vdots \\
c_{t-1} + W_1 \\
\end{bmatrix} = AC. \quad (16) \]

\[ \begin{bmatrix}
1 & x_0 & \cdots & x_{10}^3 \\
1 & x_1 & \cdots & x_{10}^3 \\
\vdots & \vdots & \ddots & \vdots \\
1 & x_{t-1} & \cdots & x_{10}^3 \\
\end{bmatrix} = \begin{bmatrix}
y \\
c_1 + W_1 \\
\vdots \\
c_{t-1} + W_1 \\
\end{bmatrix} = AC. \quad (16) \]
In this case, matrix $B$ is involved in matrix $A$ because the constant term is not allowed in Eqs. (7) and (10), and each $c_i + w_i$ is associated with $x_i$. Since matrix $A$ is still square and nonsingular, $A^{-1}$ exists and is composed of $b_p(x)$. When an intruder obtains $A^{-1}$ and attempts to modify $c_i + w_i$ to $c_i + w_i + \epsilon$, the first step of tampering attack is to recover $A$ from $A^{-1}$. After that, the noise $E$ is added to $AC$ to finish the tampering attack as:

$$AC + AE,$$

where $E = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_t \end{bmatrix}$.

The complexity of achieving the tampering attack is still $O(t^3)$ dominated by the inverse matrix operations.

From the above observations, the key to a successful tampering attack relies on whether the matrix $A$ can be obtained efficiently. In order to increase the security against obtaining $A$, in the proposed method, the sender only sends the first $(t - 1)$ rows of $A^{-1}$, an incomplete inverse matrix, to the receiver. Since only $(t - 1)$ rows of $A$ are available, an intruder, who cracks the unreliable receiver, cannot recover $A$ correctly from the non-square matrix with dimensions $(t - 1) \times t$. In other words, the dimensions of $A$ remain $t \times t$, but only partial $A^{-1}$ of dimensions $(t - 1) \times t$ is sent to the receiver. Using this strategy, the receiver still gets $t$ encrypted coefficients, but only $(t - 1)$ of them can be decrypted. This idea is realized by using a dummy variable, which will be introduced below.

In addition to sending the first $(t - 1)$ rows of $A^{-1}$ to the receiver, the randomness of $A$ should also be increased to enhance the security. For the scheme described above, see for example Eq. (16), each row of $A$ is generated by a given $x_i$, so it forms a kind of regular pattern (i.e., $x_i, x_i^2, \ldots, x_i^n$ in order) and the receiver can get an entire row of $A$ if $x_i$ is correctly guessed. Nevertheless, since $A$ is controlled by the sender, the sender can increase the randomness of $A$, making each row of $A$ be a non-regular pattern, and, thus, can enhance the security of $A$. Specifically, based on this idea, Eq. (7) can be reformulated and rewritten as:

$$g(X) = c_1 x_1 + c_2 x_2 + \ldots + c_t x_{t-1} + d_t x_t \mod p,$$

where $d_t$ is a dummy variable, $X_t$ is a set consisting of $t$ variables, namely $x_1, x_2, \ldots, x_t$, for the $i$-th row of $A$. Eq. (18) implies that $(t - 1)$ pixels/coefficients $c_i$'s and one dummy variable $d_t$ constitute a polynomial $g(X)$ with $t$ variables $X_t$'s. Basically, the formulation Eq. (18) generalizes Eq. (7) originally used in Shamir's secret sharing scheme.

The same reformulation rule is also applied to Eqs. (8)-(11) to obtain $\phi(X), G(X), f_p(X), F_p(X)$, respectively. The $X$ in $g(X)$, $\phi(X), G(X), f_p(X)$, and $F_p(X)$ means the set of all $X_t$, where $1 \leq j \leq t$. An example of encrypting $(t - 1)$ coefficients in a fingerprinted media obtained via $g(X) + \phi(X)$ is shown as:

$$\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1t} \\ x_{21} & x_{22} & \ldots & x_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t-1,1} & x_{t-1,2} & \ldots & x_{t-1,t} \\ x_{t1} & x_{t2} & \ldots & x_{tt} \end{bmatrix} + \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{t-1} \\ d_t \end{bmatrix} = \begin{bmatrix} x_{10} & x_{11} & x_{12} & \ldots & x_{1t} \\ x_{20} & x_{21} & x_{22} & \ldots & x_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t-1,0} & x_{t-1,1} & x_{t-1,2} & \ldots & x_{t-1,t} \\ x_{t0} & x_{t1} & x_{t2} & \ldots & x_{tt} \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{t-1} \\ a_t \end{bmatrix} \equiv \begin{bmatrix} z_{0} \\ z_{1} \\ \vdots \\ z_{t-1} \\ z_{t} \end{bmatrix} \pmod{p},$$

where $d_t$ is also a dummy value, and $a_i = \Sigma x_{tj} \times w_j$ and $r_t = \Sigma x_{tj} \times \epsilon_j$.

When the receiver receives the encrypted data (i.e., $b_i + r_i$), fingerprinting is performed in the encryption domain by $b_i + r_i + a_i + (-r_i)$. Then, the decryption process in getting $(c_0 + w_0), (c_1 + w_1), \ldots, (c_{t-1} + w_{t-1})$ with the help of $(t-1)$ rows of $A^{-1}$ is shown as:

$$\begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1t} \\ x_{21} & x_{22} & \ldots & x_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ x_{t-1,1} & x_{t-1,2} & \ldots & x_{t-1,t} \\ x_{t0} & x_{t1} & x_{t2} & \ldots & x_{tt} \end{bmatrix} + \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ \vdots \\ a_{t-1} + b_{t-1} \\ a_t + b_t \end{bmatrix} \equiv \begin{bmatrix} z_{0} \\ z_{1} \\ \vdots \\ z_{t-1} \\ z_{t} \end{bmatrix} \pmod{p},$$

If any of the encrypted data is tampered with, all the $(t - 1)$ decrypted data will be affected due to the fact that $(c_i + w_i) = \Sigma x_{tj} \times (a_j + b_j)$. Therefore, pre-authentication to ensure the fidelity of the received data is achieved.
3.5. Space requirement

The second problem of the JFDA method described in Section 3.2 is space consumption. The Lagrange basis polynomials in Eq. (12) containing $t^2$ coefficients that must be stored on the receiver side will increase the hardware cost. In order to reduce the space complexity, the features of VM are partitioned into several segments of equal size $t_0$, where $t_0$ can be equal to the size of the fingerprint and is usually much smaller than $t$. Under this circumstance, each segment is individually encrypted, fingerprinted, and decrypted. As a result, the cost of preserving coefficients on the receiver side will be significantly reduced from $t^2$ to $t_0^2$. In addition, the size of the lookup table $C_t$ for the receiver can also be reduced from $t$ to $t_0$.

3.6. JFDA equipped with post-authentication

Based on the proposed JFDA framework, as shown in Fig. 6, there are three possible weak points, at which an intruder can tamper with the data. At Point 1, the JFDA method presented in Section 3.2 has shown the robustness based on the secret sharing scheme. At Point 2, we also provide a solution in Section 3.3 to resist tampering attacks. At Point 3, an intruder may directly tamper with the decrypted fingerprinted image, which is found to be a relatively unexplored attack in the study of JFD.

Basically, the attacks at Points 1 and 2 are called ciphertext only attacks, and the attack at Point 3 is referred to as a receiver-side attack operated on plaintext only. To efficiently overcome the attack at Point 3, an authentication mechanism should be involved in the design of conventional JFD methods. For this, the proposed JFDA is extended to involve post-authentication. Every user can be an examiner to authenticate his/her received data.

Assume that $h(\cdot)$ is a hash function that the receiver can use, and $h(G(X))$ and $h(F(X))$ represent the hash values of the encrypted data $G(X)$ and perturbed fingerprint $F(X)$, respectively. The pair $(G(X), h(G(X)))$ could ensure the authenticity of $c_1$’s in $G(X)$. This property can also be applied to $(F(X), h(F(X)))$ to protect $w_1$’s in $F(X)$. If the sender, however, sends $(G(X), h(G(X)))$ to the receiver for authentication, the bandwidth consumed would become double. Similarly, the pair $(F(X), h(F(X)))$ would also double the space requirement for the receiver. Furthermore, when the receiver obtains the fingerprinted data, $c_1 = c_1 + w_1, h(G(X))$ and $h(F(X))$ become useless because of the absence of the message digest of $c_1 + w_1$. Thus, when the receiver is unreliable, an intruder can still arbitrarily manipulate $c_1$ after the decryption process.

Basically, the mechanism of authentication applied to JFD should satisfy three requirements: (1) it must be performed on the receiver side, (2) any possible modifications to hash values should affect decryption results, (3) it must comply with the principle of JFDA—sending one data stream and producing multiple fingerprinted copies.

To meet the three requirements, the authentication mechanism is designed as follows. Assume that $VM'$ is the fingerprinted media, and $h(VM')$ is identical (we describe how to achieve this later in this section) for different users subscribing to the same content. This assumption not only helps the realization of post-authentication of JFDA but also complies with the main principle of JFD; i.e., sending one data stream from the sender and producing multiple fingerprinted copies for different users. For convenient description, let $h_i$ in Eq. (19) be equal to $h(VM')$. Since the sender has the information of $x_{i0}, c_i$, and $h(VM')$, the dummy value $d_i$ can be determined from

\[ x_{i0} \times c_i + x_{i2} \times c_2 + \ldots + x_{i-1} \times c_{i-1} + x_{i} \times d_i = h(VM'). \]

Accordingly, $b_1, b_2, \ldots, b_t$ in Eq. (19) can be generated sequentially.

Similarly, in order to determine $d_2$ in Eq. (20), $a_1$ is set to 0 since $b_1$ is set to $h(VM')$. In fact, $a_1$ and $b_1$ can be set arbitrary but complying with the rule that $a_1 + b_1 = h(VM')$. Therefore, the remaining values of $a_t - r_i$ (i.e., the entries of the lookup table), can be evaluated. After $a_t$ is fixed, the lookup table $C_t$ can be constructed and then used repeatedly for different media files.

Based on such a design, the message digest, $h(VM')$, can be obtained directly from the fingerprinted copy via Eq. (19) plus Eq. (20), and the fingerprinted data $(c_1 + w_1)$’s are presented through the first $(t - 1)$ rows of $A^{-1}$. If the fingerprinted copy is authentic, the hash value of the fingerprinted media composed of $(c_1 + w_1)$’s should be equal to $h(VM')$. This is the basic property for a general authentication scheme to approve the fingerprinted media. Nevertheless, in a conventional JFD scheme, even if $h(VM')$ is not equal to the hash value, the decryption process would not be affected, bringing about another issue; i.e., whether $h(VM')$ can be trusted. Therefore, in addition to the basic authentication property, our message digest $h(VM')$ further plays an important role; i.e., a key to the decryption process. In other words, if $h(VM')$ is tampered with, only the shuffled image can be obtained. If the genuine fingerprinted image can be generated, $h(VM')$ should be authentic and can be trusted. This important property assures that every user gets the hash $h(VM')$ to authenticate the received data.

To satisfy the precondition that different users have the same $h(VM')$ for the same media content, we can, for example, assume that the summation of each user’s fingerprint is equal, so that $h(VM') = h(w_1 + w_2 + \ldots + w_t, c_1 + c_2 + \ldots, c_t)$ is identical for each user subscribing to the same VM. Therefore, the goal of sending one VM and producing multiple copies can be achieved under the constraint that $h(VM')$ is the same for all users for the same VM. The equivalent summation of each user’s fingerprint complies precisely with the design of anti-collusion codes (ACCs) [28], as shown in Fig. 7, where each column with a summation that is the same as others is a fingerprint for a user.

Note that the fact that $h(VM')$ is identical for different users subscribing to the same content does not mean that $h(VM')$ is only correlated with the content of VM. Since the receiver only has the fingerprinted data and knows nothing about the original data, the design of $h(VM')$ should involve not only the content of VM but also the information about the user’s fingerprint.

The fingerprinted media on the receiver side can be approved only when it is authenticated. Otherwise, we can claim that the fingerprinted media or the authentication code suffers from tampering attack. In this design model, every user subscribing to VM can be an examiner to authenticate the same VM over the Internet. Accordingly, tampered copies can be efficiently detected, and thus, illegal distributors would be deterred from imposing tampering attacks. In addition, traitor tracing problems can also be simplified.
For example, in a collusion attack, if traitors provide forged fingerprinted data to each other, in which the fingerprint has lost its authenticity, most of traitor tracing algorithms would be compromised. Since each attacker has different abilities in removing fingerprints, generally, no colluder is willing to take more risk than any other colluder due to the concern of fairness. Therefore, the authentication mechanism can play an important role to intimidate and force all colluders to present their authenticated fingerprinted data, thus, validating traitor tracing algorithms.

In Section 3.3, we have mentioned that the features (pixels/transformed coefficients) of VM partitioned into segments can save the space usage. How to apply this JFDA scheme to multiple segments is still straightforward. In Eq. (19), the fact that \( t - 1 \) c’s (which are plaintext) will produce \( t \) shuffled values (which are ciphertext) enlarges the transmission cost. If the features of VM are partitioned into \( s \) segments and each segment contains \( (t_i - 1) \) c’s, the transmission cost will be increased from \( s \times (t_i - 1) \) to \( s \times t_i \). This extra cost is considerable when \( t_i \ll t \) and \( s \) is large, such as in secure video distribution. To reduce the transmission cost, \( h(VM') \) only needs to be sent once to the receiver through the first segment, and the remaining segments repeatedly use the same \( h(VM') \) for decryption by the similar equation shown in Eq. (21). Therefore, each of the segments other than the first one only sends \( (t_i - 1) \) instead of \( t_i \) shuffled values to the receiver. In other words, in this scheme, the encryption volume is approximately equal to the decryption one with the ratio of \((s \times t_i + 1)/(s \times t_i)\approx 1\).

4. Security analysis

We analyze the security of the proposed method from five cases: (1) eluding attack, (2) ciphertext only attack, (3) linear estimation attack, (4) tampering attack, and (5) collusion attack. The secret created by the sender is these polynomials: \( g(X), \phi(X), \) and \( f_i(X) \). Assume that the range of each \( w_i \) in \( f_i(X) \) is within \( r \), where \( 0 < r < p \). Please note that the so-called "receiver-side attack operated on plaintext only" conducted on Point 3 of Fig. 6 can be detected via the authentication mechanism presented in Section 3.4.

4.1. Eluding attack

An eluding attack aims to avoid the fingerprinting process in the encrypted media on the receiver side. Nevertheless, since the receiver only has the decryption key \( C_u \) for user \( u \) and an incomplete inverse matrix without any information about \( \phi(X), g(X), \) or \( f_i(X) \), the only way for the receiver to get the decrypted media is to perform \( G(X) + F_u(X) \), in which the fingerprint has been embedded. In other words, the decrypted media cannot be obtained without executing the required operations through Point 2 in Fig. 6.

4.2. Ciphertext only attack

A ciphertext only attack aims to guess the original \( c_i \) or \( w_i \). If the attackers intercept data through the Internet, only the shuffled media can be obtained. Therefore, guessing attack is the only way one can try if the attacker wants to decrypt the shuffled media. The probability of getting the correct media is \( 1/p^t \), where \( t \) is the number of coefficients \( c_i \) involved in Eq. (19). This probability is almost zero since \( p \) and \( t \) are normally large. Similarly, the attacker can also guess \( w_i \)'s on the receiver side. However, the probability of cracking the complete fingerprint is \( 1/e^t \), which also approaches zero, where \( t \) is the length of the fingerprint and \( t < t \).

4.3. Linear estimation attack

In Celik et al.'s method [6], the encryption method shown in Eq. (1) is modeled as:

\[
e = c + TE,
\]

where \( e \) denotes the encrypted data, \( c \) is the original data, \( T \) is the session key-dependent matrix, and \( E \) is the encryption lookup table. Under this model, an attacker may use a linear estimator to derive the original data from the encrypted data or from the fingerprinted data. Celik et al. proved that the quality of the derived data would relate to the size of the lookup table. Assume that \( M \) is the size of the original data and \( L \) is the size of the lookup table. Celik et al. showed that the larger size of the lookup table is, the worse spurious image quality will be obtained.

The linear estimation analysis of our method is similar to that of Celik et al.'s method since our encryption method in Eq. (19) is structurally identical to Eq. (22) if we set:

\[
\begin{bmatrix}
\mathbf{b}_1 + r_1 \\
\mathbf{b}_2 + r_2 \\
\vdots \\
\mathbf{b}_{t_i-1} + r_{t_i-1} \\
\mathbf{b}_1 + r_1
\end{bmatrix}
\begin{bmatrix}
\mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1s} \\
\mathbf{x}_{21} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{2s} \\
\vdots & \vdots & \ddots & \vdots \\
\mathbf{x}_{(t_i-1)1} & \mathbf{x}_{(t_i-1)2} & \cdots & \mathbf{x}_{(t_i-1)s} \\
\mathbf{x}_{11} & \mathbf{x}_{12} & \cdots & \mathbf{x}_{1s}
\end{bmatrix}
\begin{bmatrix}
\mathbf{c}_1 \\
\mathbf{c}_2 \\
\vdots \\
\mathbf{c}_{t_i-1}
\end{bmatrix}
\begin{bmatrix}
\mathbf{d}
\end{bmatrix}
\Rightarrow e = AC + TE = c + TE\text{Let } c = AC.
\]

In the proposed scheme, however, even when the linear estimation attack is applied, an attacker can only estimate \( AC \) in Eq. (23) instead of the original coefficients (i.e., \( C \)). Hence, the attacker still cannot obtain the desired information. The security of the proposed scheme relies not only on \( TE \) but also on the matrix \( A \), which remains secret in the whole JFDA process.

4.4. Tampering attack

A tampering attack aims to tamper with the encrypted media in order to disturb the fingerprint in the decrypted media. If an attacker arbitrarily tampers with \( G(X) \) (at Point 1 of Fig. 6) or \( F_u(X) \) for user \( u \) (at Point 2 of Fig. 6), the summation of the incomplete inverse matrix multiplied by these tampered values would result in a major effect on the decrypted image even when the modification is insignificant. Therefore, the tampered decrypted image still remains unidentifiable. In Section 5, we will show what happens when \( G(X) \) or \( F_u(X) \) is tampered with.

A successful tampering attack may occur at Point 3 when an intruder, who cracks the unreliable receiver, uses fingerprint (watermark) removing algorithms such as Stirmark attacks [22]. Nevertheless, if JFDA with post-authentication is applied, such a tampering attack still can be detected, as already discussed in Section 3.4.

4.5. Collusion attack

There are two types of collusion attacks; namely key estimation and framing. In the case of key estimation, when the users obtain
the decrypted media, they can collude with each other to attempt to dilute the influence of their own fingerprints. Assume that the decrypted data for users $u$ and $v$ are denoted as $g(X) + f_u(X)$ and $g(X) + f_v(X)$, respectively. The key information that the two colluders can obtain is only the difference, $D(X) = f_u(X) - f_v(X)$, of their own fingerprints. Therefore, they still cannot unveil $f_u(X)$ or $f_v(X)$ from $D(X)$, and the probability of guessing the correct fingerprint is still $1/r^2$. This result is also suitable for $k(\geq 2)$ colluders.

Another collusion approach to key estimation is using the collusion of $k$ decryption keys $C_i$’s to decrypt $\phi(X)$, each of which is composed of $F_i(X)$, thus $f_i(X)$ and $g(X)$ are able to be exposed. According to Eq. (11) using $X$ defined in Eq. (18), $F_i(X)$ with $t' + 1$ variables derived as:

$$F_i(X) = -x_0 + (w_{t1} - x_t)x_1 + \ldots + (w_{tL} - x_t)x_L + \ldots$$

$$- x_t x_t \mod p,$$

only provides $t$ shadows in $C_i$, where $t' + 1 > t$. These indicate that not only $\phi(X)$ but also the fingerprint (i.e., $w_{t1}, w_{t2}, \ldots, w_{tL}$) cannot be unveiled. If the $k$ colluders contribute their decryption keys, $k \times t$ shadows can be obtained but the number of variables from $k$ $F_i(X)$’s is up to $(t' + 1) + (k - 1) \times t$ since $\phi(X)$ is identical but $F_i(X)$’s are different in $F_i(X)$’s. By the fact that $[(t' + 1) + (k - 1) \times t]$ is still greater than $k \times t$, the coefficients of $F_i(X)$’s remain secret, not to mention $\phi(X)$ and $f_i(X)$’s. Therefore, the collusion of $k$ decryption keys is still failed. Undoubtedly, the key information that each colluder can obtain using their $C_i$’s is only the differences of their fingerprints, letting the probability of guessing the correct fingerprint remain $1/r^2$.

In the case of framing, the intended conspirer forges a fingerprinted copy to frame innocent receivers or, at least, to eliminate his/her identity. The well-known attacks include linear and non-linear collusions. This powerful attack has been widely investigated in the fingerprint design [7,21,28]. One of the feasible ways to resist this kind of attack is to employ anti-collusion codes (ACCs) [4,28], which can be incorporated into the proposed method. Since the collusion attack pertaining to media fingerprinting has also been thoroughly investigated in [25–27], the focus of our paper, as described previously, is on the threat model presented in this paper in order distinguish our paper from the existing works.

5. Experimental results

Several experiments conducted on a personal computer with an Intel Core 2 2.26 GHz processor and 2 GB of RAM were used to evaluate the performance of the proposed method. In our experiments, four standard 512 × 512-pixel grayscale images shown in Fig. 8 were used as the cover images for JFDA. The effectiveness of our methods was demonstrated in both the spatial domain and frequency domain, respectively, in Sections 5.1 and 5.2 together with performance comparisons conducted in Section 5.3.

5.1. JFDA in the spatial domain

The first kind of experiments was conducted in the spatial domain, where the features of VM are all the pixels in the host image. Assume that the fingerprint size is 256 and is repeatedly added to the VM; each entry of the fingerprint is randomly set to the range of [0,7] for concern of perceptual quality; and $p$ is set to 251. Figs. 9 and 10, respectively, show the results of the encrypted and fingerprinted images obtained from the proposed method. It can be seen that the encrypted images are all sufficiently scrambled, and the hidden fingerprints in the fingerprinted images are all imperceptible. The imperceptibility of an image can be measured by the peak-signal-to noise ratio (PSNR), which is defined as:

$$PSNR = 10 \times \log 10(255^2/MSE),$$

where MSE denotes the mean square error between the original image and the encrypted image. Most of the PSNRs for the encryption results in Fig. 9 are smaller than 10. In general, PSNR
for acceptable image quality should be above 30; PSNR below 20 is already too low for human perception.

When the attacker intercepts the encrypted image $G(X)$ from the Internet, he/she may want to affect the fingerprinted image obtained via $G(X) + F_u(X)$. For example, if the tampered $G(X)$ is used, the image still remains unintelligible after decryption, as shown in Fig. 11(a). Similarly, Fig. 11(b) shows that if tampered $F_u(X)$ is applied, the decrypted image is also scrambled. These results demonstrate the resistance of our method to tampering attack.

The encryption results obtained using Celik et al.’s method [6] are shown in Fig. 12 for comparison, which are similar to those of the proposed method. Nevertheless, if the encrypted image is tampered with, in which each encrypted value is increased by only 1, the tampered decrypted image is very similar to the original one and difficult to detect, as shown in Fig. 13. In Celik et al.’s method, since tampering attacks on the encrypted and decrypted image are equivalent, each pixel of the difference images between Figs. 8 and 13 is equal to 1. In other words, if each encrypted value is increased by 1, the resultant decrypted image will also be increased by 1.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption (s)</td>
<td>0.18</td>
<td>0.28</td>
<td>0.46</td>
<td>0.73</td>
</tr>
<tr>
<td>Fingerprinting and decryption (s)</td>
<td>0.15</td>
<td>0.25</td>
<td>0.42</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 1

Time complexity of the proposed JFDA method.
Table 1 presents the average performance of the proposed JFDA method for a $512 \times 512$ image. The "Dimension" in the first row indicates the segment size of the partitioned pixels (i.e., the number of columns in the matrix $A$ in Eq. (19)). From the table, we can observe that both the encryption and decryption processes of the proposed method are very efficient, which imply that our method is suitable for real-time secure image transmission and distribution.

5.2. JFDA in the frequency domain

The second kind of experiments was conducted in the frequency domain. In this case, the compression method used here was in accordance with the JPEG standard. Fig. 14 shows the four JPEG images with quality factor 75%. Assume that the fingerprint is a binary sequence. In our method, the first 32 AC coefficients in a $8 \times 8$ block in the zigzag order were selected for encryption since adjusting the DC value easily gives rise to artifacts. The encrypted images are shown in Fig. 15, conveying the scrambled property in the context of encryption.

In contrast to the proposed method, Kundur et al.’s [14] encrypted the media by modifying the signs of DCT coefficients as shown in Fig. 16, where, with the exception of the DC value, each sign of the coefficients in a $8 \times 8$ block is flipped. From Fig. 16, it is obvious that the encrypted images are all still recognizable, leading to content leakage. Moreover, these blurred images may be recovered by using image restoration techniques.

5.3. Performance comparisons

The comparisons of the proposed method with other methods are summarized in Table 2. Let $M$ denote the number of coefficients to be encrypted, and let $L$ represent the size of the lookup table. In most cases, $L$ is much smaller than $M$. In the sign bit plane encryption methods, Kundur et al.’s method [14] and Karthik and Hatzinakos’s method [13] easily suffer from tampering attacks on the user’s fingerprint by altering signs that would not affect the perceptual quality significantly, such as modifying the coefficients in blocks with high texture entropy. Although Karthik and Hatzinakos supplement Kundur et al.’s method with perceptual models, the consequences of imperceptible fingerprinting are more complicated computations and more severe...
content leakage. Furthermore, the binary operations on sign bit encryption give rise to the difficulty in balancing the image fidelity against fingerprinting when the number of users becomes large since the magnitudes of the un-deciphered coefficients with opposite sign would be completely reflected in perceptual quality. In contrast, in our method, Lemma et al.'s method, Adelsbach et al.'s method, and Celik et al.'s method, which can be regarded as based on table lookup operations, the image fidelity can be easily maintained since the fingerprint design in these lookup-table-based methods is independent of the host image and the magnitude of the fingerprint can only cause a small disturbance to perceptual quality. Nevertheless, in Lemma et al.'s method [15], Adelsbach et al.'s method [1], and Celik et al.'s method [6], when perceptual models are applied, content leakage is natural/common in smooth regions since coefficients to be encrypted are isolated from each other and the coefficients in the smooth region tend to be unchanged in order to preserve the perceptual quality. Note that the result in Fig. 12 of Celik et al.'s method did not show the content leakage because none of perceptual methods are applied. In contrast, the proposed method will not cause content leakage in the encryption domain even when the perceptual modeling is considered because our encryption function as indicated in Eq. (9) is associated with a group of pixels/coefficients. In sum, only our method can resist both tampering attacks and content leakage.

6. Conclusions

In this paper, we have presented a joint privacy-preserving fingerprinting, decryption, and authentication (JFDA) method for secure transmission and distribution of digital images. Our framework is significantly different from the earlier works because (1) our proposed privacy-preserving fingerprinting strategy allows embedding fingerprints in the encryption domain, (2) the threat of unreliable receiver, relatively unexplored in the literature, is taken into consideration and solved in our method, (3) the threat of a tampering attack on ciphertext can be avoided by employing our proposed pre-authentication mechanism, (4) a new idea of post-authentication in JFDA is proposed to oppose a stronger attack, namely an attack operated on plaintext directly, which prompts tampering detection against the intruder who attends to crack the unreliable receiver, and (5) all the operations in JFDA are accomplished with one breath and correlated with each other, instead of a combination of methods stacked on top of each other.

Acknowledgment

This work was supported by National Science Council Taiwan, under Grants MOST 103-2221-E-468-007-MY2. We especially thank Prof. Ya-Fen Chang, Dept. of Computer Science and Information Engineering, National Taichung University of Science and Technology, for giving us precious comments on this paper.

References