Soft Comput (2009) 13:383–391
DOI 10.1007/s00500-008-0326-8

FOCUS

Genetic fingerprinting for copyright protection of multicast media

Hsiang-Cheh Huang · Yueh-Hong Chen

Published online: 4 June 2008

Abstract Generally speaking, comparing to the unicast or broadcast methods, it is more efficient to transmit multimedia data via the multicast method to massive users. However, the ease of delivery of multimedia data may cause the copyright of such multimedia to be easily infringed upon, and the fingerprinting scheme is one of effective means for conquering this problem. Fingerprint embedding process often generates the multimedia contents into many different versions, which have to be transmitted via the unicast method. In this paper, we propose a new genetic fingerprinting scheme for copyright protection of multicast video. In this method, the encryption and decryption keys, which aim at scrambling and descrambling multimedia contents, are first produced with genetic algorithms. Next, multimedia data are then encrypted and multicast to all users. At the same time, a secure channel is employed to unicast a designated decryption key to each user. When a user deploys the designated key to decrypt the received data, a corresponding fingerprint would be embedded into the contents. Once upon the reception of the fingerprinted multimedia content, the embedded fingerprint can be extracted shortly, and the copyright can be confirmed and assured. Experimental results demonstrate that the proposed method can transmit multimedia data to clients effectively and cause only a slight degradation in perceptual quality.

1 Introduction

Lots of Internet e-learning and entertainment applications contribute to the importance of Internet TV and video/multimedia on demand (VOD/MOD) services besides the existing text- or image-based retrieval applications. Nevertheless, the Internet TV and VOD/MOD applications have encountered some difficulties, including (1) the dramatic increase in bandwidth requirement for transmitting multimedia data, especially video data, and (2) the rampant pirates problem. These are burning issues to the video-based Internet applications.

There is a growing interest in digital rights protection (DRP) and digital rights management (DRM) researches because of their potential capabilities to prevent media content from being pirated (Cox et al. 2001; Pan et al. 2004a, 2007). Besides the conventional schemes by employing cryptographic schemes, or data encryption, to protect data, digital watermarking is a commonly used technique for DRP and DRM (Pan et al. 2004b). As people know, after completing the encryption process, the output looks like noisy pattern (Chang et al. 2007), which may easily cause the suspicion by the eavesdroppers. From another perspective, watermarking is a way to secretly and imperceptibly embed the specific information, called the watermark, into the original multimedia contents. After performing watermarking, the original multimedia content and its corresponding output look very similar, hence reducing the possibility by the suspicion from the eavesdroppers.

When an embedded watermark is associated with a particular user, it can be considered as a fingerprint. Once the fingerprinted media have been illegally distributed, the corresponding user could be easily traced back from the re-distributed versions. Unfortunately, the fingerprint embedding process often makes the media content into many different outputs, which have to be transmitted via the unicast method. Generally speaking, it is more efficient to transmit a unique media via the multicast method to massive users (Chuang and Sirbu 2001; Zhao and Liu 2004). It becomes

H.-C. Huang (✉)
National University of Kaohsiung, Kaohsiung, Taiwan, R.O.C
e-mail: huang.hc@gmail.com

Y.-H. Chen
National Chiao Tung University, Hsinchu, Taiwan, R.O.C

quite important in the field of video-based applications to efficiently transmit video data embedded with fingerprints to all the users. Several different techniques have been proposed to tackle the problem above. A brief review of those research efforts is described as follows.

According to the time when a fingerprint is embedded into a video, most of proposed methods can be classified into one of the following cases:

1. transmitter-side fingerprint embedding,
2. receiver-side fingerprint embedding,
3. intermediate-node fingerprint embedding, and
4. joint fingerprinting and decryption.

Each of these cases will be briefly introduced and discussed in the following sections.

1.1 Transmitter-side fingerprint embedding

The goal for transmitter-side fingerprint embedding schemes is to embed users' fingerprints into the video to be multicast at server side. Next, the video is scrambled such that each user can only descramble his/her own fingerprinted video. Wu and Wu (1997) proposed a technique that multicast most of the video and unicast a portion of the video with unique fingerprints. When a larger percentage of the video is chosen to be fingerprinted, scrambled, and unicast, the security of transmitted video gets enhanced, but the efficiency of the protocol begins to resemble that of the simple unicast model. Boneh and Shaw (1998) presented a method to distribute the fingerprinted copies of digital data with the scrambling approach, also called the encryption approach therein. In their approaches, only two watermarked versions of video were needed, and video data were transmitted in a multicast manner. However, the bandwidth requirement was almost doubled over that of the normal multicast case. The strategy was also adopted by some other methods on frames (Chu et al. 2002), packets (Parviainen and Parnes 2001), and segments of video streams (Thanos 2001).

1.2 Receiver-side fingerprint embedding

The architecture of this type of embedding methods was initially introduced in Macq and Quisquater (1995), and more recently discussion were appeared in Hartung and Girod (1997) and Bloom (2003). In this scheme, a video is protected to produce a scrambled content, also called an encrypted content, and it is then multicast to users from the server side. At the receiver side, the encrypted video is decrypted and fingerprinted with an unique mark by a decryption operator. For security reasons, tamper proof hardware must be employed in order to protect the purely decrypted host media content from eavesdropping. However, tamper proof hardware is difficult to build and it is an interesting, open research topic till now.

1.3 Intermediate-node fingerprint embedding

This method proposed to distribute a fingerprinting process over a set of intermediate nodes such as routers (Judge and Ammar 2002). Thus, by tracing the routing paths, the owner of a specific fingerprinted copy can be identified. However, this method creates a different set of challenges, such as vulnerability to intermediate node, compromise and susceptibility to standard network congestion, and packet dropping (Luh and Kundur 2004). Hence, these are also interesting, open research problems.

1.4 Joint fingerprinting and decryption

The Joint fingerprinting and decryption (JFD) method, proposed by Kunder and Karthik (2004), integrates the decryption and fingerprinting processes at the client side. In the method, a server is allowed to multicast only one encrypted video to all customers, and to unicast a designated decryption key to a specific user. The user can only decrypt a portion of the video data with the decryption key, and the video data that remain encrypted constitutes a fingerprint. However, in the method in Kunder and Karthik (2004), they perform the fingerprinting process in the discrete cosine transform (DCT) domain, and DCT coefficients are partitioned into subsets to represent binary strings, that is, the user ID. The bandwidth required for unicasting decryption keys would be increased with the increasing number of subsets.

In this paper, we propose a new JFD method based on genetic algorithms (GA). The method first generates encryption and decryption keys with GA. Multimedia data is then encrypted and multicast to all users. At the same time, a secure channel is used to unicast a designated decryption key to each user. When a user employs the designated key to decrypt the received video, a designated fingerprint would be embedded into the video. Three features of the proposed method make itself a suitable approach for protecting copyright of multicast media on the Internet. These features are:

1. Adaptive fingerprinting. The transform domain coefficients that are suitable for embedding fingerprints are selected with genetic algorithm. Therefore, different criteria can be adopted for different applications with the suitable design of fitness function.
2. Effective transmission. Only the decryption keys, that is, the random seeds in our implementation, need to be delivered with the unicast method. All the remaining data, including encrypted contents and information related to decryption, can be transmitted with the multicast method.

3. Security and imperceptibility. While a video is encrypted, on the one hand, most of the transform domain coefficients are scrambled such that it has little or no commercial value. On the other hand, a decrypted video left only a few coefficients that are still encrypted, and a fingerprint, or the encrypted coefficients left in the decrypted video, causes only imperceptible degradation in video quality.

The rest of the paper is organized as follows. Section 2 introduces the concepts of genetic algorithms. Section 3 presents the scheme about how to encrypt a video frame and how to generate decryption keys with genetic algorithms. Section 4 discusses about the fingerprint detection method of the proposed scheme. Experimental results are presented in Sect. 5. Finally, in Sect. 6, we summarize the proposed method and draw a brief concluding remarks.

2 Brief descriptions of genetic algorithms

It is commonly seen that in a non-linear function with multiple variables, finding the maximum and minimum values is a difficult task by use of conventional optimization schemes. One scheme called the "genetic algorithm" (GA), based on the concept of natural genetics, is a directed random search technique. The exceptional contribution of this method was developed by Holland (1992) over the course of 1960s and 1970s, and finally popularized by Goldberg (1989).

In the genetic algorithms, the parameters are represented by an encoded binary string, called the "chromosome." And the elements in the binary strings, or the "genes," are adjusted to minimize or maximize the fitness value. The fitness function is defined by algorithm designers, with the goal of optimizing the outcome for the specific application, for instance, the conventional traveling salesman problem (TSP) (Gen and Cheng 1997) and more recently, applications for watermarking (Pan et al. 2004c; Huang et al. 2007). It generates its fitness value, which is composed of multiple variables to be optimized by GA. For every iteration in GA, a pre-determined number of chromosomes will correspondingly produce fitness values.

GA is mainly composed of three building blocks, namely, *selection*, *crossover*, and *mutation*. Figure 1 demonstrates the flow chart for a typical binary GA. It begins by defining the parameters for optimization, the fitness function provided by algorithm designers, and the corresponding fitness value, and it ends by testing for convergence. According to the applications for optimization, designers need to carefully define the necessary elements for training with GA.

The three major building blocks in Fig. 1, or the core components for GA, can be briefly depicted as follows.

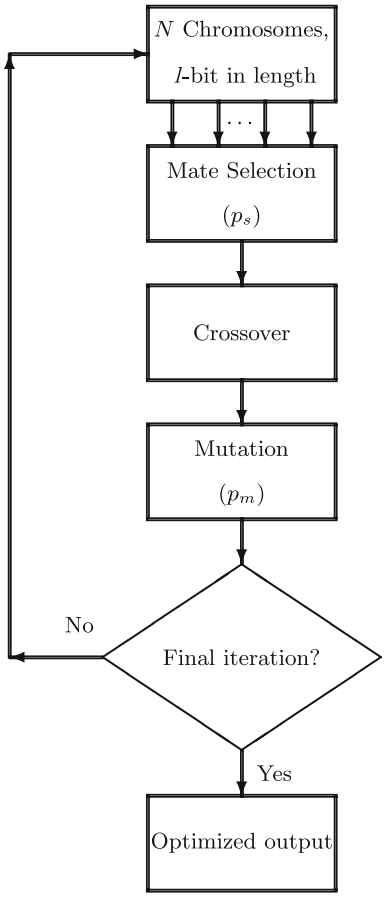

Fig. 1 Flowchart of a genetic algorithm

Mate selection: Assume that there are N chromosomes, and each has the length of l-bit for training in GA. A large portion of the chromosomes with low fitness values is discarded through this natural selection step. Algorithm designers need to provide a parameter called the "selection rate", p_s, for training in GA. The selection rate defines the portion of chromosomes with high fitness values that can be survived into the next training iteration. Consequently, there are $(N \cdot p_s)$ chromosomes that can be survived into the next iteration.

Crossover: Crossover is the first way that a GA explores a fitness surface. Two among the $(N \cdot p_s)$ survived chromosomes are chosen from the current training iteration to produce two new offsprings. A crossover point is selected, and the fractions of each chromosome after the crossover point are exchanged, and two new chromosomes are produced.

Mutation: Mutation is the second way that a GA explores a fitness surface. The mutation procedure is accomplished by intentionally flipping the bit values at the chosen positions. It introduces traits not in the original individuals, and keeps GA from converging too fast. The fraction between the number of chosen positions, and

the total lengths of chromosomes, is called the mutation rate, p_m. Consequently, a total of $(N \cdot l \cdot p_m)$ bits are intentionally flipped during the mutation procedure. The pre-determined mutation rate should be low. Most mutations deteriorate the fitness of an individual, however, the occasional improvement of the fitness adds diversity and strengthens the individual.

After obtaining the fundamental concepts in GA, we are able to design an optimized, DCT-based fingerprinting system with the aid of GA, and to evaluate the fitness function in addition to the terminating criteria with the natural selection, crossover, and mutation operations in a reasonable way (Gen and Cheng 1997).

3 Genetic fingerprinting scheme

The proposed genetic fingerprinting scheme is mainly applied to transmit and protect media content for a multicast scenario. In this section, the multicast scenario and a performance metric for multicast fingerprinting schemes are introduced. Next, the embedding and detecting methods for the proposed GA fingerprinting scheme are also described.

3.1 Performance evaluation for multicast fingerprinting schemes

Multicast transmission described in this paper is similar to that in Kunder and Karthik (2004). First, we assume that there is only a public channel between a media server and all clients. Data transmitted with the public channel can be received by all of the clients simultaneously. In other words, sending data directly with the public channel is called the broadcast transmission. If the server needs to send secret data to a specific client, the data should be encrypted with the secret key associated with the client before transmission. All clients' secret keys are delivered with a secure channel. Encrypting and transmitting data to a specific client is referred to as the unicast transmission. We use the term multicasting for the transmission of data using the combination of the unicast and broadcast methods. To speak qualitatively, the transmission of media content is efficient if it incorporates both the broadcast and unicast methods such that the broadcast channel is used only a few times, while the unicast channel is seldom employed (Kunder and Karthik 2004). From quantitative point of view, the efficiency of a distribution method is measured; it relates to the purely naive broadcasting scenario and it can be defined by the ratio given in Eq. (1),

$$\eta = \frac{m_D}{m_0}, \tag{1}$$

where m_D is a value proportional to the bandwidth used by a fingerprinting scheme, and m_0 is a value proportional to the bandwidth used in the unicast channel case. In particular, m_0 is defined to be the number of times the public channel is used when the fingerprinted content is sent to each user respectively, and m_D is the number of times the public channel is used by the fingerprinting scheme. We expect that $0 \leq \eta \leq 1$. In addition, for two fingerprinting methods, if Method 1 is more efficient than Method 2, we refer to $\eta_1 < \eta_2$.

3.2 Genetic fingerprinting embedding

As discussed in Sect. 3.1, it is much more economic to use the broadcast channel than to use the unicast channel to transmit data. Therefore, the proposed scheme employs the broadcast channel to deliver the encrypted media content and decryption-related data required by all clients. To verify the applicability of the proposed genetic fingerprinting algorithm, we use image data to serve as the multimedia content. It is directly extendable to video or other media formats under consideration with only minor modifications.

In our scheme, a method to modify the frequency domain coefficients of images is required for encrypting and fingerprinting the images. Any method having following properties, along with the watermarking requirements, can be adopted. These properties are: (1) robustness, (2) imperceptibility, and (3) reversibility. The first two properties are similar to the requirements of a general watermarking system (Pan et al. 2007; Huang et al. 2007; Shieh et al. 2004). The third property is that the modification made by the method should be reversible such that clients can decrypt the image. In this paper, we invert the sign bits of some DCT coefficients to encrypt the image. The DCT is applied to the entire image, and unlike conventional schemes, it is not divided into 8×8 blocks for reasons of perceptibility of the fingerprint. Next, decryption keys, that is, the random seeds in our implementation, are provided to clients through the unicast channel. The encrypted image and decryption-related information are broadcast to all clients. By combining decryption keys and decryption-related information, all clients have the ability to decrypt the encrypted coefficients to the most, and to obtain their own fingerprinted image.

Figure 2 demonstrates the process of fingerprint embedding. First, we randomly select a decryption key for each client. The decryption key is regarded as a random seed at the client side. Then, GA is applied to choose the suitable DCT coefficients to be encrypted, and to generate a coefficient decryption table. The coefficient decryption table is shared among all clients. Finally, the encrypted image and the coefficient decryption table are delivered to all clients with the broadcast method, while the decryption keys are sent to each client with the unicast method.

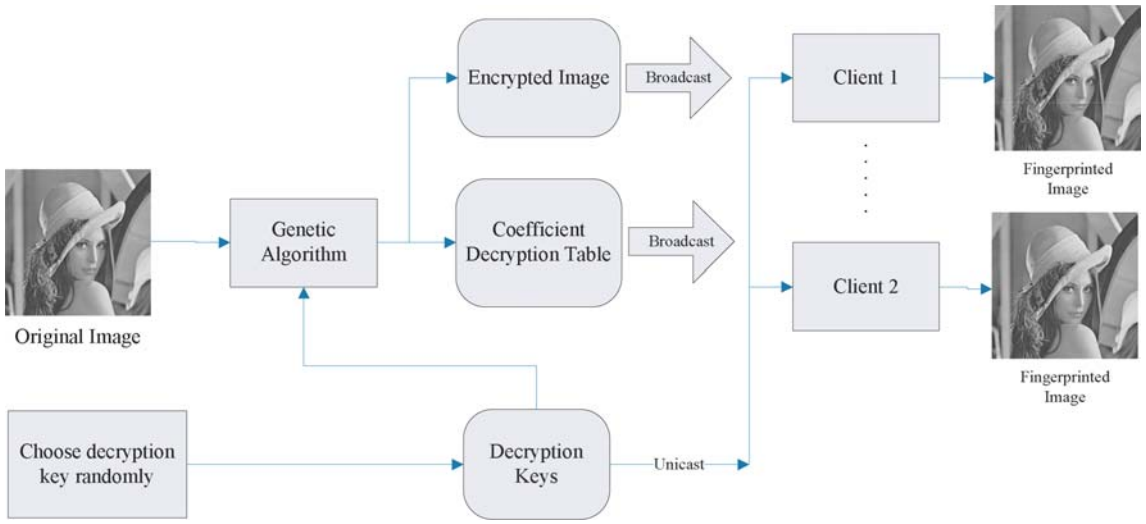

Fig. 2 The process of fingerprint embedding in this paper

Entry	Coefficient #1	· · ·	Coefficient # m
1	44	· · ·	−1
2	173	· · ·	35
3	62	· · ·	44

Fig. 3 The coefficient decryption table

An example of the coefficient decryption table is depicted in Fig. 3. An entry of the coefficient decryption table contains at most m coefficient indices. Each index corresponds to a DCT coefficient. A larger coefficient decryption table requires the more bandwidth for transmission, and consequently adds variety to the encryption and decryption processes. At the client side, the decryption key is used to serve as a random seed to select K entries from the coefficient decryption table randomly, and then to decrypt coefficients indexed by the selected entries. Some encrypted coefficients will not be decrypted and form a fingerprint for the user.

Figure 4 presents another example for the decryption and fingerprinting processes. On the left side in Fig. 4, suppose that an image with the size of 3×3 is to be protected. Both the encrypted DCT coefficients and the coefficient decryption table are ready to be broadcast to all the clients by the server. The encrypted DCT coefficients are represented by dark blocks, namely, coefficients 2, 3, 4, 6, 8 are encrypted at the server side. On the right side in Fig. 4, suppose that two clients have received the encrypted image. With their own decryption keys, the two clients choose three entries from the coefficient decryption table, and decrypt the six corresponding DCT coefficients. Since the clients do not have the knowledge about the exact combination of coefficients that are encrypted at the server side, some coefficients may be left unchanged during the decryption process. On the one hand, some encrypted coefficients that are not selected are still encrypted. On the other hand, if some coefficient is not encrypted by the server side, it may become fingerprinted at the client side due to the coefficient decryption table. We can see that for Coefficient 7 for Client 1, and for Coefficient 9 for Client 2, both of them present this phenomena.

To speak more precisely, let X be the candidate coefficients to be encrypted in the original image, and let $\hat{X}$ be the candidate coefficients extracted from fingerprinted image of client u. The fingerprint of the client u can be calculated with Eq. (2),

$$F_u(i) = \frac{1}{2}\left(1 + \text{sign}(X(i)) \cdot \text{sign}(\hat{X}(i))\right), \quad 1 \le i \le L, \quad (2)$$

where

$$\text{sign}(x) = \begin{cases} 1, & x \ge 0; \\ -1, & x < 0; \end{cases}$$

and L is the number of candidate coefficients. Thus, the fingerprint of the client is a binary string. If an encrypted coefficient is not decrypted, the corresponding bit is equal to 1; otherwise, it is equal to 0.

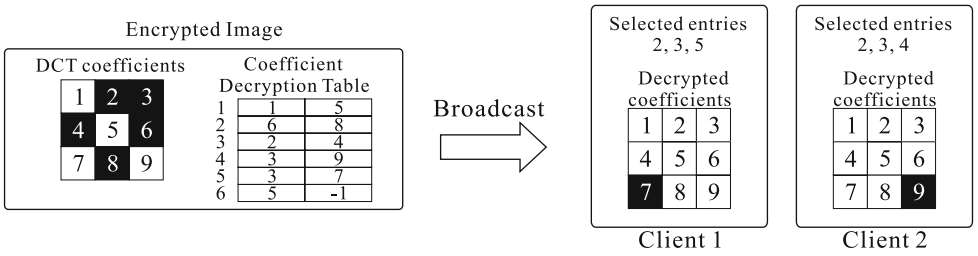

Fig. 4 An example for the decryption and fingerprinting processes of the proposed method

Since the decryption keys associated with the clients are chosen randomly, the genetic algorithm is used to select proper coefficients for encryption and to generate coefficient decryption table such that the client will leave some encrypted coefficients to form a fingerprint. Therefore, chromosomes of the genetic algorithm consist of a binary string and an integer array. The length of the binary string is equal to the length of X. When a bit is equal to '1' in the string, it indicates that the corresponding coefficient should be encrypted at the server side; and when the bit is equal to '0' the coefficient will be left unchanged. The integer array represents a coefficient decryption table. The value of each integer is regarded as the index of a coefficient to be decrypted, and a special value -1 is used to indicate that this table cell does not points to any coefficient.

From the discussions above, we observe that there are lots of flexibilities for implementing encryption and fingerprinting, and GA is suitable for conquering this task with a properly designed fitness function. For designing the fitness function, we consider the following goals.

1. The original image and the encrypted one should be as dissimilar as possible.
2. The correctly decrypted images corresponding to each client should be as similar to the original image as possible.
3. Different fingerprints should be as diverse as possible in order to differentiate the specific client.
4. The statistical difference between different fingerprints should be as apart as possible.

Therefore, the fitness function proposed in training with the genetic algorithm, containing four different parts, is depicted in Eq. (3),

$$\begin{aligned} \text{fitness} = &-\omega_1 \cdot \text{PSNR}_e \\ &+\omega_2 \cdot \sum_{u=1}^{U} \text{PSNR}_u \\ &-\omega_3 \cdot \sum_{i=1}^{U} \sum_{\substack{j=1 \\ j \neq i}}^{U} \text{sim}(F_i, F_j) \\ &+\omega_4 \cdot \sum_{i=1}^{U} \sum_{\substack{j=1 \\ j \neq i}}^{U} \text{diff}(F_i, F_j), \end{aligned} \tag{3}$$

where $\omega_i,\ i = 1, 2, 3, 4$, are weighting factors for each term, U is the number of clients. In the first two terms, corresponding to the image quality, PSNR_e is the peak signal-to-noise ratio (PSNR) value of an encrypted image, and PSNR_u is the PSNR value of the decrypted image of client u. Next, for the remaining two terms relating to the fingerprint, and in order to reduce the computational complexity, the similarity and difference measures between different fingerprints, sim(•) and diff(•), are defined as follows,

$$\text{sim}(F_i, F_j) = \begin{cases} 1, \text{ if } \frac{1}{L}\left(F_i \cdot F_j\right) > T, \\ 0, \text{ otherwise;} \end{cases} \tag{4}$$

$$\text{diff}(F_i, F_j) = \begin{cases} 1, \text{ if } \frac{1}{L}\left(F_i \cdot F_j\right) < t, \\ 0, \text{ otherwise;} \end{cases} \tag{5}$$

where F_i, F_j denotes the fingerprint i and j. T and t are the corresponding, pre-defined thresholds. By doing so, we are ready to maximize the fitness value in Eq. (3).

With the fitness function, we hope to find an encryption manner to degrade the visual quality of the encrypted image, and to enhance the quality of clients' decrypted images. Moreover, the fingerprints of any two clients should be only partly different to prevent from a comparison attack. We can also take into account the robustness of fingerprints to certain image processing methods. This would make fingerprints to be placed in more secure coefficients. Consequently, with GA described in Sect. 2, it can be employed to find proper coefficients to encrypt and form a coefficient decryption table. Since the decryption key associated with a specific client is only a random seed, the bandwidth required to send a decryption key can almost be omitted. Hence, by using a coefficient decryption table with a proper size, we can distribute fingerprinted image efficiently.

4 Fingerprint detection

When client u decrypts an encrypted image, a fingerprint F_u is embedded into the image immediately. If a redistributed copy of the image is obtained, we can detect the fingerprint F_u with Eq. (6),

$$\text{similarity} = \frac{1}{L}\left(F_u \cdot S\right), \tag{6}$$

and

$$S(i) = \frac{1}{2}\left(1 + \text{sign}(X(i)) \cdot \text{sign}(\tilde{X}(i))\right), \quad 1 \le i \le L,$$

where $\tilde{X}$ is the candidate coefficients for encryption, extracted from the redistributed image. If the similarity calculated by Eq. (6) is larger then a pre-defined threshold T, the redistributed image is considered to be embedded with the fingerprint F_u.

5 Experimental results

In this section, we present some experimental results to evaluate the performance of the proposed method. A 256×256, 8 bit/pixel gray scale image, `Lena`, is used as the test image in all experiments. This test image will be encrypted and transmitted to 5 different clients. 15,000 DCT coefficients in middle frequency are chosen as candidate coefficients for encryption, that is, $L = 15,000$. Coefficient decryption table contains 15,000 entries, and each entry can keep 2 coefficient indices at most. After receiving the decryption key, the client will randomly select 70% entries of the coefficient decryption table, and will decrypt the coefficients indexed. Four weighting factors in the fitness function are assigned to be $\omega_1 = 0.5$, $\omega_2 = 1$, $\omega_3 = 5$, and $\omega_4 = 5$. Since the coefficient modification method in this paper is inherently robust against image processing operations, the fitness function does not take the robustness measure into consideration. Even though there is no robustness measure in the fitness function, we will still evaluate the robustness of the proposed fingerprinting scheme with general image processing operations.

First, the genetic algorithm is applied to find out proper coefficients to be encrypted, and a corresponding coefficient decryption table for the five clients can be generated. Corresponding to Sect. 2 and Fig. 1, in our experiments, the number of chromosomes is $N = 200$, the length of each chromosome is $l = 45,000$, the selection and mutation rates are set to $p_s = 0.5$ and $p_m = 0.04$, respectively, and the number of training iteration is 450. Figure 5 depicts the fitness value of the best and the worst chromosomes in all the 450 generations. The resulting image encrypted by the best chromosome in GA is shown in Fig. 6, with the PSNR = 25.24 dB. The fingerprinted image with the lowest image quality among five clients is illustrated in Fig. 7, with the PSNR = 43.22 dB. Figure 8 demonstrates the resulting image decrypted with a randomly generated decryption key, which does not belong to the five clients. And we can see that the randomly decrypted image has poor image quality and is not suitable for commercial use. As shown in Fig 8, after the encryption process, from subjective point of view, the image content is still visible, but the image quality is too low to have commercial value. After decryption process, although the decrypted image contains a fingerprint, it is imperceptible to human eyes. As depicted in Table 1, PSNR values of all fingerprinted images are higher than 43 dB, and the one with the lowest PSNR value is illustrated in Fig. 7, with acceptable quality from subjective point of view. Moreover, based on our design methodology in algorithm, any two fingerprints are only partly different, so the collusion attack can be prevented.

In the second set of experiments, five fingerprinted images are attacked with three general image-processing methods. They are: (1) low pass filtering, (2) high pass filtering, and

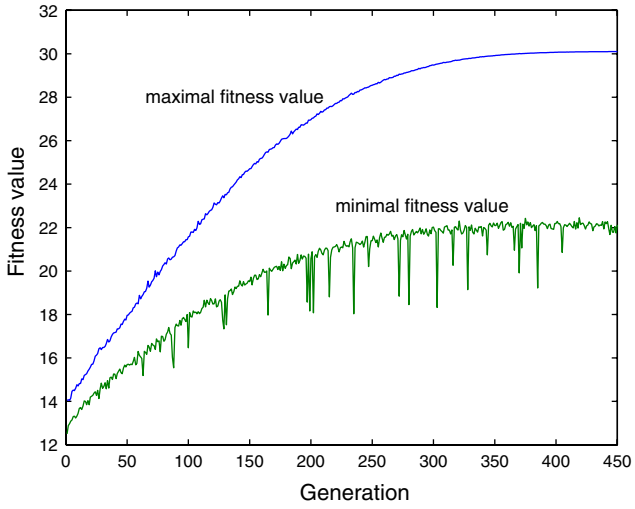

Fig. 5 Fitness values of best and worst chromosomes in 450 generations

Fig. 6 The encrypted image, PSNR = 25.24 dB

Fig. 7 The image with the worst quality among 5 decrypted and fingerprinted images, PSNR = 43.22 dB

Fig. 8 The image decrypted with a random key, PSNR = 28.64 dB

(3) JPEG compression. The detection values are presented in Table 2. From Table 2, it can be seen that although the fitness function does not take these attacks into account, the fingerprints are still robust to these image processing schemes applied intentionally, because the proposed algorithm is inherently robust to such processing. Thus, the fingerprint can hardly be removed unless the image quality is severely degraded.

Table 1 The PSNR values of five fingerprinted images, corresponding to five clients

Client number	PSNR for fingerprinted image (dB)
1	43.35
2	43.42
3	43.31
4	43.22
5	43.54

Table 2 The detection values of fingerprints after various image processing operations

Client number	Low pass filtering	High pass filtering	JPEG compression, quality factor = 15
1	0.92	0.99	0.717
2	0.93	0.98	0.706
3	0.95	0.99	0.693
4	0.91	0.99	0.660
5	0.91	0.99	0.683

Finally, in our experiments, the encrypted `Lena` image is transmitted to five clients and then fingerprinted. As discussed in Sect. 3.1, the efficiency of our method is

$$\eta = \frac{m_D}{m_0} = \frac{1 + 0.9155}{5} = 0.3831,$$

where the term 0.9155 is the bandwidth required to broadcast coefficient decryption table. This result, 0.3831, is lower than the results presented in Boneh and Shaw (1998), Chu et al. (2002), Parviainen and Parnes (2001) and Thanos (2001), which have a $\eta = 0.4$. Moreover, the fingerprinted images have acceptable to good visual quality. Summing up, the proposed method is very suitable for the application needs to transmit fingerprinted media in broadcast environment.

6 Conclusion

In this paper, we propose a new genetic JFD method. The method randomly select decryption keys for clients, and then generates an encryption key and decryption-related information with genetic algorithms. Media data is then encrypted and multicast to all client. At the same time, a secure channel is used to unicast a designated decryption key to each client. When a client uses the designated key to decrypt received media data, a designated fingerprint will be embedded in the data correspondingly. The proposed method have three features:

1. Adaptive fingerprinting,
2. Effective transmission, and
3. Security and imperceptibility.

Experimental results demonstrate that the proposed method can transmit media data to clients effectively and cause only a slight degradation in perceptual quality. Moreover, the proposed method has the capability to resist some attack methods if an appropriate encryption method is adopted.

References

Bloom JA (2003) Security and rights management in digital cinema. In: 2003 International conference on multimedia and expo, pp 621–624

Boneh D, Shaw J (1998) Collusion-secure fingerprinting for digital data. IEEE Trans Inf Theory 44:1897–1905

Chang F-C, Huang H-C, Hang H-M (2007) Layered access control schemes on watermarked scalable media. J VLSI Signal Process Syst Signal Image Video Technol 49:443–455

Chu HH, Qiao L, Nahrstedt K (2002) A secure multicast protocol with copyright protection. ACM Comput Commun Rev 32:42–60

Chuang JC-I, Sirbu MA (2001) Pricing multicast communication: a cost-based approach. Telecommun Syst 17:281–297

Cox IJ, Miller ML, Bloom JA (2001) Digital watermarking: principles & practice. Morgan Kaufman, Los Altos

Gen M, Cheng R (1997) Genetic algorithms and engineering design. Wiley, New York

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading

Hartung F, Girod B (1997) Digital Watermarking of MPEG-2 Coded Video in the Bitstream Domain. In: 1997 IEEE international conference on acoustics, speech, and signal processing, pp 2621–2624

Holland JH (1992) Adaptation in natural and artificial systems. The MIT Press, Cambridge

Huang H-C, Pan JS, Huang YH, Wang FH, Huang K-C (2007) Progressive watermarking techniques using genetic algorithms. Circuits Syst Signal Process 26:671–687

Judge P, Ammar M (2002) WHIM: watermarking multicast video with a hierarchy of intermediaries. Comput Netw 39:699–712

Kunder D, Karthik K (2004) Video fingerprinting and encryption principles for digital rights management. Proc IEEE 92:918–932

Luh W, Kundur D (2004) Digital media fingerprinting: techniques and trends, Chap. 19, Multimedia Security Handbook. CRC, Boca Raton

Macq BM, Quisquater J-J (1995) Cryptology for digital TV broadcasting. Proc IEEE 83:944–957

Pan JS, Huang H-C, Jain LC (eds) (2004) Intelligent watermarking techniques. World Scientific Publishing Company, Singapore

Pan JS, Hsin YC, Huang H-C, Huang KC (2004) Robust image watermarking based on multiple description vector quantisation. Electron Lett 40:1409–1410

Pan JS, Sung MT, Huang H-C, Liao BY (2004) Robust VQ-based digital watermarking for the memoryless binary symmetric channel. IEICE Trans Fund Electron Commun Comput Sci E-87:1839–1841

Pan JS, Huang H-C, Jain LC, Fang WC (eds) (2007) Intelligent multimedia data hiding. Springer, Berlin

Parviainen R, Parnes P (2001) Large scale distributed watermarking of multicast media through encryption. In: IFIP conference proceedings on communications and multimedia security, pp 149–158

Shieh CS, Huang H-C, Wang FH, Pan JS (2004) Genetic watermarking based on transform domain techniques. Pattern Recognit 37:555–565

Thanos D (2001) COiN-Video: a model for the dissemination of copyrighted video streams over open networks. In: 4th International workshop on information hiding, pp 169–184

Wu T-L, Wu SF (1997) Selective encryption and watermarking of MPEG video. In: International conference on image science, systems and technology, pp 261–269

Zhao H, Liu KJR (2004) Bandwidth efficient fingerprint multicast for video streaming. In: 2004 IEEE international conference on acoustics, speech, and signal processing, pp 849–852