Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **3**, July **2014**

Accelerated Tracing Scheme for Gaussian Fingerprinting

Xin-Wei Li

School of Electrical Engineering and Automation
Henan Polytechnic University
Jiaozuo 454000, China
lixinwei@hpu.edu.cn

Kang Sun

School of Electrical Engineering and Automation
Henan Polytechnic University
Jiaozuo 454000, China
sunkang@hpu.edu.cn

Lei-Da Li

School of Information and Electrical Engineering
China University of Mining and Technology
Xuzhou 221116, China
Shanghai Key Laboratory of Integrate Administration Technologies for Information Security
Shanghai 200240, Chinaa
lileida@cumt.edu.cn

Received July, 2013; revised January, 2014

Abstract. *Gaussian fingerprinting has been widely used for its well performance of resisting to collusion and common signal processing attacks, however, its computational complexity during colluder tracing is too high. This paper proposes an accelerated tracing scheme to reduce the computational complexity. The key idea is that a compact index table is created to condense the computational burden, which is similar to hashing searching. The accelerated tracing includes two matching stages. In the first stage, the feature extracted from the suspicious fingerprint is compared with each feature in the index table which had been extracted from original fingerprints. A few suspicious candidates are filtered from fingerprints database according to the obtained results. In the second stage, the true colluders are identified from the derived candidates through comparing the suspicious fingerprint with them. Two reasons make the computational complexity be reduced. One is the feature dimensions are very low in the first stage, and the other is that only a few fingerprints participate in comparison in the second stage. Theory and simulation results simultaneously show that the computation time can be reduced to 5 percent to 50 percent while keeping the correct identification ratios invariant in face of collusion attacks and traditional signal processing attacks.*

Keywords: Feature, Index table, Hashing, Fast

1. **Introduction.** The rapid development of internet and computer technologies makes the multimedia data accessing become more and more convenient. On the other hand, a large number of multimedia data are copied and spread arbitrarily, which damaged the owners rights seriously. In order to trace the source of data leakage, digital fingerprints are embedded into multimedia works to label them. Similar to human fingerprints, digital

fingerprints are exclusive data labels. Generally the labels are composed of different data components. Illegal users can be traced according to the labels extracted from suspicious works which had been registered and distributed to them. As being a special watermarking, digital fingerprinting develops rapidly recently[1],[2],[3]. The embedded fingerprints are easily been weakened when they suffered from collusion attacks, where several fingerprinted copies are compared and operated to remove fingerprints. Due to the destructiveness, most existing literatures mainly focus on improving the performance of resisting collusion attacks for digital fingerprinting. Gaussian fingerprinting was favored in the past decades for its high robustness to several kinds of attacks, especially to the collusion attacks[1],[2]. Cox[1] proposed the spread spectrum watermarking first. Since then it was followed and developed all the time. Trappe et al.[2] proposed a binary tree detection algorithm which reduced the correlations during colluder tracing effectively. Wang et al.[4],[5] studied the robustness of

Gaussian fingerprinting comprehensively. Varna et al.[6] extend the Gaussian fingerprinting to multimedia compression domain in succession. Kuribayashi[7] studied the impact of rounding error on Gaussian fingerprinting. Li[8] proposed a blind detection algorithm combined Anti-Collusion Code with Gaussian fingerprinting, and he also studied the attacks abilities of some typical collusion on Gaussian fingerprinting[9]. The common in these literatures is that their studies all focused on Gaussian fingerprinting. However, low efficiency detection affected the Gaussian fingerprinting real-time tracing seriously. Each fingerprint is scanned during colluder tracing for traditional algorithm[1],[4]. The tracing process is very time-consumed when there are lots of fingerprints in the database. The binary tree detection way[2] divides the database into two subsets and calculates the similarities between suspicious fingerprint and the sum of subsets. Then the subsets whose correlation coefficients are higher than threshold are subdivided into two subsets again. The processes were carried out until there is only one fingerprint for the subsets in the recursive manner. The way can reduce the correlations sharply compared with the traditional way when the signal-noise-ratio is high, however, the correlations will increase even exceed that of traditional way when the signal-noise-ratio decrease. More important is that the recursive manner makes the process running time increase surprisingly. The reason is that transferring a large amount of data costs much memory and time for each recursion.

Hashing is a technique that translates the large scale data into limited-dimension data which represent the original data to carry out some task. Matching is possible even when the recognizable database objects have undergone transformations or when only partial information is present. It is a contraction mapping method to improve searching efficiency. For example, multimedia hashing is that extracting some unique features from images or videos to represent them for fast searching[10],[11],[12],[13]. At the same time, some literatures[14],[15],[16] studied the fast searching algorithms for binary sequences hashed from multimedia signal. Gionis[14] examined a scheme for approximate similarity search based on hashing. Its key idea is to hash the points from the database, and the purpose is to ensure that the collision probability is much higher for objects that are close to each other than for those that are far apart. Shakhnarovich[15] introduced an algorithm that learned a set of hashing functions that efficiently index examples in a way relevant to a particular estimation task. Esmaeili[16] proposed a fast approximate nearest neighbor search algorithm for the Hamming space. Especially Wang[17] proposed an feature matching algorithm, which narrowed the candidates scope with a fast manner, and then confirmed the nearest goal. Inspired by the idea of hashing and fast searching, the Gaussian fingerprints can be represented by their features to participate in matching. In this paper, we propose an accelerated tracing scheme for Gaussian fingerprinting to improve computational efficiency. The proposed tracing algorithm includes two matching stages. The core is creating an index table and searching colluders with it. The computational efficiency is improved greatly compared with the traditional way and binary tree detection. The rest of the paper is organized as follows. Section 2 describes the accelerated tracing scheme including the scheme framework, the construction of index table, the first matching process, the second matching process and their analysis. Simulation results including computational efficiency, the ability on resisting collusion attacks and common signal processing attacks are shown in Section 3. Finally conclusions are given in Section 4.

2. **Accelerated Tracing Scheme.** Considering the typicalness of Gaussian fingerprinting, the proposed scheme is described with it being object in the paper. The main idea of the scheme is that short features and finite candidates are employed to be compared. The framework of the scheme is shown in figure 1. It includes two stages, and the first stage is shown in the upper half part. Its goal is to select a few suspicious fingerprints. The second stage is shown in the lower half part, and its function is to capture the true colluders. Before fingerprints matching the index table should be created. The details for the processes of the index table creation, the first and second matching will be analyzed in following.

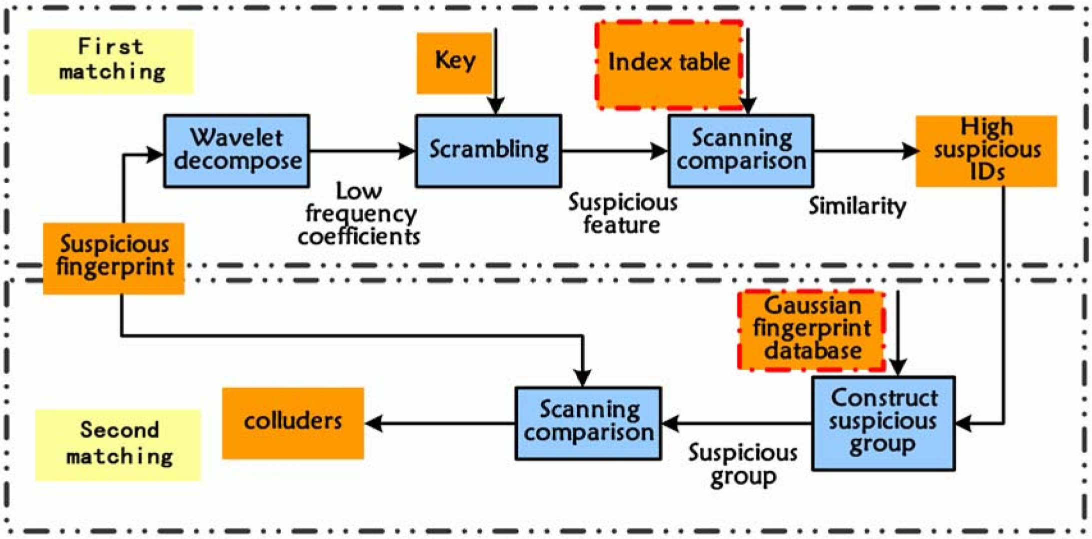

FIGURE 1. Framework of accelerated tracing

2.1. **The creation of index table.** The basic units of the index table are the feature vectors extracted from original fingerprints in essence. In order to reduce computational time and keep correct identification ratio invariant, the features must meet the compactness and robustness simultaneously. The features should be robust when they undergo attacks same as that the fingerprinted copies experience, such as collusion attacks, lossy compression, noise disturbing and filtering. To meet the premise of robustness, the features should be as short as possible. Experiments have shown that the low frequency wavelet coefficients for signal are compact and robust to the above-mentioned attacks. Therefore the index table creation is based on wavelet decomposition in this paper.

As shown in Figure 2, each Gaussian fingerprints in the database is decomposed with wavelet first, then some low frequency coefficients are scrambled according to the key created in advance to generate feature vector. When all the fingerprints are processed the index table is created. The introduction of scrambling can increase the scheme's security. The key is a random integer sequence whose values are from one to the length of features. The wavelet coefficients are adjusted according to the key's components. This way can avoid illegal users disturbing colluder tracing deliberately. The details for referred parameters setting are discussed in next section.

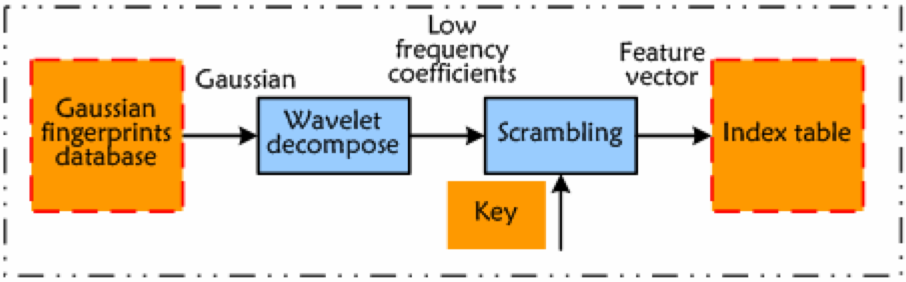

Figure 2. Framework of accelerated tracing

2.2. **Colluder tracing.** The multimedia works embedded fingerprints are called fingerprinted copies. They are distributed to users after being registered. The distributed copies may be conspired or attacked to generate an unauthorized copy. The fingerprint extracted from the unauthorized copy is called suspicious fingerprint. The details follows the process that in literature[4]. Similar to the creation process of the index table, the suspicious fingerprint is decomposed with wavelet. Before that the wavelet decomposition layer should be determined.

In order to select a suitable decomposition layer, we have done some experiments with the following parameters. The database is set to 5000 Gaussian fingerprints with 9216 dimensions. The required memory for the original database is 338M. These fingerprints are decomposed with Haar wavelet to generate the index table. 50 randomly selected fingerprints are averaged to generate a suspicious fingerprint. The decomposition layer is set to 4, 5 and 6 separately to compare the results. The results with 1000 rounds of experiments are listed in Table 1. Identification ratios mean that the probabilities of capturing the true colluders correctly. Time represents the consumed time for running 1000 rounds, and memory denotes the required memory for storing the index table.

In Table 1, the varying trends for dimensions and identification rations are both monotonic decreasing, and they are contrary to the decomposition layer. It is easy to understand that high dimensions vectors will cost more time and need more memory. Considering the robustness and compact the layer is set to 5 because the whole performance is better than the others. Wavelet decomposition layer is set to 5 for all simulations in this paper. Next some analyses are given to compare the computation complexity.

Suppose there are N fingerprints with n dimension in the original fingerprints database, and the i^{th} fingerprint is denoted by $\mathbf{f}_i(\mathrm{i} = 1, \cdots, \mathrm{n})$. For describing simply, $\mathbf{v}_i$ denotes the corresponding feature vector of the i^{th} fingerprint. Assume the decomposition layer is t, then the dimension of $\mathbf{v}_i$ is $\lfloor (n/2^t) \rfloor$, where $\lfloor (\cdot) \rfloor$ is a down rounding function. The traditional additive watermarking way is employed in this paper for representativeness. If the host vector is s, then the i^{th} fingerprinted signal is

$$\mathbf{y}_i = s + \alpha \mathbf{f}_i + \mathbf{d} \tag{1}$$

where α controls the embedding intensity, and $\mathbf{d}$ represents the noise. Assume that the previous K fingerprints are averaging colluded. Assume $\mathbf{f}'$ denotes the attacked fingerprint, it is expressed as follows

Table 1. Relations between decomposition layers and related parameters

Layer	Dimensions	Iden. Ratios	Time(Rs)	Memory
4	576	0.996	144.69s	21.1M
5	288	0.962	67.66s	10.5M
6	144	0.695	50.93s	5.28M

$$\mathbf{f}' = \frac{\alpha}{K} \sum_{i=1}^{K} \mathbf{f}_i + \mathbf{d} \tag{2}$$

Let $\mathbf{v}'$ denote the feature vector corresponding to $\mathbf{f}'$. The correlation coefficients between the two vectors are always employed to represent their approximation degree. Thus Equation (3) is used in the first scanning stage.

$$\mathbf{sim\_v}_i = \frac{\mathbf{v}_i^T \mathbf{v}'}{\|\mathbf{v}'\| \cdot \|\mathbf{v}_i\|} \tag{3}$$

where $\|\cdot\|$ is norm function, i = 1, · · ·, n. A list of correlation coefficients is obtained after scanning all the features. Then a few fingerprints corresponding to the top coefficients are selected for the second scanning. This stage can enclose a small group of fingerprints because there is a high possibility for them to participate in collusion. The simulation results shown in Section 3 will prove that.

Suppose that m (m is far less than n) fingerprints are selected as the candidates. The similarity calculations between $\mathbf{f}'$ and $\mathbf{f}_j$ can be completed fast following Equation (4) because there are only a few candidates.

$$\mathbf{sim\_f}_i = \frac{\mathbf{f}_i^T \mathbf{f}'}{\|\mathbf{f}'\| \cdot \|\mathbf{f}_i\|} \tag{4}$$

where j = 1, · · · , m. The selection of m relies on the goal of the scheme. If the goal is to catch single colluder m can be a very small number. If the goal is to catch all colluders then m should be a larger number. Once the goal is clear the users corresponding to the maximum similarities are identified to be colluders.

For the traditional scanning way, Equation (4) is calculated N correlations. There are $3n{+}2$ multiplications, 1 division, 2 square roots, and 3n − 3 additions for each correlation. In order to simplify the calculation, suppose there are 3n + 5 multiplications for each correlation. Then there are $N(3n + 5)$ multiplications in total for traditional scanning. On the other hand, in our scheme there are $N(3\lfloor n \rfloor /2^t + 5 + n(3n + 5)$ multiplications. If t is equal to 5 and m is smaller than one percent n, our multiplications will be reduced to about 5 percent to the traditional scanning.

3. Simulation Results and Analysis.

In this section, we study the behavior of our scheme. Matlab version 7.0 was employed to be the simulation environment. The computer was configured with Core-i3 CPU, 3.4GHz Clock Speed and 2G memory. As before, the suspicious signal and the orthogonal signals are assumed to be independent, and each of them is an $N = 9216$ point vector of identically distributed Gaussian samples. There are 5000 fingerprints in the database.

3.1. **Simulations for the two matchings.** In the simulations that follow, several fingerprints are colluded with averaging attack, and an equal energy Gaussian noise was added. The colluder numbers are set to 20, 30, 40, 50, 60 and 70 separately. Figure 3 shows the matching results for the first stage under different colluder numbers. The horizontal axis represents the colluder orders, and the vertical axis represents the coefficient orders to the whole 5000 correlation coefficients. The solid dots represent the top 10 coefficients. Figure 3(a) illustrates all 10 colluders can be captured in the first matching stage when there are 20 colluders. Even there are 70 colluders, 4 colluders can be delineated in the candidates group when m is set to 10. The rest subfigures can intensify this statement. Thus even scanning only 10 fingerprints in the second stage, some true colluders can be captured surely.

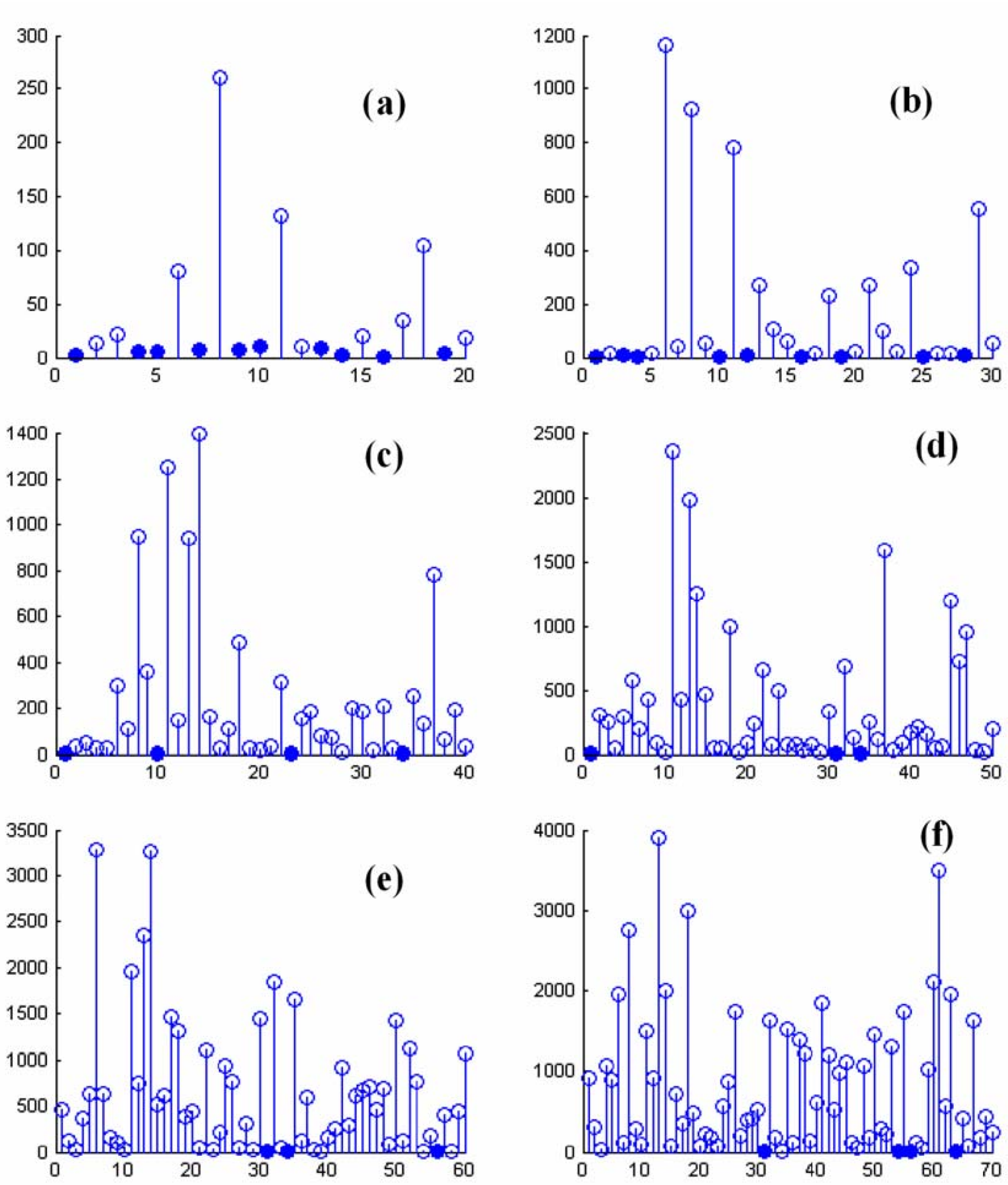

FIGURE 3. The similarity orders of first matching under different colluders (a) 20 (b) 30 (c) 40 (d) 50 (e) 60 (f) 70

Though Figure 3 shows the results of one random experiment, many tests show the similar results. In order to prove this statement, we have done 100 rounds simulations with setting different m. The false alarm rate is set to 0.0001. Figure 4 shows their average results. The vertical axis represents the ratios of capturing colluders to all. When m is set to 10 the curve descend sharply, but the captured colluders are still more. For example, the ratio 0.9525 corresponding to 10 colluders in the horizontal axis illustrates that average 9.525 colluders were captured. The ratio reduced to 0.1054 when colluders were 70, and 7.378 colluders were identified in average. In fact, when m is set to 10 more than 7 colluders can be captured correctly in the figure. It should be noted that this curve is a averaging result for that in Figure 3. The scenarios that m is set to 100, 400 and 600 are also shown. When 100 candidate fingerprints are scanned, at least about half colluders were captured. More than 75 percent colluders had been captured when m is set to 400, which is 0.8 percent to the original fingerprints in the database.

The higher m makes more colluders be captured, and that will cost more time. Figure 5 shows the consumed time corresponding to the simulations in Figure 4. The average time consumed are 0.064s, 0.11s, 0.234s, and 0.313s separately for the above four scenarios. The time is invariant for constant candidate groups.

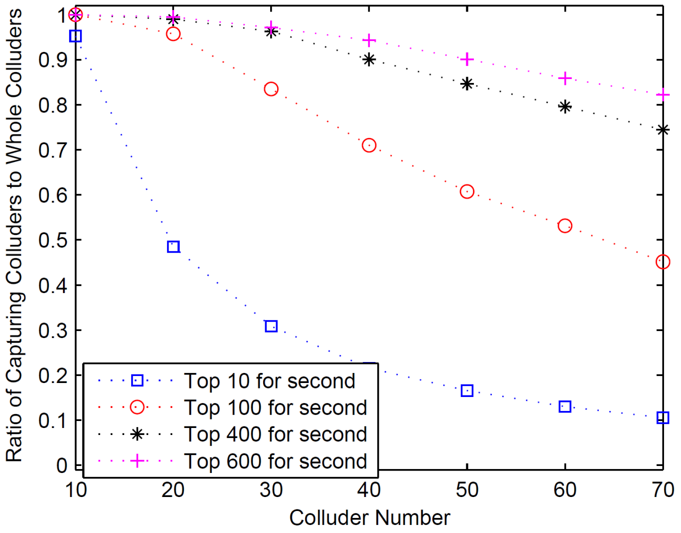

Figure 4. Ratios of correctly capturing colluders to whole colluders

In order to evaluate the advantages of our scheme to other algorithms, we selected the traditional detection way[4] and the famous binary tree detection way[2]. The traditional way is scanning each fingerprint in original database using Equation (4). The multiplications were listed in Table 2. For capturing single colluder, the multiplications ratio of proposed scheme to traditional way was 0.0334. The ratio of proposed scheme to binary tree detection was 6.8 for capturing single colluder. This number indicate the multiplications of binary tree is less, however, the binary tree detection consumed more time than ours in real running. The results in Figure 6 can prove this statement. For capturing 50 colluders, the multiplications ratio of proposed scheme to traditional way was 0.1314, and that was 0.5348 to binary tree detection. Even the goal is capturing more colluders our scheme will cost the least time.

The performance comparisons of robustness and cost time for the three ways are shown in Figure 6. Figure 6(a) shows the ratios of captured colluders to all. The red square line shows ours performance, and the blue circle line corresponds to the traditional way. It is clear that the ratios of traditional way are the highest. There was a slight reduction compared with the traditional way when m is set to 2000. In this situation, about 40 percent fingerprints are scanned in the second stage. Figure 6(b) shows the cost time of the three simulations in Figure 6(a). It is exciting that even m is 2000 the consumed time is less than half of that traditional way. The time can be reduced 50 percent of that traditional even in the worst case. The consumed time of binary tree detection is much higher than that of two previous schemes. It should be remained that all the false alarm rates were set to 0.0001. If the goal is designed to capture single colluder, the three schemes can identify the colluder with probability being equal to 1 under the above parameters setting. And the consumed time all reduced sharply.

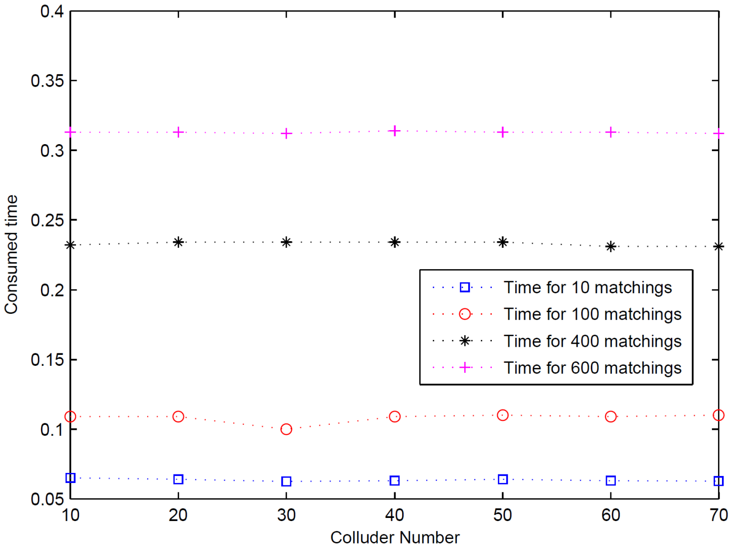

Figure 5. Consumed time for m=10,100,400 and 600

Table 2. Relations between decomposition layers and related parameters

	Multi.s (capture single)	**Multi.s (capture k=50)**
Traditional	$N(3n+5)$	$N(3n+5)$
Binary tree	$>= 2(\log_2 N)(3n+5)$	$>= 2k(\log_2 N)(3n+5)$
Proposed	$N(3(n/2^t)+5)+10(3n+5)$	$N(3(n/2^t)+5)+10k(3n+5)$
R.of Prop.to Trad.	0.0334	0.1314
R.of Prop.to B.T.	$<= 6.8$	$<= 0.5348$

3.2. **Simulations on real images.** In order to demonstrate the performance of our scheme on real images for tracing colluders, we used an additive spread spectrum watermarking scheme similar to that in [1], where the watermark was added to 8*8 block DCT coefficients. The detection of the watermark is a nonblind detection scenario performed with the knowledge of the host image via the detection statistics as shown in Equation (3) and (4). The 256*256 Lenna and Baboon images were used as the host signals for the fingerprints. The fingerprinted images have no visible artifacts with an average PSNR of 42.03 dB for Lenna and 41.35dB for Baboon. Its abilities of resisting to collusion attacks, JPEG compression attacks, adding noise attacks and filtering attacks were tested. All scenarios have been processed 100 rounds, and the data shown in the Figure 7, Figure 8 and Figure 9 are average results. The robustness to collusion attacks was shown in Figure 7. The vertical axis represents the captured true colluders. The red square curve shown in Figure 7 corresponds to the case that m is equal to 100. When there were 40 users conspired, 25 colluders were captured. When colluders are more than 50 the captured colluders reduced. Compared with the results in Figure 4 the ratios decreased slightly, however, this situation will be improved if we increase m. The blue circle curve represents the false captured colluders. It shown that while capturing the above colluders no innocent was captured.

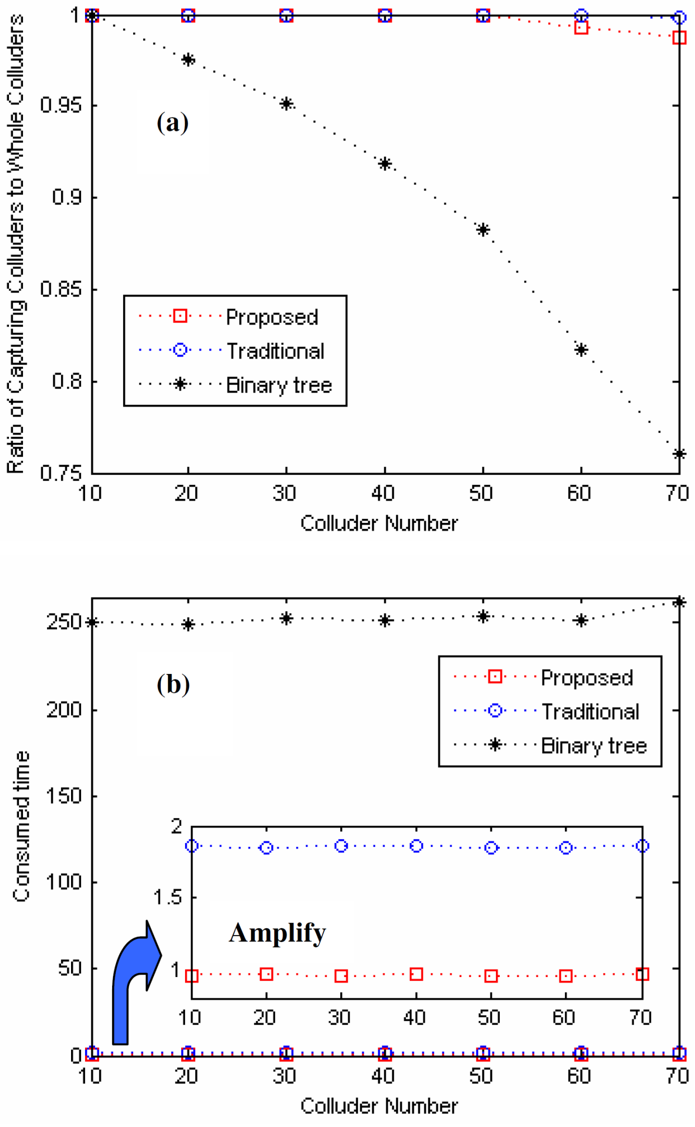

FIGURE 6. Consumed time for m=10,100,400 and 600

In order to test its robustness to lossy compression, an fingerprinted copy was JPEG compressed with different quality factors to be the suspicious copy. Figure 8 shows the detection results. The red curve illustrates the correct identification ratios. Even the compression quality factor is set to 10 the true colluder can be captured. The false ratios almost keeps 0 when the quality factor is larger than 10, revealing that no user was treated unjustly. The only varying is 0.29 percent corresponding to 10 on the horizontal axis. That means about 0.29 percent innocents would be identified colluders when the fingerprinted copy was deeply compressed.

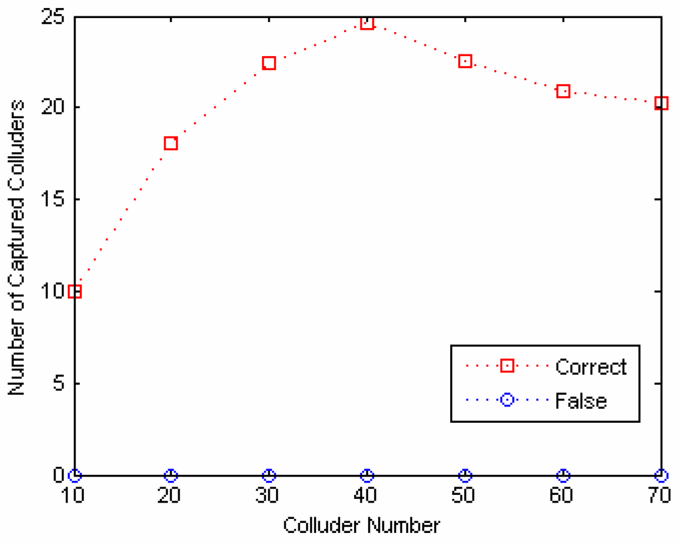

FIGURE 7. Colluder captured vs. averaging collusion attacks (m=100)

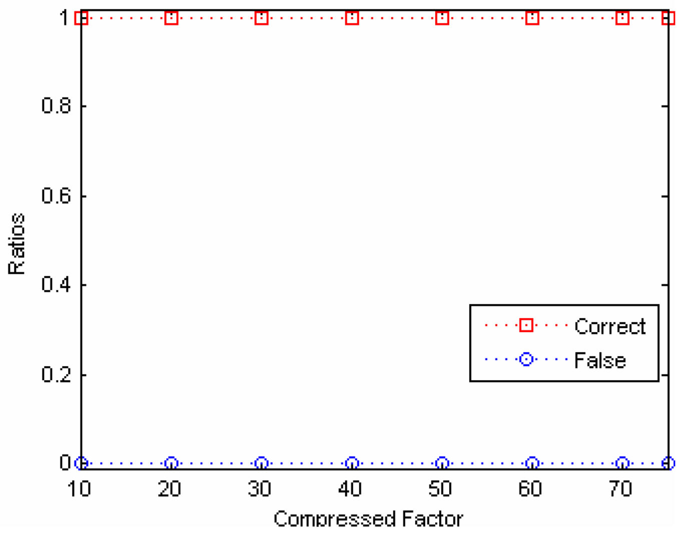

FIGURE 8. Robustness vs. JPEG compressed attacks

The simulation results for resisting noise added attacks are shown in Figure 9. Peppers-Salt and Gaussian noise were tested. There are some obvious reductions compared with the above figures, even for the case that Gaussian variance is higher than 0.02. In fact, when the Gaussian variance is equal to 0.02 the fingerprinted copies are too poor to use. So we can state that this scheme can resist noise added attacks effectively. The ability on resisting wiener and median filtering attacks were also tested. When the filter template was set to 3*3 or 5*5, the correct ratios all kept above 0.99, and the false ratios were all less than 0.01. So the accelerated tracing is also robust to filtering attacks. From the whole simulation results our scheme runs well in face of various challenges.

4. **Conclusions.** In this paper, we investigated the problem of fingerprinting multimedia content that can resist collusion attacks and trace colluders. The traditional detection schemes for Gaussian fingerprinting require an amount of correlations that is linear in the amount of fingerprints. In order to solve this problem, the binary tree detection scheme

was proposed which decreased the correlations greatly. However, this scheme employed a recursive manner and needed to transfer a large number of data, which leads to detection time being increased.

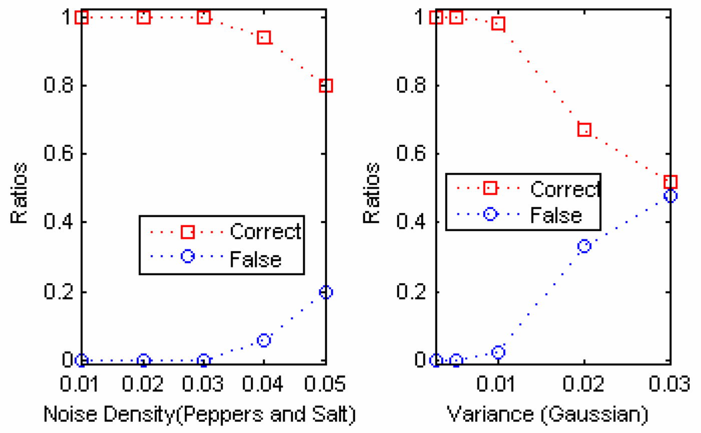

Figure 9. Robustness vs. JPEG compressed attacks

We proposed an accelerated tracing scheme whose essence was that compacting the fingerprints to construct its index. The shorten index was created through wavelet decomposition. And then it was employed to participate in comparison. We studied the relationships among the decomposition layers, robustness and time consumed. Then we studied the relationships between the candidates number and robustness. The compared simulations for the traditional scanning, the binary tree detection and ours were listed. The results proved the stated advantages of our scheme. Finally, the whole simulation results on real images also verified our scheme's effectiveness. The proposed scheme can be extended easily to other fingerprints. However, there are some problems should be studied in future. The first one is that weather there are differences if employ different wavelets decomposition, if not, which one is the best? The second one is that the theoretical model for features resisting collusion attacks should be studied. The third one is that exploring other feature extraction methods.

Acknowledgment. The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. This work is supported by Doctoral Funds of Henan Polytechnic University under Grant No. B2013-022, National Natural Science Foundation of China under Grant Nos. 61379143, 60802077,61105066 and the Opening Project of Shanghai Key Laboratory of Integrate Administration Technologies for Information Security under Grant No. AGK2012002.

REFERENCES

[1] I. J. Cox, J. Kilian, and T. Leighton et al, Secure spread spectrum watermarking for multimedia, *IEEE Trans. Image Process*, vol. 6, no. 12, pp. 1673-1687, 1997.

[2] W. Trappe, M. Wu, and Z. J. Wang et al, Anti-collusion fingerprinting for multimedia, *IEEE Trans. Signal Process*, vol. 51, no. 4, pp. 1069-1087, 2003.

[3] M. Wu, W. Trappe, Z. and Jane Wang et al, Collusion-resistant fingerprinting for multimedia, *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 15-27, 2004.

[4] Z. Jane Wang, M. Wu, and H. V. Zhao et al, Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation, textitIEEE Trans. Image Processing, vol. 14, no. 6, pp. 804-821, 2005.

[5] H. V. Zhao, M. Wu, Z. J. Wang, and et al, Forensic analysis of nonlinear collusion attacks for multimedia fingerprinting, *IEEE Trans. Image Processing*, vol. 14, no. 5, pp. 646-661, 2005.

[6] A. L. Varna, S. He, and M. Wu, Fingerprinting compressed multimedia signals, *IEEE Trans. Information Forensics and Security*, vol. 4, no. 3, pp. 330-345, 2009.

[7] M. Kuribayashi, and H. Kato, Impact of rounding error on spread spectrum fingerprinting scheme, *IEEE Trans. Information Forensics and Security*, vol. 5, no. 4, pp. 670-680, 2010.

[8] X. W. Li, B. L. Guo, L. D. Li, and et al, A novel fingerprinting algorithm with blind detection in DCT domain for images, *International Journal of Electronics and Communications*, vol. 65, no.11, pp. 942-948, 2011.

[9] X. W. Li, B. L. Guo, X. X. Wu and L. D. Li, On collusion attack for digital fingerprinting, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, no. 4, pp. 366-376, 2011.

[10] Y. N. Li, Z. M. Lu, and C. Zhu, and et al, Robust image hashing based on random gabor filtering and dithered lattice vector quantization, *IEEE Trans. Image Processing*, vol. 21, no. 4, pp. 1963-1980, 2009.

[11] Y. Zhao, S. Z. Wang, and X. P. Zhang, and et al, Robust hashing for image authentication using zernike Moments and Local Features, *IEEE Trans. Information Forensics and Security*, vol. 8, no. 1, pp. 55-63, 2013.

[12] M. Li, and V. Monga, Robust video hashing via multilinear subspace projections, *IEEE Trans. Image Processing*, vol. 21, no. 10, pp. 4397-4409, 2012.

[13] B. Kulis and K. Grauman, Kernelized locality-sensitive hashing for scalable image search, *Proc. IEEE 12th International Conference on Computer Vision*, pp. 2130-2137, 2009.

[14] A. Gionis, P. Indyk, and R. Motwani, Similarity search in high dimensions via hashing, *Proc. of the 25th International Conference on Very Large Data Bases*, pp. 518-529, 1999.

[15] G. Shakhnarovich, P. Viola, and T. Darrell, Fast pose estimation with parameter-sensitive hashing, *Proc. 9th IEEE International Conference on Computer Vision*, pp. 750 -757, 2003.

[16] M. M. Esmaeili, R. K. Ward, and M. Fatourechi, A fast approximate nearest neighbor search algorithm in the hamming space, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2481-2488, 2012.

[17] Y. Y. Wang, Z. M. Li, L. Wang, and et al, A scale invariant feature transform based method, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 2, pp. 73-89, 2013.