http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (2): 251-259, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.251.259

A Client-side Fingerprinting Method for Text Document Distribution

[1]Hui Peng, [1]Gang Luo and [2]Lingyun Xiang
[1]School of Information Science and Engineering, Hunan University, Changsha, 410082, China
[2]School of Computer and Communication Engineering,
Changsha University of Science and Technology, Changsha, 410114, China

Abstract: In this study, a new method that joint fingerprinting and decryption at the client side is proposed for the electronic distribution of text documents. In mass-scale electronic distribution systems fingerprint embedding primarily at the server which may lead to computational load of server and bandwidth burden. Fingerprinting at the client side will reduce the embedding complexity and bandwidth usage but it is easy to pose a threat to security on the fingerprint information and the protected content. It will be a challenge to design a client-side embedding method which will not leak the embedding secrets or original content. Some client-side embedding methods have been proposed for images, video and other media which contain enough redundancy information and allow various processions. But for text documents, to overcome this challenge will be more difficult. In this study, by giving each user a personalized coding dictionary which constructed based on synonym substitution, only one encoded and encrypted copy needs to be sent by the server and fingerprint embedding is jointed with decryption at the client side. Theoretical analysis and experimental results prove its practice.

Key words: Copyright protection, joint fingerprinting and decryption, text document, synonym-substitution, steganography

INTRODUCTION

In recent years, digital products gradually replace classic content distribution channels, plays a very important role in human life. How to protect copyright becomes one of the most important questions of electronic distribution. Digital Rights Management (DRM) (Jonker and Linnartz, 2004) systems try to reduce the risk of copyright infringements by using cryptographic techniques to securely sent content to client devices. Forensic tracking watermarks also introduced for electronic distribution which can be used alone or in conjunction with traditional encryption techniques.

In forensic tracking system, each copy of the distributed content was watermarked with a unique label called fingerprint (Wagner, 1983) which made that copy correspond with a particular user or a specific device. Once an unauthorized copy is found, the extracted fingerprint uniquely identifies the source of the copy and also determines the re-distributed user.

In traditional scheme, fingerprint is embedded at the server. With the need of individually marked content copies, the server computing load increases linearly with the number of users (Van der Veen *et al*., 2005). The uniqueness of each copy also prohibits the use of broadcasting, multicasting and caching. Therefore, the scheme is hard to apply in large-scale distribution systems.

Some researchers have suggested that the embedding can be performed by intermediate nodes of a network, such as routers. For instance, Crowcroft *et al*. (2000) propose using network nodes that switch between different watermarked versions of information segments to watermark an information signal in time. Their method significantly reduces the bandwidth requirements of the system but it also brings a series of new problems, such as vulnerability to intermediate node compromise, do not practical in today's heterogeneous networks and packet dropping.

In order to solve the previous problems, many researches study on client-side embedding. In such a scheme, the complexity is shifted from the server to the client device; a unique watermark could be embedded by each client device in decrypted content. Since all users will receive the same unmarked copy of the encrypted content which allowing the use of broadcast, multicast and caching mechanism. The main bandwidth bottleneck has been eliminated. The scalability of which makes it especially suitable for large-scale electronic content distribution systems.

Corresponding Author: Gang Luo, School of Information Science and Engineering, Hunan University, No. 252, Lushan South Road, Yuelu District, Changsha, 410082, China

One security issue faced by client-side embedding method designers is that original content and the embedding secrets have the risk of leak. A solution to this problem is provided by tamper-resistant hardware. However, although the tamper-resistant hardware solution is feasible, it is usually costly and inflexible.

Another way to solve the security problem is to design suitable algorithms. In this scenario, the content will transmit in encrypted form and each user will get a client-specific decryption key. Each user decrypts the content by using his or her personalized decryption key. As a result, the decryption process effectively superimposes a fingerprint sequence onto the decrypted content. Through rational design, security of client-side embedding will be comparable to the inherent security of the watermark.

Many client-side embedding algorithms have been proposed but there are still some drawbacks. For instance, Anderson and Manifavas (1997) proposed a special stream encryption method called Chameleon which is encrypt with the use of lookup table. By modifying less-significant bits of some words in the lookup table, the decryption result will be slightly different from the original plain text and those differences generated by each user's customized lookup table can be used as a forensic tracking watermark in the absence of additional noise.

Celik *et al.* (2008) contemporarily proposed a spread-spectrum watermarking method that shares the same general principle of Chameleon. The main difference is that the format of the encrypted content changed from bit streams to Discrete Cosine Transform (DCT) coefficients. Kundur and Karthik (2004) presented a joint fingerprinting and decryption scheme for images based on coefficient scrambling. In their method, images is encrypted by modifying signs of significant (mid-frequency) DCT coefficients and gives each user the necessary keys to decrypt only a subset of these coefficients. The coefficients that remain scrambled actually form a forensic mark. The main problem of this method is that the decryption key may leaked the locations of part scrambling coefficients which will lead to security flaw, especially when collusion occurs.

Lemma *et al.* (2006) overcome these problems and present a client-side embedding method which is based on partial encryption for both advanced audio coding (AAC)-encoded audio and MPEG2 encoded video streams. Their method requires profuse additional information which will increases the band load.

Recently, Lin *et al.* (2010) proposed a JFD method for vector quantization images. Their method is relatively simple but needs to be worked with specific hardware. Piva *et al.* (2010) creatively designed a secure client-side embedding system which could be used to embed spread transform dither modulation (ST-DM) watermark. Pun *et al.* (2011) proposed a client-side LUT-based watermarking method for audio which achieves better robustness and inaudibility because of the adaptive selection of embedding strength for each individual segment.

It can be seen that client-side embedding algorithms mentioned previously are proposed for images, video and other media of which have abundant redundancy and allowed various processing means. However, thoughts of these embedding schemes cannot be applied to the other media directly. There is no similar method putted for text content of which still forms the bulk of Internet traffic (Topkara *et al.*, 2004). In view of this, inspired by similar works mentioned before, this study presents a client-side embedding method that joint fingerprinting and decryption for electronic distribution of text documents. The proposed method avoids some problems that existed in the mentioned articles-it requires neither very long keys nor profuse help information and the fingerprint has excellent imperceptibility while the entire system is secure enough.

PROPOSED APPROACH

In this study, the fingerprint embedding is performed based on synonym-substitution. Before describe the approach, we have briefly to introduce the mechanism of synonym substitution-based linguistic steganography.

Synonym substitution: Synonym substitution is the most widely used linguistic transformation in text steganography since it does not need sentence parsing (Chiang *et al.*, 2003). In synonym substitution system, the hidden message is embedded by substituting changeable words so that the truth values of the modified sentences are preserved (Topkara *et al.*, 2004).

For synonym substitution, a synonym lexicon which was grouped by disjoint sets is used (Yu *et al.*, 2008). In such a synonym lexicon only the words completely express the same concept are grouped in the same synonym set. The grouping of the lexicon is largely determines the performance of the substitution algorithm. All words in a synonym set are coded and the correct word will be selected according to designed rules in order to embed messages.

Distribution scheme: Inspired by previous researches and considering with the characteristics of text content, a client-side embedding scheme has been designed for text content distribution. In the distribution scheme, each user is given a personalized coding dictionary, in which each word is encoded in order to achieve the conversion between words and its codeword. Every word of the original text will be converted to the

corresponding codeword firstly by looking up the coding dictionary of the server and then the entire encoded text will be encrypted by a session key which is chosen for the delivered content each time. Each user coding dictionary that constructed by the system is slightly different with others and such a difference is generated based on the notion of synonym substitution. With a process that joint fingerprinting and decrypt, those differences are used as the user's fingerprint information and be embedded in the decrypted text (stego-text). The stego-text keeps the same sense before and after transmission just like in other synonym substitution algorithms. Figure 1 illustrates the main procedures of the distribution scheme. Each step shown in the figure will be discussed in detail next.

Coding dictionary construction: For coding dictionary construction, it need to collect as many words as possible to construct a dictionary at first, especially that all of the commonly used words must be included. Then the collected dictionary will be encoded, so that each word in it is represented by a unique codeword with binary form which allow the conversion between word and its codeword. Details of the encoding process will not be described in this study but it is obvious that it exerts a critical influence on the efficiency and security of the distribution system.

After encoding, a server-side coding dictionary wherein each word w_i has a unique corresponding codeword C_i is constructed and let D_0 denote such a dictionary. According to the coding dictionary D_0, the client-side dictionaries can be constructed for each user by using synonym substitution. A synonym lexicon consists of synonym sets that can be used for substitution-based linguistic steganography is required in the proposed scheme which is denoted by L. Let $S_i = \{S_{i,0}, S_{i,1}, ..., S_{i,m_i-1}\}$ denote the synonym set in L, where $S_{i,j}$ $(0 \le j < m_i)$ represents the element of S_i and m_i represents the size of that synonym set. The main idea of the algorithm is performing some permutation operations on the original server-side coding dictionary.

Algorithm 1: User coding dictionary construction

Input: Original coding dictionary D_0, synonym lexicon L which with n synonym sets, pseudo-random number generator

Output: User coding dictionary D_u

Steps:

Step 1: Let $D_u = D_0$

Step 2: for I = 0 = 0 to n-1 do

Step 3: Use D_0 to find out the corresponding codeword of each element in $S_i = (S_{i,0}, S_{i,1}, ..., S_{i,m_i-1})$ $(0 \le i < n)$ and denote them by $(C_{i,0}, C_{i,1}, ..., C_{i,m_i-1})$

Step 4: Use the pseudo-random number generator to generate a number p_i with the contains $0 \le p_i < m_i$

Step 5: According to p_i do a permutation for S_i:

$$S'_{i,j} = S_{i,(j+p_i)\%m_i}$$

where, $S'_i = \{S'_{i,0}, S'_{i,1}, ..., S'_{i,m_i-1}\}$ is used to denote the new permutated synonym set and % is the complementary operation

Step 6: Do corresponding adjustments in D_u according to the permutation results. That is to say, in D_u, $C_{i,j}$ is assigned as the codeword of word S'_i in the permutated synonym set instead of the original one

Step 7: end for

Step 8: return D_u

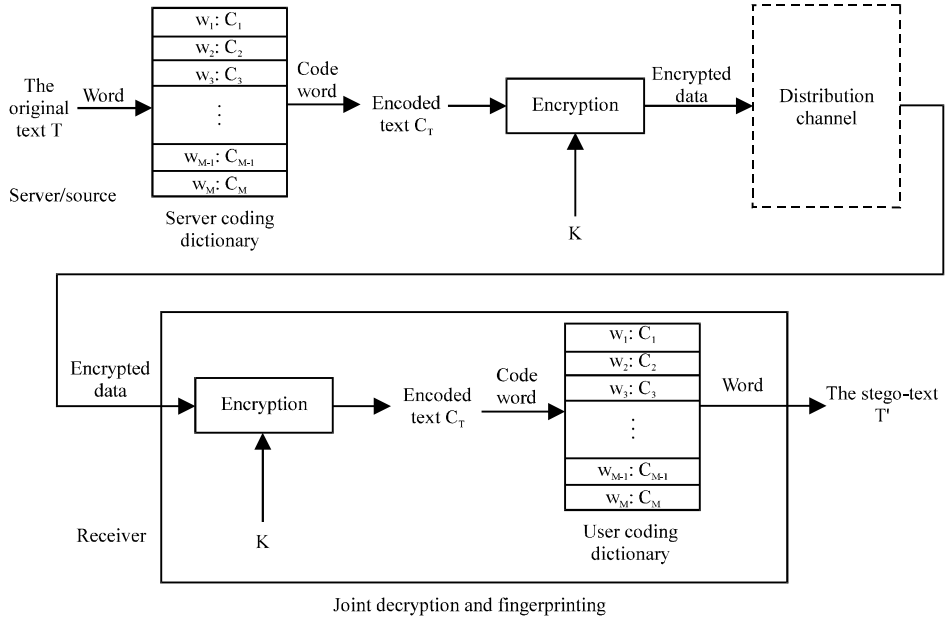

Fig. 1: Client-side fingerprinting scheme for the electronic distribution of text documents

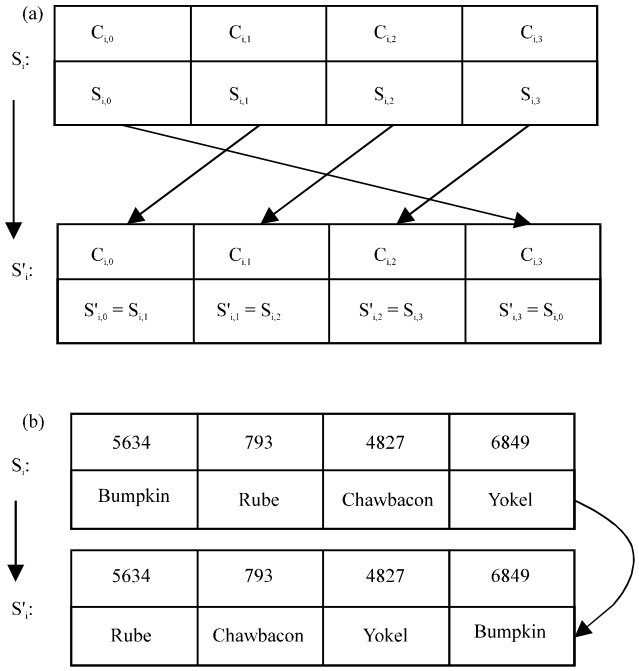

Fig. 2(a-b): An example of permutation in user coding dictionary construction, (a) Permutation in a synonym set and (b) An example of permutation results

Figure 2 shows an example to illustrate the permutation in Algorithm 1. There is a synonym set S_i = {Bumpkin, Rube, Chawbacon, Yokel}, decimal numbers is used to represent the corresponding codewords for easy to understand. The size of that synonym set is 4 and the generated pseudo-random number is assumed to be 1. Then according to step 5 in the algorithm, the result of permutation is as shown in Fig. 2b. It should be noticed that elements in a synonym set are not adjacent in the coding dictionary but dispersed.

Because the permutation for each synonym set is depended on a pseudo-random number, the result of each run in Algorithm 1 will be different. So, if there are n users in the distribution system, it just need to run the algorithm n times to construct the desired coding dictionaries and each dictionary will be sent to the corresponding user after encryption. In application, the user coding dictionary could be encapsulated (be invisible to the user) if necessary.

Text encryption and distribution: Let denote the original text data by $T = \{w_0, w_1,\ldots,w_{m-1}\}$, where m is the number of words contained in T. It should be noted that the concept of "word" also includes punctuation here. According to the server-side coding dictionary D_0, w_i is converted to its codeword C_i to generating the corresponding encoded text C_T with the form $\{C_0, C_1,\ldots, C_{m-1}\}$. If $w_i \notin D_0$, it should be given a particular temporary codeword which must be different with others. These temporary coding are recorded and added at the end of encoded text C_T as help data (additional information) for decryption. Since enough words, especially common ones are collected in the dictionary, words rarely need to be encoded temporary. Our experiments will confirm this inference.

The encoded text C_T will be compressed and then decrypted by AES with a short-term session key K which can be sent to each user in asymmetric channel at each time. The encrypted data is broadcast to authorized users in public channel.

Decryption and fingerprinting: Authorized user decrypt the received data by the same session key K and then obtained the encoded text C_T by decompression. In accordance with the customized coding dictionary, each codeword in the encoded text is converted to its corresponding word to generating a final stego-text.

Let define T^j as the corresponding stego-text which decrypted by user j, therefore, the word w_i^j which converted from codeword C_i is the corresponding word of w_i, where $w \notin T$ and $w^j \notin T^j$. It can be known from procedures of dictionary construction and encryption that $w_i^j = w_i$ if $w_i \notin L$, otherwise, w_i^j must be a synonym of w_i due to the permutation as shown in Fig. 2. Therefore, the meaning of the text remains unchanged and the stego-text can be regarded as the result of performing synonym substitution in the original text. Such substitutions will be

different in the stego-texts due to the different customized user coding dictionary and it could be used for user authentication.

To better illustrate, some definitions will be introduced in this article appropriately.

Definition 1

Keyword sequence: The keyword sequence of text T is the sequence that contains all synonym words (keywords) in the text, it denoted by seq(T) = $\{w_0, w_1, \ldots, w_{n-1}\}$, where, $w_i \in L$ and n is the number of synonym words in text T.

Thus, the keyword sequence of stego-text T_j correspond to the original text T must be $seq(T^j) = \{w_0^j, w_1^j, \ldots, w_{n-1}^j\}$ and which is uniquely determined by the coding dictionary of user j. For the sake of convenience, seq(T^j) is also called the keyword sequence of user j. We focus on the difference between keyword sequences of any two users. Let $seq(T^a) = \{w_0^a, w_1^a, \ldots, w_{n-1}^a\}$ and $seq(T^b) = \{w_0^b, w_1^b, \ldots, w_{n-1}^b\}$ be keyword sequences of user a and user b, respectively. Then $w_i^a = w_i^b$ only if same permutation is applied in the both dictionaries, so the probability of $w_i^a = w_i^b$ is given by:

$$P_r(w_i^a = w_i^b) = P_r(p_i^a = p_i^b) = 1/m_i \quad (1)$$

where, m_i is the size of synonym set which w_i belong to, p_i^a and p_i^b are the pseudo-random numbers generated by step 4 of Algorithm 1 which used to construct user coding dictionary D_a and D_b, respectively.

In other words, differences among user coding dictionaries which introduced by permutations are eventually reflected by the keyword sequences of the stego-texts. As knowing from Eq. 1, such a difference is determined by the length of sequence and a series of pseudo-random numbers used for coding dictionary construction. Therefore, if the keyword sequence is long enough and user scale is controlled within a reasonable range, each user can be distinguished by the keyword sequences of their stego-texts with a larger probability. So this sequence can be used as the user's fingerprint information (fingerprint sequence).

Fingerprint extraction and identification: Through the foregoing introduction, the identification process looks very simple and natural. For any original text, each user may get a specific fingerprint sequence which is determined by the customized user coding dictionary and be embedded in the final stego-text. Then the fingerprint identification can be achieved by extracting keyword sequence from the suspicious text and compared it with the fingerprint sequences of each user. Let us introduce the concept of "sequence similarity" while a threshold θ is also selected for identification, users who have similarity between his or her fingerprint sequence and the keyword sequence of the unauthorized text that beyond the threshold value will be judged as a traitor. The definition of sequence similarity is introduced before given the traitor tracing algorithm.

Definition 2

Sequence similarity: It measures the similarity of two sequences with equal words. For two given word sequences $S^a = \{w_0^a, w_1^a, \ldots, w_{n-1}^a\}$ and $S^b = \{w_0^b, w_1^b, \ldots, w_{n-1}^b\}$ which are contains n words, if $w_i^a = w_i^b$, we say that these two sequences is match at location i, the total number of matches is referred to as match $\{S^a, S^b\}$, the sequence similarity will be denoted by sim $\{S^a, S^b\}$ and calculated by the following expression:

$$sim\{S^a, S^b\} = \frac{match\{S^a, S^b\}}{n} \quad (2)$$

Once the text owner detected a suspicious text T'the traitor will be revealed by the following algorithm:

Algorithm 2: Traitor tracing

Input: Suspicious text T', original text T, Synonym lexicon L, Server-side coding dictionary D_0, User coding dictionaries, threshold θ

Output: The traitor set S_T

Steps:

Step 1: According to Definition 1 to obtain the keyword sequence of T' and save it as seq(T') = $\{w'_0, w'_1, \ldots, w'_{n-1}\}$

Step 2: Find out the corresponding original text T and then obtained the keyword sequence sqe(T) = $\{w_0, w_1, \ldots, w_{n-1}\}$ too

Step 3: Find the codeword sequence of seq(T) through the use of D_0 and save it as $C_S = \{C_0, C_1, \ldots, C_{n-1}\}$

Step 4: For each user j of the distribution system do

Step 5: Convert each codeword in C_S to its corresponding word by the use of user coding dictionary D_j to obtain the keyword sequence which denoted by $seq(T^j) = (w_0^j, w_1^j, \ldots, w_{n-1}^j)$, according to our distribution scheme, it must be the fingerprint sequence of user j

Step 6: Calculate the sequence similarity sim{seq(T'), seq(T^j)} between seq(T') and seq(T') according to Definition 2

Step 7: If (sim{seq(T'), seq(T^j)}>θ) then add user j to the traitor set S_T

Step 8: End for

Step 9: Return S_T

ANALYSIS AND EXPERIMENT

This method is proposed for Chinese text. At the same time, compared to the classics, the proposed scheme is particularly suitable for booming literature because of their relatively low accuracy requirements and in aspects of copyright protection needs. Therefore, text documents selected in our experiments are internet novel in Chinese.

An additional process called Chinese Word Segmentation is needed in the distribution scheme due to the lack of space which may separate each word. In this study, the Chinese lexical analysis system ICTCLAS that

developed by Chinese Academy of Sciences is selected as the segmentation tool. Experimental text documents have to be separated first and then be distributed according procedures shown in Fig. 1. The proposed method will be analyzed combined with the experimental results.

Confidentiality and imperceptibility: The use of AES provides a good confidentiality for the distribution scheme. With a leaked session key, the attacker can only get the encoded text which may not disclose the original text and fingerprint information due to the confidential and independent coding dictionary construction.

Therefore, the AES key K is a short-term key which may renewed each time the text have to encrypted and the user coding dictionary can be viewed as a long-term key, both of them are used in decryption. The system need to completed coding dictionary construction at the initial stage and update when necessary, instead of to sent with the encrypted content every time. Thus, the proposed client-side fingerprinting scheme greatly reduce the embedding complexity of server and the traffic load, especially for distribution of large text documents such as e-books.

The proposed approach is based on synonym substitution which have widely used in text steganography and shows good imperceptibility. Since synonyms appeared fewer and scattered in text, the user's fingerprint is almost imperceptible.

Traitor tracing: In non-attack background, if similarity between the keyword sequence of the unauthorized copy and the fingerprint sequence of an innocent user exceeds the predetermined threshold, this innocent user will be judged as a traitor according to the traitor tracing algorithm. Since the user fingerprint is determined by pseudo-random numbers generated during dictionary construction, such a positive falsity may existed on probability. We will focus on it because of its great impact on the performance of the distribution scheme.

For a given original text T, any two fingerprint sequence $seq(T^a) = \{w_0^a, w_1^a, ..., w_{n-1}^a\}$ and $seq(T^b) = \{w_0^b, w_1^b, ..., w_{n-1}^b\}$ of two stego-texts will be matched at location i with the probability of $1/m_i$, which can be known from Eq. 1. The total match locations of these two sequences will nearly fit the binomial distribution if we assumed that all synonym sets in the synonym lexicon have the same size. In the experiments, the average synset size of the synonym lexicon is 2.38 while the maximum is 7 and the minimum is 2. This average value ($m_a = 2.38$) is used to instead the real size of each synonym set in order to simplify the analysis.

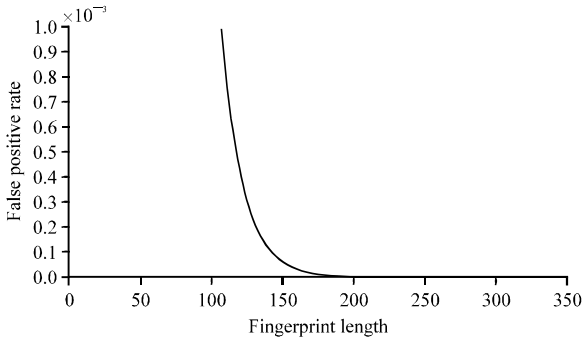

Fig. 3: Estimated relation between error probability and fingerprint length

Let us denote the mentioned false positive rate as P_f. With the above simplification, according to theorem A.1.4 which gives the estimation for binomial distributions (Alon and Spencer, 2000) and works of Chen and Zhu (2007), P_f can be estimated by:

$$P_f = P_f(sim\{seq(T^a)seq(T^b)\} > \theta < exp[-2(\theta n - n/m_i)^2/n \quad (3)$$

It reveals the relationship between the false positive rate and the length of the fingerprint sequence. Figure 3 shows the result when related parameters are substituted while the threshold value is set to 0.6. The false positive rate can be controlled within an acceptable range if fingerprint sequences are long enough for distinguished.

Certainly, the above discussion is only involved the case of two random users without considering the user scale of the system. In fact, for the entire system, the false positive rate will increase proportionally to the square of the user scale. If lower error rate or more large scale is required, the user whose fingerprint sequence is most similar with the suspicious text should be judged as the traitor instead of users with the similarity beyond a threshold.

There are 250 text documents to be used in our experiments. Stego-texts decrypted by different users are choosed randomly for authentication, Fig. 4 shows the results. Considering of the performance of the segmentation system and time overload, size of the tested documents is limited to less than 100 kb and only 10 users are selected for identification (It can be overcome through divide large document into several parts for extracting the fingerprint sequence in application).

Almost all traitors have been traced correctly when threshold value is set to 0.6. The result is consistent with the analysis and proves the feasibility of the proposed fingerprinting scheme.

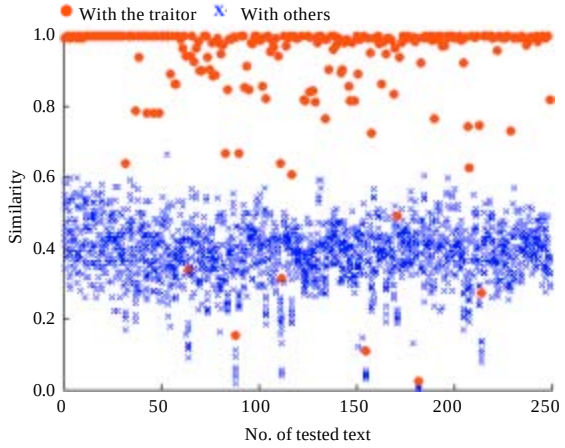

Fig. 4: Results of traitor tracing without attacks

In our method, adjusting the format of text contents has no effect on the extraction of keyword sequence, so does not affect identification. On the other hand, malicious attacks like partial adding, deleting have a huge impact on the fingerprint extraction, obviously. If any attack cause great changes in the text keyword sequence, similarity calculated using the proposed method will be decreased significantly. A more accurate extraction is needed for traitor tracing and a threshold, 0.5 for example was used in this case. We consider that the suspicious text was malicious attacked or the segmentation system did not achieve the desired effect if similarities with all users do not exceed the threshold. Then the keyword sequence must extracted sentence by sentence in order to remove sentences which can not match with the original text semantically. Time overhead must be increased in this case due to the introduced syntax analysis procedure.

Attacks of partial adding, deleting and modifying were executed on 50 stego-texts. Figure 5 shows the result, where (a) were original similarities calculated by Algorithm 2 and (b) were results calculated after the non-matching sentences have been removed. It can be seen that our proposed method achieves relative robustness to resist such attacks.

Attacks mentioned before usually carried on without embedding secrets. In this case, it is hard for attacker to remove the fingerprint without reduce quality of the carrier. Fingerprinting algorithm faced with another type of attack called collusion which refers to a coordinated attack by a group of users, called the coalition. Members of the coalition use their personalized content copies (bearing different fingerprints) to obtain an attacked version in which none of their fingerprints can be reliably detected.

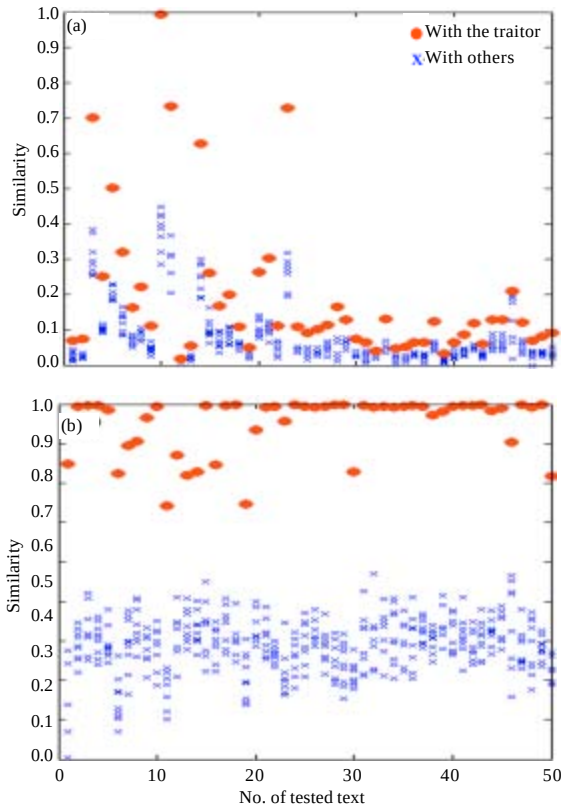

Fig. 5(a-b): Results of traitor tracing with malicious attacks, (a) Results calculated by the original traitor tracing algorithm and (b) Results calculated after the non-matching sentences have been removed

If fingerprints of all members of the coalition agree in the i-th bit, then it is considered that the i-th bit is undetectable for collusion. The marking assumption proposed by Boneh and Shaw (1998) states that the undetectable marks cannot arbitrarily be changed without rendering the content copy useless. There are two ways to obtain a content copy through collusion for the distribution scheme: One is by the personalized copies itself and the other is by coding dictionaries of each colluder, both of them will achieve the same effect. Now we have to discuss the collusion-resistance of the proposed method under the marking assumption.

If there are c members in the coalition, the probability of any location in the keyword sequences is undetectable will be $1/m_i^{c-1}$, where m_i is the size of the synonym set which the keyword in that location belong to. Then there will be n/m_i^{c-1} undetectable locations for a fingerprint sequence with the length of n. Such undetectable

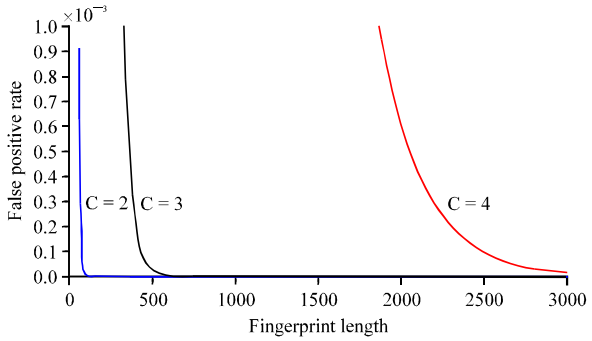

Fig. 6: Estimated relation between error probability and fingerprint length under different conditions

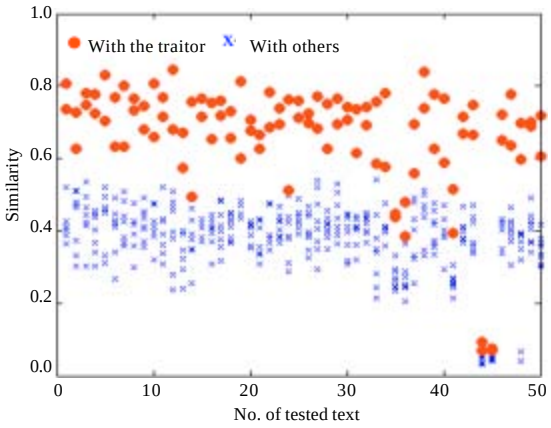

Fig. 7: Traitor tracing for collusion with two users

keywords will remain unchanged while keywords in other locations must be corresponding selected from all copies randomly to generate the redistributed copy (redistributed by colluders).

With the same assumption, similar to works of Chen and Zhu (2007), we got an estimation of false positive rate in the case of collusion (if the user who has the most similar fingerprint be judged as the traitor):

$$p_f < \exp[-2(m_i-1)^2 n/m_i^{2c}] \quad (4)$$

Although there is a slightly approximation error in this estimation, it clearly reflects the relationship between the false positive rate, the length of keyword sequence and collusion number yet. Figure 6 shows such a relationship with different collusion numbers.

It can be seen that our method has well collusion-resistance when there are only two colluders but the resistance is reduced significantly when coalition consists more than 3 members. Coalition consists of two members have been experimented and Fig. 7 shows the result. All calculations indicate the correct tracing. Experimental results of some tested documents are not as good as expected because of the segmentation system do not achieve the desired effect. For better experimental results, more accurate keyword sequence extraction need to be applied as when other attacks occurred.

CONCLUSION

In this study, a client-side embedding method was proposed for the electronic distribution of text documents. In the proposed method, only one encrypted copy needs to be sent by the server-side and fingerprint embedding is joint with decryption at the client side by the use of personalized user coding dictionary. It can greatly reduce the embedding complexity of server and the traffic load which is of great significance in today's electronic reading environments. Theoretical analysis and experiments show that the proposed method has good performance in confidentiality and imperceptibility but it also has some drawbacks, such as robustness against some attacks, especially the collusion attack, is not as good as expected. At the same time, the fingerprint extraction and identification will be impacted by the segmentation system which could take some extra time overhead. In the near future we will dedicate on fingerprint capacity and fingerprint code design which could depended on to solve the existed problem and to make the scheme more suitable in application. At the same time, a similar approach which suitable for media like images or video is also a direction of our future work.

ACKNOWLEDGMENT

This study is supported by National Natural Science Foundation of China (Grant No. 61070196, 61103215 and 61202439).

REFERENCES

Alon, N. and J.H. Spencer, 2000. The Probabilistic Method. John wiley and Sons, Inc., New York, USA., pp: 263-265.

Anderson, R. and C. Manifavas, 1997. Chamleon: A new kind of stream cipher. Proceedings of the 4th International Workshop on Fast Software Encryption, January 20-22, 1997, Haifa, Israel, pp: 107-113.

Boneh, D. and J. Shaw, 1998. Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory, 44: 1897-1905.

Celik, M.U., A.N. Lemma, S. Katzenbeisser and M. van der Veen, 2008. Lookup-table-based secure client-side embedding for spread-spectrum watermarks. IEEE Trans. Inform. Forensics Security, 3: 475-487.

Chen, X.S. and D.L. Zhu, 2007. A digital fingerprint coding and tracing algorithm based on random binary codes. J. Chinese Comput. Syst., 28: 823-825.

Chiang,Y.L., L.P. Chang, W.T. Hsieh and W.C. Chen, 2003. Natural language watermarking using semantic substitution for Chinese text. Proceedings of the 2nd International Workshop on Digital Watermarking, October 20-22, 2003, Springer, pp: 129-140.

Crowcroft, J., C. Perkins and I. Brown, 2000. A method and apparatus for generating multiple watermarked copies of an information signal. WO Patent/056059, http://patentscope.wipo.int/search/en/WO2000056059

Jonker, W. and J.P. Linnartz, 2004. Digital rights management in consumer electronics products. IEEE Trans. Signal Process., 21: 82-91.

Kundur, D. and K. Karthik, 2004. Video fingerprinting and encryption principles for digital rights management. Proceedings of the IEEE, Volume 92, May 18, 2004, IEEE., pp: 918-932.

Lemma, A., S. Katzenbeisser, M. Celik and M. van der Veen, 2006. Secure watermark embedding through partial encryption. Proceedings of the International Workshop on Digital Watermarking, November 8-10, 2006, Jeju, Island, Korea, pp: 433-445.

Lin, C.Y, W.L. Huang and T.H. Chen, 2010. Noise-resistant joint fingerprinting and decryption based on vector quantization. Proceedings of the International Conference on Broadband, Wireless Computing, Communication and Applications, November 4-6, 2010, Fukuoka, Japan, pp: 463-468.

Piva, A., T. Bianchi and A. De Rosa, 2010. Secure client-side ST-DM watermark embedding. IEEE Trans. Inform. Forensics Sec., 5: 13-26.

Pun, C.M., J.J. Jiang and C.L.P. Chen, 2011. Adaptive client-side LUT-based digital watermarking. Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), November 16-18, 2011, Changsha, China, pp: 795-799.

Topkara, M., C. Taskiran and E.J. Delp, 2004. Natural language watermarking. Proc. SPIE, 5681: 441-452.

Van der Veen, M., A. Lemma and T. Kalker, 2005. Electronic content delivery and forensic watermarking. Multimedia Syst., 11: 174-184.

Wagner, N.R., 1983. Fingerprinting. Proceedings of the Symposium on Security and Privacy, April 25-27, 1983, Oakland, California, pp: 18-22.

Yu, Z., L. Huang, Z. Chen, L. Li, X. Zhao and Y. Zhu, 2008. Detection of synonym-substitution modified articles using context information. Proceedings of the 2nd International Conference on Future Generation Communication and Networking, Dec. 13-15, IEEE Computer Society, Hainan, China, pp: 134-139.