21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

Dynamic Traitor Tracing with Near Optimal Codes

Xin-Wen Wu*

Abstract— **For static traitor tracing, the codes proposed by Tardos are the best known codes with regard to code length. Tardos codes are asymptotic optimal according to the low bound on code length proved by Peikert et al. The computational complexity of the tracing algorithm for Tardos codes is $O(N)$, where N is the number of authorized users. Recently, Tardos codes have been adapted to dynamic traitor tracing. The adapted codes have the same tracing complexity as Tardos codes, that is, $O(N)$.**

In this extended abstract, we report our research of attempting to develop dynamic traitor-tracing schemes, using the near-optimal codes obtained from the concatenation of Tardos codes and algebraic-geometric codes, which allow an efficient tracing procedure with complexity $O(\log(N))$.

Keywords: Dynamic traitor tracing, Tardos codes, algebraic-geometric codes, optimal codes.

AMS subject classifications: 11T71, 14G50.

I. INTRODUCTION

We consider the problem of protecting digital objects against illegal re-distribution. Digital fingerprinting provides a means of tracing unauthorized re-distribution [1], [2]. Before distribution the owner (or the authorized distributer) imperceptibly embeds an unique fingerprint, that is, a sequence of marks, into each authorized copy of the digital object, using a digital (invisible) watermarking procedure. In case an unauthorized copy of the object is found, by extracting and analysing the fingerprint from the observed pirate copy, the user who made the unauthorized copy (such a user is called a *traitor* or a *pirate*) can be identified.

Fingerprinting schemes address the problem of *collusion*, where a group of users (called a *coalition*) try to detect and then change the fingerprint marks in their copies, and to produce an "untraceable" version of the digital object. In order to discover the hidden fingerprint marks, the members of a coalition can simply compare the copies that belong to them, to find the locations where their copies differ. It is natural to assume that users cannot detect or alter the fingerprint marks in the positions where their copies agree. This is referred as the *Marking Assumption* in the literature [1]. There are two different models for how the pirate copy could be produced by the traitors: (1) The *mix-and-match model* allows the traitors only to replace a segment of the object which contains a mark of the fingerprint, by a corresponding segment of another copy that belongs to them; (2) The *general-digit model* allows the traitors to put any symbol, including unrecognizable symbol, to replace a fingerprint mark that they can detect, or simply remove the detectable marks. Obviously, the general-digit model includes the first model as a special case.

An effective fingerprinting scheme consists of two components: a fingerprinting code and a tracing algorithm, where the *fingerprinting code* $\mathcal{C}$ is a set of codewords, with each codeword being assigned to a unique user as the fingerprint in his/her copy of the object. A *tracing algorithm* T is a procedure that tries to identify the traitor(s) using the observed pirate copy. Two types of traitor tracing schemes haven been studied in the literature, namely, static schemes and dynamic schemes [5]. In *static schemes*, the process ends after one run of the above algorithm with a fixed code length, and the output of T (which is a subset of the set of users) is the final set of accused users. So the complete codewords are generated and distributed, the pirates generate and distribute a pirate copy, and the distributor detects the pirate copy and calculates the set of accused users. Static schemes are applied to Blu-ray disc content protection, for example, in the recently developed standard, Advanced Access Content System (AACS), and to DVD encryption in the Content Scramble System (CSS).

In *dynamic schemes*, the process of sending out symbols (i.e., fingerprint marks), detecting pirate copy, and running a tracing algorithm is executed in multiple stages during the distributing or broadcasting of the digital content. If a traitor is caught in a certain stage, he/she is immediately cutoff from the system and can no longer access the content. The dynamic schemes are applied to live broadcasts, such as pay-TV. In the dynamic scenarios, the distributor broadcasts the content, while the pirates directly output a pirate copy of the content. The distributor then listens in on the pirate broadcast, extracts the watermarks, and uses this information for the choice of watermarks for the next segment of the content. We assume that the pirates always try to keep their broadcast running so that if one of the pirates is disconnected, the other pirates will take over. Ideally, one demands that all the traitors are identified and accused, and the set of accused users exactly matches the coalition.

For a fingerprinting scheme $(\mathcal{C}, T)$, each element of $\mathcal{C}$, which is a sequence of fingerprint marks and denoted by $\mathbf{c} = (c_1, ..., c_l)$, is called a codeword. l is called code length. As a critical requirement, the embedding of the fingerprinting codewords should not affect the functionalities or quality of the digital object. Therefore, for a certain level of security, the shorter the code length, the better the code. The well-known Tardos codes proposed by G. Tardos in 2003 [10]

*Xin-Wen Wu is with School of Information and Communication Technology, Griffith University, Gold Coast, QLD 4222, Australia. Email: x.wu@griffith.edu.au

are the best known fingerprinting codes with regard to code length. Tardos codes are asymptotic optimal according to the low bound on code length proved by Peikert et al [6]. However, original Tardos codes are only applied to static traitor tracing; and the computational complexity of their tracing algorithm is at the order of number of users, N, that is, $O(N)$. The high computational complexity makes Tardos schemes infeasible to the applications which have a large number of users and require real-time tracing. In the recent paper [11], concatenating Tardos codes with algebraic-geometric codes, a class of near-optimal fingerprinting codes have been obtained. While the concatenated codes in [11] are still only for static schemes, they allow for a tracing algorithm with a significantly-reduced complexity, namely, $O(\log(N))$.

In a paper appeared in 2013 [5], the authors adapted Tardos codes and obtained a class of dynamic traitor tracing schemes. The complexity of the tracing procedure for the adapted Tardos dynamic schemes [5] remains high, that is, $O(N)$. In this extended abstract, we report our research of attempting to develop dynamic traitor-tracing schemes, using the near-optimal codes obtained from the concatenation of Tardos codes and algebraic-geometric codes, which allow a tracing procedure with complexity $O(\log(N))$.

II. Near-optimal fingerprinting codes

Peikert, et al in 2003 [6] have derived a lower bound on the code length l, that is, collusion-secure fingerprinting codes must have length at least $o(s^2\log(1/s\epsilon)$, where s is the size of coalition and ϵ is the probability that the tracing procedure fails to find any traitor. Codes that achieve the lower bound are called *optimal codes*.

In their pioneering work [1] Boneh and Shaw in 1998 presented a class of fingerprinting codes, which are now called Boneh-Shaw codes in the literature. The length of Boneh-Shaw codes is $O(Ns^2\log(N/\epsilon))$, which finds at least one traitor with a small error probability for any coalition consisting of at most s (where $s < N$) traitors. Concatenating Boneh-Shaw codes with random codes, Boneh and Shaw [1] have also constructed a class of s-secure codes of length $O(s^4\log(N/\epsilon)\log(1/\epsilon))$.

The best known codes have been constructed by Tardos [10]. The codes have a length $O(s^2\log(1/\epsilon))$ which find a least one traitor with error probability $\epsilon_T = \varepsilon^{s/4}$ for any coalition of size at most s. Boneh-Shaw codes and Tardos codes are both for static traitor tracing; and the computational complexity of the tracing procedures for these codes is $O(N)$.

In [3] Cortrina-Navau and Fernández constructed a class of s-secure codes for static schemes. The length of these codes is $O(s^6\log(s/\epsilon)\log(N))$; and the codes allow for a tracing algorithm with the computational complexity $O(\log(N))$. These codes were constructed by concatenating Boneh-Shaw codes (as inner codes) and algebraic-geometric codes (as outer codes). While the codes in [3] allow for an efficient tracing algorithm, they are very long, comparing with Tardos codes.

In this work, we start with the near optimal codes constructed by concatenating Tardos codes and algebraic-geometric codes, which are the same as in our previous work [11] and have code length $O(s^2\log(1/\epsilon)\log N)$. Motivated by the idea in [5], we will adapt the concatenated scheme in [11] to dynamic traitor tracing. We will show that the adapted dynamic traitor-tracing is efficient, and has a reduced complexity, namely, $O(\log(N))$.

In the following for the convenience of reader, we give a brief review of Tardos codes and the tracing algorithm. The details of algebraic-geometric codes can be found in the textbook [9].

Suppose M and s are positive integers, and ε is a real number with $0 < \varepsilon < 1$. Let $m = 100s^2\lceil\log(1/\varepsilon)\rceil$. An $(M \times m)$ matrix, denoted by $\mathbf{X}$, is defined as follows.

First, let p_i be independent, identically distributed random variables from $[t, 1-t]$ for $i = 1, ..., m$. Here we set $t = 1/(300s)$, and $p_i = \sin^2(r_i)$ is selected by picking uniformly at random the value $r_i \in [t', \pi/2 - t']$, where $0 < t' < \pi/4$, $\sin^2 t' = t$.

Next, for $j = 1, ..., M$ and $i = 1, ..., m$, let X_{ji} be independently from the binary alphabet $\{0, 1\}$ with $Pr(X_{ji} = 1) = p_i$. Notice that the random variables X_{ji} and $X_{j'i}$ are positively correlated as both of them tend to be 1 when p_i is large. Let $\mathbf{X} = (X_{ji})$ be the $M \times m$ binary matrix with the entries X_{ji}.

Taking each row of $\mathbf{X}$ as a codeword, we get a code (that is, the Tardos code)

$$\mathcal{C}_T = \{\mathbf{c}_j \mid \mathbf{c}_j = (X_{j1}, ..., X_{jm}),\ 1 \le j \le M\}.$$

Obviously, the code $\mathcal{C}_T$ has a length

$$m = 100s^2\lceil\log(1/\varepsilon)\rceil,$$

and consists of M codewords (users).

The tracing algorithm T for Tardos code is given in the following. First, an $(M \times m)$ matrix $U = (U_{ji})$ is defined, with entries

$$U_{ji} = \begin{cases} \sqrt{\frac{1-p_i}{p_i}} & \text{if } X_{ji} = 1, \\ -\sqrt{\frac{p_i}{1-p_i}} & \text{if } X_{ji} = 0. \end{cases}$$

Given a pirated fingerprint $\mathbf{y} = (y_1, ..., y_m) \in \{0, 1\}^m$ as the input, the tracing algorithm T includes a user $j \in \{1, 2, ..., M\}$ (or equivalently, the codeword, $\mathbf{c}_j \in \mathcal{C}_T$, assigned to user j) in its output, if

$$\sum_{i=1}^{n} y_i U_{ji} > Z,$$

where $Z = 20s\lceil\log(1/\varepsilon)\rceil$ is a threshold parameter. In other words, $T(\mathbf{y})$ contains the user j for whom the jth entry of $U\mathbf{y}^T$ exceeds Z.

From the construction and tracing algorithm above, Tardos codes are s-secure codes of length $O(s^2 \log(1/\epsilon))$ with error probability $\epsilon_T = \varepsilon^{s/4}$. As $0 < \varepsilon < 1$, we have $\epsilon_T < \varepsilon$, which can be any small real number.

Our code construction is based on the standard concatenation construction (see [3], [11]). Let W be a $[n, k, d]$ error-correcting code over $\mathbf{F}_q$, the finite field of q elements, where n is the code length, k the dimension, that is, the code has q^k codewords, and d the minimum Hamming distance. Let V be another code over an alphabet $\mathcal{A}$ (for example, $\mathcal{A} = \{0, 1\}$ for Tardos codes), which has q codewords. Then the concatenated code $\mathcal{C} = V \circ W$ is the code obtained by taking the words $\mathbf{w} = (w_1, ..., w_n) \in W \subseteq \mathbf{F}_q^n$, and mapping every symbol $w_i \in \mathbf{F}_q$ onto a codeword $V(w_i) \in V$. In this construction, V is called the inner code and W the outer code. Note that $\mathcal{C} = V \circ W$ is a code over the alphabet $\mathcal{A}$. It is a well known fact that if the inner and outer codes have lengths m and n, respectively, then the length l of the concatenated code is $m \cdot n$. In our construction, we take the binary Tardos code $\mathcal{C}_T$ as the inner code. Suppose $\mathcal{C}_T$ has $M = q$ (which is 2^K for an integer K). The outer code W is an algebraic-geometric code over $\mathbf{F}_q$ with appropriate code parameters.

The length of the inner code (i.e., Tardos code) is $m = O(s^2 \log(1/\epsilon))$. Hence, the concatenated code $\mathcal{C}$ has a length $O(s^2 \log(1/\epsilon)n)$, where n is the length of the algebraic-geometric code. As algebraic-geometric codes are linear codes, the code length $n = O(k) = O(\log N)$, where N is the number of codewords (which is equal to the number of users in traitor tracing scheme). Therefore, the length of $\mathcal{C}$ is $O(s^2 \log(1/\epsilon)n) = O(s^2 \log(1/\epsilon) \log N)$.

Theorem *Let V be a s-secure Tardos code with error probability ϵ_T. Making use of an algebraic-geometric code with appropriate code parameters as the outer code W, we have that the concatenated code $\mathcal{C} = V \circ W$ is a s-secure code with an exponentially small error probability. The length of the s-secure code $\mathcal{C}$ is $l = O(s^2 \log(1/\epsilon) \log N)$.*

III. Dynamic tracing and computational complexity

Let $\mathbf{s}^1, ..., \mathbf{s}^s$ be the codewords associated to a coalition S, where each $\mathbf{s}^j = (V(s_1^j), ..., V(s_n^j)) \in \mathcal{C}$ and $(s_1^j, ..., s_n^j) \in W$. The procedure of dynamic traitor tracing consists of the following steps.

Initial step: Set a threshold Z according to the number of users and the code construction of Tardos code and algebraic-geometric code (which is similar to that of [5]).

For each i, where $i < l = m \cdot n$ (l is the length of fingerprinting code or equivalently the number of segments of the digital content):

Step 1: Decode each of the blocks $(y_1^1, ..., y_m^1)$, ..., $(y_1^i, ..., y_m^i)$ using the tracing algorithm for the Tardos inner code V, (see [10], [5]), to obtain a sequence of fingerprint marks $Y_1, ..., Y_i \in \mathbf{F}_q$.

Step 2: Given $(Y_1, ..., Y_i) \in \mathbf{F}_q^i$, use the list decoding algorithm for the algebraic-geometric code [3], [11] (here, we actually decode the punctured code obtained by deleting the last $n - i$ coordinates of the algebraic-geometric code W), to find all the codewords $\mathbf{c}$ of the punctured code, such that $\mathbf{c}$ and $(Y_1, ..., Y_i)$ agree in at least Z coordinates.

Step 3: Disconnect all the active users corresponding to these words $\mathbf{c}$. Set $i = i + 1$ and return to Step 1. If there is no $\mathbf{c}$ satisfying the above-mentioned threshold condition, do steps 1-3 for $i = i + 1$.

To recover each of the marks $Y_1, ..., Y_i \in \mathbf{F}_q$, the computational complexity is $O(Mm)$, where M and m are the number of codewords and length of the inner code, respectively, which both are at the order of $O(q)$. Thus, the complexity is $O(nq^2)$, as the procedure requires to recover at most n symbols. For the decoding of outer code, it is well known that for an algebraic-geometric code [3], an efficient list decoding algorithm has a time complexity $O(n^3)$. Also, we know that for algebraic-geometric codes, $q = O(poly(n))$. Therefore, the overall complexity for the tracing procedure is $O(nq^2 + n^3)$, which is polynomial in $n = O(l) = O(\log(N))$.

References

[1] D. Boneh, and J. Shaw, "Collusion secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, 1998, **44**, (5), pp. 1897-1905.

[2] B. Chor, A. Fiat and M. Naor, "Tracing traitors," in *Advances in Cryptology - Crypto'94 (Lecture Notes in Computer Science)*, Berlin, Germany: Springer-Verlag, 1994, vol.839, pp.480-491.

[3] J. Cortrina-Navau, and M. Fernández, "A family of asymptotically good binary fingerprinting codes," *IEEE Trans. Inform. Theory*, 2010, **56**, (10), pp. 5335-5343.

[4] A. Fiat, and T. Tassa, "Dynamic traitor tracing," *J. Cryptology*, 2001, **14**, pp.211-223.

[5] T. Laarhoven, J. Doumen, P. Roelse, B. Škorić, and B. de Weger, "Dynamic Tardos traitor tracing schemes," *IEEE Trans. Inform. Theory*, 2013, **59**, (7), pp. 4230-4242.

[6] Peikert, C., Shelat, A., and Smith, A, "Lower Bounds for Collusion-Secure Fingerprinting," *Proc. 14th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA'03)*, 2003, 472-479.

[7] R. Safavi-Naini, and Y. Wang, "Sequential traitor tracing," *IEEE Trans. Inform. Theory*, 2003, **49**, (5), pp. 1319-1326.

[8] Škorić, B., Katzenbeisser, and S., Celik, M.U, "Symmetric Tardos fingerprinting codes for arbitrary alphabet sizes," *Des. Codes Cryptogr.*, Vol.46, 2008, 137-166.

[9] M.A. Tsfasman, and S.G. Vladut, *Algebraic-Geometric Codes*, Boston, MA: Kluwer Academic, 1991.

[10] G. Tardos, "Optimal probabilistic fingerprint codes," in *Proceedings of 35th Annual ACM Symposium on Theory of Computing (STOC)*, San Diego, USA, June 9-11, 2003, pp.116-125.

[11] X.-W. Wu, A. W.-C. Liew, "Near-Optimal Collusion-Secure Fingerprinting Codes for Efficiently Tracing Illegal Re-distribution," in: *Cyberspace Safety and Security*, LNCS 7672, (Eds., Y. Xiang, J. Lopez, C.-C. Jay Kuo, & W. Zhou), 2012, pp 352-361. (Presented at the International Symposium on Cyberspace Safety and Security, Melbourne, Australia, December 12-13, 2012).