University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information Sciences

2003

Sequential Traitor Tracing

R. Safavi-Naini
University of Wollongong, rei@uow.edu.au

Yejing Wang
University of Wollongong

Publication Details

This paper originally appeared as: Safavi-naini, R and Wang, Y, Sequestion Traitor Tracing, IEEE Transactions on Information Theory, May 2003, 49(5), 1319-1326. Copyright IEEE 2003.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Sequential Traitor Tracing

Abstract
We consider a new type of traitor tracing scheme, called sequential traitor tracing, that protects against rebroadcasting of decrypted content. Sequential traceability (TA) schemes trace all up to c traitors and remove the shortcomings of dynamic tracing schemes. We give two general constructions and show the relationship between c-TA codes and sequential tracing schemes.

Keywords
cryptography, error correction codes, watermarking

Disciplines
Physical Sciences and Mathematics

Publication Details
This paper originally appeared as: Safavi-naini, R and Wang, Y, Sequestion Traitor Tracing, IEEE Transactions on Information Theory, May 2003, 49(5), 1319-1326. Copyright IEEE 2003.

This journal article is available at Research Online: http://ro.uow.edu.au/infopapers/132

Sequential Traitor Tracing

Reihaneh Safavi-Naini, *Member, IEEE*, and Yejing Wang

***Abstract*—We consider a new type of traitor tracing scheme, called sequential traitor tracing, that protects against rebroadcasting of decrypted content. Sequential traceability (TA) schemes trace *all* up to c traitors and remove the shortcomings of dynamic tracing schemes. We give two general constructions and show the relationship between c-TA codes and sequential tracing schemes.**

***Index Terms*—Error-correcting codes, fingerprinting, traitor tracing.**

I. Introduction

TRAITOR tracing is studied in different contexts. In *broadcast encryption* [1], the information is broadcasted to a set of authorized receivers. Each receiver has a decoder with a unique key that allows him to decrypt the encrypted broadcast. Traitor tracing schemes protect against a pirate decoder that is constructed by a group of colluders that use their key information to illegally decrypt the broadcast. Traitor tracing is also used in the context of data fingerprinting, where colluders use their copies of a digital object to construct a pirate copy of the object; again the aim is to trace one of the colluders.

Dynamic tracing [2], [3] considers the scenario where the content is broadcasted to a group of authorized receivers. Here, the colluders do not construct a pirate decoder but decrypt the content and rebroadcast it. A simple solution to trace the source of rebroadcast is to embed a different watermark for each receiver and trace the source of rebroadcast by examining the embedded watermark in the rebroadcast. This solution, however, requires one copy of content for each user and so requires very high bandwidth. Dynamic tracing [2] allows tracing all colluders with much lower bandwidth. The basic idea is to break time into consecutive intervals and modify the watermarking strategy of the system in each interval using the rebroadcasted content. After observing the rebroadcast for long enough time, one or more colluders can be traced. The identified colluders are disconnected from the system and the system proceeds until all colluders are found one by one and get disconnected. Dynamic tracing has two main drawbacks. First, it is completely ineffective against an attack, called *delayed rebroadcast*, in which the attackers rebroadcast the content with some delay. Under this attack not even a single colluder can be found. Second, it requires high real-time computation and so is not suitable for systems with a large number of users. Sequential tracing, first proposed in [4], removes these two shortcomings by tracing at least one colluder if delayed rebroadcast attack is used and substantially reducing the real-time computation cost. This paper is an extended and corrected version of that paper.

The paper is organized as follows. In Section II, we recall dynamic tracing and point out its shortcomings. In Section III, we introduce the model of sequential tracing, and in Sections IV and V, describe two constructions, one based on function families and the other based on error-correcting codes. In Section VI, we investigate the relationship between sequential c-traceability (TA) schemes and c-TA codes. In Section VII, we provide a bound on the size of collusion for codes that are obtained from error-correcting codes. In Section VIII, we discuss our results and propose possible extensions.

II. Dynamic Tracing

A q-ary watermarking system with *mark set* $\mathcal{W} = \{1, 2, \ldots, q\}$ consists of two algorithms: an *embedding algorithm* I that embeds one of the q marks in a content, and a *detection algorithm* D that takes a content and outputs one of the q marks, or "?." To provide protection against removal of the mark, the two algorithms may use the same secret key information. We assume the watermarking system is *robust* and the embedded mark cannot be changed or removed. An example of such scheme is the *spread-spectrum technique* of Cox *et al.* [5].

In dynamic tracing [2], [3], the *content* is divided into consecutive *segments*, for example, one minute interval in an audio track. A watermarking algorithm is used to embed one of the q marks in the segment, hence creating q *versions* of the segment. In each interval, the user group is divided into q subsets and each subset receives one version of the segment. The subsets are varied in each interval using the rebroadcasted content. It is assumed that there is an efficient group key management scheme that allows the broadcaster to efficiently regroup the receivers in each interval and securely deliver their allocated version. Fiat and Tassa proved [2], [3] that for tracing c traitors at least $c+1$ versions must be used, and gave algorithms that use $c+1$ and $2c+1$ versions and require $O(3^c c \log n)$ and $O(c \log n)$ steps, respectively, to find *all colluders*, that is *to converge*. Berkman *et al.* [6], [7] improved these results and showed an algorithm with $c+a+1$, $a \leq c$, versions and $O(c \log n + c^2/a)$ steps for convergence, and a second one with $ca+1$, $a \geq 2$, versions and $O(c \log_a n)$ steps for convergence. The main aim of these works has been to construct schemes with the smallest number of steps for convergence when the number of versions is close to the lower bound $c+1$.

Dynamic tracing has two major shortcomings. The first shortcoming is that regrouping of the users and mark allocation to users in each interval depends on the rebroadcasted content,

Manuscript received January 7, 2002; revised January 13, 2003. The material in this paper was presented in part at CRYPTO 2000, Santa Barbara, CA, August 2000.

The authors are with the School of Information Technology and Computer Science, University of Wollongong, Wollongong 2522, Australia (e-mail: rei@uow.edu.au; yejing@uow.edu.au).

Communicated by N. I. Koblitz, Associate Editor for Complexity and Cryptography.

Digital Object Identifier 10.1109/TIT.2003.810629

0018-9448/03$17.00 © 2003 IEEE

also called *feedback* from the channel. This means that if there is no feedback from the channel no regrouping will occur and so the system is vulnerable to a *delayed rebroadcast attack*. In this attack, the attackers do not immediately rebroadcast, but record the content and rebroadcast it with some delay and so the broadcaster has no alternative but keeping the mark allocation unchanged. With this attack the system fails completely and cannot trace any colluder.

The second shortcoming of the system is high real-time computation for regrouping the users and allocating marks to subsets. This means that the length of a segment cannot be very short. In dynamic tracing, the number of segments required by the algorithm to converge grows with the number of users. Hence, to trace colluders given a fixed length content, the length of the segment must reduce as the size of the user population grows. On the other hand, the computation for repartitioning the group and allocating the versions grows with the size of the group and because of the real-time nature of the computation, the length of the segment cannot be decreased. The conflicting requirements on the segment size, that is, requiring shorter length to provide for longer convergence length, and at the same time the need to have it longer to give time for real-time computation, would result in unworkable systems for large groups.

III. Sequential Tracing

We consider the same scenario as dynamic tracing and propose a different solution which removes the above shortcomings. In sequential tracing, the channel feedback is only used for tracing and *not* for allocation of marks to users. Similar to dynamic tracing, the system can trace all colluders. The mark allocation table is predefined and there is no need for real-time computation to determine the mark allocation of the next interval. Other computations related to key management of the group can be all performed as precomputation and so the need for real-time computation will be minimized. Mark allocation in each interval will be according to the table irrespective of the channel feedback. Using a predefined table also protects against the delayed rebroadcast attack, however, the tracing ability of the system will reduce to one traitor. That is, even if the rebroadcast is delayed until the whole content is received, still once colluders start rebroadcasting at least one of them will be traced. We call the system *sequential traitor tracing scheme* to emphasise the fact that the traitors are identified sequentially, that is, when a colluder is found he is disconnected and the system proceeds to trace the remaining colluders, and at the same time differentiate it from dynamic schemes.

System Description

In sequential tracing, the *protected content* is divided into *segments*. A q-ary watermarking system is used to produce q versions of each segment. Let $U = \{u_1, u_2, \ldots, u_N\}$ denote the set of users, and $\mathcal{W} = \{1, 2, \ldots, q\}$ be the mark set. A *mark allocation table* M is an array over $\mathcal{W}$ with N rows and L columns where $M(i, j)$ is the mark allocated to the user u_i in segment j. In the jth time interval, the broadcaster uses the jth column of M to allocate marks to users. A colluding group C chooses one of their versions and rebroadcasts it. The tracer intercepts the rebroadcast, extracts the mark, and appends it to a sequence, called *feedback sequence*. Let $C \subseteq U$ denote a colluding group, and

$$W_j(C) = \{f_j \colon f_j \in \{M(i, j) \colon u_i \in C\}\}.$$

A feedback sequence $F = (f_1, f_2, \ldots, f_L)$ is called *c-feedback sequence* if there exists a $C \subseteq U$, $|C| \leq c$, such that $f_j \in W_j(C)$ for $j = 1, 2, \ldots, L$. After observing a certain number of segments from a feedback sequence, the tracer identifies one traitor (or more traitors) and disconnects him from the system, that is, excludes him from future broadcasts. The tracer continues observing the rebroadcast and identifying other traitor(s) who will be disconnected in a similar way. After observing L elements of the feedback sequence, all traitors are found and the tracing algorithm converges, here L is the *convergence length* of the algorithm. Let F_j denote the subsequence consisting of the first j elements of F.

Definition 1: A sequential c-TA scheme consists of a mark allocation table M and a tracing algorithm A with the following properties:

1) $M = (m_{ij})$ is an $N \times L$ array with entries from $\mathcal{W}$;
2) A is a mapping

$$A \colon \mathcal{W}^* \longrightarrow 2^U$$

such that for any c-feedback sequence F, there exists a sequence of integers

$$0 < d_1(F) < d_2(F) < \cdots < d_k(F) \leq L$$

such that

$$A(F_j) = \begin{cases} C_j, & \emptyset \neq C_j \subseteq U \Big\backslash \bigcup_{i=1}^{j-1} C_i, \\ & j = d_1(F), d_2(F), \ldots, d_k(F) \\ \emptyset, & \text{otherwise} \end{cases} \tag{1}$$

and

$$\bigcup_{j=1}^{k} C_{d_j(F)} = C. \tag{2}$$

The colluders are identified in k steps, where k depends on the feedback sequence and $k \leq c$ because it is possible to identify more than one colluder in one step. The working of a sequential c-TA scheme, denoted by (M, A), can be summarized as follows.

```
Set j = 1, n = N, F_0 = ()
While j < L and n > N − c
   For i = 1, ..., n
         Send version M(i, j) to u_i
   If there is feedback
         Extract f_j
         Append f_j to F_{j−1}
         If A(F_j) ≠ ∅
            disconnect users in A(F_j)
            n = n − |A(F_j)|
   j = j + 1
```

Disconnecting a user (or a group) means that they cannot contribute to future rebroadcasts and the feedback sequence will not have their contributions. Three parameters q, L, and c measure communication efficiency of the system. Parameter q is the number of versions of a segment and so higher q means more versions and higher bandwidth for sending the segment. Parameter L is the *convergence length* of the system and its higher value means more segments are required to trace all traitors. Parameter c is the maximum number of traitors the system tolerates.

Intuitively, there is a tradeoff between these two parameters. That is, using more versions would allow shorter convergence length. In Section VII, we give an expression that relates these three parameters when the mark allocation table is derived from an error-correcting code. In the following two sections, we describe two constructions of sequential TA schemes.

IV. A Construction Using a Function Family

This construction identifies one of the c colluders in c^2+1 steps, and *all* colluders in at most c^2+c steps. That is, the scheme converges in c^2+c steps and the convergence length is independent of the size of the user group. However, the number of versions is proportional to the group size and so for large groups requires high bandwidth. We will show (Section VIII) that the scheme can be recursively used to increase the number of users while the number of versions is kept fixed. This will be at the cost of higher convergence length.

A. Mark Allocation Table

Let $\mathcal{W}=\{1, 2, \ldots, q\}$ be the mark set, b and m be integers, where $b \le q$. Consider a collection of mappings

$$\Phi = \{\phi_{ij}\colon 1 \le i \le b,\ 1 \le j \le m\}$$

where

$$\phi_{ij}\colon \mathcal{W} \to \mathcal{W}$$

satisfies the following two properties.

(P1) For a fixed j, and a pair of the first indexes (i_1, i_2), $i_1 \ne i_2$, we have $\phi_{i_1 j}(x) \ne \phi_{i_2 j}(x)$ for all $x \in \mathcal{W}$.

(P2) For a pair of the first indexes (i_1, i_2), and a pair of the second indexes (j_1, j_2) with $j_1 \ne j_2$, if $\phi_{i_1 j_1}(x_1) = \phi_{i_2 j_1}(x_2)$ we have $\phi_{i_1 j_2}(x_1) \ne \phi_{i_2 j_2}(x_2)$, for all $x_1, x_2 \in \mathcal{W}$, $x_1 \ne x_2$.

Let M_0 and $\phi_{ij}(M_0)$, $1 \le i \le b$, $0 \le j \le m$, be the following $q \times 1$ matrices:

$$M_0 = \begin{pmatrix} 1 \\ 2 \\ \vdots \\ q \end{pmatrix}, \qquad \phi_{ij}(M_0) = \begin{pmatrix} \phi_{ij}(1) \\ \phi_{ij}(2) \\ \vdots \\ \phi_{ij}(q) \end{pmatrix}.$$

Define a mark allocation table M as follows:

$$M = \begin{pmatrix} M_0 & \phi_{11}(M_0) & \phi_{12}(M_0) & \cdots & \phi_{1m}(M_0) \\ M_0 & \phi_{21}(M_0) & \phi_{22}(M_0) & \cdots & \phi_{2m}(M_0) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_0 & \phi_{b1}(M_0) & \phi_{b2}(M_0) & \cdots & \phi_{bm}(M_0) \end{pmatrix}. \tag{3}$$

That is, M has b *block rows*, each containing q rows. Each row is allocated to a user and (r, k) denotes the kth row of the rth block row. For a feedback sequence $F_j = (f_1, f_2, \ldots, f_j)$, we say F_j and u_i *have s matches* if there exist indexes $j_1 < j_2 < \cdots < j_s$ such that

$$f_{j_1} = M(i, j_1),\ f_{j_2} = M(i, j_2),\ \ldots,\ f_{j_s} = M(i, j_s).$$

Let $\rho(F, u)$ denote the number of matches between F and u. Define a tracing function $A\colon \mathcal{W}^* \to 2^U$ as follows:

$$A(F_j) = \{u_i\colon \rho(F_j, u_i) = c+1\}. \tag{4}$$

The tracing algorithm can be implemented by attaching a counter to each row. The counter records the number of matches between a row and F_j. In step j, f_j is received from the channel. The row counters of rows that have f_j in their jth position will be incremented. When a row counter reaches $c+1$, the corresponding user is identified as a traitor and is disconnected from future broadcast. Let c be the following integer:

$$c = \left\lfloor \frac{-1+\sqrt{5+4m}}{2} \right\rfloor. \tag{5}$$

Theorem 1: The mark allocation table M and the tracing function A, as defined in (3) and (4), respectively, define a sequential c-TA scheme for $N = bq$ users. The convergence length is $L = m+1$ and c is given in (5).

The proof of this theorem uses Lemmas 1 and 2 that follow.

Lemma 1: Let

$$F_j = (f_1, f_2, \ldots, f_j), \qquad j \ge c+1 \tag{6}$$

be a c-feedback sequence produced by $C \subseteq U$, $|C| \le c$. Suppose $\rho(F_j, u) = c+1$. Then $u \in C$.

Proof: Assume $u \notin C$ and $\rho(F_j, u) = c+1$. Suppose u corresponds to the row (r, k) of M. Consider the $c+1$ positions that F and (r, k) match. Since F is constructed by C and $|C| \le c$, there must be a colluder who has contributed two segments in that $c+1$ positions. Let the two segments be j_1 and j_2, $j_1 < j_2$. That is, there is $(r', k') \in C$ that matches F in positions j_1, j_2, and hence matches (r, k) in positions j_1 and j_2. Because of the structure of M, we have one of the following two cases:

1) $j_1 = 1$ which gives

$$k = k' = f_{j_1}, \qquad \phi_{r, j_2}(k) = \phi_{r', j_2}(k) = f_{j_2}$$

or

2) $j_1 > 1$ which gives

$$\phi_{r, j_1}(k) = \phi_{r', j_1}(k') = f_{j_1}, \quad \phi_{r, j_2}(k) = \phi_{r', j_2}(k') = f_{j_2}.$$

The first case contradicts (P1). The second case contradicts (P2) as $j_1 \neq j_2$. □

Corollary 1: i) Tracing one traitor requires $c+1$ segments from the traitor. ii) Tracing t traitors in a single step requires at most $tc+1$ segments from them.

Proof: Part i) is a direct corollary of Lemma 1. To prove Part ii), we note that if $\rho(F, u) = c+1$ then u is a traitor. If in segment j, t traitors are simultaneously identified, it must be the case that the t users have simultaneously reached $c+1$ matches with the c-feedback sequence F_j, and so each of the t users have c matches with F_{j-1}. The number $j-1$ is bounded by $j-1 \leq tc$ and so tracing t traitors in a single step requires $j \leq tc+1$ segments. □

Consider a c-feedback sequence F. Let $d_0 = 0$. For $j \geq 1$, if $\{u \in U \backslash A(F_{j-1}) : \exists \ell > d_{j-1} \text{ such that } \rho(F_\ell, u) = c+1\} \neq \emptyset$ define

$$d_j = \min\{\ell \colon \exists\, u \in U \setminus A(F_{j-1}) \text{ such that } \rho(F_\ell, u) = c+1\}. \tag{7}$$

Lemma 2: Let $F = (f_1, f_2, \ldots, f_L)$ be a c-feedback sequence and $d_1 < d_2 < \cdots < d_k$ be defined in (7). Then, $d_j \leq c^2 + j$ for $j = 1, 2, \ldots, k$.

Proof: Let C be the collusion produced F and

$$T_j = A(F_{d_1}) \cup A(F_{d_2}) \cup \cdots \cup A(F_{d_j}).$$

For $u_t \in C \setminus T_j$, define

$$b_t = |\{h \leq d_j \colon f_h = M(t, h)\}|.$$

From (4) we have $b_t \leq c$. Applying Corollary 1 we obtain that

$$\begin{aligned} d_j &\leq \sum_{i=1}^{j} \left(|A(F_{d_i})|\, c + 1\right) + b_1 + b_2 + \cdots + b_{c-|T_j|} \\ &\leq |T_j| c + j + (c - |T_j|) c = c^2 + j \end{aligned}$$

which proves the lemma. □

Proof of Theorem 1: Consider $d_1, d_2, \ldots, d_k$ defined in (7). Using Lemma 2, we have $d_k \leq c^2 + c$ as $k \leq c$. If c is given by (5), then $c^2 + c \leq m+1$. That is, $d_k \leq L$. So (M, A) is a sequential c-TA scheme. The number of users N is the number of rows of M and is equal to bq.

B. Existence of Φ

The construction in Section IV-A uses a function family Φ that satisfies properties (P1) and (P2). In the following, we give a construction for Φ.

Theorem 2: Let p be a prime number, $Z_p^* = \{1, 2, \ldots, p-1\}$. Define a function family $\Phi = \{\phi_{ij} \colon 1 \leq i, j \leq (p-1)/2\}$ as follows:

$$\phi_{ij} : Z_p^* \to Z_p^*,\ \phi_{ij}(x) = (i+j)x \pmod p.$$

Then, Φ satisfies properties (P1) and (P2).

Proof: If $i_1 \neq i_2$, then $\phi_{i_1 j}(x) \neq \phi_{i_2 j}(x)$ for all $x \in Z_p^*$ and so (P1) is satisfied. Assume $\phi_{i_1, j_1}(x) = \phi_{i_2, j_1}(y)$. Then we have $i_1 x - i_2 y = j_1(y - x)$ and so if $j_2 \neq j_1$ we have $i_1 x - i_2 y \neq j_2(y - x)$. This implies $\phi_{i_1, j_2}(x) \neq \phi_{i_2, j_2}(y)$, and hence (P2) is satisfied. □

The mark allocation table resulting from Φ in Theorem 2 will have $N = (p-1)^2/2$ rows and $L = 1 + (p-1)/2$ columns, and will be a sequential c-TA scheme with

$$c = \left\lfloor \frac{-1 + \sqrt{3 + 2p}}{2} \right\rfloor.$$

For given N and c, p must be chosen such that $(p-1)^2/2 \geq N$ and $c^2 + c \leq 1 + (p-1)/2$ and so

$$p \geq \max(1 + \sqrt{2N},\ 2c^2 + 2c - 1).$$

C. An Example

Example 1: To provide protection for 50 users against collusion of up to two colluders, $c = 2$, we need p in Theorem 2 to be 11. M will have five block rows, each with 10 rows as shown at the top of the following page.

Suppose users $(1, 10)$ and $(4, 2)$ are the colluders, and assume the feedback sequence is $F = (10, 10, 8, 3, 6, 7)$. We expect to identify the first colluder after observing at most $c^2 + 1 = 5$ segments. That is after observing at most five segments, there will be at least one row that will match the feedback sequence in $c + 1 = 3$ positions while all other rows will have at most $c = 2$ matches. We note that length 5 corresponds to the worst (longest) case when both colluders have attempted to remain unknown for the longest period. However, it means that when the first colluder is identified (three matches), the other one has already two matches with F and will be caught after observing the next segment. The following table lists columns that match a particular segment in the feedback sequence. After observing five segments of the feedback sequence, the colluder $(1, 10)$ will have three matches and will be disconnected. The colluder $(4, 2)$ has appeared only twice which is the same number as some of the innocent users, for example, $(4, 1)$. After observing the sixth element of the feedback sequence, this colluder will also be identified.

10	10	8	3	6	7
↓	↓	↓	↓	↓	↓
(1, 10)	(1, 5)	(1, 10)	(1, 9)	(1, 10)	(1, 3)
(2, 10)	(2, 7)	(2, 2)	(2, 5)	(2, 1)	(2, 1)
(3, 10)	(3, 8)	(3, 6)	(3, 6)	(3, 4)	(3, 5)
(4, 10)	(4, 2)	(4, 5)	(4, 2)	(4, 9)	(4, 2)
(5, 10)	(5, 9)	(5, 9)	(5, 10)	(5, 8)	(5, 4)

V. Construction From Error-Correcting Codes

Mark allocation tables can be obtained from error-correcting codes by using each codeword as a row of the table.

Definition 2: An $(L, N, D)_q$-error-correcting code, or $(L, N, D)_q$-ECC for short, is a set of N codewords of length L over an alphabet of size q and having the minimum Hamming distance between any pair of codewords equal to D.

The mark allocation table in (3) corresponds to an error-correcting code with $D = L - 1$. A general construction from

Mark Allocation Table for Example 1

(1, 1):	1	2	3	4	5	6	(2, 1):	1	3	4	5	6	7	(3, 1):	1	4	5	6	7	8
(1, 2):	2	4	6	8	10	1	(2, 2):	2	6	8	10	1	3	(3, 2):	2	8	10	1	3	5
(1, 3):	3	6	9	1	4	7	(2, 3):	3	9	1	4	7	10	(3, 3):	3	1	4	7	10	2
(1, 4):	4	8	1	5	9	2	(2, 4):	4	1	5	9	2	6	(3, 4):	4	5	9	2	6	10
(1, 5):	5	10	4	9	3	8	(2, 5):	5	4	9	3	8	2	(3, 5):	5	9	3	8	2	7
(1, 6):	6	1	7	2	8	3	(2, 6):	6	7	2	8	3	9	(3, 6):	6	2	8	3	9	4
(1, 7):	7	3	10	6	2	9	(2, 7):	7	10	6	2	9	5	(3, 7):	7	6	2	9	5	1
(1, 8):	8	5	2	10	7	4	(2, 8):	8	2	10	7	4	1	(3, 8):	8	10	7	4	1	9
(1, 9):	9	7	5	3	1	10	(2, 9):	9	5	3	1	10	8	(3, 9):	9	3	1	10	8	6
(1, 10):	10	9	8	7	6	5	(2, 10):	10	8	7	6	5	4	(3, 10):	10	7	6	5	4	3
(4, 1):	1	5	6	7	8	9	(5, 1):	1	6	7	8	9	10							
(4, 2):	2	10	1	3	5	7	(5, 2):	2	1	3	5	7	9							
(4, 3):	3	4	7	10	2	5	(5, 3):	3	7	10	2	5	8							
(4, 4):	4	9	2	6	10	3	(5, 4):	4	2	6	10	3	7							
(4, 5):	5	3	8	2	7	1	(5, 5):	5	8	2	7	1	6							
(4, 6):	6	8	3	9	4	10	(5, 6):	6	3	9	4	10	5							
(4, 7):	7	2	9	5	1	8	(5, 7):	7	9	5	1	8	4							
(4, 8):	8	7	4	1	9	6	(5, 8):	8	4	1	9	6	3							
(4, 9):	9	1	10	8	6	4	(5, 9):	9	10	8	6	4	2							
(4, 10):	10	6	5	4	3	2	(5, 10):	10	5	4	3	2	1							

an error-correcting code is given in Theorem 3. We need the following lemmas.

Lemma 3: Let Γ be an $(L, N, D)_q$-ECC and

$$F_j = (f_1, f_2, \ldots, f_j), \qquad j \geq c(L-D)+1 \tag{8}$$

be a c-feedback sequence produced by $C \subseteq \Gamma$, $|C| \leq c$. If $\rho(F_j, u) = c(L-D)+1$, then $u \in C$.

Proof: Let $u \in \Gamma$ and F_j matches in $c(L-D)+1$ positions. Then, there exists a $u' \in C$ such that u' and u have at least $L-D+1$ matches within these positions, and so

$$d(u', u) = L - \rho(u', u) \leq L - (L-D+1) = D-1.$$

This shows that $u = u' \in C$. □

For an $(L, N, D)_q$-ECC, define a tracing function A: $\mathcal{W}^* \to 2^\Gamma$ as

$$A(F) = \{u \in \Gamma: \rho(F, u) = c(L-D)+1\}. \tag{9}$$

Corollary 2: The tracing function (9) for an $(L, N, D)_q$-ECC has the following properties.

i) Tracing one traitor requires $c(L-D)+1$ segments from a traitor.
ii) Tracing t traitors in a single step requires at most $tc(L-D)+1$ segments from them.

Proof: Property i) is a direct corollary of Lemma 3. To prove property ii), we note that if $\rho(F, u) = c(L-D)+1$ then u is a traitor. If in segment j, t traitors are simultaneously identified it means that the t users have reached $c(L-D)+1$ matches with the c-feedback sequence F_j simultaneously, and so each of the t users have $c(L-D)$ matches with F_{j-1}. The number $j-1$ is bounded by $j-1 \leq tc(L-D)$, so tracing t traitors in a single step requires $j \leq tc(L-D)+1$ segments. □

Consider a c-feedback sequence F. Let $d_0 = 0$. For $j \geq 1$, if $\{u \in \Gamma \backslash A(F_{j-1}) : \exists \ell > j-1 \text{ s.t. } \rho(F_\ell, u) = c(L-D)+1\} \neq \emptyset$ define

$$d_j = \min\{\ell: \exists\, u \in \Gamma \setminus A(F_{j-1}) \text{ s. t. } \rho(F_\ell, u) = c(L-D)+1\}. \tag{10}$$

Lemma 4: Let $F = (f_1, f_2, \ldots, f_L)$ be a c-feedback sequence and $d_1 < d_2 < \cdots < d_k$ be defined as in (10). Then $d_j \leq c^2(L-D)+j$ for $j = 1, 2, \ldots, k$.

Proof: Suppose F is produced by C. Let

$$T_j = A(F_{d_1}) \cup A(F_{d_2}) \cup \cdots \cup A(F_{d_j}).$$

For $u_t = (w_1^{(t)}, w_2^{(t)}, \ldots, w_L^{(t)}) \in C \setminus T_j$, define

$$b_t = |\{h \leq d_j: f_h = w_h^{(t)}\}|.$$

From (9), we have $b_t \leq c(L-D)$. Applying Corollary 2 we obtain that

$$\begin{aligned} d_j &\leq \sum_{i=1}^{j} (|A(F_{d_j})| c(L-D)+1) + b_1 + b_2 + \cdots + b_{c-|T_j|} \\ &\leq |T_j| c(L-D) + j + (c-|T_j|) c(L-D) \\ &= c^2(L-D)+j \end{aligned}$$

which proves the lemma. □

Theorem 3: Let c be an integer, Γ denote a mark allocation table obtained from an $(L, N, D)_q$-ECC satisfying

$$D \geq \left(1 - \frac{1}{c^2}\right) L + \frac{1}{c} \tag{11}$$

and A be defined as in (9). Then (Γ, A) is a sequential c-TA scheme.

Proof: Consider $d_1, d_2, \ldots, d_k$ defined in (10). Using Lemma 4, we have $d_k \leq c^2(L-D)+c$. The condition (11)

gives $c^2(L-D)+c \le L$ and so (Γ, A) is a sequential c-TA scheme. □

Theorem 3 shows that sequential tracing schemes can be constructed from q-ary error-correcting codes with large minimum distances and

$$c = \left\lfloor \frac{-1+\sqrt{1+4L(L-D)}}{2(L-D)} \right\rfloor.$$

Examples of codes that satisfy (11) are given in what follows.

Reed–Solomon Codes

A Reed–Solomon code (RS-code) over GF (q) is a linear code with $L = q-1$, $D = L-k+1$, and $N = q^k$ codewords. An RS-code defines a mark allocation table for a sequential c-TA scheme with

$$c = \left\lfloor \frac{-1+\sqrt{1+4(q-1)(k-1)}}{2(k-1)} \right\rfloor.$$

Algebraic-Geometry Codes

An algebraic-geometry code (AG-code) over GF (q), denoted by $[L, k, L+1-k-g]_q$, is a linear code of length L, $D = L+1-k-g$ where k and g are the dimension of the code and the genus of the algebraic curve, respectively. It is known [8] that AG-codes with parameters $[L, k, L+1-k-g]_q$ exist, if there exists an algebraic curve of genus g over GF (q) having L rational points. For $g = 1$, the curves of genus 1 are elliptic curves which are known to exist for any $k \le L-1$ and any $L \le N_q(1)$ where $q = p^m$ and $N_q(1)$ is defined as

$$N_q(1) = \begin{cases} q+\lceil 2\sqrt{q}\rceil, & p \mid \lceil 2\sqrt{q}\rceil,\ m \ge 3 \text{ is odd} \\ q+\lceil 2\sqrt{q}\rceil+1, & \text{else.} \end{cases}$$

When $g = 2$, the curve of genus 2 exists for any $k \le L-2$ and any $L \le N_q(2)$, where $q = p^m$ and $N_q(2)$ is given as follows.

If $m \equiv 0 \pmod 2$

$$N_q(2) = \begin{cases} q+4\sqrt{q}+1, & q \ne 4, 9 \\ 10, & q = 4 \\ 20, & q = 9. \end{cases}$$

If $m \equiv 1 \pmod 2$

$$n_q(2) = \begin{cases} q+2\lceil 2\sqrt{q}\rceil+1, & q \text{ nonspecial} \\ q+2\lceil 2\sqrt{q}\rceil, & \\ \qquad q \text{ special}, 2\sqrt{q}-\lceil 2\sqrt{q}\rceil > (\sqrt{5}-1)/2 & \\ q+2\lceil 2\sqrt{q}\rceil-1, & \\ \qquad q \text{ special}, 2\sqrt{q}-\lceil 2\sqrt{q}\rceil < (\sqrt{5}-1)/2. & \end{cases}$$

Here q is *special* means that either $p \mid \lceil 2\sqrt{q}\rceil$ or q is of the forms: $q = \ell^2+1$, $q = \ell^2+\ell+1$, or $q = \ell^2+\ell+2$ for some integer ℓ.

An AG-code defines the mark allocation table of a sequential c-TA scheme for which

$$c = \left\lfloor \frac{-1+\sqrt{1+4L(k+g-1)}}{2(k+g-1)} \right\rfloor.$$

Low-Rate Codes

Theorem 4 ([9, Lemma III.3]): For positive integers p, N, let $L = 8p\log N$. Then there exists an $(L, N, D)_{2p}$-ECC where $D > (1-\frac{1}{p})L$.

The code in Theorem 4 defines the mark allocation table of a sequential c-TA scheme for which

$$c = \left\lfloor \frac{-p+\sqrt{p^2+4pL^2}}{2L} \right\rfloor.$$

VI. Sequential c-Traceability Schemes and c-TA Codes

Mark allocation table in a sequential TA scheme is closely related to TA codes.

Definition 3 [10]: Let Γ be a q-ary code, and c be an integer, $C = \{u_1, \ldots, u_b\} \subseteq \Gamma$ be a collusion of size $b \le c$, where $u_i = (w_1^{(i)}, \ldots, w_L^{(i)})$. Define

$$\operatorname{desc}(C) = \{(x_1, \ldots, x_L): x_j \in \{w_j^{(i)}: 1 \le i \le b\},\ 1 \le j \le L\}.$$

Then Γ is called a c-TA code if the following condition is satisfied: for any $(x_1, \ldots, x_L) \in \operatorname{desc}(C)$ for some C with $|C| \le c$, there is a $u_i \in C$ such that

$$|\{j: x_j = w_j^{(i)}\}| > |\{j: x_j = w_j\}|$$

for any $(w_1, \ldots, w_L) \in \Gamma \setminus C$.

Staddon *et al.* [10] proved the following theorem.

Theorem 5: Let Γ be an $(L, N, D)_q$-ECC, and c be an integer. If

$$D > \left(1-\frac{1}{c^2}\right)L \tag{12}$$

then Γ is a c-TA code.

For an $(L, N, D)_q$-ECC, let Γ_j denote the $(L_j, N, D_j)_q$-ECC obtained from Γ by restricting each codeword to its first j components.

Theorem 6: Let Γ be sequential$c$-TA scheme obtained from an $(L, N, D)_q$-ECC with D satisfying (11). Then, there exists a sequence of integers $L \ge L_1 \ge L_2 \ge \cdots \ge L_c$ such that Γ_{L_j}, $1 \le j \le c$, is a $(c-j+1)$-TA code.

Proof: Let $L_j = c(c-j+1)(L-D)+1$. That is,

$$D-L = -\frac{L_j}{c(c-j+1)} + \frac{1}{c(c-j+1)}.$$

The minimum Hamming distance D_j of Γ_{L_j} satisfies

$$\begin{aligned} D_j &\ge D-(L-L_j) = D-L+L_j \\ &= -\frac{L_j}{c(c-j+1)} + \frac{1}{c(c-j+1)} + L_j \\ &> \left(1-\frac{1}{(c-j+1)^2}\right)L_j. \end{aligned}$$

From Theorem 5, Γ_{L_j} is a $(c-j+1)$-TA code. □

Corollary 3: Mark allocation table of a sequential c-TA scheme is a c-TA code.

VII. Asymptotic Bounds

Intuitively, we expect a tradeoff among q, L, and c. That is, we expect higher bandwidth results in a shorter convergence length. In the following, we give a relation between these parameters for mark allocation tables that are based on error-correcting codes.

Theorem 7 (Plotkin Bound, [11]): For an $(L, N, D)_q$-ECC, if $D > (1-1/q)L$, then

$$N \le \frac{D}{D-(1-1/q)L}.$$

The following Theorem shows a bound on c for a sequential c-TA scheme obtained from an $(L, N, D)_q$-ECC with D satisfying (11). This bound was stated in [12] for a c-TA obtained from an $(L, N, D)_q$-ECC with D satisfying (12). Here, we give an alternative proof.

Theorem 8: Let c be an integer, Γ be an $(L, N, D)_q$-ECC with D satisfying (11) and $\lim_{L\to\infty} \frac{L}{N} = 0$. Then

$$c < \sqrt{q + \left(\frac{q}{2L}\right)^2} - \frac{q}{2L}$$

for sufficiently large L.

Proof: Assume otherwise, that is,

$$c \ge \sqrt{q + \left(\frac{q}{2L}\right)^2} - \frac{q}{2L}.$$

This implies that $Lc^2 + qc - qL \ge 0$, or equivalently

$$\left(1 - \frac{1}{c^2}\right) L + \frac{1}{c} \ge \left(1 - \frac{1}{q}\right) L.$$

If D satisfies (11), then

$$D > \left(1 - \frac{1}{q}\right) L.$$

Applying Theorem 7, we have

$$N \le \frac{D}{D-(1-\frac{1}{q})L}$$

and so

$$D \le \frac{N}{N-1}\left(1 - \frac{1}{q}\right) L.$$

Note that

$$\begin{aligned}\lim_{L\to\infty} &\left(\frac{N}{N-1}\left(1-\frac{1}{q}\right)L - \left(1-\frac{1}{q}\right)L\right)\\ &= \lim_{L\to\infty} \frac{L}{N-1}\left(1-\frac{1}{q}\right)\\ &= 0.\end{aligned}$$

For sufficiently large L

$$\left(1 - \frac{1}{q}\right) L < D \le \frac{N}{N-1}\left(1 - \frac{1}{q}\right) L$$

gives a contradiction as $D$is an integer. □

VIII. Discussion

As noted earlier, for a fixed-size group there is a tradeoff between the alphabet size and the convergence length. That is, for shorter convergence length larger alphabet size is required. In the following, we show a method of composing TA codes that results in systems for larger group sizes while keeping the alphabet size constant but increasing the convergence length.

An (L, N) code over a q-ary alphabet $\mathcal{W}$ is a subset of $\mathcal{W}^L$ of size N. With this definition, a c-TA code and a mark allocation table of a sequential c-TA scheme are codes over q-ary alphabets.

Let Γ_0 be a q-ary (L_0, N_0) code and Γ_1 be an N_0-ary (L_1, N_1) code. Define the *composition* of the two codes to be a q-ary code Γ obtained by i) associating each codeword of Γ_0 with a symbol in the alphabet set of Γ_1 and ii) in each codeword of Γ_1, replacing symbols by their associated codewords of Γ_0.

The code Γ will be a q-ary code of length L_0L_1 with N_1 codewords. We refer to Γ_0 and Γ_1 as the *inner* and the *outer* code, respectively.

Theorem 9: Let Γ_0 and Γ_1 be a q-ary (L_0, N_0) code and an N_0-ary (L_1, N_1) code, respectively, and let Γ denote the composition of the inner code Γ_0 and the outer code Γ_1. Suppose Γ_0 is a c-TA code and Γ_1 is a sequential c-TA scheme. Then Γ is an (L_0L_1, N_1) sequential c-TA scheme over a q-ary alphabet.

Proof: Let Γ be the mark allocation table of a sequential c-TA scheme and assume there is a group of c colluders. We construct a sequential tracing algorithm for Γ.

The algorithm starts by initializing the feedback sequence $F = (\cdot)$. The rebroadcasted content is considered in blocks of L_0 segments. Each block represents a pirate word in Γ_0. Using the tracing algorithm of Γ_0, the block is traced to one of the colluders and hence a symbol in Γ_1 alphabet associated with the colluder. This symbol is appended to F to form the feedback sequence of Γ_1. Now the tracing algorithm of Γ_1 is employed to trace colluders.

The correctness of the tracing algorithm follows from the correctness of the tracing algorithms of Γ_0 and Γ_1. □

Theorem 6 showed that a sequential TA code is a c-TA code. This means that the code Γ obtained above can be used as a c-TA code and be composed with a sequential c-TA scheme again. By repeating the composition n times, one can construct a q-ary (L, N) sequential c-TA scheme where $L = \prod_0^n L_i$.

A. Delayed Rebroadcast

Sequential tracing alleviates delayed rebroadcast attack. The mark allocation of each interval is determined solely by a column of the mark allocation table and does not depend on the feedback from the channel. The broadcaster follows the mark allocation table even if there is no rebroadcast. In the worst case, the colluders wait for the broadcast to be completed and then start the rebroadcast. This is the same as the traditional TA systems and since a sequential tracing scheme is a c-TA code then at least one of the colluders can be found. In fact, the first d_0 segments of the content can be used to identify a colluder. However, tracing more than one colluder requires the contribution of the traced one to be removed from the remaining content which is not possible in this case and so the system guarantees one colluder to be found. This is in contrast to the dynamic scheme that will become completely insecure against this attack.

B. Time–Bandwidth Tradeoff

Two important efficiency parameters of dynamic traitor tracing schemes are i) the number of marks q which determines the communication efficiency of the system, and ii) the convergence length L. Fiat and Tassa, and later Berkman *et al.* [6] concentrated on the communication efficiency and presented efficient algorithms when q is close to its theoretical minimum

$c+1$. Berkman *et al.* showed that if $q = c+a+1$, $1 \leq a \leq c$, it is possible to find traitors in $O(c \log N + c^2/a)$ steps. Schemes constructed in Section V (for example, the one obtained from Theorem 4) use $q = 2p$ versions and require $O(p \log N)$ steps to converge, where $p = Lc^2/(L-c)$.

In Theorem 9 we showed a way of trading convergence length with alphabet size. It is worth noting that the actual convergence time is the product of the length of a segment and the convergence length. As noted earlier, sequential tracing reduces the real-time computation and so can use shorter segments. However, the actual convergence time in these schemes might be lower in practice.

C. Conclusions

Sequential TA schemes fit between static and dynamic TA schemes. The application scenario in sequential schemes and dynamic schemes are the same and is different from static c-TA schemes. Also, the goal of the former two is the same (tracing all traitors) and is different from c-TA schemes in which the aim is to identify one colluder. Sequential schemes do not use the feedback from the channel to allocate marks and so in general would require higher bandwidth and a higher number of segments to converge. However, they provide security against delayed rebroadcast attack and reduce the real-time computation. The main construction of sequential TA schemes is by using error-correcting codes with large minimum distance. We showed that the mark allocation table in a sequential TA scheme gives a sequence of c-TA schemes. Determining necessary and sufficient conditions for c-TA schemes to be used as sequential c-TA schemes is an interesting open problem.

ACKNOWLEDGMENT

The authors would like to thank anonymous referees who provided useful comments.

REFERENCES

[1] A. Fiat and M. Naor, "Broadcast encryption," in *Advances in Cryptology—CRYPTO'93 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1994, vol. 773, pp. 480–491.

[2] A. Fiat and T. Tassa, "Dynamic traitor tracing," in *Advances in Cryptology—CRYPTO'99 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 1999, vol. 1666, pp. 354–371.

[3] ——, "Dynamic traitor tracing," *J. Cryptol.*, vol. 14, no. 3, pp. 211–223, 2001.

[4] R. Safavi-Naini and Y. Wang, "Sequential traitor tracing," in *Advances in Cryptology—CRYPTO 2000 (Lecture Notes in Computer Science)*. Berlin, Germany: Springer-Verlag, 2000, vol. 1880, pp. 316–332.

[5] I. Cox, J. Killian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673–1687, Dec. 1997.

[6] O. Berkman, M. Parnas, and J. Sgall, "Efficient dynamic traitor tracing," in *Proc. 11th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA 2000)*, 2000, pp. 586–595.

[7] ——, "Efficient dynamic traitor tracing," *SIAM J. Computing*, vol. 30, no. 6, pp. 1802–1828, 2001.

[8] M. A. Tsfasman and S. G. Vlăduţ, *Algebraic-Geometric Codes*. Norwell, MA: Kluwer, 1991.

[9] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1897–1905, Sept. 1998.

[10] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 1042–1049, Mar. 2001.

[11] J. H. van Lint, *Introduction to Coding Theory (Graduate Texts in Mathematics)*. New York: Springer-Verlag, 1999.

[12] R. Safavi-Naini and Y. Wang, "Collusion secure q-ary fingerprinting for perceptual content," in *Security and Privacy in Digital Rights Management (SPDRM 2001), Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 2002, vol. 2320, pp. 57–75.