Collusion-Secure Fingerpinting for Digital Data

Research Seminar in Cryptography

Cyber Security Master Programme

University of Tartu

Yauhen Yakimenka
Supervised by Vitaly Skachek

Abstract

This is the report based on the paper [BS98]. This report introduces fingerprinting for digital data and problem of collusion, and then presents results obtained in the paper. Some possible ideas for further research are discussed at the end.

Contents

1 Basic definitions and notation

An *alphabet* Σ of size s is just a set of s elements. We usually consider $\Sigma = \{0, 1, \ldots, s-1\}$.

For string $u \in \Sigma^l$ and the set of positions $I = \{i_1, i_2, \ldots, i_s\} \subset \{1, 2, \ldots, l\}$ we define[1]

$$u|_I = (u_{i_1}, u_{i_2}, \ldots, u_{i_s}).$$

The *Hamming distance* between two strings u and v of length l over the same alphabet Σ is the number of positions at which the corresponding symbols are different:

$$\mathsf{d}(u, v) = \big|\{i \mid i = 1, 2, \ldots, l,\ u_i \neq v_i\}\big|.$$

The *Hamming weight* of the string $u \in \Sigma^l$ is the hamming distance from the all-zero string of the same length:

$$\mathsf{w}(u) = \mathsf{d}(u,\ \underbrace{00\ldots0}_{l \text{ times}}).$$

Let Σ be an alphabet of size s. A set $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(n)}\}$, where every element $w^{(i)}$ (called codeword) is a string of length l over Σ, is called an (l, n)-*code*.

The (L, N)-code over an alphabet of p letters is said to be an $(L, N, D)_p$-*Error-Correcting Code* (or ECC, in short), if the Hamming distance between every pair of codewords is at least D.

Let η be a binomial random variable over k experiments with success probability $1/2$, i.e.

$$\begin{aligned} \mathsf{P}\{\eta = r\} &= \frac{1}{2^k}\binom{k}{r} \text{ for all } 0 \leq r \leq k, \\ \mathsf{E}\{\eta\} &= \frac{k}{2}. \end{aligned}$$

For any $a > 0$ the standard *Chernoff bound* states the following:

$$\mathsf{P}\left\{\eta \leq \frac{k}{2} - a\right\} \leq e^{-2a^2/k}.$$

2 Problem definition and variations

Illegal copying is a major problem in many areas. For digital material this is especially true, because copying such material is quite easy and no information is lost in the process. In addition, the growth of the Internet makes it possible to distribute the material in a much larger scale than before. And because of both technical and legal issues it is often difficult to find and prosecute the pirates.

Digital fingerprinting was introduced for the first time by Wagner [Wag83] in 1983. In digital fingerprinting the vendor embeds a secret unique mark in each copy of the digital object. This mark, the fingerprint, makes it possible to identify the buyers. Those guilty buyers who participated in illegal distribution will be occasionally called pirates.

In fact, the set of all these unique marks is a code.

But the next challenge is collusions of users. Since every user is given a slightly altered copy of the file, two or more users (the *coalition*) can simply

[1] We assume that $i_1 < i_2 < \ldots i_s$

identify different bits in their copies (e.g. by running some kind of `diff` utility). We will often equate the coalition and set of their codewords.

For further analysis the *Marking assumption* is introduced. This property claims that users cannot change the state of an undetected mark without rendering the object useless. It is assumed that marks satisfying this property exist for the objects being fingerprinted.[2]

When pirates from the coalition compare their copies they cannot define marking positions if they have the same marks in these positions. They just cannot distinguish them from all the other information in the file. Formally this could be defined in the following way.

Definition 1 ([BS98, Definition II.2]). *Let $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(n)}\}$ be an (l, n)-code and C be a coalition of users. For $i \in \{1, 2, \ldots, l\}$ we say that position i is **undetectable** for C if the words assigned to users in C match in their ith position.*

For example let us consider the following fragments of the files of two pirates:

		↓					↓			↓				
...	0	**1**	1	**0**	1	1	**0**	1	1	**1**	**0**	0	0	...
...	0	**0**	1	**0**	1	1	**1**	1	1	**0**	**0**	0	0	...

Bits in bold are embedded marks, i.e. the first pirate has the copy marked with 10010 while the second copy has the mark 00100. Note that pirates do not know the positions of marks. That's why they can only detect bits marked with arrows on the picture. When we say that position i is undetectable, i means number of position in the embedded fingerprint, not the entire file.

Next let us describe what the coalition is able to do. If the ith mark is detectable by coalitions C then the coalition can generate an object in which the ith mark is in any of its s states *or* is unreadable (wiped out). We denote wiped out mark by '?'.

Definition 2 ([BS98, Definition II.3]). *Let $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(n)}\}$ be an (l, n)-code and C be a coalition of users. Let R be the set of undetectable positions for C. Define the **feasible set** of C as*

$$F_\Gamma(C) = \left\{ w \in (\Sigma \cup \{?\})^l \;\middle|\; w|_R = w^{(u)}|_R \right\}$$

for some user u in C (no matter which particular user, since all of the users in C have the same characters in undetectable positions).

We sometime will omit Γ in subscript if it is obvious which code is considered.

So coalition ability is just producing codewords from its feasible set. The ability to make the mark unreadable is crucial here. As it will be shown later, it makes impossible to construct non-trivial *deterministic* secure fingerprinting schemes.

[2] In fact, this is not a simple task in real world. For instance, marking assumption for software is widely believed to be true, but exact universal algorithms have not been introduced yet.

3 Boneh and Shaw results

We now turn to the particular results of the paper [BS98]. The structure of the paper is sequential. The schemes are introduced starting with the simplest and then subsequent schemes are based on previous ones.

3.1 c-frameproof codes

To get familiarized with the domain, we will study problem which a bit distant from the rest of this report. We will discuss the following. One wants to build the code which will not allow any coalition *to frame* an innocent user. This means that coalition cannot generate any correct codeword except those already in coalition. This property will be relaxed by limiting the size of coalition to c users. Formally we define it as follows.

Definition 3. *A code Γ is c-**frameproof** if for every coalition C of size at most c it holds that $F(C) \cap \Gamma = W$, where W is the set of words of users from coalition C.*

Next the code $\Gamma_0(n)$ is defined over the alphabet $\Sigma = \{0, 1\}$.

Construction 1. *$\Gamma_0(n)$ is the (n, n)-code containing all the words of Hamming weight 1.*

For instance, $\Gamma_0(3) = \{100, 010, 001\}$.

Let us show that $\Gamma_0(n)$ is n-frameproof. Indeed, any coalition of size c detects exactly c bits in the positions they have 1's. Since any other user not in the coalition has 1 in another position (which remains undetectable for the coalition). Finally the coalition of n users contains all the users therefore they cannot frame anyone not in the coalition simply because there are no users left.

But $\Gamma_0(n)$ is rather useless in practice since its length is equal to the number of users. To construct more efficient code, we could compose it to any $(L, N, D)_p$-*Error-Correcting-Code*.

Now let us describe the construction of c-frameproof code.

Construction 2. *Let $\Gamma = \{w^{(1)}, w^{(2)}, \ldots, w^{(n)}\}$ be an (l, n)-code over Σ and let Υ be an $(L, N, D)_n$-ECC. Then composition code $\Gamma' = \Gamma \circ \Upsilon$ is defined as follows: for a codeword $v = v_1 v_2 \ldots v_L \in \Upsilon$ let*

$$W_v = w^{(v_1)} \| w^{(v_2)} \| \ldots \| w^{(v_L)} \in \Sigma^{lL}.$$

Then $\Gamma' = \{W_v \mid v \in \Upsilon\}$.

Let us prove that constructed code is indeed c-frameproof.

Lemma 1 ([BS98, Lemma III.2])**.** *If Construction 2 satisfies*

$$D > L\left(1 - \frac{1}{c}\right),$$

then Γ' is c-frameproof.

Proof. By composing codes Υ and Γ we map[3] codewords in Υ to the codewords in Γ'. We could consider coalition C either as $\{v^{(1)}, v^{(2)}, \ldots, v^{(c)}\} \subseteq \Upsilon$ or as $\{W_{v^{(1)}}, W_{v^{(2)}}, \ldots, W_{v^{(c)}}\} \subseteq \Gamma'$.

Let us suppose by contradiction that there is a codeword $z \in \Upsilon$ such that

$$W_z \notin \{W_{v^{(1)}}, W_{v^{(2)}}, \ldots, W_{v^{(c)}}\} \text{ but } W_z \in F_{\Gamma'}(C).$$

Since the minimum distance between each pair of codewords from Υ is D, we have

$$\mathbf{d}(z, v^{(k)}) \geq D > L\left(1 - \frac{1}{c}\right) \text{ for all } k = 1, \ldots, c.$$

This means that z and $v^{(k)}$ match in less than L/c positions. Hence, there exists a position $1 \leq j \leq L$ such that $z_j \neq v_j^{(k)}$ for all $k = 1, \ldots, c$.

Consider next the following coalition in Γ (which is in a certain sense projection of C on jth position):

$$C_j = \{w^{v_j^{(1)}}, w^{v_j^{(2)}}, \ldots, w^{v_j^{(c)}}\}.$$

Since Γ is a c-frameproof we know that $w^{z_j} \notin F_\Gamma(C_j)$. Then since w^{z_j} is a subword of W_z, this implies that $W_z \notin F_{\Gamma'}(C)$.

This contradiction proves the lemma. □

The question which remains open whether there exists required ECC. By picking the codewords randomly it is possible to obtain such a code. This is immediate from the Chernoff bound and we will state this in the following lemma.

Lemma 2 ([BS98, Lemma III.3])**.** *For any positive integers n and N let $L = 8n \log N$. Then there exists a $(L, N, D)_{2n} - ECC$ which has*

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

The main result on c-frameproof codes is the following.

Theorem 1 ([BS98, Theorem III.4])**.** *For any integers $n > 0$ and $c > 0$ let $l = 16c^2 \log_2 n$. Then there exists an (l, n)-code which is c-frameproof.*

Proof. From Lemma 2 we know that there exists $(L, n, L(1-1/c))_{2c}$-ECC where $L = 8c \log n$. Combining this with the code $\Gamma_0(2c)$ and Lemma 1 we get a c-frameproof code for n users whose length is $2cL = 16c^2 \log n$. □

However explicit construction of such a code is not so good due to a difficulty of explicit constructing of ECC.

3.2 c-secure codes

Now we turn our attention to the following task. Suppose that a coalition C generates the word $x \in F_\Gamma(C)$. When the object marked by x is found, the distributor would like to detect some (at least one) pirates from C. Hence, a totally c-secure code is the combination of a c-frameproof code and a tracing algorithm A.

[3] Mapping is in fact bijective.

Definition 4 ([BS98, Definition IV.1]). *A code Γ is **totally c-secure** if there exists a tracing algorithm A such that if a coalition C of at most c users generates a word x then $A(x) \in C$.*

Note that any code is totally 1-secure provided the Marking assumption is true. Indeed, one pirate cannot detect any of marks and therefore could only distribute his own copy.

But unfortunately the freedom of coalition to make marks unreadable leads to the fact that there are no c-secure codes for $c > 1$.

Theorem 2 ([BS98, Theorem IV.2]). *For $c \geq 2$ and $n \geq 3$ there are no totally c-secure (l, n)-codes.*

Proof. First let us show that there are no totally 2-secure codes. Let Γ be an arbitrary (l, n)-code. Let $w^{(1)}, w^{(2)}, w^{(3)}$ be three distinct codewords assigned to users u_1, u_2, u_3, respectively. Define the majority word M by

$$M = \begin{cases} w_i^{(1)}, & \text{if } w_i^{(1)} = w_i^{(2)} \text{ or } w_i^{(1)} = w_i^{(3)} \\ w_i^{(2)}, & \text{if } w_i^{(2)} = w_i^{(3)} \\ ?, & \text{if all } w_i^{(1)}, w_i^{(2)}, w_i^{(2)} \text{ are different.} \end{cases}$$

One can see that the word M could be produced by any of the coalitions $\{u_1, u_2\}$, $\{u_1, u_3\}$, $\{u_2, u_3\}$. But we cannot determine any user who is guilty with certainty. Hence, Γ is not 2-secure.

Next we should point out that if the code is not 2-secure, then it is also not c-secure for all $c > 2$. Indeed, we could take the coalition of 2 pirates which is able to produce untraceable word x and just add $c - 2$ users who are not traceable from the word x. Therefore this new coalition of c pirates produces untraceable word x and hence the code is not totally c-secure.

□

Thus we need to relax the requirement. And exploiting the *randomness* will help us.[4]

Definition 5 ([BS98, Definition IV.2]). *A fingerprint scheme Γ_r with a random string r is c-**secure with ϵ-error** if there exists a tracing algorithm A satisfying the following condition: for any coalition C of at most c users and any word $x \in F_\Gamma(C)$ we have*

$$\mathbf{P}\{A(x) \in C\} > 1 - \epsilon$$

where the probability is taken over the random bits r and the random choices made by the coalition.

With security of fingerprinting scheme defined as above we are able to build good schemes.

3.3 Replication scheme

This section will discuss the n-secure codes that are the building block (the inner code) for the logarithmic c-secure codes that will be discussed in the next

[4] The piece of randomness used below could also be viewed as a secret key needed for fingerprinting.

section. This n-secure code is called the *Boneh and Shaw Replication Scheme* (BS-RS).

Let y_m be a column of height n in which the first m bits are 1 and the rest are 0. Let us construct the following matrix (we describe it by enumerating its columns):

$$Y(n,d) = (\underbrace{y_1 y_1 \dots y_1}_{d \text{ times}} \underbrace{y_2 y_2 \dots y_2}_{d \text{ times}} \dots \underbrace{y_{n-1} y_{n-1} \dots y_{n-1}}_{d \text{ times}})$$

We define $\Gamma_0(n,d)$ as an $(n(d-1),n)$-code whose codewords are rows of the matrix $Y(n,d)$. The amount of duplication d determines the error probability ϵ. For example, $\Gamma_0(4,3)$ for users A,B,C,D is defined by

$$Y(4,3) = \begin{pmatrix} 111111111 \\ 000111111 \\ 000000111 \\ 000000000 \end{pmatrix} \begin{matrix} A \\ B \\ C \\ D \end{matrix}$$

Let $\Gamma_0(n,d) = \{w^{(1)}, w^{(2)}, \dots, w^{(n)}\}$. Before using this, the distributor applies to the columns of $Y(n,d)$ random permutation π, hence user u_i receives fingerprint $\pi(w^{(i)})$. The same permutation π is used for all the users and is kept secret from them[5].

Next let us introduce some notation.

B_m is the set of positions where columns y_m are mapped by π, $|B_m| = d$. In other words, if $\pi = (\pi_1, \pi_2, \dots, \pi_{d(n-1)})$, then

$$B_m = \{\pi_i \mid (m-1)d + 1 \le i \le md\}.$$

Note that[6] $\{1,2,\dots,d(n-1)\} = B_1 \sqcup B_2 \sqcup \dots \sqcup B_{n-1}$.

In fact, the permutation of columns of $Y(n,d)$ is defined only by partition of $\{1,2,\dots,d(n-1)\}$ into $B_1, B_2, \dots, B_{n-1}$ because of repetitive columns. Therefore there are only[7]

$$\binom{d(n-1)}{d,d,\dots,d} = \frac{(d(n-1))!}{(d!)^{n-1}}$$

really different permutations of $Y(n,d)$.

For $2 \le s \le n-1$ define $R_s = B_{s-1} \sqcup B_s$.

For instance, suppose for $\Gamma_0(4,3)$ we use the following permutation $\pi = (7,3,2,4,9,5,1,6,8)$. Then

$$\pi\left(Y(4,3)\right) = \begin{pmatrix} 111111111 \\ 100111011 \\ 100001010 \\ 000000000 \end{pmatrix}$$

$$B_1 = \{2,3,7\},\ B_2 = \{4,5,9\},\ B_3 = \{1,6,8\}.$$
$$R_2 = \{2,3,4,5,7,9\},\ R_3 = \{1,4,5,6,8,9\}.$$

[5] Therefore the scheme uses $\log(d(n-1))! \approx d(n-1)\log d(n-1)$ bits of randomness to represent permutation π.

[6] $\sqcup$ denotes union for pairwise nonoverlapping sets.

[7] The fact which is known from combinatorics

Before we turn to a strict proof, let us give some intuition. If we remove sth row from $\pi(Y(n,d))$, all the columns in positions R_s will become the same. Hence, for $i \in R_s$ one cannot tell if i lies in B_{s-1} or B_s. If user s is not in the coalition, the pirates cannot reconstruct more than $\pi(Y(n,d))$ with sth row removed and therefore they cannot tell if i lies in B_{s-1} or B_s. So whichever strategy the pirates use to produce a word x, the 1's in $x|_{R_s}$ will be evenly distributed between $x|_{B_{s-1}}$ and $x|_{B_s}$ (with high probability). Hence, if the 1's in $x|_{R_s}$ are not evenly distributed then, with high probability, user s is a member of the coalition that generated x.

Algorithm 1. *Given $x \in \{0,1,?\}^{d(n-1)}$, find a subset of the coalition that produced x.*

1. Set all ?-bits to 0.
2. If $\mathbf{w}(x|_{B_1}) > 0$ then output "User 1 is guilty".
3. If $\mathbf{w}(x|_{B_{n-1}}) < d$ then output "User n is guilty".
4. For $s = 2, 3, \ldots, n-1$ do: let $k = \mathbf{w}(x|_{R_s})$. If
$$\mathbf{w}(x|_{B_{s-1}}) < \frac{k}{2} - \sqrt{\frac{k}{2}\log\frac{2n}{\epsilon}}$$
then output "User s is guilty".

To prove the correctness of algorithm we will prove first the next two lemmas.

Lemma 3 ([BS98, Lemma V.2])**.** *Consider the code $\Gamma_0(n,d)$ with*
$$d = 2n^2 \log(2n/\epsilon).$$
Let x be the word produced by coalition C and S be the set of users which Algorithm 1 pronounces as guilty. Then
$$\mathbf{P}\{S \subseteq C\} \geq 1 - \epsilon.$$

Proof. Suppose $1 \in S$. This implies that $\mathbf{w}(x|_{B_1}) > 0$. If user 1 would not be a member of C, all the pirates would have 0's in B_1. Therefore the bits in B_1 would be undetectable for C and $\mathbf{w}(x|_{B_1}) = 0$. Contradiction shows that user 1 is indeed member of C.

With almost the same argument we have that if $n \in S$ then $n \in C$.

Now suppose that algorithm pronounces user $1 < s < n$ as guilty but user s is innocent, i.e. $s \notin C$. As it was shown above, the coalition cannot tell which bits in R_s belong to B_{s-1} and which belong to B_s. Since π was chosen uniformly at random, we could assume that coalition placed the 1's in $x|_{R_s}$ randomly.

Let $k = \mathbf{w}(x|_{R_s})$ and $\xi_k = \mathbf{w}(x|_{B_{s-1}})$ given that $\mathbf{w}(x|_{R_s}) = k$. For any $\max(0, k-d) \leq r \leq \min(d,k)$:
$$\mathbf{P}\{\xi_k = r\} = \frac{\binom{d}{r}\binom{d}{k-r}}{\binom{2d}{k}}.$$

From definition of ξ_k we know that expectation of ξ_k is $k/2$. Next, to estimate the probability that s was pronounced guilty we need to estimate

$$\mathbf{P}\left\{\xi_k < \frac{k}{2} - \sqrt{\frac{k}{2}\log\frac{2n}{\epsilon}}\right\}$$

given the distribution above.

Let η be a binomial random variable over k experiments with success probability $1/2$. One could check that for any r we have that $\mathbf{P}\{\xi_k = r\} \le 2\mathbf{P}\{\eta = r\}$. Then for any $a > 0$

$$\begin{aligned}\mathbf{P}\left\{\xi_k < \frac{k}{2} - a\right\} &\le 2\mathbf{P}\left\{\eta < \frac{k}{2} - a\right\} \\ &\le 2\mathbf{P}\left\{\eta \le \frac{k}{2} - a\right\} \le e^{-2a^2/k}\end{aligned}$$

where the last inequality follows from the standard Chernoff bound. Setting $a = \sqrt{(k/2)\log(2n/\epsilon)}$ leads to

$$\mathbf{P}\left\{\xi_k < \frac{k}{2} - \sqrt{\frac{k}{2}\log\frac{2n}{\epsilon}}\right\} \le 2e^{-\log(2n/\epsilon)} = \frac{\epsilon}{n}.$$

Hence if user s is innocent then the probability of him being pronounced guilty by Algorithm 1 is at most ϵ/n.[8] Therefore, the probability that some innocent user will be pronounced as guilty is not more than ϵ. This proves the lemma. □

Lemma 4 ([BS98, Claim V.4])**.** *Consider the code $\Gamma_0(n, d)$ with*

$$d = 2n^2\log(2n/\epsilon).$$

Let x be the word produced by coalition C and Algorithm 1 pronounces no users as guilty. Then for all s we have

$$\mathbf{w}(x|_{B_s}) \le 2s^2\log\frac{2n}{\epsilon}.$$

Proof. By induction on s.

For $s = 1$, this is trivial since $\mathbf{w}(x|_{B_1}) = 0$.

Now we assume the claim holds for $s < n - 1$ and prove it for $s + 1$. Define

$$\begin{aligned}k &= \mathbf{w}(x|_{B_s}) \\ k' &= \mathbf{w}(x|_{B_{s+1}}).\end{aligned}$$

Then inductive hypothesis is

$$k \le 2s^2\log\frac{2n}{\epsilon}.$$

Since $s + 1$ was not pronounced guilty and since $\mathbf{w}(x|_{R_{s+1}}) = k + k'$, we have

$$k \ge \frac{k + k'}{2} - \sqrt{\frac{k + k'}{2}\log\frac{2n}{\epsilon}}.$$

[8] Or, to be precise, $\frac{n-2}{n}\epsilon$.

From last two inequalities noting that $s \geq 1$ we have

$$k' \leq 2(s+1)^2 \log \frac{2n}{\epsilon}.$$

Therefore the lemma also holds for $s+1$ and by induction we have proved the lemma. □

Now we are ready to prove the correctness of Algorithm 1.

Theorem 3 ([BS98, Theorem V.1]). *For $n \geq 3$ and $\epsilon > 0$ let $d = 2n^2 \log_2(2n/\epsilon)$. The fingerprinting scheme $\Gamma_0(n, d)$ is n-secure with ϵ-error.*

Proof. From Lemma 3 we have that either Algorithm 1 is correct, or it pronounces no-one guilty.

If no-one was pronounced guilty, for user n we know that

$$\mathbf{w}(x|_{B_{n-1}}) = d = 2n^2 \log \frac{2n}{\epsilon}.$$

On the other hand, from Lemma 4 for $s = n-1$ we have

$$\mathbf{w}(x|_{B_{n-1}}) \leq 2(n-1)^2 \log \frac{2n}{\epsilon}.$$

This contradiction proves the theorem. □

The length of this code is $d(n-1) = O(n^3 \log_2(n/\epsilon))$.

3.4 Concatenated scheme

The idea here is to compose replication scheme from previous section with the Random Code (RC) scheme due to [Che96]. n-secure code is used as the alphabet over which RC can be applied. A particular RC which was used is kept secret. This is an addition to keeping hidden the L permutations used when embedding the L copies of $\Gamma_0(n, d)$ in the object. The resulting code is called the *Boneh and Shaw Concatenated Scheme* (BS-CS). The concatenation is achieved as described in section 3.1.

Theorem 4 ([BS98, Theorem V.5]). *Given an integers N, c, and $\epsilon > 0$ set $n = 2c, L = 2c \log(2N/\epsilon)$ and $d = 2n^2 \log(4nL/\epsilon)$. Then BS-CS is a code which is c-secure with ϵ-error. The code contains N words and has a length*

$$l = O(Ldn) = O\left(c^4 \log \frac{N}{\epsilon} \log \frac{1}{\epsilon}\right).$$

Proof. Proof of the theorem is based on the results of [Che96]. As for now we will only point out the tracing algorithm.

Algorithm 2. *Given $x \in \{0,1\}^l$, find a member of the guilty coalition that produced x.*

1. Apply Algorithm 1 to each of the L components of x. For each component $i = 1, \ldots, L$ arbitrarily choose one of the outputs of Algorithm 1. Set g_i to be this chosen output. Note that $g_i \in \{1, \ldots, n\}$. Next, form the word $g = g_1 \ldots g_L$.

2. Find the word w in the RC which was used such that Hamming distance between w and g is minimal.

3. Let u be the user whose codeword is derived from w. Output "User u is guilty".

□

3.5 Lower bound on the codes length

The theoretical lower bound on the length of c-secure codes is given in the following theorem.

Theorem 5 ([BS98] VI.1)**.** *Let Γ be an (l, n) fingerprinting scheme over a binary alphabet. Suppose Γ is c-secure with ϵ error. Then the code length is at least*

$$l \geq \frac{1}{2}(c-3)\log\left(\frac{1}{\epsilon c}\right).$$

Proof notes. In the proof authors suppose by contrary that $l < \frac{1}{2}(c-3)\log_2\left(\frac{1}{\epsilon c}\right)$ and show that every coalition of c pirates can produce untraceable codeword.

To do so, the coalition builds the codeword which is explicitly constructed in the paper. But the coalition succeeds in the word construction if the following occurs simultaneously: a) it correctly guessed $k_0 \in \{2, 3, \ldots, c-2\}$ (with probability $\frac{1}{c}$); b) it correctly set bits in some locations (with probability $c\epsilon$) . The probability of simultaneous occurrence of both events is not less than ϵ. □

Note also that the mentioned lower bound is not necessarily tight, i.e. it may happen that the codes with such a length do not exist. However, the difference between the lower bound:

$$l \geq \frac{1}{2}(c-3)\log\left(\frac{1}{\epsilon c}\right).$$

and the results from Composition Scheme:

$$l = O\left(c^4 \log\left(\frac{N}{\epsilon}\right)\log\left(\frac{1}{\epsilon}\right)\right)$$

is non-negligible therefore it seems that shorter n-secure schemes exist.

Schaathun in [Sch06] and then in a joint paper with Fernandez [SF06] came up with two schemes with good rates, where the codewords are significantly shorter than in the Boneh-Shaw scheme.

4 Further research ideas

4.1 Better results for small coalitions

The results in [BS98] are obtained for arbitrary c. On the other hand intuition suggests that for small values of c (for instance, $c = 2$ or $c = 3$) better results could be obtained.

Sebe and Domingo-Ferrer in [SDF02a] and [SDF02b] constructed 3-secure codes that (given a relatively small number of possible buyers) are much shorter than the general construction of Boneh and Shaw. This was improved by Schaathun in [Sch04].

4.2 Changed abilities of the collusive coalition

Abilities of the collusive coalition is a very important settings of the problem. Let us remind that in [CFNP00] coalition was not able to make the marks unreadable. That allowed authors to get much better results than in [BS98].

It would be interesting to study coalitions with different abilities to change the data. It seems that empowering is not really interesting as the coalition is powerful enough already. Alternatively, we can study coalitions with more restricted abilities. For instance, it could be a maximum number of changed marks or marks made unreadable.

Some computational capabilities limitations could be sound limitation too.

4.3 Detecting more pirates

The fingerprinting scheme suggested in [BS98] allows for (with high probability) exposure of only one pirate. However it could be useful to find more of them, ideally – all the pirates that actively participated in the generation of the codeword.

List decoding, introduced in [Sch06] as outer code decoding, facilitates the tracing of more than one pirate. This should be studied in greater detail.

4.4 Using less randomness

The Boneh-Shaw Concatenated Scheme uses a lot of randomness. It would be interesting to find if less random data usage will still allow to construct secure fingerprinting schemes (maybe for small coalitions).

References

[BS98] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *Information Theory, IEEE Transactions on*, 44(5):1897–1905, 1998.

[CFNP00] Benny Chor, Amos Fiat, Moni Naor, and Benny Pinkas. Tracing traitors. *Information Theory, IEEE Transactions on*, 46(3):893–910, 2000.

[Che96] Yeow Meng Chee. *Turán-type problems in group testing, coding theory and cryptography*. PhD thesis, Waterloo, Ont., Canada, Canada, 1996. AAINN15293.

[Sch04] Hans Georg Schaathun. Fighting three pirates with scattering codes. In *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, page 202. IEEE, 2004.

[Sch06] Hans Georg Schaathun. The Boneh-Shaw fingerprinting scheme is better than we thought. *Information Forensics and Security, IEEE Transactions on*, 1(2):248–255, 2006.

[SDF02a] Francesc Sebé and Josep Domingo-Ferrer. Scattering codes to implement short 3-secure fingerprinting for copyright protection. *Electronics Letters*, 38(17):958–959, 2002.

[SDF02b] Francesc Sebé and Josep Domingo-Ferrer. Short 3-secure fingerprinting codes for copyright protection. In *Information Security and Privacy*, pages 316–327. Springer, 2002.

[SF06] Hans Georg Schaathun and Marcel Fernandez. Soft decision decoding of Boneh-Shaw fingerprinting codes. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(10):2603–2608, 2006.

[Wag83] Neal R Wagner. Fingerprinting. In *Proceedings of the 1983 IEEE Symposium on Security and Privacy*, page 18. IEEE Computer Society, 1983.