On collusion-secure codes for copyright protection

Hans Georg Schaathun

Abstract

With a digital fingerprinting scheme a vendor of digital copies of copyrighted material marks each individual copy with a unique fingerprint. If an illegal copy appears, it can be traced back to one or more guilty pirates, due to this fingerprint. To work against a coalition of several pirates the fingerprinting scheme must be based on a collusion-secure code.

We make a new error analysis for the well-known collusion-secure code due to Boneh-Shaw, proving that it is much better than originally assumed. We also point out a problem of adverse selection to which schemes by Tardos, and by Le, Burmester, and Hu appears to be vulnerable.

1 Introduction

The problem of digital fingerprinting was introduced in [14], studied in [2], and given increasing attention following [3]. A vendor selling digital copies of copyrighted material wants to prevent illegal copying. Digital fingerprinting is supposed to make it possible to trace the guilty user (pirate) when an illegal copy is found. This is done by embedding a secret identification mark, called a fingerprint, in each copy, making every copy unique.

The fingerprint must be embedded in such a way that it does not disturb the information in the data file too much. It must also be impossible for the user to remove or damage the fingerprint, without damaging the information contents beyond any practical use. In particular, the fingerprint must survive any change of file format (e.g. gif to tiff) and any reasonable compression including lossy compression. This embedding problem is essentially the same as the problem known as watermarking in the literature.

If a single pirate distributes unauthorised copies, they will carry his fingerprint. If the vendor discovers the illegal copies he can trace them back to the pirate and prosecute him. If several pirates collude, they can to some extent tamper with the fingerprint. When they compare their copies they see some bits (or symbols) which differ and thus must be part of the fingerprint. Identified bits may be changed, and thus the pirates create a hybrid copy with a false fingerprint. In order to trace at least one pirate from such a collution, we use what is known as a collusion-secure code.

Collusion-secure codes are also employed in traitor tracing [5]. Whereas fingerprinting protects the digital data in themselves, traitor tracing protects broadcast encryption keys. Other important variants of the problems are dynamic traitor tracing (e.g. [9]) and anonymous fingerprinting [8].

A collusion-secure code can be probabilistic or combinatorially. In a probabilistic scheme, the vendor shall be able to trace a pirate with probability at least $1-\epsilon$ for some small error rate ϵ. Combinatorially collusion-secure codes allow successful tracing with probability 1.

Many schemes have been suggested over the past few years, with various pros and cons. In this paper we make a comparison of binary collusion-secure codes, both for some reasonable parameters, from a thousand to a billion users, and for the asymptotic case. The fingerprinting model is slightly refined, and we show that the error rates stated for existing schemes are not necessarily comparable.

We also make a new error-analysis to show that the Boneh and Shaw scheme from [3] is better than previously assumed. In particular the Boneh-Shaw scheme yields asymptotic classes of codes with positive rate and exponentially decreasing error rate, a property first proved for the BBK scheme [1]. We also introduce a couple of new schemes, in a sense variants of the Boneh-Shaw scheme, based on the new error analysis. We get particularly good improvements in the two-pirate case.

It should be mentioned that the design of collusion-secure codes is only a small part of the problem of fingerprinting. Another major problem is how actually to embed the fingerprint in the digital data; this problem has been studied in the field of *watermarking* on which there is an extensive literature. Little work exists on how to combine this into complete fingerprinting system. The present work shows that the length of the fingerprint can be reduced for a given number of users and pirates. Thus fewer symbols have to be embedded in the digital work to obtain the desired protection, which implies less distortion of the copyrigted work if or when a complete system is constructed.

2 The fingerprinting game

We use notation and terminology from coding theory. An $(n, M)_q$ code is a set of M n-tuples (words) over an alphabet of q symbols. We will refer to the set of fingerprints used for embedding as an $(n, M)_q$ code. Obviously, this provides for up to M buyers and requires n symbols to be embedded in the digital file. Each symbol has to take q different values.

A fingerprinting scheme consists of an (n, M) code C and a tracing algorithm A. Each codeword from C identifies a legitimate user, and is embedded as a fingerprint in the digital copies sold to this user. If several users collude to make illegal copies, they can make copies with some hybrid fingerprint $\mathbf{x}$ which combines information from their respective fingerprints. The algorithm A takes $\mathbf{x}$ as the input, and outputs a set $L \subseteq C$. If successful, the output is a non-empty subset of the pirates.

The fingerprinting scheme will usually take a randomising parameter, which we can call the key K. This is used to reduce the information available to the pirates; the key being known to the vendor and unknown by the pirates.

The game proceeds in the following steps.

1. The vendor chooses the fingerprinting scheme (C_K, A_K) to use for the product he is selling; this is the vendor strategy.

2. The key K is chosen at random.

3. The copies of the digital data are generated using the fingerprinting scheme and the key, and distributed to the users.

4. A coalition of potential pirates get together and compare their copies. At this stage they are allowed to opt out of the game, and refrain from illegal copying and distribution.

5. If the pirates choose to play, they choose a strategy for garbling the fingerprint, make the copies, and sell the copies with the false fingerprint.

6. If and when an illegal copy is discovered, the vendor runs the tracing algorithm A and prosecutes any users traced.

In accordance with Kerchoff's principles, all the information chosen in Step 1 is assumed to be public knowledge. The key chosen in Step 2 however is known only by the vendor.

If the pirates choose not to make illegal copies, no crime is committed and it makes sense to consider this as the normal or default outcome. In this case the game ends after Step 4.

A second outcome, which is usually neglected, and which probably do not have much impact on the design of fingerprinting scheme, is the situation where a crime is committed, but never revealed. This corresponds to the above game terminating before Step 6. We will not think any more of this outcome.

If the game continues until the end of Step 6, there are several possible outcomes. The tracing algorithm returns a set of users, which can be (1) only guilty pirates, (2) only innocent users, (3) some guilty and some innocent users, or (4) no user at all (void). If at least one pirate and no innocent user is returned, we say that the tracing algorithm is successful. If no guilty pirate is returned, we say we have an error of Type I, and if one or more innocent users are accused, then we have an error of Type II. Clearly, in case (3) above, we have both Type I and Type II errors.

Exactly what happens after Step 6 is outside the model. A criminal investigation is likely to provide further evidence of the crime, and a prosecution might fail even when a pirate has been traced, or succeed when an innocent user is accused. If innocent users are accused, we may hope that other investigational methods can clear them.

For the sake of the model and consistent with previous works, we will consider the vendor to be the winner in case (1) where no error occurs, and the pirates win if some error occurs. If the vendor wins, the pirates are penalised and pay compensation to cover the vendor's losses. The pirates are worse off than in the default case, and the vendor is at least as well off. If the pirates win, they get away with gains from the illegal sales, and the vendor is no better off than in the default case.

One of the most important parameters for the fingerprinting scheme is a bound on the error probability. Unfortunately, the error probabilities stated for various published schemes are not comparable. Viewed at the start of the game, before the key is drawn and before the fingerprints are distributed to the users, there is an *a priori error probability* that the pirates will be caught, assuming that they will never opt out of the game in Step 4.

When the pirates compare their copies in Step 4, they gain some information about their fingerprints. This information is very imperfect in most cases, but it can still result in an *(a posteriori) error probability* which is significantly different from the a priori probability.

It goes without saying that pirates who perceive a relatively high error probability after comparing their copies are more likely to go on with the crime, because they face a lower risk of being discovered and penalised. This is called *adverse selection*; the game is played only when the error probability is in favour of the pirates.

It is hard to argue that the pirates should not be allowed to opt out in a real setting, and thus adverse selection is a major problem for some proposed schemes where only an

a priori error probability is stated, i.e. [7, 13]. Still it may be possible to extend the error analysis for these schemes and prove that the probability of a pirate coalition seeing a dangerously low a posteriori error probability is negligible. This is a question for future research.

3 The marking assumption

The fingerprinting system must include some method to embed the fingerprints in the digital data, in addition to the fingerprinting code and tracing algorithm briefly described above. Some theoretical embeddings are suggested in [3]. We will base our collusion-secure codes on the following Marking Assumption. Alternative assumptions have been proposed, and some overview of this can be found in [1]. The present one says that the pirates can produce any fingerprint $(x_1, \ldots, x_n)$ if and only if for every position $i = 1 \ldots n$, at least one of the pirates see the symbol x_i in position i. Formally we define the feasible set $F_C(P)$ of false fingerprints which might be generated, as follows.

Definition 1 (The Marking Assumption)
Let $P \subseteq C$ be the set of fingerprints held by a coalition of pirates. The pirates can produce a copy with a false fingerprint $\mathbf{x}$ for any $\mathbf{x} \in F_C(P)$, where

$$F_C(P) = \{(c_1, \ldots, c_n) : \forall i, \exists (x_1, \ldots, x_n) \in P, x_i = c_i\}.$$

We call $F_C(P)$ the feasible set of P with respect to C.

There is an example of a simple and comprehensible embedding in the traitor tracing setting [5]. The system uses a $q \times n$ matrix of permanent keys $K_{j,i}$. Each row corresponds to an alphabet symbol and each column to a coordinate position. A fingerprint is an n-tuple $(a_1, \ldots, a_n)$ where each a_i is in the range $1, \ldots, q$. The user with fingerprint $(a_1, \ldots, a_n)$ receives the key $K_{a_i,i}$. The session key is the exclusive or of n elements s_1 to s_n. An enabling block is transmitted at the start of each session consisting of $e_{K_{j,i}}(s_i)$ for each i and j, where e_K is the encryption function for key K. To get the session key, one key from each column of the matrix is required, and that is exactly what each user has. When the pirates make a pirate decoder box, they must supply it with a key for each coordinate position from one of their true fingerprints, and thus the marking assumption is satisfied.

When the pirates opt to make false fingerprints, they choose a strategy S which will define a probability distribution on $F_C(P)$. However, since the strategy must be based on what the pirates actually can see, their choice is restricted. Most fingerprinting schemes use a secret permutation of the base code, meaning that when the pirates detect a column where they see more than one symbol, they cannot know where in the codewords it belong. Two columns $(x_1 \ldots x_t)$ and $(y_1 \ldots y_t)$ are indistinguishable if there is a permutation ϕ on the alphabet such that $y_i = \phi(x_i)$ for all i. A column $(0 \ldots 0)$ is of course not at all detectable.

The most general type of pirate strategies is a *fractional strategy*. For the presentation we assume $q = 2$ for simplicity. For each type $\mathbf{x}$ of indistinguishable columns, the pirates choose $f_\mathbf{x} \in [0, 1]$, and if $N_\mathbf{x}$ columns of this type exist, they choose at random $f_\mathbf{x} \cdot N_\mathbf{x}$ of the columns where they output the symbol seen by the first pirate. In the remaining columns of the type they output the opposite bit value.

It is customary in the literature to assume *column-independent strategies*. In this case a probability $p_\mathbf{x} \in [0, 1]$ is chosen for column type $\mathbf{x}$, and independently for each column

of the type the bit matching the first pirate is chosen with probability $p_{\mathbf{x}}$ and the opposite symbol is chosen with probability $1 - p_{\mathbf{x}}$. Clearly, by the law of large numbers, if the number of columns of each type is moderate, then the column-independent strategies are fair approximations to fractional strategies, and this is the case for the proposed schemes.

A fingerprinting scheme is a pair (C_K, A_K) where C_K is an (n, M) code and A_K is an algorithm taking a vector $\mathbf{x}$ of length n and outputting a subset $L \subseteq C_K$. If $\mathbf{x}$ is a false fingerprint produced by some coalition $P \subseteq C$, then A is successful if L is a non-empty subset of P. We have an error of Type I if $L \cap P = \emptyset$, and an error of Type II if $L \setminus P \neq \emptyset$. We say that (C_K, A_K) is *a priori* $(t, \epsilon_{\mathrm{I}}, \epsilon_{\mathrm{II}})$-secure if, when $\#P \leq t$, the a priori probabilities of errors of Type I or II are at most ϵ_{I} and ϵ_{II} respectively. The scheme is a priori (t, ϵ)-secure, if the total a priori error probability is at most ϵ when there are at most t pirates.

If the scheme is (*a posteriori*) $(t, \epsilon_{\mathrm{I}}, \epsilon_{\mathrm{II}})$-secure ($(t, \epsilon)$-secure), then for any pirate coalition of size t or less, ϵ_{I} and ϵ_{II} (ϵ) bound the error rates as perceived by the pirates after Step 4 for any pirate strategy they might choose. It is clear that if the scheme is a posteriori $(t, \epsilon_{\mathrm{I}}, \epsilon_{\mathrm{II}})$-secure, then it is also a priori $(t, \epsilon_{\mathrm{I}}, \epsilon_{\mathrm{II}})$-secure.

The distinction between a priori and a posteriori t-security has not previously been made in the literature as far as we know. The definition used has been either that of a priori security or ambiguous, but still most (though not all) of the schemes proposed are in fact a posteriori secure. This will be a key issue when we compare schemes in subsequent chapters.

In the literature we find two notable binary collusion-secure codes which are a posteriori (t, ϵ)-secure and constructible for arbitrary t. There is one due to Boneh and Shaw (which we will call 'RS-RC') [3] and one due to Barg, Blakley, and Khabatiansky (which we call 'BBK') [1]. RS-RC is the most well-known scheme. BBK forms an asymptotic family of codes with asymptotically declining error probability and non-zero rate. Furthermore, BBK can be decoded in $O(\log M)$ whereas RS-RC requires $O(M)$. In this paper we prove that also RS-RC has asymptotically vanishing error probability and non-zero rate. The rate of BBK is very good for small t, but RS-RC is better for larger t.

4 A little coding theory

The Hamming distance between two words $\mathbf{x}$ and $\mathbf{y}$ is denoted $d(\mathbf{x}, \mathbf{y})$, and the minimum distance of a code C is denoted $d(C)$ or just d. The normalised minimum distance is $\delta = d/n$. The code book C is a matrix where the rows are the codewords of C. The rate of the code is $R = (\log M)/n$.

Closest neighbour decoding is any algorithm which takes a word $\mathbf{x}$ and returns a word $\mathbf{c} \in C$ such that $d(\mathbf{c}, \mathbf{x})$ is minimised. This can always be performed in $O(M)$ operations, and for some codes it may be faster.

Concatenation is a standard technique from coding theory, and it has proven extremely useful in fingerprinting.

Definition 2 (Concatenation)
Let C_1 be a $(n_1, Q)_q$ and let C_2 be an $(n_2, M)_Q$ code. Then the concatenated code $C_1 \circ C_2$ is the $(n_1 n_2, M)_q$ code obtained by taking the words of C_2 and mapping every symbol on a word from C_1. Each set of n_1 symbols corresponding to one word of the inner code will be called a block.

Concatenated codes are often decoded by first decoding each block using some decoding algorithm for the inner code, so that a word of symbols from the outer code

alphabet is obtained. This word can finally be decoded with a decoding algorithm designed for the outer code.

For the error analysis, we will use the well known Chernoff bound as given in the following theorem. See e.g. [6] for a proof. The relative entropy function is defined as

$$D(\sigma||p) = \sigma \log \frac{\sigma}{p} + (1-\sigma) \log \frac{1-\sigma}{1-p}, \quad \text{for } \sigma, p \in (0,1). \tag{1}$$

Theorem 1 (Chernoff)
Let $X_1, \ldots, X_t$ be bounded, independent, and identically distributed stochastic variables in the range $[0,1]$. Let x be their (common) expected value. Then for any $0 < \delta < 1$, we have

$$P\left(\sum_{i=1}^{t} X_i \leq t\delta\right) \leq e^{-tD(\delta||x)}, \quad \text{when } \delta < x.$$

We write $\mathcal{B}(n,p)$ for the binomial distribution with n trials with probability p. If X is distributed as $\mathcal{B}(n,p)$, we write $X \sim \mathcal{B}(n,p)$.

Another useful concept for collusion-secure codes is separating codes. Such codes have been applied in various fields for more than three decades, see [10] for a survey.

Definition 3
A (t,u)-separating code or (t,u)-SS has the property for any two disjoint sets T and U of respectively t and u codewords, there is at least one coordinate position where every codeword of T is di*ff*erent from any codeword of U.

It can be shown that $(t,1)$-separating codes are frameproof, in the sense that it makes it impossible for a coalition of size t to generate a fingerprint identical to that of an innocent user.

5 Concatenated schemes

In this chapter we develop a general analysis of concatenation of collusion-secure codes. Decoding is done by decoding each block with a tracing algorithm for the inner code, in order to obtain a word of symbols from the outer code alphabet. If inner decoding is always successful, then this returns a word in the feasible set of the pirates, viewed as a subset of the outer code.

When Boneh and Shaw used this technique, they chose the parameters such that inner decoding succeeds in every position with probability $1-\epsilon/2$, and and such that outer decoding, given perfect inner decoding, succeeds with probability $1-\epsilon/2$. Thus the total error probability was less than ϵ. Demanding that inner decoding be correct in every position is a strong requirement, because its probability declines exponentially in the code length. An idea pointed out in [1] is that a bounded fraction of failures from inner decoding can be corrected by making the outer code slightly more powerful. This idea works for the Boneh and Shaw scheme as well, and we will see that their scheme is far better than they proved.

We suggest to decode the outer code with list decoding. Apart from the obvious advantage of allowing us to trace more than one pirate in many cases, it also makes the error analysis simpler, and it becomes clear how to adapt the error analysis for other choices for inner and outer codes in the scheme. Even though an error analysis for closest neighbour decoding can be made, it is not certain to give better error bounds.

List decoding of concatenated codes

Let C_I be an (n_1, q) inner code which is $(t, \epsilon_{\text{in}})$-secure, and C_O an $(n_2, M)_q$ outer code. Let R_I and R_O denote the rates of C_I and C_O respectively.

Our decoding algorithm works as follows. Let P be a pirate coalition of size at most t, and $\mathbf{x} \in F_C(P)$. First each block is decoded with respect to the inner code, to produce a q-ary vector $\mathbf{y}$ of length n_2. The algorithm returns the set L of codewords $\mathbf{c} \in C_O$ at a distance $d(\mathbf{c}, \mathbf{y}) \leq D$, for some decoding threshold D.

Let F be the number of positions where inner decoding is incorrect. Clearly, $F \sim \mathcal{B}(n, \epsilon_{\text{in}})$. The pirates match $\mathbf{y}$ in at least $(n - F)/t$ positions on average, which means that if $F \leq tD - (t-1)n_2$, then at least one guilty pirate is caught. The following theorem follows by the Chernoff bound.

Theorem 2
Using a concatenated code of an (n_1, q) t-secure inner code with ϵ_{in}-error, and an (n_2, M) outer code, with outer list decoding with threshold $D = n_2\Delta$, the probability of identifying no guilty user is

$$\epsilon_{\text{I}} \leq P(F \geq (1 - t + t\Delta)n_O), \quad F \sim \mathcal{B}(n_O, \epsilon_{\text{in}}),$$

and

$$\epsilon_{\text{I}} \leq 2^{-n_O D(1-t+t\Delta||\epsilon_{\text{in}})}, \quad \text{if} \quad \epsilon_{\text{in}} < 1 - t + t\Delta.$$

Corollary 1
If $D(1 - t + t\Delta||\epsilon_{\text{in}}) > 0$, then the probability of Type I error tends exponentially to zero with increasing code length n_2.

Note that the bound on ϵ_{I} is valid for any codes, and it depends only on n_2, Δ, t, and ϵ_{in}. The Type II error rate ϵ_{II} will depend on the design of the outer code.

Random codes (RC)

Boneh and Shaw used random codes, for which Chee [4] was credited. Let C_O be a $(n_2, M)_q$ code, where each symbol in each codeword is chosen uniformly at random from the alphabet. The entire code is kept secret by the vendor.

Theorem 3
If a random code is used as outer code for concatenation and $1/q < 1 - \Delta$, the probability of including a given innocent user $\mathbf{c}$ in the output list is bounded as

$$P(\mathbf{c} \in L) \leq \hat{\epsilon} \leq 2^{-n_2 D(1-\Delta||1/q)},$$

and the total Type II error rate is bounded as

$$\epsilon_{\text{II}} \leq 2^{n_2(R_O \log q - D(1-\Delta||1/q))}.$$

Proof: Consider the output $\mathbf{y}$ from inner decoding and an innocent user $\mathbf{c} \notin P$. Let $X = n_2 - d(\mathbf{c}, \mathbf{y})$. Clearly X is a stochastic variable with distribution $B(n_2, 1/q)$, and $P(\mathbf{c} \in L) = P(X \geq n_2 - D)$. The error probability is bounded as

$$\epsilon_{\text{II}} \leq \sum_{\mathbf{c} \in C \setminus P} P(\mathbf{c} \in L) \leq M \cdot P(X \geq n_2(1 - \Delta)),$$

and the theorem follows by Chernoff's bound. □

Corollary 2
The Type II error rate tends exponentially to zero with increasing length if $R_O < D(1-\Delta||1/q)/\log q$.

One great advantage of random codes is that they can be made for any number of users quite trivially. Observing the error bounds, we note that ϵ_{I} is unaltered, and ϵ_{II} degrades gracefully when M increases.

Replication scheme with random codes

The following construction was introduced by Boneh and Shaw to serve as inner code. We will call it the Boneh-Shaw replication scheme (BS-RS).

BS-RS uses a binary $(r(M-1), M)$ code which is M-secure with ϵ-error. The code book has $M-1$ distinct columns replicated r times. A set of identical columns will be called a type. Every column has the form $(1\ldots10\ldots0)$, such that the i-th $(1 \le i \le M)$ user has zeroes in the first $i-1$ types and a one in the rest. We can see that unless user i is a pirate, the pirates cannot distinguish between the $(i-1)$-th and the i-th type. Hence they have to use the same probability of choosing a 1 in both these types. If r is large enough we can use statistics to test the null hypothesis that user i be innocent. The output is a list of users for which the null hypothesis may be rejected.

We have

$$\hat{\epsilon} \le 2^{1-\frac{r}{2M^2}}.$$

Theorem 4 (Boneh and Shaw)
The BS-RS with replication factor r is (M, ϵ)-secure whenever $r \ge 2M^2 \log(2M/\epsilon)$.

Suppose we use an (n_1, q) BS-RS as an inner code. This scheme has several control parameters which may be used to tune the performance of the system. The inner code cardinality q is the trickiest one. Most of the time we will follow Boneh and Shaw and set $q = 2t$. Obviously n_2 and r control a trade-off between code length and error rate. Finally, we have Δ to control the trade-off between the two error types.

Theorem 5
If we use $q = 2t$, $\Delta = t/(t+1)$, and $\epsilon_{\mathrm{in}} = 1/2t$, then RS-RC is an (t, ϵ)-secure fingerprinting scheme accommodating M users requiring length

$$n = (2t-1)\left\lceil 8t^2(3+2\log t)\right\rceil n_2,$$

where

$$n_2 = \frac{\max\{-\log\epsilon_{\mathrm{I}}, \log M - \log\epsilon_{\mathrm{II}}\}}{D(\frac{1}{t+1}||\frac{1}{2t})}.$$

Asymptotically, the length is

$$n = \Theta\left(t^4(\log t)(\log M - \log\epsilon)\right).$$

In this theorem, Δ is made only slightly greater than the minimum value of $(t-1)/t$. By Corollary 1 we require $\epsilon_{\mathrm{in}} < 1/(t+1)$, but to make n_2 linear in t, ϵ_{in} must in fact be much smaller than $1/(t+1)$.

$t = \log M$	Boneh and Shaw	New analysis
10	$6.64 \cdot 10^8$	$3.14 \cdot 10^8$
15	$3.91 \cdot 10^9$	$1.82 \cdot 10^9$
20	$1.40 \cdot 10^{10}$	$6.56 \cdot 10^9$
25	$3.80 \cdot 10^{10}$	$1.80 \cdot 10^{10}$
30	$8.68 \cdot 10^{10}$	$4.15 \cdot 10^{10}$

Table 1: Some lengths when $t = \log M$.

Proof: Theorems 2 and 3 give two bounds on n_2, so we get

$$n_2 = \max\left\{ \frac{-\log \epsilon_{\mathrm{I}}}{D(\frac{1}{t+1}||\frac{1}{2t})}, \frac{\log M - \log \epsilon_{\mathrm{II}}}{D(\frac{1}{t+1}||\frac{1}{2t})} \right\}.$$

It can be shown that $D(1/(t+1)||1/(2t)) = \Theta(t^{-1})$, and hence

$$n_2 = \Theta(t(\log M - \log \epsilon)).$$

For the inner code, we have

$$n_1 = (q-1)2q^2(\log(2q) - \log \epsilon_{\mathrm{in}}) = (2t-1)8t^2(3 + 2\log t) = \Theta(t^3 \log t).$$

The theorem follows since $n = n_1 n_2$. □

For comparison, we include the original theorem from [3].

Theorem 6 (Boneh and Shaw)
BS-RS with replication factor r and $q = 2t$ users for the inner code, is a t-secure (n, M) code with ϵ-error, where

$$n_2 = \left\lceil 2t \log \frac{2M}{\epsilon} \right\rceil, \quad r = \left\lceil 8t^2 \log \frac{8tn_2}{\epsilon} \right\rceil,$$

$$n = n_2 r(2t-1) \approx 16t^3(2t-1)\left(\log \frac{2M}{\epsilon} \right)\left(\log \frac{8tn_2}{\epsilon} \right).$$

The decoding complexity was $\Theta(n + M)$.

The most interesting point in the original theorem is that $r = \Theta(\log n_2)$, such that n grows faster than linearly in n_2. Since n_2 depends on M and on ϵ, the length was much more dependent on ϵ and M than is with our analysis. In Table 1 we see some real sample lengths for these codes, with our and Boneh and Shaw's formulæ. The new analysis appears to make an improvement by a factor of 2.1 (as pointed out by an anonymous referee). It might be possible analytically to prove such an improvement factor in general, but so far we have not looked into that.

Considering asymptotic classes of codes, Δ can be made smaller. The following theorem gives the better rates.

Theorem 7
There exists an asymptotic class of fingerprinting codes with exponentially declining error rate for any rate R satisfying

$$R < \frac{D(\frac{1-2q2^{-r/(2q^2)}}{t}||1/q)}{r(q-1)}, \tag{2}$$

if q and r are natural numbers such that $(1 - 2q2^{-r/(2q^2)})/t > 1/q$.

	RS-RC			BBK	
t	q	r	Rate	C_I	Rate
2	4	238	$2.42 \cdot 10^{-4}$	$(126, 2^{14})$	0.0172
3	5	410	$3.62 \cdot 10^{-5}$	$(2046, 2^{7})$	$3.98 \cdot 10^{-4}$
4	7	847	$9.62 \cdot 10^{-6}$	$(32766, 2^{10})$	$1.82 \cdot 10^{-5}$
5	9	1457	$3.53 \cdot 10^{-6}$	$(1048572, 2^{12})$	$4.36 \cdot 10^{-6}$
7	13	3223	$8.04 \cdot 10^{-7}$	$(10^{28} - 1, 2^{12})$	$0.116 \cdot 10^{-8}$

Table 2: Asymptotic rates and maximising values of q and r for the RS-RC codes for some numbers of pirates.

Proof: Asymptotically, ϵ_{in} can be taken arbitrarily close to $1 - t + t\Delta$, or in other words

$$\Delta \approx \frac{t - 1 + \epsilon_{\text{in}}}{t} = \frac{t - 1 + 2q2^{-r/2q^2}}{t}.$$

By Theorem 3, the outer rate can be chosen arbitrarily close to $D(1 - \Delta || 1/q)/\log q$. We get the following component code rates

$$R_O \approx \frac{D(\frac{1-2q2^{-r/2q^2}}{t} || 1/q)}{\log q}, \quad R_I = \frac{\log q}{r(q-1)},$$

which gives the total rate as stated in the theorem. □

In Table 2, we can see some asymptotic rates for our codes. The BBK codes given are the best we could find using constructible inner codes from the literature, namely duals of BCH codes [12]. We can see that BBK is better for few pirates, but for larger t we could not find (t, t)-separating codes which are good enough. It is also interesting to note that $2t$ is not the maximising value of q asymptotically, except for $t = 2$.

Outer code with large distance

We recall that codes with sufficiently large distance give combinatorially secure codes. The BBK scheme introduced outer codes where the minimum distance is large enough not only to successfully trace, but also to correct for some decoding errors from the inner decoding. We present an error analysis for such codes, following the lines from the previous section, and show how it can be combined with $(t, \epsilon_{\text{in}})$-secure inner codes. The BBK code used (t, t)-separating inner codes.

Let C_I be an inner code t-secure with ϵ_{in}-error. Let $\hat{\epsilon}_{\text{in}}$ be an upper bound on the probability of accusing any given innocent user $\mathbf{c}$. Even though this is a parameter traditionally never explicitly stated for constructed fingerprinting schemes, it is often known by a bound at least as good as that for ϵ_{in}, which is often bounded as $\epsilon_{\text{in}} \leq M\hat{\epsilon}_{\text{in}}$.

Let C_O be the outer code with minimum distance δn, and $P = \{\mathbf{a}_1, \ldots, \mathbf{a}_t\} \subseteq C_O$ a pirate coalition. Consider a false fingerprint $\mathbf{x}$ after inner decoding and an arbitrary innocent user $\mathbf{c} \notin P$. For each i, $\mathbf{c}$ matches $\mathbf{a}_i$ in at most $n(1 - \delta)$ positions. If inner decoding were perfect, $\mathbf{x}$ would match $\mathbf{c}$ in at most $nt(1 - \delta)$ positions.

The outer code is decoded by list decoding with threshold Δ. First we study the probability $\pi(\mathbf{c})$ that an innocent user $\mathbf{c}$ be accused. Let S be the set of coordinates where $\mathbf{c}$ is different from any pirate, and let S^C be the complement, i.e. the set of positions

where **c** match at least one pirate. Let X_i be a stochastic variable which is one if and only if $c_i = x_i$. We get that

$$s(\mathbf{c}, \mathbf{x}) = \sum_{i \in S} X_i + \sum_{i \in S^C} X_i \leq \sum_{i \in S} X_i + \#S^C. \tag{3}$$

We have $\#S^C \leq nt(1-\delta)$. If we let $S' \subseteq S$ be any subset of size $n(1-t(1-\delta))$, we get

$$s(\mathbf{c}, \mathbf{x}) \leq X + nt(1-\delta)), \quad \text{where } X = \sum_{i \in S'} X_i. \tag{4}$$

We have that X_i is 1 with probability $\hat{\epsilon}_{\text{in}}$ and 0 otherwise. We get

$$\epsilon_1 \leq P(s(\mathbf{c}, \mathbf{x}) > (1-\Delta)n) \leq P(X > ((1-\Delta) - t(1-\delta))n). \tag{5}$$

Using Chernoff, we get the following theorem.

Theorem 8
Using outer codes with normalised minimum distance δ, inner code with probability $\hat{\epsilon}_{\text{in}}$ of accusing a given innocent user, and list decoding with threshold Δ, we get the following Type II error probability:

$$\hat{\epsilon} \leq 2^{-nD(\sigma || \hat{\epsilon}_{\text{in}})}, \quad \text{where} \quad \sigma = (1-\Delta) - t(1-\delta). \tag{6}$$

Combining Theorems 2 and 8, we get that

$$\delta > 1 - \frac{1 - \epsilon_{\text{in}} - t\hat{\epsilon}_{\text{in}}}{t^2}. \tag{7}$$

It follows immediately that $q > t^2$, but exactly how much larger q needs to be is less clear. A good candidate as an outer code with large minimum distance is the $[n_O, k_O, n_O - k_O + 1]_q$ Reed-Solomon (RS) codes. For asymptotic classes of codes, algebraic geometry (AG) codes can be used. Both RS and AG codes can be decoded with the Guruswami-Sudan algorithm, with complexity $O(n_O)$.

The rates obtained with BS-RS as inner codes and RS or AG outer codes are not so good. Medium sized inner codes are needed, and BS-RS have poor rate except for very small size. In the two-pirate case [11] simplex codes is a better alternative, giving an asymptotic rate of 0.062. The following proposition state the general result with AG outer codes. More research is required to make it useful for $t > 2$.

Proposition 1
If there is an (n_I, q) $(t, \epsilon_{\text{in}})$-secure code where the probability of accusing a given innocent user is at most $\hat{\epsilon}_{\text{in}}$, then there is an asymptotic family of (t, ϵ)-secure codes with rate $R_O(\log q)/n_I$, where R_O solves

$$R_O \log q = D\Big(\frac{1 - \epsilon_{\text{in}}}{t} - t\Big(R_O + \frac{1}{\sqrt{q} - 1} \Big) \;||\; \hat{\epsilon}_{\text{in}} \Big),$$

and where ϵ vanishes exponentially.

The proof is very similar to that of Theorem 7, so it is ommitted.

q	r	Outer code	Δ	n	$\log M$	ϵ
49	53 500	$[49,2]$	0.785	125 832 000	11.2	$0.653 \cdot 10^{-10}$
49	62 690	$[49,3]$	0.746	147 446 880	16.8	$0.987 \cdot 10^{-10}$
49	78 690	$[49,4]$	0.71	185 078 880	22.5	$0.977 \cdot 10^{-10}$
49	119 000	$[49,5]$	0.685	279 888 000	28.1	$0.806 \cdot 10^{-10}$
64	130 000	$[64,5]$	0.715	524 160 000	30	$0.959 \cdot 10^{-10}$

Table 3: Some RS-RS codes against three pirates.

References

[1] A. Barg, G. R. Blakley, and G. A. Kabatiansky. Digital fingerprinting codes: Problem statements, constructions, identification of traitors. *IEEE Trans. Inform. Theory*, 49(4):852–865, April 2003.

[2] G. R. Blakley, C. Meadows, and G. B. Purdy. Fingerprinting long forgiving messages. In *Advances in cryptology—CRYPTO '85 (Santa Barbara, Calif., 1985)*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 180–189. Springer, Berlin, 1986.

[3] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Trans. Inform. Theory*, 44(5):1897–1905, 1998. Presented in part at CRYPTO'95.

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

[5] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inform. Theory*, 46(3):893–910, May 2000. Presented in part at CRYPTO'94.

[6] Torben Hagerup and Christine Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1990.

[7] Tri Van Le, Mike Burmester, and Jiangyi Hu. Short c-secure fingerprinting codes. In *Proceedings of the 6th Information Security Conference*, October 2003. Available at `http://websrv.cs.fsu.edu/~burmeste/`.

[8] Birgit Pfitzmann and Michael Waidner. Anonymous fingerprinting. In *Advances in cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 88–102. Springer, Berlin, 1997.

[9] Reihaneh Safavi-Naini and Yejing Wang. Sequential traitor tracing. In *Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA)*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 316–332. Springer, Berlin, 2000.

[10] Yu. L. Sagalovich. Separating systems. *Problems of Information Transmission*, 30(2):105–123, 1994.

[11] Hans Georg Schaathun. Binary collusion-secure codes: Comparison and improvements. Technical Report 275, Department of Informatics, University of Bergen, 2004. Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`.

[12] Hans Georg Schaathun and Tor Helleseth. Separating and intersecting properties of BCH and Kasami codes. In *Cryptography and Coding*, volume 2898 of *Springer Lecture Notes in Computer Science*. Springer-Verlag, December 2003. 9th IMA International Conference.

[13] G. Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003. `http://www.renyi.hu/~tardos/fingerprint.ps`.

[14] Neal R. Wagner. Fingerprinting. In *Proceedings of the 1983 Symposium on Security and Privacy*, 1983.