On watermarking/fingerprinting for copyright protection

Hans Georg Schaathun

Abstract

Collusion-secure codes are used for digital fingerprinting and for traitor tracing. In both cases, the goal is to prevent unauthorised copying of copyrighted material, by tracing at least one guilty user when illegal copies appear. Many recent works have introduced collusion-secure codes based on the Marking Assumption. In this paper we study how error-correction makes it possible to relax this assumption, and we modify two existing schemes to enable error-correction.

Keywords

collusion-secure fingerprinting, copyright protection, traitor tracing, soft-decision decoding

1 Introduction

Unauthorised copying and distribution of copyrighted material has received increasing attention over many years, both in research communities and in the daily press. Authors and artists depend on their income from legal sales, and unauthorised copying is often seen as a threat to these sales. For the movie or music industry, this is a question of big money.

The American *International Intellectual Property Alliance* [1] claim that losses from piracy of U.S. copyrighted material amounts to 25-30 billion US$ annually, excluding internet piracy. Even though such estimates are often disputed, there is no doubt that big money is at stake, and the issue receives tremendous interest. Several countries, including Norway, are in the process of changing their legislation to deal more effectively with illegal distribution in new media.

There are several technological approaches to battling copyright piracy. Digital Rights Management (DRM) encompass different techniques to prevent copying or restrict use of a digital file. Such technology is controversial because it also restricts normal use which is traditionally legal. So-called forensic techniques do not prevent copying, instead, when unauthorised copies appear, they enable the copyright holder to trace the pirates and prosecute. Since forensic techniques only come into play when a crime is evident, it is less controversial than DRM. Still, no perfect or generally accepted solution exists yet, giving ample room for new research.

Digital fingerprinting was suggested as a forensic technique in [14], and following [3, 4] this problem has received increasing interest. Each user is identified by a «fingerprint», which is embedded in the file in such a way that the user cannot remove it. If an unauthorised copy appear, the embedded fingerprint reveals the identity of the guilty party. Of course the pirate(s) will do what they can to remove or damage the fingerprint, and making the fingerprinting system robust to any conceivable attack is a challenging task.

A large fraction of fingerprint literature has focused on making collusion-secure codes for fingerprinting. The goal is to counter attacks where a coalition of pirates get together and cut-and-paste from their individual copies in order to make an illegal copy with a hybrid fingerprint. Most authors have assumed that an underlying watermarking system is used to embed the fingerprints. The most well-known fingerprinting model [4] make rather strong assumptions about the watermarking system.

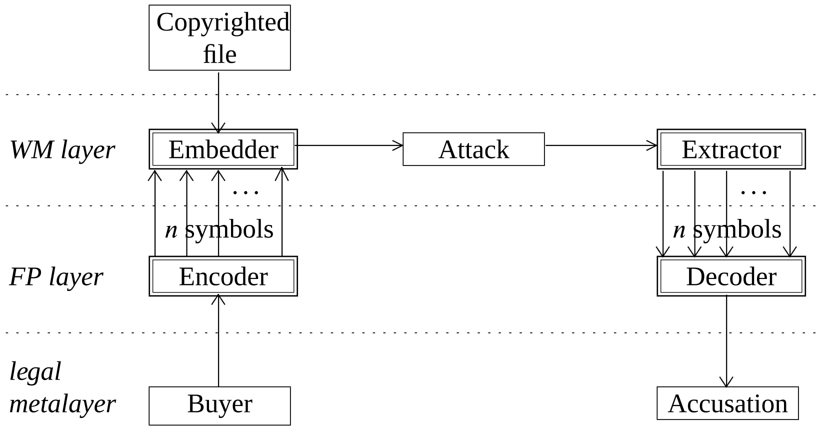

Figure 1: Watermarking/Fingerprinting model.

Different alternative models exist. Guth and Pfitzmann [7] among others have argued that a more realistic model should correct random errors as well as being collusion-secure. In Section 2, we present the layered fingerprinting model of Guth and Pfitzmann, discuss possible attacks, and add a few details. In Section 3, we show how we can make efficient codes for this model by combining the theory of error-correcting codes with some fundamental ideas from [4]. A few conclusions will be made in the last section.

2 The layered fingerprint/watermark model

Most works on fingerprinting assume that the fingerprint is a string $(c_1, \ldots, c_n)$ of n symbols. The file is assumed to be divided into n segments, and one symbol is embedded in each segment using a watermarking scheme.

Watermarking, briefly defined, is a technique to embed a message in a digital file in such a way that an adversary is unable to remove or change this message. Neither the existence nor the contents of the message is assumed to be secret, and thus watermarking is different from steganography.

We first present a two-layer model for fingerprinting before we discuss respectively fingerprinting and watermarking attacks on the system. Towards the end of the section, we will discuss a three-layer model.

The two-layer model

Guth and Pfitzman pointed out how the standard fingerprinting models fitted into a layered structure, with one watermarking layer and one fingerprinting layer. The layout is depicted in Figure 1.

The file is divided into n segments which are fed directly and independently to *the watermarking layer*. The *embedding algorithm* embeds a message from the set Q in the segment, in such a way that an adversary cannot change or remove it. The watermarked segment is called a *mark*. The *extraction algorithm* takes a mark and returns a message from Q.

In *the fingerprinting layer*, each user is identified by an element from some (n, M) code C over Q, that is a subset $C \subseteq Q^n$. If Q has q elements, we say that C is an $(n, M)_q$ code. The code C has M elements, so M users can be catered for. When a copy is sold, the buyer is assigned a fingerprint $\mathbf{c} \in C$. Each element of $\mathbf{c}$ is fed to the watermarking layer to be embedded in the digital file. Observe that the copyrighted file is not used in the fingerprinting layer at all.

When an illegal copy is found, the file is split into n marks which are fed to the *extraction algorithm* in the watermarking layer. The n outputs give a *false fingerprint* $\mathbf{x} = (x_1, \ldots, x_n) \in Q^n$ which is fed to the fingerprinting layer. The *fingerprinting decoder* takes $\mathbf{x}$ and outputs $L \subseteq C$ identifying a number of users expected to be guilty.

If nothing foul happens, any fingerprinted copy can be traced back to a user. In fact $\mathbf{x} \in C$ and corresponds to the buyer of this copy. A pirate however, does not want to be identified, so he will make attack, either on the watermarking or on the fingerprinting scheme, in order to cause the decoding to fail. Possible attacks are studied in subsequent subsections.

The mapping from coordinate positions of C onto segments is assumed to be uniformly random and secret. In other words, the pirates have no information about which coordinate position $i = \{1, \ldots, n\}$ of C is embedded in a given mark.

We have assumed that the watermark extractor always returns a single element of Q. This is a simplification. Many real systems will also be able to return 'erasure' when no one element appears as more likely than others. In case of an erasure, we return a random element of Q, and treat it as an error.

Attacks in the fingerprinting layer

Pirates can mount attacks against either or both layers. The goal is always to get an illegal copy which cannot be traced back to them, i.e. where the output L of the fingerprinting decoder does not contain any of the pirates.

When a group of t pirates collude, they have access to t distinct copies with different fingerprints/watermarks. Comparing the copies, they will see some segments which are different (called detectable marks) and some which are identical (called undetectable marks).

In the fingerprinting layer, there is one known attack available, namely the *Cut-and-paste attack*, where the pirates take some segments from each of their copies and paste them together. The result is a hybrid fingerprint where each symbol matches at least one of the pirate copies.

Many traditional works on fingerprinting considered only cut-and-paste attacks. They assumed that the output x_i from the watermarking layer would always match the i-th symbol of at least one of the pirates. The classic phrasing of this definition is as follows [4].

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.

A code C is said to be (t, ϵ)-secure under the Marking Assumption, if, when there are at most t pirates, the output L of the fingerprinting decoder is a non-empty subsets of the pirates with probability at least $1 - \epsilon$.

The most well-known solution under the Marking Assumption, is due to Boneh and Shaw [3, 4]. A handful of other schemes have also appeared over the years; see [11] for an overview. Collusion-secure codes are also employed in traitor tracing [5, 6]. Whereas fingerprinting protects the digital data in themselves, traitor tracing protects broadcast encryption keys.

Attacks in the watermarking layer

A real watermarking scheme cannot be expected to be infallible. We say that the extraction algorithm fail in position i if the output x_i does not match the i-th symbol of any of the pirate fingerprints. Such failure can be either accidental or due to pirate attacks, and the following causes are known.

1. *Random unintentional noise.* In digital distribution, we do not expect accidental distortions of the file, but sometimes a fingerprinted file will be transmitted in an analog medium (like radio) where no error-correction is used. When the file is distorted, the watermark may be distorted as well.

2. *Non-collusive watermarking attack.* Non-collusive watermarking attacks can be applied to any mark. By garbling the segment, the pirates cause the extraction algorithm to fail with some probability.

3. *Collusive watermarking attack.* A collusive watermarking attack applies to detectable marks. By combining different versions of the same mark, for instance by averaging, the pirates can weaken the watermark and cause extraction to fail with some probability.

4. *Cropping a segment.* A pirate can crop the file by removing certain segments.

If the pirates use a very strong watermarking attack or extensive cropping, they will also ruin the file so that it no longer be useful. This limits the success probability of the attacks. Let p_e be an upper bound on the probability that the extraction algorithm fail. This leads to a weaker Marking Assumption [7] as follows.

Definition 2 (Marking Assumption with Random Errors)
Let $P \subseteq C$ be the set of fingerprints held by a coalition of pirates, and let x_i be the output x_i from the watermarking layer in position i. The probability that for all $(c_1 \ldots c_n) \in P$, $c_i \neq x_i$, is at most p_e, independently of the output x_j for all other columns $j \neq i$.

Note that when $p_e = 0$, this coincides with the Boneh-Shaw Marking Assumption. An error-correcting adaption of the Boneh-Shaw scheme was proposed in [7]. A non-binary solution was presented in [9], protecting against deletion as well as errors, but this solution used Generalised Reed-Solomon codes requiring a very large alphabet.

The assumption of independent segments is crucial in order to use simple statistical models and formulæ. In real applications it may not be true. It is likely that some segments are independent whereas others are more or less correlated. Now it is important to remember that the pirates do not know to which code column a given segment corresponds. Thus, they will have no means to predict the correlation between to code columns, and it seems reasonable to assume independence as a fair approximation on average.

We assume that the receiver is able to synchronise before passing segments to the watermark extractor, such that the decoder will know to which code position each symbol corresponds even in presence of cropping. This can always be done if the receiver has access to the original file. The missing symbols are erasures, replaced by random symbols and treated like errors. Some authors argue that synchronisation is not always trivial and devise collusion-secure codes with deletion-correction in order to synchronise in the fingerprinting layer.

The three-layer model

It can be argued that schemes based on [4] actually use a three-layer model. The fingerprinting code of [4] is a concatenated code. It can be instructive to place the inner and outer codes in different layers. In order to simplify notation, we consider only binary schemes where $Q = \{0, 1\}$.

Lets first define a concatenated code. Take an inner binary (n_1, q) code C_1 and an outer $(n_2, M)_q$ code C_2 over Q_2. Each symbol of Q_2 is mapped on a codeword from C_1, and the codewords of the concatenated code C is formed by taking each word of C_2 and replace the symbols by words from C_1. Thus we get an $(n_1 n_2, M)$ code C.

1. *Watermarking layer.* The watermarking scheme takes a bit 0 or 1 from the fingerprinting layer and embeds it in a segment of the copyrighted file, like in the two-layer model.

2. *Inner fingerprinting layer.* The inner fingerprinting code C_1 is a (n_1, q_2) code over Q. The encoder takes a symbol $x \in Q_2$ and encodes it as a word $\mathbf{c} \in C_1$. Each bit of $\mathbf{c}_1$ is passed to the watermarking layer for embedding. The segments or bits corresponding to the same codeword of C_1, is called a *block*.

3. *Error-correcting layer.* The outer code is an (n_2, M) code C_2 over Q_2. This code has to be error-correcting, and will correct errors whether they are caused in the watermarking layer or in the inner fingerprinting layer. The encoder takes a buyer and encodes it as a codeword $\mathbf{c}_2$. Each symbol of $\mathbf{c}_2$ is then passed to the inner fingerprinting layer for encoding and embedding.

The rationale for the middle layer, is to expand the alphabet. Efficient known solutions for the top layer requires huge alphabets.

Any collusion-secure code can be used for the inner code C_1, and the output of the decoder for C_2 can be passed to the layer above. However, we can also use soft decision, as we do in this paper, and pass soft information from the middle to the top layer. The details will be explained later.

The inner fingerprinting code need not be very strong if the outer code can correct many errors. Error-correction is actually the only required property for the outer code. Decoding in the inner layer can also be allowed to be relatively costly, because the code is relatively small. It is much more important to have an efficient decoder in the top layer.

With Kerchoff's principle in mind, we assume that most of the system is public knowledge. Only parameters which can be randomly chosen at initialisation of each new application can be kept secret. Therefore, we assume that the pirates know how to divide the file into segments. On the other hand, they do not know which segment correspond to which column of C, because this mapping is a random secret permutation.

3 The Boneh-Shaw code

The Boneh-Shaw code [4] probably is the fingerprinting code most frequently referred to in the literature. It assumes an underlying watermarking layer which is error-free. They do not explicitly divide the fingerprinting problem into a middle and top layers, but the code is concatenated and fits well in the three-layer model.

A problem in the error analysis of [4] is that the middle layer is also assumed to be error-free in the sense that a very strong code is used in this layer so that the probability of error in one or more blocks is negligible. Since there are many blocks, this is rather demanding.

We have seen in [10, 12] that the outer code of the Boneh-Shaw system has some error-correcting capability. By only slightly increasing the code length in the top layer, we could correct a lot of errors from the middle layer. This allowed us to use a considerably weaker code in the middle layer, improving the overall system.

In this paper we take this one step further. The error-correcting code in the top layer can be used to correct errors both from the middle and bottom layers. We shall see that the Boneh-Shaw code essentially works fine even if random errors are permitted in the watermarking layer. Following [12], we use a soft output in the middle layer decoding.

On the BS inner code

The inner code will be called the BS code and is depicted in Figure 2. It is a binary $(r(M-1), M)$ code which is (M, ϵ)-secure. 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 $(1 \dots 10 \dots 0)$, such that the i-th $(1 \leq i \leq M)$ user has ones in the first $i-1$ types and a zeroes 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 for both these types. If r is large enough we can

$$\begin{array}{ll}
\mathbf{c}_1 & \overbrace{11\cdots1}^{r}\overbrace{11\cdots1}^{r}\overbrace{11\cdots1}^{r}\cdots\overbrace{11\cdots1}^{r} \\
\mathbf{c}_2 & 00\cdots0\,11\cdots1\,11\cdots1\cdots11\cdots1 \\
\mathbf{c}_3 & 00\cdots0\,00\cdots0\,11\cdots1\cdots11\cdots1 \\
\vdots & \quad\vdots\qquad\vdots\qquad\vdots\quad\ddots\quad\vdots \\
\mathbf{c}_q & \underbrace{00\cdots0\,00\cdots0\,00\cdots0\cdots00\cdots0}_{q-1\text{ column types}}
\end{array}$$

Figure 2: The Boneh-Shaw inner code.

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.

Theorem 1 (Boneh and Shaw)
The BS code with replication factor r is M-secure with ϵ-error whenever $r \geq 2 \cdot M^2 \cdot \log(2M/\epsilon)$.

A hybrid fingerprint is characterised by the number F_i of ones for each column type i. Let $F_0 = 0$ and $F_q = r$ by convention (as if there were a column type 0 with all zeroes, and a type q with all ones). The F_i are stochastic variables with distributions depending on the pirate strategy. If user i be innocent, the pirates cannot distinguish between column types i and $i - 1$, and consequently $F_i \sim F_{i-1}$.

The decoding algorithm of the original Boneh-Shaw scheme is based on hypotheses tests of the null hypothesis 'user i be innocent'. This hypothesis can be rejected if the auxiliary null hypothesis $F_i \sim F_{i-1}$ can be rejected. This gives a threshold such that if $|F_i - F_{i-1}|$ is sufficiently high, then user i can be accused. This provides hard input to the outer decoding algorithm.

A more efficient idea [12] is to use soft decision decoding. This means that the the inner decoding returns a reliability v_j for each $j \in C_1$. A high reliability indicates that j is likely to be a correct decoding, whereas a low value indicates that j is likely to be incorrect. We propose the following

$$(v_j : j \in C_1) \quad \text{where} \quad v_j = \frac{F_j - F_{j-1}}{r}. \tag{1}$$

Observe that all the v_j sum to 1 and $v_j \in [-1, 1]$ for all j. Furthermore, if the pirates cannot see symbol j, then $E(v_j) = 0$.

This choice may seem odd. In the case that j is innocent, F_j is expected to be close to F_{j-1}, making v_j close to zero. Whether v_j is close to $+1$ or -1, it is unlikely that j is innocent. To this we can only say that this definition worked well through the analysis, as we shall see. We did try other definitions, but the analysis became too technical to complete.

It was shown in [12] that the error probabilities are independent of the replication factor, and the introduction of random errors does not change this. Consequently we fix $r = 1$ for the rest of this paper.

On the outer code

Boneh and Shaw suggested to concatenate the BS inner code with a random code, which is constructed by picking every symbol in every codeword and decode it with closest neighbour decoding. The random code has to be kept secret by the vendor. Later [10] it has been shown that both random codes and algebraic codes with large distance, like AG or Reed-Solomon codes, have advantages as outer code. The original decoding of the outer code used closest neighbour, but in [10] we argued the utility of list decoding of the outer code, i.e. returning all codewords within a certain distance of the hybrid word after inner decoding.

After inner decoding of each block, we form the $q \times n$ reliability matrix $R = [r_{i,j}]$ where the i-th row is the vector $\mathbf{v}$ from inner decoding of the i-th block. The output of the soft decision list decoder is a list $L \subseteq C$ of codewords

$$L = \{\mathbf{c} : W(\mathbf{c}) \geq \Delta n\},$$
$$W((c_1, \ldots, c_n)) = \sum_{i=1}^{n} r_{c_i,i}.$$

We employ the common assumption that the pirates make independent decisions in each column (segment), such that all the F_i are independent and distributed as $B(1, p_i)$ for some probability p_i. This assumption is reasonable by the laws of large numbers, if there is at least a moderately large number of columns indistinguishable for the pirates.

Using Rees-Solomon or AG outer codes, this list decoding can be implemented using the Kötter-Vardy algorithm, with complexity $O(\log M)$.

It is an important property that the terms r_{i,c_i} of the sum are stochastically independent. Each term is also bounded in the interval $[-1, 1]$ and has a fairly simple distribution. This will allow us to use the well-known Chernoff bound in the error analysis.

Theorem 2 (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

$$P\left(\sum_{i=1}^{t} X_i \leq t\delta\right) \leq 2^{-tD(\delta||x)}, \quad \text{when } \delta < x,$$
$$P\left(\sum_{i=1}^{t} X_i \geq t\delta\right) \leq 2^{-tD(\delta||x)}, \quad \text{when } \delta > x,$$

where

$$D(\sigma||p) = \sigma \log \frac{\sigma}{p} + (1-\sigma)\log\frac{1-\sigma}{1-p}.$$

For an understanding of the proof of this bound, we recommend to read [8].

Error analysis

In this section, we shall bound the error probability for concatenated codes with Boneh-Shaw inner codes and soft decision decoding as defined in the previous section. This analysis follows the previous works without error-correction [12] and also works using hard decision [10]. The error analysis considers the two fingerprinting layers jointly, and the resulting error bound will depend on the parameters of both the inner and outer fingerprinting codes.

Consider the decoding of a single block, using the BS inner code. To each user is assigned a stochastic variable $X_\gamma = F_\gamma - F_{\gamma-1}$. Since $F_q = 1$ and $F_0 = 0$, we get that all the X_γ sum to 1, even with the introduction of errors.

If $j \notin P$ and $j \notin \{1, q\}$, then $E(X_j) = 0$, since F_j and F_{j-1} are identically distributed. However, for $j = 1$ we have $F_{j-1} = 0$ and for $j = q$ we have $F_j = 1$. Thus $E(X_j) = p_e$ for $j \in \{1, t\} \backslash P$. This means that user 1 and q, if they are innocent, have an increased expected decoding heuristic with the introduction of errors. For the other users, the error probability does not change a thing.

Theorem 3 (Probability of failure)
Suppose there are at most t pirates, and that they have probability at most $p_e < 1/2$ of making an error in an undetectable position. Using the concatenated code with a BS inner code and soft input

list decoding with threshold $\Delta < (1-2p_e)/t$, for the outer code, the probability of failing to accuse any guilty user is given as

$$\epsilon_I \leq 2^{-nE}, \text{ where } E = D\left(\frac{1+\Delta}{2}\,\middle|\middle|\,\frac{t+1}{2t} - \frac{p_e}{t}\right). \qquad (2)$$

This bound is independent of the choice of outer code.

We observe that for $p_e = 0$, the above theorem reduces to the original result of [12].

Proof: The probability ϵ_I that the decoding algorithm outputs no guilty user, is bounded as

$$\epsilon_I \leq P\left(\frac{1}{t}\sum_{i=1}^{n}\sum_{\mathbf{c}\in P} r_{i,c_i} \leq \Delta n\right) = P\left(\sum_{i=1}^{n} Y_i \leq \Delta n\right).$$

where

$$Y_i = \sum_{\mathbf{c}\in P}\frac{r_{i,c_i}}{t} = \frac{1}{t}\sum_{\mathbf{c}\in P} F_{c_i} - F_{c_i-1}.$$

Obviously

$$\sum_{\gamma\in Q} F_\gamma - F_{\gamma-1} = 1.$$

Let $P_i \subseteq Q$ be the set of symbols seen by the pirates in position i, i.e. $P_i = \{c_i : \exists(c_1,\ldots,c_n) \in P\}$. Write $a = (\min P_i) - 1$, and $b = \max P_i$. Then we have $E(F_a) \leq p_e$ and $E(F_b) \geq (1-p_e)$. Hence

$$E(\sum_{i=a}^{b} r_i) \geq 1 - 2p_e,$$

and since $E(r_{i,\gamma}) = 0$ when $\gamma \notin P_i \cup \{1,t\}$, we get

$$E(Y_i) = E(\frac{1}{t}\sum_{\mathbf{c}\in P} r_{i,c_i}) \geq \frac{1-2p_e}{t},$$

Note that $Y_i \in [-1,1]$, so in order to get a stochastic variable in the $[0,1]$ range, we set $X_i = (1+Y_i)/2$. Thus

$$\epsilon_I \leq P\left(\sum_{i=1}^{n} X_i \leq \frac{1+\Delta}{2}n\right).$$

If $\Delta < (1-2p_e)/t$, the Chernoff bound is applicable, proving the theorem. □

The probability of failure as derived above is independent of the choice of outer codes. The probability of false accusations on the other hand, must be derived separately for different classes of outer codes. Below, we find this error rate for random outer codes, as suggested by Boneh and Shaw.

Using random codes as outer codes, the probability of Type II errors is independent of p_e. We recall that each symbol of a codeword is drawn uniformly and independently at random. We express this as the following theorem.

Theorem 4 (Error rate for random codes)
Concatenating a $(q-1,q)$ BS inner code with a random outer code using soft input list decoding with threshold $\Delta > 1/q$ for the outer code, the probability of accusing an innocent user is given as

$$\epsilon_{\mathrm{II}} \leq 2^{(R_O\log q - E)n}, \text{ where } E = D\left(\frac{1+\Delta}{2}\,\middle|\middle|\,\frac{q+1}{2q}\right).$$

Proof: Let $\mathbf{c} \notin P$ be an innocent user. The probability of accusing $\mathbf{c}$ is

$$\pi(\mathbf{c}) = P\left(\sum_{i=1}^{n} Y_i \geq \Delta n\right), \tag{3}$$

where $Y_i = r_{i,c_i}$ where c_i is drawn uniformly at random from C_1. Recall that the r_{i,c_i} for $i = 1,\ldots,q$ sum to 1. Hence $E(r_{i,c_i}) = 1/q$. Like in the last section, we make a stochastic variable in the $[0,1]$ range,

$$X_i = \frac{1+Y_i}{2}, \tag{4}$$

$$E(X_i) = \frac{q+1}{2q}, \tag{5}$$

and

$$\pi(\mathbf{c}) = P\left(\sum_{i=1}^{n} X_i \geq \frac{1+\Delta}{2} n\right). \tag{6}$$

□

Theorem 5

For any $q > t$, there is an asymptotic class of (t,ϵ)-secure codes with $\epsilon \to \infty$ and rate given by

$$R_t \approx \frac{D\left(\frac{t+1-2p_e}{2t} || \frac{q+1}{2q}\right)}{q-1}, \quad \text{if} \quad \frac{t+1-2p_e}{2t} > \frac{q+1}{2q}.$$

Proof: For asymptotic codes, $\epsilon_I \to 0$ if $\Delta < (1-2p_e)/t$, so we can take $\Delta \approx (1-2p_e)/t$. Likewise, $\epsilon_{\text{II}} \to 0$ if $\Delta > 1/q$ and

$$R_O < \frac{D\left(\frac{t+1-2p_e}{2t} || \frac{q+1}{2q}\right)}{\log q}.$$

Since $R_I = \log q/(q-1)$, we get the theorem. □

Unfortunately, we cannot see any nice expression for the optimal value of q. Clearly, we require $q = \Omega(t)$, and if $q = \Theta(t)$, we get $R_t = \Omega(t^{-3})$. The only scheme with $R_t = \Omega(t^{-3})$ is the Tardos scheme with $R_t = \Theta(t^{-2})$, but that scheme is subject to adverse selection. Figure 3 gives an impression of code rates for $q = 2t, 3t, 4t$.

Theorem 6

Suppose there are at most t pirates, and that they have probability at most $p_e < 1/2$ of making an error in an undetectable position. Concatenating a $(q-1,q)$ BS inner code with a $(n, 2^{R_O n}, \delta n)$ outer code using soft input list decoding with threshold Δ for the outer code, the probability of accusing an innocent user is given as

$$\epsilon_{\text{II}} \leq 2^{(R_O \log q - [1-t(1-\delta)] D(\sigma || (1+p_e)/2))n},$$

provided $\Delta > t(1-\delta)$, and

$$\sigma = \frac{1}{2} + \frac{\Delta - t(1-\delta)}{2(1-t(1-\delta))}.$$

Proof: Let $\mathbf{c} \notin P$ be some innocent user. We want to bound the probability of accusing $\mathbf{c}$,

$$\pi(\mathbf{c}) \leq P\left(\sum_{i=1}^{n} r_{i,c_i} \geq \Delta n\right). \tag{7}$$

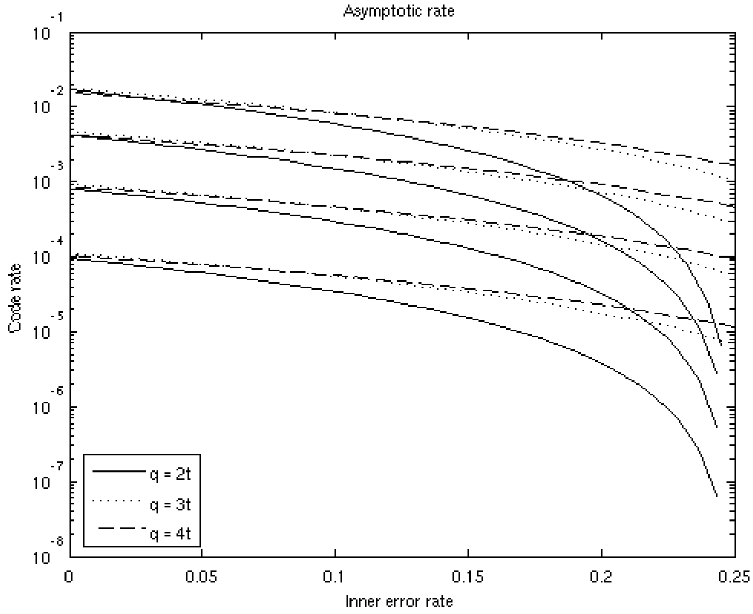

Figure 3: Code rates for concatenated codes with BS inner codes and random codes for varying underlying error rates and varying q for $t = 2, 3, 5, 10$.

An innocent user $\mathbf{c}$ can match a given pirate in at most $(1-\delta)n$ positions. Thus there are at most $t(1-\delta)n$ positions where $\mathbf{c}$ matches some pirate. For the purpose of a worst case analysis, we assume that $r_{i,c_i} = 1$ whenever c_i matches a pirate. There are at least $N = [1 - t(1-\delta)]n$ positions $i_1, \ldots, i_N$, where $r_{i,j} = v_j$ is given by (1) with $F_j \sim F_{j-1}$. Thus we get

$$\pi(\mathbf{c}) \leq P\left(\sum_{i=1}^{N} r_{i,c_i} \geq \tau N\right), \tag{8}$$

$$N = [1 - t(1-\delta)]n, \tag{9}$$

$$\tau = \frac{\Delta - t(1-\delta)}{1 - t(1-\delta)}. \tag{10}$$

Clearly, τ increases in δ as well as in Δ.

Suppose c_i is not seen by any pirate. Recall that if $c_i \notin \{1, q\}$, then $F_{c_i} \sim F_{c_i - 1}$ and consequently $E(r_{i,\gamma}) = 0$, independently of p_e. For $c_i \in \{1, q\}$ however, we get $E(r_{i,c_i}) = p_e$. Setting $Y_i = (1 + r_{i,c_i})/2$, we get $E(Y_i) \leq (1 + p_e)/2$ and

$$\pi(\mathbf{c}) \leq P\left(\sum_{j=1}^{N} Y_j \geq \frac{1+\tau}{2} N\right). \tag{11}$$

The theorem follows by the Chernoff bound. □

For asymptotic codes, $\epsilon_{\mathrm{I}} \to 0$ if $\Delta < 1/t$, so we can take $\Delta \approx 1/t$. Likewise, $\epsilon_{\mathrm{II}} \to 0$ if both $\Delta > t(1-\delta)$ and

$$R_O < \frac{1 - t(1-\delta)}{\log q} D(\sigma || (1 + p_e)/2).$$

Using AG codes with

$$R = 1 - \delta - \frac{1}{\sqrt{q} - 1},$$

where q is an even prime power, we can get codes with R_O solving the following

$$R_O = \frac{1 - t\left(R_O + \frac{1}{\sqrt{q}-1}\right)}{\log q} D\left(\frac{1}{2} + \frac{1}{2} \cdot \frac{1 - t^2\left(R_O + \frac{1}{\sqrt{q}-1}\right)}{t - t^2\left(R_O + \frac{1}{\sqrt{q}-1}\right)} \,\Big|\Big|\, \frac{1 + p_e}{2}\right), \tag{12}$$

$$0 < \frac{1 - t^2\left(R_O + \frac{1}{\sqrt{q}-1}\right)}{t - t^2\left(R_O + \frac{1}{\sqrt{q}-1}\right)}. \tag{13}$$

The total rate is $R_t(q) = R_I \cdot R_O$ where

$$R_I = \frac{\log q}{q - 1}.$$

The number of pirates t, is a property of the resulting codes, whereas q is a control parameter chosen so as to maximise R_t.

4 Conclusion

We have showed that an efficient collusion-secure code with error-correction can be built based on the Boneh-Shaw code. The error-correction helps to build a complete watermarking/fingerprinting scheme resistant to attacks on the watermarking layer. The impact of errors on the information rate is surprisingly low.

We also note that, unlike past schemes for this model [7], we have incorporated all the latest improvements of Boneh-Shaw, using soft-decision decoding and error-correcting codes. No formulæ for the information rate are given in [7] so an exact comparison is omitted. However, our information rate, even with 25% errors, are better than those of the original BS scheme with 0 error rate, and [7] did not introduce anything to obtain such improvement.

We expect to use the same techniques to make collusion-secure codes with error-correction based on other known codes, such as [2, 13]. A more challenging problem is probably to make good non-binary codes with error-correction.

References

[1] International Intellectual Property Alliance, fact sheet. `http://www.iipa.com/aboutiipa.html`.

[2] 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.

[3] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology - CRYPTO'95*, volume 963 of *Springer Lecture Notes in Computer Science*, pages 452–465, 1995.

[4] 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.

[5] B. Chor, A. Fiat, and M. Naor. Tracing traitors. In *Advances in Cryptology - CRYPTO '94*, volume 839 of *Springer Lecture Notes in Computer Science*, pages 257–270. Springer-Verlag, 1994.

[6] 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.

[7] Hans-Jürgen Guth and Birgit Pfitzmann. Error- and collusion-secure fingerprinting for digital data. In *Information Hiding '99, Proceedings*, volume 1768 of *Springer Lecture Notes in Computer Science*, pages 134–145. Springer-Verlag, 2000.

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

[9] R Safavi-Naini and YJ Wang. Traitor tracing for shortened and corrupted fingerprints. In *Digital rights management*, volume 2696 of *Springer Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[10] Hans Georg Schaathun. The Boneh-Shaw fingerprinting scheme is better than we thought. Technical Report 256, Dept. of Informatics, University of Bergen, 2003. Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`.

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

[12] Hans Georg Schaathun and Marcel Fernandez-Muñoz. Boneh-Shaw fingerprinting and soft decision decoding. In *Information Theory Workshop*, 2005. Rotorua, NZ.

[13] Gábor Tardos. Optimal probabilistic fingerprint codes. *Journal of the ACM*, 2005. `http://www.renyi.hu/~tardos/fingerprint.ps`. To appear. In part at STOC'03.

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