Multimedia Systems manuscript No.
(will be inserted by the editor)

Hans Georg Schaathun

On error-correcting fingerprinting codes for use with watermarking

the date of receipt and acceptance should be inserted later

Abstract Digital fingerprinting has been suggested for copyright protection. Using a watermarking scheme, a fingerprint identifying the buyer is embedded in every copy sold. If an illegal copy appears, it can be traced back to the guilty user. By using collusion-secure codes, the fingerprinting system is made secure against cut-and-paste attacks.

In this paper we study the interface between the collusion-secure fingerprinting codes and the underlying watermarking scheme, and we construct several codes which are both error-correcting and collusion-secure. Error-correction makes the system robust against successful attacks on the watermarking layer.

Keywords collusion-secure fingerprinting, copyright protection, error-correcting codes, watermarking, 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 instance, 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 are in the process of changing their legislation to deal more effectively with illegal distribution in new media.

H.G. Schaathun
University of Surrey
Dept. Computing
GU2 7XH Guildford
England
E-mail: H.Schaathun@surrey.ac.uk

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 (FP) was suggested as a forensic technique in [25], 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 FP system robust to any conceivable attack is a challenging task. One typical attack would be the cut-and-paste attack, where a collusion of pirates cut segments from their individual copies and paste them together to form a hybrid copy with a hybrid fingerprint.

An FP system is often divided into two modules. A watermarking (WM) scheme (see e.g. [7]) is used to embed the fingerprint in the digital file. The fingerprint is a sequence $(c_1, \ldots, c_n)$ where each symbol c_i is embedded independently into one segment of the file in the WM layer. The set of all fingerprints is called a *code*. In order to make the system resistant against collusive attacks (e.g. cut-and-paste attacks), a *collusion-secure code* can be used. Most of the literature studies the WM scheme and the collusion-secure code separately as black boxes, and we follow and refine this view.

The FP literature has defined its requirements for the WM scheme as a Marking Assumption. Different variants of this assumption exist. The most common one [4] says that when a collusion make a hybrid copy, each fingerprint symbol in the hybrid copy will match the fin-

gerprint of at least one of the members. Unfortunately, this assumptions is unrealistically strong. Regular WM attacks can cause decoding errors in the WM layer for some of the symbols.

In this paper, following Guth and Pfitzmann [9], we use a weaker Marking Assumption, which allows for some successful attacks in the WM layer as well as the cut-and-paste attack. The solution is codes that are both collusion-secure and error-correcting.

One of the most celebrated collusion-secure codes is due to Boneh and Shaw [4]. Codes for the Guth-Pfitzmann Marking Assumption were developed in in [9,15,27], and all of these were based on the Boneh-Shaw code. Recently, we have seen [20] that the Boneh-Shaw code is theoretically more secure than originally assumed for the Boneh-Shaw Marking Assumption. In [22] we saw that the code can be further improved by using soft decision decoding.

Collusion-secure fingerprinting has been studied both from experimental and theoretical angles. A survey of the field can be found in [26]. A recent paper [11] made an experimental analysis of a joint WM/FP system. Although it did not directly use the theoretical properties of the collusion-secure code used, it was ground-breaking as the first attempt to combine collusion-secure codes with real WM schemes.

In this paper, we analyse the Boneh-Shaw scheme with soft decision decoding (BS-SD) in view of the Guth-Pfitzmann Marking Assumption. Our goals are (1) to show that BS-SD is theoretically secure under the Guth-Pfitzmann Marking Assumption, and that it is theoretically more efficient than previously known solutions; and (2) to demonstrate that it works in conjunction with a real WM scheme, with efficiency comparable to other known joint WM/FP schemes. Thirdly, we propose a variant of BS-SD using algebraic outer codes. This has theoretical advantages for large parameters, but could not be realised for the parameters used in our experiments (up to 10 000 users) for comparison with previous works.

We will start by defining the watermark/fingerprinting model in Section 2. The model is a refined variant of [9]. We emphasise the advantages of a modular model and the use of black boxes. We also discuss how various attack relates to the model. The main result, namely the analysis of BS-SD [21], is presented in Sections 3 (theoretical analysis) and 4 (experimental analysis). In the final section we present conclusions and open problems.

2 The layered fingerprint/watermark model

In this section, we develop the layered model which will allow us to analyse each component of the system theoretically. We will first define basic notation and terminolgy from coding theory, which we will use throughout. We present a two-layer model for fingerprinting in Section 2.2, before we discuss respectively fingerprinting and watermarking attacks on the system in Sections 2.3 and 2.4. In Section 2.5, we will discuss a three-layer model. At the end of the section we stress the advantages of a layered model.

2.1 Coding theory

An (n, M) *code* C is a set of M distinct codewords $(c_1, \ldots, c_n)$ of length n. Each element c_i is drawn from some *alphabet* Q, i.e. a discrete set. If Q has q elements, we also say that C is an $(n, M)_q$ code.

The *minimum (Hamming) distance* between two distinct codewords is denoted by $d = \delta n$. An (n, M, d) code is an (n, M) code with minimum distance d. The (information) *rate* of a code is defined as $R = (\log_q M)/n$.

If Q is a finite field F_q for q elements, and if C is a vector space over F_q, then C is a *linear code*. An $[n, k, d]_q$ code is a linear code of dimension k (size $M = q^k$).

In general the decoding problem is NP-complete, but many linear codes with known algebraic structure have faster decoding algorithms. We will use two such codes in this paper. The Reed-Solomon (RS) codes allow constructions of $[n, k, n + 1 - k]_q$ codes for any $n \leq q$ and any k, $1 \leq k \leq n$ (see e.g. [14]). Algebraic Geometry (AG) codes allow asymptotic constructions according to the proposition from [24] below. CRT codes were used in [27,16], but the parameters appear to be the same as for Reed-Solomon codes so we do not study them separately.

Proposition 1 *For any $\alpha > 0$ there are constructible, infinite families of codes with parameters $[N, NR, N\delta]_q$ for $N \geq N_0(\alpha)$ and*

$$R + \delta \geq 1 - (\sqrt{q} - 1)^{-1} - \alpha,$$

where q is an even prime power.

A *concatenated code* is defined as follows. Take an inner binary $(n_1, q)_{q'}$ code C_{I} and an outer $(n_2, M)_q$ code C_{O} over Q. Each symbol of Q is mapped on a codeword from C_{I}, and the codewords of the concatenated code C is formed by taking each word of C_{O} and replace the symbols by words from C_{I}. Thus we get an $(n_1 n_2, M)_{q'}$ code C.

2.2 The two-layer model

Guth and Pfitzmann [9] pointed out how the standard FP models fitted into a layered structure, with a WM layer and a FP layer. (Cf. 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

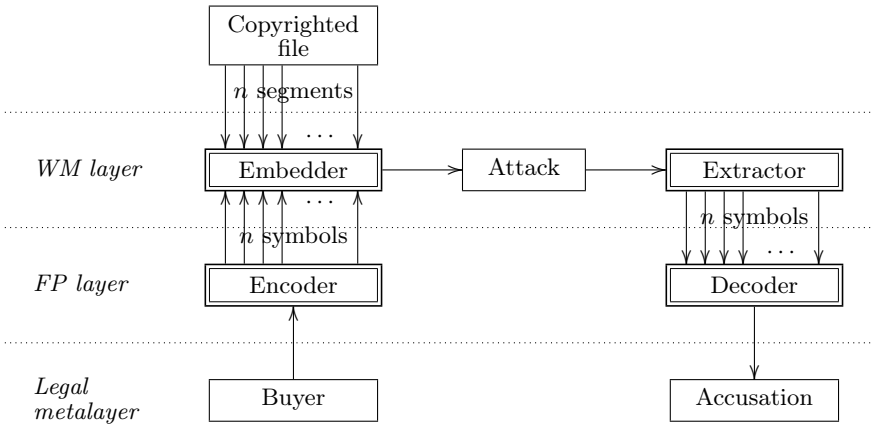


Fig. 1: Watermarking/Fingerprinting model.

a *mark*. The *extraction algorithm* takes a mark and returns a message from Q.

In *the fingerprinting layer*, each user is identified by a codeword from some (n, M) code C over Q. The code C has M codewords (fingerprints), 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 WM layer to be embedded in the digital file. Observe that the copyrighted file is not used in the FP layer at all. All interaction with the media file occurs in the WM layer; the FP layer is completely media-independent.

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

If no change has been made to the fingerprinted file, the copy can be traced back to a user. In fact $\mathbf{x} \in C$ and it corresponds to the buyer of this copy. A pirate however, does not want to be identified, so he will make attack, either on the WM or the FP scheme, in order to cause the decoding to fail. We will discuss possible attacks shortly.

Let $P \subseteq C$ be the set of pirate fingerprints, and $L \subseteq C$ the output for the fingerprint decoder. If $L = \emptyset$, we say that we have an error of *Type I* or a *failure*. If $L \not\subseteq P$, there is an innocent user accused by the decoder, and we call this a *false accusation* or an error of *Type II*. If we have no error of either type, then L is a non-empty subset of the pirates, and the decoder has been successful.

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 segment.

We have assumed that the WM extractor always returns a single element of Q. This is a simplification. Some systems will also be able to return 'erasure' when no one element appears as more likely than others. In order to keep the model simple, we replace any erasure by a random symbol and treat it as an error.

2.3 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 from the fingerprinting decoder does not contain any of the pirates.

A collusion of t pirates have access to t distinct copies with different fingerprints. Comparing the copies, they will see some segments which are different (called detectable marks) and some which are identical (called undetectable marks).

One known attack is available in the FP layer, 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 assume that only cut-and-paste attacks are possible. The classic phrasing of this assumption 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 $\mathbf{x} \in F(P)$, where

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

We call $F(P)$ 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 subset 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 [19] 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.

2.4 Attacks in the watermarking layer

A real WM 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.* If the file is transmitted over an analog medium (like radio), it will be distorted by random noise. When the file is distorted, the watermark may be distorted as well.
2. *Non-collusive watermarking attack.* Non-collusive WM attacks can be applied to any mark. By garbling the segment, the pirates cause the extraction algorithm to fail with some probability. Lossy compression can also work as a non-collusive attack.
3. *Collusive watermarking attack.* A collusive WM attack applies to detectable marks. This is similar to collusive attacks in the FP layer, but operates on a single segment, and may result in a hybrid segment different from all of the pirates' copies. The most common example is the averaging attack, where the pirates use an average of all their copies.
4. *Cropping a segment.* A pirate can crop the file by removing certain segments.

If the pirates use a very strong WM 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 ϵ_{in} be an upper bound on the probability that the extraction algorithm fail. This leads to a weaker Marking Assumption [9] 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 $(c_1 \dots c_n) \in P$, $c_i \neq x_i$, is at most ϵ_{in}, independently of the output x_j for all other columns $j \neq i$.

Note that when $\epsilon_{\text{in}} = 0$, this coïncides with the Boneh-Shaw Marking Assumption. An error-correcting adaption of the Boneh-Shaw scheme was proposed in [9]. A non-binary solution was presented in [17], 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. Statistical dependence is most likely to exist within local neighbourhoods. 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 two code columns, and it seems reasonable to assume independence as a fair approximation, though we have to assert it for potential WM schemes.

We assume that the receiver is able to synchronise, even in presence of cropping, before passing segments to the WM extractor. 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 FP layer.

It is an open question if this Marking Assumption be true for various known watermarking schemes. The assumption is reasonable, as for the most widely accepted schemes, there is no known attack which allows the attacker to succeed with certainty. For each level of distortion, the success rate of the attack can be limited by ϵ_{in}. It is irrelevant to our construction which attack is used, as long as the success rate is bounded by ϵ_{in}.

When the FP code is binary, collusive attacks in the WM layer have no effect. A collusive attack depends on the colluders seeing different watermarks. In the binary case, this means that they see all possible watermarks, and any decoding would be correct.

2.5 The three-layer model

It can be argued that schemes based on [4] actually use a three-layer model. The FP code of [4] is a concatenated code. It can be instructive to place the inner and outer codes in different layers, as in Figure 2.

1. *Watermarking layer.* The WM embedder takes an element $x_i \in Q'$ and a segment u_i of the file, and outputs a watermarked segment w_i. The WM extractor inverts this; given w_i it outputs x_i.
2. *Inner fingerprinting layer.* The inner fingerprinting code C_{I} is an (n_1, q) code over Q'. The encoder takes a symbol $x \in Q$ and encodes it as a word $\mathbf{c}_1 \in C_{\text{I}}$. Each position of $\mathbf{c}_1$ is passed to the watermarking layer for embedding. The symbols corresponding to one symbol from Q is called a *block*.
3. *Error-correcting (EC) layer.* The outer code is an (n_{O}, M) code C_{O} over Q. 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

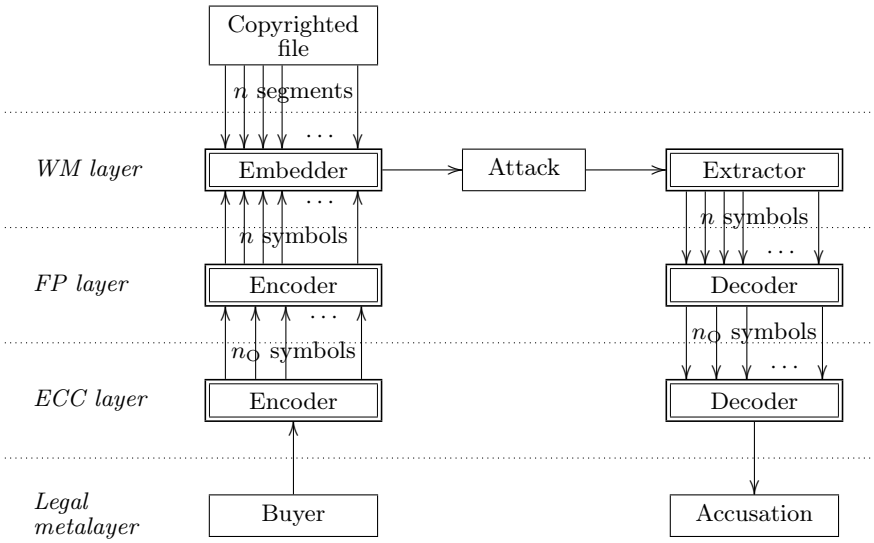


Fig. 2: The three-layer model.

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 require huge alphabets. The inner FP code need not be very strong if the outer code can correct many errors [18]. 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 ECC layer.

With Kerchoffs' principle [12] 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; these parameters are known as the *key*. 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, as this mapping is a random secret permutation.

2.6 The advantages of layered models

Each layer in the model uses different types of information processing. The WM layer, interacting with the media file, is mainly signal processing. The FP and ECC layers use coding theory and small, discrete alphabets. The FP code may have to be designed specifically for fingerprinting, but the ECC code is commonly a standard-issue code known from other applications.

The layered model allows us to farm out the different components to experts of various fields, and reuse components from other areas of research. However, this is only possible if the interfaces between layers are well-defined and agreed.

Our focus in this paper will be on the FP and ECC layers, and we will only make minor suggestions for the WM layer. Hopefully, this will inspire further research on the topic by watermarking experts.

3 The Boneh-Shaw code

The BScode [4] is probably the FP code most frequently referred to in the literature. In [18], we proved that by viewing the outer code as an error-correcting code, we can improve the error bound significantly, without changing the FP scheme. A second step was made in [21], where a new modified scheme was developed, using soft decision decoding for the BS inner code.

In this paper, we investigate how the BS scheme with Soft Decision decoding (BS-SD) performs in the presence of random errors. New error bounds are proved in Theorems 2, 3, and 5, taking random errors into account. We also prove theorems about the asymptotic performance of the construction. We note in particular that the achievable rate degrades only slowly in the error rate from the WM layer. All the theorems are generalisations of previous results for the Boneh-Shaw Marking Assumption [21].

3.1 On the BS inner code

Boneh and Shaw used an $(r(q-1), q)$ inner code. We saw in [21], that the optimal choice is $r = 1$, giving a $(q-1, q)$ code where the codewords form an upper triangular 0-1

matrix, with ones on and above the diagonal, i.e.

$$\begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \\ \mathbf{c}_3 \\ \vdots \\ \mathbf{c}_{q-1} \\ \mathbf{c}_q \end{bmatrix} = \begin{bmatrix} 1\,1\,1\cdots 1 \\ 0\,1\,1\cdots 1 \\ 0\,0\,1\cdots 1 \\ \vdots\,\vdots\,\vdots\,\ddots\,\vdots \\ 0\,0\,0\cdots 1 \\ 0\,0\,0\cdots 0 \end{bmatrix}.$$

Note that Q is represented as natural numbers $\{1, 2, \ldots, q\}$, and $i \in Q$ is encoded as the i-th row $\mathbf{c}_i$ of the inner code.

Let $(X_1, \ldots, X_{q-1})$ be a hybrid fingerprint. Let $X_0 = 0$ and $X_q = 1$ by convention. Note that unless user i is a pirate, the pirates cannot distinguish between the $(i-1)$-th and the i-th column. Hence the probability of outputting a 1 is equal for the two columns, i.e. $X_i \sim X_{i-1}$.

The output from the inner decoding algorithm is the vector

$$(V_j : j \in C_{\mathrm{I}}) \quad \text{where} \quad V_j = X_j - X_{j-1}. \tag{1}$$

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 and $j \notin \{1, q\}$, then $E(V_j) = 0$.

3.2 On the outer code

Boneh and Shaw suggested to concatenate the BS inner code with a random code, decoded using closest neighbour decoding. Each symbol in each codeword is drawn independently and uniformly at random. The random code has to be kept secret by the vendor. Muratani [15, 16] and others have combined the BS inner code and algebraic outer codes with large minimum distance in a different model with a stronger marking assumption. Outer codes with large distance was also studied in the Boneh-Shaw model in [18], and even though the code rates are inferior to random codes, they are reasonably good, and algebraic codes have the advantage of more efficient decoding.

The closest neighbour decoding used in [4] returns the codeword $\mathbf{c} \in C_{\mathrm{O}}$ minimising the Hamming distance $d(\mathbf{x}, \mathbf{c})$, where $\mathbf{x}$ is the hybrid fingerprint (after each block has been decoding using the inner code). In [18] we argued the utility of list decoding of the outer code, i.e. we return all codewords $\mathbf{c}$ within a certain distance of $d(\mathbf{c}, \mathbf{x}) \leq \Delta n_{\mathrm{O}}$.

Soft decision decoding of the Boneh-Shaw scheme was introduced in [21]. 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 $(V_1, \ldots, V_q)$ from inner decoding of the i-th block. The output of the soft decision list decoder is a list $L \subseteq C$ of codewords

$$L = \{\mathbf{c} : W(\mathbf{c}) \geq \Delta n\}, \tag{2}$$

$$W((c_1, \ldots, c_n)) = \sum_{i=1}^{n} r_{i,c_i}. \tag{3}$$

For random codes, the list decoding has to be implemented as an exhaustive search with complexity $O(M)$. Using Reed-Solomon or AG outer codes, we can use the Kötter-Vardy algorithm [13], which is a soft-decision variant of the Guruswami-Sudan algorithm [8] and has complexity $O(\log M)$.

We employ the common assumption that the pirates make independent decisions in each column (segment), such that all the X_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. Most importantly, this assumption implies that the r_{i,c_i} for different i are stochastically independent, allowing us to use the well-known Chernoff bound, defined as follows.

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 $\delta \in [0, 1]$, we have*

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

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

where

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

A nice presentation of this result and its proof can be found in [10].

The probability of failure is independent of the choice of outer code, and we present the result below. The probability of false accusations on the other hand, must be derived separately for different classes of outer codes, and this is done in subsequent subsections.

Theorem 2 (Probability of failure) *Suppose there are at most t pirates, and that they have probability at most $\epsilon_{\mathrm{in}} < 1/2$ of causing an error in an undetectable position. Using the concatenated code with a BS inner code and soft input list decoding with threshold $\Delta < (1 - 2\epsilon_{\mathrm{in}})/t$, for the outer code, the probability of failing to accuse any guilty user is given as*

$$\epsilon_{\mathrm{I}} \leq \exp -n_{\mathrm{O}} D\left(\frac{1+\Delta}{2} || \frac{t+1}{2t} - \frac{\epsilon_{\mathrm{in}}}{t}\right).$$

This bound is independent of the choice of outer code.

For $\epsilon_{\mathrm{in}} = 0$, the above theorem reduces to the original result of [21].

Proof The probability ϵ_{I} that the decoding algorithm outputs no guilty user, is bounded as

$$\epsilon_{\mathrm{I}} \le P\left(\frac{1}{t}\sum_{i=1}^{n_{\mathrm{O}}}\sum_{\mathbf{c}\in P} r_{i,c_i} \le \Delta n_{\mathrm{O}}\right) = P\left(\sum_{i=1}^{n_{\mathrm{O}}} Y_i \le \Delta n_{\mathrm{O}}\right).$$

where

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

Obviously $\sum_{\gamma\in Q} V_\gamma = 1$.

Let $P_i \subseteq Q$ be the set of symbols seen by the pirates in position i, i.e. $P_i = \{c_i : \exists(c_1,\ldots,c_{n_{\mathrm{O}}}) \in P\}$. Write $a = (\min P_i)-1$, and $b = \max P_i$. Then we have $E(X_a) \le \epsilon_{\mathrm{in}}$ and $E(X_b) \ge (1-\epsilon_{\mathrm{in}})$. (We have $E(X_a) = \epsilon_{\mathrm{in}}$ if $a = 1$ and $E(X_a) = 0$ if $a > 1$, and $E(X_b) = 1-\epsilon_{\mathrm{in}}$ for $b = q$ and $E(X_b) = 1$ for $b < q$.) Hence

$$E(\sum_{i=a}^{b} r_{i,c_i}) \ge 1 - 2\epsilon_{\mathrm{in}},$$

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

$$E(Y_i) = E\left(\frac{1}{t}\sum_{\mathbf{c}\in P} r_{i,c_i}\right) \ge \frac{1-2\epsilon_{\mathrm{in}}}{t},$$

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

$$\epsilon_{\mathrm{I}} \le P\Big(\sum_{i=1}^{n} Z_i \le \frac{1+\Delta}{2} n_{\mathrm{O}}\Big).$$

If $\Delta < (1-2\epsilon_{\mathrm{in}})/t$, the Chernoff bound is applicable, proving the theorem. □

3.3 Random outer codes

In this section, we consider random outer codes, where each symbol of every codeword is drawn uniformly at random. The analysis follows [21,18].

Theorem 3 (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*

$$\epsilon_{\mathrm{II}} \le 2^{(R_{\mathrm{O}}\log q - E)n_{\mathrm{O}}}, \text{ where } E = D\left(\frac{1+\Delta}{2}\Big|\Big|\frac{q+1}{2q}\right).$$

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

$$\pi(\mathbf{c}) = P\left(\sum_{i=1}^{n_{\mathrm{O}}} Y_i \ge \Delta n_{\mathrm{O}}\right),$$

where $Y_i = r_{i,c_i}$ where c_i is drawn uniformly at random from Q. Recall that the r_{i,c_i} for $c_i = 1,\ldots,q$ 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, $Z_i = (1+Y_i)/2$, to get

$$E(Z_i) = \frac{q+1}{2q} \quad \text{and} \quad \pi(\mathbf{c}) = P\left(\sum_{i=1}^{n_{\mathrm{O}}} Z_i \ge \frac{1+\Delta}{2} n_{\mathrm{O}}\right).$$

The theorem follows by applying the Chernoff bound and multiplying by the number of innocent users $\approx 2^{n_{\mathrm{O}} R_{\mathrm{O}} \log q}$. □

Example 1 Suppose we require a Boneh-Shaw scheme with $t = 20$, $M = 2^{20}$, $\epsilon_{\mathrm{in}} = 2\%$, $\epsilon_{\mathrm{II}} = 10^{-3}$, and $\epsilon_{\mathrm{I}} = 10^{-6}$. We use $q = 3t$. Setting equality in Theorems 2 and 5, we get

$$3\log 10 = D\left(\frac{1+\Delta}{2}\Big|\Big|\frac{21}{40} - \frac{\epsilon_{\mathrm{in}}}{20}\right) n_{\mathrm{O}},$$
$$6\log 10 = D\left(\frac{1+\Delta}{2}\Big|\Big|\frac{61}{120}\right) n_{\mathrm{O}} - 20.$$

We solve the equations to get $\Delta \approx 0.03757$ and $n_{\mathrm{O}} \approx 126\,660$, and consequently $n = 7\,472\,940$.

Remark 1 Similar calculations to the example for fewer pirates give length 5 655 for $t = 2$, 21 744 for $t = 3$, 109 074 for $t = 5$, and 915 385 for $t = 10$, still assuming a million users and 2% WM errors.

Theorem 4 *There is an asymptotic class of (t,ϵ)-secure codes with $\epsilon \to \infty$ and rate given by*

$$R_t \approx \frac{D\left(\frac{t+1-2\epsilon_{\mathrm{in}}}{2t}\Big|\Big|\frac{q+1}{2q}\right)}{q-1}, \quad \textit{for any} \quad q > \frac{t}{1-2\epsilon_{\mathrm{in}}}.$$

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

$$R_{\mathrm{O}} < \frac{D\left(\frac{t+1-2\epsilon_{\mathrm{in}}}{2t}\Big|\Big|\frac{q+1}{2q}\right)}{\log q}.$$

Since $R_{\mathrm{I}} = \log q/(q-1)$, we get the theorem. □

The theorem obviously demands $q = \Omega(t)$, but we cannot see any nice expression for the optimal value of q. Asymptotic error rates are shown in Figure 3 for various alphabet sizes. Large alphabets appear to be better asymptotically.

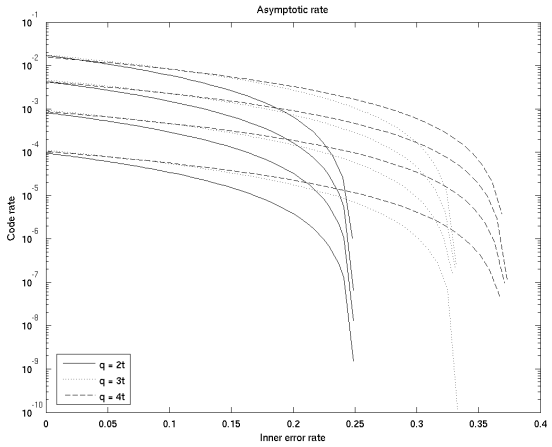


Fig. 3: Code rates for concatenated codes with BS inner codes and random codes for varying underlying error rates and varying q for $t = 2, 3, 5, 10$.

3.4 Outer codes with large distance

Theorem 5 *Suppose there are at most t pirates, and that they have probability at most $\epsilon_{\text{in}} < 1/2$ of causing an error in an undetectable mark. Concatenating a $(q-1, q)$ BS inner code with an $(n_{\text{O}}, 2^{R_{\text{O}} n_{\text{O}}}, \delta n_{\text{O}})$ outer code using soft input list decoding with threshold Δ for the outer code, the probability of accusing an innocent user is given as*

$$\epsilon_{\text{II}} \leq \exp\left(R_{\text{O}} \log q - E\right) n_{\text{O}},$$

where

$$E = [1 - t(1-\delta)] D\left(\frac{1}{2} + \frac{\alpha}{2} || \frac{1}{2} + \frac{\epsilon_{\text{in}}}{q}\right),$$
$$\alpha = \frac{\Delta - t(1-\delta)}{1 - t(1-\delta)}, \tag{4}$$

provided that $\alpha > 2\epsilon_{\text{in}}/q$.

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

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

An innocent user $\mathbf{c}$ can match a given pirate in at most $(1-\delta)n_{\text{O}}$ positions. Thus there are at most $t(1-\delta)n_{\text{O}}$ 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 $N = [1-t(1-\delta)]n_{\text{O}}$ positions $i_1, \ldots, i_N$, where $r_{i,c_i} = V_{c_i}$ and the pirates do not see c_i. Thus we get

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

where α is given by (4) and

$$N = [1 - t(1-\delta)]n_{\text{O}}.$$

Clearly, α 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 $X_{c_i} \sim X_{c_i - 1}$ and consequently $E(r_{i,\gamma}) = 0$, independently of ϵ_{in}. For $c_i \in \{1, q\}$ however, we have $E(r_{i,c_i}) = \epsilon_{\text{in}}$. Since the encodings ϕ_i are uniform and random, we have $P(c_i \in \{1, q\}) = 2/q$, and hence $E(r_{i,c_i}) = 2\epsilon_{\text{in}}/q$. Setting $Y_i = (1 + r_{i,c_i})/2$, we get $E(Y_i) \leq 1/2 + \epsilon_{\text{in}}/q$ and

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

The theorem follows by applying the Chernoff bound and multiplying by the number of innocent users $\approx 2^{n_{\text{O}} R_{\text{O}} \log q}$. □

Remark 2 It is possible to construct a variant where the bound on ϵ_{II} is independent of ϵ_{in}. This is achieved by adding an all-one and an all-zero column to the inner code, making C_{I} a $(q+1, q)$ code. The result would be that $E(V_i) = 0$ when $i = 1, q$ and i is innocent, and thus the ϵ_{in}/q term in the second argument of $D(\cdot||\cdot)$ disappears. When q is relatively large however, the effect of this is marginal.

Example 2 We use the same target values as in Example 1, and a $[2^{20}, 1]_{2^{20}}$ RS (or repetition) code. The Type I error rate is bounded as

$$\epsilon_{\text{I}} \leq \exp -1024 D\left(\frac{1+\Delta}{2} || \frac{21}{40} - \frac{\epsilon_{\text{in}}}{20}\right).$$

Setting the bound equal to 10^{-3} and solving, we get $\Delta \approx 0.04437$. Inserting this into the bound on Type II errors, we get $\epsilon_{\text{II}} \leq 0.34 \cdot 10^{-442}$. Using the shortened (or generalised) $[2^{15}, 1]_{2^{20}}$ RS code as outer code, we get $\epsilon_{\text{II}} \approx 0.10 \cdot 10^{-7}$. This gives a code of length $n \approx 2^{35}$.

In order to make asymptotic construction, we take $\Delta \approx (1 - 2p_1)/t$ just at we did for random codes. To get $\epsilon_{\text{II}} \to 0$ we require that $\alpha > \epsilon_{\text{in}}/q$ in Theorem 5, and that

$$R_{\text{O}} < \frac{1 - t(1-\delta)}{\log q} D\left(\frac{1}{2} + \frac{\alpha}{2} || \frac{1}{2} + \frac{\epsilon_{\text{in}}}{q}\right).$$

We use codes with $R_O \approx 1 - \delta - 1/(\sqrt{q} - 1)$. It follows that we can have outer code rate arbitrarily close to the R_{O} solving the following equation,

$$R_{\text{O}} = \frac{1 - t\left(R_{\text{O}} + \frac{1}{\sqrt{q}-1}\right)}{\log q} D\left(\frac{1}{2} + \frac{\alpha}{2} || \frac{1}{2} + \frac{\epsilon_{\text{in}}}{q}\right), \tag{5}$$
$$\alpha = \frac{1 - 2\epsilon_{\text{in}} - t^2\left(R_{\text{O}} + \frac{1}{\sqrt{q}-1}\right)}{t - t^2\left(R_{\text{O}} + \frac{1}{\sqrt{q}-1}\right)},$$
$$\frac{\epsilon_{\text{in}}}{q} < \frac{\alpha}{2}.$$

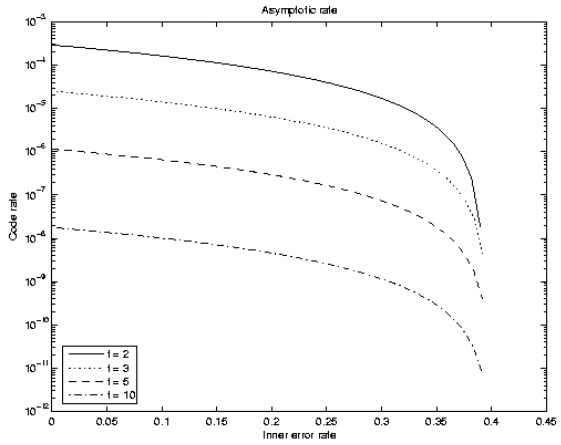


Fig. 4: Code rates for concatenated codes with BS inner codes and AG outer codes for $t = 2, 3, 5, 10$ and $q = 25t^4$.

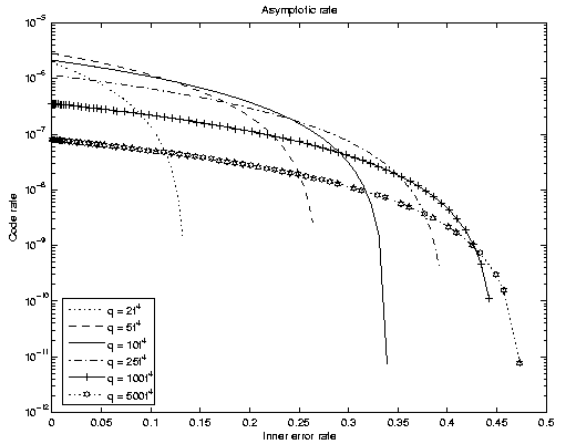


Fig. 5: Asymptotic code rates against $t = 5$ pirates for different alphabet sizes.

The total rate is $R_t(q) = R_{\mathrm{I}} \cdot R_{\mathrm{O}}$ where

$$R_{\mathrm{I}} = \frac{\log q}{q-1}.$$

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. We have calculated some sample rates by solving (5) by fix-point iteration. A graphical view on the asymptotic result is shown in Figures 4 and 5. The former shows performance for different numbers of pirates t, and the latter for 5 pirates with different alphabet sizes.

3.5 Comparison

The most notable codes for the Guth-Pfitzmann Marking Assumption are due to Guth-Pfitzmann (GP) [9], Muratani [15], and Yoshioka and Matsumoto [27]. Among these, only [9] offer theoretical error bounds. The other two give only experimental analyses.

Since our code takes advantage of soft decision decoding, it can, even at high error rates, use shorter codewords than Boneh and Shaw [4]. The GP code requires longer codewords than Boneh and Shaw did, and is thus clearly outperformed by our code.

As an example, we calculated the length for the GP code using the same parameters as in Example 1, and found $n \approx 5.9 \cdot 10^9$; more than a thousand times more than our code. Asymptotically, the GP code compares even less favourably, as it has $n = O((\log M)^2)$. In addition to the outer code being linear in $\log M$, the inner code of uses a replication factor $r = O(\log M)$.

Observe that the error bounds theoretically proved in this paper hold for *any attack* under the Marking Assumption with random errors. An experimental analysis can only lead to conclusions about the particular attacks tried in the experiment. A clever attacker may very well come up with a more efficient attack than the analyst could. Thus, theoretical analyses tend to give *upper* bounds on error probabilities, whereas experimental analyses give *lower* bounds. For critical security applications, upper bounds will usually be required by the specification. The same will be the case for forensic systems used as evidence in a criminal court. A lower bound will not limit the level of doubt, and thus will not allow conviction.

4 Experimental results

We have made a series of experiments to describe possible operation of the Error-Correcting Boneh-Shaw scheme.

We start by presenting simulations of the Boneh-Shaw scheme as an isolated black box for the FP and ECC layers. For comparison with [27], we use $M = 10\,000$ users in Section 4.1, and to compare with [11], we use $M = 1024$ in Section 4.2. In order better to match the scenario studied in [11], in Section 4.3 we introduce a variant of our scheme taking soft input to the decoder.

In Section 4.4, we demonstrate how the Boneh-Shaw scheme could be used in conjunction with a simple image WM scheme, and demonstrate its performance opposite the standard attacks used in similar analyses in the literature, such as averaging, cut-and-paste, and Gaussian noise.

Experimental tests show that the alphabet size should be kept relatively small to minimise the rate of decoding error for given M, n, and t. This is contrary to the asymptotic case, where q should be large, especially for high error rates. Unfortunately, the algebraic outer codes require very large alphabets. We did some tests with $M = 10\,000$ and $q = 101$, but we had almost 100% decoding error, and we were nowhere close to the performance of random codes in Section 4.1.

1. Generate a random $[n, M]$ Boneh-Shaw code C
2. Draw a set $P \subset C$ of t random fingerprints
3. Make a hybrid fingerprint $\mathbf{y}'$ from P, by dividing the code positions into t disjoint groups, and set all bits in group i equal to the fingerprint of the ith pirate.
4. Flip a fraction ϵ_{in} of the bits in $\mathbf{y}'$ uniformly at random to obtain $\mathbf{y}$.
5. Use $\mathbf{y}$ to trace one user $\mathbf{c}$, and note *success* if $\mathbf{c} \in P$ and *error* otherwise.

Table 1: Simulation procedure for the fingerprinting layer.

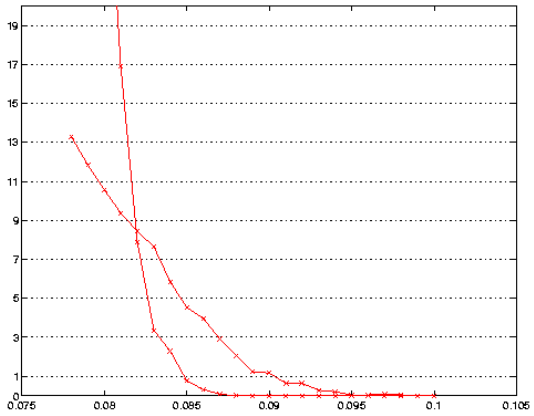


Fig. 6: Comparison of different thresholds at $\epsilon_{\text{in}} = 37\%$.

Still, algebraic outer codes may be an alternative for applications with more users. The key advantage is efficient decoding. For the small examples commonly considered in the literature , decoding efficiency is not likely to be an issue. With 10 000 users the calculation of a single sample (incl. code generation, encoding, and decoding) typically took 42s, using a crude Matlab implementation.

4.1 Simulations with 10 000 users

Yoshioka and Matsumoto [27] proposed a code which they analysed by simulations with $M = 10000$ users, $t = 15$ colluders, and a length of $n = 293\,000$. We present a similar simulation of our coding using similar parameters.

Experimentally, using $q = 15$ and $n_{\text{O}} = 20\,000$ give decent performance for the Boneh-Shaw scheme with random outer code. The total length is then $n = 280\,000$, which is slightly shorter than [27]. We use the average of 50 samples to calculate each data point throughout this subsection.

Yoshioka and Matsumoto used list decoding. They claim zero false positives throughout the experiments, using a sample size of 100 coalitions. The number of true positives was shown as a declining function in ϵ_{in}. They pointed to an error rate $\epsilon_{\text{in}} = 37\%$ as the limit where the average number of true positives was 1.

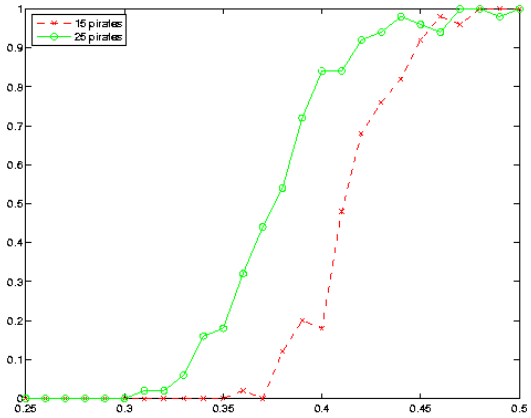


Fig. 7: Simulation results for closest neighbour decoding with $M = 10\,000$ users.

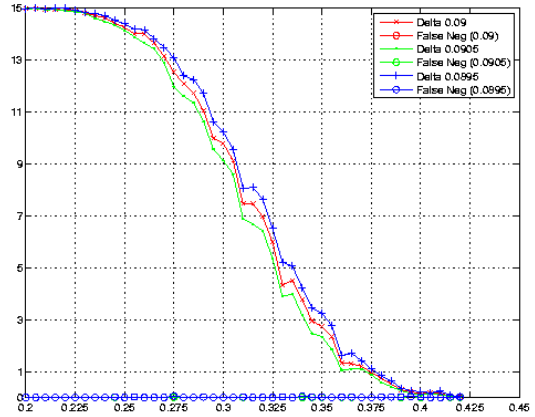


Fig. 8: Simulation results for list decoding with $M = 10\,000$ users.

Figure 6 shows a comparison of different threshold values Δ for our scheme, at an error rate of 37%. Using this plot we find that thresholds around 0.09 give $\epsilon_{\text{II}} \approx 0$.

Figures 7 and 8 show the performance of our scheme with closest neighbour and list decoding respectively. We observe the same breakpoint of $\epsilon_{\text{in}} \approx 37\%$ as [27], where the average number of correctly identified users is 1. However, contrary to [27], we detect all the colluders up to $\epsilon_{\text{in}} > 20\%$, whereas [27] reports detection of only 9 colluders on average for $\epsilon_{\text{in}} = 0$ and 7 users for $\epsilon_{\text{in}} = 20\%$. On the other hand, for $\epsilon_{\text{in}} > 40\%$, we detect almost no colluders whereas [27] does some of the time. We also observe that closest neighbour decoding has negligible error rate up to $\epsilon_{\text{in}} = 37\%$.

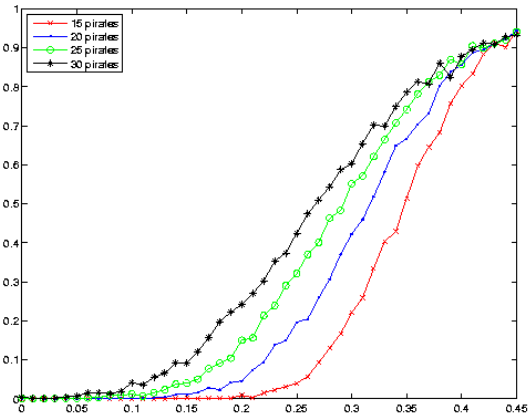


Fig. 9: Simulation results for $M = 1024$ users.

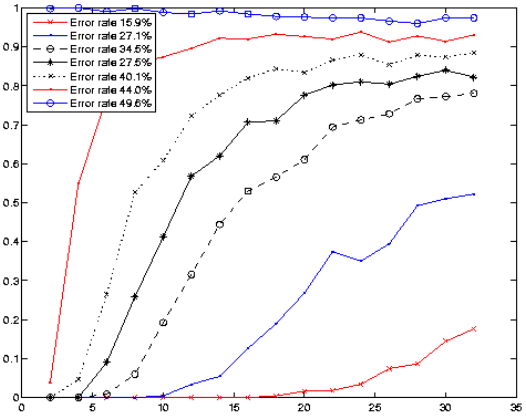


Fig. 10: Simulation results for $M = 1024$ users.

4.2 Simulations with 1024 users

In [11], the analysis was based on simulations using $M = 1024$ users and a codelength of $n = 30\,000$. Their watermarks are sequences of ± 1. In the experiments, they added the watermark to random Gaussian signals (simulating the host signal). The detector was non-blind, subtracting the the host signal before decoding using a correlation decoder. They made experiments with watermark-to-noise rations (WNR) ranging from -20dB to 0dB, but they did not specify the distribution of the noise.

Matching the parameters of [11], we use $q = 11$ and $n_{\mathrm{O}} = 3000$. Figures 9 and 10 show experimental performance for different collusion sizes and error rates.

In a practical scenario, we suppose that the binary codeword over $\{0, 1\}$ will be translated to a watermark $\mathbf{w}$ with symbols from $\{\pm 1\}$ by mapping $0 \mapsto -1$, which can be added to the host signal, possibly after multiplication by a watermarking strength α. We assume a non-blind hard-decision decoder, which decodes negative values to 0 and positive values to 1.

WNR	σ	ϵ_{in}
0dB	1	15.9%
-4dB	0.6310	27.1%
-8db	0.3981	34.5%
-12dB	0.2512	40.1%
-16dB	0.1585	44.0%
-20dB	0.1000	49.6%

Table 2: Error rate ϵ_{in} resulting from White Gaussian noise with standard deviation σ, and corresponding to given WNR.

Under White Gaussian noise, using hard decision decoding as suggested, the error rate ϵ_{in} generated at various WNR is given by Table 2. The table has been derived theoretically from the normal distribution. The curves in Figure 10 correspond to WNR from 0dB to -20dB in intervals of 4dB.

Experimental error rates are depicted in Figures 9 and 10. At $\epsilon_{\mathrm{in}} = 15.9\%$ (WNR 0dB) the performance is decent, but it is clear, as one would expect, that at higher noise level this scheme is not good enough. A hard-decision demodulator with $\epsilon_{\mathrm{in}} \approx 49.6\%$ would require an extremely long code; however, a soft-input decoder at a WNR of -20dB is feasible. Indeed, soft input is used in [11].

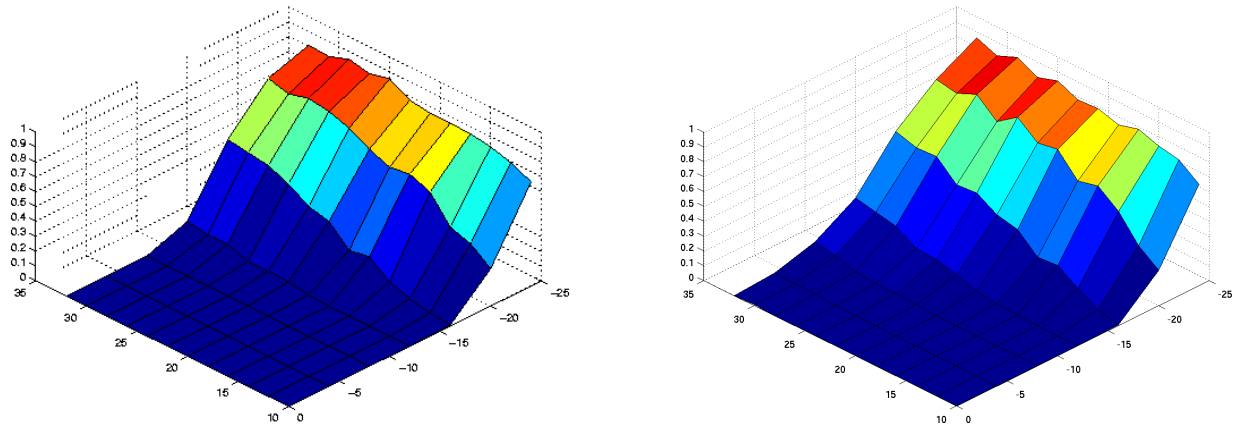
4.3 Soft input decoder

In order to match [11], we make an *ad hoc* modification to our system, to allow soft input to the decoder. This is relatively simple to do.

We use the same embedding as before, mapping $0 \mapsto -1$. The non-blind receiver will subtract the host signal, to get an observed, received sequence $\mathbf{r}'$. The input to the fingerprint decoder is $\mathbf{r} = (\mathbf{r}' + \mathbf{1})/2$ The only change introduced is that the decoder input, which used to be elements of $\{0, 1\}$, are now real numbers. Consider the heuristic $V_j = X_j - X_{j-1}$ from equation (1) used by the inner decoder. This is still well-defined and real for any real X_j.

We made a new simulation using this approach. The flip attack in Table 1, Step 4 was replaced by addition of Gaussian noise with $\mu = 0$ and varying σ. Results are shown in Figures 11, and 13a. We also tried with an averaging attack instead of cut-and-paste in Figures 12 and 13b.

Comparing the figures against the figures of [11], we see that both have negligible error rates high WNR (close to zero) and few colluders. And both become useless, with error rates above 50%, when both the WNR is low and there are many colluders. In between these two extremes, it can be observed that our scheme outperforms [11].



(a) Averaging attack

(b) Cut-and-paste attack

Fig. 13: Soft input decoding in presence of Gaussian noise.

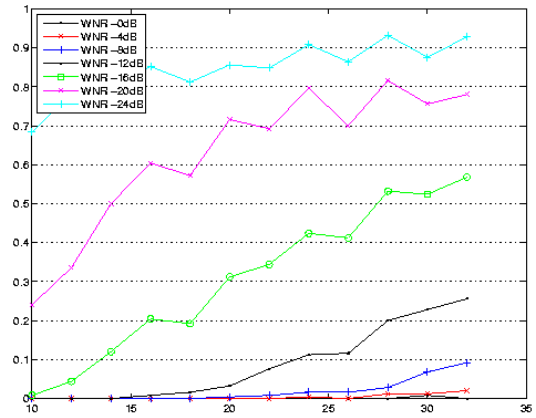


Fig. 11: Soft input decoding against the cut-and-paste attack and Gaussian noise.

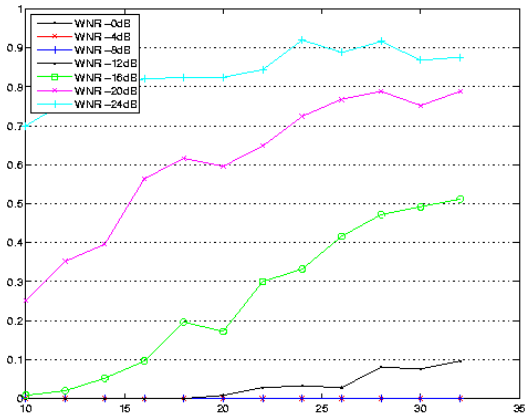


Fig. 12: Soft input decoding against the averaging attack and Gaussian noise.

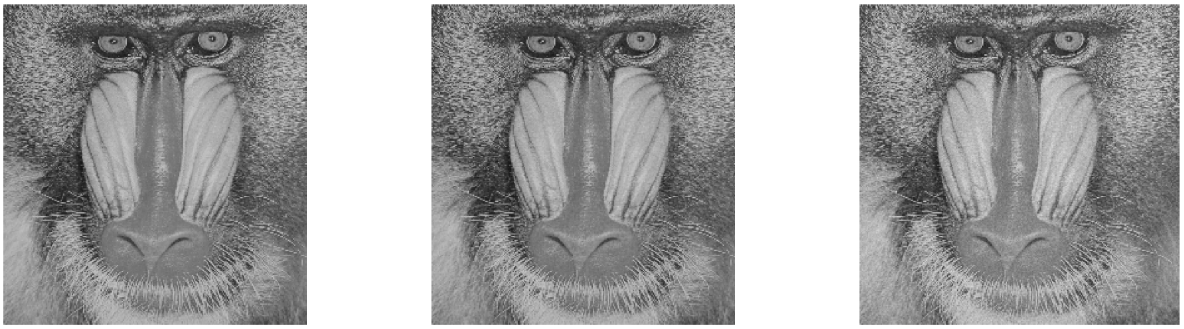
For the cut-and-paste attack, we have negligible error rate for $t = 32$ pirates at WNR 0db, whereas [11] dips sharply for $t \geq 28$ and this WNR, with an error rate of about 10% for $t = 28$ and 20% for $t = 30$. Against 10 pirates, we have negligible error rate down to WNR -16dB, whereas [11] has an error rate around 20%.

For the averaging attack, we see that for 10 colluders and WNR of -16dB, we have negligible error rate, whereas [11] has an error rate about 10%. For 30 pirates at WNR -12dB, we have an error rate of less than 10% against more than 40% in [11].

4.4 Image Watermarking

As a simple proof of concept, we use the embedding described in the previous subsection to fingerprint real images in the blockwise DCT domain. We use 30 mid-frequency coefficients from each block, and use a *single* coefficient to embed each code bit. Among the eligible coefficients, 30 000 were selected at random. The embedding strength is $\alpha = 4$. After embedding, the image was subjected to a cut-and-paste attack and random Gaussian noise in the spatial domain, with standard deviation $\sigma = 5\alpha = 20$.

We used three standard test images, in grayscale 256× 256. The resulting error rates are shown in Figure 15. One sample set of images is shown in Figure 14. The



(a) Original (b) Watermarked (c) Attacked

Fig. 14: Sample fingerprinted and attacked image. The watermarked image has PSNR 37.66dB, and the attacked one has PSNR 24.84dB.

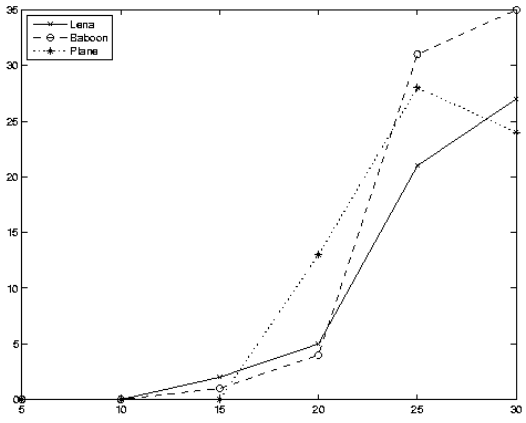


Fig. 15: Error rates soft-decision tracing with real images.

mean PSNR of the watermarked image was consistently 38.7dB (lena), 37.7dB (baboon), and 37.9dB (plane), with empirical variance less than 10^{-26}. The attacked image had PSNR 25.9dB (lena), 24.9dB (baboon), and 25.1dB (plane) with empirical variance less than $3 \cdot 10^{-4}$.

As our focus has been on the code development, we are unable to say whether this is the best embedding technique to use. Our intention is simply to demonstrate that decent results are possible using the code presented. To find optimal approaches, further research is needed from a watermarking point of view.

5 Conclusion and future research

The layered WM/FP models illustrate how fingerprinting and watermarking can be studied separately and combined as black boxes, *if* we have a clear and common understanding of the interface. Past works on fingerprinting for the Boneh-Shaw model have often suggested to use an underlying WM scheme, without comparing the assumptions about the interface. Similarly, watermarking works have referred to the Boneh-Shaw FP scheme without discussing the interface. The GP model [9] is more realistic for WM applications than Boneh-Shaw model [4], as it allows for errors created in the watermarking layer.

We propose two construction for the GP model. Algebraic outer codes are only practical for large parameters. The construction with random outer codes is the more flexible and was analysed experimentally and theoretically. We found a huge discrepancy between the theoretical and the experimental performance. This is not surprising; the theoretical analysis giving an upper bound on error probability, and the experimental analysis giving a lower bound. The theoretical upper bound is correct for any adversary attack, whereas the experimental estimate only consider specific attacks which may not be optimal.

It is unfortunate that other known systems from the literature generally have been presented either with an experimental or with a theoretical analysis; rarely with both. Theoretically, our system has a much better information rate than that of Guth-Pfitzmann [9]. Experimentally, it has similar performance to [27]; its main advantage being that a theoretical analysis is available as well.

We have given a simple example of how our system can be built into a WM scheme, and compared it with the joint WM/FP scheme of [11].

A major advantage of the code presented, is that the error-correction capability is strong enough so that we may not need any error-correction in the WM layer. This allows us to use a single signal sample for each code symbol. As a single sample cannot be subdivided for a cut-and-paste attack, such attack can no longer be mounted in the WM layer.

The present work raises a long list of open questions. Some of the most important ones are the following.

1. Can other collusion-secure codes for the Boneh-Shaw model be extended for the Guth-Pfitzmann model? The Tardos scheme [23] is particularly interesting for this study.
2. Is it possible to close the gap between the theoretical and experimental error rates? One approach to this question is to identify more effective attacks for use in the simulations.
3. Develop theoretical error bounds for
   - the collusion-secure code of [27].
   - our system with soft input from the WM layer (as in Section 4.3).
   - the joint WM/FP system of [11].
4. What is the optimal watermarking system to use in conjunction with out collusion-secure code?

It is also possible to adapt the collusion-secure code of [2] to handle random errors using the same techniques as in this paper, but this code is only workable for large M and thus could not be compared using the parameters in our experiments.

Acknowledgements The author is grateful for many useful discussions with dr. Stefan Katzenbeisser, dr. Marcel Fernandez, and prof. Gérard Cohen. Also, many thanks to dr. Xunzhan Zhu and dr. Pedro Comesaña, for access to Matlab code for watermarking; I learnt a lot from it. Finally, thanks to the anonymous referees for their useful comments.

The theoretical research in this paper was completed as an employee of the University of Bergen, supported by the Norwegian Research Council. The experimental research was completed in the author's present post at the University of Surrey.

References

1. International Intellectual Property Alliance, fact sheet. `http://www.iipa.com/aboutiipa.html`
2. Barg, A., Blakley, G.R., Kabatiansky, G.A.: Digital fingerprinting codes: Problem statements, constructions, identification of traitors. IEEE Trans. Inform. Theory **49**(4), 852–865 (2003)
3. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. In: Advances in Cryptology - CRYPTO'95, *Springer Lecture Notes in Computer Science*, vol. 963, pp. 452–465 (1995)
4. Boneh, D., Shaw, J.: Collusion-secure fingerprinting for digital data. IEEE Trans. Inform. Theory **44**(5), 1897–1905 (1998). Presented in part at CRYPTO'95
5. Chor, B., Fiat, A., Naor, M.: Tracing traitors. In: Advances in Cryptology - CRYPTO '94, *Springer Lecture Notes in Computer Science*, vol. 839, pp. 257–270. Springer-Verlag (1994)
6. Chor, B., Fiat, A., Naor, M., Pinkas, B.: Tracing traitors. IEEE Trans. Inform. Theory **46**(3), 893–910 (2000). Presented in part at CRYPTO'94
7. Cox, J., Miller, M., Bloom, J.: Digital Watermarking. Morgan Kaufmann (2002)
8. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometry codes. IEEE Trans. Inform. Theory **45**(6), 1757–1767 (1999)
9. Guth, H.J., Pfitzmann, B.: Error- and collusion-secure fingerprinting for digital data. In: Information Hiding '99, Proceedings, *Springer Lecture Notes in Computer Science*, vol. 1768, pp. 134–145. Springer-Verlag (2000)
10. Hagerup, T., Rüb, C.: A guided tour of Chernoff bounds. Information Processing Letters **33**, 305–308 (1990)
11. He, S., Wu, M.: Joint coding and embedding techniques for multimedia fingerprinting. IEEE Trans. Information Forensics and Security **1**, 231–248 (2006)
12. Kerchoffs, A.: La cryptographie militaire. Journal des sciences militaires **IX**, 5–38 (1883)
13. Koetter, R., Vardy, A.: Algebraic soft-decision decoding of Reed-Solomon codes. IEEE Trans. Inform. Theory **49**(11), 2809–2825 (2003)
14. MacWilliams, F.J., Sloane, N.J.A.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
15. Muratani, H.: A collusion-secure fingerprinting code reduced by Chinese remaindering and its random-error resilience. In: I.S.M. (Ed.) (ed.) Information Hiding 2001, *Springer Lecture Notes in Computer Science*, vol. 2137, pp. 303–315 (2001)
16. Muratani, H.: Optimization and evaluation of randomized c-secure CRT code defined on polynomial ring. In: J.F. (Ed.) (ed.) Information Hiding 2004, *Springer Lecture Notes in Computer Science*, vol. 3200, pp. 282–292 (2004)
17. Safavi-Naini, R., Wang, Y.: Traitor tracing for shortened and corrupted fingerprints. In: Digital rights management, *Springer Lecture Notes in Computer Science*, vol. 2696. Springer-Verlag (2002)
18. Schaathun, H.G.: The Boneh-Shaw fingerprinting scheme is better than we thought. Tech. Rep. 256, Dept. of Informatics, University of Bergen (2003). Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`
19. Schaathun, H.G.: Binary collusion-secure codes: Comparison and improvements. Tech. Rep. 275, Dept. of Informatics, University of Bergen (2004). Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`
20. Schaathun, H.G.: The boneh-shaw fingerprinting scheme is better than we thought. IEEE Transaction on Information Forensics and Security (2006)
21. Schaathun, H.G., Fernandez-Muñoz, M.: Boneh-Shaw fingerprinting and soft decision decoding. In: Information Theory Workshop (2005). Rotorua, NZ
22. Schaathun, H.G., Fernandez-Muñoz, M.: Soft decision decoding of boneh-shaw fingerprinting codes. IEICE Transactions (2006). To appear.
23. Tardos, G.: Optimal probabilistic fingerprint codes. Journal of the ACM (2005). `http://www.renyi.hu/~tardos/fingerprint.ps`. To appear. In part at STOC'03.
24. Tsfasman, M.A.: Algebraic-geometric codes and asymptotic problems. Discrete Appl. Math. **33**(1-3), 241–256 (1991). Applied algebra, algebraic algorithms, and error-correcting codes (Toulouse, 1989)
25. Wagner, N.R.: Fingerprinting. In: Proceedings of the 1983 Symposium on Security and Privacy (1983)
26. Wu, M., Trappe, W., Wang, Z.J., Liu, K.J.R.: Collusion resistant fingerprinting for multimedia. IEEE Signal Processing Magazine (2004)
27. Yoshioka, K., Matsumoto, T.: Random-error resilience of a short collusion-secure code. IEICE Trans. Fundamentals **E86-A**(5) (2003)