

**REPORTS
IN
INFORMATICS**

ISSN 0333-3590

**Boneh-Shaw fingerprinting and soft decision
decoding**

**Hans Georg Schaathun
Marcel Fernandez-Muñoz**

REPORT NO 289

January 2005



Department of Informatics
UNIVERSITY OF BERGEN
Bergen, Norway

This report has URL

<http://www.ii.uib.no/publikasjoner/texrap/pdf/2005-289.pdf>

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at <http://www.ii.uib.no/publikasjoner/texrap/>.

Requests for paper copies of this report can be sent to:

**Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway**

Boneh-Shaw fingerprinting and soft decision decoding

Hans Georg Schaathun* and Marcel Fernandez-Muñoz†

21st January 2005

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. The most well-known collusion-secure code is due to Boneh and Shaw (1995/98). In this paper we improve the decoding algorithm by using soft output from the inner decoder, and we show that this permits using significantly shorter codewords.

Keywords

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

1 Introduction

The problem of digital fingerprinting was introduced in [12], studied in [2], and given increasing attention following [3, 4]. A vendor selling digital copies of copyrighted material wants to prevent illegal copying. Digital fingerprinting is supposed to make it possible to trace the guilty user (pirate) when an illegal copy is found. This is done by embedding a secret identification mark, called a fingerprint, in each copy, making every copy unique.

The fingerprint must be embedded in such a way that it does not disturb the information in the data file too much. It must also be impossible for the user to remove or damage the fingerprint, without damaging the information contents beyond any practical use. In particular, the fingerprint must survive any change of file format (e.g. gif

*Dr. Schaathun is with the Selmer Centre, University of Bergen. Email: <georg@ii.uib.no>.

†Dr. Fernandez is with the the Telecommunications Engineering School, Universitat Politecnica de Catalunya. Email: <marcelf@mat.upc.es>.

to tiff) and any reasonable compression including lossy compression. This embedding problem is essentially the same as the problem of watermarking.

If a single pirate distributes unauthorised copies, they will carry his fingerprint. If the vendor discovers the illegal copies he can trace them back to the pirate and prosecute him. If several pirates collude, they can to some extent tamper with the fingerprint. When they compare their copies they see some bits (or symbols) which differ and thus must be part of the fingerprint. Identified bits may be changed, and thus the pirates create a hybrid copy with a false fingerprint. A collusion-secure code is a set of fingerprints which enables the vendor to trace pirates even when they collude, given that there are no more than t pirates for some threshold t .

Collusion-secure coding is also employed in traitor tracing [5, 6]. Whereas fingerprinting protects the digital data in themselves, traitor tracing protects broadcast encryption keys.

A collusion-secure code can be deterministic or probabilistic. In the first case we demand that we can always find a guilty user when at most t users have colluded. In the latter case, we are satisfied if the vendor is able to trace a pirate with probability at least $1 - \epsilon$ for some small error rate ϵ . Deterministic schemes are possible only over large alphabets.

The most well-known collusion-secure code is the probabilistic scheme due to Boneh and Shaw [3, 4]. A handful of other schemes have also appeared over the years; see [11] for an overview. A new analysis of the error probability for the Boneh-Shaw scheme was made in [10], showing that the codewords could be made much shorter than initially assumed. In this paper, we make further improvement by using soft output from the inner decoding. The error analysis follows the approach of [10]. The major novelty of this paper is to find a good output parameter from the inner decoding. Soft decision decoding has also previously been applied for other collusion-secure codes, in [7] among others.

2 On collusion-secure codes

2.1 Some coding theory

We use notation and terminology from coding theory. The set of fingerprints is an $(n, M)_q$ code, which provides for up to M buyers, uses an alphabet of q symbols, and requires n such symbols embedded in the digital file. The code book is the matrix formed by taking the codewords (fingerprints) as rows. The Hamming distance $d(\mathbf{x}, \mathbf{y})$ between two words \mathbf{x} and \mathbf{y} is the number of positions where the two words differ, and the minimum distance of a code C is denoted $d(C)$ or just d . The normalised minimum distance is $\delta = d/n$. The rate of the code is $R = (\log M)/n$.

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

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

Definition 1 (Concatenation)

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

Concatenated codes are often decoded by first decoding each block using some decoding algorithm for the inner code, so that a word of symbols from the outer code alphabet is obtained. This word can finally be decoded with a decoding algorithm designed for the outer code.

2.2 The fingerprinting problem

To understand the fingerprinting problem, we must know what the pirates are allowed to do. This is defined by the Marking Assumption.

Definition 2 (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

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

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

The Marking Assumption defines the requirements for the embedding of the fingerprint in the digital data. Constructing appropriate embeddings is non-trivial, though it is not theoretically impossible [4]. Alternative assumptions have been proposed, and some overview of this can be found in [1].

A *tracing algorithm* for the code C is any algorithm A which takes a vector \mathbf{x} as input and outputs a set $L \subseteq C$. If \mathbf{x} is a false fingerprint produced by some coalition $P \subseteq C$, then A is successful if L is a non-empty subset of P . We say that we have an error of Type I if $L \cap P = \emptyset$ and an error of Type II if $L \setminus P \neq \emptyset$. A Type I error means that we do not find any guilty pirate, whereas Type II means accusing an innocent user. Let ϵ_1 and ϵ_2 denote the probabilities of Type I and Type II errors respectively. Given our juridical system, Type II is clearly a graver error than Type I, so we might accept ϵ_1 higher than we can accept ϵ_2 .

A code is said to be combinatorially t -secure if it has a tracing algorithm which succeeds with probability 1 when there are at most t pirates. It is said to be t -secure with ϵ -error if the probability of error (of either type) is at most ϵ when there are at most t pirates.

A binary fingerprinting scheme consists of a binary (n, M) code C , a tracing algorithm A , and a bijection ι between C and the set of users. The tracing algorithm A is

public information. The code C may be secret information, but it is drawn at random from some probability distribution which is publicly known. The mapping ι may be secret or public. The ensemble of secret information is called the *key*.

Our challenge is, for a given number of users M and a maximum number of pirates t , to find a code with the shortest possible length n and the best possible error rate ϵ . It is also advantageous if the tracing algorithm A is efficient.

3 The code

3.1 On the inner code

The Boneh-Shaw code is a concatenated code. The inner code will be called BS-RS (Boneh-Shaw replication scheme); 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 zeroes in the first $i-1$ types and a one in the rest. We can see that unless user i is a pirate, the pirates cannot distinguish between the $(i-1)$ -th and the i -th type. Hence they have to use the same probability of choosing a 1 for both these types. If r is large enough we can use statistics to test the null hypothesis that user i be innocent. The output is a list of users for which the null hypothesis may be rejected.

Theorem 1 (Boneh and Shaw)

The BS-RS 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.

Our idea is to return soft information, i.e. a function of $F_i - F_{i-1}$, to be used by the outer decoding algorithm. We have played with many variants, but most of them have been very difficult to work through the error analysis. The following might not be optimal, but it does work well. The output is a vector $\mathbf{v} = (v_1, \dots, v_q)$, given as

$$v_j = \frac{F_j - F_{j-1}}{r}. \quad (1)$$

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 $E(v_j) = 0$.

3.2 On the outer code

Boneh and Shaw suggested to concatenate the inner code with a random q -ary code which would be decoded using closest neighbour decoding. In the improved error analysis [10], this was replaced by list decoding, returning all codewords within a certain distance of the hybrid word after inner decoding. It has also been suggested to use codes with large minimum distance, typically AG codes, but unfortunately such codes need much larger alphabets than random codes, and the inner Boneh-Shaw code is bad in that case, having length linear in q . We will return to outer codes with a large distance in a later section.

After inner decoding of all the blocks, 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. We employ the common assumption that the pirates make independent decisions in each column, such that $F_i \sim B(r, p_i)$ for some probability p_i .

The outer decoding algorithm takes the $q \times n$ reliability matrix R as input and returns all codewords $\mathbf{c} = (c_1, \dots, c_n)$ that satisfy

$$W(\mathbf{c}) = \sum_{i=1}^n r_{i,c_i} \geq \Delta n. \quad (2)$$

We call $W(\mathbf{c})$ the weight of \mathbf{c} . The decoding can always be made in time $O(n \cdot M)$.

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, \dots, 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

$$P\left(\sum_{i=1}^t X_i \leq t\delta\right) \leq 2^{-tD(\delta||x)}, \quad \text{when } \delta < x, \quad (3)$$

$$P\left(\sum_{i=1}^t X_i \geq t\delta\right) \leq 2^{-tD(\delta||x)}, \quad \text{when } \delta > x, \quad (4)$$

where

$$D(\sigma||p) = \sigma \log \frac{\sigma}{p} + (1 - \sigma) \log \frac{1 - \sigma}{1 - p}. \quad (5)$$

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

4 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. The probability of failing to accuse any guilty user is independent of the outer code used. We study this error probability in the first subsection. The probability of accusing innocent users does depend on the outer code, and this will be studied for random codes in Section 4.2 and for codes with large distance in Section 4.3.

4.1 Probability of failing

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

$$\epsilon_I \leq P\left(\frac{1}{t} \sum_{i=1}^n \sum_{c \in P} r_{i,c_i} \leq \Delta n\right) = P\left(\sum_{i=1}^n Y_i \leq \Delta n\right). \quad (6)$$

where

$$Y_i = \sum_{c \in P} \frac{r_{i,c_i}}{t} = \frac{1}{t} \sum_{c \in P} \frac{F_{c_i} - F_{c_i-1}}{r}. \quad (7)$$

Obviously

$$\sum_{\gamma \in Q} \frac{F_\gamma - F_{\gamma-1}}{r} = 1, \quad (8)$$

and $E(F_\gamma - F_{\gamma-1}) = 0$ when γ is not seen by the pirates. Hence we get $E(Y_i) = 1/t$. Observe that $-1 \leq Y_i \leq 1$. In order to get a stochastic variable in the range $[0, 1]$, we set $X_i = (1 + Y_i)/2$. Thus

$$E(X_i) = \bar{x} = \frac{t+1}{2t}, \quad (9)$$

and we get

$$\epsilon_I \leq P\left(\sum_{i=1}^n X_i \leq \frac{1+\Delta}{2}n\right). \quad (10)$$

If $1/t > \Delta$, the Chernoff bound is applicable, giving the following theorem.

Theorem 3

Using the concatenated code with a BS-RS inner code and soft input list decoding with threshold $\Delta < 1/t$ for the outer code, the probability of failing to accuse any innocent user is given as

$$\epsilon_I \leq 2^{-nE}, \text{ where } E = D\left(\frac{1+\Delta}{2} \parallel \frac{t+1}{2t}\right). \quad (11)$$

This bound is independent of the choice of outer code.

4.2 Random outer codes

In this section we study a concatenated code, where the outer code is constructed by drawing each symbol for each codeword independently and uniformly at random from the alphabet. This random code is kept secret by the vendor. The bound on ϵ_I from Theorem 3 is still valid. In this section we bound ϵ_{II} .

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

$$\pi(\mathbf{c}) = P\left(\sum_{i=1}^n r_{i,c_i} \geq \Delta n\right). \quad (12)$$

Clearly $E(r_{i,c_i}) = 1/q$. Like in the last section, we make a stochastic variable in the $[0, 1]$ range,

$$X_i = \frac{1 + r_{i,c_i}}{2}, \quad (13)$$

$$E(X_i) = \frac{q+1}{2q}, \quad (14)$$

and

$$\pi(\mathbf{c}) = P\left(\sum_{i=1}^n X_i \geq \frac{1+\Delta}{2}n\right). \quad (15)$$

Theorem 4

Concatenating a $(r(q-1), q)$ BS-RS 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

$$\epsilon_{II} \leq 2^{(R_0 \log q - E)n}, \text{ where } E = D\left(\frac{1+\Delta}{2} \parallel \frac{q+1}{2q}\right). \quad (16)$$

Interestingly, both the bounds on ϵ_I and ϵ_{II} are independent of r , and hence we are going to choose $r = 1$ to minimise the length.

In Table 1, we show some constructions of RS-RC-Soft. The parameters have been found by trial and error, and cannot be expected to be optimal. However, major improvements appear to be impossible.

Theorem 5

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

$$R_t \approx \frac{D\left(\frac{t+1}{2t} \parallel \frac{q+1}{2q}\right)}{q-1}.$$

$\log M$	t	q	Δ	n_O	n	ϵ
10	10	35	0.06667	42400	1 441 600	$9.09 \cdot 10^{-11}$
10	50	153	0.01370	1 200 000	1 824 000 000	$8.84 \cdot 10^{-11}$
10	32	100	0.02128	480 000	4 752 000 000	$9.88 \cdot 10^{-11}$
20	20	66	0.03448	200 000	13 000 000	$9.39 \cdot 10^{-11}$
20	100	335	0.006897	4 918 000	1 642 612 000	$9.97 \cdot 10^{-11}$
20	1000	3350	0.0006897	$4.95 \cdot 10^8$	$1.657755 \cdot 10^{12}$	$8.75 \cdot 10^{-11}$
30	30	99	0.02353	494 000	48 412 000	$9.61 \cdot 10^{-11}$
30	150	500	0.004701	12 230 000	6 102 770 000	$9.99 \cdot 10^{-11}$

Table 1: Some constructions with random outer codes.

Proof: For asymptotic codes, $\epsilon_I \rightarrow 0$ if $\Delta < 1/t$, so we can take $\Delta \approx 1/t$. Likewise, $\epsilon_{II} \rightarrow 0$ if $\Delta > 1/q$ and

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

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

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 $R_I = \Omega(t^{-3})$. The only scheme with $R_I = \Omega(t^{-3})$ is the Tardos scheme with $R_I = \Theta(t^{-2})$, but that scheme is subject to adverse selection. Table 4 presents asymptotic rates for some constructions against few pirates.

4.3 Outer code with large distance

Suppose now that we use an outer code with large minimum distance δ , typically a Reed-Solomon (RS) code in the finite case or an algebraic geometry (AG) code asymptotically. We want to bound the probability of accusing \mathbf{c} when \mathbf{c} is innocent, i.e. to bound the probability

$$\pi(\mathbf{c}) \leq P\left(\sum_{i=1}^n r_{i,c_i} \geq \Delta n\right). \quad (17)$$

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, \dots, i_N , where $r_{i,j} = v_j$ is given by (1) with $F_j \sim F_{j-1}$.

Thus we get

$$\pi(\mathbf{c}) \leq P \left(\sum_{j=1}^N r_{i_j, c_{i_j}} \geq \tau N \right), \quad (18)$$

$$N = [1 - t(1 - \delta)]n, \quad (19)$$

$$\tau = \frac{\Delta - t(1 - \delta)}{1 - t(1 - \delta)}. \quad (20)$$

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

When $F_j \sim F_{j-1}$, we have $E(F_j - F_{j-1}) = 0$. Setting $Y_j = (1 + r_{i_j, c_{i_j}})/2$, we get $E(Y_j) = 1/2$ and

$$\pi(\mathbf{c}) \leq P \left(\sum_{j=1}^N Y_j \geq \frac{1 + \tau}{2} N \right), \quad (21)$$

$$(22)$$

This results in the following theorem.

Theorem 6

Concatenating a $(r(q-1), q)$ BS-RS code with a $(n, 2^{R_0 n}, \delta n)$ outer code using soft input list decoding with threshold Δ for the outer code, the probability of accusing an innocent user is given as

$$\epsilon_{\text{II}} \leq 2^{(R_0 \log q - [1 - t(1 - \delta)]D(\sigma \| 1/2))n}, \quad (23)$$

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

$$\sigma = \frac{1}{2} + \frac{\Delta - t(1 - \delta)}{2(1 - t(1 - \delta))}. \quad (24)$$

Again, we see that the error rate is independent on r , so $r = 1$ for maximum rate. In Table 2, we show some good constructions of RS-RS-SD. These lengths are better than those of RS-RC, but RS-RC appears to be better for large t . In particular, RS-RS-SD requires $q > t^2$. This is also illustrated by the fact that we got a shorter length for $t = 100$ when $M = 2^{30}$ than when $M = 2^{20}$ in the table.

In Table 3, we compare lengths for the different variants. We observe that random codes provide the shortest lengths, but for moderate t , even Reed-Solomon codes with soft decision beat the old random codes with hard decision.

4.3.1 Asymptotic codes

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

$$R_0 < \frac{1 - t(1 - \delta)}{\log q} D(\sigma \| 1/2). \quad (25)$$

$\log M$	t	$[n_O, k_O]$	m	Δ	n	ϵ
10	10	[1024, 1]	22	0.053	23046144	$0.31 \cdot 10^{-10}$
20	20	[1024, 2]	252	0.0364	263983104	$0.98 \cdot 10^{-10}$
20	100	$[2^{20}, 1]$	3	0.006	3298531737600	$0.12 \cdot 10^{-10}$
30	30	$[2^{15}, 2]$	8	0.02	8589672448	$0.76 \cdot 10^{-10}$
30	100	$[2^{15}, 2]$	169	0.00707	181456830464	$0.83 \cdot 10^{-10}$
30	150	$[2^{15}, 2]$	1861	0.005781	1998172553216	$1.0 \cdot 10^{-10}$

Table 2: Some good finite constructions with Reed-Solomon outer codes ($q = n_O$). To get the desired error rate, the code is also concatenated with an $[m, 1]$ repetition code.

		Hard dec. [10]	Random codes	Large distance
$\log M$	t	n	n	n
10	10	306548964	1441600	23046144
10	50	$0.233 \cdot 10^{12}$	182400000	
10	32	$0.265 \cdot 10^{11}$	47520000	
20	20	$6.44 \cdot 10^9$	13000000	260840448
20	100	$5.10 \cdot 10^{12}$	1642612000	3298531737600
20	1000	$7.02 \cdot 10^{16}$	$1.657755 \cdot 10^{12}$	
30	30	$4.09 \cdot 10^{10}$	48412000	8589672448
30	150	$1.38 \cdot 10^{23}$	6102770000	1991730298880

Table 3: Comparison of finite constructions of the two new schemes and the original Boneh-Shaw code with improved error analysis.

	Random codes		AG codes		Old record
t	q	Rate	q	Rate	Rate
2	5	0.0180	9^2	$6.79 \cdot 10^{-4}$	0.0688 [10]
3	8	0.00466	19^2	$6.14 \cdot 10^{-5}$	0.000638 [1]
4	11	0.00187	32^2	$1.10 \cdot 10^{-5}$	
5	14	0.000930	49^2	$2.89 \cdot 10^{-6}$	

Table 4: Asymptotic rates for some constructions of RS-AG with soft decision decoding.

Using AG codes with

$$R = 1 - \delta - \frac{1}{\sqrt{q}-1}, \quad (26)$$

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

$$R_O = \frac{1-t \left(R_O + \frac{1}{\sqrt{q}-1} \right)}{\log q} D \left(\frac{1}{2} + \frac{1}{2} \cdot \frac{1-t^2 \left(R_O + \frac{1}{\sqrt{q}-1} \right)}{t-t^2 \left(R_O + \frac{1}{\sqrt{q}-1} \right)} \middle| \middle| \frac{1}{2} \right), \quad (27)$$

$$0 < \frac{1-t^2 \left(R_O + \frac{1}{\sqrt{q}-1} \right)}{t-t^2 \left(R_O + \frac{1}{\sqrt{q}-1} \right)}. \quad (28)$$

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

$$R_I = \frac{\log q}{q-1}. \quad (29)$$

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 computed some asymptotic rates in Table 4, by choosing q by trial and error, and solving (27) by fix point iteration.

5 On complexity, conclusions, open problems

We have constructed a new collusion-secure coding scheme with extremely good rates. The only existing scheme with comparable or better rates is the Tardos scheme, which is unfortunately subject to adverse selection (see [11]).

Our decoding algorithms have complexity $O(M \log M)$, since the weight has to be calculated and compared for each codeword. This complexity is typical for collusion-secure codes. The only schemes with better complexity are those using Guruswami-Sudan decoding for the outer code.

The inspiration for this article started with an attempt to use Kötter-Vardy (KV) decoding [9], which is a soft-input variant of Guruswami-Sudan. At present we are not sure if this algorithm can be used. Firstly, we would have to change the reliability matrix to get non-negative entries. For instance, we might use $R' = [r'_{i,j}]$, where $r'_{i,j} = (1 + r_{i,j})/2$, and compare $W(\mathbf{c})$ to the threshold $\Delta'n$ where $\Delta = (1 + \Delta)/2$. The problem is that KV decoding uses a decoding threshold of $\sqrt{1-\delta}\sqrt{n}\|R\| + \epsilon$. Since $\|R\|$ depends on all the columns, it is not possible to use the Chernoff bound in a simple way. We leave the complicated ways as an open problem.

References

- [1] A. Barg, G. R. Blakley, and G. A. Kabatiansky. Digital fingerprinting codes: Problem statements, constructions, identification of traitors. *IEEE Trans. Inform. Theory*, 49(4):852–865, April 2003. 2.2, 4.3.1
- [2] G. R. Blakley, C. Meadows, and G. B. Purdy. Fingerprinting long forgiving messages. In *Advances in cryptology—CRYPTO '85 (Santa Barbara, Calif., 1985)*, volume 218 of *Lecture Notes in Comput. Sci.*, pages 180–189. Springer, Berlin, 1986. 1
- [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. 1
- [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. 1, 2.2
- [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. 1
- [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. 1
- [7] Marcel Fernández and M. Soriano. Fingerprinting concatenated codes with efficient identification. In *Information Security (ISC'02)*, volume 2433 of *Springer Lecture Notes in Computer Science*, pages 459–470, 2002. 1
- [8] Torben Hagerup and Christine Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1990. 3.2
- [9] Ralf Koetter and Alexander Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 49(11):2809–2825, 2003. 5
- [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/>. 1, 3.2, 4.3, 4.3.1
- [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/>. 1, 5

- [12] Neal R. Wagner. Fingerprinting. In *Proceedings of the 1983 Symposium on Security and Privacy*, 1983. 1