REPORTS
IN
INFORMATICS

ISSN 0333-3590

Binary Collusion-Secure Codes:
Comparison and Improvements

Hans Georg Schaathun

REPORT NO 275 **June 2004**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL http://www.ii.uib.no/publikasjoner/texrap/pdf/2004-275.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

Binary Collusion-Secure Codes: Comparison and Improvements

Hans Georg Schaathun

8th June 2004

Abstract

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

Quite a few collusion-secure codes have been proposed over the past few years, but since the authors have addressed different settings, it is hard to compare the schemes. In this paper we study and compare how existing schemes behave both asymptotically in the number of users, asymptotically in the number of pirates, and for practical parameters with from a thousand to a billion users.

For the Boneh and Shaw scheme, we make a new error analysis, proving that their scheme is in fact much better than originally assumed. We also point out a problem of adverse selection to which schemes by Tardos, and by Le, Burmester, and Hu appears to be vulnerable.

Keywords

collusion-secure codes, digital fingerprinting, copyright protection, traitor tracing, separating codes

1 Introduction

The problem of digital fingerprinting was introduced in [Wag83], studied in [BMP86], and given increasing attention following [BS95, BS98]. A vendor selling digital copies of copyrighted material wants to prevent illegal copying. Digital fingerprinting is supposed to make it possible to trace the guilty user (pirate) when an illegal copy is found. This is done by embedding a secret identification mark, called a fingerprint, in each copy, making every copy unique.

The fingerprint must be embedded in such a way that it does not disturb the information in the data file too much. It must also be impossible for the user to remove or damage the fingerprint, without damaging the information contents beyond any practical use. In particular, the fingerprint must survive any change of file format (e.g. gif to tiff) and any reasonable compression including lossy compression. This embedding problem is essentially the same as the problem 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. Collusion-secure coding is required to enable to trace at least one pirates where a coalition of pirates have colluded.

Collusion-secure coding is also employed in traitor tracing [CFN94, CFNP00]. Whereas fingerprinting protects the digital data in themselves, traitor tracing protects broadcast encryption keys. Other important variants of the problems are dynamic traitor tracing (e.g. [SNW00]) and anonymous fingerprinting [PW97].

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

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

We also make a new error-analysis to show that the Boneh and Shaw scheme from [BS95, BS98] is better than previously assumed. In particular the Boneh-Shaw

scheme yields asymptotic classes of codes with positive rate and exponentially decreasing error rate, a property first proved for the BBK scheme [BBK03]. We also introduce a couple of new schemes, in a sense variants of the Boneh-Shaw scheme, based on the new error analysis. We get particularly good improvements in the two-pirate case.

1.1 Acknowledgements

The author has had great benefit from discussions with colleagues at Mathematics Department at the Royal Holloway, University of London, and at Selmersenteret, University of Bergen. The work was supported by the Norwegian Research Council under Grant 146874/20.

2 Preliminaries

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

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

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

Definition 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.

For the error analysis, we will use the well known Chernoff bound as given in the following theorem. See e.g. [HR90] for a proof. The relative entropy function is defined as

$$D(\sigma||p) = \sigma \log \frac{\sigma}{p} + (1-\sigma) \log \frac{1-\sigma}{1-p}. \tag{2.1}$$

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

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

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

Another useful concept for collusion-secure codes is separating codes. Such codes have been applied in various fields for more than three decades, see [Sag94] for a survey.

Definition 2
A (t,u)-separating code or (t,u)-SS has the property for any two disjoint sets T and U of respectively t and u codewords, there is at least one coordinate position where every codeword of T is di*ff*erent from any codeword of U.

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

We will also use the separating weight $\theta(T,U)$ which is the number of positions separating T and U. The (t,u)-separating weight $\theta_{t,u}$ of the code C is such that $\theta(T,U) \geq \theta_{t,u}$ for any disjoint $T,U \subseteq C$ of sizes t and u.

2.1 The fingerprinting game

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

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

The game proceeds in the following steps.

1. The vendor chooses the fingerprinting scheme (C_K, A_K) to use for the product he is selling; this is the vendor strategy.
2. The key K is chosen at random.
3. The copies of the digital data are generated using the fingerprinting scheme and the key, and distributed to the users.
4. A coalition of potential pirates get together and compare their copies. At this stage they are allowed to opt out of the game, and refrain from illegal copying and distribution.
5. If the pirates choose to play, they choose a strategy for garbling the fingerprint, make the copies, and sell the copies with the false fingerprint.
6. If and when an illegal copy is discovered, the vendor runs the tracing algorithm A and prosecutes any users traced.

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

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

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

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

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

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

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

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

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

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

where only an a priori error probability is stated, i.e. [LBH03, Tar03]. Still it may be possible to extend the error analysis for these schemes and prove that the probability of a pirate coalition seeing a dangerously low a posteriori error probability is negligible. This is a question for future research.

2.2 The marking assumption

The fingerprinting system must include some method to embed the fingerprints in the digital data, in addition to the fingerprinting code and tracing algorithm briefly described above. Some theoretical embeddings are suggested in [BS98]. We will base our collusion-secure codes on the following Marking Assumption. Alternative assumptions have been proposed, and some overview of this can be found in [BBK03]. The present one appears to be dominant in the literature.

Definition 3 (The Marking Assumption)
Let $P \subseteq C$ be the set of fingerprints held by a coalition of pirates. The pirates can produce a copy with a false fingerprint $\mathbf{x}$ for any $\mathbf{x} \in F_C(P)$, where

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

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

There is an example of a simple and comprehensible embedding in the traitor tracing setting [CFNP00]. The system uses a $q \times n$ matrix of permanent keys $K_{j,i}$. Each row corresponds to an alphabet symbol and each column to a coordinate position. The user with fingerprint $(a_1, \ldots, a_n)$ receives the key $K_{a_i,i}$. The session key is the exclusive or of n elements s_1 to s_n. An enabling block is transmitted at the start of each session consisting of $e_{K_{j,i}}(s_i)$ for each i and j, where e_K is the encryption function for key K. To get the session key, one key from each column of the matrix is required, and that is exactly what each user has. When the pirates make a pirate decoder box, they must supply it with a key for each coordinate position from one of their true fingerprints, and thus the marking assumption is satisfied.

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

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

It is customary in the literature to assume *column-independent strategies*. In this case a probability $p_{\mathbf{x}} \in [0,1]$ is chosen for column type $\mathbf{x}$, and independently for each column of the type the bit matching the first pirate is chosen with probability $p_{\mathbf{x}}$ and the opposite symbol is chosen with probability $1-p_{\mathbf{x}}$. Clearly, by the law of large numbers, if the number of columns of each type is moderate, then the column-independent strategies are fair approximations to fractional strategies, and this is the case for the proposed schemes.

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

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

A code is said to be combinatorially t-secure, if it is $(t, 0)$-secure. It is well known that any code with $\delta > 1 - t^{-2}$ is a so-called t-traceability code, which is combinatorially t-secure using closest neighbour decoding. Unfortunately, this large minimum distance is only possible when the alphabet is large. A binary code cannot be combinatorially collusion-secure.

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

3 Comparison of schemes

Most of the collusion-secure codes from the literature are binary. Non-binary random codes are used in [CFNP00], and one non-binary scheme is presented in [BK01]. In this paper, we will concentrate on binary schemes. The following schemes are known to us.

BS-RS The Boneh and Shaw Replication Scheme is a posteriori (t, ϵ)-secure [BS95, BS98] with parameters $[r(t-1), t]$ See Section 5.3 for details.

RS-RC BS-RS with random codes. The scheme is due to Boneh and Shaw [BS95, BS98], but we present a new error analysis in Chapter 5, considerably improving the performance. The code is asymptotically good, with exponentially decreasing error rate. It is also scalable and can be constructed for any number of users M and any error rate ϵ. See Section 5.3.

BBK Barg, Blakley, and Khabatiansky [BBK03]. There is a lot to be said about this scheme, and we shall look at some of it in Chapter 4. The code is asymptotically good, and it is the first scheme to introduce a decoding algorithm with complexity logarithmic in M.

SS(2) Any $(2, 2)$-separating code is $(2, \epsilon_I, 0)$-secure and there exists infinite families of such with exponentially declining ϵ_I [Sch03b]. A special case is the simplex codes analysed in [HJDF00].

SC-SS(3) Scattering Codes give rise to a 3-secure scheme. The original scheme was due to Sebé and Domingo-Ferrer [SDF02b, SDF02a] but this is insecure. A modified scheme [Sch04] achieves better rates and is secure.

LBH Le, Burmester, and Hu [LBH03]. This scheme is susceptible to adverse selection. The length is $n = O(4^t \log(M/\epsilon))$, which is good for small t but terrible for large t. Codes may be constructed for any M and ϵ.

Tardos Tardos' [Tar03] scheme is an a priori (t, ϵ)-secure (n, M) code with $n = 100t^2 \log(M/\epsilon)$, for any M and ϵ. The length is extremely good for large t, but unfortunately, the scheme is subject to adverse selection. A remarkable feature in this scheme is that the probability of accusing a given innocent user is independent of the Marking Assumption and the number of pirates. This makes it more flexible in that even though an over-sized pirate coalition can still get away, but they will rarely frame anyone.

New schemes This paper presents a couple of new schemes, by combining different inner and outer codes using the techniques developed for RS-RC. See Chapters 5 and 6.

Scheme	Toy	Practical	Asymptotic in M	Scalability in t	Other
BS-RS	Yes	No	No	$t = M$	
SS(2)	Yes	A few	Yes	$t = 2$	$\epsilon_{\mathrm{II}} = 0$
SS-RC(2)	Yes	Flexible	Yes	$t = 2$	
SS-RS(2)	Few	Inflexible	(SS-AG)	$t = 2$	$\epsilon_{\mathrm{II}} = 0$
SC-SS(3)	—	Inflexible	No	$t = 3$	
RS-RC	Yes	Flexible	Yes	$n = \Theta(t^4 \log t)$	
RS-RS	None	Not good	(RS-AG)	$n = \Omega(t^6 \log t)$	
BBK	Some	Inflexible	Yes	$n = \Omega(t \cdot 2^t)$	
LBH	Yes	Flexible	Yes	$n = \Omega(4^t)$	Adv. sel.
Tardos	Yes	Flexible	Yes	$n = \Theta(t^2)$	Adv. sel.

Table 3.1: Categorisation of fingerprinting schemes.

3.1 Scalability and flexibility

A general scheme is expected to scale well in the number of pirates M and preferably also in the error rate ϵ. By scalability we often think of only the asymptotic behaviour of the parameters, but this is not sufficient for fingerprinting because the most interesting cases have only moderate values of M. With six billion people on Earth, it is not realistic to sell ten million copies of a digital file, and even one billion would be impressive.

In Table 3.1, we try to categorise the available schemes according to their performance for different parameter ranges. By toy schemes we mean schemes which are good for few users, i.e. $M < 1000$. We use toy schemes as inner codes for concatenation.

We think of practical parameters as $\log M \in [10, 35]$ and error rates no worse than 10^{-4}. These values are a bit arbitrary, and rather on the liberal side. Some codes may be constructed with good properties for virtually any size and error probability in this range. Such codes are described as 'flexible' in the table. The 'inflexible' codes have good parameters for some practical parameter values, but are not constructible for others.

There is no clear definition of 'good' properties in these cases. Basically, when a scheme is listed as not being good for practical or toy parameters, it appears to be so bad that it has not been worth computing exact properties.

A family of (n_M, M) codes is said to be asymptotically good if $\lim_{M\to\infty} R > 0$. Asymptotic fingerprinting schemes have asymptotically good codes. All the schemes which are asymptotically good, also have error rates tending to zero. Comparison of some asymptotic properties are presented in Table 3.3.

General schemes should also scale well in t. There are known schemes working for only small fixed values of t, and Table 3.1 shows how the length depends on t asymptotically. In Table 3.2 we present some sample lengths for the schemes which

M	t	RS-RC	Tardos	LBH
2^{10}	10	478 110 526	300 000	94 283 604
2^{10}	50	$0.362 \cdot 10^{12}$	7 500 000	$0.114 \cdot 10^{33}$
2^{10}	32	$0.570 \cdot 10^{11}$	3 072 000	$0.166 \cdot 10^{22}$
2^{20}	20	$6.556 \cdot 10^{9}$	1 480 000	$1.217 \cdot 10^{14}$
2^{20}	100	$5.196 \cdot 10^{12}$	37 000 000	$1.778 \cdot 10^{62}$
2^{20}	1000	$71.565 \cdot 10^{15}$	3 700 000 000	$1.271 \cdot 10^{604}$
2^{30}	30	$41.527 \cdot 10^{9}$	3 960 000	$1.516 \cdot 10^{20}$
2^{30}	150	$33.124 \cdot 10^{12}$	99 000 000	$2.678 \cdot 10^{92}$
2^{30}	2^{15}	$140.303 \cdot 10^{21}$	4 724 464 026 000	$2.634 \cdot 10^{19730}$

Table 3.2: Lengths for some fingerprinting codes with various M and $t = \log M$, $t = 5 \log M$, and $t = \sqrt{M}$, for $\epsilon \leq 10^{-10}$.

behave reasonably well for practical parameters and moderate and large t.

If we require the scheme to be resistant to adverse selection, RS-RC appears to be the best choice against moderate or large t.

3.2 Key size

All the fingerprinting scheme presented need a randomising secret key to limit the information available to the pirates. The only work to address the key size so far has been [BBK03]; the BBK scheme has a key size of $n_O \log q! = O(\log M)$ bits, where n_O is the outer code length and q is the outer code alphabet size. The key consists of n_O mappings from the outer code alphabet onto the inner code, one for each position in the outer code. Some other schemes based on outer codes with large distance will have a similar key size, with one subkey per outer code coordinate position. Each subkey will typically be a permutation of the inner code and a mapping from the outer code alphabet size.

Separating codes against two pirates (SS(2)) use a permutation on the entire code. When random codes are used (Tardos, LBH, RS-RC), the entire random code is a secret key, thus creating a huge key space.

In practice, the schemes will be pseudo-randomly generated, such that the actual random (and secret) key may be much smaller. No work exists so far on how pseudo-randomness will affect error probabilities in the various schemes, and how much the actual key size can be reduced. This could be an interesting topic for future research.

3.3 Complexity and efficiency of tracing

Few authors address decoding complexity. Most of the concatenated schemes use closest neighbour decoding for the outer code, and this has complexity $O(M \log M)$

in general, because the entire codebook, an $M \times n_O$ matrix, has to be compared to the false fingerprint, and $n_O = O(\log M)$ for asymptotically good codes. The LBH code use a similar technique, though the heuristic they use is not the Hamming metric. The BBK scheme and other codes with AG outer codes, can benefit from the Guruswami-Sudan algorithm of complexity $O(\log M)$.

The problem with the BBK decoding is the complexity of the inner decoding. Every possible t-set of codewords has to be considered, giving exponential complexity in t. Even for moderately large t, the BBK decoding is likely to be slow. The Tardos and LBH decoding complexities depend on t only through the dependency on n. Thus Tardos has complexity $O(t^4)$ whereas LBH is exponential in t.

There are a few schemes having worse complexity than $O(n_O M)$ too, see Table 3.3 for details.

3.4 Length formulæ

There exist a few lower bounds on the length of collusion-secure codes. A weak bound was given in [BS98]. The following more recent bounds are stronger, but still no bound is known incorporating the size M of the code, or which is valid for fixed ϵ.

Proposition 1 [PSS03]
If C is t-secure (n, M) code with ϵ-error where $M > t$, then $n = \Omega(-t^2 \log(c\epsilon))$ when $-\ln \epsilon \geq Kk \log t$, where k is the expected number of distinct column types and K is some sufficiently large constant.

Proposition 2 [Tar03]
Let C be an (n, M) code over an arbitrary alphabet. Let $3 \leq t \leq M$ be an integer and $0 < \epsilon < (100t^a)^{-1}$ for some constant $a > 1$. If C satisfies 1 and 2 below, then $n > -d_a t^2 \log \epsilon$ where $d_a > 0$ depends only on a.

1. For any coalition $P \subseteq C$ of size at most $t-1$, and for any pirate strategy, the probability of accusing a given user $\mathbf{c} \notin P$ is at most ϵ
2. The probability of failing to accuse any guilty user $\mathbf{a} \in P$ is at most 0.01.

The flexible schemes have closed form formulæ for the length and rate in terms of the other parameters. In the following we rephrase such formulæ for the RS-RC, LBH, and Tardos schemes.

For RS-RC, we have

$$n = \frac{(2t-1)\left\lceil 8t^2(3+2\log t)\right\rceil \max\{-\log\epsilon_1, \log M - \log\epsilon_2\}}{D\left(\frac{1}{t+1}\middle\|\frac{1}{2t}\right)}, \tag{3.1}$$

$$R = \frac{D\left(\frac{1}{t+1}\middle\|\frac{1}{2t}\right)}{(2t-1)\left\lceil 8t^2(3+2\log t)\right\rceil}. \tag{3.2}$$

Name	Asymp. rate		Complexity
	$t = 2$	$t = 3$	
RS-RC	$2.42 \cdot 10^{-4}$	$3.62 \cdot 10^{-5}$	$O(M \log M)$
BBK	0.015	0.000638	$O(\log M)$
SS-RC(2)	0.062	N/A	$O(M \log M)$
SS-AG(2)	0.0476	N/A	$O(\log M)$
RS-AG	(SS-AG)	$0.96 \cdot 10^{-9}$	$O(\log M)$
SS(2)	0.026	N/A	$O(M^2 \log M)$
Tardos	0.0025	0.00111	$O(M(\log M)^2)$
LBH	0.0267	0.00707	$O(M \log M)$

Table 3.3: Comparison of asymptotic properties for various schemes. Note that $R = 0.000638$ for BBK might not be constructible even though existence is proven; $R = 0.000156$ is expected to be constructible.

For the LBH scheme, we have

$$n = \frac{1}{a}\ln(M/\varepsilon), \quad \text{where} \quad a = \frac{1}{12}\max_{p\in[0,1]}\min\left\{\frac{(1-p)^2p^{2t}}{(1-p)p^t+1}, \frac{p^2(1-p)^{2t}}{p(1-p)^t+1}\right\}.$$

The operands to min appears to have a unique local maximum on $(0,1)$, and they evaluate to 0 at 0 and at 1. This means that the maximum is obtained for $p = 1/2$ where the two operands are equal, and we get

$$a^{-1} = 3 \cdot 2^{t-1}(2^{t+1}+1).$$

The resulting rate and length are

$$n = 3 \cdot 2^{t-1}(2^{t+1}+1)\ln(M/\varepsilon), \tag{3.3}$$

$$R = \frac{1}{3 \cdot 2^{t-1}(2^{t+1}+1)\cdot \ln 2}. \tag{3.4}$$

For the Tardos scheme, we have $n = 100t^2\lceil\ln(M/\varepsilon)\rceil$ and $R = (100t^2 \log e)^{-1}$.

4 BBK constructions

Barg, Blakley, and Khabatiansky [BBK03] introduced a new scheme, which we call the BBK scheme. They use separating codes as inner codes, and codes with large distance as outer codes. The outer code distance must be larger than what is required for traceability codes, because the codes have to correct tracing errors from inner decoding in addition to the tracing. An important idea behind this scheme is that the inner code may have a very high error rate, because the outer code can be made powerful enough to correct it. The next chapter will show that this idea applies to the Boneh and Shaw scheme as well.

The theory on (t,t)-separating codes, which are used as inner codes, is still limited. Only the following existence lemma is provided in [BBK03],

Lemma 1 [BBK03]
For any n and t, there exists a (t,t)-separating codes of length n and rate $R_{t,t}$ given by

$$R_{t,t} = \frac{-\log(1-2^{-(2t-1)})}{2t-1} - \frac{1}{n}. \tag{4.1}$$

There is no efficient construction technique to obtain the codes guaranteed by the lemma. To get an (t,t)-separating (n,M) code, the proof suggests to start with a random $(n,2M)$ code, and check every possible $2t$-set of codewords for separability. If the $2t$-set is not (t,t)-separated, one of the words is removed from the code. The expected number of codewords removed this way is at most M. We have to check $\binom{M}{2t}$ $2t$-sets, which is hardly feasible except for small M and small t.

When t is large, the rate of these codes is not very good. In fact, we have that

$$\frac{-\log(1-2^{-(2t-1)})}{2t-1} = \Theta(2^{-t}/t),$$

so it follows that the minimum length is $\Theta(t \cdot 2^t)$. Unless better separating codes can be constructed, the BBK scheme is therefore worse than exponential in t.

The length of BBK codes is very large even for moderate t. It is easily checked that for $t \geq 20$, even the first term of $R_{t,t}$ is much smaller than than $1/n'$ where n' is the length of BS-RS in Table 3.2. Thus even the inner code alone need codewords longer than BS-RS.

Separating codes can also be constructed from duals of BCH codes [SH03], and for $t = 2$, the BCH-duals have rates better than $R_{2,2}$. It is not known whether there

exist better separating codes, which can give good, practical BBK codes against many pirates. The advantages that BBK has in terms of decoding complexity and key size are not likely to be relevant for the code parameters from Table 3.2. The decoding complexity depends heavily on t due to the inner decoding, so the actual running time must be expected to be long. The key size increase as rapidly in the length of the inner code for BBK, as it does in the total length for other codes, and the above discussion shows that even the inner length alone is greater than the total length of other schemes.

We will make some constructions against two or three pirates though. As outer codes, we will use Reed-Solomon codes as suggested in [BBK03]. Reed-Solomon codes have parameters $[n, k, n+1-k]_q$ for any $n \leq q$ and any $k < n$. Strictly speaking the codes with $n < q-1$ should probably be called punctured Reed-Solomon codes and with $n = q$ they are extended Reed-Solomon codes; but we will not be that strict. The Reed-Solomon codes are list decodable in time $O(n)$ [GS99].

According to [BBK03], we have

$$\epsilon \leq 2^{-n_O D(\sigma || \frac{t-1}{q-1})} \cdot \boldsymbol{M}, \tag{4.2}$$

where

$$\sigma = \frac{1}{t} - (1-\delta)t, \tag{4.3}$$

and we require $\sigma > (t-1)/(q-1)$. Using $[q, k]_q$ Reed-Solomon codes, δ, and thus ϵ, is a function of q, $\boldsymbol{M}$, and t. Codes with given parameters $\boldsymbol{M}$ and t can be constructed for inner codes with different sizes q, resulting in different error rates. Table 4.1 gives some examples.

Asymptotically, [BBK03] gives the following result.

Proposition 3 (Asymptotic BBK)
If there is an (n_I, q) (t, t)-separating code where q is an even power of a prime, then there is an asymptotic family of (t, ϵ)-secure codes with rate $R_O(\log q)/n_I$, where R_O solves

$$R_O \log q = D\left(\frac{1}{t} - t\left(R_O + \frac{1}{\sqrt{q}-1}\right), \frac{t-1}{q-1}\right),$$

and where ϵ vanishes exponentially.

Remark 4.1
A asymptotically good 3-secure code with rate 0.000156 is obtained by concatenating a $(983, 2^8)$ $(3,3)$-SS with AG outer codes. To construct the inner code, between $\binom{2^8}{6} \approx 2^{38.4}$ and $\binom{2^9}{6} \approx 2^{44.5}$ 6-sets must be checked for $(3,3)$-separation, which is probably feasible on present-day computers.

Using a $(1201, 2^{10})$ or a $(2075, 2^{18})$ inner code, we get rates 0.000372 and 0.000638 respectively. However, the first of these codes might not be constructible, and the second one quite certainly is not.

$\log M$	t	Inner code	Outer code	Length	Error rate
37.2	2	BBK, $(209, 5483)$	$[22, 2]_{5483}$	4807	$0.302 \cdot 10^{-10}$
28	2	$\mathrm{BCH}^{\perp}(2)$, $(126, 2^{14})$	$[11, 2]_{2^{14}}$	1386	$0.228 \cdot 10^{-12}$
42	2	$\mathrm{BCH}^{\perp}(2)$, $(126, 2^{14})$	$[21, 2]_{2^{14}}$	4914	$0.131 \cdot 10^{-10}$
14	3	BBK, $(874, 2^7)$	$[2^7, 2]_{2^7}$	111872	$0.321 \cdot 10^{-33}$
14	3	BBK, $(874, 2^7)$	$[57, 2]_{2^7}$	49818	$0.605 \cdot 10^{-10}$
18	3	BBK, $(1092, 2^9)$	$[41, 2]_{2^9}$	44772	$0.785 \cdot 10^{-10}$
20	3	BBK, $(1201, 2^{10})$	$[2^{10}, 2]_{2^{10}}$	1229824	$0.127 \cdot 10^{-628}$
20	3	BBK, $(1201, 2^{10})$	$[37, 2]_{2^{10}}$	43236	$0.257 \cdot 10^{-10}$
21	3	BBK, $(874, 2^7)$	$[2^7, 3]_{2^7}$	111872	$0.769 \cdot 10^{-27}$
21	3	BBK, $(874, 2^7)$	$[76, 3]_{2^7}$	66424	$0.624 \cdot 10^{-10}$
30	3	BBK, $(1201, 2^{10})$	$[2^{10}, 3]_{2^{10}}$	1229824	$0.203 \cdot 10^{-618}$
30	3	BBK, $(1201, 2^{10})$	$[53, 3]_{2^{10}}$	63653	$0.334 \cdot 10^{-10}$
30	3	BBK, $(1747, 2^{15})$	$[2^{15}, 2]_{2^{15}}$	57245696	$0.693 \cdot 10^{-36954}$
30	3	BBK, $(1747, 2^{15})$	$[27, 2]_{2^{15}}$	47169	$0.902 \cdot 10^{-10}$
32	3	BBK, $(983, 2^8)$	$[2^8, 4]_{2^8}$	251648	$0.110 \cdot 10^{-83}$
32	3	BBK, $(983, 2^8)$	$[82, 4]_{2^8}$	80606	$0.548 \cdot 10^{-10}$

Table 4.1: Some BBK constructions against two or three pirates.

With the constructible $[2046, 7]$ $\mathrm{BCH}^{\perp}(3)$, we get a concatenated code with rate $0.703 \cdot 10^{-5}$. A non-linear subcode of the $\mathrm{BCH}^{\perp}$ of size 121 is used as inner code, to get q to be an even prime power fitting with the AG outer codes.

In the binary case, [BBK03] provides stronger results. The error rate is given as

$$\epsilon \leq M \cdot (q-2)^{-n_O(1-6(1-\delta))}, \tag{4.4}$$

which becomes for Reed-Solomon codes,

$$\epsilon \leq M \cdot (q-2)^{-(n_O-6(k_O-1))}. \tag{4.5}$$

In [BBK03], it is suggested to use $(2, 2)$-separating codes described in [PS72], but they appear to be inferior to the codes guaranteed by Barg et al. themselves. The best constructions, finite as well as asymptotic, that we found, all use the $[126, 14]$ BCH inner code. The asymptotic rate is $R = 0.015$.

Remark 4.2
It is possible to construct some toy codes from BBK, by using a repetition code as outer code. Very small codes is not possible though, because even q has to be rather large.

5 Concatenated schemes

In this chapter we develop a general analysis of concatenation of collusion-secure codes. Even though the BBK scheme uses concatenated codes, the inner codes of BBK are not themselves collusion-secure with the decoding algorithm in use, and consequently the following analysis does not relate to the BBK scheme.

Two combinatorially t-secure codes can simply be concatenated to obtain a larger t-secure code. The fingerprints can be viewed alternately as words of the outer code C_O or of the concatenated code C. Decoding starts with an element of the feasible set with respect to C. Successful inner decoding of a block gives an outer code symbol which is seen by one of the pirates; thus inner decoding yields a vector in the feasible set with respect to the outer code, which can be decoded.

Having two probabilistically t-secure codes, concatenation is non-trivial, but it can still be done [BS98]. Suppose we want to construct a t-secure code C with ϵ-error. Boneh and Shaw chose the parameters such that inner decoding succeeds in every position with probability $1-\epsilon/2$, and and such that outer decoding, given perfect inner decoding, succeeds with probability $1-\epsilon/2$. Thus the total error probability is less than ϵ.

The scheme is actually far better than proved by Boneh and Shaw. Demanding that inner decoding be correct in every position is a strong requirement, because its probability declines exponentially in the code length. This requirement is not necessary. A small fraction of failures from inner decoding will only slightly increase the error probability in outer decoding and improve the overall error rate significantly. This observation was put to use in the BBK scheme, but it should be remembered for any concatenated scheme.

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

5.1 List decoding of concatenated codes

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

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

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

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

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

and

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

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

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

5.2 Random codes (RC)

Boneh and Shaw used random codes, for which Chee [Che96] was credited. Let C_O be a $(n_2, M)_q$ code, where each symbol in each codeword is chosen uniformly at random from the alphabet. The entire code is kept secret by the vendor. Thus the key for the random code scheme is $M \cdot n_2 \cdot \log q$ bits, not counting the keys required by the inner code.

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

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

and the total Type II error rate is bounded as

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

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

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

and the theorem follows by Chernoff's bound. □

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

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

5.3 Replication scheme with random codes

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

BS-RS uses a binary $(r(M-1), M)$ code which is M-secure with ϵ-error. The code book has $M-1$ distinct columns replicated r times. A set of identical columns will be called a type. Every column has the form $(1 \ldots 10 \ldots 0)$, 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 in both these types. If r is large enough we can use statistics to test the null hypothesis that user i be innocent. The output is a list of users for which the null hypothesis may be rejected.

We have

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

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

The key space consists of all the permutations of the columns of the code book, so the key size in bits is

$$K = \log \frac{(r(M-1))!}{(r!)^{M-1}}.$$

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

Theorem 5
If we use

$$q = 2t, \quad \Delta = \frac{t}{t+1}, \quad \epsilon_{\text{in}} = \frac{1}{2t},$$

then RS-RC is an a posteriori (t, ϵ)-secure fingerprinting scheme accommodating M users requiring length

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

where

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

Asymptotically, the length is

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

In this theorem, Δ is made only slightly greater than the minimum value of $(t-1)/t$. By Corollary 1 we require $\epsilon_{\text{in}} < 1/(t+1)$, but to make n_2 linear in t, ϵ_{in} must in fact be much smaller than $1/(t+1)$.
Proof: Theorems 2 and 3 give two bounds on n_2, so we get

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

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

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

For the inner code, we have

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

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

For comparison, we include the original theorem from [BS98].

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

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

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

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

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

Table 5.1: Some lengths when $t = \log M$.

The most interesting point in the original theorem is that $r = \Theta(\log n_2)$, such that n grows faster than linearly in n_2. Since n_2 depends on M and on ϵ, the length was much more dependent on ϵ and M than is with our analysis. In Table 5.1 we see some real sample lengths for these codes, with our and Boneh and Shaw's formulæ.

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

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

$$R < \frac{D(\frac{1-2q2^{-r/(2q^2)}}{t}||1/q)}{r(q-1)}, \tag{5.1}$$

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

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

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

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

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

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

In Table 5.2, we can see some asymptotic rates for our codes. The BBK codes given are the best we could find using constructible inner codes from the literature, namely duals of BCH codes [SH03]. We can see that BBK is better for few pirates, but for larger t we could not find (t,t)-separating codes which are good enough. It is also interesting to note that $2t$ is not the maximising value of q asymptotically, except for $t = 2$.

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

Table 5.2: Asymptotic rates and maximising values of q and r for the RS-RC codes for some numbers of pirates.

5.4 Outer code with large distance

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

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

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

The outer code is decoded by list decoding with threshold Δ. First we study the probability $\pi(\mathbf{c})$ that an innocent user $\mathbf{c}$ be accused. Let S be the set of coordinates where $\mathbf{c}$ is different from any pirate, and let S^C be the complement, i.e. the set of positions where $\mathbf{c}$ match at least one pirate. Let X_i be a stochastic variable which is one if and only if $c_i = x_i$. We get that

$$s(\mathbf{c}, \mathbf{x}) = \sum_{i \in S} X_i + \sum_{i \in S^C} X_i \leq \sum_{i \in S} X_i + \#S^C. \tag{5.2}$$

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

$$s(\mathbf{c}, \mathbf{x}) \leq X + nt(1-\delta)), \quad \text{where } X = \sum_{i \in S'} X_i. \tag{5.3}$$

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

$$\epsilon_1 \leq P(s(\mathbf{c}, \mathbf{x}) > (1-\Delta)n) \leq P(X > ((1-\Delta) - t(1-\delta))n). \tag{5.4}$$

Using Chernoff, we get the following theorem.

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

$$\hat{\epsilon} \leq 2^{-nD(\sigma||\hat{\epsilon}_{\text{in}})}, \quad \text{where} \quad \sigma = (1-\Delta) - t(1-\delta). \tag{5.5}$$

Combining Theorems 2 and 8, we get that

$$\delta > 1 - \frac{1 - \epsilon_{\text{in}} - t\hat{\epsilon}_{\text{in}}}{t^2}. \tag{5.6}$$

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

Example 5.1 *A RS outer code can be combined with a BS-RS inner code. Take for instance, $t = 20$ and $M = 2^t$. Let $q = 2^{10}$ and $r = 3.1 \cdot 10^7$, and use a $(r(q-1), q)$ BS-RS as inner code. As an outer code, we use a $[690, 2]_q$ generalised Reed-Solomon code. With a decoding threshold of $\Delta = 0.958$, we get a total error rate of $\epsilon \leq 0.356 \cdot 10^{-10}$. The total length is $2.139 \cdot 10^{10}$. These parameters are not really bad, but they are not as good as for RS-RC in Table 3.2.*

Concatenations of BS-RS inner codes and RS outer codes will be denoted RS-RS. We have $n_I = \Theta(q^3 \log q)$ from BS-RS, and $q = \Omega(t^2)$ due to the distance requirement. This gives us $n = \Omega(t^6 \log t)$, which is inferior to RS-RC. Furthermore, it is rather difficult to find the optimal choices for the various parameters.

To make asymptotic classes of codes, we can use AG outer codes, as follows. The rates obtained with RS-AG, using BS-RS as inner codes, are not impressive though. Other medium sized inner codes should be sought for.

Proposition 4
If there is an (n_I, q) $(t, \epsilon_{\text{in}})$-secure code where the probability of accusing a given innocent user is at most $\hat{\epsilon}_{\text{in}}$, then there is an asymptotic family of (t, ϵ)-secure codes with rate $R_O(\log q)/n_I$, where R_O solves

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

and where ϵ vanishes exponentially.

Proof: We see from Theorem 2, that exponentially declining ϵ_{I} is obtained if $\Delta > 1 - 1/t + \epsilon_{\text{in}}/t$, but Δ can be taken arbitrarily close to this bound. From Theorem 8, we get that ϵ_{II} will decline exponentially if $R_O \log q < D(\sigma||\hat{\epsilon}_{\text{in}})$, where

$$\sigma = (1-\Delta) - t(1-\delta) \approx \frac{1 - \epsilon_{\text{in}}}{t} - t(1-\delta).$$

q	r	Outer code	Δ	n	$\log M$	ϵ
49	53 500	[49, 2]	0.785	125 832 000	11.2	$0.653 \cdot 10^{-10}$
49	62 690	[49, 3]	0.746	147 446 880	16.8	$0.987 \cdot 10^{-10}$
64	94 000	[64, 3]	0.7765	379 008 000	18	$0.901 \cdot 10^{-10}$
49	78 690	[49, 4]	0.71	185 078 880	22.5	$0.977 \cdot 10^{-10}$
64	106 000	[64, 4]	0.737	427 392 000	24	$0.907 \cdot 10^{-10}$
49	119 000	[49, 5]	0.685	279 888 000	28.1	$0.806 \cdot 10^{-10}$
64	130 000	[64, 5]	0.715	524 160 000	30	$0.959 \cdot 10^{-10}$
49	405 000	[49, 6]	0.66995	952 560 000	33.7	$0.766 \cdot 10^{-10}$

Table 5.3: Some RS-RS codes against three pirates.

Again R_O can be taken arbitrarily close to this bound. Using AG outer codes, we get

$$\delta \approx 1 - R_O - \frac{1}{\sqrt{q}-1},$$

giving

$$\sigma \approx \frac{1-\epsilon_{\text{in}}}{t} - t\Big(R_O + \frac{1}{\sqrt{q}-1}\Big).$$

It follows that R_O can be taken arbitrarily close to the solution x of

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

as required. □

Comparing this proposition to the BBK result of Proposition 3, we can see that ϵ_{in} and $\hat{\epsilon}_{\text{in}}$ must be rather small to obtain high R_O. This is not possible with RS-RS without getting a depressingly low R_I. For instance, with $q = 13^2$ and $r = 10^6$, we obtained $R = 0.96 \cdot 10^{-9}$. We found nothing better. The above proposition may still be useful though, if better inner codes can be constructed.

For $t = 2$, it is better to use Simplex inner codes than BS-RS, and this will be considered in Chapter 6. The lengths and rates obtained show that Reed-Solomon and AG codes are much better than random codes if the inner code can be made large enough. Some lengths for $t = 3$ are shown in Table 5.3, but they cannot compete with the BBK scheme. The BS-RS inner codes simply grow too long for sufficient sizes.

6 Fighting two pirates

We mentioned that the BS replication codes may not be the ideal choice for inner codes. For two pirates we have good alternatives, which we consider now.

It was proven in [BBK03, Lemma 3.3], that any $(2,2)$-SS is 2-secure with 1/3-error. A much stronger result is the following lemma from [Sch03b].

Lemma 2
Any $(2,2)$-SS is $(2,\epsilon_I,0)$-secure, where

$$\epsilon_I \leq (M-2)\max\left\{\frac{1}{2}\binom{d_1}{\theta_{2,1}}^{-1}, \binom{d_1}{d_1/2}^{-1}\right\}, \tag{6.1}$$

and d_1 is the minimum distance and $\theta_{2,1}$ is the $(2,1)$-separating weight.

The separating properties of duals of BCH codes were analysed in [CELS03, SH03]. Some $(2,2)$-SS with good error rates are presented in Table 6.1.

A special case of Lemma 2 appeared in [HJDF00], where it was proved that the $[2^k-1,k,2^{k-1}]$ Simplex code is $(2,\epsilon_I,0)$-secure where $\epsilon_I = 2^{k-2^{k-1}}$. This appears as a slightly stronger bound than in Lemma 2, but the reason for that is that they assume that the pirates choose the bit for each position independently. This assumption is expected to be reasonable if the Simplex code is used as an inner code for concatenation, but for a Simplex in itself, it is not true.

6.1 Asymptotic constructions

The best asymptotic rate offered for $t=2$ in [BBK03] was 0.015, using the $[126,14]$ BCH-dual as inner code and an AG outer code. On the other hand, [Sch03b] offered a rate of 0.026 for an asymptotic class $(2,2)$-SS.

Code		(n,M)	ϵ_I
$\mathrm{BCH}^{\perp}(2)$	$m=7$	$[126,14]$	$0.193\cdot 10^{-10}$
$\mathrm{BCH}^{\perp}(2)$	$m=9$	$[511,18]$	$0.16\cdot 10^{-37}$
$\mathrm{BCH}^{\perp}(3)$	$m=8$	$[255,24]$	$0.26\cdot 10^{-20}$
$\mathrm{BCH}^{\perp}(3)$	$m=10$	$[1023,30]$	$0.80\cdot 10^{-78}$

Table 6.1: Some $(2,2)$-separating codes which are good collusion-secure codes.

Theorem 9 (SS-RC(2))
By concatenating the $[126, 14]$ punctured dual of the two-error-correcting BCH code with a random code, we get an infinite class of 2-secure codes with ϵ-error and rate R, for any $R < 0.0476$ and exponentially declining error rates given as

$$\epsilon_{\mathrm{I}} \leq 2^{-n\frac{D(2\Delta-1||0.2\cdot 10^{-10})}{126}} \quad \text{and} \quad \epsilon_{\mathrm{II}} \leq 2^{n(R-D(1-\Delta||2^{-14})/126)},$$

where Δ may be chosen freely in the interval $1/2 + 10^{-11} < \Delta < 1 - 2^{-14}$.

Remark 6.1 (SS-RC(3))
Any $(3, 3)$-SS is 3-secure with $4/7$-error, a fact which follows by an argument similar to that of [BBK03, Lemma 3.3]. Using the $(4092, 2^{12})$ subcode of the dual of BCH(3) [SH03], and a random outer code, we get an asymptotic class of codes which are $(3, \epsilon)$-secure with rate $2.74 \cdot 10^{-4}$, where ϵ vanishes. This rate is inferior to BBK though.

Instead of random outer codes, we can use AG codes. Since these codes have algebraic structure, it is possible to take advantage of the fact that the inner codes have $\epsilon_{\mathrm{II}} = 0$ and make concatenated schemes which also have $\epsilon_{\mathrm{II}} = 0$. We call this scheme SS-AG.

For any innocent user $\mathbf{c}$, we have $s(\mathbf{c}, \mathbf{x}) \leq 2(1-\delta)n_O$. Hence $\mathbf{c}$ will never be accused if $\Delta > 1 - 2(1-\delta)$. Asymptotically, δ can be taken arbitrarily close to $(1+\Delta)/2$. The bound on ϵ_{II} is found from Theorem 2,

$$\epsilon_{\mathrm{I}} \leq 2^{-n_O D(-1+2\Delta||\epsilon_{\mathrm{in}})}.$$

It is necessary that $\Delta > (1+\epsilon_{\mathrm{in}})/2$, which gives us

$$\delta \approx 1/2 + (1+\epsilon_{\mathrm{in}})/4.$$

The outer code rate can be brought arbitrarily close to

$$R_O \approx 1 - \delta - \frac{1}{\sqrt{q}-1} \approx \frac{1}{2} - \frac{1+\epsilon_{\mathrm{in}}}{4} - \frac{1}{\sqrt{q}-1}.$$

Using the $[126, 14]$ inner code, we get $R_O \approx 0.242$, and overall rate 0.0269. This is not as good as using random codes, but it is better than BBK. Furthermore, it can be decoded with the GS algorithm in time $O(\log M)$, just like BBK.

Another alternative for the inner codes is simplex codes. The SS-RC scheme with $[15, 4, 8]$ inner codes, get a remarkable rate, as the following theorem show. This code is too small to work with AG outer codes, though, and larger simplex codes would be inferior to the $[126, 14]$ code.

Theorem 10
The SS-RC scheme forms an infinite class of $(2, \epsilon)$-secure codes with rate R, for any $R < 0.062$, and exponentially declining error rates given as

$$\epsilon_1 \leq 2^{-n\frac{D(2\Delta-1||1/16)}{15}} \quad \text{and} \quad \epsilon_2 \leq 2^{n(R-D(1-\Delta||1/16)/15)},$$

where Δ may be chosen freely in the interval $17/32 < \Delta < 15/16$.

Corollary 3
The SS-RC codes with $[15, 4, 8]$ inner codes are probabilistically $(2, \epsilon)$-secure with length

$$n = 15 \left\lceil \max \left\{ \frac{\log \epsilon_1}{D(2\Delta - 1||1/16)}, \frac{\log \epsilon_2 - \log M}{D(1-\Delta||1/16} \right\} \right\rceil,$$

for any Δ such that $17/32 < \Delta < 15/16$.

6.2 Practical codes

In Table 6.3, we present code lengths for 1000 to a billion users with the schemes we know. The RS-RC codes are computed with $q = 4$, $\epsilon_{\text{in}} = 0.002$. The error rates were set such that both ϵ_{I} and ϵ_{II} both are less than $10^{-10}/2$. We used $\Delta = 0.655$ for 2^{10} users, $\Delta = 210/320$ for 2^{15} users, $\Delta = 52/80$ for 2^{20} users, $\Delta = 41/64$ for 2^{25} users, and $\Delta = 203/320$ for 2^{30} users. Note that the simplex codes will have much better error rate than the 10^{-10} that we require. Also observe that the $[126, 14]$ and $[254, 24]$ BCH duals are $(2, 10^{-10})$-secure with shorter codewords than SS-RS; but unfortunately we do not know of good separating codes for other sizes.

Following the analysis of SS-AG, it is reasonable to propose SS-RS: separating inner codes and Reed-Solomon outer codes. This will also result in a $(2, \epsilon_{\text{I}}, 0)$-secure code. The $[31, 5, 16]$ Simplex code with $\epsilon_{\text{I}} = 2^{-11}$ appears to be a good inner code. It is also possible to use the $[126, 14]$ code as an inner code. Constructions are shown in Table 6.2, and the general result is given in the following theorem.

Theorem 11
A concatenated code of an $(2, \epsilon_{\text{in}}, 0)$-secure inner code and a $[n_O, k_O]_q$ RS code is 2-secure with $\epsilon_{\text{II}} = 0$ and

$$\epsilon_{\text{I}} \leq P(Y \leq 2k_O - 3), \quad \text{where } Y \sim \mathcal{B}(n_O, 1 - \epsilon_{\text{in}}). \tag{6.2}$$

Tracing is done with the Guruswami-Sudan algorithm.

Proof: We have $\epsilon_{\text{I}} \leq P(X > n_O(1 - 2(1-\delta)) + 1)$ where $X \sim \mathcal{B}(n_O, \epsilon_{\text{in}})$, from the proof of Theorem 2. Setting $Y = n_O - X$ gives the theorem. □

There is a variant from [FS03] using Simplex inner codes, Reed-Solomon outer codes, and a more complicated inner decoding algorithm.

From Chapter 4, we have a $(1\,386, 2^{28})$ BBK code, as well as a $(4\,807, 5483^3)$ code, which are also only beaten by SS-RC and SS-RS.

Inner code	Outer code	Concatenated code	Error rate
$[15,4]$	$[5,2]_{32}$	$[155,10]$	$0.3\cdot 10^{-12}$
$[15,4]$	$[7,3]_{32}$	$[217,15]$	$0.2\cdot 10^{-11}$
$[15,4]$	$[9,4]_{32}$	$[279,20]$	$0.7\cdot 10^{-11}$
$[15,4]$	$[11,5]_{32}$	$[341,25]$	$0.2\cdot 10^{-10}$
$[126,14]$	$[3,2]_{2^{14}}$	$[378,28]$	$0.111\cdot 10^{-20}$
$[15,4]$	$[13,6]_{32}$	$[403,30]$	$0.4\cdot 10^{-10}$
$[126,14]$	$[4,3]_{2^{14}}$	$[504,42]$	$0.454\cdot 10^{-11}$

Table 6.2: Some SS-RS codes.

$\log M$	BS-CS	RS-RC	Simplex	SS-RC	SS-RS
10	759 330	299 889	1 023	1 305	155
15	848 085	334 359	32 767	1 455	217
20	937 440	367 359	1 048 575	1 545	279
25	1 026 684	401 001	$2^{25}-1$	1 605	341
30	1 116 408	435 471	$2^{30}-1$	1 695	403

$\log M$	Tardos	LBH
10	12 000	25 884
15	13 600	28 878
20	14 800	31 872
25	16 400	34 867
30	17 600	37 861

Table 6.3: Code lengths against two pirates for 1000 to a billion users and error rate $\epsilon \leq 10^{-10}$.

7 Open problems

We have made a new error analysis of the Boneh-Shaw fingerprinting scheme and shown how the component codes can be replaced by others to improve the scheme. Using outer codes with large distance gives $O(\log M)$ decoding complexity, but requires large inner codes. It is realised that good inner codes are yet unknown in the general case. A certain alphabet size is needed for the outer codes, and BS-RS has a too rapidly growing lengths. Constructing better t-secure codes for 'toy' parameters is an important open question.

We have pointed out the control parameters in the RS-RC scheme, and these may be used to tune the performance of the scheme to actual applications. Good and general statements on the optimal choices of control parameters is still an open problem.

Barg, Blakley, and Khabatiansky [BBK03] ask whether it is possible to compute a channel capacity for the fingerprinting problem. As we are able to construct schemes with higher and higher rates, it is of increasing interest to know the theoretical capacity limit. Even though some bounds on the code lengths are known, these are independent of the code size M, and they assume an error rate ϵ which is upper bounded in terms of the collusion size t. In particular, an asymptotic lower bound on the length n in terms of t for fixed ϵ would be very interesting.

Schemes like Tardos and LBH are subject to adverse selection, but this is a problem which can possibly be controlled by limiting the probability of such selection. Concatenating a Tardos code with a larger outer code, is also likely to hide the information leading to adverse selection. How good such a scheme would be is also an open problem.

All the schemes studied have security proofs based on truly random keys or permutations. It is an open problem how the various schemes work with pseudo-random keys and reduced key sizes.

Bibliography

[BBK03] 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. 1, 2.2, 3, 3.2, 4, 1, 4, 4, 4, 4, 6, 6.1, 6.1, 7

[BK01] A. Barg and G. Kabatiansky. A class of i.p.p. codes with efficient identification. Technical report, DIMACS, 2001. 3

[BMP86] 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

[BS95] 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, 3

[BS98] 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, 3, 3.4, 5, 5.3

[CELS03] Gérard D. Cohen, Sylvia B. Encheva, Simon Litsyn, and Hans Georg Schaathun. Intersecting codes and separating codes. *Discrete Applied Mathematics*, 128(1):75–83, 2003. 6

[CFN94] 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

[CFNP00] B. Chor, A. Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Trans. Inform. Theory*, 46(3):893–910, May 2000. 1, 2.2, 3

[Che96] Yeow Meng Chee. *Turán-type problems in group testing, coding theory and cryptography*. PhD thesis, University of Waterloo, Canada, 1996. 5.2

[FS03] Marcel Fernandez and Miguel Soriano. Protecting intellectual property by guessing secrets. 2003. EC-Web Sept. 2-5 2003 Prague, Czech Rep. 6.2

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Trans. Inform. Theory*, 45(6):1757–1767, 1999. 4

[HJDF00] J Herrera-Joancomarti and Josep Domingo-Ferrer. Short collusion-secure fingerprints based on dual binary Hamming codes. *Electronics Letters*, 36:1697–1699, September 2000. 3, 6

[HR90] Torben Hagerup and Christine Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33:305–308, 1990. 2

[LBH03] Tri Van Le, Mike Burmester, and Jiangyi Hu. Short *c*-secure fingerprinting codes. In *Proceedings of the 6th Information Security Conference*, October 2003. Available at `http://websrv.cs.fsu.edu/~burmeste/`. 2.1, 3

[PS72] M. S. Pinsker and Yu. L. Sagalovich. A lower bound on the size of automata state codes. *Problems of Information Transmission*, 8(3):59–66, 1972. 4

[PSS03] Chris Peikert, Abhi Shelat, and Adam Smith. Lower bounds for collusion-secure fingerprinting. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2003. 1

[PW97] Birgit Pfitzmann and Michael Waidner. Anonymous fingerprinting. In *Advances in cryptology—EUROCRYPT '97*, volume 1233 of *Lecture Notes in Comput. Sci.*, pages 88–102. Springer, Berlin, 1997. 1

[Sag94] Yu. L. Sagalovich. Separating systems. *Problems of Information Transmission*, 30(2):105–123, 1994. 2

[Sch03a] Hans Georg Schaathun. The Boneh-Shaw fingerprinting scheme is better than we thought. Technical Report 256, Department of Informatics, University of Bergen, 2003. Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`.

[Sch03b] Hans Georg Schaathun. Fighting two pirates. In *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes*, volume 2643 of *Springer Lecture Notes in Computer Science*, pages 71–78. Springer-Verlag, May 2003. 3, 6, 6.1

[Sch04] Hans Georg Schaathun. Fighting three pirates with scattering codes. Technical Report 263, Department of Informatics, University of Bergen, 2004. Also available at `http://www.ii.uib.no/~georg/sci/inf/coding/public/`. 3

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

[SDF02b] Francesc Sebé and Josep Domingo-Ferrer. Short 3-secure fingerprinting codes for copyright protection. In *ACISP 2002*, volume 2384 of *Springer Lecture Notes in Computer Science*, pages 316–327. Springer-Verlag, 2002. 3

[SH03] Hans Georg Schaathun and Tor Helleseth. Separating and intersecting properties of BCH and Kasami codes. In *Cryptography and Coding*,

volume 2898 of *Springer Lecture Notes in Computer Science*. Springer-Verlag, December 2003. 9th IMA International Conference. 4, 5.3, 6, 6.1

[SNW00] Reihaneh Safavi-Naini and Yejing Wang. Sequential traitor tracing. In *Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA)*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 316–332. Springer, Berlin, 2000. 1

[Tar03] G. Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003. `http://www.renyi.hu/~tardos/fingerprint.ps`. 2.1, 3, 2

[Wag83] Neal R. Wagner. Fingerprinting. In *Proceedings of the 1983 Symposium on Security and Privacy*, 1983. 1