Traitor Tracing for Shortened and Corrupted Fingerprints

Reihaneh Safavi-Naini and Yejing Wang
School of Information Technology and Computer Science
University of Wollongong
Wollongong 2522, Australia
email:[rei,yejing]@uow.edu.au

Abstract

We consider tracing fingerprinted media data such as images and video data. We consider pirate objects that are constructed by a group of up to c colluders who have used a range of attacks including 'cut and paste', averaging to weaken the embedded marks, and cropping to remove part of the fingerprint. We have two main results: First, we give an efficient algorithm for tracing shortened fingerprints that are obtained from a class of generalized Reed-Solomon codes. Second, we propose combined mark detection and tracing using soft-decision decoding to give a more powerful tracing algorithm. We conclude the paper by discussing our results and giving possible directions for future research.

1 Introduction

Traitor tracing [3] is studied in two related scenarios: *Key fingerprinting* in the context of broadcast encryption [3, 26, 18] and *data fingerprinting* in the context of copy protection. In broadcast encryption, a decoder box has a unique set of keys and the aim is to trace at least one of c colluders who have constructed a pirate decoder that can decrypt the broadcast. In data fingerprinting the distributer of a digital object, for example a software, embeds an individual fingerprint which is a q-ary string (also called a *mark sequence*) in each copy of the object, enabling a pirate copy to be traced to a traitor. Collusion secure data fingerprinting is considered in [1, 2, 19]. A pirate copy is constructed by a set of at most c colluders who compare their objects, find some of the mark positions and construct an object that has one of their marks in the found positions.

c-Traceability codes (c-TA) for data fingerprinting [25] use Hamming distance between a pirate word and codewords to trace one of the at most c possible colluders who have constructed a pirate object. Tracing in this case is the same as the traditional decoding problem in error-correcting codes.

We consider *fingerprinting for media data* such as images or video clips. To protect against illegal copying and to be able to trace pirates, a seller embeds a unique fingerprint which is a q-ary string of length ℓ, in each copy of the protected object. We assume embedding is by dividing the object into blocks (for example 50×50 pixel blocks) and using a robust watermarking algorithm to embed symbols of a fingerprinting sequence one by one in each block [5]. When a pirate object is found, the embedded fingerprint sequence is recovered and traced to a traitor. The output of the watermarking algorithm is either one of the q-ary symbols of the alphabet, or $'?'$ which means the algorithm cannot decide the embedded mark.

We consider the case that the pirate object is constructed by a collusion of up to c colluders each having a distinct fingerprinted copy of the object. We assume the colluders do not know which blocks are used for embedding marks. However they may compare their objects to find marked blocks. That is although we do not assume that marked blocks in different objects have exactly the same values in different copies (for

example have the same pixel values in different copies of an image) but they are more similar compared to a block marked with different symbols.

The pirate object is constructed by (i) cut and paste of parts of the different copies, (ii) using averaging attack to weaken the marks and defeat the mark detection algorithm, and (iii) cropping the object and removing some parts of the fingerprint. The pirate fingerprint recovered from a pirate object will be a sequence, possibly of shorter length compared to the original fingerprints, having some erased marks and in each of the non-erased positions, a mark from one of the colluders. The result of cut and paste attack on the object is that each component of the pirate fingerprint is from the fingerprint of one of the colluders or is an *erased mark*. In averaging attack, the pirate object is obtained by finding the average values of object elements, for example pixels in an image. This attack reduces the strength of the embedded marks and in watermark recovery phase introduces errors in the recovered fingerprints. If this attack is used on all blocks and if there are enough colluders [12], then majority of the marks will be erased and tracing will fail. In [6] a bound on the size of collusion to make the mark undecidable is derived. Using c-traceability codes for fingerprinting sequences as above, will still have the same upper bound on collusion security. However if large enough portions of the fingerprinting sequence is recovered, the pirate object can be traced to one of the colluders.

Cropping attack will remove parts of the fingerprint and will result in a shorter fingerprint. Collusion secure fingerprinting codes and traceability codes protect against attacks of type (i) and (ii), as long as the number of erased (unrecognisable) marks are not too many. However they fail completely if the pirate fingerprint is shortened even by one component.

Tracing shortened fingerprints was considered in [22] and a tracing algorithm based on Levenshtein distance was proposed. The tracing algorithm in this case, instead of Hamming distance, used the length of the longest substring common between the pirate word and each codeword to measure similarity between the two, and choose the most similar codeword as a colluder. Instead of error-correcting codes deletion correcting codes were proposed. The drawback of this method is that (i) tracing algorithm is computationally expensive and (ii) construction of deletion correcting codes that satisfy the required conditions is an open problem.

In this paper we have two main contributions.

1. We consider the problem of tracing shortened fingerprints and show that using certain generalized Reed-Solomon (GRS) codes for fingerprinting allows shortened fingerprints to be correctly traced. We show that for GRS codes deletion decoding can be formulated as a polynomial interpolation problem and the list decoding approach of Gurswami and Sudan [11] can be used to find the closest, in this case 'most similar', code vector to the given shortened pirate word. This removes both shortcomings of the previous method by giving a construction for codes that can protect against deletion of fingerprint components, and also giving an efficient algorithm for tracing. We use list decoding algorithm of Guruswami and Sudan [11] for GRS codes to find a bound on the number of deletions that can be tolerated.

2. We consider fingerprints in which strength of the marks is weakened and so the mark detection algorithm cannot easily produce a single output for each mark. Attacks such as averaging results in uncertainty in detection of the marks and so the recovered fingerprints will be likely to have errors which would result in incorrect tracing. We propose a *combined mark detection and tracing algorithm* and show that it can be used to construct a more powerful tracing algorithm. We will consider two complementary types of tracing algorithms: 'hard-tracing' algorithms of traceability codes applied to a 'hard-detected' fingerprint (pirate sequence), and 'soft-tracing' algorithms that will be used on 'soft-detected' fingerprints. A 'soft-detected' fingerprint, also called a *pirate matrix*, is a matrix whose columns correspond to the probability distributions on the watermarking symbols. That is column j

gives a probability distribution on the symbols of the watermarking code to occur in the j^{th} position of the fingerprint.

Soft-tracing uses the notion of *generalized distance* and tracing is by finding the codeword that has the highest 'similarity' with the pirate matrix. This tracing algorithm in general is computationally expensive but if the fingerprinting code is RS code, then the soft-decision decoding of [16] can be used to efficiently find a colluder.

Pirate sequences can be thought of as special types of pirate matrices where each column has a single one and the rest of the elements zeros. A tracing algorithm takes a pirate matrix and outputs a colluder and is powerful if it can trace a large set of pirate matrices. We will show that the set of pirate matrices that can be traced by hard-tracing is different from the set that can be traced by soft-tracing and so combining the two will result in a more powerful tracing compared to using only hard-tracing.

An interesting result of soft-traceability is that if the fingerprinting code is obtained from an error-correcting code whose minimum distance satisfies $d > \ell(1-1/c)+e$, then pirate matrices that are constructed by colluders of size up to c can be correctly traced. In this bound e is the maximum number of columns that are ambiguous and have at least two symbols with the same maximum probabilities. These columns are undecidable and effectively erased marks. Assuming no erasure ($e = 0$) we have $d > \ell(1 - 1/c)$. Comparing this bound with the bound $d > \ell(1 - 1/c^2)$ for traceability codes [25] shows that the required minimum distance for soft-tracing is less than that of hard-tracing.

An interesting open problem is to combine the above two results and consider fingerprints that are shortened and weakened.

Using list decoding for deletion correction to our knowledge is the first algebraic method for deletion correction and has wider applications such as synchronising signals. Although the tracing algorithms are for GRS codes but the general method of tracing shortened fingerprints and combined watermarking and tracing algorithm is applicable to all c-TA codes that are based on error-correcting codes and have list decoding and (or) soft-decision decoding algorithm including algebraic geometry codes (AG-codes) [16, 10].

1.1 Related Works

Fingerprinting media data

A traditional method of fingerprinting a media object is by using watermarking where a perceptually tolerable signal is embedded in the object such that a detector with a correct key can recover the watermark [4]. A common attack for removing watermark in such objects is 'cropping' where parts of the object are removed. This is a very effective and depending on the size of the cropped part can completely remove (detector cannot recover) the mark or weaken it. An alternative fingerprinting method [5] is to divide the object into blocks and embed a q-ary fingerprint in the object by using a watermarking algorithm to embedded each element of the fingerprint in a separate block. Now if the object is cropped, the recovered pirate fingerprint will be generally shorter than the embedded one. This is the approach considered in this paper.

Collusion attack in the case of media data is similar to collusion attack in data fingerprinting. That is, colluders will compare their objects, detect mark places where the embedded marks (symbols) are different (different versions of the same block) and construct a pirate object such that each block contains one of the versions that they have. We refer to this attack as basic *collusion attack*. Colluders may also try to make the marks unreadable in which case an *erasure* occurs. We note that a 'cut and paste' attack where colluders construct the pirate object by pasting parts obtained from their individual copies can be described as a combination of cropping and basic collusion attack with erasure.

We note that cropping and collusion substitution are two independent attacks and need separate protection.

Collusion secure fingerprinting

Boneh and Shaw studied collusion security for data fingerprinting and defined and constructed *c-frame-proof codes* [1, 2] in which collusions of up to c colluders cannot frame another user, and *c-secure codes* with ε-error, in which given a pirate copy that is constructed by a collusion of up to c colluders, at least one traitor can be traced. To construct a pirate object, colluders may use a basic collusion attack with erasure. This is captured in a 'marking assumption'. In [12] the marking assumption is extended to the case that all positions, including undetected positions, are erasable.

Staddon et al [25] defined q-ary *c-traceability codes* where the construction of pirate word is as above but does not include erasure, and tracing algorithm uses the Hamming distance between the pirate word and the set of codewords. In [24] it was shown that by using list decoding algorithm [28, 11] for GRS codes, a set of traitors who are at distance at most $\ell - \ell/c$ from the pirate word can be found.

Traceability systems are also studied in the context of broadcast encryption schemes [3, 26, 27, 18, 9, 21]. In [15] pirate strategies are discussed and corresponding decoders are categorized. It has been shown [14] that tracing traitor is impossible for some type of decoders when the number of traitors exceeds a bound.

A related notion is IPP-codes, or codes with *Identifiable Parent Property*. In a c-IPP code the intersection of all collusions that can construct a pirate word is non-empty and so all traitors in the intersection of all such subsets are identifiable traitors. IPP-code are defined in [13] and constructed in [25, 23].

Other related works are *dynamic tracing* scheme [7] and *sequential tracing* scheme [20] which require the feedback from the channel.

In [22] q-ary fingerprinting for perceptual content and the question of tracing with shortened fingerprint is considered. As noted earlier the tracing algorithm is computationally expensive and construction of a good deletion correcting code is an open problem.

The rest of the paper is organized as follows. In Section 2 we propose a deletion decoding algorithm for GRS codes and show that GRS codes can be used to trace shortened fingerprints. In Section 3 we introduce combined mark detection and tracing algorithm and in Section 4 show that soft-decision decoding of [17, 16] can be used for correctly tracing a colluder. In Section 5 we conclude the paper.

2 An Algebraic Approach to Tracing Shortened Fingerprints

List decoding for an error-correcting code of minimum distance d can correct error patterns with Hamming weight higher than $\frac{d-1}{2}$. List decoding [11] for a GRS code of length ℓ and dimension k, takes a received word and outputs a list of codewords that are at distance up to $\ell - \sqrt{k\ell}$. We model tracing of shortened fingerprint as a list decoding problem and show the correctness of the algorithm. First we show GRS codes that satisfy a certain condition can correct deletions (Section 2.2), and then formulate decoding of shortened words as a conventional decoding problem (Section 2.3) and give an efficient tracing algorithm for shortened fingerprints (Section 2.4) that are constructed by up to c colluders.

2.1 Preliminaries

In this subsection we give definitions and review known results that will be used in the rest of this paper.

Traceability codes with shortened pirate words

Let Σ be a q-ary alphabet, Σ^* the set of strings over Σ, and Σ^ℓ the set of vectors, also called words, of length ℓ over Σ. Staddon et al [25] defined c-traceability code (c-TA) as follows.

Definition 2.1 *([25]) Let Γ be code of length ℓ over an alphabet Σ having n codewords. Let $C = \{u^{(1)}, \cdots, u^{(b)}\} \subseteq \Gamma$ be a collusion set where $u^{(i)} = (u_1^{(i)}, u_2^{(i)}, \cdots, u_\ell^{(i)})$, for $1 \leq i \leq b \leq c$. Define*

$$desc(C) = \{(y_1, \cdots, y_\ell) : y_j \in \{u_j^{(i)} : 1 \leq i \leq b\}, 1 \leq j \leq \ell\}$$

Γ is called a c-$TA_q(\ell, n)$ code if the following condition is satisfied: for any $C \subseteq \Gamma$, $|C| \leq c$, and for any $(y_1, y_2, \cdots, y_\ell) \in desc(C)$, there is a $u^{(i)} \in C$ such that

$$|\{j : y_j = u_j^{(i)}\}| > |\{j : y_j = v_j\}|$$

for any $(v_1, v_2, \cdots, v_\ell) \in \Gamma \setminus C$.

Safavi-Naini et al [22] extended c-TA codes to allow deletions and erasures in the pirate word. A *subword* y of $u = (u_1, \cdots, u_\ell)$ is a vector,

$$y = (u_{i_1}, u_{i_2}, \cdots, u_{i_{\ell'}}), \quad 1 \leq i_1 < i_2 < \cdots < i_{\ell'} \leq \ell.$$

Let $|y|$ denote the length of y. A *common subword* of two codewords $u^{(1)}$ and $u^{(2)}$ is a subword of both $u^{(1)}$ and $u^{(2)}$. For $u^{(1)}, u^{(2)} \in \Gamma$, define

$$\rho(u^{(1)}, u^{(2)}) = \max\{|y| : y \text{ is a common subword of } u^{(1)} \text{ and } u^{(2)}\}$$

For a code Γ denote by

$$\rho(\Gamma) = \max_{u^{(1)}, u^{(2)} \in \Gamma, u^{(1)} \neq u^{(2)}} \rho(u^{(1)}, u^{(2)}).$$

the length of the maximum common substring between two codewords. The value $\rho(u^{(1)}, u^{(2)})$ can be seen as a measure of similarity between two vectors.

The following definition generalizes c-TA codes to allow tracing when the pirate fingerprint is shorter than the embedded one.

Definition 2.2 *([22]) Let r and c be integers, Γ be a code of length ℓ over Σ with n codewords, and $C \subseteq \Gamma$. Define*

$$\begin{aligned} desc(C; r) &= \{y = (y_1, \cdots, y_{\ell'}) : y \text{ is a subword of some } z \in desc(C), \ell - r \leq \ell' \leq \ell\} \\ \Sigma^{\ell, r} &= \{y \in \Sigma^* : \ell - r \leq |y| \leq \ell\} \end{aligned}$$

Γ is called a c-$TA_q(\ell, n; r)$ if there is a tracing function $A : \Sigma^{\ell,r} \to \Gamma$ such that $A(y) \in C$ for any $C \subseteq \Gamma$, $|C| \leq c$, and any $y \in desc(C; r)$.

Theorem 2.1 gives a sufficient condition for c-$\mathrm{TA}_q(\ell, n; r)$ codes in terms of $\rho(\Gamma)$.

Theorem 2.1 *(Lemma 1, [22]) Let Γ be a code of length ℓ over Σ, and $r, c > 0$ be integers. If*

$$\rho(\Gamma) < \frac{\ell - r}{c^2}$$

then Γ is a c-$TA_q(\ell, |\Gamma|; r)$.

For these codes, given a pirate word x, to trace a colluder, the codeword u with $\max_{u\in\Gamma}\rho(u,x)$ must be found. The tracing algorithm in general is an exhaustive search and the cost of search grows exponentially with the code dimension.

GRS codes and list decoding

Generalized Reed-Solomon codes (GRS codes) are defined in [29]. Let F_q be a field of q elements, $\ell \le q$ be an integer, $\alpha_1, \alpha_2, \cdots, \alpha_\ell \in F_q$ be distinct elements, $v_1, v_2, \cdots, v_\ell \in F_q$ be non-zero elements. Write $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_\ell)$ and $v = (v_1, v_2, \cdots, v_\ell)$. A $(k+1)$-dimension GRS code is the set of all vectors

$$(v_1 f(\alpha_1),\ v_2 f(\alpha_2),\ \cdots,\ v_\ell f(\alpha_\ell))$$

where f runs over all polynomials with $\deg(f) \le k$ over F_q. This code is denoted by $\mathrm{GRS}_{k+1}(\alpha, v)$. Sudan and Guruswami [28, 11] gave an elegant list decoding algorithm for GRS codes. Let

$$F_q[x]_k = \{f(x) : f(x) \text{ is a polynomial over } F_q \text{ with } \deg(f) \le k\}.$$

Theorem 2.2 *([11]) Let Γ be a $GRS_{k+1}(\alpha, v)$, and k, ℓ, t be integers such that*

$$\ell \ge \log q, \quad t > \sqrt{k\ell}, \tag{1}$$

$(x_1, y_1), \cdots, (x_\ell, y_\ell) \in F_q^2$ be given. There is an algorithm which outputs all $p(x) \in F_q[x]_k$ satisfying $y_i = p(x_i)$ for at least t values of $i = 1, 2, \cdots, \ell$ and the running time is

$$O\left(\max\left\{\frac{k^3\ell^6 t^6}{(t^2 - k\ell)^6}, \frac{t^6}{k^3}\right\}\right).$$

In a $\mathrm{GRS}_{k+1}(\alpha, v)$ code, any $(k+1)$-tuple in $k+1$ chosen positions determines a unique codeword. In the following we show that a substring of length $2k+2$ determines a unique codeword.

2.2 Deletion correcting capability for GRS codes

Let Γ be a $\mathrm{GRS}_{k+1}(\alpha, v)$ code of length ℓ where $\alpha_1, \alpha_2, \cdots, \alpha_\ell \in F_q$ are ℓ distinct elements, $v_1, v_2, \cdots, v_\ell \in F_q^*$ are ℓ non-zero elements. We will show that Γ, for certain choice of α and v, satisfies the condition of Theorem 2.1 and hence is a c-$\mathrm{TA}_q(\ell, |\Gamma|; r)$ and determine c and r. The main result of this section is given in Theorem 2.4 which can be proved using Lemma 2.1 and Theorem 2.3 (Proofs of this Lemma and Theorem are omitted due to space limitation).

Suppose $\ell > 2k+2$. Let $I = \{i_1, i_2, \cdots, i_{2k+2}\}$, $I' = \{i'_1, i'_2, \cdots, i'_{2k+2}\}$ be two $(2k+2)$-subsets of $\{1, 2, \cdots, \ell\}$ such that

$$\begin{cases} 1 \le i_1 < i_2 < \cdots < i_{2k+2} \le \ell \\ 1 \le i'_1 < i'_2 < \cdots < i'_{2k+2} \le \ell \\ i_j = i'_j \quad \text{for at most } k \text{ values of } j \in \{1, 2, \cdots, 2k+2\} \end{cases} \tag{2}$$

Consider the following $K \times K$ matrix

$$\begin{pmatrix} v_{i_1} & v_{i_1}\alpha_{i_1} & v_{i_1}\alpha_{i_1}^2 & \cdots & v_{i_1}\alpha_{i_1}^k & v_{i'_1} & v_{i'_1}\alpha_{i'_1} & v_{i'_1}\alpha_{i'_1}^2 & \cdots & v_{i'_1}\alpha_{i'_1}^k \\ v_{i_2} & v_{i_2}\alpha_{i_2} & v_{i_2}\alpha_{i_2}^2 & \cdots & v_{i_2}\alpha_{i_2}^k & v_{i'_2} & v_{i'_2}\alpha_{i'_2} & v_{i'_2}\alpha_{i'_2}^2 & \cdots & v_{i'_2}\alpha_{i'_2}^k \\ v_{i_3} & v_{i_3}\alpha_{i_3} & v_{i_3}\alpha_{i_3}^2 & \cdots & v_{i_3}\alpha_{i_3}^k & v_{i'_3} & v_{i'_3}\alpha_{i'_3} & v_{i'_3}\alpha_{i'_3}^2 & \cdots & v_{i'_3}\alpha_{i'_3}^k \\ \vdots & & & & & \vdots & & & & \\ v_{i_K} & v_{i_K}\alpha_{i_K} & v_{i_K}\alpha_{i_K}^2 & \cdots & v_{i_K}\alpha_{i_K}^k & v_{i'_K} & v_{i'_K}\alpha_{i'_K} & v_{i'_K}\alpha_{i'_K}^2 & \cdots & v_{i'_K}\alpha_{i'_K}^k \end{pmatrix} \tag{3}$$

where $K = 2k+2$.

Lemma 2.1 *Let I and I' be two $(2k+2)$-sets satisfying (2), and the rank of (3) be $2k+2$. Then there are no non-zero polynomials $f, g \in F_q[x]_k$, $f \neq g$, such that*

$$(v_{i_1} f(\alpha_{i_1}), v_{i_2} f(\alpha_{i_2}), \cdots, v_{i_{2k+2}} f(\alpha_{i_{2k+2}})) = (v_{i'_1} g(\alpha_{i'_1}), v_{i'_2} g(\alpha_{i'_2}), \cdots, v_{i'_{2k+2}} g(\alpha_{i'_{2k+2}})) \tag{4}$$

Theorem 2.3 *Let $GRS_{k+1}(\alpha, v)$ code of length $\ell > 2k+2$ over F_q be given. If α and v satisfy that*

$$\text{the rank of (3) for any two } (2k+2)\text{-sets } I, I' \text{ satisfying (2) is } 2k+2 \tag{5}$$

then $\rho(\Gamma) \leq 2k+1$.

Theorem 2.3 bounds the length of the longest common subwords of any two codewords of Γ and can be used to determine deletion correcting capability of the code.

Theorem 2.4 *Let a $GRS_{k+1}(\alpha, v)$ code of length $\ell > 2k+2$ be given. If α and v satisfy that (5), then $GRS_{k+1}(\alpha, v)$ is a c-$TA_q(\ell, q^k; r)$ and*

$$r < \ell - (2k+1)c^2.$$

2.3 Decoding a shortened word

Theorem 2.4 shows that using a $\mathrm{GRS}_{k+1}(\alpha, v)$ code for fingerprinting allows a pirate word of length at least $(2k+1)c^2 + 1$ to be traced to one of the c colluders. Using Theorem 2.1, the tracing algorithm has to find the codeword that has a common substring of length at least equal to $(\ell - r)/c$ with the pirate word. As noted before the cost of an exhaustive search will grow exponentially with the code dimension. In this section we formulate deletion decoding problem of a shortened pirate word as an error-correction problem and in Section 2.4, will use the list decoding algorithm of [11] to find at least one of the colluders.

Let Γ be a c-$\mathrm{TA}_q(\ell, n; r)$ code and assume a shortened pirate word,

$$y = (y_1, y_2, \cdots, y_{\ell - r}) \in desc(C; r) \tag{6}$$

is given. Then y is a subword of $z \in desc(C)$ and is obtained by r deletions from z. However, the positions where deletions have occurred are not known. Denote by Y_j the set of possible values of the j^{th} component of z, for $j = 1, 2, \cdots, \ell$. Then we have,

$$\begin{aligned} Y_j &= \{y_j, y_{j-1}, \cdots, y_1\}, \quad \text{for } j \in \{1, 2, \cdots, r\} \\ Y_j &= \{y_j, y_{j-1}, \cdots, y_{j-r}\}, \quad \text{for } j \in \{r+1, r+2, \cdots, \ell - r\} \\ Y_j &= \{y_{\ell - r}, y_{\ell - r - 1}, \cdots, y_{j-r}\}, \quad \text{for } j \in \{\ell - r + 1, \ell - r + 2, \cdots, \ell\} \end{aligned} \tag{7}$$

The sets $Y_1, Y_2, \cdots, Y_\ell$ define a set Z of words

$$Z = \{(z_1, z_2, \cdots, z_\ell) \in \Sigma^\ell : z_j \in Y_j, 1 \leq j \leq \ell\}$$

which contains words having y as a subword and could result in y. Not all elements of Z are in $desc(C)$. Tracing problem is to find a codeword u which is close (Hamming distance) to some $z \in Z \cap desc(C)$.

2.4 Tracing algorithm

We use the list decoding algorithm of GRS codes to give an efficient tracing algorithm. Let Γ be a $\mathrm{GRS}_{k+1}(\alpha, v)$ code of length ℓ over F_q with α, v satisfy (5). It is known in [22] that if a collusion $C \subseteq \Gamma$, $|C| \leq c$, produces a sequence $y \in desc(C; r)$, then there exists a codeword $u \in C$ such that

$$\rho(y, u) \geq \frac{\ell - r}{c} \tag{8}$$

and $\rho(y, v) < \rho(y, u)$ for all $v \in \Gamma \setminus C$. That is a vector $u \in \Gamma$ satisfying (8) is a member of C. For a $\mathrm{GRS}_{k+1}(\alpha, v)$, (8) gives the fact that there exist $\alpha_{i_1}, \alpha_{i_2}, \cdots, \alpha_{i_{\ell-r}}$ such that at least $(\ell - r)/c$ of the following equations

$$f(\alpha_{i_1}) = y_1,\ f(\alpha_{i_2}) = y_2,\ \cdots,\ f(\alpha_{i_{\ell-r}}) = y_{\ell-r} \tag{9}$$

are satisfied, where $f \in F_q[x]_k$ is the polynomial corresponding to u. The tracing algorithm for a shortened pirate word $y \in desc(C; r)$ outputs polynomials that satisfy at least $(\ell - r)/c$ equations in (9). We will use the list decoding algorithm in [11] to solve this problem.

For $y \in desc(C; r)$, we define a set

$$\begin{aligned} S(y) &= \left(\cup_{j=1}^{r} \{(\alpha_j, y_j), (\alpha_j, y_{j-1}), \cdots, (\alpha_j, y_1)\}\right) \\ &\bigcup \left(\cup_{j=r+1}^{\ell-r} \{(\alpha_j, y_j), (\alpha_j, y_{j-1}), \cdots, (\alpha_j, y_{j-r})\}\right) \\ &\bigcup \left(\cup_{j=\ell-r+1}^{\ell} \{(\alpha_j, y_{\ell-r}), (\alpha_j, y_{\ell-r-1}), \cdots, (\alpha_j, y_{j-r})\}\right) \end{aligned} \tag{10}$$

$S(y)$ is the set of all possible points in all polynomials (codewords) that have y as a subword. This is obtained by considering all possible values of such polynomials at point α_j, $j = 1, 2, \cdots, \ell$ as was shown in the set Y_j in (7). It is easy to see that the following holds.

$$\begin{aligned} |S(y)| &\leq \sum_{j=1}^{r} j + (\ell - 2r)(r + 1) + \sum_{j=1}^{r} j \\ &= (\ell - r)(r + 1) \end{aligned} \tag{11}$$

Now a colluder has a polynomial that passes through at least $(\ell - r)/c$ points of $S(y)$. Note that a polynomial can only go through one of the points of the form $(\alpha_j, y_{j'})$, $y_{j'} \in Y_j$ for every j. We use list decoding algorithm of [11] to find these polynomials.

Theorem 2.5 *Let a $GRS_{k+1}(\alpha, v)$ code of length $\ell > 2k + 2$ over F_q be given, where α, v satisfy (5), r and c be integers such that*

$$r < \min\left\{\frac{\ell - c^2 k}{c^2 k + 1},\ \ell - (2k + 1)c^2\right\} \tag{12}$$

There is an algorithm, for which on any input $y = y_1\, y_2 \cdots y_{\ell'} \in desc(C; r)$, it outputs the following list: $\{u \in C : \rho(u, y) \geq \ell'/c\}$. The running time is

$$O\left(max\left\{\frac{k^3 |S(y)|^6 \ell'^6}{(t^2 - k|S(y)|)^6 c^6},\ \frac{\ell'^6}{k^3 c^6}\right\}\right).$$

Theorem 2.5 shows that tracing is a polynomial time algorithm. This is a major improvement compared to the exponential time required by brute force decoding algorithm.

3 Soft-Decision Decoding

In this section we consider corrupted fingerprints. We assume up to c colluders construct a pirate object by first applying cut-and-paste (but not cropping) to their copies (hence resulting in a pirate sequence that is in the feasible set of the colluders) and then trying to weaken the marks using averaging or other attacks that make the mark detection less reliable. We note that if the collusion size is not large and so the marks cannot be completely removed, the above strategy will have a good chance of creating an untraceable object. This is because using the traditional method of mark detection followed by tracing will fail if marks are not correctly detected.

We propose a *combined mark detection and tracing* algorithm that increases the chance of successful tracing by carrying the useful information from the mark detection stage to the tracing stage. Traditional mark detection methods are *hard-detection* and in each position output the symbol with the maximum probability. We consider mark detectors that have a 'soft' output. That is instead of a single mark for each position, they output a *reliability vector* for that position and so the input to the tracing algorithm is a $q \times \ell$ reliability matrix. A reliability matrix, also called a *pirate matrix*, constructed by a group of c colluder will have each of its columns contributed by one colluder. We assume a c-TA code is used for fingerprinting and the tracing algorithm uses minimum Hamming distance for tracing. Replacing 'hard-detected' pirate words with reliability matrices is similar to *soft-decision* decoding in error-correcting codes where the channel output is a reliability matrix and the soft-decision decoding algorithm is used to find the most likely codeword [16]. However, in the case of tracing the aim is to find a *definite* pirate and there is no guarantee that the output of a soft-decision decoder is a colluder. It is worth noting that using minimum Hamming distance for tracing in c-TA codes, guarantees correct tracing of a colluder because the number of symbols contributed by the colluder who is closest to the pirate word is bigger than a certain bound. In another words, success of tracing is not due to the maximum likelihood property of minimum distance decoding.

The remaining question is whether it is possible to use the information in the reliability matrix to have a 'more powerful' tracing algorithm. We answer this question in affirmative by firstly giving a definition of 'more powerful' in terms of the size of the set of pirate matrices that are traceable, and then showing an algorithm that compared to the traditional hard-detection tracing is more powerful.

3.1 Background

Soft-decision decoding was proposed by Forney [8], and later developed by numerous authors. Koetter et al [17, 16] extended the list decoding algorithm of Gurswami and Sudan [11] to give an algebraic soft decoding algorithm for RS codes. In this section we review soft-decision decoding of [16] and characterize a set of reliability matrices that will be decoded to a unique codeword u using the generalized distance measure (or similarity) used in [16].

Let $\Sigma = \{1, 2, \cdots, q\}$ be an alphabet and Γ be a code of length ℓ over Σ. A $q \times \ell$ matrix $\Pi = (\pi_{i,j})_{1 \le i \le q, 1 \le j \le \ell}$ is called a *reliability matrix*, or a *pirate matrix*, if $\sum_{1 \le i \le q} \pi_{i,j} = 1$ for every j, $1 \le j \le \ell$. The inner product of two $q \times \ell$ matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ is defined as

$$\langle A, B \rangle = \sum_{i=1}^{q} \sum_{j=1}^{\ell} a_{i,j} b_{i,j} \tag{13}$$

A vector $x = (x_1, x_2, \cdots, x_\ell) \in \Sigma^\ell$ can be identified with a $q \times \ell$ matrix in which the entry (x_j, j) is 1 and the entry (i, j) is 0 for $i \neq x_j$, for every $j, 1 \le j \le \ell$. This matrix is called *exact matrix* of x. We use x to denote a q-ary word and its exact matrix both when dealing with inner product. The inner product of Π

and the word x using its matrix representation and (13) is as follows.

$$\langle \Pi, x \rangle = \sum_{j=1}^{\ell} \pi_{x_j,j}$$

The inner product gives a measure of similarity between the two matrices. For the special case that the two matrices are from two q-ary words x_1 and x_2, $\langle x_1, x_2 \rangle = \ell - d_H(x_1, x_2)$ which is the number of their common components and $\langle x, x \rangle = \ell$. This measure is used in [17, 16] to give an algebraic soft-decision decoding algorithm that is optimal and minimizes the chance of error-decoding. Given a reliability matrix Π, the output of the decoder is vector(s) that maximize $\langle \Pi, x \rangle$.

Theorem 3.1 *(Theorem 12, [17]) The algebraic soft-decision decoding algorithm outputs a list of codewords consisting of $u \in \Gamma$ satisfying*

$$\frac{\langle \Pi, [u] \rangle}{\sqrt{\langle \Pi, \Pi \rangle}} \geq \sqrt{k+1} + o(1)$$

where $o(1)$ denotes a function of an integer s that tends to zero as $s \to \infty$.

3.2 Uniquely decodable matrices

For a reliability matrix $\Pi = (\pi_{i,j})$, let

$$\begin{aligned} \pi_j^{max} &= \max_{1 \leq i \leq q} \pi_{i,j} \\ \bar{\pi}_j^{max} &= \max_{i:\pi_{i,j} \neq \pi_j^{max}} \pi_{i,j} \\ E_\Pi &= \{j : \exists\, i_1 \neq i_2, \pi_{i_1,j} = \pi_{i_2,j} = \pi_j^{max}\} \end{aligned}$$

that is π_j^{max} is the maximum element in column j, $\bar{\pi}_j^{max}$ is the next biggest element (strictly less than the maximum) in that column, and E_Π is the set of columns that more than one element has the maximum value and corresponds to an ambiguous position. Although we will use this set in our proofs, without loss of generality we can ignore it in comparing our result with hard-detection case because erased positions do not provide any information in either case.

A column in the reliability matrix is a probability distribution on the mark set. More uniform distribution in a column corresponds to more ambiguity in detecting a mark. A column with uniform distribution corresponds to an erased position and a column with more than one symbol with maximum probability corresponds to an ambiguous position.

We say a word $u = (u_1, u_2, \cdots, u_\ell)$ *matches* a reliability matrix Π in column j if $\pi_{u_j,j}$ is maximum in column j of Π. Consider a code Γ with minimum Hamming distance d. A reliability matrix Π is *faithful* to a codeword $u \in \Gamma$ if there is a set $S_\Pi \subseteq \{1, 2, \cdots, \ell\}$ that satisfies (A1), (A2) and (A3) given below.

(A1) $|S_\Pi| > \ell - d + |E_\Pi|$;

(A2) for each $s \in S_\Pi$, $\pi_{u_s,s} = \pi_s^{max}$;

(A3) for each $s \in S_\Pi \setminus E_\Pi$,

$$\pi_{u_s,s} - \bar{\pi}_s^{max} > \frac{1}{d + |S_\Pi| - \ell - |E_\Pi|} \sum_{j \notin S_\Pi} (\pi_j^{max} - \pi_{u_j,j})$$

Informally, a matrix that is faithful to a codeword has at least $\ell - d$ columns that 'match' u and, in each of these places the 'strength' $\pi_s^{max} - \bar{\pi}_s^{max}$ of the match is higher than a certain threshold. Denote by Φ_u the set of reliability matrices that are faithful to codeword u.

The following theorem shows that all matrices that are faithful to a codeword will be decoded to that word if soft-decision decoding based on inner product is used, and that codeword is unique.

Theorem 3.2 *Let Γ be a code and $\Pi = (\pi_{i,j})$ a reliability matrix. Then Π is faithful to at most one codeword $u \in \Gamma$.*

Proof: Suppose Π is faithful to $u = (u_1, u_2, \cdots, u_\ell) \in \Gamma$ and S is a set corresponding to (A1), (A2) and (A3). Let $v = (v_1, v_2, \cdots, v_\ell) \in \Gamma$ be an arbitrary codeword. Define subsets $J_0, J_e, J_u, J_v \subseteq \{1, 2, \cdots, \ell\}$ as follows.

$$\begin{aligned} J_0 &= \{j : u_j = v_j\} \\ J_e &= \{j : u_j \neq v_j, \pi_{u_j,j} = \pi_{v_j,j}\} \\ J_u &= \{j : u_j \neq v_j, \pi_{u_j,j} > \pi_{v_j,j}\} \\ J_v &= \{j : u_j \neq v_j, \pi_{v_j,j} > \pi_{u_j,j}\} \end{aligned}$$

The sets $\{J_0, J_e, J_u, J_v\}$ is a partition of $\{1, 2, \cdots, \ell\}$, and $S \subseteq J_0 \cup J_e \cup J_u$, $J_v \subseteq \{1, 2, \cdots, \ell\} \setminus S$. It follows that $S = (S \cap J_0) \cup (S \cap J_e) \cup (S \cap J_u)$, and we have

$$|S| = |S \cap J_0| + |S \cap J_e| + |S \cap J_u| \tag{14}$$

Note that $|S \cap J_0| \leq |J_0| = \lambda(u, v) = \ell - d_H(u, v)$, and $S \cap J_e \subseteq E_\Pi$. That is $|S \cap J_e| \leq |E_\Pi|$. Then equality (14) gives

$$\begin{aligned} |S \cap J_u| &= |S| - |S \cap J_0| - |S \cap J_e| \\ &\geq |S| - (\ell - d_H(u, v)) - |E_\Pi| \end{aligned} \tag{15}$$

Since Π is faithful to u, for every $s \in S \setminus E_\Pi$ we have

$$\pi_{u_s,s} - \bar{\pi}_s^{max} > \frac{1}{d + |S| - \ell - |E_\Pi|} \sum_{j \notin S} (\pi_j^{max} - \pi_{u_j,j})$$

This implies that

$$\begin{aligned} \sum_{s \in S \cap J_u} (\pi_{u_s,s} - \bar{\pi}_s^{max}) &> \sum_{s \in S \cap J_u} \left(\frac{1}{d + |S| - \ell - |E_\Pi|} \sum_{j \notin S} (\pi_j^{max} - \pi_{u_j,j}) \right) \\ &= |S \cap J_u| \cdot \frac{1}{d + |S| - \ell - |E_\Pi|} \sum_{j \notin S} (\pi_j^{max} - \pi_{u_j,j}) \\ &\geq \sum_{j \notin S} (\pi_j^{max} - \pi_{u_j,j}), \quad \text{(from (15))} \\ &\geq \sum_{j \in J_v} (\pi_j^{max} - \pi_{u_j,j}) \end{aligned} \tag{16}$$

The last inequality is because of $J_v \subseteq \{1, 2, \cdots, \ell\} \setminus S$. By definitions of S and J_u, we know that $\pi_{v_s,s} < \pi_{u_s,s} = \pi_s^{max}$ for $s \in S \cap J_u$ and hence $\pi_{v_s,s} \leq \bar{\pi}_s^{max}$, and so $\sum_{s \in S \cap J_u} (\pi_{u_s,s} - \pi_{v_s,s}) > \sum_{s \in S \cap J_u} (\pi_{u_s,s} - \bar{\pi}_s^{max})$ follows. From (16) we obtain that

$$\sum_{s \in S \cap J_u} (\pi_{u_s,s} - \pi_{v_s,s}) > \sum_{j \in J_v} (\pi_{v_j,j} - \pi_{u_j,j})$$

Then we have

$$\sum_{j\in J_u}(\pi_{u_j,j}-\pi_{v_j,j}) \geq \sum_{s\in S\cap J_u}(\pi_{u_s,s}-\pi_{v_s,s}) > \sum_{j\in J_v}(\pi_{v_j,j}-\pi_{u_j,j})$$

That is

$$\sum_{j\in J_u}\pi_{u_j,j}+\sum_{j\in J_v}\pi_{u_j,j} > \sum_{j\in J_u}\pi_{v_j,j}+\sum_{j\in J_v}\pi_{v_j,j} \tag{17}$$

By definition

$$\begin{aligned}\langle \Pi, u\rangle &= \sum_{j\in J_0}\pi_{u_j,j}+\sum_{j\in J_e}\pi_{u_j,j}+\sum_{j\in J_u}\pi_{u_j,j}+\sum_{j\in J_v}\pi_{u_j,j}\\ \langle \Pi, v\rangle &= \sum_{j\in J_0}\pi_{v_j,j}+\sum_{j\in J_e}\pi_{v_j,j}+\sum_{j\in J_u}\pi_{v_j,j}+\sum_{j\in J_v}\pi_{v_j,j}\end{aligned}$$

When $j \in J_0 \cup J_e$, $\pi_{u_j,j} = \pi_{v_j,j}$. So (17) implies that $\langle \Pi, u\rangle > \langle \Pi, v\rangle$.

□

Theorem 3.2 shows that a reliability matrix Π can be faithful to only one codeword. A geometric interpretation of the above result as suggested in [16], can be obtained by defining the angle $\theta(A, B)$ between two matrices $A = (a_{i,j})$ and $B = (b_{i,j})$ as follows,

$$\cos\theta(A,B) = \frac{\langle A, B\rangle}{\sqrt{\langle A, A\rangle}\sqrt{\langle B, B\rangle}}$$

The following corollary is the direct result of the above theorem and definition (13).

Corollary 3.1 *Let Γ be a code and $\Pi = (\pi_{i,j})$ a reliability matrix. If Π is faithful to a codeword $u \in \Gamma$, then $\theta(\Pi, u) < \theta(\Pi, v)$ for all $v \in \Gamma \setminus \{u\}$.*

In hard decision decoding, given a received word $x = (x_1, x_2, \cdots, x_\ell) \in \Sigma^\ell$, there is at most one codeword u in the ball $B_{d/2}(x) = \{w \in \Sigma^\ell : d_H(x, w) < d/2\}$. Faithful matrices generalize this ball when soft-decision is used. The following corollary of Theorem 3.2 is a re-statement of the known result in hard decision decoding of codes with minimum Hamming distance d.

Corollary 3.2 *Let $x = (x_1, x_2, \cdots, x_\ell) \in \Sigma^\ell$, Π be the exact matrix of x. Denote by $\mathcal{B}(\Pi) = \{w \in \Sigma^\ell :$ Π is faithful to $w\}$. Then $B_{d/2}(x) \subseteq \mathcal{B}(\Pi)$.*

Corollary 3.2 shows that if x has at least $\ell - d/2$ in common with a codeword (distance at most $(d-1)/2$), the codeword is unique and that hard-decision decoding using Hamming distance is a special case of soft-decision decoding using inner product as the metric.

4 Tracing Colluders

Reliability matrices represent the set of all possible outputs of the mark detection stage including the cases that the fingerprint is damaged beyond tracing. For example if the pirate matrix has uniform or nearly uniform distribution in all columns, the fingerprint has effectively been removed. A pirate matrix Π can be always converted to the most likely pirate word by replacing each column with the symbol that has the maximum probability in that column. This word is called the *hard-detected word* associated with Π. If marks are not damaged beyond recovery, this hard detected word will be a descendant of the codewords

held by the colluders and can be correctly traced. Traditional hard-detection tracing implicitly assumes that the marks are correctly recovered.

Let $\mathcal{P}$ denote the set of all pirate matrices. For a tracing algorithm A, denote by $\mathcal{A}_c \subset \mathcal{P}$ the set of all matrices that can be traced to a colluder. The size of the set $\mathcal{A}_c$ is an indication of the effectiveness of the tracing algorithm. A more powerful algorithm can trace a larger set of pirate matrices. Without loss of generality, for a c-TA codes obtained from error-correcting codes with minimum distance d, we assume the code cannot tolerate any erased symbols and tracing succeeds if all marks in the fingerprint are correctly detected. Note that c-TA codes may tolerate erasure but this directly translates into higher minimum distance or lower c. Let $\mathcal{D}_c(\Gamma)$ denote the set of all descendants of all colluding sets of size at most c, and $\mathcal{R}_c(\Gamma)$ denote the set of reliability matrices whose hard-detected words are in the set $\mathcal{D}_c(\Gamma)$. This is the set of reliability matrices that can be correctly traced using hard-detection tracing. In the following we will show that using soft-detection allows another set of pirate matrices to be correctly traced. This set has non-empty overlap with $\mathcal{R}_c(\Gamma)$ but includes many matrices that are not in $\mathcal{R}_c(\Gamma)$ and so the set of traceable reliability matrices has been effectively enlarged.

Reliability matrices obtained from colluding groups

We show that a reliability matrix that is produced by a collusion of size at most c and is faithful to a codeword $u \in \Gamma$ can be correctly traced to u. For a collusion $C \subseteq \Gamma$, define the following set:

$$T(C) = \{\Pi : \Pi \text{ is faithful to a } u \in C \text{ with a set } S_\Pi \text{ of matched columns, } |S_\Pi| \geq \ell/c, \text{ and } |E_\Pi| \leq e\} \quad (18)$$

The following theorem shows that the inner product $\langle \Pi, u \rangle$ can be used to correctly trace all matrices in $T(C)$.

Theorem 4.1 *Let Γ be a code of length ℓ and minimum Hamming distance d over Σ, $C \subseteq \Gamma$, $|C| \leq c$, and c and e be non-negative integers. Then $\Pi \in T(C)$ can be traced if*

$$c < \frac{\ell}{\ell - d + e} \quad (19)$$

4.1 Tracing algorithm for soft-detection tracing

Let $\mathcal{T}_c(\Gamma) = \cup_{C,|C|\leq c} T(C)$. Theorem 4.1 shows that using inner product to measure similarity of a reliability matrix $\Pi \in \mathcal{T}_c(\Gamma)$ and codewords can correctly trace a colluder. We note that $\Pi \in \mathcal{T}_c(\Gamma)$ means that Π is faithful to one of the codewords but its hard-detected word might not be in $\mathcal{D}_c(\Gamma)$. Faithful matrices have high similarity with one of the codewords but they might contain columns that are constructed by other colluders and correspond to undetectable marks. Also note that $\Pi \in \mathcal{R}_c(\Gamma)$ might not be faithful to any codeword as although there will be a codeword with at least ℓ/c matches but the strength of the match might be less than what is required by faithfulness property. To summarize, we have $\mathcal{T}_c(\Gamma) \not\subset \mathcal{R}_c(\Gamma)$ and $\mathcal{R}_c(\Gamma) \not\subset \mathcal{T}_c(\Gamma)$.

Matrices in $\mathcal{R}_c(\Gamma)$ can be hard-detected and traced. For a reliability matrix $\Pi \in \mathcal{T}_c(\Gamma)$, the tracing algorithm must find the codeword u which maximizes $\langle \Pi, u \rangle$ over all codewords. Finding u in general will be computationally expensive with the computational cost growing exponentially with the code dimension $(\log N)$. However using RS codes as the c-TA code allows us to use the soft-decision decoding of [16] to find this codeword and there will be an efficient tracing algorithm. According to theorem 3.1 the output of the soft-decision algorithm will be a list of codewords for which the inner product is above $\sqrt{k+1} + o(1)$. Among these codewords there will be a unique codeword to which the pirate matrix will be faithful (satisfy (A1), (A2) and (A3)).

Hence, the soft-tracing algorithm will have two steps.

1. Use soft-decision decoding to find a list of suspects.
2. Examine each vector in the suspect list to identify the traitor.

4.2 Codes for soft-tracing

In the above we assume that the fingerprinting code is a c-TA code. It was proved [25] that an error-correcting code of length ℓ with the minimum Hamming distance d is a c-TA code as long as $c^2 < \ell/(\ell - d)$. From (19) we have $c < \ell/(\ell - d)$. This means that soft-tracing can trace larger collusions for the same code.

This improvement is because of requiring that the 'strength' of a mark, given by $\pi_{u_s,s} - \bar{\pi}_s$ in a 'matched position' to be higher than

$$\frac{1}{d + |S| - \ell - |E_\Pi|} \sum_{j \notin S} (\pi_j^{max} - \bar{\pi}_j^{max})$$

Hence, the larger $|S|$ requires the smaller strength for matched positions.

5 Conclusion

Tracing pirate media objects by fingerprinting the object can only succeed if we limit the range of possible attacks. We considered a number of attacks that could be used by colluders and gave efficient tracing algorithms that can trace one of at most c colluders if corruption of the pirate fingerprint is below certain level. In this paper we considered the widest (compared to other works) range of attacks against fingerprinting sequences.

The main contributions of this paper are (i) efficient tracing of shortened fingerprints and (ii) efficient tracing of a larger class of corrupted fingerprints.

Shortened fingerprints had already been considered but the construction of codes that protect against r deletions, and efficient tracing of shortened fingerprints had been open problems. Our results in (i) provide solutions to both these problems.

For corrupted fingerprints, that is fingerprints for which mark detection stage can not be very reliable, we proposed a combined mark detection and tracing approach to allow useful information from mark detection stage to be used in the tracing stage. Using pirate matrices for tracing provides a generalized setting for tracing and allows various tracing algorithms to be compared in terms of the subset of matrices that they can trace. We showed that the traditional method of hard-detection followed by tracing, referred to as hard-tracing, can be complemented by soft-tracing which allows the subset of pirate matrices that are faithful to a codeword to be traced. Such matrices, although might not be traceable by hard-tracing, but have such a strong similarity to one of the codewords that is possible to correctly trace them. Our definition of similarity follows that used for a recently proposed soft-decision decoding algorithm for RS codes, and so we can use this decoding algorithm to give an efficient soft-decision tracing for faithful matrices. These results can also be extended to other codes such as AG codes that can use this algorithm.

An interesting open question is to find other tracing algorithms that can trace a larger set of pirate matrices. We left combining (i) and (ii) as an open problem. That is, allowing the pirate fingerprint to be shorter than the original one and have corrupted marks and so representable by a pirate matrix with less than ℓ columns.

Finally, characterization of the class of GRS codes that can be used for deletion correction is an open problem.

Acknowledgement The authors would like to thank Luke McAven for constructing examples of GRS codes with deletion correcting property and indicating some errors in an earlier version of this paper.

References

[1] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. In *Advances in Cryptology - CRYPTO'95, Lecture Notes in Computer Science*, volume 963, pages 453–465. Springer-Verlag, Berlin, Heidelberg, New York, 1995.

[2] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, Vol. 44, No. 5:1897–1905, 1998.

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

[4] I. Cox, M. Miller, and J. Bloom. *Digital Watermarking.* Mulimedia Information and Systems. Morgan Kaufman Publishers, 2002.

[5] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, and J. Ueberberg. Combining digital watermarks and collusion secure fingerprinting for digital images. In *Proceedings of SPIE*, volume 3657, pages 171–182, 1999.

[6] F. Ergun, J. Kilian, and R. Kumar. A note on the limits of collusion-resistant watermarks. In *Advances in Cryptology - EUROCRYPT'99, Lecture Notes in Computer Science*, volume 1592, pages 140–149. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[7] A. Fiat and T. Tassa. Dynamic traitor tracing. In *Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science*, volume 1666, pages 354–371. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[8] Jr. G. D. Forney. Generalized minimum distance decoding. *IEEE Transactions on Information Theory*, Vol. IT-12, No. 2:125–131, 1966.

[9] E. Gafni, J. Staddon, and Y. L. Yin. Efficient methods for integrating traceability and broadcast encryption. In *Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science*, volume 1666, pages 372–387. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[10] V. Guruswami. List decoding of error-correcting codes. Ph.D Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, USA, 2001.

[11] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, Vol. 45, No. 6:1757–1767, 1999.

[12] H. Guth and B. Pfitzmann. Error- and collusion-secure fingerprinting for digital data. In *Information Hiding'99, Lecture Notes in Computer Science*, volume 1768, pages 134–145. Springer-Verlag, Berlin, Heidelberg, New York, 2000.

[13] H. D. L. Hollmann, J. H. van Lint, J. P. Linnartz, and L. M. G. M. Tolhuizen. On codes with the identifiable parent property. *Journal of Combinatorial Theory, Series A*, 82:121–133, 1998.

[14] A. Kiayias and M. Yung. Self protecting pirates and black-box traitor tracing. In *Advances in Cryptology - CRYPTO'01, Lecture Notes in Computer Science*, volume 2139, pages 63–79. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[15] A. Kiayias and M. Yung. On crafty pirate and foxy tracers. In *Security and Privacy in Digital Rights Management (SPDRM 2001), Lecture Notes in Computer Science*, volume 2320, pages 22–39. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[16] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. preprint, 2000.

[17] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. In *International Symposium on Information Theory (ISIT 2000)*, page 61, 2000.

[18] K. Kurosawa and Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Advances in Cryptology - EUROCRYPT'98, Lecture Notes in Computer Science*, volume 1462, pages 502–517. Springer-Verlag, Berlin, Heidelberg, New York, 1998.

[19] B. Pfitzmann. Trials of traced traitors. In *Information Hiding, Lecture Notes in Computer Science*, volume 1174, pages 49–64. Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[20] R. Safavi-Naini and Y. Wang. Sequential traitor tracing. In *Advances in Cryptology - CRYPTO 2000, Lecture Notes in Computer Science*, volume 1880, pages 316–332. Springer-Verlag, Berlin, Heidelberg, New York, 2000.

[21] R. Safavi-Naini and Y. Wang. New results on frameproof codes and traceability schemes. *IEEE Transactions on Information Theory*, Vol. 47, No. 7:3029–3033, 2001.

[22] R. Safavi-Naini and Y. Wang. Collusion secure q-ary fingerprinting for perceptual content. In *Security and Privacy in Digital Rights Management (SPDRM 2001), Lecture Notes in Computer Science*, volume 2320, pages 57–75. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[23] P. Sarkar and D. R. Stinson. Frameproof and IPP codes. In *Progress in Cryptology - INDOCRYPT 2001, Lecture Notes in Computer Science*, volume 2247, pages 117–126. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[24] A. Silverberg, J. Staddon, and J. Walker. Efficient traitor tracing algorithms using list decoding. In *Advances in Cryptology - ASIACRYPT'01, Lecture Notes in Computer Science*, volume 2248, pages 175–192. Springer-Verlag, Berlin, Heidelberg, New York, 2001.

[25] J. N. Staddon, D. R. Stinson, and R. Wei. Combinatorial properties of frameproof and traceability codes. *IEEE transactions on information theory*, Vol. 47, No. 3:1042–1049, 2001.

[26] D. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes. *SIAM Journal on Discrete Mathematics*, 11:41–53, 1998.

[27] D. R. Stinson and R. Wei. Key preassigned traceability schemes for broadcast encryption. In *Proceedings of SAC'98, Lecture Notes in Computer Science*, volume 1556, pages 144–156. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[28] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13:180–193, 1997.

[29] J. H. van Lint. *Introduction to Coding Theory.* Graduate texts in mathematics. Springer-Verlag, New York, 1999.