Applications of List Decoding to Tracing Traitors

Alice Silverberg, Jessica Staddon, *Member, IEEE*and Judy L. Walker, *Member, IEEE*

***Abstract*— We apply results from algebraic coding theory to solve problems in cryptography, by using recent results on list decoding of error-correcting codes to efficiently find traitors who collude to create pirates. We produce schemes for which the TA (traceability) traitor tracing algorithm is very fast. We compare the TA and IPP (identifiable parent property) traitor tracing algorithms, and give evidence that when using an algebraic structure, the ability to trace traitors with the IPP algorithm implies the ability to trace with the TA algorithm. We also demonstrate that list decoding techniques can be used to find all possible pirate coalitions. Finally, we raise some related open questions about linear codes, and suggest uses for other decoding techniques in the presence of additional information about traitor behavior.**

***Index Terms*— Algebraic geometry code, identifiable parent property, list decoding, traceability code, traitor tracing, Reed-Solomon code.**

I. Introduction

An important problem in the protection of intellectual property is how to deter piracy. This leads to the question of how to efficiently trace traitors, i.e., legitimate users who collude to create pirate software, pirate decoder boxes, etc. Traceability schemes were introduced in 1994 in [6] and have been extensively studied in the intervening years. We focus on one of the few aspects of this area of work that has received little attention: the complexity of the traitor tracing algorithms. We show that powerful new techniques for the list decoding of error-correcting codes enable us to construct traceability schemes with very fast traitor tracing algorithms. These schemes guarantee the identification of at least one traitor. We also show that the same techniques can be used to build an algorithm for finding *all* possible traitor coalitions. Over time, this algorithm can be used to accumulate convincing evidence against additional users. Further, we discuss potential applications of other decoding methods to the problem of tracing traitors, suggest alternative approaches when additional information is known about the way the traitors are operating, examine the relationship between two important tracing algorithms, and raise some open questions about linear codes.

An example where our schemes could be applied is a pay-TV scenario where each subscriber is given a sequence of keys, one key for each broadcast segment, and traitors pool their keys to create a pirate decoder. Another example, given in [5], is a movie where different copies have segments filmed from different camera angles. Our scheme would trace pirated copies to colluding owners of copies of the movie.

Our approach takes advantage of recent powerful list decoding methods, which originated with the work of Sudan [19] and were refined in [11]. In list decoding the input is a received word and the output is the list of all codewords within a given Hamming distance of the received word. The results in [19] are not strong enough to be applicable in the setting in which the TA algorithm succeeds in finding traitors (as opposed to only identifying probable traitors), since the decoding procedure in [19] is not capable of correcting enough errors in the code. However, the improvements in [11] are precisely sufficient to be applicable to the setting where the TA algorithm succeeds. An additional advantage of using list decoding is that the associated traitor tracing algorithm gives a list containing one or more traitors, rather than only one. Efficient list decoding algorithms now exist for Reed-Solomon codes, more general algebraic geometry codes, and some concatenated codes. List decoding techniques are receiving wide attention in the coding theory community, and improvements and generalizations are being rapidly produced.

Although error-correcting techniques are used to attain traceability in [3], our paper gives the first applications of list decoding to the problem of traceability. List decoding is applied to the related problem of watermarking in [21].

We note that algebraic geometry codes appear to have been under-utilized in cryptological applications. For example, the results of [17] can be used to give better explicit examples of c-frameproof codes than those obtained in [5]. The codes constructed in [17] are concatenated codes where the outer code is an algebraic geometry code coming from a Hermitian curve, while those used in [5] come from pseudo-random graphs.

II. Background on Codes and Traceability

A. Definitions and Notation

If C is a code of length r on a (finite) alphabet Q, we write the codewords as $x = (x^{(1)}, \cdots, x^{(r)})$, where $x^{(i)} \in Q$ for $1 \leq i \leq r$. Subsets of C will be called *coalitions*. For any coalition $C_0 \subseteq C$, we define the set of *descendants* of C_0, denoted $\mathsf{desc}(C_0)$ by

$$\mathsf{desc}(C_0) = \{w \in Q^r : w^{(i)} \in \{x^{(i)} : x \in C_0\}, 1 \leq i \leq r\}.$$

The set $\mathsf{desc}(C_0)$ consists of the r-tuples that could be produced by the coalition C_0. We define $\mathsf{desc}_c(C)$ to be the set of all $w \in Q^r$ for which there exists a coalition C_0 of size at most c such that $w \in \mathsf{desc}(C_0)$. In other words, $\mathsf{desc}_c(C)$

A. Silverberg is with the Department of Mathematics, Ohio State University, Columbus, OH, USA; email: silver@math.ohio-state.edu. Silverberg would like to thank MSRI, Bell Labs Research Silicon Valley, Xerox PARC, NSA, and NSF.

J. Staddon is with the Palo Alto Research Center Inc., Palo Alto, CA, USA. Much of this work was completed while Staddon was employed by Bell Labs Research Silicon Valley.

J. Walker is with the Department of Mathematics and Statistics, University of Nebraska, Lincoln, NE, USA. Walker is partially supported by NSF grants DMS-0071008 and DMS-0071011.

A version of this paper appeared in ASIACRYPT 2001.

consists of the r-tuples that could be produced by a coalition of size at most c. For $x, y \in Q^r$, let $I(x,y) = \{i : x^{(i)} = y^{(i)}\}$. The *(Hamming) distance* between x and y is $r - |I(x,y)|$.

Definition 1: A code C is a c-TA (traceability) code if for all coalitions C_i of size at most c, if $w \in \mathsf{desc}(C_i)$ then there exists $x \in C_i$ such that $|I(x,w)| > |I(z,w)|$ for all $z \in C - C_i$.

In other words, C is a c-TA code if, whenever a coalition of size at most c produces a pirate word w, there is an element of the coalition that is closer to w than any codeword not in the coalition.

Codes with the identifiable parent property (IPP) are another type of traceability code.

Definition 2: A code C is a c-IPP code if for all $w \in \mathsf{desc}_c(C)$, the intersection of the coalitions C_i of size at most c such that $w \in \mathsf{desc}(C_i)$ is nonempty.

If C is a c-IPP code and $w \in \mathsf{desc}_c(C)$, then the *traitors* that can produce the *pirate* w are the codewords that lie in all coalitions C_i of size at most c such that $w \in \mathsf{desc}(C_i)$.

When implementing one of these traceability codes, the users are given encrypted versions of their codewords, and not the codewords themselves. Encrypting the codewords thwarts algebraic attacks.

Suppose X is a smooth, absolutely irreducible curve of genus g defined over a finite field $\mathbf{F}_q$, $\mathcal{P} = \{P_1, \ldots, P_r\}$ is a set of r distinct $\mathbf{F}_q$-rational points on X, P_0 is an $\mathbf{F}_q$-rational point on X that is not in the set $\mathcal{P}$, and ℓ is an integer. Then the one-point algebraic geometry (AG) code $C_X(\mathcal{P}, \ell P_0)$ consists of the r-tuples $(f(P_1), \ldots, f(P_r))$, with f running over the rational functions on X whose only pole is P_0, where the multiplicity is at most ℓ. If $2g - 2 < \ell < r$, this code has dimension $\ell + 1 - g$ and minimum distance at least $r - \ell$. Reed-Solomon codes can be viewed as algebraic geometry codes by taking X to be the projective line, $\mathcal{P}$ to be the set of points corresponding to the r chosen field elements, P_0 to be the point at infinity, and $\ell = k - 1$, where k is the code's dimension.

B. Background Traceability Results

Lemma 3: ([18], Lemma 1.3) Every c-TA code is a c-IPP code.

As shown in [18], there are c-IPP codes that are not c-TA. We give a simple example of a 2-IPP code that is not 2-TA.

Example 4: Let $u_1 = (0,0,1)$, $u_2 = (1,0,0)$, and $u_3 = (2,0,0)$. The code $\{u_1, u_2, u_3\}$ is clearly 2-IPP, since the first entry of a pirate determines a traitor. The coalition $\{u_1, u_2\}$ can produce the pirate $w = (0,0,0)$. However, $|I(u_1,w)| = |I(u_2,w)| = |I(u_3,w)| = 2$, so the code is not 2-TA.

Note that for c-IPP codes, traitor tracing is roughly an $O(\binom{N}{c})$ process, where N is the total number of codewords in the code. A traitor tracing algorithm for a c-TA code takes as input a $w \in \mathsf{desc}_c(C)$ and outputs a codeword x such that $|I(x,w)|$ is largest. Hence for c-TA codes, tracing is an $O(N)$ process, in general.

The next result, which is proved in [18] (see also [6] and [7]), shows that for codes with large enough minimum distance the TA algorithm suffices, and consists of finding codewords within distance $r - \frac{r}{c}$ from the pirate. Further, all codewords within this distance will be traitors.

Theorem 5: ([18], Theorem 4.4) Suppose C is a code of length r, c is a positive integer, and the minimum distance d of C satisfies $d > r - \frac{r}{c^2}$. Then

(i) C is a c-TA code;
(ii) if C_0 is a coalition of size at most c, and $w \in \mathsf{desc}(C_0)$, then:
  a) there exists an element of C_0 within distance $r - \frac{r}{c}$ of w, and
  b) every codeword within distance $r - \frac{r}{c}$ of w is in the coalition C_0.

Proof: Let C_0 and w be as in the statement of the theorem. By the pigeonhole principle, there exists $x \in C_0$ such that $|I(x,w)| \geq \frac{r}{c}$, and so the distance from x to w is at most $r - \frac{r}{c}$. Now suppose $z \in C \setminus C_0$. Then since $w \in \mathsf{desc}(C_0)$ and any two distinct codewords of C share fewer than $\frac{r}{c^2}$ coordinates, we have

$$|I(z,w)| \leq \sum_{y \in C_0} |I(z,y)| < c\frac{r}{c^2} = \frac{r}{c}.$$

Thus z has distance greater than $r - \frac{r}{c}$ from w, and by definition, C is a c-TA code. ■

III. Efficient Tracing Algorithms via List Decoding

In this section we show how the efficiency of the TA tracing algorithm can be greatly improved when the traceability scheme is based on certain error-correcting codes, and the tracing algorithm uses fast list decoding methods. What is an $O(N)$ process in general becomes a process that runs in time polynomial in $c \log N$, where c is the maximum coalition size and N is the number of users. These constructions match the best previously known traceability schemes in this model in terms of the alphabet size that is required to support a given level of traceability and codeword length (roughly speaking, the alphabet size is $O(N^{\frac{c^2}{r}})$), and exceed all earlier schemes in the speed with which they trace (at least) one traitor. The following theorem describes constructions based on Reed-Solomon, algebraic geometry, and concatenated codes. One advantage of considering all three types of codes is that the appropriate code choice for the traceability scheme depends on the desired parameters.

Theorem 6:

(i) Let C be a Reed-Solomon code of length r and dimension k over a finite field $\mathbf{F}_q$ of size at most 2^r. If c is an integer, $c \geq 2$, and $r > c^2(k-1)$, then C is a c-TA code and there is a traitor tracing algorithm that runs in time $O(r^{15})$. If $r = (1+\delta)c^2(k-1)$ then the algorithm runs in time $O(\frac{r^3}{\delta^6})$. For $r = \Theta(c^2 k)$, the runtime is $O(c^{30} \log_q^{15} N)$.
(ii) Let X be a nonsingular plane curve of genus g defined over a finite field $\mathbf{F}_q$, $\mathcal{P}$ a set of r distinct $\mathbf{F}_q$-rational points on X, P_0 an $\mathbf{F}_q$-rational point on X that is not in $\mathcal{P}$, and k an integer such that $k > g - 1$. Let c be an integer such that $c \geq 2$ and $r > c^2(k+g-1)$, assume that $q \leq 2^r$, and assume the pre-processing described in [11] has occurred. Then the one-point AG code $C_X(\mathcal{P}, (k+$

$g-1)P_0)$ is a c-TA code with a traitor tracing algorithm that runs in time polynomial in r.

(iii) If k and c are positive integers, q is a prime power, $q > c^2 \geq 4$, and δ is a real number such that $0 < \delta \leq \frac{q/c^2-1}{q-1}$, then there exists an explicit linear c-TA code over the field $\mathbf{F}_q$ of length $r = O(\frac{k^2}{\delta^3 \log(1/\delta)})$ (or length $r = O(\frac{k}{\delta^2 \log^2(1/\delta)})$) and dimension k with a polynomial (in r) traitor tracing algorithm.

Proof:

(i) Since C is a Reed-Solomon code, the minimum distance d satisfies $d = r - k + 1$. The condition $r > c^2(k-1)$ is then equivalent to the condition $d > r - r/c^2$. By Theorem 5, C is a c-TA code and traitor tracing amounts to finding a codeword within distance $r - r/c$ of the pirate. Theorem 12 and Corollary 13 of [11] imply that if $t > \sqrt{(k-1)r}$ then all codewords within distance $r - t$ of a given word can be listed in time $O(r^{15})$, and if $t^2 = (1+\delta)(k-1)r$ then the runtime is $O(\frac{r^3}{\delta^6})$. Taking $t = r/c$ gives the desired results. (Note that $k = \log_q N$.)

(ii) The minimum distance d of the code satisfies $d \geq r - k - g + 1$ (see, for example, Theorem 10.6.3 of [15]). By our choice of c we have $d \geq r - k - g + 1 > r - r/c^2$ and $r - r/c < r - \sqrt{r(k+g-1)}$. By Theorem 27 of [11], there exists an algorithm that runs in time polynomial in r that outputs the list of codewords of distance less than $r - \sqrt{r(k+g-1)}$ from a given word. Now apply Theorem 5.

(iii) Theorems 7 and 8 and Corollaries 2 and 3 of [12] imply that there exists an explicit concatenated code over $\mathbf{F}_q$ of the correct length r and dimension k, with minimum distance $d \geq (1 - \frac{1}{q})(1-\delta)r$, with a polynomial time list decoding algorithm for e errors, as long as $e < (1 - \sqrt{\delta})(q-1)r/q$. The condition $\delta \leq \frac{q/c^2-1}{q-1}$ implies that $d > r - r/c^2$ and that the upper bound on the number of errors is satisfied when $e \leq r - r/c$. The result now follows from Theorem 5.

■

We emphasize that further improvements in the runtime of list decoding algorithms are being rapidly produced. It seems that some of these results will bring the runtime down to $O(r \log^3 r)$ for Reed-Solomon codes, at least in certain cases (see [9]). The list decoding algorithm in [11] for AG codes is improved in [20] (see Theorem 4.1), where an explicit runtime is also given.

In related work, a public-key traitor tracing scheme is given in [3]. One of the nice properties of the scheme in [3] is that it is possible to identify *all* traitors. We note that although our algorithms in this section can only guarantee the identification of one traitor, they do so in significantly faster time (polynomial in $c \log N$, versus polynomial in N, with N the number of codewords and c the maximum coalition size). In addition, we note that probabilistic tracing has been shown to be possible on potentially shorter codewords in [5]. The tracing algorithm in [5] appears to have a longer expected running time than those of Theorem 6.

IV. Comparative Analysis of TA and IPP Traceability

The results in this section justify a focus on TA (as opposed to more general IPP) schemes. In this paper we have been using linear codes to construct schemes for which the TA tracing algorithm is efficient. We know by Lemma 3 that c-TA codes are also c-IPP codes. However the converse fails ([18]; see also Example 4 above). If constructions of schemes for which the IPP tracing algorithm is efficient (i.e., significantly reduced from $O(\binom{N}{c})$ time) are possible, it is reasonable to expect this to be accomplished by introducing an algebraic or linear structure. Here we give evidence that doing so may enable the inherently more efficient TA algorithm to be used to identify traitors.

First, we prove a necessary condition on Reed-Solomon codes, under which they yield c-TA set systems. This condition is that the minimum distance is greater than $r - r/c^2$, where r is the length of the codewords. This result suggests a potential method for generating examples of schemes that are c-IPP but not c-TA, namely, decreasing the minimum distance. Next we demonstrate through a family of counterexamples that in fact this approach does not work in general; when the minimum distance is $r - r/c^2$ it is possible to find Reed-Solomon codes for which both the IPP and TA tracing algorithms fail.

There is a natural way to produce unordered sets from the ordered sets that constitute the code: to a codeword

$$x = (x^{(1)}, \ldots, x^{(r)}),$$

associate the set

$$x' = \{(1, x^{(1)}), \ldots, (r, x^{(r)})\}.$$

We define TA and IPP set systems (as opposed to TA and IPP codes) in the natural way, with the noteworthy difference that a pirate *unordered set* consists of r elements such that each element is a member of some coalition member's set. This is a generalization of our earlier definition because it is not necessary to have one element of the form $(i, y^{(i)})$ for each $i = 1, \ldots, r$. The following theorem is a partial converse of Theorem 5.

Theorem 7: If $c \geq 2$ is an integer and C is a Reed-Solomon code of length r with minimum distance $d \leq r - \frac{r}{c^2}$, then the set system corresponding to C is not a c-TA set system.

Proof: As above, if $x \in C$, write

$$x' = \{(1, x^{(1)}), \ldots, (r, x^{(r)})\}$$

for the associated element of the set system. Choose a codeword $v = (v^{(1)}, \ldots, v^{(r)})$ in C. We will show that a coalition of size at most c exists that does not contain v', but that can implicate v'. In other words, we will construct a pirate set w which can be created by a coalition $\{u'_1, \ldots, u'_b\}$ with $b \leq c$ that does not contain v', but which satisfies $|v' \cap w| \geq |u'_i \cap w|$ for every i. Let $\delta = r - d = k - 1$, where k is the dimension of the code C. By assumption, $\delta \geq r/c^2$.

First, assume $c\delta \leq r$. For $i = 1, \ldots, c$, choose $u_i \in C$, distinct from v, but which agrees with v on the δ positions $(i-1)\delta + 1, \ldots, i\delta$. (To do this, simply find a polynomial h_i of degree δ that vanishes on the δ field elements corresponding to these δ positions, and let u_i be the codeword corresponding to the polynomial $f - h_i$, where f is the polynomial corresponding to v.) Notice that, since two distinct codewords can agree on at most δ positions, each u'_i contains at least $r - c\delta$ elements that are not in v' or in u'_j for any $j \neq i$. Since $r - c\delta \geq 0$ and $c \geq 2$, we have $r - c\delta \geq \lceil \frac{r-c\delta}{c} \rceil = \lceil \frac{r}{c} \rceil - \delta$. We now form a pirate set w from the coalition $\{u'_1, \ldots, u'_c\}$ as follows: First, choose the elements $((i-1)\delta + l, u_i{}^{((i-1)\delta+l)})$ from u'_i for $1 \leq i \leq c$ and $1 \leq l \leq \delta$. Next, choose $\lceil \frac{r}{c} \rceil - \delta$ elements from each u'_i, that are not in any u'_j for $j \neq i$ and that are not in v'. This gives a total of $c(\delta + (\lceil \frac{r}{c} \rceil - \delta)) \geq r$ elements of w. If the inequality is strict, simply throw away as many of the $c(\lceil \frac{r}{c} \rceil - \delta)$ elements which were chosen in the second step as necessary. We see that for every i, $|u'_i \cap w| \leq \delta + (\lceil \frac{r}{c} \rceil - \delta) = \lceil \frac{r}{c} \rceil$ and $|v' \cap w| = c\delta \geq \lceil \frac{r}{c} \rceil$. Thus the TA algorithm will mark v' as a traitor.

If on the other hand $c\delta > r$, simply choose $u_1, \ldots, u_j$ as above, where $j = \lfloor \frac{r}{\delta} \rfloor < c$, and choose $u_{j+1} \neq v$ to agree with v on the last $r - j\delta$ positions. The coalition $\{u'_1, \ldots, u'_{j+1}\}$ can create v' as a pirate set. ■

Theorem 7 leaves open the question of whether Reed-Solomon codes with minimum distance at most $r - \frac{r}{c^2}$ might still have traceability when the IPP algorithm is used even though the TA algorithm may no longer correctly identify traitors. The next theorem gives a family of counterexamples illustrating that this is not generally the case by giving examples of Reed-Solomon codes of length r and minimum distance $r - r/c^2$ which are not c-IPP.

Theorem 8: Let s and c be positive integers with $c \geq 2$, and let p be a prime number greater than c^2. For $i = 1, \ldots, c$, let $a_i = (i-1)c$. For $i = 1, \ldots, c$, if s is not divisible by p, let $g_i(x) = x^s - i$; otherwise let $g_i(x) = x^s + x - i$. Let T be the set of roots of all the c^2 polynomials $g_i - a_j$. Let q be a sufficiently high power of p so that T is a subset of the finite field $\mathbf{F}_q$. Then T consists of $c^2 s$ distinct elements of $\mathbf{F}_q$. Let C be the Reed-Solomon code in which the codewords are the evaluations at the elements of T of all polynomials over $\mathbf{F}_q$ of degree at most s. Then the dimension of the code C is $s+1$, the length r of the codewords is $r = c^2 s$, the minimum distance of C is $r - r/c^2$, and C is not c-IPP.

Proof: We first show that T consists of $c^2 s$ distinct elements. Let $h_{ij} = g_i - a_j$. Then $h_{ij}(x) - h_{mn}(x) = -i-(j-1)c+m+(n-1)c$. If $h_{ij}(x) - h_{mn}(x) = 0$, then $m - i$ is divisible by c. Since m and i are both in the range $1, \ldots, c$, they must be equal. Thus $(j-1)c = (n-1)c$, and so $j = n$. Therefore the set $\{h_{ij}\}$ consists of c^2 distinct polynomials of degree s, any two of which differ by a non-zero constant. Therefore no two can have a root in common. Further, the derivative of h_{ij} is sx^{s-1} if s is not divisible by p, and is 1 otherwise. In both cases this derivative is relatively prime to h_{ij} (in the first case, note that h_{ij} is always of the form x^s+(a non-zero constant), so it never has 0 as a root). Therefore all the roots of h_{ij} are simple. So T consists of $c^2 s$ distinct elements, and it makes sense to consider the Reed-Solomon code defined by evaluating polynomials of degree at most s at the elements of T. The code clearly has the stated parameters. The two coalitions corresponding to the polynomials in the sets $\{a_1, \ldots, a_c\}$ and $\{g_1, \ldots, g_c\}$ are disjoint, and each coalition can produce the pirate word defined as follows: for each β in T, the β-th entry of the pirate word is $g_i(\beta) = a_j$, for the unique i and j such that the equality holds. It follows that the code is not c-IPP. ■

By evaluating the polynomials at subsets of T of size at least $s+1$ (to ensure that $k \leq r$), we can take the length r to be anything between $s+1$ and $c^2 s$. The resulting minimum distance $r - s$ is then at most $r - r/c^2$.

We remark that if s is not divisible by p, then we can always find a q that works and is a divisor of p^s.

Example 9: The Reed-Solomon code obtained by evaluating constant and linear polynomials over $\mathbf{F}_{11}$ at all the points (0 to 10) of $\mathbf{F}_{11}$ is not 4-IPP, since the disjoint coalitions corresponding to the polynomials $\{0, 1, 2, 3\}$ and $\{x, x-4, x-8\}$ can create the pirate $(0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2)$. For this code we have $d < r - r/c^2$, since $d = r - k + 1 = 11 - 2 + 1 = 10$ and $r - r/c^2 = \frac{165}{16}$.

Example 10: The Reed-Solomon code obtained by evaluating polynomials over $\mathbf{F}_{11}$ of degree at most 2 at 0, 1, 2, 3, 4, 5, 6, 7 is not 2-IPP, since the 3 disjoint coalitions:

$$\{(7,9,0,2,4,6,8,10), (0,6,2,10,8,7,7,8)\},$$
$$\{(6,7,2,2,7,6,10,8), (0,9,9,0,4,10,7,6)\},$$
$$\{(5,8,7,2,4,2,7,8), (0,9,2,1,6,6,1,2)\}$$

can create the pirate $(0, 9, 2, 2, 4, 6, 7, 8)$. For this code we have

$$d = r - k + 1 = 8 - 3 + 1 = 6 = 8 - 8/4 = r - r/c^2.$$

The results in this section lead to the following questions, which are of independent interest.

Question 11: Is it the case that $d > r - r/c^2$ for all c-IPP Reed-Solomon codes of length r and minimum distance d?

It is easy to see that this would be false if "Reed-Solomon" were replaced by "linear". For example, one-dimensional linear codes are always both c-IPP and c-TA, but can have $d \leq r - r/c^2$ if they are not Reed-Solomon codes (for one-dimensional codes, the minimum distance d is the number of non-zero entries in the non-zero codewords; the codewords of distance less than d from the pirate lie in every coalition that can create the pirate).

If the answer to Question 11 were yes, combining it with Theorem 5 would imply that all Reed-Solomon c-IPP codes are c-TA. We raise as an open question:

Question 12: Is it the case that all *linear* c-IPP codes are c-TA?

V. Finding All Possible Coalitions

A coding theoretic approach can be used to amass additional piracy information: a list of all coalitions that are capable of creating a given pirate. Such information is useful in two respects. It clears all codewords not appearing in any of these coalitions of involvement in constructing the pirate word, and it constitutes useful audit information (circumstantial evidence) that may be helpful in the prosecution of a traitor later on. The algorithms of this section require only that the code have minimum distance greater than $r - \frac{r}{c^2}$, and therefore are applicable to the codes in Theorem 6. The algorithms are fast when fast list decoding techniques exist. In addition, for every code meeting this minimum distance requirement and having fast list decoding, the algorithms enable the IPP traitor tracing algorithm [13], [2], [18] to run more efficiently (as that algorithm works by intersecting all coalitions that are capable of creating a given pirate word).

At a high level, the main algorithm builds a "tree" from which all c-coalitions capable of constructing a pirate w can be extracted. At the root of the tree lie all codewords that we know must be in *every* such coalition. The children are then candidate codewords for the next member of the coalition. Branches of the tree are extended until the current coalition "covers" w (i.e., is capable of constructing w), or until it becomes clear that this is impossible (e.g., because the coalition is already of size c and still cannot create w). In the latter case that "dead-end" coalition is discarded and other branches of the tree are explored.

Before describing the algorithm in detail, we need two definitions. First, given a subset S of $\{1,\ldots,r\}$ of size s, define the map $f_S : \mathbf{F}_q^r \to \mathbf{F}_q^{r-s}$ to be the projection that omits the entries in positions corresponding to elements of S. Second, a *minimal* c-coalition for w is a subset U of C such that $|U| \le c$, $w \in \mathsf{desc}(U)$, but w is not in $\mathsf{desc}(V)$ for any proper subset V of U. Since one may obtain all coalitions of size at most c that can create w from the minimal ones by appending arbitrary elements of the code, it is enough to find all minimal c-coalitions.

Algorithm Sketch:
Input: Integer $c > 1$, code C of length r and minimum distance greater than $r - \frac{r}{c^2}$, pirate word $w \in \mathsf{desc}_c(C)$.
Output: A list of coalitions of size at most c that can create w, including all minimal c-coalitions for w.
Steps:

(i) Use list decoding to find all codewords $u_1,\ldots,u_a \in C$ ($a \le c$) within distance $r - r/c$ of w. Let S be the subset of $\{1,\ldots,r\}$ on which w agrees with at least one of $\{u_1,\ldots,u_a\}$, and set $s = |S|$. Set $r_1 = r - s$, $c_1 = c - a$, $C_1 = f_S(C)$, and $w_1 = f_S(w)$. If $r_1 = 0$, quit and output $\{u_1,\ldots,u_a\}$. Set $i = 1$.

(ii) Use list decoding to find all codewords $v_{i1},\ldots,v_{ib_i} \in C_i$ within distance $r_i - r_i/c_i$ of w_i. If this outputs the empty-set, exit to Step (iii). Otherwise, let S_i be the subset of $\{1,\ldots,r_i\}$ on which w_i agrees with v_{ib_i}, and set $s_i = |S_i|$. Set $r_{i+1} = r_i - s_i$, $c_{i+1} = c_i - 1$, $C_{i+1} = f_{S_i}(C_i)$, and $w_{i+1} = f_{S_i}(w_i)$, and repeat this step.

(iii) Beginning with u_1, …, u_a, form coalitions to output by adding (lifts to C of) v_{1b_1}, v_{2b_2}, and so on, until w is a descendant of the list. When this process succeeds or dead-ends (i.e., w is not a descendant of the current list but there are no codewords within distance $r_i - r_i/c_i$ of w_i or the list already has c codewords), move back through the v_{ij}'s to find the first unexplored branch and repeat Step (ii) with a different v_{ij} in place of v_{ib_i}. The algorithm terminates when all branches have been explored.

Analysis of the Algorithm:
The output of the algorithm is clearly a list of coalitions of size at most c that can create the pirate, and includes each minimal c-coalition at least once. Note that in Step (iii), all lifts of each v_{ij} should be considered. By Theorem 5, $u_1,\ldots,u_a$ are in every coalition that can create w. In Step (ii), if $d_i > r_i - r_i/c_i^2$ where d_i is the minimum distance of C_i, then every coalition that can produce the original pirate w will contain some lift to the original code of some v_{ij}. Moreover, if a lift to C of v_{ij} is in some coalition that can create the original pirate w, then there exists a codeword within $r_i - r_i/c_i$ of v_{ij} (by the pigeonhole principle), and the algorithm will proceed. If Step (ii) returns the empty-set, then the current path is a dead-end. When C satisfies any of the sets of conditions in Theorem 6, then Step (i) can be done efficiently (time polynomial in r). Note that if C is a Reed-Solomon (resp., algebraic geometry) code, then so is C_i.

The brute force method for finding all coalitions runs in time $O(crN^c)$, where N is the total number of codewords in the code and $N \gg c$ (for each of the at most N^c coalitions of size at most c, compare each of the r entries of the pirate to the corresponding entry of each member of the coalition). For Reed-Solomon codes with $r = \Theta(c^2k)$, this gives a runtime of $O(c^3N^c \log N)$.

Variation:

A variation of the above algorithm is to list decode to find all codewords $u_1,\ldots,u_a$ ($1 \le a \le c$) within distance $r - r/c$ of the pirate (as in Step (i) above), and then use brute force to determine the remaining (at most) $c - a$ members of the coalitions. When C is a Reed-Solomon code satisfying the conditions in Theorem 6(i) with $r = \Theta(c^2k)$, the dominant term in the runtime is $O(c^3N^{c-a} \log N)$. This is clearly an improvement over brute force alone, since $a \ge 1$.

VI. Future Directions: Tracing with Extra Information

In this section, we describe how other coding theoretic techniques may be applied to the traitor tracing problem when additional information about traitor behavior is available.

One possible approach to tracing traitors is to try to second-guess the traitors' strategy. For example, if you believe that one traitor has contributed more than the other members of the coalition to the pirate, you can apply bounded-distance decoding up to the error-correction bound to find such traitors very quickly. This might involve a "ringleader" or "scapegoat" scenario. If on the other hand you believe that all traitors contributed roughly equal amounts, then list decoding should be tried first. Traitors can be searched for in sequences of expanding Hamming balls around the pirate. These searches can be run in parallel or sequentially. The runtime of bounded-distance decoding up to the error-correction bound for Reed-Solomon codes is at most quadratic in the length of the codewords. Note that [16] gives a fast algorithm for list decoding Reed-Solomon codes beyond the error-correction bound (also quadratic in the codeword length), but does not go as far as the Guruswami-Sudan algorithm. It therefore will not be guaranteed to find a traitor, but would quickly find a ringleader.

In [11], list decoding is considered not just in the case of errors, but also in the case of erasures and errors (and another potentially useful case that is referred to as "decoding with uncertain receptions"). For concatenated codes, [12] also deals with the problem of decoding from errors and erasures. Building on [11], [14] presents a high-performance soft-decision list decoding algorithm. We believe that these results also have potential for use in traitor tracing problems, in cases where some additional information is known about the traitors or how they are operating.

If one has information about the traitors or their modes of operation, one can build that information into a reliability matrix, and apply soft-decision decoding algorithms to trace. For example, suppose we know that a traitor who contributed the first entry to the pirate contributed at least r/c entries to the pirate. One can use this information to construct a skewed reliability matrix. If the underlying code is a Reed-Solomon code over a finite field of size q, one can then apply the soft-decision algorithm in [14] to find such a "dominant" traitor. The channel that models this situation is a q-ary symmetric channel. The first column of the reliability matrix will have a 1 in the entry corresponding to the field element that occurs in the first position of the pirate, and 0's elsewhere. For $j > 1$, the jth column of the reliability matrix will have $1 - \epsilon$ in the entry corresponding to the field element in the jth entry of the pirate, and the other entries will all be $\frac{\epsilon}{q-1}$, where $\epsilon < \frac{q-1}{q}$ is chosen so as to optimize the soft-decision decoding algorithm in [14]. If one does not know which entry was contributed by the traitor who contributed the most, one possible search method is to choose entries at random from the pirate and apply the above strategy to search for traitors that contributed that entry.

Erasure-and-error decoding may be useful in fingerprinting or watermarking scenarios, such as those presented in [4], [5], [10], [1]. In one model, a coalition creates a pirate copy of the digital content by leaving fixed all codeword entries where they all agree, and choosing the values of the remaining positions from $Q \cup \{?\}$, where Q is the alphabet. The ?'s can be viewed as erasures.

VII. Conclusion

We have demonstrated that traitor tracing algorithms can be quite efficient when the construction of the traceability scheme is based on error-correcting codes and the method of tracing is based on fast list decoding algorithms. For the TA algorithm, traitors can be identified in time polynomial in r, where r is roughly $c^2 \log_q N$, rather than in time $O(N)$. We also give evidence for a close relationship between the TA and IPP properties, for linear codes, and raise some open questions about this relationship. Finally, we suggest avenues for future research, including explorations of applications of soft-decision and erasure decoding techniques to traitor tracing in scenarios where additional information has been obtained about the traitors or their mode of operation.

Acknowledgments.: The authors thank Dan Boneh, Gui-Leng Feng, Tom Høholdt, Ralf Kötter, and Madhu Sudan for useful conversations.

References

[1] A. Barg, G. R. Blakley and G. Kabatiansky. Digital fingerprinting codes: problems statements, constructions, identification of traitors. Presented in part at the IEEE International Sumposium on Information Theory, June 2001, Washington, DC and at the Workshop on Coding and Cryptography, National University of Singapore, September 2001.

[2] A. Barg, G. Cohen, S. Encheva, G. Kabatiansky and G. Zémor. A hypergraph approach to the identifying parent property: the case of multiple parents. In Coding and Cryptography (WCC 2001), Discrete Applied Mathematics **111** (2001), Elsevier.

[3] D. Boneh and M. Franklin. An efficient public key traitor tracing scheme. In "Advances in Cryptology – Crypto '99", *Lecture Notes in Computer Science* **1666** (1999), 338–353.

[4] D. Boneh and J. Shaw. Collusion secure fingerprinting for digital data. In "Advances in Cryptology – Crypto '95", *Lecture Notes in Computer Science* **963** (1995), 452–465.

[5] D. Boneh and J. Shaw. Collusion secure fingerprinting for digital data. *IEEE Transactions on Information Theory* **44** (1998), 1897–1905.

[6] B. Chor, A. Fiat and M. Naor. Tracing traitors. In "Advances in Cryptology – Crypto '94", *Lecture Notes in Computer Science* **839** (1994), 480–491.

[7] B. Chor, A. Fiat, M. Naor and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory* **46** (2000), 893–910.

[8] I. Cox, J. Kilian, T. Leighton and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Information Theory* **6** (1997), 1673–1687.

[9] G.-L. Feng. Very Fast Algorithms in Sudan Decoding Procedure for Reed-Solomon Codes. Preprint.

[10] A. Fiat and T. Tassa. Dynamic traitor tracing. In "Advances in Cryptology – Crypto '99", *Lecture Notes in Computer Science* **1666** (1999), 354–371.

[11] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory* **45**(6) (1999), 1757–1767.

[12] V. Guruswami and M. Sudan. List decoding algorithms for certain concatenated codes. In Proc. 32nd ACM Symposium on Theory of Computing (STOC 2000), 181–190.

[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 A* **82** (1998), 121–133.

[14] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. Submitted to IEEE Transactions on Information Theory, 2001,

[15] J. H. van Lint. Introduction to coding theory. Third edition. Graduate Texts in Mathematics **86**, Springer-Verlag, Berlin (1999).

[16] R. M. Roth and G. Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory* **46** (2000), 246–257.

[17] B.-Z. Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory* **39** (1993), 239–242.

[18] J. N. Staddon, D. R. Stinson and R. Wei. *Combinatorial properties of frameproof and traceability codes. IEEE Transactions on Information Theory* **47** (2001), 1042–1049.

[19] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity* **13**(1) (1997), 180–193.

[20] X.-W. Wu and P. H. Siegel. Efficient Root-Finding Algorithm with Application to List-Decoding of Algebraic-Geometric Codes. Submitted to *IEEE Transactions on Information Theory*.

[21] F. Zane. Efficient Watermark Detection and Collusion Security. In Proceedings of Financial Cryptography 2000, Anguilla, February 2000.