EFFICIENT TRAITOR TRACING ALGORITHMS USING LIST DECODING

ALICE SILVERBERG, JESSICA STADDON, AND JUDY WALKER

Abstract. We apply powerful, recently discovered techniques for the list decoding of error-correcting codes to the problem of efficiently tracing traitors. Traitor tracing schemes have been extensively studied for use as a piracy deterrent. In a widely studied model for protecting digital content, each user in the system is associated with a unique set of symbols. For example, the sets may be used to install a software CD or decrypt pay-TV content. The assignment of sets is done in such a way that if a bounded collection of sets is used to form a new set to enable piracy, at least one of the traitor sets can be identified by applying a traitor tracing algorithm to the newly formed set. Much work has focused on methods for constructing such traceability schemes, but the complexity of the traitor tracing algorithms has received little attention. A widely used traitor tracing algorithm, the TA algorithm, has a running time of $O(N)$ in general, where N is number of sets in the system (e.g., the number of copies of the CD), and therefore is inefficient for large populations. In this paper we use a coding theoretic approach to produce traceability schemes for which the TA algorithm is very fast. We show that when suitable error-correcting codes are used to construct traceability schemes, and fast list decoding algorithms are used to trace, the run time of the TA algorithm is polynomial in the codeword length. We also use the strength of the error-correcting code approach to construct traceability schemes with more efficient algorithms for finding all possible traitor coalitions. Finally, we provide evidence that amongst traceability schemes in general, TA traceability schemes are the most likely to be amenable to efficient tracing methods.

1. Introduction

Traceability schemes are introduced in [7] 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 list decoding techniques are receiving wide attention in the coding theory community, and improvements and generalizations are being rapidly produced. This paper gives the first applications of these important tools to the problem of tracing traitors.

A popular model for traceability schemes is one in which a unique set (possibly ordered) of r symbols is associated with each user. For example, the set may be embedded in a software CD possessed by the user, or contained in a smartcard the user has for the purpose of viewing encrypted pay-TV programs (in the latter case, the set corresponds to a set of keys). When a coalition forms to commit piracy, it must construct a set to associate with the pirate object. In the case of unordered sets, this pirate set consists of r symbols, each of which belongs to at least one coalition member's set. If the sets are ordered, the coalition members have less freedom and must form an ordered pirate set in which the symbol in each position is identical to the symbol in the same position in the ordered set of some coalition member. In either scenario a traitor tracing algorithm is applied to the pirate, and the sets are constructed in such a way that the algorithm only identifies an actual traitor or traitors. In practice, one randomly chooses a set of symbols $\{s_{(i,y)}\}$ with $i \in \{1, ..., r\}$ and y in a finite alphabet Q, and the collection of symbols corresponding to a given user is determined by the set associated with that user. For example, if the ordered set $x = (x_1, ..., x_r)$ is associated with user u, then the set of symbols associated with user u is $S_u = \{s_{(1,x_1)}, ..., s_{(r,x_r)}\}$. It is S_u, not x, that the user stores (e.g., S_u is embedded in the user's CD or stored on their smartcard). This additional step makes the model of pirate behavior that we consider reasonable. Since the symbols are generated randomly

Much of this work was completed while Staddon was employed by, and Silverberg was visiting, Bell Labs Research Silicon Valley. Walker is partially supported by NSF grants DMS-0071008 and DMS-0071011. Silverberg would like to thank MSRI, Bell Labs Research Silicon Valley, NSA, and NSF.

it is essentially impossible to guess a symbol, and hence a coalition is only able to form pirate words out of its pooled collection of symbols. In other words, moving from codewords to symbols thwarts algebraic attacks (see, for example, [15]). Although a coalition may be able to write down any codeword on which a user's set is based (this information may be public) it can only generate the symbol associated with an entry in the codeword if there is a coalition member that agrees with the codeword in that position. In this paper, we do not define an associated encryption mechanism, anticipating instead that if one is needed it will likely be a broadcast encryption scheme ([12]), as such a scheme enables certain users to be prevented from recovering the content from the encrypted broadcast.

The approach we take here is to use error-correcting codes to construct traceability schemes in which the sets are ordered. The ordered (as opposed to the unordered) set scenario yields naturally to coding theoretic techniques and has many practical applications ([8, 6]). In addition, we note that when combined with certain types of broadcast encryption schemes, a combination we expect in practice, our assumptions on the traitors' behavior are validated. More precisely, many broadcast encryption schemes have been studied (see, for example, [7]) in which the only pirate sets that are capable of decrypting content are those constructed by choosing one symbol each from amongst the symbols the traitors have in each particular position. Hence, with such broadcast encryption schemes, if traitors do not behave as modeled here they will not create a valid pirate set, and piracy attempts will be thwarted without any need for traceability.

We focus on the TA traitor tracing algorithm (following the terminology in [34]), that identifies as traitors all users whose set shares the most with the pirate set. In general the TA algorithm runs in $O(N)$ time, where N is the number of users. However this paper shows that for suitable constructions based on error-correcting codes, tracing can be accomplished in time polynomial in the length of each codeword, a significant improvement. The constructions in this paper match the best previously known schemes in this model in terms of the alphabet size that is required to achieve a certain level of traceability for a given codeword length, and exceed all earlier schemes in the speed with which they trace (at least) one traitor. Additional justification for focusing on the TA algorithm is derived in Section 5, where evidence is given that adding enough structure to a traceability scheme to enable fast tracing appears to make the properties of TA and IPP indistinguishable.

Our approach takes advantage of recent powerful methods for list decoding of linear codes, that originated with work of Sudan [39]. 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. Sudan's results by themselves are not strong enough to be applicable in the setting in which the TA algorithm succeeds in finding traitors (as opposed to identifying probable traitors), since the decoding procedure in [39] is not capable of correcting enough errors in the code. However, Sudan's work has recently been extended to enable it to efficiently correct more errors; in other words, it extends the radius of the Hamming ball around the received word in which it can find all the codewords in polynomial time. The improvements in [17] are precisely sufficient to be applicable to the setting where the TA algorithm succeeds. Efficient list decoding algorithms now exist for Reed-Solomon codes, more general algebraic geometry codes, and some concatenated codes. The results are rapidly undergoing improvement and generalization, and hold promise for greater improvements in the construction of efficient traceability schemes.

Traceability should be viewed as one weapon in an arsenal against piracy. Traceability is a worthwhile addition to a system provided the associated algorithms add sufficiently little cost, as we believe the techniques presented in this paper do. For example, as noted in [15], traceability can be a useful addition to a long-lived broadcast encryption scheme. If keys are allocated to smartcards in such a way as to ensure some traceability, it is possible to keep a list of traitor smartcards over time. If the smartcard of one particular user appears on the list frequently despite many smartcard refreshments (i.e., key changes) this mounting evidence makes it increasingly likely that the user is actually guilty, and not simply a victim of smartcard theft. Hence, as long as traceability schemes are efficient, they can quickly yield useful information during system audits.

Overview. The rest of the paper is organized as follows. Section 1.1 covers related work in the areas of traceability and broadcast encryption and Section 2 covers the necessary background on traceability and

gives a brief overview of the coding theoretic ideas used in this paper. Section 3 describes how to construct traceability schemes for which the TA algorithm is efficient. Section 4 discusses an efficient way to find all (minimal) coalitions of traitors. Section 5 considers the relationship between TA and IPP (a term defined in Section 2) traceability schemes. A discussion of other potential applications of coding theoretic ideas and techniques to traceability questions is given in Section 6.

1.1. **Related Work.** The phrase *traitor tracing* is coined in [7] (see also the extended version [8]). In traceability schemes, users are each given an ordered (as in [7, 6, 13, 34], for example), or unordered (as in [37], for example) set of keys. In many of these papers, an encryption scheme is specified in a way that enables the TA tracing algorithm to identify at least one traitor provided the coalition of traitors that colluded to produce the pirate is of bounded size and the pirate set has been constructed in accordance with the encryption scheme.

In [5] (see also the revised version [6]), methods for creating TA traceability codes are given for the purpose of fingerprinting digital data. Lower bounds and additional constructions of TA traceability schemes are given in [37], while lower bounds are also proven in [25, 24]. In addition, [24] provides a tracing algorithm for schemes in [25].

The problem of combining broadcast encryption and traceability is studied in [38, 14, 27, 43].

Some variations on the models of [8, 6] have been studied in recent years. *Dynamic* models (here we study a static model), in which it is possible to get additional evidence of piracy in order to "test" traitor guesses, are studied in [13, 2, 31]. A public-key traitor tracing scheme is given in [4]. One of the nice properties of the scheme in [4] is that it is possible to identify *all* traitors. We note, however, that although our algorithms in Section 3 can only guarantee the identification of one traitor, they do so in significantly faster time (polynomial in the code length r, versus $O(N\log N\log N\log\log N)$ in [4], where N is the number of codewords).

In [29, 9], ways in which accountability can be added to the model are discussed. For example, to improve upon the strength of the deterrent, in [9] committing piracy efficiently necessitates revealing sensitive information. In [15], a system in which pirate pay-TV decoders can only work for short periods of time is presented.

Recently, the identifiable parent property (IPP) tracing algorithm has garnered attention [21, 1, 34] (also, very similar ideas are studied in [36]). In [21], a combinatorial characterization of 2-IPP schemes is presented. Additional constructions of and bounds for IPP schemes appear in [1, 34].

A coding theoretic approach is taken in [23] to study the related problem of blacklisting users in a broadcast encryption scheme, but that paper does not address the question of tracing.

2. Background on Codes and Traceability

In this section we give definitions, notation, and background on codes, traceability, and decoding.

2.1. **Definitions and Notation.** A *code* C of *length* r is a subset of Q^r, where Q is a finite alphabet. The elements of C are called *codewords*; each codeword has the form $x = (x_1, \cdots, x_r)$, where $x_i \in Q$ for $1 \le i \le r$. 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

$$\mathsf{desc}(C_0) = \{w \in Q^r : w_i \in \{x_i : x \in C_0\}, \text{ for all } 1 \le i \le r\}.$$

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

For $x, y \in Q^r$, let $I(x, y) = \{i : x_i = y_i\}$.

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 $z \in C - C_i$.*

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.*

Suppose C is a code of length r. The *(Hamming) distance* between two elements x and y of Q^r is $r - |I(x,y)|$. The *minimum distance* of the code C is the smallest distance between distinct codewords of C.

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 $w \in \mathsf{desc}(C_i)$.

Since the traitor tracing problem is trivial when $c = 1$, we will always take c to be at least 2.

2.2. **Background on c-TA codes.** The following result, which is Lemma 1.3 of [34], is very useful for showing that a code is c-IPP.

Lemma 1. *Every c-TA code is a c-IPP code.*

As shown in [34], there are c-IPP codes which are not c-TA. We give a simple example of a 2-IPP code which is not 2-TA.

Example 1. *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 an $O(\binom{N}{c})$ process, in general, 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 the codewords x such that $|I(x,w)|$ is largest. Hence for c-TA codes, tracing is an $O(N)$ process, in general, where N is the number of codewords.

The next result, which is proved in [34] (Theorem 4.4 of that paper; see also [7] and [8]), 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. In fact, all codewords within this distance will be traitors.

Theorem 1. *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 a traitor within distance $r - \frac{r}{c}$ of w, and*
  (b) *every codeword within distance $r - \frac{r}{c}$ of w is a traitor.*

Proof: If $|C_0| \le c$ and $w \in \mathsf{desc}(C_0)$, then there exists $x \in C_0$ such that $|I(x,w)| \ge \frac{r}{c}$. Since $d > r - \frac{r}{c^2}$, if $x_1, x_2, \ldots, x_c, x_{c+1}$ are $c+1$ distinct codewords and $w \in \mathsf{desc}(\{x_1, \ldots, x_c\})$, we have

$$|I(w, x_{c+1})| \le \sum_{i=1}^{c} |I(x_i, x_{c+1})| < c\frac{r}{c^2} = \frac{r}{c}.$$

It follows that C is c-TA, and that the traitor tracing algorithm will only output codewords in C_0. In addition this demonstrates that to trace traitors in this construction, it suffices to find codewords within distance $r - r/c$ of the pirate w. □

2.3. **Linear Codes.** Linear codes are a very important class of codes. We will say that a code of length r is *linear*, or linear over F_q, if the alphabet is a finite field F_q and the code is a linear subspace of the vector space F_q^r. The *dimension* of the code is its dimension as a vector space. If C is a linear code over F_q of dimension k, then $|C| = q^k$.

Reed-Solomon codes are among the most widely-used linear codes, with many useful applications and properties. To obtain a Reed-Solomon code of length r and dimension k over the finite field F_q, fix r distinct elements $\alpha_1, \ldots, \alpha_r$ of F_q. The codewords are exactly the r-tuples $(f(\alpha_1), \ldots, f(\alpha_r))$ as f runs over (the zero polynomial and) all polynomials of degree $< k$ in $F_q[x]$. Note that a basis for the code over F_q can be taken to be

$$\{(1, \ldots, 1), (\alpha_1, \ldots, \alpha_r), (\alpha_1^2, \ldots, \alpha_r^2), \ldots, (\alpha_1^{k-1}, \ldots, \alpha_r^{k-1})\}.$$

Since two distinct polynomials of degree less than k agree on at most $k-1$ points, the minimum distance of the code is $r-k+1$.

A useful generalization of Reed-Solomon codes are *algebraic geometry (AG) codes* (see for example [16, 35, 41]). The linear codes with the "best" known parameters asymptotically are AG codes [42]. One advantage of AG codes is that they are not, in general, bound by the restriction that $r \leq q$, as was the case for the Reed-Solomon codes above. Being freed of this constraint allows us to have a smaller alphabet (and in applications, fewer keys), for given choices of the other parameters. Hermitian codes, coming from Hermitian curves, are examples of AG codes that have nice properties and can be defined explicitly. For those familiar with the below terminology (such knowledge is not essential for appreciating the results of this paper), we note that for our purposes it will suffice to consider the one-point codes $C_X(P, \ell P_0)$ which can be defined as follows. One begins with a smooth, absolutely irreducible curve X of genus g defined over a finite field F_q, a set $P = \{P_1, \ldots, P_r\}$ of F_q-rational points on X, another F_q-rational point P_0 on X which is not in the set P, and an integer ℓ. The codewords are then the r-tuples $(f(P_1), \ldots, f(P_r))$, where f is any element of $L(\ell P_0)$, the vector space of rational functions on X whose only poles occur at P_0, and with multiplicity at most ℓ. Under the assumption $2g-2 < \ell < r$, one finds that this code has dimension $\ell+1-g$ and minimum distance at least $r-\ell$. Notice that Reed-Solomon codes can be viewed as algebraic geometry codes by taking X to be the projective line, P to be the set of points corresponding to the r chosen field elements, P_0 to be the point at infinity, and $\ell = k-1$.

Concatenated codes are codes which are "concatenated" from two other codes. When two linear codes are concatenated, the product of their lengths (resp., dimensions, resp., minimum distances) is the length (resp., dimension, resp., minimum distance) of the (linear) concatenated code. There are linear concatenated codes for small alphabets which have good list decoding capabilities, i.e., a small list of possible codewords can be recovered even when a large percentage of the symbols are in error or have been erased [18].

We refer the reader to [16, 26, 35, 41] for more information on coding theory.

2.4. **Decoding.** In the theory of error-correcting codes, a codeword is transmitted through a noisy channel and an element of Q^r (i.e., a *word*) is received. The receiver (or *decoder*) then tries to determine as accurately as possible which codeword was transmitted. In *maximum-likelihood decoding*, the decoding process consists of finding the closest codeword to the received word. If d is the minimum distance of the code, then the receiver can "correct" $\frac{d-1}{2}$ errors; i.e., there is at most one codeword within distance $\frac{d-1}{2}$ of the received word. In the maximum-likelihood decoding decision problem, the inputs are a linear code over a given finite field, a received word, and a specified distance t, and the output is a yes or no answer to the question of whether there is a codeword within distance t of the received word. This decision problem is known to be NP-complete [3].

In *list decoding*, the goal is to output the list of all codewords within a specified distance of the received word. In [39] and [40], Sudan gave the first efficient methods for list decoding that run in time polynomial in the length of the codewords. Since then, Sudan's list decoding technique has been improved, generalized, and refined [32, 33, 17, 18, 19, 20, 22, 28, 30, 44, 10, 11]. The runtimes for the steps of the algorithm have been improved, the number of errors that can be "corrected" has been increased, and the technique has been shown to be applicable to a larger class of codes. Sudan's original algorithm is for Reed-Solomon codes. Other codes for which the techniques have been shown to apply include AG codes (for which the focus has been on Hermitian codes) and certain concatenated codes (see [18], where the "outer code" is a Reed-Solomon or AG code and the "inner code" is a Hadamard code).

In *erasure decoding*, some positions of the received word are garbled or "erased", and cannot be identified. In this case the decoder knows that errors occurred in those positions.

In *erasure-and-error decoding*, the decoder receives a word with some erasures and some errors, and determines the transmitted word, or a list of possible transmitted words (given some appropriate bounds on the numbers of errors and erasures).

In *soft-decision decoding*, instead of receiving a (*hard-decision*) word, the decoder receives a reliability matrix, that states the probability that any given element of the alphabet was sent in any given position. Using this "soft" information, a soft-decision decoder outputs the most likely transmitted codeword(s).

3. 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 the codeword length r. 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 $O(N^{\frac{c^2}{r}})$). We describe constructions based on Reed-Solomon, algebraic geometry, and concatenated codes.

3.1. **Reed-Solomon codes.** A widely used and extremely important class of codes are the Reed-Solomon codes. This is the class of codes used in compact disks, for example. This is also the class of codes which has received the most attention by coding theorists looking for fast list decoding techniques, and to which Sudan first applied his method. While Sudan's original result is not strong enough to allow us to use list decoding to trace traitors, the later results of Guruswami and Sudan are exactly strong enough to accomplish this.

Theorem 2. *Let C be a Reed-Solomon code of length r and dimension k over a finite field 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 $O(\frac{r^3}{\delta^6})$.*

Proof: 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 1, 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 [17] 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 result. □

We note that further improvements in the runtime are being rapidly produced, and it seems that some of these results will bring the runtime down to $O(rlog^3 r)$, at least in certain cases (see [10]).

3.2. **AG Codes.** In [17], a polynomial-time algorithm for list decoding an AG code defined from a nonsingular plane curve is given. This algorithm depends on the (reasonable) assumption that a certain amount of pre-processing has occurred which provides the decoder with some additional information about the code, for example a list of certain rational functions on the curve. Under this same assumption, we have:

Theorem 3. *Let X be a nonsingular plane curve of genus g defined over a finite field F_q, P a set of r F_q-rational points on X, P_0 an F_q-rational point on X which is not in 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 above has occurred. Then the one-point AG code $C_X(P, (k+g-1)P_0)$ is a c-TA code with a traitor tracing algorithm that runs in time polynomial in r.*

Proof: The minimum distance d of the code satisfies $d \geq r - k - g + 1$ (see Theorem 10.6.3 of [26]). 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 [17], 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. The result now follows from Theorem 1. □

The list decoding algorithm in [17] for AG codes was improved in [44] (see Theorems 3.4 and 4.1), where an explicit runtime was also given.

3.3. **Concatenated Codes.** As pointed out earlier, there exist nice linear concatenated codes for small alphabets which have good list decoding capabilities.

Theorem 4. *Given a prime power q and positive integers k and c such that $q > c^2 \geq 4$, and given a real number δ such that $0 < \delta \leq \frac{q/c^2 - 1}{q-1}$, then there exists an explicit linear c-TA code over the field 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: Theorems 7 and 8 and Corollaries 2 and 3 of [18] imply that there exists an explicit concatenated code over 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 therefore follows from Theorem 1. □

4. Finding all Possible Coalitions

In this section, we describe how a coding theoretic approach can be used to amass additional piracy information: a list of all (minimal) coalitions that are capable of creating a given pirate. Such information is useful in two respects. Codewords not appearing in any of these coalitions were not involved in constructing the pirate word, and it constitutes useful audit information that may be helpful in the prosecution of a traitor later on. In addition, the algorithm we present enables the IPP traitor tracing algorithm [21, 1, 34] 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 algorithm builds a "tree" from which all c-coalitions capable of constructing w can be extracted. At the root of the tree lie all codewords that we know must be in *any* such coalitions. 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, 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 more detail, we introduce some of the ideas used. If S is a subset of $\{1, \ldots, r\}$ and $s = |S|$, define a map $f_S : F_q^r \to F_q^{r-s}$ by "forgetting" the entries in positions corresponding to elements of S. If C is a code, then the image code $f_S(C)$ is the *punctured code*, where we view the code C as having been punctured at the positions corresponding to the elements of S. If u is in $f_S(C)$ we call a *lift* of u to C any codeword v such that $f_S(v) = w$.

We say that C_0 is a *minimal* c-coalition for w if $|C_0| \leq c$, $w \in \mathsf{desc}C_0$, but w is not in $\mathsf{desc}C_i$ for any proper subset C_i of C_0.

Algorithm Sketch:

Input: positive integer c, Reed-Solomon Code C of length r having N codewords 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 which can create w, that includes all minimal c-coalitions for w.

The basic steps of the algorithm are as follows:

(i) Use list decoding to find all codewords $u_1, ..., u_a \in C$ $(a \leq c)$ 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 $\{u_1, ..., u_a\}$, and let $s = |S|$. Let $r_1 = r - s$, $c_1 = c - a$, $C_1 = f_S(C)$, and $w_1 = f_S(w)$. (Thus C_1 is the punctured code, r_1 is its length, w_1 is the image of the pirate word in C_1, and c_1 is the number of coalition members still to be found.) If $r_1 = 0$, quit and output $\{u_1, ..., u_a\}$. Set $i = 1$.

(ii) Use list decoding to find all codewords $v_{i1}, ..., v_{ib_i} \in C_i$ $(b_i \leq c_i)$ within distance $r_i - r_i/c_i$ of w_i. (Note that the first time this is executed, the output is non-empty.) 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 let $s_i = |S_i|$. Let $r_{i+1} = r_i - s_i$, $c_{i+1} = c_i - 1$, $C_{i+1} = f_{S_i}(C_i)$, and $w_{i+1} = f_{S_i}(w_i)$.

(iii) To create the coalitions to output, always start with $u_1, ..., u_a$. Then add (a lift to C of) v_{1b_1}, v_{2b_2}, and so on. Continue until the list of codewords "covers" the pirate w. When this process succeeds or dead-ends (i.e, the current list does not yet cover w, but either we cannot find any codewords within the required distance $r_i - r_i/c_i$ of w_i, or we already have c codewords in our list), then move up the "tree" of v_{ib_j}'s to find the first unexplored branch and continue from there. The algorithm terminates when all branches have been explored.

Analysis of the Algorithm:

By Theorem 2, Step (i) can be done efficiently (time polynomial in r). By Theorem 1, $u_1, ..., u_a$ are in every coalition that can create w. Further, in Step (ii), if $d_i > r_i - r_i/c_i^2$ where d_i is the minimum distance of the new (punctured) code C_i, then every coalition that can produce the original pirate w will contain some lift to the original code of some v_{i,b_j}. Moreover, if a lift to C of v_{i,b_j} 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_{i,b_j} (by the pigeonhole principle), and the algorithm will proceed. If Step (ii) returns the empty-set, then v_{i,b_j} was a dead-end. Note that list decoding a punctured code and then lifting accomplishes the same as erasure-and-error decoding. One may therefore use erasure-and-error decoding algorithms to accomplish this step. Any codeword found in Step (i) of the algorithm (and at least one codeword must be found in this step) must appear in every coalition considered by the algorithm. Hence the algorithm will certainly not consider the coalitions that do not include the codeword(s) produced in Step (i), and the number of such coalitions is at least $\binom{N-1}{c} = \frac{N-c}{N}\binom{N}{c}$, where N is the number of codewords. If $N \gg c$, then the number of coalitions that are not considered by the algorithm is $\Theta(\binom{N}{c})$. The algorithm is therefore a significant improvement over the brute force method.

5. The TA and IPP Tracing Algorithms

The results in this section justify a focus on TA (as opposed to IPP) schemes. In this paper we have been using the additional structure provided by linear codes to construct schemes for which the TA tracing algorithm is efficient. We know by Lemma 1 that c-TA codes are also c-IPP codes. However the converse fails ([34]; see also Example 1 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 structure to the scheme. Here we give evidence that doing so may enable the inherently more efficient TA algorithm to be used to identify traitors. In particular, we show that one natural approach to adding such structure, that is via Reed-Solomon codes, fails to construct IPP schemes that are not also TA schemes. Hence, since it is unclear that c-IPP schemes yield any advantage over c-TA schemes, most of our work focuses on the latter.

First, we prove a necessary condition on the minimum distance of Reed-Solomon codes, under which Reed-Solomon codes yield c-TA set systems. This result suggests a potential method for generating examples of schemes that are c-IPP but not c-TA. Next, we demonstrate through a family of counterexamples that in fact this approach does not work; as soon as the minimum distance is decreased it is possible to find examples of codes where both the IPP and TA tracing algorithms fail.

We first recall that there is a natural way to produce unordered sets from the ordered sets that constitute the code: to a codeword $x = (x_1, ..., x_r)$, associate the set $x' = \{(1, x_1), ..., (r, x_r)\}$. 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 $i = 1, ..., r$.

The following theorem is a partial converse of Theorem 1.

Theorem 5. *If c is an integer, $c \geq 2$, 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 $u \in C$, write $u' = \{(1, u_1), ..., (r, u_r)\}$ 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 which does not contain v', but which 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, which agrees with v on the positions $(i-1)\delta + 1, \ldots, i\delta$. (To do this, simply find a polynomial h_i of degree δ which 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 which 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 can therefore form a pirate set w so 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. □

The previous theorem 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 following family of counterexamples illustrates that this is not generally the case. The key idea behind the examples is that if the underlying field is sufficiently large, a function defined as the difference between two polynomials of degree α, has α roots in the field. From this, it can be argued that there exist two disjoint collections of c polynomials, such that if a polynomial is chosen from each collection, the resulting pair agrees on as many distinct points as their degree. In addition, the sets of points of agreement of each pair are disjoint. Hence, we can generate two disjoint coalitions that are capable of creating the same pirate word.

Theorem 6. *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 F_q. Then T consists of $c^2 s$ distinct elements of 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 F_q of degree at most s. Then C is not c-IPP. The length r of the codewords is $r = c^2 s$, the dimension of the code is $s + 1$, and the minimum distance is equal to $r - r/c^2$.*

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 $(j-1) = (n-1)c$, 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 define 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 which is a divisor of p^s.

The results in this section lead to the following questions which, while peripheral to the traitor tracing problem, are of independent interest. Is it the case that all Reed-Solomon codes of length r with minimum distance $d \leq r - r/c^2$ are not c-IPP? It is easy to see that this is false for linear codes in general. 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 the above question were yes, combining it with Theorem 1 would imply that all Reed-Solomon c-IPP codes are c-TA. We raise as an open question whether all *linear* c-IPP codes are c-TA.

6. 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.

In [17], 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, [18] also deals with the problem of decoding from errors and erasures. Building on [17], [22] 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 user 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 [22] 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 [22]. If one does not know which entry was contributed by the user 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.

Another possible approach to tracing traitors is to try to second-guess their strategy. For example, if you believe that one traitor has contributed more than the other members of the coalition, you can apply maximum-likelihood decoding 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. The optimal decoding algorithm to use will depend on the radius of the Hamming ball. These algorithms can be run in parallel or sequentially.

Erasure-and-error decoding may be useful in fingerprinting or watermarking scenarios, such as those presented in [5, 6, 13]. In one model for such a scenario, 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.

7. Conclusion

In this paper 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, the length of the code, rather than in the much larger parameter N, the number of codewords. In addition, list decoding on successive punctured codes gives a method for identifying all possible traitor coalitions of size at most c more efficiently than a brute force search (which runs in $O(\binom{N}{c})$ time). Finally, we suggest avenues for future research in this area, 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 Gui-Leng Feng, Tom Høholdt, Ralf Kötter, and Madhu Sudan for useful conversations.

References

[1] A. Barg, G. Cohen, S. Encheva, G. Kabatiansky and G. Zémor. A hypergraph approach to the identifying parent property: the case of multiple parents, DIMACS Technical Report 2000-20.

[2] O. Berkman, M. Parnas and J. Sgall. Efficient dynamic traitor tracing, in 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2000), 586–595.

[3] E. R. Berlekamp, R. J. McEliece and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory* **24** (1978), 384–386.

[4] 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.
[5] 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.
[6] D. Boneh and J. Shaw. Collusion secure fingerprinting for digital data, *IEEE Transactions on Information Theory* **44** (1998), 1897–1905.
[7] B. Chor, A. Fiat and M. Naor. Tracing traitors, in "Advances in Cryptology – Crypto '94", *Lecture Notes in Computer Science* **839** (1994), 480–491.
[8] B. Chor, A. Fiat, M. Naor and B. Pinkas. Tracing traitors, *IEEE Transactions on Information Theory* **46** (2000), 893–910.
[9] C. Dwork, J. Lotspiech and M. Naor. Digital Signets: Self-Enforcing Protection of Digital Information, in Proc. 28th ACM Symposium on Theory of Computing (STOC 1997), 489–498.
[10] G.-L. Feng. Very Fast Algorithms in Sudan Decoding Procedure for Reed-Solomon Codes. Preprint.
[11] G.-L. Feng. Fast Algorithms in Sudan Decoding Procedure for Hermitian Codes. Preprint.
[12] A. Fiat and M. Naor. Broadcast Encryption, in "Advances in Cryptology – Crypto '93", *Lecture Notes in Computer Science* **773** (1994), 480–491.
[13] A. Fiat and T. Tassa. Dynamic traitor tracing, in "Advances in Cryptology – Crypto '99", *Lecture Notes in Computer Science* **1666** (1999), 354–371.
[14] 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* **1666** (1999), 372–387.
[15] J. Garay, J. Staddon and A. Wool, Long-Lived Broadcast Encryption, in "Advances in Cryptology – Crypto 2000", *Lecture Notes in Computer Science* **1880** (2000), 333-352.
[16] V. D. Goppa. Geometry and codes. Kluwer Academic Publishers, Dordrecht, 1988.
[17] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes, *IEEE Transactions on Information Theory* **45**(6) (1999), 1757–1767.
[18] 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.
[19] T. Høholdt and R. R. Nielsen. Decoding Reed-Solomon codes beyond half the minimum distance, in Coding theory, cryptography and related areas (Guanajuato, 1998), Springer, Berlin (2000), 221–236.
[20] T. Høholdt and R. R. Nielsen. Decoding Hermitian codes with Sudan's algorithm. To appear in the 13th AAECC Symposium.
[21] 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.
[22] R. Koetter and A. Vardy. Algebraic soft-decision decoding of Reed-Solomon codes. Preprint. `http://www.dia.unisa.it/isit2000/lavori/455.ps`.
[23] R. Kumar, S. Rajagopalan and A. Sahai. Coding constructions for blacklisting problems without computational assumptions, in "Advances in Cryptology – Crypto '99", *Lecture Notes in Computer Science* **1666** (1999), 609–623.
[24] K. Kurosawa, M. Burmester and Y. Desmedt. A proven secure tracing algorithm for the optimal KD traitor tracing scheme. DIMACS Workshop on Management of Digital Intellectual Properties, April, 2000, and Eurocrypt 2000 rump session.
[25] K. Kurosawa and Y. Desmedt. Optimal traitor tracing and asymmetric schemes, in "Advances in Cryptology – Eurocrypt '98", *Lecture Notes in Computer Science* **1438** (1998), 145–157.
[26] J. H. van Lint. Introduction to coding theory. Third edition. Graduate Texts in Mathematics **86**, Springer-Verlag, Berlin (1999).
[27] M. Naor and B. Pinkas. Efficient Trace and Revoke Schemes, to appear in Proceedings of Financial Crypto 2000.
[28] V. Olshevsky and A. Shokrollahi. A displacement structure approach to efficient decoding of algebraic geometric codes, in Proc. 31st ACM Symposium on Theory of Computing (STOC 1999), 235–244.
[29] B. Pfitzmann. Trials of traced traitors, in Information Hiding, First International Workshop, *Lecture Notes in Computer Science* **1174** (1996), 49–64.
[30] 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.
[31] R. Safavi-Naini and Y. Wang. Sequential Traitor Tracing, in "Advances in Cryptology – CRYPTO 2000", *Lecture Notes in Computer Science* **1880** (2000), 316–332.
[32] M. A. Shokrollahi and H. Wassermann. Decoding Algebraic-Geometric Codes Beyond the Error-Correction Bound, in Proc. 30th ACM Symposium on Theory of Computing (STOC 1998), 241–248.
[33] M. A. Shokrollahi and H. Wassermann. List Decoding of Algebraic-Geometric Codes. *IEEE Transactions on Information Theory* **45** (1999), 893–910.
[34] J. N. Staddon, D. R. Stinson and R. Wei. *Combinatorial properties of frameproof and traceability codes.* To appear in *IEEE Transactions on Information Theory.*
[35] H. Stichtenoth. Algebraic Function Fields and Codes. Springer-Verlag, Berlin, 1993.
[36] D. R. Stinson, Tran van Trung and R. Wei. Secure frameproof codes, key distribution patterns, group testing algorithms and related structures, *Journal of Statistical Planning and Inference* **86** (2000), 595-617.

[37] D. R. Stinson and R. Wei. Combinatorial properties and constructions of traceability schemes and frameproof codes, *SIAM Journal on Discrete Mathematics* **11** (1998), 41–53.
[38] D. R. Stinson and R. Wei. Key preassigned traceability schemes for broadcast encryption, in "Selected Areas in Cryptology – SAC '98", *Lecture Notes in Computer Science* **1556** (1999), 144–156.
[39] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity* **13**(1) (1997), 180–193.
[40] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction diameter, in Proc. 35th Annual Allerton Conference on Communication, Control and Computing (1997), 215–224.
[41] M. A. Tsfasman and S. G. Vlăduţ. Algebraic-geometric codes. Kluwer Academic Publishers, Dordrecht, 1991.
[42] M. A. Tsfasman, S. G. Vlăduţ and Th. Zink. Modular curves, Shimura curves, and Goppa codes, better than Varshamov-Gilbert bound. *Math. Nachr.* **109** (1982), 21–28.
[43] W.-G. Tzeng and Z.-J. Tzeng. A Traitor Tracing Scheme Using Dynamic Shares, to appear in PKC2001.
[44] X.-W. Wu and P. H. Siegel. Efficient List Decoding of Algebraic Geometric Codes Beyond the Error Correction Bound, submitted to *IEEE Transactions on Information Theory*.

Dept. of Mathematics, Ohio State University and MSRI.
E-mail address: silver@math.ohio-state.edu *URL:* http://www.math.ohio-state.edu/∼silver

E-mail address: jstaddon@yahoo.com

Dept. of Mathematics and Statistics, University of Nebraska, Lincoln, NE 68588-0323.
E-mail address: jwalker@math.unl.edu *URL:* http://www.math.unl.edu/∼jwalker