Efficient Methods for Integrating Traceability and Broadcast Encryption

Eli Gafni[1⋆], Jessica Staddon[2*], and Yiqun Lisa Yin[3*]

[1] University of California at Los Angeles
eli@cs.ucla.edu
[2] Bell Laboratories Research Silicon Valley
jstaddon@yahoo.com
[3] NTT Multimedia Communications Laboratories
yiqun@nttmcl.com

Abstract. In many applications for content distribution, broadcast channels are used to transmit information from a distribution center to a large set of users. Broadcast encryption schemes enable the center to prevent certain users from recovering the information that is broadcast in encrypted form, while traceability schemes enable the center to trace users who collude to produce pirate decoders. In this paper, we study general methods for integrating traceability and broadcasting capability. In particular, we present a method for adding *any* desired level of broadcasting capability to *any* traceability scheme and a method for adding *any* desired level of traceability to *any* broadcast encryption scheme. To support our general methods, we also present new constructions of broadcast encryption schemes which are close to optimal in terms of the total number keys required. Our new schemes are the first to be both maximally resilient and fully scalable.

1 Introduction

In many applications for content distribution, broadcast channels are used to transmit a message from a distribution center to a large set of users. It is often desirable for the center to be able to exclude certain users from recovering the message that is broadcast in encrypted form. One such example is the pay television industry, in which only privileged users (i.e., active subscribers) are permitted to view shows. Many solutions to this problem have been proposed using broadcast encryption schemes [3,9,10,4,5,15,13,20,11,19,16,6,1]. In such schemes, keys are allocated to users in such a way that broadcasts can be made to selected sets with security. To broadcast to the selected set, a subset of the encryption keys is used to encrypt the message based on the protocol being used. The basic attribute of a broadcast encryption scheme is its *broadcasting capability*, which is generally measured by the number of users that can be prevented from recovering the message from the broadcast.

⋆ Most of this work was done while the authors were working at RSA Laboratories.

Michael Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 372–387, 1999.

Clearly, security is an important attribute of broadcast encryption schemes. Two commonly used measures of security are *resiliency* [9] and *traceability* [7,17,12]. A scheme is said to have m-resiliency if no set of at most m *excluded* users can pool their keys together to recover a message from a broadcast. A scheme is said to have c-traceability if when a set of at most c users (who are not necessarily excluded) pool their keys together to construct a "pirate decoder", at least one of the users involved can be identified by examining the keys in the decoder. Traceability can offer protection against the piracy that is often a serious problem in content distribution applications.

Although a natural goal for constructing broadcast encryption schemes is to have both high broadcasting capability and high traceability, these two attributes have been studied separately in the past, with the exceptions of [13,18]. In [13], Staddon determines the traceability of various specific broadcast encryption schemes and proves lower bounds on the traceability of certain (protocol dependent) broadcast encryption schemes. In [18], Stinson and Wei develop the first method for combining the two attributes by adding broadcasting capability to a given traceability scheme. Their method is quite general in that the construction is based on an arbitrary traceability scheme.

The first contribution of this paper is to study general methods for integrating traceability and broadcasting capability. In particular, we approach the integration problem from both directions: (1) we develop the first method for adding *any* desired level of traceability to an *arbitrary* broadcast encryption scheme; (2) we develop a new method for adding *any* desired level of broadcasting capability to an *arbitrary* traceability scheme.

The central idea behind our method for adding traceability to broadcast encryption schemes is that using "randomness" when allocating keys to users allows the users' key sets to be dispersed, and hence, is conducive to traceability. Based on this observation, our method adds a "dimension" of randomness to an underlying broadcasting encryption scheme. In the other direction, our method adds adding broadcasting capability to a traceability scheme. The main idea behind the latter method is to leverage on the inherent broadcasting capability in the underlying traceability scheme. We show that by exploiting such inherent broadcasting "structure", significant efficiency improvements can be achieved over the method in [18].

For both of the general methods that we present here, keys are allocated to users according to a certain matrix. The keys appear to be randomly assigned to users along one dimension of the matrix but well structured along the other dimension. The random dimension contributes to traceability and the structured dimension contributes to broadcasting capability. Hence, the two methods are complementary to each other and are conceptually quite simple.

An important feature of these methods is their preservation of the properties of the underlying broadcast encryption schemes. In addition to resiliency, another important property is *full scalability*. This means that the set of keys for each existing user remains unchanged when new users are introduced into the system. While scalability is clearly a desirable attribute for any large con-

tent distribution system, it has been largely ignored in the context of broadcast encryption. Our second contribution is to propose two new maximally-resilient fully-scalable broadcast encryption schemes, to which one may add traceability by our general method.

One of our schemes is based on a geometric construction and the other on an algebraic construction. Both schemes employ the so-called "*OR* protocols" [1,10,11], which have the desirable property of yielding maximally resilient schemes. We show that our new schemes are close to optimal with respect to the total number of keys by proving a lower bound that almost matches this number. This lower bound is obtained by demonstrating a concise combinatorial characterization of broadcast encryption systems with *OR* protocols. These results establish a relationship between the number of keys per user, r, and the total number of keys, K. Although individual bounds on r and K have been shown [11], the relationship between the two has not been studied prior to our work.

The organization of this paper is as follows: Section 2 contains notation and definitions. Section 3 summarizes related work in broadcast encryption and traceability schemes. Section 4 describes the new broadcast encryption schemes and proves a tight lower bound relating the number of keys per user and the total number of keys. Section 5 presents the general methods for integrating broadcasting capability and traceability.

2 Preliminaries

In this section, we provide the notation and definitions for broadcast encryption schemes and their attributes. At a very high level, a broadcast encryption scheme consists of users, keys, a key allocation method for assigning keys to users, and a broadcast protocol that the center uses to transmit information to certain sets of users.

Let $\{u_1, ..., u_n\}$ denote the set of all users. We call the users who have the permission to receive a message that's broadcast in encrypted form, the set of *privileged users*, and the users who don't have permission, *excluded users*. We use $\mathcal{P}$ to denote the collection of privileged sets of users and m to denote the number of excluded users. So, $\mathcal{P}$ is the collection of all subsets of users of size $n - m$.

Let $S = \{k_1, ..., k_K\}$ denote the set of all keys. The set of keys assigned to user u, is denoted by $U \subseteq S$. Since we mostly focus on the maximum number of keys per user as an important measure of the efficiency, it is without loss of generality that we assume all users have the maximum number of keys, r. That is, for each user u, $|U| = r$.

For a set of privileged users, $P \in \mathcal{P}$, the set of keys that the center uses to broadcast to P will be denoted by $S_P \subseteq S$. The *number of transmissions* for a broadcast encryption scheme is defined to be $t = \max_{P \in \mathcal{P}} |S_P|$. This is the number of keys used in the communication.

In most applications for content distribution, the center first establishes a broadcast key, B_P, with the set of privileged users P, and encrypts subsequent broadcasts with the broadcast key. For each privileged set P, there is a *broadcast protocol* which defines which subsets of keys in S_P are used to encrypt and recover B_P. Hence, a protocol yields an access structure on S_P because it defines which subsets of S_P suffice to recover B_P. Therefore, to implement any protocol for broadcasting to P, one can use the keys in $S_P = \{k_1, ..., k_t\}$, to generate shares, $B_P^1, ..., B_P^t$, according to the access structure and the choice of secret sharing scheme (see [14] for more on access structures and secret sharing). Each share B_P^i is then encrypted in a computationally secure way, so that key k_i is necessary to decrypt it. We assume that a user u for which $k_i \notin U$ gains no information about B_P^i from its encrypted form.

In this paper, we concentrate on *OR* protocols. If the center is broadcasting to a set P with an *OR* protocol, then a user needs only one out of the t keys in S_P to decrypt B_P. Consequently, to implement *OR* protocols, one can use a $(1, t)$-threshold scheme to generate the shares[1]. We focus on *OR* protocols because a broadcast encryption scheme that employs them is secure against arbitrary coalitions of excluded users. An excluded user has none of the keys in S_P, therefore a coalition of excluded users of arbitrary size still cannot recover B_P (consequently, *OR* protocols are said to be *arbitrarily resilient*).

Clearly, many other protocols are possible. In fact, any formula of a certain form (see [11]) defines a protocol. For example, in an *AND* protocol, all the keys in S_P are necessary to recover B_P. An *AND* protocol can be implemented with a (t, t)-threshold scheme.

As discussed earlier, traceability offers a form of security that is complementary to resiliency. Traceability protects against a coalition of users, $\mathcal{C}$, who build a "pirate decoder", F. The decoder can be modeled as a subset of their pooled keys. That is, $F \subseteq \cup_{u \in \mathcal{C}} U$, such that $|F| \geq r$. In addition to the basic components in broadcast encryption, a traceability scheme also consists of an algorithm which identifies one user in $\mathcal{C}$ by analyzing the keys in F. Informally, we say that a scheme has c-traceability if when the size of $\mathcal{C}$ is at most c, at least one of the users involved in coalition can be identified with very high probability. In other words, an innocent user will be identified as "guilty" with only negligible probability. This is called a *c-resilient* traceability scheme in [7].

Definition 1. *Let $\mathcal{C}$ be any coalition of at most c users who produce a pirate decoder F. A scheme is called a* **c-traceability scheme** *if for any user u, such that for all users $w \neq u$ the following inequality holds:*

$$|U \cap F| \geq |W \cap F|$$

then the probability that u is not a member of the coalition $\mathcal{C}$ is negligible.

Another desirable attribute of a broadcast encryption or a traceability scheme is that it scale well. This means that as the number of users grows, only a small

[1] Note that we've described a natural way to implement an *OR* protocol and from the results in [10] it follows that it is as efficient as possible.

amount of rekeying is necessary for the old users. In certain systems, it might be required that no rekeying is needed.

Definition 2. *A scheme is* **fully scalable** *or has* **full scalability** *if when new users are added, no rekeying of existing users is necessary.*

For ease of notation, we assume that when discussing a scheme, the parameters of the scheme (such as the number of keys per user, etc.) are represented by the notation summarized in Table 1, unless otherwise specified.

Summary of Terms and Notation

- $\{u_1, ..., u_n\}$ is the set of all users.
- $\{k_1, ..., k_K)$ is the set of all keys.
- S_P is the set of keys used to broadcast to privileged set P.
- B_P is the message (e.g., a broadcast key) that is broadcast to P in encrypted form.
- n is the total number of users.
- K is the total number keys.
- r the number of keys per user.
- m is the number of users who are excluded.
- t is the number of transmissions. Note that $|S_P| \leq t$.
- c is the traceability of the scheme.
- OR Protocol for Broadcasting to P: Any one of the keys in S_P suffices to recover B_P from the broadcast.
- AND Protocol for Broadcasting to P: All of the keys in S_P are necessary to recover B_P from the broadcast.

Table 1. Summary of Terms and Notation

3 Related Work

3.1 Broadcast Encryption

The early works in broadcast encryption are [3,9,10,4]. In [3], a *one-time* broadcast encryption scheme is presented. It can be used once with security as information about each user's key is leaked to the privileged set during broadcast. Our model for broadcast encryption is a formalization of the one in [9]. In [9], the concept of resiliency is formalized, and broadcast encryption schemes of various resiliencies are constructed. In [10], the authors consider broadcast encryption schemes with OR protocols (although this terminology is not used) and prove that the entropy of a broadcast is at least the size of the entropy of the message times the number of users in the privileged set. The work in [4] concentrates on broadcast encryption schemes in which only one transmission is needed by

the broadcasting center (called *zero-message schemes*) and on schemes in which users interact. Some information theoretic lower bounds are also derived.

In several subsequent works [5,15,13,11,19,16], the trade-off between communication cost and storage in broadcast encryption is studied. Many new schemes are proposed, some of which are combinatorial in nature. In [5,15,19,16], the trade-off is measured using an information theoretic ratio, while in [13,11] it is measured through a comparison of the number of keys (per user and in total) versus the number of keys used in the communication. Consequently, schemes that are optimal under one measurement may not be optimal under the other. We note that most of the schemes in [5,13,19,16] use (as a component) a construction in [9] that does not scale very well.

The recent work in [1] focuses on constructing broadcast encryption systems in which the user storage is very limited. In their proposed systems, the reduction in storage is achieved by allowing a controlled number of excluded users to receive the broadcast. They prove some lower bounds under this framework and and present an algorithm for efficiently finding such schemes while minimizing the communication cost.

A quite different approach to solving the problem of broadcast encryption appears in [20,6]. The model differs from all of the above mentioned works in that when some user is removed from the system, keys of existing users are updated (called *rekeying*). In the Internet draft [20], a hierarchical tree-based scheme is recommended for use in a broadcast encryption system. The system is maximally resilient but not fully scalable. This work is later built upon in [6], which demonstrates a method for reducing center's storage in the tree-based scheme by considering the trade-off between storage and the rekeying communication cost.

Our model is consistent with those in [13,11,1], and is a formalization of the one in [9]. These works ([13,11,1]) focus on two important quantities: the number of keys per user, and the total number of keys. These are important quantities because they give a concrete bound on storage requirements which is very useful for implementation. In addition, when *OR* protocols are used (as in this paper), the resulting broadcast size is just a multiple of the number of transmissions, therefore bandwidth is a straightforward calculation. Although the schemes in this paper are not tight with the bound in [11], no broadcast encryption schemes with *OR* protocols are known that are tight with this bound for the number of transmissions required by these schemes (see Table 1). In addition, there is evidence that the bound in [11] is not tight for certain values of the parameters. For example, if t is on the order of $\sqrt{n}$ and m is small, then to be tight with the bound means that K and r must be essentially 1, which is clearly impossible. Further, we emphasize that [11] does not establish a relationship between r and K, but rather, it proves that both r and K are $\Omega(\binom{n}{m}^{1/t})$. In addition, our schemes are fully scalable and allow the implementor complete control over the number of keys per user. These are not features of any other maximally resilient broadcast encryption scheme.

In overall comparison with the previous work, we emphasize that our new broadcast encryption schemes (in Section 4) are the first that are both maximally resilient and fully scalable under this model. In addition, the schemes are very flexible in terms of the number of keys per user. We note that in this paper we do not consider other models such as one-time schemes [3], zero-message schemes [4], and schemes that allow rekeying [20,6].

3.2 Traceability Schemes

Traceability schemes are first introduced in [7] and further studied in [17]. Several constructions for traceability schemes are given and lower bounds on the number of keys per user and the total number of keys are proven.

A generalization of traceability called threshold traceability, is considered in [12]. Threshold traceability schemes are designed to trace the source of a pirate decoder which can decrypt with only a probability *larger than some threshold*. By relaxing the decryption probability requirement, a significant reduction in storage and communication is achieved.

Our model for traceability schemes is the same as the one in [7,17]. The methods for integrating broadcast encryption and traceability (in Section 5) can be extended to the generalized model in [12].

3.3 Integrating Broadcast Encryption and Traceability

There are only two previous works [13,18] that study the integration of broadcast encryption and traceability. In [13] the traceability of various specific broadcast encryption schemes is determined and lower bounds on the traceability of certain (protocol dependent) broadcast encryption schemes is proven. The focus of this work is to determine the traceability of certain broadcast encryption schemes, rather than to demonstrate how to achieve a certain level of traceability with a specific broadcast encryption scheme.

In [18] the model of traceability is a generalization of the model in [7]. They allow decoders to hold any number of keys and they allow the set of excluded users to be a *proper* subset of the complement of the set of privileged users. In [18], broadcasting capability is added to a traceability scheme by using a construction in [9] to expand each key into a set of keys. Our method (Method 2 in Section 5) differs from their method in that we take full advantage of the *inherent* broadcasting capability in the underlying traceability scheme, and therefore our method requires much less keys per user than [18] in most situations. We remark, however, that the model in [18] is more general, and hence their method may be applicable to more situations than our method which follows the earlier model in [7].

4 Optimal Broadcast Encryption Schemes with *OR* Protocols

In this section we describe two new constructions for broadcast encryption schemes with *OR* protocols. Recall that *OR* protocols are desirable because of their inherent resiliency. The first construction, which we call the *cube scheme*, is based on a geometric construction. The second one, which we call the *polynomial scheme*, is based on an algebraic construction. Both schemes are fully scalable and m-resilient (due to the use of *OR* protocols. In particular, we emphasize that when new users are added, the set of keys for any existing user remains unchanged. We also show that both schemes are close to optimal in terms of the total number of keys by proving a matching lower bound.

4.1 The Cube Scheme

The cube scheme is a parameterized scheme. For a fixed number of keys per user, r, the construction is based on an r-dimensional cube. Informally speaking, users are represented by *entries* of the cube (i.e., points), and keys are represented by *slices* of the cube (i.e., subspaces of dimension $r-1$).

First we describe the case in which $r = 2$, as it is easier to understand and suggests a natural generalization. Consider a $n^{1/2} \times n^{1/2}$ square, and associate each of the n users with an entry in this square indexed by (i_1, i_2), where $i_1, i_2 \in \{1, 2, ...n^{1/2}\}$. For $1 \leq i \leq n^{1/2}$, let C_i denote the set of users in column i and let R_i denote the set of users in row i. For each i, we create two unique keys and allocate one of the keys only to the users in C_i and allocate the other only to the users in R_i. Therefore, each user has exactly 2 keys. To exclude a given user u, the center broadcasts according to an *OR* protocol with all the keys except the 2 keys stored by user u. Since each two users share at most 1 key, every user except u can receive the broadcast.

We can easily generalize the above scheme to dimension r by associating each user with an entry in an r-dimensional cube. An r-dimensional cube has entries indexed by r-tuples, $(i_1, ..., i_r)$ where each $i_j \in \{1, 2, ..., n^{1/r}\}$. We define a *slice* of the cube to be the $(r-1)$-dimensional analog of rows and columns, that is, a subspace of dimension $r-1$. More precisely, for each pair (j, w) such that $1 \leq j \leq r$ and $1 \leq w \leq n^{1/r}$, we define a slice, $S_{j,w}$:

$$S_{j,w} = \{(i_1, i_2, ..., i_r) : i_j = w\}.$$

In other words, a slice consists of all the r-tuples which are identical in the jth entry. As in the 2-dimensional case, we create a unique key for each slice. Therefore, each user has exactly r keys. To exclude a given user u, the center broadcasts according to an *OR* protocol with all the keys except the r keys that u has. Since each pair of users share at most $r-1$ keys, every user except u can recover B_P from the broadcast. Note that the cube scheme can exclude one user.

We now present a simple extension of the above construction to exclude m users by making "copies" of the cube scheme. Specifically, we assign independent keys to m different r-dimensional cube schemes, therefore each user has rm keys in total. We can exclude m users, $\{u_1, u_2, ..., u_m\}$ by excluding the r keys that user i has in the ith cube scheme. The broadcast protocol is then an *AND* on the union of the sets of keys left in each cube scheme. The resulting scheme is still 1-resilient. In summary, for this scheme, the total number of keys is $K = mrn^{1/r}$, the number of keys per user is mr, the number of transmissions is $K - mr$, and the resiliency is 1.

Finally, we note that the cube scheme and it extension scale well as the number of users grows. For example, we can add $n^{(r-1)/r}$ users by expanding the cube by the size of one slice. This requires the addition of only one new key. The new users are given that new key and old keys corresponding to the other slices in which they are contained. No rekeying is necessary for old users. This is significantly better than in the previously known schemes. For example, in the *OR* scheme in [11], there is a key for each set of $\frac{n-m}{t}$ users. Therefore, adding one new user necessitates the creation of $\binom{n-1}{\frac{n-m}{t}-1}$ new keys, and each old user needs $\binom{n-2}{\frac{n-m}{t}-2}$ new keys.

4.2 The Polynomial Scheme

The polynomial scheme described in this section is a parameterized scheme depending on both r, the number of keys per user, and m, the number of excluded users. The scheme uses a set system construction[2] based on polynomials over a finite field. Speaking informally, users are represented by polynomials and keys are represented by points on the polynomials.

Let p be a prime larger than r, and let A be a subset of the finite field F_p of size r. Consider the set of all polynomials over F_p of degree at most $\frac{r-1}{m}$. (For simplicity, we assume that $m|(r-1)$.) There are $p^{\frac{r-1}{m}+1}$ such polynomials. We associate each of the n users with a different polynomial. Therefore, p needs to satisfy the condition that $p^{\frac{r-1}{m}+1} \geq n$, or equivalently, $p \geq n^{\frac{m}{r-1+m}}$. The keys are created and assigned to users as follows: We create a unique key, $k_{(x,y)}$, for each pair (x, y) where $x \in A$ and $y \in F_p$. Note that the polynomials may be public information, as knowledge of a user's polynomial reveals only the *indices* of that user's keys, not the keys themselves. For a user u who is associated with a given polynomial f, u is allocated all the keys in the set $\{k_{(x,f(x))}|x \in A\}$. Since any two of the polynomials intersect in at most $\frac{r-1}{m}$ points, it follows that any two users share at most $\frac{r-1}{m}$ keys. This ensures that if all the keys belonging to the m excluded users are removed, then each privileged user will still have at least 1 key. Therefore, the center can broadcast with an *OR* protocol to any set of $n - m$ users. In summary, the total number of keys is $K = rp \geq rn^{\frac{m}{r-1+m}}$, the number of keys per user is r, the number of transmissions is at most $K - r$, and the resiliency is m.

[2] The construction appeared in [2] in a purely combinatorial context.

This scheme is also fully scalable, since increasing the size of the field, F_p, allows significantly more users to be added with no rekeying of the old users. For example, if K is doubled, then $2^{\frac{r-1}{m}+1}$ more users can be added to the scheme. The new users will get some of the new keys and some of the old, while the old users key sets will remain unchanged.

Finally, we note that for certain values of the parameters this scheme may be closely related to the cube scheme of the previous section.

4.3 Lower Bound on the Total Number of Keys

In this section, we establish a lower bound on the total number of keys in a broadcast encryption scheme in terms of the number of keys assigned to each user. This lower bound shows that the total number of keys is close to optimal in both the cube scheme and the polynomial scheme. To prove the bound, we first demonstrate a combinatorial characterization of broadcast encryption schemes with *OR* protocols.

Lemma 3. *A collection of n sets can be used as a broadcast encryption scheme with OR protocols that can exclude any set of m users if and only if*

$$\forall \quad U_{i_1}, ..., U_{i_{m+1}} \text{ distinct, } U_{i_1} \not\subseteq \cup_{j=2}^{m+1} U_{i_j}$$

Proof: $\Rightarrow$: Assume we have such a broadcast encryption scheme and there exists a set of $m+1$ users, u_1,...,u_{m+1}, such that $U_1 \subseteq \cup_{j=2}^{m+1} U_j$. Then, if *OR* protocols are used, at least one of u_2,...,u_{m+1} will be able to recover the message from a broadcast to u_1. This is a contradiction.

$\Leftarrow$: If for every set of m users $u_1, ..., u_m$ and for every user, u, outside of this set, $U \not\subseteq \cup_{j=1}^{m} U_j$, then to broadcast to $P = \{u_{m+1}, ..., u_n\}$, let $S_P = S - \cup_{i=m+1}^{n} U_i$. This S_P (or possibly even a subset of it) can be used to broadcast to P with *OR* protocols. □

The following result by Erdös, Frankl, Füredi [8] is very useful in determining the relationship between the parameters of a set system satisfying the condition in the previous lemma.

Theorem 4 ([8]). *Let $U = \{k_1, k_2, ..., k_K\}$ be a set of K elements. Let $U_1, ..., U_n$ be a collection of n subsets of U such that $\forall j$, $|U_j| = r$, and $\forall \quad U_{i_1}, ..., U_{i_{m+1}}$ distinct, $U_{i_1} \not\subseteq \cup_{j=2}^{m+1} U_{i_j}$, then*

$$n \leq \frac{\binom{K}{\lceil r/m \rceil}}{\binom{r-1}{\lceil r/m \rceil - 1}}$$

Combining Lemma 3 and Theorem 4, we can establish a relationship between the total number of keys and the number of keys per user.

Theorem 5. *In a broadcast encryption scheme with OR protocols, the total number of keys, K, is $\Omega((n/m)^{m/r} r)$, where $r \geq m$ is the number of keys per user and m is the number of users that can be excluded in the scheme.*

Proof: From Lemma 3, it follows that any broadcast encryption scheme with *OR* protocols must satisfy the condition of Theorem 4. Then the lower bound on K can be easily derived from the inequality given in Theorem 4. □

The lower bound given in Theorem 5 enables one to first choose the number of keys per user when constructing a broadcast encryption scheme (e.g. based on the storage capabilities of a smart card), and then determine the minimum total number of keys that is necessary. Indeed, this is the approach that we have used in both the cube scheme and the polynomial scheme. Table 2 summarizes these schemes. Based on our lower bound, it is easy to see that both schemes are close to optimal in terms of the total number of keys. We remark that fixing the number of keys per user ahead of time (so that it is independent of the total number of users) is very useful in constructing fully scalable broadcast encryption schemes.

Scheme	**Number of users can exclude**	**Resiliency**	**Total number of keys**	**Number of keys per user**
r-dimensional cube scheme	1	1	$rn^{1/r}$	r
m copies of the cube scheme	m	1	$mrn^{1/r}$	mr
(r, m)-polynomial scheme	m	m	$\geq n^{\frac{m}{r-1+m}} r$	r

Table 2. A Summary of the Broadcast Encryption Schemes in Sections 4.1 and 4.2.

We also note that for certain values of the parameters, Theorem 5 may yield a larger bound on K than is proven in [11], and is, therefore, an improvement. For example, when t is large, the bound in [11] (K is $\Omega(\binom{n}{m}^{1/t})$) is only trivially true, as it is quite small.

5 Integrating Traceability and Broadcast Encryption

In this section, we present two methods for integrating traceability with broadcasting capability. Our methods are both efficient and conceptually quite simple.

In Section 5.1 we describe a method that adds any desired level of traceability to any given broadcast encryption scheme, $\mathcal{B}$. A scheme constructed by this method can be viewed as a two dimensional matrix, in which broadcasting capability is drawn from one dimension, and traceability from the other. In other words, if we assume that all the keys in $\mathcal{B}$ are arranged in one column, then the method extends the column of keys into a matrix in such a way that the horizontal dimension contributes traceability.

In Section 5.2 we describe a method that adds any desired level of broadcasting capability to any traceability scheme, $\mathcal{T}$. A scheme constructed with this

method can also be viewed as a two dimensional matrix. If we arrange all the keys in $\mathcal{T}$ in one row, then this method extends this row of keys into a matrix in such a way that the vertical dimension contributes broadcasting capability.

Together, these complementary approaches solve the problem of integrating traceability and broadcasting capability from both directions.

5.1 Adding Traceability to Broadcast Encryption Schemes

We first consider how much traceability is inherent in a broadcast encryption scheme.

Lemma 6. *Any broadcast encryption scheme that can exclude m ($m \geq 1$) users has at least 1-traceability. In addition, a broadcast encryption scheme that can exclude m users may have no more than 1-traceability.*

Proof: The first statement follows from the definitions. To prove the second statement, it suffices to produce a broadcast encryption scheme with 1-traceability. In [13], a scheme using *AND* protocols is described and it's proven that the scheme has 1-traceability for sufficiently large n. □

From this lemma, it is clear that the traceability of an arbitrary broadcast encryption scheme can be quite limited. We now turn to our method for adding traceability to an arbitrary broadcast encryption scheme.

The schemes in [7] gain traceability from "randomness" in the key assignments. The random nature of the key assignments forces the key sets of the individual users to be distinct enough that traitors can be identified with high probability upon examination of the keys in a decoder. In most broadcast encryption schemes, however, keys are assigned to users in a very structured way. Therefore, the central idea in our method is to incorporate some randomness into the way in which the keys are assigned to users in a broadcast encryption scheme. Our method is motivated by the constructions of traceability schemes in [9]. The method is described in Table 3.

The following theorem gives the precise parameter values for an implementation of our method using the "open one-level" scheme of Fiat and Naor [9], which defines a practical way of assigning the keys in step 2 of Method 1 using hash functions.

Theorem 7. *Let $\mathcal{B}$ be a broadcast encryption scheme with parameters (n, m, K, r, t). If $r > 4c^2 \log n$, then there exists a broadcast encryption scheme, $\mathcal{B}'$, which has c-traceability and parameters (n, m, K', r', t'), where $K' = 2c^2K$, $r' = r$, and $t' = 2c^2t$.*

Proof: All the assertions about $\mathcal{B}'$ except its c-traceability follow from the construction of Method 1 given in Table 3. The argument for traceability is very similar to the argument for the "open one-level" scheme in [7]. In particular, if we set $h = 2c^2$ and a pirate decoder contains at least $s > 4c^2 \log n$ keys, then the probability that a user who has at least $\frac{s}{c}$ keys in common with the decoder is innocent, is negligible. By definition, $\mathcal{B}'$ has c-traceability. □

Method 1

Input:

- a broadcast encryption scheme, $\mathcal{B}$
- an integer, c, the desired level of traceability

Output:

- a broadcast encryption scheme, $\mathcal{B}'$, with c-traceability

Construction:

1. Let $\{k_1, ..., k_K\}$ be the set of keys in scheme $\mathcal{B}$. For each key k_j, create a set of h keys $W_j = \{k_{j,1}, k_{j,2}, ..., k_{j,h}\}$, where h (value to be determined) depends on c.
2. If a user u has key k_j in $\mathcal{B}$, then in $\mathcal{B}'$, u gets one key randomly chosen from the set W_j.
3. To broadcast a secret B_P to a set of privileged users, P, where $S_P = \{k_{i_1}, ..., k_{i_t}\}$, the center first generates shares $B_P^1, ..., B_P^t$, according to the protocol used for P in $\mathcal{B}$ (as described in Section 2). Then for each key k_{i_j} in S_P, the center encrypts B_P^j with each of the keys in the set W_j.

Table 3. A method for integrating traceability into broadcast encryption schemes.

We emphasize here that this method is not specific to the "open one-level" scheme in [9]. Rather, all that is needed to execute this method is a mechanism for assigning the keys in step 2 of Method 1. For example, another scheme such as the "open two-level" scheme in [9] may be used as well.

5.2 Adding Broadcasting Capability to Traceability Schemes

In this section, we take an approach that's similar to the one in Section 5.1, by analyzing how much broadcasting capability is inherent in a traceability scheme. We start by considering some combinatorial properties of both types of schemes. The following lemma is used in [7] and [17] to prove lower bounds on the number of users in a c-traceability scheme.

Lemma 8 ([7,17]). *In a c-traceability scheme with users u_1,...,u_n, the following must be true:*

$$\forall \quad U_{i_1}, ..., U_{i_{c+1}} \text{ distinct, } U_{i_1} \not\subseteq \cup_{j=2}^{c+1} U_{i_j}$$

Using the above lemma and Lemma 3, we prove a result on the broadcasting capability inherent in an arbitrary traceability scheme.

Theorem 9. *A c-traceability scheme can be used as a broadcast encryption scheme with OR protocols that can exclude any set of m users for any $m \leq c$.*

Method 2

Input:

- c-traceability scheme, $\mathcal{T}$
- an integer, m, the desired number of excluded users
  (for simplicity, we assume that $s = m/c$ is an integer.)

Output:

- a c-traceability scheme, $\mathcal{T}'$, which can exclude m users

Construction:

1. Let $\{k_1, ..., k_K\}$ be the set of keys in $\mathcal{T}$. For each $j = 1, ..., s$, create independent sets of keys $\{k_{j,1}, ..., k_{j,K}\}$. These sets of keys can be viewed as the copies of the scheme $\mathcal{T}$.
2. If a user u has key k_ℓ in $\mathcal{T}$, then in $\mathcal{T}'$, u is allocated the s keys, $k_{1,\ell}, ..., k_{s,\ell}$.
3. Let E be a set of m users to be excluded and let $P = \{u_1, ..., u_n\} - E$ be the set of privileged users. Partition E into s subsets $E_1, ..., E_s$ such that each E_i has size $c = m/s$. Let $P_i = \{u_1, ..., u_n\} - E_i$.
4. To broadcast a secret B_P to set P, the center first generates s shares of B_P, B_P^1,...,B_P^s, such that all s shares are necessary to recover B_P (i.e. it's an (s, s)-threshold scheme). Then the center broadcasts B_P^i to the set P_i in accordance with the protocol for P_i in $\mathcal{T}$ (by Theorem 9 this could be an *OR* protocol).

Table 4. A method for integrating broadcasting capability into traceability schemes.

Proof: If for some $m \leq c$, there exist distinct sets $U_1, ..., U_{m+1}$ such that $U_1 \subseteq \cup_{j=2}^{m+1} U_j$ then clearly those sets cannot be part of a c-traceability scheme. The result follows from Lemma 3. □

Hence, a c-traceability scheme can easily be used to construct a broadcast encryption scheme that can exclude any set of c users. Since the resulting scheme is based on *OR* protocol, it's also c-resilient.

To achieve more broadcasting capability (i.e. the ability to exclude more users), we need to add more "structure" to the way in which keys are assigned to users. A simple method for accomplishing this is to make "copies" of a single traceability scheme. This method is presented in Table 4.

The traceability of the scheme constructed by Method 2 is inherited from the traceability scheme that is input to the method. Informally, this is because the number of keys per user grows with the number of copies of the original traceability scheme. Therefore, a sufficiently large number of the keys in a decoder must all be contained in one of the copies, and then a traitor tracing algorithm can be applied to those keys.

Theorem 10. *Let $\mathcal{T}$ be a traceability scheme with parameters (n, c, K, r, t) and broadcasting capability c. Then there exists a traceability scheme $\mathcal{T}'$ which*

has broadcasting capability m *and parameters* (n, c, K', r', t')*, where* $K' = \frac{mK}{c}$, $r' = \frac{mr}{c}$, *and* $t' = \frac{mt}{c}$.

Proof: We first show the broadcasting capability of $\mathcal{T}'$. Since for any excluded user u, there exists a j such that $u \notin P_j$, u is unable to obtain B_P^j, and hence, u is unable to obtain the message B_P.

To see that $\mathcal{T}'$ has c-traceability, we note that a decoder contains sr keys, where $s = m/c$. Since there are s copies of $\mathcal{T}$, one of the copies must contain at least r keys. Hence, the c-traceability of $\mathcal{T}'$ follows from the c-traceability of $\mathcal{T}$.

All the other assertions about $\mathcal{T}'$ follow from Method 2. □

5.3 Comments on These Methods

For both of the general methods presented here, keys are allocated to users according to a certain matrix. If we look at the key allocations in schemes constructed under either method, the keys appear to be randomly assigned to users along one dimension, but well structured along the other dimension. The random dimension facilitates traceability because it disperses the users' key sets and the structured dimension contributes to broadcasting capability because it indicates which keys to use to exclude different sets of users. Method 1 adds a dimension of randomness to broadcast encryption to achieve high traceability, while Method 2 adds a dimension of structure to traceability schemes to achieve high broadcasting capability. Hence, the two methods can be viewed as complementary to each other.

We also remark that using the new broadcast encryption schemes in Section 4 in conjunction with Method 1, one can construct broadcast encryption schemes with high traceability, high resiliency, and full scalability.

Acknowledgments

We would like to thank Beverly Schmoock, Yuan Ma, Satomi Okazaki and everyone at RSA Labs in San Mateo, CA for their help in preparing this paper. We would also like to thank the anonymous referees for their useful comments.

References

1. M. Abdalla, Y. Shavitt and A. Wool. *Towards Making Broadcast Encryption Practical.* To appear in the Proceedings of Financial Cryptography '99, Lecture Notes in Computer Science.
2. L. Babai and P. Frankl. *Linear Algebra Methods in Combinatorics with Applications to Geometry and Computer Science.* Preliminary Version 2, September 1992. Available from the authors.
3. S. Berkovits. *How to Broadcast a Secret.* In Advances in Cryptology - Eurocrypt '91, Lecture Notes in Computer Science **547** (1992), pp. 536-541.

4. C. Blundo and A. Cresti. *Space Requirements for Broadcast Encryption.* In Advances in Cryptology - Eurocrypt '94, Lecture Notes in Computer Science **950** (1994), pp. 287-298.
5. C. Blundo, L. A. Frota Mattos and D. Stinson. *Trade-offs Between Communication and Storage in Unconditionally Secure Systems for Broadcast Encryption and Interactive Key Distribution.* In Advances in Cryptology - Crypto '96, Lecture Notes in Computer Science **1109** (1996), pp. 387-400.
6. R. Canetti, T. Malkin and K. Nissim. *Efficient Communication-Storage Tradeoffs for Multicast Encryption.* In Advances in Cryptology - Eurocrypt '99, Lecture Notes in Computer Science.
7. B. Chor, A. Fiat, and M. Naor. *Tracing Traitors.* In Advances in Cryptology - Crypto '94, Lecture Notes in Computer Science **839** (1994), pp. 257-270. Final version with B. Pinkas, preprint.
8. P. Erdös, P. Frankl and Z. Füredi. *Families of Finite Sets in which No Set is Covered by the Union of r Others.* Israel Journal of Mathematics **51** (1985), pp.75-89.
9. A. Fiat and M. Naor. *Broadcast Encryption.* In Advances in Cryptology - Crypto '93, Lecture Notes in Computer Science **773** (1994), pp. 480-491.
10. M. Just, E. Kranakis, D. Krizanc and P. van Oorschot. *On Key Distribution via True Broadcasting.* In Proceedings of 2nd ACM Conference on Computer and Communications Security, November 1994, pp. 81-88.
11. M. Luby and J. Staddon. *Combinatorial Bounds for Broadcast Encryption.* In Advances in Cryptology - Eurocrypt '98, Lecture Notes in Computer Science, **1403**(1998), pp. 512-526.
12. M. Naor and B. Pinkas. *Threshold Traitor Tracing.* In Advances in Cryptology - Crypto '98, Lecture Notes in Computer Science, **1462** (1998), pp. 502-517.
13. J. Staddon. *A Combinatorial Study of Communication, Storage and Traceability in Broadcast Encryption Systems.* Ph.D. thesis, University of California at Berkeley, 1997.
14. D. Stinson. *Cryptography: Theory and Practice.* CRC Press, 1995.
15. D. Stinson. *On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption.* Designs, Codes and Cryptography **12** (1997), pp. 215-243.
16. D. Stinson and T. van Trung. *Some New Results on Key Distribution Patterns and Broadcast Encryption.* Designs, Codes and Cryptography **14** (1998), pp. 261-279.
17. D. Stinson and R. Wei. *Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes.* SIAM J. Discrete Math, **11** (1998), pp. 41-53.
18. D. Stinson and R. Wei. *Key Preassigned Traceability Schemes for Broadcast Encryption.* In the Proceedings of SAC '98, Lecture Notes in Computer Science, **1556** (1999), pp. 144-156.
19. D. Stinson and R. Wei. *An Application of Ramp Schemes to Broadcast Encryption.* Information Processing Letters **69** (1999), pp. 131-135.
20. D. Wallner, E. Harder and R. Agee. *Key Management for Multicast: Issues and Architectures.* Internet Draft, 1997.