Long-Lived Broadcast Encryption

Juan A. Garay[1], Jessica Staddon[2], and Avishai Wool[1]

[1] Bell Labs, 600 Mountain Ave., Murray Hill, NJ 07974, USA.
E-mail: {garay,yash}@research.bell-labs.com.
[2] Bell Labs Research Silicon Valley, 3180 Porter Drive, Palo Alto, CA 94304, USA.
E-mail: staddon@research.bell-labs.com.

Abstract. In a broadcast encryption scheme, digital content is encrypted to ensure that only privileged users can recover the content from the encrypted broadcast. Key material is usually held in a "tamper-resistant," replaceable, smartcard. A coalition of users may attack such a system by breaking their smartcards open, extracting the keys, and building "pirate decoders" based on the decryption keys they extract.

In this paper we suggest the notion of *long-lived broadcast encryption* as a way of adapting broadcast encryption to the presence of pirate decoders and maintaining the security of broadcasts to privileged users while rendering all pirate decoders useless. When a pirate decoder is detected in a long-lived encryption scheme, the keys it contains are viewed as compromised and are no longer used for encrypting content. We provide both empirical and theoretical evidence indicating that there is a long-lived broadcast encryption scheme that achieves a steady state in which only a small fraction of cards need to be replaced in each epoch. That is, for any fraction β, the parameter values may be chosen in such a way to ensure that eventually, at most β of the cards must be replaced in each epoch.

Long-lived broadcast encryption schemes are a more comprehensive solution to piracy than traitor-tracing schemes, because the latter only seek to identify the makers of pirate decoders and don't deal with how to maintain secure broadcasts once keys have been compromised. In addition, long-lived schemes are a more efficient long-term solution than revocation schemes, because their primary goal is to minimize the amount of recarding that must be done in the long term.

1 Introduction

Broadcast encryption (BE) schemes define methods for encrypting content so that only privileged users are able to recover the content from the broadcast. Keys are allocated in such a way that users may be prevented on a short-term basis from recovering the message from the encrypted content. This short-term exclusion of users occurs, for example, when a proper subset of users request to view a movie. The long-term exclusion (or, revocation) of a user is necessary when a user leaves the system entirely.

In practice most BE systems are smartcard-based. It has been well documented (see, for example, [19]) that pirate smartcards (also called pirate "decoders") are commonly built to allow non-paying customers to recover the content. Broadcast encryption schemes can be coupled with traceability schemes to offer some protection against piracy. If a scheme has x-traceability, then it is possible to identify at least one of the smartcards used to construct a given pirate card provided at most x cards are used in total. When a pirate card is discovered, the keys it contains are necessarily compromised and this must be taken into account when encrypting content. Earlier work in traceability does not deal with this; instead, the analysis stops with the tracing of smartcards (or, traitor users).

In this paper, we introduce the notion of *long-lived broadcast encryption* schemes, whose purpose is to adapt to the presence of compromised keys and continue to broadcast securely to privileged sets of users.

Our basic approach is as follows. Initially, every user has a smartcard with several decryption keys on it, and keys are shared by users according to a predefined scheme. When a pirate decoder is discovered, it is analyzed and the keys it contains are identified. Such keys are called "compromised," and are not used henceforth. Similarly, when a user's contract runs out and she is to be excluded, the keys on her smartcard are considered compromised. Over time, we may arrive at a state in which the number of compromised keys on some legitimate user's smartcard rises above the threshold at which secure communication is possible using the broadcast encryption scheme.[1] In order to restore the ability to securely broadcast to such a user, the service provider *replaces* the user's old smartcard with a new one containing a fresh set of keys.

The events driving the service provider's actions are the card compromises: either due to pirate decoders or the expiration of users' contracts. We use these events to divide time into administrative *epochs* of d compromises each. At the end of an epoch, the service provider computes which legitimate users need their cards replaced, and replaces those cards. Therefore, the primary cost in a long-lived BE scheme is the amount of recarding that needs to be done in each epoch.

We assume in this paper that the schemes use perfect encryption, i.e., a pirate can access the content only by obtaining the key. Hence, we are requiring a strong form of security; information theoretic security. However, we argue that this model is very realistic for encrypted pay TV applications. In fact only a handful of the successful attacks described in [19] can be classified as cryptanalytic attacks, and only against analog equivalents of simple substitution ciphers. All the rest exploit breaches in the smartcard, using attacks such as those described in [2]. Furthermore, since smartcard-based systems are widely used in practice [19] it is natural to study this security model.

[1] It costs little to assume that in addition to the shared broadcast keys, each user's smartcard contains a key unique to him. Thus the service provider can always revert to unicast communication to any user if all the user's broadcast keys are compromised.

In this model, and towards the goal of designing an efficient long-lived BE scheme, we consider the performance of three different schemes that have been suggested for broadcast encryption: two randomized and one deterministic. We start with a short-term analysis[2], which focuses on the first epoch. The parameters we analyze are the total number of keys the service provider needs, and the expected number of compromised cards the scheme can tolerate before replacement cards need to be issued. The analysis shows that the costs of the three schemes are quite similar. Hence, we use as the basis for our long-lived construction the simplest of the three, which is a randomized scheme. We provide both empirical and theoretical evidence using an expected-case analysis, that a steady state is achieved in which only a bounded number of users need to be recarded in any epoch of this long-lived scheme.

Related work. Our methods are based on efficient BE schemes. The study of broadcast encryption is initiated in [4, 13, 15] and the efficiency of BE schemes is studied in [5, 6, 18]. The model of BE that we consider here is a formalization of the deterministic (i.e. resilient with probability 1) model of [13] and is consistent with [1, 18].

We are particularly interested in BE schemes that are based on cover-free families (see Section 2.1). Cover-free families are studied in [12] and BE schemes involving such families are studied in [17, 14]. Our long-lived system may be based on (short-term) BE schemes that are tight with the proven lower bounds on the total number of keys in such schemes [12, 14].

The recent papers of [20, 3] propose novel revocation schemes.[3] In these schemes, in order to maintain the ability to revoke t users, the center must make a private communication to each of the remaining users when a single user is revoked. Our goals are fundamentally different from this in that we seek to minimize the amount of communication (e.g. recarding) that's necessary, and so we adapt to the presence of compromised cards (or equivalently, revoked users) by simply removing the keys on these cards from the encryption process. When this approach is no longer possible due to a large number of compromised keys, we recard the affected users only. We show that through an appropriate choice of the parameters, the affected number of users can be a small fraction of the users in the steady state. Hence, our solutions seem useful in either the smartcard scenario or in a network-based system as studied in [20]. Whereas, it may be difficult to apply the techniques in [20] in a smartcard scenario as the cost of reprogramming or replacing a large number of cards, may be prohibitive.

Broadcast encryption schemes and multicast encryption schemes (see for example, [8–10]) are designed with many common goals in mind. BE schemes and multicast schemes are similar in that a pirate smartcard in the former is essentially treated the same as the card of a revoked user in a multicast scheme.

[2] Throughout this paper we focus on an expected case analysis. Some justification for this approach comes from the law of large numbers [21].

[3] The term "revocation" is typically used to indicate the permanent exclusion of a user from the system, rather than the prevention of a user from recovering a particular message.

In either case, care must be taken in future broadcasts to ensure that the keys contained in the card are useless for recovering future messages. The main difference is that in the multicast scenario, the primary goal is to maintain a secure group key at all times, whereas, in broadcast encryption schemes, the group itself varies over time as a subset of the universe of users, and so any group key is only established when the privileged subset is identified. Moreover, widespread rekeying of users (typically via a series of encrypted messages) in the multicast group may be required as part of the process of establishing the new group key. For example, the tree-based scheme in [26] specifies rekeying of all users when a single user leaves the system. This is to be contrasted with the approach taken in long-lived BE, in which we view rekeying of users as the most significant cost of the system, and hence, we rekey as infrequently as possible. Independently of our work, a recent paper [22] confirms that rekeying upon each change in the multicast group membership is prohibitively expensive, and proposes rekeying the entire group at fixed intervals at a cost of some latency in membership adjustments.

Our work is related to the goals of traitor-tracing schemes [11, 24, 7] in that such schemes are also concerned with coalitions of users who conspire to build pirate smartcards (which we refer to as compromised cards). However, we emphasize that there are substantial differences between a traitor-tracing scheme and the schemes we present here. The most important difference is that traceability schemes do not describe how to broadcast securely to privileged sets of users after pirate decoders have been located. The purpose of an x-traceability scheme is to make the practice of building pirate smartcards risky. This is accomplished by allocating keys to users in such a way that once a pirate smartcard is confiscated, at least one of the cards that was used to construct it, can be identified. Clearly, the keys in a pirate smartcard must be viewed as compromised. Hence, after a pirate smartcard is confiscated, the set of keys used to encrypt content must be modified to avoid allowing a user with compromised keys to recover the content. Traitor tracing does not deal with this.

Another difference is that the security achieved in traceability schemes is limited by the necessity of having a bound on the number of users in a coalition. Our approach can handle a pirate decoder built by a coalition of any size. Furthermore, in the secret key model that we consider here, there is a sizable gap between the proven lower bound on the number of keys in a x-traceability scheme and the number of keys in the best known construction. Since our methods rely on efficient BE schemes we are able to keep the total number of keys tight with the aforementioned lower bounds, while retaining an ability to broadcast securely in the presence of pirate smartcards, and consequently, compromised keys.

Finally, we note that if traceability is desired in the long-lived system, this may be achieved by basing the system on a BE scheme with some traceability (as, for example, in [25, 14]).

Our results. The contributions of this paper can be summarized as follows:

- We introduce the notion of *long-lived* broadcast encryption, whose purpose is to continue to broadcast securely to privileged sets of users as cards are compromised over time;
- an analysis and comparison of three BE schemes based on cover-free families with respect to total number of keys and expected number of cards that the scheme can tolerate before recarding; and, based on this analysis,
- an efficient long-lived BE scheme. We provide empirical and theoretical evidence that for any fraction β, there is a scheme that recards at most a β fraction of the users in the steady state.

Organization of the paper. Definitions and notation, as well as the formalization of the long-lived approach to broadcast encryption, are presented in Section 2. The short-term analysis and comparison of the three BE schemes is given in Section 3. The long-lived BE scheme, analysis and experimental evaluation are in Section 4. We conclude with some final remarks and directions for future work in Section 5.

2 Preliminaries

2.1 Definitions and notation

We consider broadcast encryption schemes in the secret key scenario. For our purposes, a broadcast encryption scheme consists of a collection of subsets of keys (one for each user) and a broadcasting protocol, which indicates how to securely distribute content to privileged sets of users. Let $\{u_1, ..., u_n\}$ denote the set of all users, and let the ith user's set of keys be denoted by U_i. Typically, a user's keys are contained in a card, and consequently, we will often refer to U_i as u_i's *card*.

We denote the universe of keys by $\mathcal{K} = \{k_1, ..., k_K\}$ (K keys in total), and each user has r keys in $\mathcal{K}$ ($\forall i, |U_i| = r$). In this paper, the privileged sets of users are of fixed size $n - m$. A privileged set of users will be denoted by P, and the corresponding excluded set of m users will be denoted by X.

The *broadcasting protocol* specifies which subsets of keys in $\mathcal{K}$ suffice to recover the content from the encrypted broadcast. In this paper we are interested in *s-threshold* protocols [17] in which a user needs to use s keys out of r in order to decode the content. When $s = 1$, this broadcasting protocol is sometimes called an *OR* protocol and has been studied in [1, 14, 15, 18].

Definition 1. *An* $(s, |S_P|)$**-threshold protocol** *is used to broadcast a message, M, to users $P = \{u_1, ..., u_{n-m}\}$, in the following manner. K shares of M, M_{k_1}, M_{k_2},..., M_{k_K}, are created in such a way that any s of the shares suffices to recover M. The shares corresponding to keys held by users in $X = \{u_{n-m+1}, ..., u_n\}$ are discarded, and each remaining share is encrypted with its corresponding key and these encrypted messages are broadcast to the universe of users.*

We focus on threshold protocols because they are simple and yield broadcast encryption schemes with maximal *resilience*. A scheme is said to be $m-$*resilient* if m excluded (i.e., not privileged) users cannot recover the content even by pooling their keys. A broadcast encryption scheme with $(s, |S_P|)$-threshold protocols for every privileged set P, is m-resilient. In addition, for some values of s, the techniques of [17] can be used to reduce the broadcast transmission length.

When using threshold protocols for broadcasting we must ensure that a user has sufficiently many keys left after the keys of m other users are excluded to recover the content from the broadcast. Traditionally, this has been guaranteed by allocating keys to users in such a way that the set system is a *cover-free family.*

Definition 2. *Let $\mathcal{K}$ be a collection of elements. A set of subsets of $\mathcal{K}$, $\{U_1,\dots,U_n\}$, is an* **(m,α)-cover-free family**, *if for all $i = 1,...,n$, and for all sets of m indices, $\{j_1,\dots,j_m\}$ not containing i, $|U_i \cap (\cup_{s=1}^{m} U_{j_s})| \leq (1-\alpha)|U_i|$.*

Note that $\frac{1}{r} \leq \alpha \leq 1$. In the original construction of [12] $\alpha = \frac{1}{r}$; i.e., no m users cover *all* of another user's keys.

In most of our work, we adhere to the cover-free requirement to allow comparisons with earlier work. Note that this is a very strong requirement: It guarantees that it is impossible for *any* coalition of m cards to cover an α-fraction of another card's keys. As a result, the constructions need very large key sets, roughly, K is $\Omega(n^{m/r})$, when $r \geq m$, and $\Omega(n)$, otherwise [14]. These bounds may well be prohibitive for large user populations.

However, in a long-lived system, the cover-free requirement seems less relevant, simply because a cover-free scheme gives no guarantee on the system's behavior after $m+1$ cards are compromised. In addition, in the randomized attack model that we are considering, it can easily be shown (see Lemma 8) that even in a system with significantly fewer keys than an m-cover-free system, a set of m compromised cards will cover another card only with negligible probability. Thus, in the long-term analysis, and in the experimental results, we do not define values of m and adhere to the cover-free requirement for those values. Instead we de-couple the number of users n from the total number of keys K number, and observe the behavior of the resulting schemes in terms of how many cards need to be issued per epoch.

We are interested in how the broadcast encryption scheme is affected by pirate smartcards, which we assume to be cards containing r keys.[4] A card may be *compromised* either because of piracy or simply because a user ceases to be an active subscriber and leaves the system. In either case, the keys on the card become permanently unavailable for use as encryption keys. A compromised card may be a clone of some user's card or may contain a set of r keys that does not exactly match any of the n users in the system.

When the keys on a card are all unavailable because it is a compromised card or belongs to an excluded user, we say that the keys it contains and the card itself are *dead.* A key that is not on a compromised card and does not belong

[4] We believe it is reasonable to assume that a pirate decoder contains at least as many keys as legitimate cards. It is sometimes the case that the pirate cards even use *better* technology than the legitimate ones [16], i.e., they can store more keys.

to an excluded user is said to be *active*. A card is said to be *clean* if it contains only active keys. We use d as a counter for the number of dead cards (i.e., either due to piracy or exclusion).

We note that the reason behind the unavailability of a key has an effect on our behavior. When a key is dead because it appears on a compromised card, the key is permanently unavailable; whereas if it simply appears on an excluded user's card, its unavailability may be short-term as the excluded user may be a privileged user at a later time. For more on this issue see Section 2.2.

– $\{u_1, \ldots, u_n\}$ is the set of all users. – U_i is the set of keys held by u_i. – $\mathcal{K} = \{k_1, ..., k_K\}$ is the set of all keys. – S_P is the set of keys used to broadcast to privileged set P. – 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. – d is the number of unavailable (dead) cards at a certain point in time. – C_j^i is the set of cards in epoch i that were created in epoch j.

Table 1. Summary of notation

2.2 The long-lived approach

In this section we describe our basic method for securely broadcasting to privileged users as cards get compromised. The method is based on knowledge of compromised cards and consists of two basic components:

1. Adjusting the set S_P of keys that are used to encrypt the broadcast; and
2. recarding of users.

The method is *reactive* in the sense that actions are taken responding to the number of compromised cards (which, for example, might hamper the continuity of service for privileged users, or bring transmission costs to unacceptable levels). This divides the life of the system into *epochs*. At the end of epoch i, $i = 1, 2, \cdots$, a decision is made about which cards need to be replaced, and new cards are issued. We now describe the structure of our long-lived reactive recarding scheme more formally.

A long-lived broadcast encryption scheme consists of:

- Underlying structure: An efficient (short-term) broadcast encryption scheme consisting of an (m,α)-cover-free family and a unicast key between each user and the center. An $(\alpha r,\ |S_P|)$-threshold protocol is used to broadcast to privileged set P. If some user is unreachable under the $(\alpha r,\ |S_P|)$-threshold protocol (i.e., too many of their keys appear on dead cards) then the unicast key will be used to reach that user.

- A distribution on the compromised cards: In this paper we assume the cards are drawn independently at random (with replacement) from the key space.
- Reaction to a newly compromised card: To render the compromised card useless as a decoder we exclude all its keys from S_P, creating a new set S_P^1. Broadcasting to privileged set P is with an (αr, $|S_P^1|$)-threshold protocol, relying on unicast keys if necessary.
- Recarding policy: A recarding session is entered whenever d cards become unavailable. During a recarding session, any user with less than αr active keys receives a new card.

The parameter d in the fixed schedule will be based on the number of compromised users and the desired transmission length. In a recarding session, new values are chosen randomly for all dead keys.

Recall that a key is dead either because it belongs to an excluded user or is on a compromised card. In the former case, the key is unavailable on what may be a short-term basis, as an excluded user may well be a privileged user at another time. Hence, we note that our long-term analysis (Section 4) is best applied to a stable privileged set P, or to the whole set of users when the number of excluded users, m, is small. Given this, it is very likely that users will only be recarded when more than $(1-\alpha)r$ of their keys are *permanently* unavailable (i.e., contained in compromised cards) rather than simply temporarily unavailable, due to the current set of excluded users. As stated in Section 1, the primary motivation for recarding users should be the presence of compromised cards.

A summary of terms and notation is given in Table 1.

3 A Short-Term Analysis of Three BE Schemes

In this section we describe three schemes, each based on a cover-free family. The first BE scheme is a randomized bucket-based construction from [17] and the second is a deterministic construction based on polynomials [14, 17]. Both constructions yield (m, α)-cover-free families. The third scheme is a very simple randomized method for producing (m, α)-cover-free families. We present a short-term analysis of all three schemes, which indicates that they are remarkably similar in terms of efficiency. Specifically, the three schemes only differ by a constant fraction in the number of dead cards (i.e., compromised or belonging to excluded users) they can tolerate before recarding is needed. Hence, given the simplicity of the randomized scheme, we choose to focus our long term analysis on it (see Section 4).

3.1 A bucket-based BE scheme

In this section we consider a reactive recarding scheme based on the randomized cover-free family construction of [17].[5] In [17], the construction is presented for an $(m, 1/2)$-cover-free family. We present the construction of an (m, α)-cover-free family for completeness. The set of all keys $\{k_1, ..., k_K\}$ is partitioned into sets of size K/r. Each card contains a randomly selected key from each set. To

[5] See the randomized construction of an "inner code" in Section 4 of [17].

broadcast to a set P of privileged users, we use an $(\alpha r, |S_P|)$-threshold protocol as described in Section 2.1.

The following lemma gives a lower bound on the total number of keys K, for which this construction yields a (m, α)-cover-free family with high probability. Note that in this scheme m depends on r and K.

Lemma 1. *Let ϵ be a positive fraction. If $r = \frac{K \ln(1/\alpha)}{4m}$ and K is $\Omega(m(m \ln n + \ln(1/\epsilon)))$, then the bucket-based construction is an (m, α)-cover-free family with probability $1 - \epsilon$.*

Proof: Consider $m + 1$ users, $u, u_1, ..., u_m$. We'll calculate the probability that $u_1, .., u_m$ cover enough of user u's keys to violate the cover-free condition. The partitions are each of size $K/r = \frac{4m}{\ln(1/\alpha)}$. The probability that a key k is not in $\cup_{i=1}^m U_i$ is $(1 - \frac{\ln(1/\alpha)}{4m})^m$. Therefore,

$$E(|U \cap (\cup_{i=1}^m U_i)|) = r(1 - (1 - \frac{\ln(1/\alpha)}{4m})^m)$$

When m is sufficiently large relative to $1/\alpha$, $(1 - \frac{\ln(1/\alpha)}{4m})^{\frac{4m}{\ln(1/\alpha)}} \geq \frac{1}{e^2}$. Hence, the above expected value is at most $r(1 - \sqrt{\alpha})$. Hence,

$$Pr[|U \cap (\cup_{i=1}^m U_i)| > (1-\alpha)r] \leq Pr[|U \cap (\cup_{i=1}^m U_i)| > (1 + \sqrt{\alpha})\mu \,|\, \mu = r(1 - \sqrt{\alpha})]$$

Using Chernoff bounds, that probability is at most, $e^{\frac{-\alpha r(1-\sqrt{\alpha})}{3}}$. When $K \geq \frac{4(m+1)m \ln n + 4m \ln(1/\epsilon)}{\alpha(1-\sqrt{\alpha}) \ln(1/\alpha)}$, it follows that, $\binom{n}{m+1} e^{\frac{-\alpha r(1-\sqrt{\alpha})}{3}} < \epsilon$. □

Now we consider the short-term behavior of this scheme. In particular, we are interested in how many compromised cards this scheme can tolerate before recarding is necessary. Since we are interested in an expected-case analysis, we calculate how many dead cards, chosen randomly with replacement, cause a user to need to be recarded. The lower bound proven in Lemma 2 is very close to the bounds proven for the other two reactive recarding schemes (see Lemma 3 and Lemma 5), however, we note that since r is likely to be quite large in this scheme, the tolerable number of dead cards may be fairly small.

Lemma 2. *Consider a user, u. In the bucket-based construction, the expected number of dead cards that can be tolerated before it is necessary to recard u is greater than $\ln(1/\alpha)(\frac{K}{r} - 1)$.*

Proof: First we show that it suffices to only consider dead cards that are clones (i.e., cards that contain exactly one key from each bucket). To see this, note that the probability a key is in a randomly chosen set of r keys (i.e., a cloned card or otherwise) is $\frac{r}{K}$. From Lemma 1, we know that the probability a randomly chosen key is on a cloned card is $\frac{\ln(1/\alpha)}{4m} = \frac{r}{K}$. Hence in our expected case analysis it suffices to assume the dead cards are clones.

Consider d dead cards, U_1,...,U_d. As calculated in the previous lemma, for a random user u,

$$E(|U \cap (\cup_{i=1}^{d} U_i|) = r(1 - (1 - \frac{\ln(1/\alpha)}{4m})^d)$$

This quantity is greater than $(1-\alpha)r$ when $\alpha > (1 - \frac{\ln(1/\alpha)}{4m})^d$. Solving for d, we get,

$$d > \frac{\ln(1/\alpha)}{\ln(\frac{4m}{4m - \ln(1/\alpha)})} = \ln(1/\alpha)(\frac{K}{r} - 1)$$

Since $\ln(\frac{4m}{4m-\ln(1/\alpha)}) \leq \frac{\ln(1/\alpha)}{4m-\ln(1/\alpha)}$, it follows that $d > 4m - \ln(1/\alpha)$. □

3.2 A deterministic BE scheme

In this section we consider the polynomial-based broadcast encryption scheme of [14]. This scheme differs from the polynomial-based scheme in [17] in that r is an independent parameter, and not a function of the other variables. This scheme uses polynomials to construct a deterministic (m, α)-cover-free family. An $(\alpha r, S_P)$-threshold protocol is used to broadcast to a privileged set, P.

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 $\frac{r(1-\alpha)}{m}$. (For simplicity, we assume that $m|r(1-\alpha)$.) There are $p^{\frac{r(1-\alpha)}{m}+1}$ such polynomials. We associate each of the n users with a different polynomial. Therefore, p needs to satisfy the condition that $p^{\frac{r(1-\alpha)}{m}+1} \geq n$. For each pair (x, y), where $x \in A$ and $y \in F_p$, we create a unique key $k_{(x,y)}$. Hence, the total number of keys, K, is $rp \geq rn^{\frac{m}{r(1-\alpha)+m}}$. If a user u is associated with a given polynomial f, u's smartcard contains the keys in the set $\{k_{(x,f(x))} | x \in A\}$. Since any two of the polynomials intersect in at most $\frac{r(1-\alpha)}{m}$ points, it follows that any two users share at most $\frac{r(1-\alpha)}{m}$ keys. This ensures that if all the keys belonging to the m excluded users are removed, each privileged user will still have at least αr keys. Hence, the center can broadcast to users a privileged set P, with an $(\alpha r, S_P)$-threshold protocol. A user needs to be recarded when the number of active keys on their card falls below αr.

Lemma 3. *Consider a user, u. For K sufficiently large, the expected number of dead cards that can be tolerated in the deterministic scheme before it is necessary to recard u is at least $\frac{K}{r}\left(\frac{\ln(1/\alpha)}{2^{r(1-\alpha)+2}}\right)$.*

The proof of this lemma is very similar to the proof of Lemma 2.

3.3 A simple randomized BE scheme

In this scheme, each user is allocated a randomly selected set of r keys out of a universe of K keys total, where K is chosen large enough to ensure that we have an (m, α)-cover-free family with high probability (see Lemma 4). Broadcasting to a set P of privileged users is accomplished as discussed in the previous two sections. Hence, when a user has less than αr active keys, the user's card needs to be replaced. In the recarding procedure, new keys are generated for all dead keys, and active keys are unchanged. The total number of active keys (i.e., keys that need to be stored by the broadcasting center) is unaffected by the recarding procedure. We first prove a lower bound on the total number of keys.

Lemma 4. *Given any positive fraction ϵ, if the total number of keys, K is $\Omega(\frac{n^{m+1}}{\epsilon}^{\frac{1}{r(1-\alpha)+1}})$ then the randomized reactive recarding scheme is an (m,α)-cover-free family with probability at least $1-\epsilon$.*

Proof: Consider $m+1$ users, u, u_1, u_2,...,u_m. First we bound the probability that u_1,...,u_m cover more than $(1-\alpha)r$ of u's keys. Since $|\cup_{i=1}^m U_i| \leq mr$, we have the following bound:
$Pr(|U \cap (\cup_{i=1}^m U_i)| > (1-\alpha)r) \leq \frac{\binom{mr}{(1-\alpha)r+1}\binom{K-(1-\alpha)r-1}{\alpha r-1}+\binom{mr}{(1-\alpha)r+2}\binom{K-(1-\alpha)r-2}{\alpha r-2}+\ldots+\binom{mr}{r}}{\binom{K}{r}}$

Using binomial bounds and simplifying, we have:
$Pr(|U \cap (\cup_{i=1}^m U_i)| > (1-\alpha)r) \leq \frac{(rem)^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}}$

Hence, the probability that $|U \cap (\cup_{i=1}^m U_i)| \leq \alpha r$ is at least $1-\frac{(rem)^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}}$. There are $n-m$ privileged users, therefore the probability that there is at least one privileged user who shares more than $r(1-\alpha)$ keys with $u_1,...,u_m$, is at most $1-[1-\frac{rem^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}}]^{n-m}$. To account for all possible excluded sets of m users, it suffices to multiply by $\binom{n}{m}$:

$$\binom{n}{m}(1-[1-\frac{(rem)^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}}]^{n-m})$$

Substituting a binomial approximation,

$$1-(\frac{m}{ne})^m\epsilon \leq (1-\frac{(rem)^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}})^{n-m}$$

If $K > r^{r+1}m^r e^{2r-1}$ (this is reasonable since we expect r to be small), then we can use the fact that $(1-x)^{n-m} \geq 1-(n-m)x$ when $x \leq 1$, to simplify this expression. With this substitution, it suffices to show that $(n-m)\frac{(rem)^r e^{\alpha r-1}\alpha r}{K^{r(1-\alpha)+1}} \leq (\frac{m}{ne})^m\epsilon$. Solving for K yields the statement of the lemma. □

Remark 1. Note that the factor of $n^{m+1/(r(1-\alpha)+1)}$ in the bound on K is due to the cover-free requirement, that with very high probability it is *impossible* for m cards to cover another. However, the construction itself remains viable for any value of K. In fact, in the randomized attack model, in which the pirates pry open randomly selected cards, much smaller values of K suffice to guarantee a low probability of m cards covering another (see Lemma 8).

Lemma 5. *Consider a user u. The expected number of dead cards that can be tolerated in the randomized recarding scheme before it is necessary to recard u is at least $(\frac{K-r}{r})\ln(1/\alpha)$.*

The proof of this lemma is very similar to the proof of Lemma 2.

3.4 Comparison of the BE schemes

As the previous lemmas show, the three BE schemes have very similar costs. Each yield (m,α)-cover-free families with high probability when the total number of keys is close to the optimal bound[6] of $n^{m/r}$. In the deterministic and randomized schemes, this is clear. To see that the bucket-based scheme is close to this bound, note that the proven bound is roughly $\Omega(m^2 \ln n)$ and $K \geq n^{m/r}$ is equivalent to $K \ln K \geq \frac{4m^2 \ln n}{\ln(1/\alpha)}$, when r has the value stated in the lemma.

In addition to the above similarity, all three schemes can tolerate approximately $\frac{K}{r}$ dead cards before recarding a particular user is necessary. We note that this means we expect to need to recard a user only after $\frac{K}{r} - m$ cards are compromised (due to piracy or contract expiration). As the schemes are so close in terms of efficiency and cost, we use the third scheme as the basis of our long-lived system. It is the most simple, as it is entirely random, and it has the advantage over the bucket-based scheme that r and m are independent parameters.

4 Long-Lived Broadcast Encryption

In this section we extend the randomized BE scheme from Section 3.3 to a long-lived scheme. The extension is reactive as defined in Section 2.2—recarding is performed once every d dead cards—and for simplicity we consider the $(m, 1/r)$-cover-free family version of the scheme (OR protocols). We emphasize that this analysis is best applied to a stable set of privileged users, or to the entire set of users when m is small. In either case, we expect to only have to recard a user when too many of their keys appear on compromised (i.e., permanently unavailable) cards.

The main cost associated with long-lived schemes is the number of cards that must be replaced. We present a scheme in which given a positive fraction β, the parameters may be chosen so that eventually at most βn of the cards need to be replaced during any recarding session. This property is demonstrated both empirically and theoretically.

We now turn to the description of the long-lived extension and its analysis. Assume that d cards are compromised. The process for generating the new cards is as follows. Let Z be the set of keys these cards contain and let $z = |Z|$; note that $z \leq dr$. The scheme

1. discards all the keys in Z, and
2. generates a set Z' of new keys, $|Z'| = z$. The new set of all keys becomes $\mathcal{K}' = (\mathcal{K} \backslash Z) \cup Z'$.

The resulting number of keys is again K in total. Every user that needs to be recarded receives the fresh values of the same keys.[7]

[6] This bound is for $\alpha = \frac{1}{r}$.

[7] This is to preserve the cover-free property of the scheme used as basis. In the randomized scheme, the same will hold if for every user that needs to be recarded, r keys are again picked at random from the updated set $\mathcal{K}'$.

As keys become compromised and users are recarded, the users can be partitioned into sets of users with cards with fresh keys, and users with cards containing keys some of which are dead. This process is depicted in Figure 1. We let C_j^i denote the set of cards in epoch i that were created in epoch j. Initially (epoch 1), $C_1^1 = \{U_1, ..., U_n\}$. Selecting (randomly) d dead cards from C_1^1 yields C_2^2, the set of users that need to be recarded, as well as $C_1^2 = C_1^1 \backslash C_2^2$; in epoch 2, selecting d random cards from C_1^2 and C_2^2 yields C_3^3 as well as C_1^3 and C_2^3; and so on.

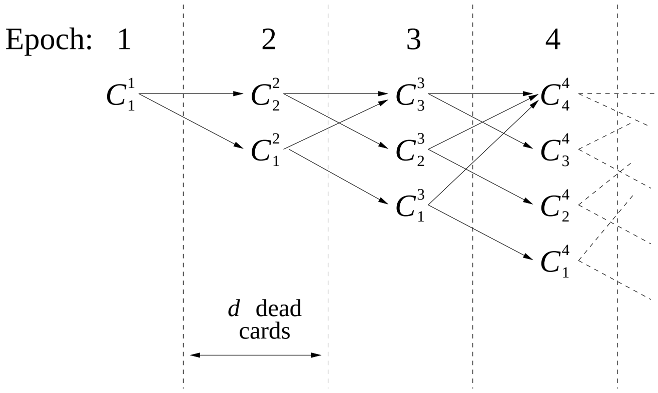

Fig. 1. Randomized long-lived BE scheme. d dead cards determine the epochs; C_j^i is the set of cards in epoch i that were created in epoch j.

Towards bounding the necessary number of recards per epoch, namely, the (expected) size of set C_j^j in epoch j, we first prove recurrence relations relating the expected number of cards in epoch j that were created in epoch $i \leq j$, $E(|C_i^j|)$.

Lemma 6. *In the randomized long-lived BE scheme with $\alpha = 1/r$ and a fixed recarding schedule of once every d dead cards, the following hold for all $i \geq 1$:*

1. $E(|C_{i+1}^{i+1}|) \leq \sum_{j=1}^{i} E(|C_j^i|)[1 - (1 - \frac{r}{K})^{(i+1-j)d}]^r$;
2. $\forall\ j, 1 \leq j \leq i$, $E(|C_j^{i+1}|) = E(|C_j^i|)(1 - [1 - (1 - \frac{r}{K})^{(i+1-j)d}]^r)$.

Proof: To see the first inequality, note that if a user is recarded (or created) in epoch j, then during the time interval from the beginning of epoch j to the end of epoch i, $d(i + 1 - j)$ randomly chosen cards become unavailable. If these cards cover the user's card, then the user must be recarded. Due to the random nature of the scheme, a user is covered with probability, $[1 - (1 - \frac{r}{K})^{(i+1-j)d}]^r$. We have a weak inequality rather than equality, because a user may be covered by fewer than $d(i - j + 1)$ cards.

The second equation is obtained by noting that all users who were recarded (or created) in epoch j, and who are not covered by the end of epoch i, become the set of users C_j^{i+1}. $\square$

We now use the first part of Lemma 6 to demonstrate that an upper bound on the number of recards per epoch holds in the limit, and that this upper bound can be made small through appropriate choices of K, r and d. We emphasize that this is an approximate analysis, and is provided largely to give some intuition for the experimental results in Section 4.1. A more rigorous analysis will appear in the full-length version of this paper.

The analysis contains three components. First, given fixed values of the parameters, we show that there exists an integer ℓ_1, such that the probability that a card is covered (and hence, needs to be refreshed) within ℓ_1 epochs, is negligible. The intuition for this result is that if a card has been refreshed recently, then it is unlikely that it will be covered again within a small number of epochs. This result indicates that the contribution to $E(|C_{i+1}^{i+1}|)$ from the first ℓ_1 terms of inequality 1 in Lemma 6, is fairly small. In addition, the later terms in inequality 1 in Lemma 6 may also not contribute much to the upper bound on $E(|C_{i+1}^{i+1}|)$. In particular, there exists an integer ℓ_2 (greater than ℓ_1), such that it is unlikely that a card will *not* be covered within ℓ_2 epochs. Note that this implies that when $i-j \geq \ell_2$, $E(|C_j^i|)$ is fairly small, and hence, won't contribute much to the upper bounds on $E(|C_{i+1}^{i+1}|)$. Finally, we show that both ℓ_1 and ℓ_2 are on the order of K/rd, hence the dominating terms are those for which $i-j$ is $\Theta(\frac{K}{rd})$, and this leads to an approximation for the upper bound of the steady state recard rate, β. The following lemma makes these ideas more precise.

Lemma 7. *Assume n, K, r, d and $\epsilon > 0$ are given. The following are true:*

1. *If $\ell_1 \in O(\frac{\epsilon^{1/r}K}{rd})$, then the probability that a card is covered within ℓ_1 epochs is less than ϵ.*
2. *If $\ell_2 \in \Omega(\frac{(1-\epsilon)^{1/r}K}{rd})$, then the probability that a card survives for more than ℓ_2 epochs before it is covered, is less than ϵ.*
3. *If $i-j \in \Theta(\frac{K}{rd})$ then the coefficient of $E(|C_j^i|)$ in inequality 1 of Lemma 6, is approximately, $(1-(\frac{1-\frac{r}{K}}{e^c})^d)^r$, where c is a constant.*

Proof:

1. The probability that a card is covered by $d\ell_1$ randomly chosen cards is $(1-(1-\frac{r}{K})^{d\ell_1})^r$. Setting this quantity less than ϵ and solving for ℓ_1 yields, $\ell_1 < \frac{\ln(1-\epsilon^{1/r})}{d\ln(1-r/K)}$.
2. The probability that a card is not covered within ℓ_2 epochs is, $1-(1-(1-\frac{r}{K})^{d\ell_2})^r$. Setting this less than ϵ and solving for ℓ_2 yields, $\ell_2 > (\frac{K}{rd}(1-\epsilon)^{1/r})$.
3. Assuming that $i-j=\frac{cK}{rd}$, for some constant c, we'll bound the contribution of $E(|C_j^i|)$ to the inequality in Lemma 6 (i.e., we'll bound the coefficient of $E(|C_j^i|)$ — given the earlier results, this bound is an approximate upper bound to β, the long-term steady state).
   When $i-j=\frac{cK}{rd}$, the coefficient of $E(|C_j^i|)))$, is $(1-((1-r/K)^d)^{cK/rd+1})^r$.
   When K is sufficiently large, this is of the order of $(1-\frac{(1-r/K)^d}{e^c})^r$. $\square$

When combined with parts 1 and 2, part 3 of the lemma indicates that the steady state recard rate β should decrease with K, which agrees with the experimental results that follow. The quantity also increases with d, which agrees with the basic intuition that the longer we wait to recard, the more recarding we will have to do.

4.1 Numerical experimentation

In order to get a better understanding of the card replacement dynamics, we present some numerical experiments. In these experiments we evaluate Equation 1 of Lemma 6 (assuming an equality rather than an inequality) for a variety of parameter settings, and track the number of cards that were issued in every epoch.

We focus on the random attack model, and assume that, in each epoch, the dead cards are selected uniformly at random from the set of user cards. We do not require that the system be cover-free, so K is not constrained by the bound of Lemma 4. Instead, we let K be a free parameter which we vary.

To justify this decoupling of K from n, we present the following simple lemma that provides a lower bound on K such that with high probability, none of the n user cards are covered by d randomly chosen compromised cards (i.e., some cover-freeness is achieved with high probability). As mentioned in Section 2.1, this lower bound may be much smaller than the size of K in a d-cover-free family.

Lemma 8. *Let $\epsilon > 0$, and n, r and d be given. If $K > \frac{r}{c}$, where c is a constant that depends on ϵ, n, r, and d, then the probability that any user's card is covered by d randomly chosen cards is less than ϵ.*

Proof: The probability that n (randomly chosen) user's cards aren't covered by d randomly chosen cards is $[1-(1-(1-\frac{r}{K})^d)^r]^n$. Hence, we solve the following inequality for K, $[1-(1-(1-\frac{r}{K})^d)^r]^n > 1-\epsilon$, which yields, $K > \frac{r}{1-(1-[1-(1-\epsilon)^{1/n}]^{1/r})^{1/d}}$. $\square$

As an example, for the values of n, r and d used in Figure 2 and $\epsilon = .1$, this lemma gives a lower bound on K of approximately 69, far less than the lower bound of approximately 10^{10} for a d-cover-free family.

In all of our experiments we use a user population of size $n = 100,000$, which we view as being on the low end of real population sizes. The card capacity r ranges between 10 and 50, which is realistic for current smartcards with 8KB of memory and keys requiring, say, 64 bytes each including overhead. We let the epoch length be $10 \leq d \leq 50$ dead cards. We use $1000 \leq K \leq 5000$, hence K may be much smaller than the number of keys required by Lemma 4, which calls for $K \geq n^{m/r}$ keys (note the dependency on m, the number of users the underlying BE is able to exclude).

Figure 2 shows the dynamics of the card re-issue strategy, and the effect of the total number of keys K. We see that the curves begin with oscillations. In the first epochs no cards are re-issued since the first dead cards do not cover

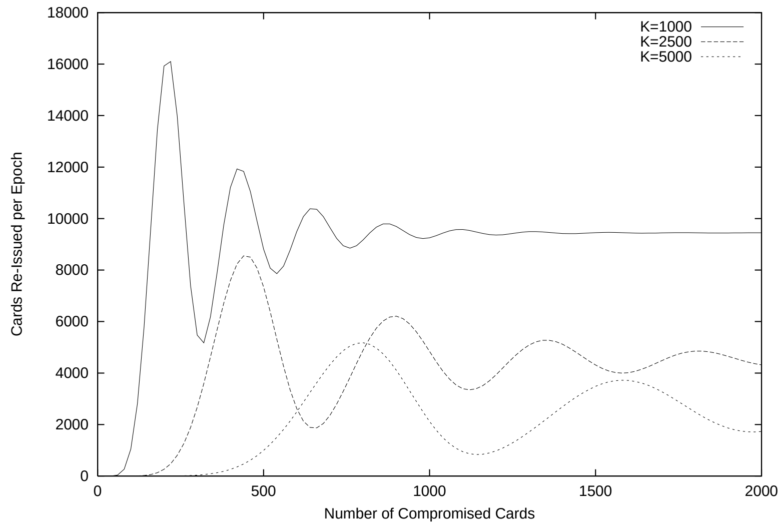

Fig. 2. Number of cards re-issued per epoch, with $n = 100,000$, $r = 10$, $d = 20$, for different values of the total number of keys K.

any user. But after a certain number d_c of dead cards are discovered, enough keys are compromised and there is a rapid increase in re-issued cards. This in turn "cleans" the card population, and the re-issue rate drops. We see that the oscillations are dampened and a steady state appears fairly quickly.

The parameter K affects several aspects of the dynamics: the first card re-issue point d_c is later for larger K ($d_c \approx 40$ for $K = 1000$ but $d_c \approx 200$ for $K = 5000$); the oscillations are gentler, have a smaller amplitude, and lower peak rate, for larger K; and most importantly, the steady state rate of re-issue is lower for larger K (≈ 9400 cards per epoch for $K = 1000$ but ≈ 2000 cards per epoch for $K = 5000$). Overall, we see that increasing K improves all the aspects of the re-issue strategy. Thus we conclude that it is better to use the largest possible K that is within the technological requirements.

Figure 3 shows the effect of increasing the card capacity r. The diagram indicates that larger values of r cause greater re-issue costs: larger r's have a higher steady state re-issue rate, and higher peak re-issue rates. This agrees with the fact that as r increases, we expect each key to be contained in more cards, so the effect of a compromised key is more widespread. Also, as indicated by Lemma 5, we expect to have to recard users sooner when r is large (and K is fixed). However, with a smaller r the expected transmission length is longer; at the extreme, setting $r = 1$ gives optimal re-issue rates (no cards need to be re-issued), with very long transmissions.

Figure 4 shows the effect of increasing the epoch length d. From the figure, it is clear that a longer epoch results in a smaller total number of re-issued cards.

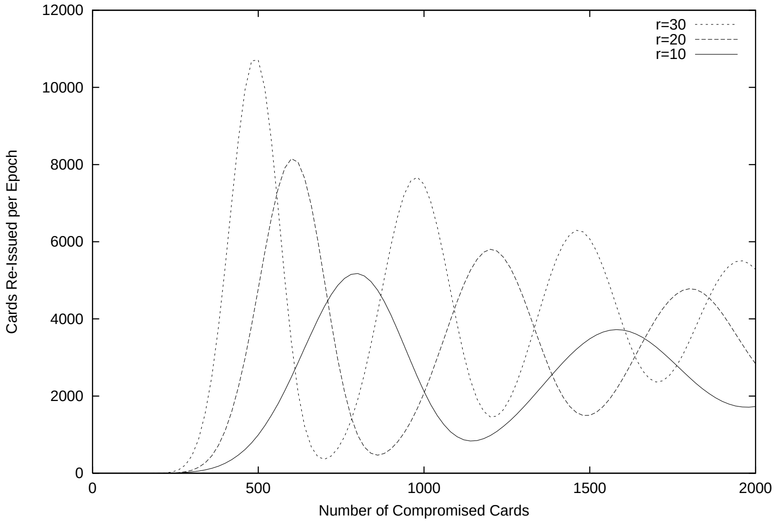

Fig. 3. Number of cards re-issued per epoch, with $n = 100,000$, $K = 5000$, $d = 20$, for different values of the number of keys per card, r.

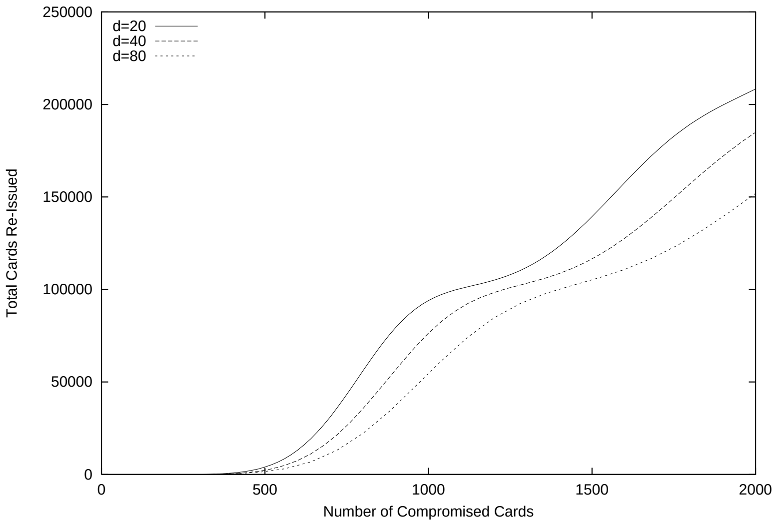

Fig. 4. The accumulated total number of cards re-issued, with $n = 100,000$, $K = 5000$, $r = 10$, for different values of the epoch length d.

However, a long epoch also means that many keys are compromised during the epoch, and consequently, it may be impossible to broadcast securely to some users during the epoch without unicasts. Hence, recarding costs and transmission costs may influence the choice of d.

5 Summary and Directions for Future Work

In this paper, we consider making broadcast encryption schemes resistant to piracy by introducing a policy of permanently revoking compromised keys. This is to be distinguished from the short-term revocation of keys that is typically done in a BE scheme in order to prevent users from recovering a particular message (e.g., a movie) and is instead more analogous to the revocation of users in a multicast group.

There are many open questions with respect to the analysis of the simple model we've proposed. For example, it would be interesting to look at different distributions on the compromised cards (i.e., other than independently at random) and to determine how transmission length is affected by parameters such as d.

It would also be interesting to consider modifications to the overall approach taken here. The long-lived scheme presented in Section 4 is reactive in the sense that actions are taken responding to the number of compromised cards. Are methods that do not count on pirate smartcard intelligence—oblivious, "proactive"—viable?

Acknowledgements. The authors thank Adi Shamir and the anonymous referees for their many valuable comments.

References

1. M. Abdalla, Y. Shavitt, and A. Wool. Towards making broadcast encryption practical. In M. Franklin, editor, *Proc. Financial Cryptography'99*, Lecture Notes in Computer Science **1648** (1999), pp. 140–157. To appear in IEEE/ACM Trans. on Networking.
2. R. Anderson and M. Kuhn. Low cost attacks on tamper resistant devices. In *5th Security Protocols Workshop*, Lecture Notes in Computer Science **1361** (1997), pp. 125–136.
3. J. Anzai, N. Matsuzaki and T. Matsumoto. *A Quick Group Key Distribution Scheme with "Entity Revocation"* In Advances in Cryptology - Asiacrypt '99, Lecture Notes in Computer Science (1999), pp. 333-347.
4. S. Berkovits. *How to Broadcast a Secret.* In Advances in Cryptology - Eurocrypt '91, Lecture Notes in Computer Science **547** (1992), pp. 536-541.
5. 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.
6. 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.

7. 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), pp. 338-353.
8. R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor and B. Pinkas. *Multicast Security: A Taxonomy and Efficient Constructions.* In Proc. INFOCOM 1999, Vol. 2, pp. 708-716, New York, NY, March 1999..
9. 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.
10. R. Canetti and B. Pinkas. *A Taxonomy of Multicast Security Issues.* Internet draft. Available at: `ftp://ftp.ietf.org/internet-drafts/draft-canetti-secure-multicast-taxonomy-00.txt`
11. B. Chor, A. Fiat, M. Naor and B. Pinkas. *Tracing Traitors.* Full version to appear in IEEE Transactions on Information Theory. Preliminary version in Advances in Cryptology - Crypto '94, Lecture Notes in Computer Science **839** (1994), pp. 257-270.
12. 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.
13. A. Fiat and M. Naor. *Broadcast Encryption.* In Advances in Cryptology - Crypto '93, Lecture Notes in Computer Science **773** (1994), pp. 480-491.
14. E. Gafni, J. Staddon and Y. Yin. *Efficient Methods for Integrating Braodcast Encryption and Traceability.* In Advances in Cryptology - Crypto '99, Lecture Notes in Computer Science **1666** (1999), pp. 372-387.
15. 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.
16. M. Kuhn. Personal communication, 1999.
17. 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), pp. 609-623.
18. 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.
19. J. McCormac. *European Scrambling Systems 5.* Waterford University Press, 1996.
20. M. Naor and B. Pinkas. *Efficient Trace and Revoke Schemes.* In *Proc. Financial Cryptography 2000*, Anguila, February 2000.
21. J. Pitman. *Probability.* Springer-Verlag, 1993.
22. S. Setia, S. Koussih, S. Jajodia and E. Harder. *Kronos: A Scalable Group Re-Keying Approach for Secure Multicast.* In 2000 IEEE Symposium on Security and Privacy, pp. 215-228.
23. D. Stinson. *Cryptography: Theory and Practice.* CRC Press, 1995.
24. D. Stinson and R. Wei. *Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes.* SIAM J. Discrete Math, **11** (1998), pp. 41-53.
25. D. Stinson and R. Wei. *Key Preassigned Traceability Schemes for Broadcast Encryption.* In Proc. SAC '98, Lecture Notes in Computer Science **1556** (1999), pp. 144-156.
26. D. Wallner, E. Harder and R. Agee. *Key Management for Multicast: Issues and Architectures.* Internet Request for Comments, **2627** (June 1999). Available at: `ftp.ietf.org/rfc/rfc2627.txt`.