Broadcast Encryption *

Amos Fiat[†] Moni Naor[‡]

Abstract

We introduce new theoretical measures for the qualitative and quantitative assessment of encryption schemes designed for broadcast transmissions. The goal is to allow a central broadcast site to broadcast secure transmissions to an arbitrary set of recipients while minimizing key management related transmissions. We present several schemes that allow a center to broadcast a secret to any subset of privileged users out of a universe of size n so that coalitions of k users not in the privileged set cannot learn the secret. The most interesting scheme requires every user to store $O(k \log k \log n)$ keys and the center to broadcast $O(k^2 \log^2 k \log n)$ messages regardless of the size of the privileged set. This scheme is resilient to *any* coalition of k users. We also present a scheme that is resilient with probability p against a random subset of k users. This scheme requires every user to store $O(\log k \log(1/p))$ keys and the center to broadcast $O(k \log^2 k \log(1/p))$ messages.

*Preliminary version appeared in Advances in Cryptology - CRYPTO '93 Proceedings, Lecture Notes in Computer Science, Vol. 773, 1994, pp. 480–491.

†Dept. of Computer Science, School of Mathematics, Tel Aviv University, Tel Aviv, Israel, and Algorithmic Research Ltd. E-mail fiat@math.tau.ac.il.

‡Incumbent of the Morris and Rose Goldman Career Development Chair, Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. Research supported by an Alon Fellowship and a grant from the Israel Science Foundation administered by the Israeli Academy of Sciences. E-mail: naor@wisdom.weizmann.ac.il.

1 Introduction

We deal with broadcast encryption. We consider a scenario where there is a center and a set of users. The center provides the users with prearranged keys when they join the system. At some point the center wishes to broadcast a message (e.g. a key to decipher a video clip) to a *dynamically* changing privileged subset of the users in such a way that non-members of the privileged class cannot learn the message. Naturally, the non-members are curious about the contents of the message that is being broadcast, and may try to learn it.

The obvious solution is: give every user its own key and transmit an individually encrypted message to every member of the privileged class. This requires a very long transmission (the number of members in the class times the length of the message). Another simple solution is to provide every possible subset of users with a key, i.e. give every user the keys corresponding to the subsets it belongs to. This requires every user to store a huge number of keys.

The goal of this paper is to provide solutions which are efficient in both measures, i.e. transmission length and storage at the user's end. We also aim that the schemes should be computationally efficient.

To achieve our goal we add a new parameter to the problem. This parameter represents the number of users that have to collude so as to break the scheme. The scheme is considered broken if a user that does not belong to the privileged class can read the transmission. For a given parameter k, our schemes should be resilient to any subset of k users that collude and any (disjoint) subset (of any size) of privileged users.

We also consider another scheme parameter, the *random-resiliency* of a scheme which refers to the expected number of users, chosen uniformly at random, that have to collide so as to break the scheme.

In many applications, it suffices to consider only the (weaker) random-resiliency measure. For example, if decryption devices are captured from random users, (or were assigned at random to users), it is the random resiliency that determines how many devices need be captured so as to break the scheme. We discuss a number of different scenarios with differing assumptions on the adversary strength. We show that even powerful and adaptive adversaries are incapable of circumventing the protection afforded by our schemes.

The final goal of the broadcast encryption scheme is to securely transmit a message to all members of the privileged subset. If cryptographic tools such as one-way functions exist then this problem can be translated into the problem of obtaining a common key. Let the security parameter be defined to be the length of this key.

1.1 Definitions

A broadcast scheme allocates keys to users so that given a subset T of U, the center can broadcast messages to all users following which all members of T have a common key.

A broadcast scheme is called *resilient* to a set S if for every subset T that does not intersect with S, no eavesdropper, that has all secrets associated with members of S, can obtain "knowledge" of the secret common to T. Knowledge here can have two different interpretations:

- The secret common to T has some a-priori distribution (usually the uniform distribution) and given the keys of S and the message transmitted by the center the conditional distribution of the secret is not changed. This is the "information-theoretic" definition of security.

- The secret of T is pseudo-random, i.e. no computationally bounded (by probabilistic polynomial time) eavesdropper can distinguish between the secret and a truly random string; even if the eavesdropper is provided with the keys of the coalition S the secret of T remains pseudo-random. This is the computational definition of security. For more information on pseudo-randomness, see [15], [4] or [10].

A scheme is called k*-resilient* if it is resilient to *any* set $S \subset U$ of size k. We also deal with random coalitions: a scheme is called (k, p)*-random-resilient* if with probability at least $1 - p$ the scheme is resilient to a set S of size k, chosen at random from U. Let $|U| = n$, we use n and $|U|$ interchangeably hereinafter.

The relevant "resources" which we attempt to optimize are

- The number of transmissions used by the center to create the common secret. (this is "wasted" bandwidth).
- The number of keys associated with each user. Since the user may be weak, i.e. a smart card, this should be minimized.
- The computation effort involved in retrieving the common key by the members of the privileged class.

1.2 Results

As a function of the resiliency required, we provide a large set of schemes that offer a tradeoff between the two relevant resources: memory per user and transmission length.

If nothing is known about the privileged subset T, any broadcast scheme requires that the transmission be sufficiently long to uniquely identify the privilege subset T. Otherwise, by a simple counting argument, there would be two non-identical sets, T and T', both of which somehow manage to obtain the same common key.

Thus, in general, simply representing a subset $T \subset U$ requires $|U|$ bits. Using our schemes, transmitting an additional $o(|U|)$ bits guarantees security against all coalitions of size $\tilde{O}(\sqrt{|U|})$ users and randomly chosen coalitions of $\tilde{O}(|U|)$ users. The computational and memory requirements for these schemes are $\tilde{O}(\sqrt{U})$. Thus, in some sense, security is available for "free".

In fact, in many contexts the privileged set may be identified by sending a relatively short transmission. *E.g.*, if the set can somehow be computed from an old privileged set or the set representation can be compressed. Thus, we distinguish between the *set identification* transmission and the *broadcast encryption* transmission. Our goal is the study of broadcast encryption transmissions and their requirements. In general, the center will identify every user with a unique identification number, and thus the set representation can be a bit vector. There are distinct advantages that the identification numbers be assigned at random to new users, we discuss this hereinafter in the context of random resiliency.

We distinguish between zero-message schemes and more general schemes. Zero-message schemes (Section 2) have the property that knowing the privileged subset T suffices for all users $x \in T$ to compute a common key with the center without any transmission. Of course, to actually use a zero-message scheme to transmit information implies using this key to encrypt the data transmitted.

More general schemes (Section 3) may require that the center transmit many messages. All the schemes we describe require that the length of the center generated messages be equal in length to the security parameter. Thus, when counting messages transmitted by the center, each messages is s bits in length.

Our general approach to constructing schemes is to use a two stage approach. First, we construct low resiliency zero-message schemes and then use these to construct higher resiliency schemes. The latter are not zero-message type schemes.

For low resiliency schemes, we describe assumption-free constructions, that are based upon no cryptographic assumption (the equivalent of a one-time pad). Then, we describe more efficient schemes based upon a some cryptographic assumptions, either the existence of a one way function or the more explicit assumption that RSA is secure. These results are described in Theorems 1, 2, 3.

We then deal with the more general case, and describe schemes of high resiliency (Section 3). For clarity of exposition, we describe our constructions in terms of the number of "levels" involved in the scheme construction. Informally, the levels refer to a sets of hash functions that partition and group users in a variety of ways. Our proofs are all based upon applications of the probabilistic method [1].

To obtain a resiliency of k, it suffices to store $k \log k \log n$ keys per user, while the number of messages transmitted by the center is $O(k^2 \log^2 k \log n)$ (Theorem 5). To obtain a random resiliency of k, with probability p, it suffices to store $\log k \log(1/p)$ keys per user, while the number of messages transmitted by the center is $O(k \log k \log(1/p))$ (Corollary 2). Other points along the tradeoff between memory and transmission length are given in Theorem 4.

1.3 Related Work

Several papers considered the problem of a center who wants to broadcast to a group (cf. [2, 6, 14]). However, all these schemes are "one-time", and the keys must be updated after every use.

Suppose that a user subscribes to a Pay-TV service, receives a decryption box and then opens it and duplicates it. There is nothing to stop him or her from doing so (except for tamper-proof hardware, which may be problematic). However, suppose that given an illegal box manufactured by such a user, or a coalition of k such users, it is possible to trace at least one "traitor". Then such a tracing scheme would work very well in conjunction with broadcast encryption: given the illegal box, a traitor is traced and its name is removed the privileged list. These can be repeated until the box is rendered useless. Chor, Fiat and Naor [7] have recently designed such traitor tracing schemes. The complexity of their schemes (in terms of the length of broadcast and number of keys stored) is similar to the schemes of this paper.

Blundo and Cresti [3] have recently provided tight lower bound for the information-theoretic version of the broadcast encryption problem discussed in this paper.

2 Zero Message Schemes

In this section we present several schemes that do not require the center to broadcast any message in order for the member of the privileged class to generate a common key. The main significance of the schemes presented in this section is their application as building blocks for the schemes presented in Section 3.

2.1 The Basic Scheme

The basic scheme we define allows users to determine a common key for every subset, resilient to any set S of size $\leq k$. The idea is very simple.

For every set $B \subset U$, $0 \leq |B| \leq k$, define a key K_B and give K_B to every user $x \in U - B$. The common key to the privileged set T is simply the exclusive or of all keys K_B, $B \subset U - T$. Clearly, every coalition of $S \leq k$ users will all be missing key K_S and will therefore be unable to compute the common key for any privileged set T such that $S \cap T$ is empty.

The memory requirements for this scheme are that every user is assigned $\sum_{i=0}^{k} \binom{n}{k}$ keys. With these requirements we need make no assumptions whatsoever. We therefore have

Theorem 1 *There exists a k-resilient scheme that requires each user to store $\sum_{i=0}^{k} \binom{n}{k}$ keys and the center need not broadcast any message in order to generate a common key to the privileged class.*

2.2 1-Resilient Schemes using Cryptographic Assumptions

We now see how to improve the memory requirements of the scheme described above using cryptographic assumptions such as "one-way functions exist" and that extracting prime roots modulo a composite is hard. The improvements are applicable to any k, however they are the most dramatic for $k = 1$.

2.3 A 1-resilient scheme based on one-way functions

Consider the 1-resilient version of the scheme described above. It requires every user to store $n + 1$ different keys. However, this can be reduced to $lceil \log n \rceil$ keys per user if the keys are *pseudo-randomly* generated from a common seed, as we explain below.

Assume that one-way functions exist and hence pseudo-random generators exist (see [13, 12])). Let $f : \{0,1\}^{\ell} \mapsto \{0,1\}^{2\ell}$ be a pseudo-random generator (the length of the output of f is twice the length of the input). We first explain how the key distribution is done. Associate the n users with the leaves of a balanced binary tree on n nodes. The root is labeled with the common seed $s \in \{0,1\}^{\ell}$ and other vertices are labeled recursively as follows: apply the pseudo-random generators f to the root label and taking the left half (first ℓ bits) of $f(s)$ to be the label of the left subtree while the right half (last ℓ bits) of $f(s)$ to be the label of the right subtree. This is similar to the construction of the tree in the generation of a pseudo-random function in [11].

By the scheme of Section 2.1, every user x should get all the keys except the one associated with the singleton set $B = \{x\}$. To meet this goal remove the path from the leaf associated with the user x to the root. The result is a forest of $\lceil \log n \rceil$ trees. Provide user x with the labels associated with the roots of these trees. Given a label of a root of a subtree it is easy to compute the labels of the leaves of that subtree. Hence user x can compute the all leaf labels (except $K_{\{x\}}$) without additional help.

On the other hand, given this information, $K_{\{x\}}$ is still pseudo-random for user x, as can be seen by a hybrid argument: if the labels provided to users x are (truly) random, then $K_{\{x\}}$ is indistinguishable from a random key (recall that $K_{\{x\}}$ was generated by an iterative application of a pseudo-random generator, which is in itself a pseudo-random generator (cf. [15])). Consider the distributions $\{\mathcal{D}_i | 1 \leq i \leq \lceil \log n \rceil\}$ such that $\mathcal{D}_i$ is the distribution where the first i labels are random and the rest are pseudo-random. Since by assumption $\mathcal{D}_0$ and $\mathcal{D}_{\lceil \log n \rceil}$ are distinguishable, there must be an i such that $\mathcal{D}_i$ and $\mathcal{D}_{i+1}$ are distinguishable. However, this is impossible since it would imply a distinguisher for f. Therefore we have:

Theorem 2 *If one-way functions exist, then there exists a* 1*-resilient scheme that requires each user to store* $\log n$ *keys and the center need not broadcast any message in order to generate a common key to the privileged class.*

This scheme is not 2-resilient, since *any* two users have (together) all the keys K_B. For instance, x and x' such that x is associated with a leaf in the left subtree of the root and x' is associated with a leaf in the right subtree of the root have the labels of both subtrees.

2.4 A 1-resilient scheme based on Computational Number Theoretic Assumptions

A specific number theoretic scheme, cryptographically equivalent to the problem of root extraction modulo a composite, can further reduce the memory requirements for $1 - resilient$ schemes. This scheme is cryptographically equivalent to the RSA scheme [17] and motivated by the Diffie-Hellman key exchange mechanism, and the original Shamir cryptographically secure pseudo-random sequence. [8, 18].

The center chooses a random hard to factor composite $N = P \cdot Q$ where P and Q are primes. It also chooses a secret value g of high index. User i is assigned key $g_i = g^{p_i}$, where p_i, p_j are relatively prime for all $i, j \in U$. (All users know what user index refers to what p_i). A common key for a privileged subset of users T is taken as the value $g_T = g^{p_T} \bmod N$ where $p_T = \prod_{i \in T} p_i$. Every user $i \in T$ can compute g_T by evaluating

$$g_i{}^{\prod_{j \in T-\{i\}} p_j} \bmod N$$

Suppose that for some $T \subset U$ and some $j \notin T$ user j could compute the common key for T. We claim that it implies that the user could also compute g: given $a^x \bmod N$ and $a^y \bmod N$ and x and y one can compute $a^{GCD(x,y)} \bmod N$ by performing a sequence of modular exponentiations/divisions on a^x and a^y (see [18]; this sequence is derived from applying the Euclidean GCD algorithm on the modular $\log_a$ of a^x and a^y). As the GCD of p_j and $\prod_{h \in T} p_h$ is 1, it follows that g can be computed by user j in this manner. Thus, the user could compute the p_j'th root of g^{p_j} while knowing only the composite N. Therefore if this is assumed to be hard, then the user cannot get the key common to T. Note however that this is not strong enough for our definition of security 9even the computational one), since the key for T is pseudo-random. If we relax this requirement to one that says that it is computationally hard to construct the common key, then we have:

Theorem 3 *If extracting root modulo composites is hard, then there exists a* 1*-resilient (under the relaxed definition) scheme that requires each user to store one key (of length proportional to the composite) and the center need not broadcast any message in order to generate a common key to the privileged class.*

This scheme is not 2-resilient since any two user can collude and compute g.

3 Low Memory k-Resilient Schemes

The zero message k-resilient schemes described in the proceeding section require for $k > 1$ a great deal of memory, exponential in k. In this section we provide several efficient constructions of k-resilient schemes for $k > 1$. Our schemes are based on a method of converting 1-resilient schemes into k-resilient schemes. Throughout this section we assume the existence of a 1-resilient scheme for any number of users. This can be taken as the no-assumption scheme, or any of the cryptographic assumption variants.

Let w denote the number of keys that a user is required to store in the 1-resilient scheme. I.e $w = n + 1$ if no cryptographic assumptions are made, $w = \log n$ if we assume that one-way functions exists and $w = 1$ if we assume that it is hard to extract roots modulo a composite. The efficiency of our schemes will be measured by how many w's they require.

3.1 One Level Schemes

Consider a family of functions $f_1, \ldots, f_l$, $f_i : U \mapsto \{1, \ldots, m\}$, with the following property: For every subset $S \subset U$ of size k, there exists some $1 \leq i \leq l$ such that for all $x, y \in S$: $f_i(x) \neq f_i(y)$. This is equivalent to the statement that the family of functions $\{f_i\}$ contains a perfect hash function for all size k subsets of U when mapped to the range $\{1, \ldots, m\}$. (See [16] or [9] for more information on perfect hash functions.)

Such a family can be used to obtain a k-resilient scheme from a 1-resilient scheme. For every $1 \leq i \leq l$ and $1 \leq j \leq m$ use an independent 1-resilient scheme $R(i, j)$. Every user $x \in U$ receives the keys associated with schemes $R(i, f_i(x))$ for all $1 \leq i \leq \ell$. In order to send a secret message M to a subset $T \subset U$ the center generates random strings $M^1, \ldots, M^\ell$ such that $\bigoplus_{i=1}^{l} M^i = M$. The center broadcasts for all $1 \leq i \leq \ell$ and $1 \leq j \leq m$ the message M^i to the privileged subset $\{x \in T | f_i(x) = j\}$ using scheme $R(i, j)$. Every user $x \in T$ can obtain all the messages $M^1, \ldots M^\ell$ and by Xoring them get M.

The number of keys each user must store is m times the number needed in the 1-resilient scheme. The length of the transmission is $\ell \cdot m$ times the length of the transmission for a zero message 1-resilient scheme, equal to the security parameter.

Claim 1 *The scheme described above is a k-resilient scheme*

Proof. For any coalition S of size at most k there is an $1 \leq i \leq \ell$ such that f_i is 1-1 on S. In the schemes $R(i, j)$, $1 \leq j \leq m$ the coalition S has at most the keys of a single user (which is not part of T). Given the transmissions of $R(i, j)$ only, then by assumption, S gets no information about M_i (in the information theoretic definition of security) or M_i is pseudo-random (in the computational definition of security). Furthermore, given the combined information of the schemes $R(i, j)$, M_i is still random (in the information theoretic case) and remains pseudo-random in the computational case. The latter can be seen by hybrid argument. Therefore, even if $M^{i'}$ is known to the eavesdropper for all $i' \neq i$, no knowledge is gained about $M = \bigoplus_{i=1}^{l} M^i$. $\square$

We now see what values can m and ℓ take. It turns out that setting $m = 2k^2$ and $\ell = k \log n$ is sufficient. This can be seen via a probabilistic construction. Fix $S \subset U$ of size k. The probability that a random f_i is 1-1 on S is at least

$$1 - \binom{k}{2} \cdot \frac{1}{m} = 1 - \frac{k(k-1)}{2k^2} \geq \frac{3}{4}.$$

Therefore the probability that for no i we have that f_i is 1-1 on S is at most $1/4^\ell = 1/n^{2k}$. Hence the probability that for all subsets $S \subset U$ of size k there is a 1-1 f_i is at least $1 - \binom{n}{k} \cdot \frac{1}{n^{2k}} \geq 1 - \frac{1}{n^k}$. We therefore conclude

Theorem 4 *There exists a k-resilient scheme that requires each user to store $O(k \log n \cdot w)$ keys and the center to broadcast $O(k^3 \log n)$ messages. Moreover, the scheme can be constructed effectively with arbitrarily high probability by increasing the scheme parameters appropriately.*

The proof implies that against a randomly chosen subset $S \subset U$ of size k we can have a more efficient scheme, since we can take ℓ to be $\log_4(1/p)$:

Corollary 1 *For any $1 \leq k \leq n$ and $0 \leq p \leq 1$ there exists a (k,p)-random-resilient scheme that requires each user to store $O(\log(1/p) \cdot w)$ keys and the center to broadcast $O(k^2 \log(1/p))$ messages. Simply choose $m = k^2$ and $\ell = \log p$. Moreover, the scheme can be constructed effectively.*

As for explicit constructions for the family $f_1, \ldots f_\ell$, they seem to be at least a factor of k more expensive. One possibility of construction is via error-correcting-codes of large relative distance (say $1 - 1/k^2$) over an alphabet of size $O(k^2)$. For a simple construction, Consider the family

$$F = \{f_p(x) = x \bmod p | p \leq k^2 \log n \text{ and is a prime}\}$$

F satisfies the above requirement.

The number of keys stored per user in this explicit construction is $O(k^2 \log n / \log \log n)$ and the number of messages that the center broadcasts is $O(k^4 \log^2 n / \log \log n)$.

3.2 Remarks

After having seen the single-level schemes above, we wish to clarify certain points that can be discussed only after seeing an example of the types of schemes we deal with. We continue with more efficient multi-level schemes in the next section. The remarks of this section are applicable to both single and multi level schemes.

3.2.1 Representing the Functions.

In some applications using probabilistic constructions is problematic because of representation problem, i.e that storing the resulting structure may be prohibitively expensive. However, as described above, our schemes do not absolutely require that the f_i functions be computable, the user could simply be assigned $f_i(x)$. This could be chosen at random. The center could in fact generate all required functions from a pseudo-random function and a single seed.

Alternatively, instead of using completely random functions one can use function with limited independence, such as random polynomials of degree d (see [1] for information on limited independence functions). The results regarding the probabilistic construction of this section require only pairwise independence (we need to worry about collisions), and those of the next section require $\log k$-wise independence. The advantage is that there is a *succinct* representation for the functions now. Storing such function representations in the user decryption devices is not much more expensive than storing the keys required in the above schemes.

3.2.2 Reducing Storage.

Suppose that we are interested in limiting the number of keys that a user must store (at the the expense of the number of keys that the center must broadcast). We can get a certain tradeoff: instead of hashing to a range of size $2k^2$ we hash to range of size $m = a \cdot k^2$. The results that we get in this case are that the memory requirements are smaller by a $\log a$ factor and the broadcast requirements are larger by a factor of a. This is true for both k-resilient schemes and for (k, p)-random-resilient schemes.

We now describe yet another tradeoff that may reduce storage requirements. Every $R(i, j)$ scheme above deals with a subset of the users. If we assume that the f_i functions can be computed by anyone (e.g., k-wise independent functions as described above), then the $R(i, j)$ 1-resilient schemes can be devised so as to deal with the true number of users associated with the scheme, depending on the underlying 1-resilient scheme, this leads to a saving in the memory requirements described in the scheme, at the expense of some additional computation.

3.2.3 Adversary Limitations and Resiliency.

A k-resilient scheme is resilient to any coalition of size k, this means that irrespective of how the adversary goes about choosing the coalition, no coalition of size smaller than k will be of any use to the adversary. However, the scheme is resilient to many sets of size much larger than k.

The adversary may capture devices at random, in this case the random resiliency measure is directly applicable. Given a $(V, 1/2)$ randomized resilient scheme, the expected number of devices that the adversary must capture to break the scheme is at least $V/2$.

A possibly legitimate assumption is that a user of the decryption device does not even know his unique index amongst all users. For example, the user index and all user secrets could be stored on a (relatively) secure smartcard, such a smartcard is probably vulnerable, but not to a casual user. Thus, if user indices are assigned at random any set of devices captured will be a random set irrespective of the adversary strategy used.

The definition of (k, p) random resiliency is somewhat problematic for two reasons:

1. The probability p is an absolute probability, this does not make sense if the underlying one resilient schemes we are using can be themselves broken with relatively high probability (e.g., by guessing the short secret keys).

2. The assignment of users ids (index numbers) to users is assumed to be random and secret. But, it may be possible to learn the user identification by monitoring transmissions and user behavior.

To avoid both these problems we define a new notion of resiliency and say that a scheme is (k, p)-*immune* if for any adversary choosing adaptively a subset S of at most k users and a disjoint subset T we have: the probability that the adversary (knowing all the secrets associated with S) guesses the value the center broadcasts to T is larger by at most (additive) p than the probability the adversary would have guessed it without knowing the secrets of S.

If we assume that the functions f are kept secret then the results we can get for (k, p)-immune schemes are very similar to the results for (k, p)-random-resilient schemes. However, we do not know whether this holds in general for all random-resilient schemes. This is true since the random constructions for both single level schemes and multi level schemes (described in the next section), the analysis fixes the subset S and evaluates the probability that it is good for a random construction. Since the adversary does not know the values of the hash functions (f_i for single level schemes) when adding a user to S, any choice of S has the same probability of being bad.

For completeness, we note that yet another attack is theoretically possible, although it may be rather difficult in practice. The adversary may attempt to actively subvert the system by publishing a solicitation for dishonest users that meet certain criteria. Specifically, it would be very useful for the adversary to capture pairs of devices that belong to the same 1-resilient $R(i, j)$ scheme described above, if he captures ℓ pairs (a_i, b_i) such that $f_i(a_i) = f_i(b_i)$ then he has corrupted our scheme above. In this case, a true k-resilient scheme is the only prevention. If k is sufficiently large and the number of traitors does not exceed k then the scheme is secure.

3.3 Multi-Level Schemes

We now describe a general multi-level scheme that converts a scheme with small resiliency to one with large resiliency. Consider a family of functions $f_1, \dots, f_l$, $f_i : U \mapsto \{1, \dots, m\}$ and a collection of sets of schemes,

$$\{R(i, j) | 1 \leq i \leq l, 1 \leq j \leq m\},$$

where each $R(i, j)$ consists of w schemes labeled $R(i, j, 1), \dots, R(i, j, w)$. These functions and schemes obey the following condition: For every subset $S \subset U$ of size k, there exists some $1 \leq i \leq l$ such that for all $1 \leq j \leq m$ there exists some $1 \leq r_j \leq w$ such that the scheme $R(i, j, r_j)$ is resilient to the set $\{x \in S | f_i(x) = j\}$.

We claim that such a structure can be used to obtain a k-resilient scheme: Generate independently chosen keys for all schemes $R(i, j, r)$. A user $x \in U$ receives for every $1 \leq i \leq l$ and every $1 \leq r \leq w$ the keys associated with x in scheme $R(i, f_i(x), r)$. Given a subset $T \subset U$ and a secret message M, the center generates:

- Strings $M^1, \dots, M^l$ such that $\bigoplus_{i=1}^{l} M^i = M$ and $M^1, \dots, M^{l-1}$ are chosen at random.
- For every $1 \leq i \leq l$, and $1 \leq j \leq m$ random strings $M_1^{(i,j)}, \dots, M_w^{(i,j)}$, such that $\bigoplus_{t=1}^{w} M_t^{(i,j)} = M^i$.

The center broadcasts for all $1 \leq i \leq \ell$ and $1 \leq j \leq m$ and $1 \leq r \leq w$ the message $M_r^{(i,j)}$ to the privileged subset $\{x \in T | f_i(x) = j\}$ using scheme $R(i, j, r)$. Every user $x \in T$ can obtain for all $1 \leq i \leq \ell$ and $1 \leq r \leq w$ messages $M_r^{(i, f_i(x))}$. To reconstruct the message M, the user $x \in T$ takes the bitwise exclusive or of all messages transmitted to the user in all schemes to which the user belongs, i.e., in all schemes $R(i, j, r)$ such that $f_i(x) = j$.

The number of keys associated with user x is therefore the number of keys associated with a scheme $R(i, j, r)$ times $l \times w$. The length of a broadcast is equal to the number of messages transmitted in an $R(i, j, r)$ scheme times $l \times m \times w$.

Claim 2 *The scheme described above is a k-resilient scheme.*

Proof. For any coalition S of size at most k there is, by assumption, an $1 \leq i \leq \ell$ and $r_1, r_2, \dots r_m \in \{1 \dots w\}$ such that the schemes $R(i, j, r_j)$ are resilient to S. Therefore, for all $1 \leq j \leq m$ the value of $M_{r_j}^{(i,j)}$ is random or pseudo-random for S and hence the value of $M^i = \bigoplus_{t=1}^{w} M_t^{(i,j)}$ is random or pseudo-random for S which implies that no knowledge is gained about M. $\square$

We now describe a concrete two level scheme using this method. Set $\ell = 2k \log n$, $m = k/\log k$, $t = 2e \log k$ and $w = \log k + 1$. The first level consists of a family of ℓ functions $f_1, \ldots, f_l$, $f_i : U \mapsto \{1, \ldots, m\}$. The second level consists of functions $g_r^{(i,j)} : U \mapsto \{1, \ldots 2t^2\}$ for all $1 \leq i \leq \ell$, $1 \leq j \leq m$ and $1 \leq r \leq w$. Every such (i, j, r) and $1 \leq h \leq 2t^2$ defines a 1-resilient scheme $R(i, j, r, h)$ as in the scheme of Section 3.1. Every user x receives the keys of schemes $R(i, f_i(x), r, g_r^{(i,f_i(x))}(x))$ for all $1 \leq i \leq \ell$ and $1 \leq r \leq w$.

For a set $S \subset U$ of size k we say that i is *good* if:
for all $1 \leq j \leq m$

1. $|\{x \in S | f_i(x) = j\}| \leq t$.
2. there exists $1 \leq r \leq w$ such that $g_r^{(i,j)}$ is 1-1 on $\{x \in S | f_i(x) = j\}$.

By Claim 2 we know that if for every set $S \in U$ of size k there is a good i, then the scheme is k-resilient.

We prove that randomly chosen f_i and $g_r^{(i,j)}$ constitute a good scheme with reasonably high probability.

Fix a subset $S \subset U$ of size k and $j \in \{1 \ldots m\}$. The probability that Condition 1 above is not satisfied is at most

$$\binom{k}{t} \cdot (\frac{1}{m})^t \leq (\frac{ek}{2e \log k})^{2e \log k} \cdot (\frac{\log k}{k})^{2e \log k} = (\frac{1}{2})^{2e \log k} = \frac{1}{k^{2e}}$$

Suppose that condition 1 is satisfied, then for any $1 \leq r \leq w$ the probability that $g_r^{(i,j)}$ is 1-1 on $\{x \in S | f_i(x) = j\}$ is at least $1 - t\frac{1}{2t} = \frac{1}{2}$. Hence the probability that condition 2 is not satisfied is at most $1/2^w = 1/2k$ and therefore the probability that Conditions 1 and 2 are both satisfied for every $1 \leq j \leq m$ is at least $1/2$. The probability that *no* i is good for S is at most $1/2^\ell = 1/n^{2k}$. Hence the probability that all subsets $S \subset U$ of size k have a good i is at least

$$1 - \binom{n}{k} \cdot \frac{1}{n^{2k}} \geq 1 - \frac{1}{n^k}.$$

We therefore conclude:

Theorem 5 *There exists a k-resilient scheme that requires each user to store $O(k \log k \log n \cdot w)$ keys and the center to broadcast $O(k^2 \log^2 k \log n)$ messages. Moreover, the scheme can be constructed effectively with high probability.*

As in Theorem 4, the proof implies that against a randomly chosen subset $S \subset U$ of size k we can have a more efficient scheme:

Corollary 2 *For any $1 \leq k \leq n$ and $0 \leq p \leq 1$ there exists a (k, p)-random-resilient scheme with the property that the number of keys each user should store is $O(\log k \log(1/p) \cdot w)$ and the center should broadcast $O(k \log^2 k \log(1/p))$ messages. Moreover, the scheme can be constructed effectively with high probability.*

4 An Example and Implementation Considerations

The schemes described in this paper are valid for all possible values of the parameters. However, if random resiliency suffices, and if one seeks a solution to a concrete example then other considerations creep in.

Say we've got a user group of one billion subscribers. Also, assume that our goal is that to discourage any possible pirate box manufacturer, and thus the expectation should be that he is required to capture $k = 100,000$ devices before seeing any return on his or her investment.

Basing our 1-resilient scheme on the number theoretic scheme, and using our randomized $(100000, 1/2)$-resilient scheme, the number of keys stored in every subscriber decryption device is less than 20, and the length of a broadcast enabling transmission is on the order of two million keys. (Vs., one billion keys transmitted for standard schemes).

However, there is a major problem, with the set identification transmission. It seems that all subscribers will have to listen to one billion bits of set identification transmission without making a single error. In fact, the subscriber is apathic to the presence or absence of most of the users. It is only users that belong to the same underlying 1-resilient schemes that he belongs to that matter. Thus, there are advantages to splitting up users into independent broadcast encryption schemes, determining what user gets assigned to what scheme at random. By appropriately resynchronizing and labeling schemes, the decryption device will only have to deal with the set identification transmission dealing with one (smaller) scheme.

There is a tradeoff between error control issues and security. If the number of broadcast encryption schemes gets too large, and the resiliency gets too small, then the (multiple) birthday paradox enters into consideration. (We say such a scheme is broken if any of it's component broadcast encryption schemes is broken).

Say we split the billion users above into randomly assigned broadcast encryption groups of 1000 users. We use a non-random 5-resilient broadcast encryption scheme which requires about 10 keys stored per user, and 100 keys transmission per broadcast encryption scheme, for a total of 10^8 key transmissions. The total random resiliency is approximately $1,000,000^{5/6} = 100,000$. (The adversary must randomly select devices until he has 5 different devices from the same broadcast encryption scheme). Transmissions are 50 times longer than before, but still significantly shorter than individual transmissions. This is a practical scheme since there is no longer any serious error control problem.

Another advantage of the scheme presented in this section is that if the adversary is in fact successful, after collecting 100,000 decryption devices, and if we have captured one of the adversary eavesdropping devices, all is not lost. It is still a relatively simple matter to disable all adversary devices by disabling one group of 1000 users, splitting these users amongst other groups, the adversary effort has been in vain.

References

[1] N. Alon and J. Spencer, **The Probabilistic Method**, Wiley, 1992.

[2] S. Berkovits, *How to Broadccast a Secret*, Advances in Cryptology - Eurocryp'91, Lecture Notes in Computer Science 547, Springer, 1991, pp. 536–541.

[3] C. Blundo and A. Cresti, *Space Requirements for Braodcast Encryption*, Advances in Cryptology - Eurocryp'94, Lecture Notes in Computer Science 950 Springer, 1995, pp. 287–298.

[4] R. Boppana and R. Hirshfeld *Pseudorandom generators and complexity classes*, Advances in Computing Research; Volume 5 on Randomness and Computation.

[5] J. L. Carter and M. N. Wegman, *Universal Classes of Hash Functions*, Journal of Computer and System Sciences 18 (1979), pp. 143–154.

[6] G. H. Chiou and W. T. Chen, *Secure Broadcasting using the Secure Lock*, IEEE Trans. on Software Engineering, vol 15, 1989, pp. 929–934.

[7] B. Chor, A. Fiat and M. Naor, *Tracing traitors*, Advances in Cryptology - Crypto'94, Lecture Notes in Computer Science No. 839, Springer Verlag, 1994, 257–270.

[8] W. Diffie and M. Hellman, *New Directions in Cryptography* , IEEE Trans. on Information Theory, vol. IT-22, 6 (1976), pp. 644–654.

[9] M.L. Fredman, J. Komlós and E. Szemerédi, *Storing a Sparse Table with* $O(1)$ *Worst Case Access Time*, Journal of the ACM, Vol 31, 1984, pp. 538–544.

[10] O. Goldreich, *Foundations of Cryptography (Fragments of a Book)*, available in "http://www.eccc.uni-trier.de/eccc/info/eccc-books.html".

[11] O. Goldreich, S. Goldwasser and S. Micali, *How to Construct Random Functions* Journal of the ACM **33**, 1986, pp. 792–807.

[12] J. Hastad, *Pseudo-Random Generators under Uniform Assumptions*, Proc. 19th Symposium on Theory of Computing, 1990.

[13] R. Impagliazzo, L. Levin and M. Luby, *Pseudo-random Generation given from a One-way Function*, Proc. of the 20th ACM Symp. on Theory of Computing, 1989.

[14] C. Laih, J. Lee and L. Harn, *A New Threshold Scheme and its Application is Desiging the conference Key Distribution Cryptosystem*, Information Processing Letters, vol 32, 1989, pp. 95–99.

[15] M. Luby, **Pseudo-randomenss and Applications**, Princeton Universtity Press, To appear.

[16] K. Mehlhorn, **Data Structures and Algorithms: Sorting and Searching**, Springer-Verlag, Berlin Heidelberg, 1984.

[17] R. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signature and Public Key Cryptosystems*, Comm. of ACM, 21 (1978), pp. 120–126.

[18] A. Shamir, *On the Generation of Cryptographically Strong Pseudo-Random Number Sequences*, ACM Trans. Comput. Sys., 1 (1983), pp. 38–44.

[19] M. N. Wegman and J. L. Carter, *New Hash Functions and Their Use in Authentication and Set Equality*, Journal of Computer and System Sciences 22, pp. 265-279 (1981).