On Key Distribution via True Broadcasting

Mike Just*‡ (just@scs.carleton.ca)
Evangelos Kranakis*§ (kranakis@scs.carleton.ca)
Danny Krizanc*§ (krizanc@scs.carleton.ca)
Paul van Oorschot*† (paulv@bnr.ca)

August 30, 1994

Abstract

We consider true broadcast systems for the secure communication of session keys. These schemes provide for parallel rather than serial construction of broadcast messages, while avoiding selective broadcasting. We begin by introducing a conceptual framework for true broadcasting and illustrate its design with a secure key broadcast scheme based on probabilistic encryption. The framework provides for a system requiring user anonymity, as a result of the absence of addressing for the broadcast message. We also illustrate how Shamir's threshold scheme can be altered to allow for parallel broadcasting. We then present a formal model and use information theoretic techiques to establish a lower bound on the size of the broadcast message for a class of true broadcast schemes. Finally, we improve upon the aforementioned threshold scheme such that it achieves the lower bound.

1991 AMS Classification: 94A60
CR Categories: D.4.6
Key Words and Phrases: Key distribution, broadcast encryption, secret sharing.

1 Introduction

Consider a system consisting of a set U of users and a trusted center T. Suppose that T wishes to share a large message M with a subset P of *privileged* users. There are two ways for this exchange to take place.

T can send M individually to each user in P. This is known as a *point-to-point* exchange. It involves sending *multiple* copies of the *same* message to the privileged users. It also requires the address of the intended user to be appended to each message. This is inefficient. Alternatively, T could *broadcast* the message M. This is known as a *point-to-multipoint* exchange (see [12]). The set P will consist of those users from U who share a communication line with T (e.g. a local cable company). This method has the advantage that no addressing of the message is required and only a single message is distributed. Therefore, broadcasting a message is preferable to a point-to-point exchange.

Now suppose that T would like to initiate a *secure* communication, i.e. send a message to a subset of the users who share a communication line with T. As an example, for *pay-tv*, even though many users have a communication line with the cable company, not all of these users will receive the pay-tv channels.

To facilitate encryption, suppose that T shares a secret key *a priori* with each user in U, where there are t users in P. Implementation of a point-to-point exchange now requires t separate encryptions of M along with the individual addressing and distribution of each of these encrypted messages. An alternative is to concatenate these individual encryptions and broadcast them as a single message. However, this does not remove the need for the individual encryptions and addressing (each user must be able to identify his own encrypted piece to recover M).

The major problem with these secure point-to-point solutions is that if M is very large, T will have an enormous amount of computation to perform t distinct encryptions of M, and will use a large amount of bandwidth to distribute the t encryptions. To solve this problem, suppose T were to first share a *session key* K with the members in P. Since K will be much smaller than M, the amount of work performed by T to encrypt M will be much less. Henceforth, T can use K to encrypt a single copy of M. Since only users in P possess K, the broadcast of M encrypted under K is secure.

The problem now becomes one of distributing K. One naive solution would be for T to share a number of keys with users in U during system setup. Each key would correspond to a particular privileged set that a user could belong to. However, since the number of possi-

*Carleton University, School of Computer Science, Ottawa, ON, Canada. K1S 5B6
†Bell Northern Research, P.O. Box 3511, Station C, Ottawa, ON, Canada. K1Y 4H7
§Research supported in part by NSERC grant.
‡Research supported by NSERC graduate fellowship.

ble sets is exponential in the number of users, each user would be required to store a large number of keys.

A second solution would be for T to share K with users in P using a secure point-to-point communication. However, this solution inherits the problems encountered when attempting to securely exchange M point-to-point. Namely, t separate encryptions of K are required, along with the need for an address to be appended to each encryption. A more subtle problem with point-to-point exchanges in general is that they do not allow for user *anonymity*. In other words, the address on the message that allows each user to decrypt the correct message, also allows others to identify this user as a receiver of a message from T.[1]

One solution suggested to us to solve the anonymity problem using a point-to-point exchange would be to send an individual encrypted message to each user. The privileged users would receive K upon decryption of their message while all other users would receive "junk" upon decryption. This is a common solution for protecting against traffic flow analysis. However, individual encryptions and the individual addressing of each encryption are still required for *all* users, rather than just the members of the privileged set.

The broadcasting of a single encrypted version of K avoids the problems of multiple encryptions, addressing and anonymity. The broadcast message is created such that *all* users receive the message, yet only those users in P have the ability to recover K from this message. Such a set-up is ideal for multimedia broadcast applications such as pay-per-view (see [10]). In this paper, we examine methods of broadcasting a session key K.

1.1 Outline and Summary of Results

The remainder of this Section introduces the terms and notation used throughout the paper, and summarizes previous work in key broadcasting. Section 2 presents a general conceptual framework for the broadcasting of session keys, and discusses its advantages over previous models. In Section 3, we discuss an implementation which satisfies the new framework with security based on Goldwasser-Micali [11] probabilistic encryption. Section 4 examines how the concept of a threshold scheme can be applied to broadcasting session keys. We review an implementation from Berkovits [1] that uses Shamir's threshold scheme. In Section 5, we define and formalize a model for a general class of session key broadcasting schemes. A lower bound for the number of bits transmitted by T using such a model is presented. This lower bound corresponds to the bandwidth required for t copies of the session key, for a privileged set of size t. We then improve upon Berkovits' implementation, resulting in an optimal scheme with respect to the lower bound.

[1] If the address were encrypted along with the key, this would introduce more work for the users as they must decrypt every piece up to their own rather than just a search through every piece. Also, this option is also not always available to T, e.g. in end-to-end encryption, destination addresses are required for intermediate nodes.

1.2 Definitions and Notation

We examine systems in which there is a set U of n users $U_1, U_2, \ldots, U_n$, and a trusted center T. T wishes to send a large message to a privileged subset P of users, where $|P| = t$ for some integer $t \leq n$. Every U_i will share a secret s_i with T, where each s_i is an s-bit random string. The secure distribution of s_i can be achieved when the user subscribes with T. Without loss of generality, we assume that $\{s_1, \ldots, s_t\}$ is the set of shares for users in P.

We present a model satisfying the definition of a *true broadcast system*, for broadcasting a session key K. According to [1], this is a scheme "in which the broadcast message contains the same information for each and every listener", yet only members in P can recover K from this message. A true broadcast system is referred to as a *parallel* construction of the broadcast message since T only performs one encryption of K and each user uses the same broadcast message to recover K. On the other hand, a *serial* construction would involve separate encryptions of K for each user in P, followed by the generation of a broadcast message from the separate encryptions. From the broadcast message, each user would first recover his own encrypted piece and then recover K.

1.3 Previous Work

The concept of *secure broadcasting* was proposed in [8]. Subsequent variations appear in [7, 16]. Associated with each user U_i is a secret key s_i and an integer l_i. The sender X of messages could equally be a user or a trusted center. Distribution of a session key assumes the existence of a secure cryptosystem between X and each user in P, hence U_i shares s_i with X. Secure broadcasting uses a serial construction. K is separately encrypted by X with each s_i to produce w_i for $U_i \in P$. A so-called *sealed lock* L is then constructed from these w_i and broadcast to all users. Using their l_i, a user can recover their w_i and decrypt it with s_i.[2]

One such sealed lock construction proceeds as follows. Suppose that l_i is a prime number, distinct for each U_i. Upon separately encrypting K with each s_i to produce $w_i, \forall i : U_i \in P$, X will use the system of equations $y_i \equiv w_i \bmod l_i$ and the Chinese Remainder Theorem to produce a unique sealed lock $0 < L < \prod_{i:U_i \in P} l_i$. Upon

[2] Two variations exist. X and U_i can share a common s_i, or U_i can publicize a key e_i and keep a secret key d_i. X would encrypt K with e_i while U_i would use d_i to decrypt (see [8]).

receipt of L, each $U_i \in P$ reduces L modulo his l_i to recover w_i. Subsequently, s_i is used to decrypt w_i and recover K.

As described, the term sealed lock is misleading. Since the l_i are made public, anyone can recover a user's piece from L (yet only U_i can subsequently recover K). The purpose of publicizing them is to allow one user to broadcast a single message from which only the privileged group of users can recover K, without having to share an l_i with each of them. The security of the system will depend on the security of the encryption function used.

A major shortcoming of this system arises if it is necessary to update the session key for P. Updating the session key requires re-encrypting K for each user, followed by computing a new lock L. We present a method using a parallel construction, which alleviates this problem.

Blundo *et al.* [4] consider schemes whereby a trusted off-line server initially distributes shares to all users, and subsequently t-subsets of users can recover a predefined key as a function of their own share *and the identities of* $t-1$ *other users*. However, neither these "non-interactive" schemes, nor the interactive ones from [4] provide user anonymity.

Broadcast encryption was introduced in [9]. Subsequent bounds on the size of user's shares were presented by Blundo *et al.* in [5]. Each user shares a number of keys with a trusted center T. The scheme is designed to prevent coalitions of users from conspiring to decrypt the broadcasted message. However, the solution uses *selective broadcasting*. Rather than broadcasting the same message for all users, they require that messages be sent only to specific intended destinations. This is not a characteristic of a true broadcast system. Also, this system requires each user to store a number of keys; the number, and the size of the broadcast message are both dependent upon the size of the anticipated coalition. Our proposed schemes avoid these properties.

Using *secret sharing* to broadcast session keys was introduced by Laih *et al.* [15], and subsequently by Berkovits [1]. We examine this model in more detail, in Section 4.

2 A Conceptual Framework for True Broadcasting

In this Section we present a conceptual framework for broadcasting a session key K with user anonymity. Users have no knowledge of the other members in P, nor do they require this knowledge.

Once the privileged set P has been defined, T generates a broadcast encryption key K_P using the secret keys of the users in P as follows:

$$K_P = f(s_1, s_2, \ldots, s_t), \tag{1}$$

for an appropriate function f. T then computes

$$C = E_{K_P}(K), \tag{2}$$

where C is the broadcast message. E is an "encryption function" parameterized by the "key" K_P. f and E are designed such that only knowledge of a single s_i from $U_i \in P$ is required to recover K from C, i.e.

$$K = D_{s_i}(C), \forall U_i \in P,$$

for the "decryption function" D corresponding to E. This will become more meaningful with the scheme presented in Section 3.1.

An important difference from previous methods is the ease with which one can update K. By this general approach, one need only recalculate and broadcast (2) to establish a new session key K, whereas [8] and others require separately re-encrypting the new K for each user, followed by recalculation of the sealed lock. In fact, our model could more suitably be called a sealed lock, with master key K_P and equally effective keys s_i for each user $U_i \in P$.

3 Using Number Theory to Achieve True Broadcasting

In Section 3.1, we will present a secure number theoretic scheme satisfying the model of Section 2. We begin here by motivating the setup with a preliminary, albeit insecure scheme.

Let T share a distinct, independent prime number p_i with each of n users. The broadcast key K_P from (1) is simply N where $N = p_1 \cdots p_t$. In this way, the broadcast key contains information for all users in P. T randomly selects an integral session key K in the range $0 < K < min\{p_1, p_2, \ldots, p_t\}$. The broadcast message C from (2) is the integer $C = K^{-1} \bmod N$. Upon receipt of C, each $U_i \in P$ computes

$$K = C^{-1} \pmod{p_i} \tag{3}$$

In accordance with the model of Section 2, given only the single piece that each $U_i \in P$ shares with T, allows recovery of K. Consider that if $KK^{-1} \equiv 1 \pmod N$, then by the Chinese Remainder Theorem, $KK^{-1} \equiv 1 \bmod p_i$ for each prime factor p_i of N. Therefore, for each p_i that $K < p_i$ holds true, computation of K from K^{-1} with only p_i is realized. This is done by reducing K^{-1} modulo p_i, and computing its inverse in $Z^*_{p_i}$.

Thus, we have constructed a scheme whereby T can encrypt a message K using the secrets of *all* users in

P, yet only a *single* key from a user in P is sufficient to recover K. However, this scheme is not secure since coalitions of users can, over time, recover information about the secret keys of other users. This is a result of the fact that the equation $KK^{-1} \equiv 1 \pmod{p_i}$ is valid for all $U_i \in P$.

3.1 True Broadcasting With Quadratic Residues

We now consider a secure scheme built upon the framework introduced in Section 2. First, we recall some important number theoretic properties.

Given a prime number p, q is a *quadratic residue mod* p (denoted $q \in QR_p$) if $\exists x \in Z_p^*$ such that $x^2 \equiv q \pmod{p}$ for $q \in Z_p^*$. If q is not a quadratic residue, then q is a *quadratic non-residue* (denoted $q \in QNR_p$).

Now given an integer $N = p_1 p_2 \cdots p_t$, where each p_i is a distinct prime,

$$q \in QR_N \quad \Leftrightarrow \quad q \in QR_{p_1} \cap \cdots \cap QR_{p_t} \tag{4}$$

$$q \in QNR_N \quad \Leftrightarrow \quad q \in QNR_{p_1} \cup \cdots \cup QNR_{p_t} \tag{5}$$

Notice that in (4), all of $q \in QR_{p_i}$ must be true, while in (5), only one of $q \in QNR_{p_i}$ need be true.

Given an integer q, one can determine whether or not q is a quadratic residue modulo a prime p, by performing the following test

$$\begin{aligned} q^{\frac{p-1}{2}} &\equiv 1 \pmod{p} \Rightarrow q \in QR_p \\ &\equiv -1 \pmod{p} \Rightarrow q \in QNR_p \end{aligned} \tag{6}$$

We now present a secure scheme based on quadratic residuosity.

T shares a distinct, independent prime number p_i with each of n users. Let $K = k_0 k_1 \cdots k_{l-1}$ be the binary representation of the session key. K_P is constructed as follows. First T calculates

$$N = p_1 p_2 \cdots p_t \tag{7}$$

where p_i is the secret prime that T shares with user $U_i \in P$. T will produce a y such that $y \in QNR_N$. We require that $y \in QNR_{p_i}$ for each p_i in (7), which may be done by choosing random $y_i \in Z_p^*$ and using (6) to determine if $y_i \in QNR_{p_i}$. Alternatively, T can maintain a long-term y_i for each U_i. In either case, T can then use the Chinese Remainder Theorem with each y_i and p_i to solve for y such that $y \in QNR_N$. Here, $K_P = (y, N)$ serves as the broadcast key.

For the encryption function E in (2), we use a variant of the method of *probabilistic encryption* from [11]. Sending one bit of K will require broadcasting a $\log_2 N$ bit integer. For T to broadcast a bit k_b such that only $U_i \in P$ can recover k_b, T selects a random $x \in Z_N^*$ and computes $C_b = x^2 y^{k_b} \bmod N$ and broadcasts C_b to all users.

Now consider the following two possibilities. If $k_b = 0$ then $C_b \equiv x^2 \bmod N$ and thus $C_b \in QR_N$. From (4), this implies $C_b \in QR_{p_i}$ for each prime p_i. By (6), each U_i who possesses a prime divisor p_i of N can determine that $C_b \in QR_N$ and conclude that $k_b = 0$. If $k_b = 1$, then $C_b \equiv x^2 y \bmod N$ and thus $C_b \in QNR_N$. Since y was chosen such that $y \in QNR_{p_i}$ for each p_i in the factorization of N, again by (6) each privileged user can determine that $C_b \in QNR_N$ and conclude that $k_b = 1$. Users $U_j \notin P$ cannot recover k_b as they lack the appropriate primes p_i.

The size of the broadcast message is $l \log_2 N$ bits, where N is encoded with $t \log_2 p$ bits, for $p = max(p_i)$ and an l-bit session key K. This clearly is not a practical method of broadcasting. However, it does illustrate a secure scheme, built upon the framework presented in Section 2.

Once a suitable y has been selected, the amount of work performed by the center to produce C_b is at most 2 modular multiplications in Z_N^*. To broadcast all of K, this process is repeated l times. Each U_i using only their prime p_i to recover K (in the manner described above), requires at most $2\lceil \log_2(p_i) \rceil$ modular multiplications in $Z_{p_i}^*$ for each bit of C received (subsequent to the reduction of the $\log_2 N$ bit C with the modulus p_i). Since l bits are broadcast, this operation is repeated l times. Due to the significant degree of data expansion, resulting from the $\log_2 N$ per bit expansion, we consider the scheme to be of theoretical interest only.

The security of this scheme is based on the assumption that an opponent can not determine the quadratic residuosity of an integer $q \bmod N$ without knowledge of N's prime factors. Given an integer $q \in Z_N^*$ and N, it is shown in [11] that if determining quadratic residuosity was easy to solve for some q, then it could be solved easily for all q. The fact that N is kept secret, gives an even stronger result.

4 Using Secret Sharing to Achieve True Broadcasting

Recall that a (t, n) *threshold scheme* implies a method for sharing a secret key K among a finite set U of n users such that a subset P of at least $t \leq n$ users from U can recover K while no group of size less than t can do so (see [2, 17, 19]). A trusted center T randomly selects a key K and uses a *concealer function* to produce n *shares* or *shadows* on input K. The n shares are then secretly distributed to the n users, each user receiving exactly one share. To recover K, the users in the priviliged set P, where $|P| \geq t$, combine their shares and input them to a *revealer function* to produce K. For every set X, where $|X| < t$, if inputting less than t shares to the revealer function will give the users in X no advantage in

obtaining K, the threshold scheme is said to be *perfect*.

The major obstacle to using secret sharing to directly broadcast session keys is that it requires the shares of all members of the set P to be input to the revealer function. In reality, this requires either the physical presence of each user or some way of securely communicating their shares. We proceed to alter the method slightly to allow the broadcast of session keys.

Let each user U_i continue to share a long-term secret s_i with T. T computes a random session key K and will share it with a privileged set P where $|P| = t$. T proceeds to compute the following broadcast message :

$$\mathcal{B} = g(K, s_1, s_2, \ldots, s_t) \tag{8}$$

for an appropriate share-generating function g, where s_i is the shared secret of $U_i \in P$. $\mathcal{B}$ is a set of t shares distinct from the shares input to the function g, and is broadcast to all users. Just as in Section 2, each $U_i \in P$ needs only their share s_i to recover K.

Notice how this fits the framework of Section 2. If f from (1) simply returns its parameters as output, then E from (2) and g from (8) are equivalent. This suggests the use of secret sharing to essentially achieve encryption.

We are broadcasting t shares which were formed from the session key K and the shares of each of the members of P. The intent is that a perfect $(t+1, n)$ threshold scheme for sharing the secret K is implied by g, where t shares from $\mathcal{B}$ plus any one share s_i for $U_i \in P$ reaches the threshold. Recall that in a perfect threshold scheme, $(t+1) - 1$ shares reveal no information about the intended secret.

This general idea was considered by Laih *et al.* [15], and subsequently by Berkovits [1]. In the following Sections, we present one of the schemes from [1] and subsequently show how to decrease the size of the broadcast message and the amount of work required by each user to recover K, while maintaining the "perfect" nature of the threshold scheme.

4.1 Using Shamir Interpolation

Berkovits [1] uses the threshold scheme from Shamir [17] to allow a trusted center T to broadcast messages to a privileged set P as follows.[3] Each user $U_i \in U$ shares a secret point $(x_i, y_i) \in Z_p^* \times Z_p^*$ with T, where $x_i \neq x_j$ for $i \neq j$ for a prime p. To share a session key with members of P, T first selects a random $K \in Z_p^*$. T then inputs t points $(x_i, y_i), \forall i : U_i \in P$ and $(0, K)$ to g in (8). The function g will first produce a polynomial of degree t,

$$p(x) = K + a_1 x^1 + \cdots + a_t x^t \pmod{p}. \tag{9}$$

Note that $t+1$ points are required to uniquely define $p(x)$. $\mathcal{B}$ from (8) will be made up of t points in $Z_p^* \times Z_p^*$ on the polynomial, distinct from any shares of a user $U_i \in P$. Since t distinct points are both input to and output from g and $t \leq n$, we require $p \geq 2n + 1$.

For this scheme, the size of $\mathcal{B}$ is $2t \log_2 p$ bits. From $\mathcal{B}$ and his own share, each $U_i \in P$ has enough points to reconstruct $p(x)$, and compute $p(0) = K$. A user in possession of only $\mathcal{B}$ does not have enough points to reconstruct $p(x)$ and hence has no advantage in recovering K.

The reconstruction of $p(x)$ by each user is relatively expensive. Given t points, $O(t^2)$ multiplications in Z_p^* are required.

[3] We simplify the scheme somewhat for our purposes.

5 A Lower Bound

In this Section, we take the general idea for broadcasting session keys by secret sharing, as given in Section 4, and add specific conditions to define a formal model. We then prove a lower bound on the size of the broadcast message required to broadcast a session key within this model and note that the scheme given in Section 5.1 meets this bound.

Let $\mathcal{B}$ and K be the respective broadcast message and session key from Section 4 and s_i be the share of $U_i \in P$. Without loss of generality, let $\mathcal{S}_P = (s_1, \ldots, s_t)$ be the t shares of the users in P. The following are assumptions for the model. H refers the entropy of a given element, while I refers to the transinformation. Refer to Appendix A.1 for background.

A1 $H(s_i|\mathcal{B}) = H(s_i)$. The broadcast message does not decrease the uncertainty in a user's share.

A2 $H(K|\mathcal{B}) = H(K)$. The broadcast message does not decrease the uncertainty in the session key.

A3 $H(s_i|s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_t) = H(s_i)$. The share of one user is independant of the shares of other users.

A4 $H(K|\mathcal{B}s_i) = 0$. The session key is uniquely defined by the broadcast message and the share of any member of the privileged set.

A5 $H(K|\mathcal{S}_P) = H(K)$. The shares of the users alone, reveal no information about the session key.

A6 $H(\mathcal{B}|\mathcal{S}_P K) = 0$. The shares of the users in P, and the session key, define the broadcast message.

The following are two technical lemmas required for the proof of Theorem 1.

Lemma 1 $H(\mathcal{B}) = I(\mathcal{B}|\mathcal{S}_P K)$.

Proof:

$$\begin{aligned} I(\mathcal{B}|\mathcal{S}_P K) &= I(\mathcal{B}|s_1) + I_{s_1}(\mathcal{B}|s_2) + I_{s_1 s_2}(\mathcal{B}|s_3) \\ &\quad + \cdots + I_{\mathcal{S}_P}(\mathcal{B}|K) \\ &= H(\mathcal{B}) - H(\mathcal{B}|s_1) + H(\mathcal{B}|s_1) \\ &\quad - H(\mathcal{B}|s_1 s_2) + H(\mathcal{B}|s_1 s_2) \\ &\quad - H(\mathcal{B}|s_1 s_2 s_3) + \cdots + H(\mathcal{B}|\mathcal{S}_P) \\ &\quad - \underbrace{H(\mathcal{B}|\mathcal{S}_P K)}_{0 \text{ by } A6} \\ &= H(\mathcal{B}) \end{aligned}$$

□

Lemma 2 *Let $\mathcal{D} \subset P$ be a non-empty subset of privileged participants such that $|\mathcal{D}| \leq (t-1)$, and let $\mathcal{S}_\mathcal{D}$ be the set of shares of participants in $\mathcal{D}$. Also let s_i be the share of $U_i \in P$ such that $U_i \notin \mathcal{D}$. Given a session key K and broadcast message $\mathcal{B}$ from (8), $H(s_i) - H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) \geq H(K)$.*

Proof:

The term $H(s_i K|\mathcal{B}\mathcal{S}_\mathcal{D})$ simplifies to either $H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) + H(K|\mathcal{B}\mathcal{S}_\mathcal{D} s_i)$ or $H(K|\mathcal{B}\mathcal{S}_\mathcal{D}) + H(s_i|\mathcal{B}K\mathcal{S}_\mathcal{D})$ (cf. [6, Lemma 3.3]). From A4, we have

$$H(K|\mathcal{B}\mathcal{S}_\mathcal{D} s_i) = H(K|\mathcal{B}\mathcal{S}_\mathcal{D}) = 0,$$

giving

$$H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) = H(s_i|\mathcal{B}K\mathcal{S}_\mathcal{D}) \tag{10}$$

We also have,

$$\begin{aligned} I_\mathcal{B}(s_i|K) &= I_\mathcal{B}(K|s_i) \\ \overbrace{H(s_i)}^{H(s_i) \text{ by } A1} - H(s_i|\mathcal{B}K) &= \overbrace{H(K|\mathcal{B})}^{H(K) \text{ by } A2} - \overbrace{H(K|\mathcal{B}s_i)}^{0 \text{ by } A4} \\ H(s_i) - H(s_i|\mathcal{B}K) &= H(K) \end{aligned} \tag{11}$$

And,

$$I_{\mathcal{B},K}(s_i|\mathcal{S}_\mathcal{D}) = H(s_i|\mathcal{B}K) - H(s_i|\mathcal{B}K\mathcal{S}_\mathcal{D}) \geq 0$$

$$\Rightarrow H(s_i|\mathcal{B}K) \geq H(s_i|\mathcal{B}K\mathcal{S}_\mathcal{D}) = H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}). \tag{12}$$

The last equality in (12) is obtained from (10). The result follows by applying (12) to (11). □

On its own, lemma 2 implies that if the entropy (uncertainty) of a privileged user's share s_i is equal to the entropy of the session key K, then given the broadcast message, any privileged user(s) have no uncertainty in s_i, as illustrated by the following corollary.

Corollary 1 *If $H(s_i) = H(K)$, $H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) = 0$.*

Proof:

Follows from lemma 2, where $H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) \leq H(s_i) - H(K)$. □

It is well-known (see [19]) that in a perfect secret sharing scheme, the size of the shares must be at least as large as the secret key. The following corollary highlights the fact that the number of bits used to encode a user's secret must be at least as large as those used to encode the session key.

Corollary 2 $H(s_i) \geq H(K)$.

Proof:

From lemma 2, we have

$$H(s_i) \geq H(K) + H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}).$$

Since $H(s_i|\mathcal{B}\mathcal{S}_\mathcal{D}) \geq 0$, the result follows. □

The following theorem implies that for the model presented in this Section, the minimum number of bits required to encode the broadcast message $\mathcal{B}$ is at least as large as t copies of the session key K, where $|P| = t$.

Theorem 1 $H(\mathcal{B}) \geq tH(K)$.

Proof:

Recall that $\mathcal{S}_P$ is the set of t shares of users in P.

$$\begin{aligned} H(\mathcal{B}) &= I(\mathcal{B}|\mathcal{S}_P K) && (\text{ by Lemma 1 }) \\ &= I(\mathcal{B}|\mathcal{S}_P) + I_{\mathcal{S}_P}(\mathcal{B}|K) \\ &= I(\mathcal{S}_P|\mathcal{B}) + I_{\mathcal{S}_P}(K|\mathcal{B}) \\ &= H(\mathcal{S}_P) - H(\mathcal{S}_P|\mathcal{B}) + \overbrace{H(K|\mathcal{S}_P)}^{H(K) \text{ by } A5} \\ &\quad - \underbrace{H(K|\mathcal{B}\mathcal{S}_P)}_{0 \text{ by } A4} \end{aligned} \tag{13}$$

Now, $H(\mathcal{S}_P) = H(s_1) + H(s_2) + \cdots + H(s_t)$ (by A3)

$$\begin{aligned} H(\mathcal{S}_P|\mathcal{B}) &= H(s_1 s_2 \ldots s_t|\mathcal{B}) \\ &= H(s_1|\mathcal{B}) + \sum_{j=2}^{t} H(s_j|\mathcal{B}s_1 \cdots s_{j-1}) \end{aligned}$$

So, completing (13), we have

$$\begin{aligned} H(\mathcal{B}) &= \overbrace{H(s_1) - H(s_1|\mathcal{B})}^{0 \text{ by } A1} \\ &\quad + \sum_{j=2}^{t} (H(s_j) - H(s_j|\mathcal{B}s_1 \cdots s_{j-1})) + H(K) \\ &\geq (t-1)H(K) + H(K) && (\text{ by Lemma 2 }) \\ &= tH(K) \end{aligned}$$

□

From this theorem, we obtain a lower bound for the size of the broadcast message $\mathcal{B}$ from (8) in Section 4.

5.1 Meeting the Lower Bound

To improve on this technique, we employ an idea of Krawczyk [14] (used in another context). Observe that after constructing $p(x)$, user $U_i \in P$ recovers one of the $t+1$ polynomial coefficients, namely the constant, as the session key K.

Define the shares created by $\mathcal{B}$ to be the coefficients $a_1, \ldots, a_t$ of $p(x)$. To recover $p(x)$, $U_i \in P$ will create the polynomial in (9) missing only K. Using his secret share (x_i, y_i) (a point on $p(x)$), U_i can easily solve for the constant of the polynomial, K. This can be done with a simple substitution, followed by an application of Horner's rule with $O(t)$ multiplications.

This scheme both satisfies the conditions for the model of Section 2 and realizes the lower bound; it is thus optimal in this regard. Notice that the size of $\mathcal{B}$ is now $t \log_2 p$ bits where the session key K is $\log_2 p$ bits. We are able to satisfy the conditions because of the fact that the scheme is based on a "perfect" threshold scheme. In other words, $t-1$ shares reveal no additional information about the key K. We satisfy A3 for this scheme by defining the share of each user to simply be the y-component. In this way, the shares of each user are independent.

If only the secrecy of the y-component is maintained, we now have $H(s_i) = \log_2 p$ for a randomly chosen point (x_i, y_i) on $p(x)$. Given the broadcast message $\mathcal{B}$ and the point (x_j, y_j) on $p(x)$ for user $U_j \in P$, we have $H(s_i | \mathcal{B}\mathcal{S}_\mathcal{D}) = 0$, where $\mathcal{S}_\mathcal{D}$ need only consist of s_j (see Lemma 1). This results from the fact that user U_j has the ability to recover $p(x)$ and the corresponding y-component of user U_i's share, since the secrecy of the x-component is not maintained. To solve this problem and maintain the information theoretic security of the scheme, one can apply the work of Blakley *et al.* [3] that solves the same problem with secret sharing.

6 Conclusion and Open Problems

We have provided a conceptual model for true broadcasting by which a trusted center can initiate a secure point-to-multipoint communication with a set of privileged users. The center broadcasts only one message, requiring no addressing for the message to reach the intended recipients. Moreover, each user need only use a single key to decrypt the message. An important feature is that user anonymity is preserved, i.e. the identities of users in the privileged set are neither required nor revealed in the broadcast message. This feature is absent in many other schemes, yet is considered a crucial aspect in many practical applications, such as *pay-per-view*.

In the model of Section 5, the shares of the users are independent of one another. It may be possible to achieve a tighter bound if the user's shares are not independent. It would also be of interest to find additional schemes which satisfy the framework of Section 2, in particular, more practical schemes.

The model in Section 2 creates a master key K_P associated with the privileged set P. While the message that is "locked" by this master key can be opened by that same key, it can also be opened using a single key from any one of the members of P. This differs from secret sharing where a set of user's keys are jointly required to open the lock. This arrangement whereby a single user (key) suffices to recover a message has many other potential applications, including access control.

A Appendix

A.1 Information Theory Background

We recall some standard information theoretic properties that are used in Section 5. For further reference, consult [13, 18].

Given a finite set X of size n and a probability distribution $\{p(x_i)\}_{x_i \in X}$, the *entropy of* X is defined to be

$$H(X) = \sum_{i=1}^{n} p(x_i) \log_b \left(\frac{1}{p(x_i)} \right)$$

where $\log_b(\frac{1}{0})$ is defined to be 0. The subscript b refers to our base of reference. For our purposes, we will be using $b = 2$ and hence computing *bits* of entropy.

Entropy has some useful interpretations. $H(X)$ defines one's average uncertainty in X, i.e. uncertainty about which element of X has been chosen, given the probability distribution. $H(X)$ is also useful for approximating the minimum number of bits required to encode elements of X.

The range of $H(X)$ is $0 \leq H(X) \leq \log |X|$, where the lower bound is obtained when $p(x_i) = 1$ for some i, while $p(x_j) = 0, \forall j \neq i$. The upper bound is achieved when $p(x_i) = \frac{1}{|X|}, \forall i$.

The *equivocation* of X given Y is defined to be

$$H(X|Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} p(x_i) p(x_i|y_j) \log \left(\frac{1}{p(x_i|y_j)} \right)$$

where Y is a finite set of size m. $H(X|Y)$ can be thought of as the uncertainty in X, given that Y has been observed *a priori*. Note that $H(X|Y) \geq 0$.

The *mutual information* or *transinformation* of X and Y is defined as

$$I(X|Y) = H(X) - H(X|Y)$$

and can be thought of as the amount of information that Y reveals about X. Note that $I(X,Y)$ and $I(X;Y)$ are also equivalent notations. If X and Y are independent,

then $H(X|Y) = H(X)$ and $I(X|Y) = 0$. In other words, Y contributes no information about X. Similarly,

$$I(X|YZ) = H(X) - H(X|YZ),$$

for finite sets X, Y and Z. Transinformation has the properties that

$$\begin{aligned} I(X|Y) &= I(Y|X), \\ I(X|Y) &\geq 0. \end{aligned}$$

From the latter statement, note that $H(X) \geq H(X|Y)$.

The *conditional transinformation* of the pair X, Y given Z is defined as

$$I_Z(X|Y) = H(X|Z) - H(X|YZ),$$

and can be interpreted as the amount of information that Y provides about X, given that Z has already been observed. From this we obtain

$$I(X|YZ) = I(X|Y) + I_Y(X|Z).$$

References

[1] Berkovits, S., "How to Broadcast a Secret", *Advances in Cryptology: Proceedings of EUROCRYPT '91*, Springer-Verlag, 1992, pp.536-541.

[2] Blakley, G., "One-time pads are Key Safeguarding Schemes, not Cryptosystems: Fast Key Safeguarding Schemes (Threshold Schemes) Exist", *Proceedings of IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, 1980, pp.108-113.

[3] Blakley, B., Blakley, G., Chan, A., Massey, J., "Threshold Schemes With Disenrollment", *Advances in Cryptology: Proceedings of CRYPTO '92*, Springer-Verlag.

[4] Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M., "Perfectly-Secure Key Distribution for Dynamic Conferences", *Advances in Cryptology: Proceedings of CRYPTO '92*, Springer-Verlag, 1993, pp.478-493.

[5] Blundo, C., Cresti, A., "Space Requirements for Broadcast Encryption", to appear in *Advances in Cryptology: Proceedings of Eurocrypt '94*, Springer-Verlag.

[6] Capocelli, R., De Santis, A., Gargano, L., Vaccaro, U., "On the Size of Shares for Secret Sharing Schemes", *Advances in Cryptology: Proceedings of CRYPTO '91*, Springer-Verlag, 1992, pp.101-113.

[7] Chang, C.C., Hwang, S.J., "A Secure Broadcasting Scheme Based on Discrete Logarithms", *Control and Computers*, Vol.20, No.2, 1992, pp.49-53.

[8] Chiou, G.H., Chen, W.T., "Secure Broadcasting Using the Secure Lock", *IEEE Transactions on Software Engineering*, Vol.15, No.8, August 1989, pp.929-934.

[9] Fiat, A., Naor, M., "Broadcast Encryption", to appear in *Advances in Cryptology: Proceedings of CRYPTO '93*, Springer-Verlag.

[10] Fortier, M., "A Store-and-Forward Architecture for Video-on-Demand Service", presented at *Multimedia Communications '93*, Banff, Alberta, Aug.13-16, 1993, pp.262-268.

[11] Goldwasser, S., Micali, S., "Probabilistic Encryption", *Journal of Computer and System Sciences*, Vol.28, 1984, pp.270-299.

[12] Gopal, J., Jaffe, M., "Point-to-multipoint Communication over Broadcast Links", *IEEE Transactions on Communications*, COM-32(9), 1982, pp.1034-1044.

[13] Jumarie, G., *Relative Information: Theories and Applications*, Springer-Verlag, Berlin, 1990.

[14] Krawczyk, H., "Secret Sharing Made Short", to appear in *Advances in Cryptology: Proceedings of CRYPTO '93*, Springer-Verlag.

[15] Laih, C., Lee, J., Harn, L., "A New Threshold Scheme and its Applications in Designing the Conference Key Distribution Cryptosystem", *Information Processing Letters*, Vol.32, 1989, pp.95-99.

[16] Lin, C.H., Chang, C.C., Lee, R.C., "A Conference Key Broadcasting System Using Sealed Locks", *Information Systems*, Vol.17, No.4, 1992, pp.323-328.

[17] Shamir, A., "How to Share a Secret", *Communications of the ACM*, Vol.22, No.11, November 1979, pp.612-613.

[18] Shannon, C., "Communication Theory of Secrecy Systems", *Bell System Technical Journal*, Vol.28, 1949, pp.656-715.

[19] Simmons, G., "An Introduction to Shared Secret and/or Shared Control Schemes and their Application", *Contemporary Cryptology*, IEEE Press, 1991, pp.441-497.