Design of Self-Healing Key Distribution Schemes

Carlo Blundo, Paolo D'Arco, Alfredo De Santis, and Massimiliano Listo

[1] Dipartimento di Informatica ed Applicazioni
Università di Salerno, 84081 Baronissi (SA), Italy
e-mail: {carblu, paodar, ads}@dia.unisa.it
e-mail: maslis@tiscalinet.it

December 9, 2003

Abstract

A self-healing key distribution scheme enables dynamic groups of users of an *unreliable* network to establish group keys for secure communication. In such a scheme, a group manager, at the beginning of each session, in order to provide a key to each member of the group, sends packets over a broadcast channel. Every user, belonging to the group, computes the group key by using the packets and some private information. The group manager can start multiple sessions during a certain time-interval, by adding/removing users to/from the initial group. The main property of the scheme is that, if during a certain session some broadcasted packet gets lost, then users are still capable of recovering the group key for that session simply by using the packets they have received during a previous session and the packets they will receive at the beginning of a subsequent one, without requesting additional transmission from the group manager. Indeed, the only requirement that must be satisfied, in order for the user to recover the lost keys, is membership in the group both before and after the sessions in which the broadcast messages containing the keys are sent.

This novel and appealing approach to key distribution is quite suitable in certain military applications and in several Internet-related settings, where high security requirements need to be satisfied. In this paper we continue the study of self-healing key distribution schemes, introduced by Staddon et al. in [37]. We analyse some existing constructions: we show an attack that can be applied to one of these constructions, in order to recover session keys, and two problems in another construction. Then, we present a new mechanism for implementing the self-healing approach, and we present an efficient construction which is optimal in terms of user memory storage. Finally, we extend the self-healing approach to key distribution, and we present a scheme which enables a user to recover from a single broadcast message all keys associated with sessions in which he is member of the communication group.

Keywords: Group Communication, Key Distribution, Self-Healing.

1 Introduction

Self-Healing Key Distribution. How to distribute session keys for secure communication to groups of users of a network, in a manner that is resistant to packet loss, is an issue that has not been addressed in-depth in the past. Indeed, the greatest part of the literature assumes an underlying reliable network. Recently, in [37], an interesting approach to deal with this scenario has been proposed. A *self-healing key distribution scheme* [37] enables a dynamic group of users to establish a group key over an unreliable network. In such a scheme, a group manager, at the beginning of each session, in order to provide a key to each member of the group, sends packets over a broadcast channel. Every user, belonging to the group, computes the group key by using the packets and some private information. The group manager can start multiple sessions during a certain time-interval, by adding/removing users to/from the initial group. The main property of the scheme is that, if at the beginning of a certain session some broadcasted packet gets lost, then users are still capable of recovering the group key for that session simply by using the packets they have received at the beginning of a previous session and the packets they will receive at the beginning of a subsequent one, without requesting additional transmission from the group manager. Indeed, the only requirement that must be satisfied, in order for the user to recover the lost keys, is membership in the group both before and after the sessions in which the broadcast messages containing the key are sent. In other words, the user can recover lost keys associated with sessions "sandwiched" between two sessions in which the user is member of the group and correctly receives the broadcast messages.

The benefits of such an approach basically are: reduction of network traffic, reduction of the work load on the group manager, and a lower risk of user exposure through traffic analysis.

Applications. Several settings in which session keys need to be used for a short time-period, in order to reduce the amount of ciphertext available to an adversary, or need to be updated, due to frequent changes in the group structure, can profictably use such schemes. Military-oriented applications [32] as well as Internet applications [37] (e.g., broadcast transmissions, pay-per-view TV, information services, et cetera) are few important examples. In a battle field, communication is often carried over a wireless and unreliable network (tactical ad-hoc network), where user mobile devices are powered by batteries: once a battery is off, or a user mobile device is caught by the enemy, the device must be removed from the group; as well as it can rejoin the group once the power is on again or the user device has been recovered. Moreover, due to the hierarchical structure of military groups, some information might need to reach only a certain subsets of the whole communication group. Hence, several such restricted subgroups of the main communication group can be active during the same time interval (i.e., multiple sessions), and some users (e.g., troop officials) can belong to all subgroups. Finally, there could be a need for adding new devices at any time. In such a setting, the group manager can start in a certain time-interval several secure communication sessions, and it is necessary to have a flexible mechanism to distribute cryptographic keys to the recipients.

A more useful and comfortable setting is broadcast communication over low-cost channels: live-event transmissions (e.g., concerts, formal cerimonies, ice-hockey games, ...) for users who have subscribed to (and paid for) the service, can benefit from such schemes too. Electronic services delivering sensitive content/information to authorized recipients can take advantage from self-healing key distribution schemes as well.

Previous work. Broadcast Encryption is one of the closest area to the subject of this paper. Originated in [2], and formally defined in [19], it has been extensively studied (e.g., [3, 8, 22, 39, 29, 40]), and it has grown up in different directions: mainly, re-keying schemes for *dynamic* groups of users (see, [44, 10, 11, 35, 15] to name a few), and broadcast schemes with tracing capability for dishonest users [12, 34, 16, 20, 41, 42, 43, 38, 21, 36, 24, 25]. Moreover, several papers have addressed the special case of *users revocation* from a privileged subset [26, 1, 31, 30, 23].

However, all the above papers assume that the underlying network is reliable. The authors of [33] and [45], have considered a setting in which packets can get lost during transmission. In the first case, error correction techniques have been employed. In the second, short hint messages have been appended to the packets. The schemes given in [26], by accurately choosing the values of the parameters, can provide resistance to packet loss as well. Recently, in [37, 27] the issue of packet loss due to the presence of an unreliable network has been addressed, and the key recovery approach pursued in both papers is quite similar: each packet enables the user to recover the current key and a share of previous and subsequent ones. In [14] also this problem is considered. The paper generalises several known constructions in order to gain resistance to packet loss. Finally, in [32], the schemes given in [37] have been improved in terms of both memory storage and communication complexity.

Our Contribution. In this paper we continue the study of self-healing key distribution schemes. We analyse some existing constructions: we show an attack that can be applied to one of these constructions, in order to recover session keys, and two problems in another construction. The first problem implies that group members can be excluded from the communication group in presence of packet loss. The second one gives to a user who joins the group the possibility of recovering past session keys, associated to groups in which she does not belong to. Then, we present a new mechanism for implementing the self-healing property, and we present an efficient construction which is optimal in terms of user memory storage. Finally, we extend the self-healing approach, and we present a scheme which enables a user to recover from a single broadcast message all keys associated with sessions in which he is member of the communication group.

2 Background

In this section we briefly recall some basic notions of Information Theory [13]. Indeed, in the following section, we will use the entropy function to state the properties that self-healing key distribution schemes have to satisfy.

A discrete random experiment is defined by a finite set, called *sample space*, consisting of all elementary events, and a *probability measure* assigning a non-negative real number to every elementary event, such that the sum of all these probabilities is equal to 1. An *event* of a discrete random experiment is a subset of the sample space, and the probability assigned to it is the sum of the probabilities of its elementary events.

A *discrete random variable* $\mathbf{X}$ is a mapping from a sample space to a certain range X, and is characterized by its probability distribution $\{P_{\mathbf{X}}(x)\}_{x\in X}$ that assigns to every $x \in X$ the probability $P_{\mathbf{X}}(x)$ of the event that $\mathbf{X}$ takes on the value x.

The *entropy* of $\mathbf{X}$, denoted by $H(\mathbf{X})$, is a real number that measures the uncertainty about the value of $\mathbf{X}$ when the underlying random experiment is carried out. It is defined by

$$H(\mathbf{X}) = -\sum_{x\in X} P_{\mathbf{X}}(x)\log P_{\mathbf{X}}(x),$$

assuming that the terms of the form $0\log 0$ are excluded from the summation, and where the logarithm is relative to the base 2. The entropy satisfies $0 \leq H(\mathbf{X}) \leq \log|X|$, where $H(\mathbf{X}) = 0$ if and only if there exists $x_0 \in X$ such that $Pr(\mathbf{X} = x_0) = 1$; whereas, $H(\mathbf{X}) = \log|X|$ if and only if $Pr(\mathbf{X} = x) = 1/|X|$, for all $x \in X$. The deviation of the entropy $H(\mathbf{X})$ from its maximal value can be used as a measure of non-uniformity of the distribution $\{P_{\mathbf{X}}(x)\}_{x\in X}$.

Given two random variables $\mathbf{X}$ and $\mathbf{Y}$, taking values on sets X and Y, respectively, according to a probability distribution $\{P_{\mathbf{XY}}(x,y)\}_{x\in X, y\in Y}$ on their Cartesian product, the conditional uncertainty of $\mathbf{X}$, given the random variable $\mathbf{Y}$, called *conditional entropy* and denoted by $H(\mathbf{X}|\mathbf{Y})$, is defined as

$$H(\mathbf{X}|\mathbf{Y}) = -\sum_{y\in Y}\sum_{x\in X} P_{\mathbf{Y}}(y)P_{\mathbf{X}|\mathbf{Y}}(x|y)\log P_{\mathbf{X}|\mathbf{Y}}(x|y).$$

Notice that the conditional entropy is not the entropy of a probability distribution but the *average* over all entropy $H(\mathbf{X}|\mathbf{Y} = y)$. Simple algebra shows that

$$H(\mathbf{X}|\mathbf{Y}) \geq 0 \tag{1}$$

with equality if and only if X is a function of Y.

The *mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ is a measure of the amount of information by which the uncertainty about $\mathbf{X}$ is reduced by learning $\mathbf{Y}$, and viceversa. It is given by

$$I(\mathbf{X};\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X}).$$

Since,

$$I(\mathbf{X};\mathbf{Y}) = I(\mathbf{Y};\mathbf{X}) \text{ and } I(\mathbf{X};\mathbf{Y}) \geq 0, \tag{2}$$

it is easy to see that

$$H(\mathbf{X}) \geq H(\mathbf{X}|\mathbf{Y}), \tag{3}$$

with equality if and only if $\mathbf{X}$ and $\mathbf{Y}$ are independent. Along the same line, given three random variables, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, the *conditional mutual information* between $\mathbf{X}$ and $\mathbf{Y}$ given $\mathbf{Z}$ can be written as

$$\begin{aligned} I(\mathbf{X};\mathbf{Y}|\mathbf{Z}) &= H(\mathbf{X}|\mathbf{Z}) - H(\mathbf{X}|\mathbf{Z}\ \mathbf{Y}) \\ &= H(\mathbf{Y}|\mathbf{Z}) - H(\mathbf{Y}|\mathbf{Z}\ \mathbf{X}) = I(\mathbf{Y};\mathbf{X}|\mathbf{Z}). \end{aligned} \tag{4}$$

Since the conditional mutual information $I(\mathbf{X};\mathbf{Y}|\mathbf{Z})$ is always non-negative, it holds that

$$H(\mathbf{X}|\mathbf{Z}) \geq H(\mathbf{X}|\mathbf{Z}\ \mathbf{Y}). \tag{5}$$

A useful equality, widely applied in information-theoretic proofs, is given by the so-called *chain rule*. It is stated as follows: given n random variables, $\mathbf{X}_1 \dots \mathbf{X}_n$, the entropy of $\mathbf{X}_1 \dots \mathbf{X}_n$ can be written as

$$H(\mathbf{X}_1, \dots, \mathbf{X}_n) = H(\mathbf{X}_1) + H(\mathbf{X}_2|\mathbf{X}_1) + \cdots + H(\mathbf{X}_n|\mathbf{X}_1 \dots \mathbf{X}_{n-1}). \tag{6}$$

3 The Model

The model we consider in this paper is a slightly modified version of the one given in [37].

Let $\mathcal{U}$ be the finite universe of users of a network. A broadcast unreliable channel is available, and time is defined by a global clock. Let GM be a group manager who sets up and manages, by means of join and revoke operations, a communication group, which is a dynamic subset of users of $\mathcal{U}$. Let $G_j \subseteq \mathcal{U}$ be the communication group established by GM in session j. Each user $U_i \in G_j$ holds a personal key S_i, received from GM before or when joining G_j. A personal key S_i can be seen as a sequence of elements from a finite set, and can be used as long as user U_i is not removed by GM from the group. Individual personal keys can be related.

We denote the number of sessions, supported by the scheme, by m, the set of users revoked by GM in session j by R_j, and the set of users who join the group in session j by $Join_j$. Hence, $G_j = (G_{j-1} \cup Join_j) \setminus R_j$. Moreover, for $j = 1, \dots, m$, let K_j be the session key chosen by GM and communicated to the group members through a broadcast message, B_j. For each $U_i \in G_j$, the key K_j is determined by B_j and the personal key S_i.

Let $\mathbf{S}_i, \mathbf{B}_j, \mathbf{K}_j$ be random variables representing the personal key for user U_i, the broadcast message B_j and the session key K_j for session j, respectively. The probability distributions according to whom the above random variables take values are determined by the key distribution scheme and the random bits used by GM. In particular, we assume that session keys K_j are chosen independently and according to the uniform distribution.

Then, using the entropy function, we state the following definition[1]:

[1] In order to simplify the notation, for any subset of users $G = \{U_{i_1}, \dots, U_{i_g}\} \subseteq \mathcal{U}$, where $i_1 < i_2 < \dots < i_g$, we will denote the random variables $\mathbf{X}_{i_1} \dots \mathbf{X}_{i_g}$ by means of $\mathbf{X}_G$.

Definition 3.1 *Let $\mathcal{U}$ be the universe of users of a network, let m be the maximum number of sessions, and let t be the maximum number of users that can be revoked by GM.*

1. *$\mathcal{D}(t, m, \mathcal{U})$ is a session key distribution scheme if the following are true:*

   (a) *For each member $U_i \in G_j$, the key K_j is determined by $\mathbf{B}_j$ and $\mathbf{S}_i$.* Formally, it holds that:
   $$H(\mathbf{K}_j | \mathbf{B}_j, \mathbf{S}_i) = 0.$$

   (b) *What users learn from the broadcast B_j and their own personal key cannot be determined from the broadcast or personal keys alone.* Formally, it holds that:
   $$\begin{aligned} H(\mathbf{K}_1, \ldots, \mathbf{K}_m | \mathbf{B}_1, \ldots, \mathbf{B}_m) &= H(\mathbf{K}_1, \ldots, \mathbf{K}_m | \mathbf{S}_1, \ldots, \mathbf{S}_n) \\ &= H(\mathbf{K}_1, \ldots, \mathbf{K}_m), \end{aligned}$$
   where $\mathbf{S}_1, \ldots, \mathbf{S}_n$ denote the personal keys of users $U_1, \ldots, U_n$ in $G_1 \cup \ldots \cup G_m$.

2. *$\mathcal{D}(t, m, \mathcal{U})$ has t-revocation capability if, for each session j, let $R = R_j \cup R_{j-1} \cup \ldots \cup R_1$ such that $|R| \leq t$, the group manager GM can generate a broadcast message B_j such that all revoked users in R cannot recover K_j.* Formally, it holds that:
   $$H(\mathbf{K}_j | \mathbf{B}_j, \mathbf{B}_{j-1}, \ldots, \mathbf{B}_1 \mathbf{S}_R) = H(\mathbf{K}_j),$$
   where $\mathbf{S}_R$ denotes the personal keys of all users in R.

3. *$\mathcal{D}(t, m, \mathcal{U})$ is self-healing if the following properties are satisfied:*

   (a) *Every $U_i \in G_r$, not revoked before session s, from broadcasts B_r and B_s, where $1 \leq r < s \leq m$, can recover all keys K_ℓ, for $\ell = r, \ldots, s$.* Formally, it holds that:
   $$H(\mathbf{K}_r, \ldots, \mathbf{K}_s | \mathbf{S}_i, \mathbf{B}_r, \mathbf{B}_s) = 0.$$

   (b) *Let $B \subseteq R_r \cup R_{r-1} \cup \ldots \cup R_1$ be a coalition of users removed before session r and let $C \subseteq Join_s \cup Join_{s+1} \cup \ldots \cup Join_m$ be a coalition of users who join the group from session s. Let $|B \cup C| \leq t$. Then, such a coalition does not get any information about keys K_j, for any $r \leq j < s$.* Formally, it holds that:
   $$H(\mathbf{K}_r, \ldots, \mathbf{K}_{s-1} | \mathbf{B}_1, \ldots, \mathbf{B}_m, \mathbf{S}_B, \mathbf{S}_C) = H(\mathbf{K}_r, \ldots, \mathbf{K}_{s-1}).$$
   where $\mathbf{S}_B$ denotes the personal keys of users in B, and $\mathbf{S}_C$ denotes the personal keys of users in C.

The definition is divided in three parts: the first one states the conditions that must be satisfied in a session key distribution scheme. The second and the third parts define the additional t-revocation and self-healing properties. More precisely, condition *(a)* of point 1. establishes that every user U_i, belonging to the communication group during session j, can compute K_j, using only his own personal key S_i and the broadcast message B_j. Condition *(b)* essentially establishes that neither the full sequence of broadcast messages nor the personal keys of all users alone, give information about session keys.

Point 2. formally defines the t-revocation capability of the scheme: the property establishes that, for any subgroup of at most t users that must be revoked, the group manager can broadcast a message enabling non-revoked users to compute a new session key, while revoked users do not gain any information about such a new key.

Point 3. characterizes the self-healing approach. Condition *(a)* simply states that a user belonging to the group during sessions r and s, where $r < s$, can recover every lost key K_j, sent by the group manager during session j, for $r < j < s$. On the other hand, condition *(b)* establishes a security requirement that must be satisfied. If B denotes a subset of users belonging to the group until session r and C denotes a subset of users who joined the group in session s, and $|B \cup C| \leq t$ then, by pooling together all information the two subsets of users received when they were members of the group, they do not gain any information about keys K_j, used in sessions in which they were not member of the group, i.e., sessions $r \leq j < s$.

Our model is slightly different from [37]. First of all, we have removed from the definition the following condition:

> *For any subset $F \subseteq \mathcal{U}$, such that $|F| \leq t$, and for each $U_i \notin F$, the users in F cannot determine anything about S_i.* Formally, it holds that:
>
> $$H(\mathbf{S}_i|\mathbf{S}_F, \mathbf{B}_1, \ldots, \mathbf{B}_m) = H(\mathbf{S}_i),$$
>
> where $\mathbf{S}_F$ denotes the personal keys of all users in F.

It has recently been pointed out in [32] that the schemes given in [37] do not meet the above requirement. Also the construction given in [4] does not meet such a condition. In [6], by using information theoretic arguments, it is shown that the above condition is impossible to obtain.

Then, in the model given in [37], a random variable $\mathbf{Z}_{i,j}$ is used for representing the total amount of information that user $U_i \in G_j$ gets from a broadcast message B_j and his own personal key S_i (amount of information that can be greater than the key K_j). By using such a variable point 1.(a) of the definition, for example, is therein stated by saying that $H(\mathbf{Z}_{i,j}|\mathbf{B}_j, \mathbf{S}_i) = 0$, and $H(\mathbf{K}_j|\mathbf{Z}_{i,j}) = 0$. We have preferred to give a simplified formalization of the conditions by focusing directly on the secret keys.

Points 1.(b), 3.(a), 3.(b), and 2. are stronger in our model. Indeed, we have expressed conditions 1.(b), 3.(a) and 3.(b), in terms of the *joint* entropy of the keys instead of considering a single key (e.g., $H(\mathbf{K}_1, \ldots, \mathbf{K}_m|\mathbf{S}_1, \ldots, \mathbf{S}_n) = H(\mathbf{K}_1, \ldots, \mathbf{K}_m)$ instead of $H(\mathbf{K}_i|\mathbf{S}_1, \ldots, \mathbf{S}_n) = H(\mathbf{K}_i)$, for $i = 1, \ldots, m$); while, in condition 2., we have required that revoked users do not get any information on a new key even if they pool together their personal keys and previous broadcast messages. In [37] broadcast messages are not considered in condition 2. However, all constructions therein given satisfy such stronger requirements.

The constructions we will consider in the rest of the paper are implemented over a finite *prime* field F_q. Hence, group keys, broadcast messages, and personal keys will be sequences of elements in F_q, and all operations of the schemes will take place in F_q.

In the above model, the group manager can change the structure of the initial group of users by means of revoke and join operations. However, this *does not* mean that the scheme is dynamic in the sense that the group manager can provide a session key to *any* group of users at *any* time. Indeed, according to point 2., the users that can be permanently revoked from the communication group are at most t for all lifetime of the scheme. Moreover, the model does not fix any upperbound on the number of new users that can join the system but in all constructions we will see the number of join operations depends on the size of the finite prime field F_q. The bigger the size, the more users can join the scheme. However, the drawback is that memory storage and communication complexity rise up proportionally to the size of F_q. If an upper bound g on the size of G_j, for $j = 1, \ldots, m$, is known, and for security reason a group key must be at least 80-bit long, then F_q must be such that $q > \max\{2^{80}, g\}$.

4 A Construction and an Attack

In this section we analyze the self-healing key distribution schemes proposed in [37]. We start by recalling the first construction therein given. The aim is twofold: to offer to the reader a real example of a method implementing a self-healing key distribution scheme, and to point out the difficulty the design of these schemes gives rise to. Indeed, even if the following construction (which does not provide user revocation) satisfies several conditions of the definition of a self-healing key distribution scheme, we show that, for $j = 2, \ldots, m-1$, an adversary who gets the sequence of broadcasts B_{j-1}, B_j, B_{j+1}, can recover K_j.

The idea of the following construction is the distribution, with each broadcast, of shares of past and future session keys. More precisely, a user U_i, who belongs to the group communication in session ℓ, recovers from the broadcast message B_ℓ the session key K_ℓ but also shares of past and future session keys. Hence, two broadcast messages for sessions ℓ and r enable recovering keys for "sandwiched" sessions $r < j < \ell$. The shares of past session keys are implemented by means of a sequence of polynomials $< p_1(x), \ldots, p_m(x) >$; while, the shares of future session keys are implemented by $< q_1(x), \ldots, q_m(x) >$.

Let $F_q[x]$ be the set of all univariate polynomials with coefficients in F_q, and let t be a positive integer.

Construction 1 (given in [37]):

Assume that $G_1 = \{U_1, \ldots, U_n\}$.

Set-up

The group manager:

- chooses uniformly at random $2m$ polynomials in $F_q[x]$, each of degree t, say $h_1(x), \ldots, h_m(x), p_1(x), \ldots, p_m(x)$, and m session keys, $K_1, \ldots, K_m \in F_q$
- defines, for each $j = 1, \ldots, m$, a polynomial in $F_q[x]$, say $q_j(x) = K_j - p_j(x)$
- sends in a private way to user U_i, for $i = 1, \ldots, n$, as personal key the sequence of values $S_i =< h_1(i), \ldots, h_m(i) >$.

Broadcast

In session $j \in \{1, \ldots, m\}$, the group manager broadcasts

$$\begin{aligned} B_j \;=\; & < h_1(x) + p_1(x), \ldots, h_{j-1}(x) + p_{j-1}(x), h_j(x) + K_j, \\ & h_{j+1}(x) + q_{j+1}(x), \ldots, h_m(x) + q_m(x) >, \end{aligned}$$

where each term of B_j represents a *single* polynomial obtained by applying the sum operator.

Session Key and Shares Recovery in Session j:

For all $i \in \{1, \ldots, n\}$, user U_i from broadcast B_j:

- recovers K_j by evaluating $h_j(x) + K_j$ at i and by subtracting $h_j(i)$
- recovers session key shares $< p_1(i), \ldots, p_{j-1}(i) >$ evaluating, for $\ell = 1, \ldots, j-1$, the polynomials $h_\ell(x) + p_\ell(x)$ at i and subtracting $h_\ell(i)$
- recovers session key shares $< q_{j+1}(i), \ldots, q_m(i) >$ evaluating, for $\ell = j+1, \ldots, m$, the polynomials $h_\ell(x) + q_\ell(x)$ at i and subtracting $h_\ell(i)$.

The self-healing mechanism works as follows: from broadcast B_ℓ user U_i recovers, among others, shares $q_{\ell+1}(i), \ldots, q_m(i)$ of future sessions j, for $j = \ell+1, \ldots, m$. Then, in session r, from the broadcast B_r, she gets K_r as well as, among others, shares of past sessions $p_1(i), \ldots, p_{r-1}(i)$. Therefore, for $\ell < j < r$, she can compute $K_j = p_j(i) + q_j(i)$.

Unfortunately, the structure of the broadcast message in the above protocol allows an adversary who gets the sequence of broadcasts $B_1, \ldots, B_m$, to recover K_j, for any $j = 2, \ldots, m-1$.

Attack. An adversary can recover a session key as follows: if the adversary has received B_{j-1}, B_j, and B_{j+1}, then he has got $h_j(x) + q_j(x)$, $h_j(x) + K_j$, and $h_j(x) + p_j(x)$, respectively. But,

$$2(h_j(x) + K_j) - [(h_j(x) + q_j(x)) + (h_j(x) + p_j(x))] = K_j,$$

since $p_j(x) + q_j(x) = K_j$.

The attack points out that with such an approach of using redundance in each broadcast, in order to provide the possibility of recovering lost keys, the design of the scheme must carefully consider the composition of different pieces of the broadcast messages.

The attack does not apply to Constructions 3, 4 and 5 of [37], due to the use in those schemes of unrelated polynomials in each broadcast. In other words, in session j a sequence of (different) polynomials $s_{j,1}(), \ldots, s_{j,m}()$ is used for masking the session keys and the shares of past and future session keys.

However, in the above construction, if the structure of the broadcast is opportunely modified, it is possible render useless the attack described before. More precisely, let

$$\begin{aligned} B_j \quad = \quad & < h_1(x) + p_1(x), \ldots, h_{j-1}(x) + p_{j-1}(x), 2 \cdot h_j(x) + K_j, \\ & h_{j+1}(x) + q_{j+1}(x), \ldots, h_m(x) + q_m(x) > . \end{aligned}$$

In this case it is not difficult to see that a straightforward application of the proposed attack does not work since

$$2h_j(x) + K_j - [(h_j(x) + q_j(x)) + (h_j(x) + p_j(x))] = 0.$$

More precisely, we can show the following result[2]:

Theorem 4.1 *Let F_q be a finite prime field where $q > 2$, and assume that the modified* Construction 1 *is implemented over such a field. An adversary, once received $B_1, \ldots, B_m$, does not learn any information about $K_1, \ldots, K_m$.*

Proof. To simplify the discussion we can assume, without loss of generality[3], that in Construction 1 the polynomials $h_j(x), p_j(x)$ and $q_j(x)$ are simple constants h_j, p_j and q_j.

The information that can be recovered from $B_1, \ldots, B_m$ is given by $C_1 = 2 \cdot h_1 + K_1, \ldots, C_m = 2 \cdot h_m + K_m, P_1 = h_1 + p_1, \ldots, P_{m-1} = h_{m-1} + p_{m-1}, Q_2 = h_2 + q_2, \ldots, Q_m = h_m + q_m$. It is pretty easy to see that the available values do not enable us to infer any information about the m session keys. Indeed, we can show that the following system of $3m$ equations and $3m$ variables $p_1, \ldots, p_m, h_1, \ldots, h_m, q_1, \ldots, q_m$, for any fixed m-tuple of values for $K_1, \ldots, K_m$, has one and only one solution. More precisely,

[2]Notice that, instead of 2, we could use any other field element r different from 0 and 1, i.e., changing the broadcast message with $r \cdot h_j(x) + K_j$.

[3]In the general case with t-degree polynomials, we can apply the same proof technique exactly t times considering, for $i = 0, \ldots, t-1$, the systems of equations given by the coefficients of the polynomials for each term x^i.

$$\left\{\begin{array}{l} h_1 + p_1 = P_1 \\ \ldots \\ h_{m-1} + p_{m-1} = P_{m-1} \\ 2 \cdot h_1 + K_1 = C_1 \\ \ldots \\ 2 \cdot h_m + K_m = C_m \\ h_2 + q_2 = Q_2 \\ \ldots \\ h_m + q_m = Q_m \\ p_1 + q_1 = K_1 \\ p_m + q_m = K_m \end{array}\right. \quad \text{is equivalent to} \quad \left\{\begin{array}{l} p_1 = P_1 - h_1 \\ \ldots \\ p_{m-1} = P_{m-1} - h_{m-1} \\ h_1 = (C_1 - K_1) \cdot 2^{-1} \\ \ldots \\ h_m = (C_m - K_m) \cdot 2^{-1} \\ q_2 = Q_2 - h_2 \\ \ldots \\ q_m = Q_m - h_m \\ q_1 = K_1 - p_1 \\ p_m = K_m - q_m \end{array}\right.$$

and has solution given by

$$\left\{\begin{array}{l} p_1 = P_1 - (C_1 - K_1) \cdot 2^{-1} \\ \ldots \\ p_{m-1} = P_{m-1} - (C_{m-1} - K_{m-1}) \cdot 2^{-1} \\ h_1 = (C_1 - K_1) \cdot 2^{-1} \\ \ldots \\ h_m = (C_m - K_m) \cdot 2^{-1} \\ q_2 = Q_2 - (C_2 - K_2) \cdot 2^{-1} \\ \ldots \\ q_m = Q_m - (C_m - K_m) \cdot 2^{-1} \\ q_1 = K_1 - [P_1 - (C_1 - K_1) \cdot 2^{-1}]. \\ p_m = K_m - [Q_m - (C_m - K_m) \cdot 2^{-1}]. \end{array}\right.$$

Thus, an adversary holding the sequence $B_1, \ldots, B_m$, does not learn *any information about the whole sequence* $K_1, \ldots, K_m$. ■

Notice that in the original construction the above property is not satisfied because some sequences $< K_1, \ldots, K_m >$ are not possible given the sequence of broadcast messages $< B_1, \ldots, B_m >$.

5 Lower Bounds

The size of the personal key, each user has to store, and the size of the broadcast the GM has to send at the beginning of every session, in order to establish a new group key, can be lower bounded by using Information Theory Tools. We start by recalling the following simple lemma (a slightly more general version is given in [37]):

Lemma 5.1 *Let* $\mathbf{X}$, $\mathbf{Y}$, *and* $\mathbf{W}$ *be three random variables. If* $H(\mathbf{X}|\mathbf{Y},\mathbf{W}) = 0$ *and* $H(\mathbf{X}|\mathbf{W}) = H(\mathbf{X})$, *then*

$$H(\mathbf{Y}) \geq H(\mathbf{X}).$$

Proof. Notice that $I(\mathbf{Y};\mathbf{X}|\mathbf{W})$ can be written as:

$$H(\mathbf{Y}|\mathbf{W}) - H(\mathbf{Y}|\mathbf{W},\mathbf{X}) = H(\mathbf{X}|\mathbf{W}) - H(\mathbf{X}|\mathbf{W},\mathbf{Y}).$$

From (1) we get that $H(\mathbf{Y}|\mathbf{W},\mathbf{X}) \geq 0$, while from the hypothesis we have that $H(\mathbf{X}|\mathbf{W}) = H(\mathbf{X})$ and $H(\mathbf{X}|\mathbf{W},\mathbf{Y}) = 0$. Therefore, applying (5), we get that

$$H(\mathbf{Y}) \geq H(\mathbf{Y}|\mathbf{W}) \geq H(\mathbf{X}).$$

Hence, the result holds. ■

Since all session keys are chosen uniformly at random, in the following we denote by $H(\mathbf{K})$ the entropy of a random variable $\mathbf{K}$, assuming values over a finite set K according to the uniform probability distribution.

Using the above lemma, we can show the following theorem (which gives the same result proven for the model in [37]):

Theorem 5.2 *In any self-healing key distribution scheme, for any U_i belonging to the group since session j, where $j \in \{1, \ldots, m\}$, it holds that*

$$H(\mathbf{S}_i) \geq H(\mathbf{K}_j, \ldots, \mathbf{K}_m) \geq (m - j + 1) \log q.$$

Proof. Since condition 3.(a) implies that $H(\mathbf{K}_j, \ldots, \mathbf{K}_m | \mathbf{B}_j, \mathbf{B}_m, \mathbf{S}_i) = 0$ and condition 1.(c) that $H(\mathbf{K}_j, \ldots, \mathbf{K}_m | \mathbf{B}_1, \mathbf{B}_m) = H(\mathbf{K}_j, \ldots, \mathbf{K}_m)$, from Lemma 5.1 we get that $H(\mathbf{S}_i) \geq H(\mathbf{K}_j, \ldots, \mathbf{K}_m)$. From the independence of $K_j, \ldots, K_m$, and the assumption that all keys are chosen uniformly at random, it holds that

$$\begin{aligned} H(\mathbf{K}_j, \ldots, \mathbf{K}_m) &= H(\mathbf{K}_j) + \ldots + H(\mathbf{K}_m) = (m - j + 1) H(\mathbf{K}) \\ &= (m - j + 1) \log q. \end{aligned}$$

■

Notice that, since for any random variable $\mathbf{X}$ it holds that $H(\mathbf{X}) \leq \log |X|$, we get that $\log |S_i| \geq (m - j + 1) \log q$. The above inequality says that every user who belongs to G_1 has to store a personal key of at least $m \log q$ bits.

Using close techniques, it is not difficult to show a lower bound on the size of the broadcast. We start by recalling another simple lemma.

Lemma 5.3 *Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{W}$ be three random variables. If $H(\mathbf{Y}|\mathbf{W}) = 0$ then*

$$H(\mathbf{X}|\mathbf{Y}, \mathbf{W}) = H(\mathbf{X}|\mathbf{W}).$$

Proof. Notice that $H(\mathbf{Y}|\mathbf{W}) = 0$ implies $H(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = 0$. Indeed, (1) and (5) yield

$$0 \leq H(\mathbf{Y}|\mathbf{X}, \mathbf{W}) \leq H(\mathbf{Y}|\mathbf{W}) = 0.$$

The mutual information $I(\mathbf{X}; \mathbf{Y}|\mathbf{W})$ can be written either as $H(\mathbf{X}|\mathbf{W}) - H(\mathbf{X}|\mathbf{W}, \mathbf{Y})$ or as $H(\mathbf{Y}|\mathbf{W}) - H(\mathbf{Y}|\mathbf{X}, \mathbf{W})$. Since $H(\mathbf{Y}|\mathbf{W}) - H(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = 0$, it holds that $H(\mathbf{X}|\mathbf{W}) = H(\mathbf{X}|\mathbf{Y}, \mathbf{W})$.

■

Theorem 5.4 *In any self-healing key distribution scheme, for any $j = 2, \ldots, m$,*

$$H(\mathbf{B}_j) \geq (j - 1) \log q.$$

Proof. Notice that

$$\begin{aligned} I(\mathbf{B}_j; \mathbf{K}_1, \ldots, \mathbf{K}_j | \mathbf{S}_i \mathbf{B}_1) &= H(\mathbf{B}_j | \mathbf{S}_i \mathbf{B}_1) - H(\mathbf{B}_j | \mathbf{S}_i, \mathbf{B}_1, \mathbf{K}_1, \ldots, \mathbf{K}_j) \\ &= H(\mathbf{K}_1, \ldots, \mathbf{K}_j | \mathbf{S}_i, \mathbf{B}_1) - H(\mathbf{K}_1, \ldots, \mathbf{K}_j | \mathbf{S}_i, \mathbf{B}_1, \mathbf{B}_j). \end{aligned}$$

Applying the chain rule (6), we have that

$$H(\mathbf{K}_1, \ldots, \mathbf{K}_j | \mathbf{S}_i, \mathbf{B}_1) = H(\mathbf{K}_1 | \mathbf{S}_i, \mathbf{B}_1) + H(\mathbf{K}_2, \ldots, \mathbf{K}_j | \mathbf{S}_i \mathbf{B}_1, \mathbf{K}_1).$$

Condition 1.(a) implies that $H(\mathbf{K}_1|\mathbf{S}_i, \mathbf{B}_1) = 0$, while Lemma 5.3 and condition 1.(b) yield

$$H(\mathbf{K}_2, \ldots, \mathbf{K}_j|\mathbf{S}_i, \mathbf{B}_1, \mathbf{K}_1) = H(\mathbf{K}_2, \ldots, \mathbf{K}_j|\mathbf{S}_i, \mathbf{B}_1) = H(\mathbf{K}_2, \ldots, \mathbf{K}_j).$$

Moreover, due to condition 3.(a), it holds that

$$H(\mathbf{K}_1, \ldots, \mathbf{K}_j|\mathbf{S}_i, \mathbf{B}_1, \mathbf{B}_j) = 0.$$

Hence,

$$\begin{aligned} H(\mathbf{B}_j) &\geq H(\mathbf{B}_j|\mathbf{S}_i, \mathbf{B}_1) - H(\mathbf{B}_j|\mathbf{S}_i, \mathbf{B}_1, \mathbf{K}_1, \ldots, \mathbf{K}_j) \\ &\geq H(\mathbf{K}_2, \ldots, \mathbf{K}_j) = (j-1)\log q. \end{aligned}$$

■

By applying a similar argument, it is possible also to show that $H(\mathbf{B}_1) \geq H(\mathbf{K}) = \log q$.

6 A New Construction

In this section we describe some new self-healing key distribution schemes. The novelty of these schemes, compared to the ones given in [37], lies in a *different* self-healing technique. Instead of sending, with every broadcast message, shares for recovering past and future session keys, we send *additional* information which enables the users to recover lost keys from two received broadcast messages.

The first construction we propose does not support user revocation and collusion attacks, but these features can be easily built upon this basic scheme. We start by giving such a simplified scheme, in order to emphasize the self-healing technique we have employed.

Scheme 1.

Assume that $G_1 = \{U_1, \ldots, U_n\}$.

Set-up

The group manager:

- chooses uniformly at random m values, say $h_1, \ldots, h_m$, and m session keys, $K_1, \ldots, K_m$ in F_q
- defines, for each $j = 1, \ldots, m$, the value $z_j = h_j + K_j$
- sends in a private way, for $i = 1, \ldots, n$, to user U_i as personal key the sequence of values $S_i =< h_1, \ldots, h_m >$

Broadcast

The group manager GM, in sessions 1 and 2 broadcasts $B_1 = z_1$ and $B_2 = z_2$, respectively. In session $j \in \{3, \ldots, m\}$, he broadcasts

$$B_j =< z_1 + z_2, \ldots, z_1 + z_{j-1}, z_j > .$$

Key Computation in Session j

For all $i \in \{1, \ldots, n\}$, user U_i from broadcast B_j recovers $K_j = z_j - h_j$.

Notice that a new user can always join the group at the j-th session: The group manager GM gives him a public identifier r in $F_q \setminus \{1, \ldots, n\}$, and sends him $S_r =< h_j, \ldots, h_m >$ as a personal key.

We can show that the above construction realizes a self-healing key distribution scheme without collusion resistance and revocation capabilities. More precisely, we can prove the following theorem.

Theorem 6.1 SCHEME 1 *satisfies conditions* $1.(a), 1.(b)$ *and* $3.(a)$ *of Definition 3.1.*

Proof. Condition $1.(a)$ easily follows noticing that, for all $i \in \{1, \ldots, n\}$, user U_i recovers K_j from the broadcast B_j, by evaluating $z_j - h_j$. Condition $1.(b)$ can be shown in two parts. It is straightforward to see that the personal keys alone do not give any information about any key: Indeed, the session keys are independent of the personal keys, and they are chosen according to the uniform probability distribution. Moreover, we can prove that the broadcast messages $B_1, \ldots, B_m$ alone do not give any information about the session keys as follows: The broadcast messages are given by:

$$\begin{cases} B_1 =< z_1 > \\ B_2 =< z_2 > \\ B_3 =< z_1 + z_2, z_3 > \\ B_4 =< z_1 + z_2, z_1 + z_3, z_4 > \\ \vdots \\ B_m =< z_1 + z_2, z_1 + z_3, \ldots, z_1 + z_{m-1}, z_m > \end{cases}$$

It is easy to see that several pieces are repeated a few times. Eventually, an adversary gets $z_1, z_2, \ldots, z_m$. However, since every z_j, for $j = 1, \ldots, m$, *perfectly* hides keys K_j, by means of the corresponding value of the personal key, the adversary does not gain any information about the keys.

On the other hand, condition $3.(a)$ holds since a broadcast message, say B_r, sent after B_j, enables the user to compute $z_1 = (z_1 + z_j) - z_j$, by means of which the user can subsequently recover $z_{j+1}, \ldots, z_{r-1}$ from B_r and, hence, the whole sequence of session keys, $K_j, \ldots, K_r$. ∎

Along the same lines of [37], applying the technique developed in [1, 31], we can use univariate t-degree polynomials for providing user revocation. To this aim, notice that Lagrange's interpolation formula for a polynomial $P(x)$ from $t + 1$ values at points $x_0, \ldots, x_t$ different from 0 says that we can compute

$$P(0) = \sum_{i=0}^{t} \lambda_i P(x_i)$$

where the $\lambda_i = \prod_{j \neq i} \frac{x_j}{x_j - x_i}$ are the Lagrange's coefficients and depend on the points x_i.

The following construction describes a self-healing session key distribution scheme with revocation capability and resilient to collusion attacks.

SCHEME 2.

Assume that $G_1 = \{U_1, \ldots, U_n\}$.

Set-up

The group manager GM:

- chooses, independently and uniformly at random, m polynomials of degree t, say $s_1(x), \ldots, s_m(x) \in F_q[x]$, and m session keys, $K_1, \ldots, K_m \in F_q$
- defines, for each $j = 1, \ldots, m$, the value $z_j = K_j + s_j(0)$
- sends in a private way, for $i = 1, \ldots, n$, to user U_i as personal key the sequence of values $S_i =< s_1(i), \ldots, s_m(i) >$.

Broadcast

Let $R_j \subseteq G_{j-1}$ denote the set of users revoked in session j, and let $R = R_j \cup R_{j-1} \cup \ldots \cup R_2$, such that $|R| \leq t$. The group manager GM:

- chooses a set of indices (different from 0) $W = \{\omega_1^j, \ldots, \omega_t^j\}$, such that the indices of the users in R, denoted by the set I_R, are contained in W, i.e., $I_R \subseteq W$, but $W \cap I_{G_j} = \emptyset$, where I_{G_j} represents the set of indices of the users in G_j.
- broadcasts in session j, a message B_j given by the concatenation $B_j^1||B_j^2$ of the sequences B_j^1 and B_j^2 where, for $j = 1, 2$

$$B_j =< z_j > || < \omega_1^j, \ldots, \omega_t^j, s_j(\omega_1^j), \ldots, s_j(\omega_t^j) >$$

while, in session $j \in \{3, \ldots, m\}$,

$$B_j^1 =< z_1 + z_2, \ldots, z_1 + z_{j-1}, z_j >,$$

and

$$B_j^2 =< \omega_1^j, \ldots, \omega_t^j, s_j(\omega_1^j), \ldots, s_j(\omega_t^j) > ||B_{j-1}^2$$

denoting by B_0^2 the empty sequence.

Key Computation in Session j

User U_i:

- recovers $s_j(0)$ applying Lagrange's formula to $\{(\omega_\ell^j, s_j(\omega_\ell^j))\}_{\ell=1,\ldots,t}$ and $(i, s_j(i))$
- computes K_j computing $z_j - s_j(0)$

A new user U_i, can join the group at the j-th session: the group manager GM gives him a public identifier r in $F_q \setminus \{1, \ldots, n\}$ and sends him $S_r =< s_j(r), \ldots, s_m(r) >$ as a personal key. Notice that if a certain user U_i is revoked in session j, then he must be revoked in all future sessions. In other words, the pair $(i, s_\ell(i))$ must be part of the broadcast message B_ℓ, for any $\ell = j, \ldots, m$. Hence, the scheme allows for revoking up to t users from the group.

We can now show the following result:

Theorem 6.2 SCHEME 2 *is a self-healing key distribution scheme.*

Proof. All conditions required by Definition 3.1 are satisfied.

- Condition 1.(a) easily follows noticing that, for all $i \in \{1, \ldots, n\}$, user U_i recovers K_j from the broadcast message B_j computing $s_j(0)$ and then $K_j = z_j(i) - s_j(0)$.

- Condition 1.(b) can be shown in two parts. It is straightforward to see that the personal keys alone do not give any information about any key: Indeed, the session keys are still independent of the personal keys, and they are chosen according to the uniform probability distribution. Moreover, we can prove that the broadcast messages $B_1, \ldots, B_m$ alone do not give any information about the session keys as follows: The broadcast messages are given by:

$$\begin{cases} B_1 = < z_1, \omega_1^1, \ldots, \omega_t^1, s_1(\omega_1^1), \ldots, s_1(\omega_t^1) > \\ B_2 = < z_2, \omega_1^2, \ldots, \omega_t^2, s_2(\omega_1^2), \ldots, s_2(\omega_t^2) > \\ B_3 = < z_1 + z_2, z_3, \omega_1^3, \ldots, \omega_t^3, s_3(\omega_1^3), \ldots, s_3(\omega_t^3), \\ \quad \omega_1^2, \ldots, \omega_t^2, s_2(\omega_1^2), \ldots, s_2(\omega_t^2), \omega_1^1, \ldots, \omega_t^1, s_1(\omega_1^1), \ldots, s_1(\omega_t^1) > \\ \vdots \\ B_m = < z_1 + z_2, z_1 + z_3, \ldots, z_1 + z_{m-1}, z_m, \\ \quad \omega_1^m, \ldots, \omega_t^m, s_m(\omega_1^m), \ldots, s_m(\omega_t^m), \ldots, \omega_1^1, \ldots, \omega_t^1, s_1(\omega_1^1), \ldots, \\ \quad s_1(\omega_t^1) > \end{cases}$$

  It is possible to compute $z_1, z_2, \ldots, z_m$ from the broadcast messages. It is not difficult to see that every z_j, for $j = 1, \ldots, m$, *perfectly* hides keys K_j, by means of $s_j(0)$ since $z_j = K_j + s_j(0)$.

- Condition 2 is satisfied because, a user U_i, revoked in session j, does not get any information about $s_j(0)$: he can count only on t points since the point $s_j(i)$, provided by his own personal key, is part of the broadcast message B_j sent by the group manager, i.e., the values $i, s_j(i)$ are in B_j^2. For any guess of a $s_j(0)$ he can interpolate a different polynomial $s_j(x)$. Therefore, K_j is completely safe.

- Condition 3.(a) (i.e., the self-healing property) follows by noticing that, from any two broadcasts B_r and B_s, where $r < s$, user U_i can compute all keys K_j, for $r \le j \le s$. The mechanism is exactly the same described in SCHEME 1: the only difference in this case is that, once the user has recovered $z_{r+1}, \ldots, z_{s-1}$ then he needs, for $j = r+1, \ldots, s-1$, the values of t points (plus the point provided by his own personal key), in order to compute $s_j(0)$. But these collections of points are provided by the second part of the broadcast messages B_s, i.e., B_s^2.

- Condition 3.(b) holds because users revoked before session r, do not get any information about K_j, for $j \ge r$, by themselves. At the same time new users, who join the group after session s, cannot contribute with any point to recover a key K_j, communicated during sessions $j < s$. Hence, session keys K_j, for any $r \le j < s$, are completely safe with respect to joint coalitions of size at most t of new and revoked users.

■

In terms of memory storage and communication complexity, our construction requires that user U_i stores a personal key S_i of size $|S_i| = (m - j + 1)\log q$, which is optimal with respect to Theorem 5.2, and, for $j \ge 3$, it has broadcast size $|B_j| = j \log q + 2tj \log q$. Hence, there is a substantial improvement compared to [37], where $|S_i| = m^2 \log q$ and $|B_j| = m(t+1)\log q + t\log q + mt(t+1)\log q$. Moreover, our scheme is still a bit more efficient than the scheme recently given in [32], where the personal key of every user is $2(m - j + 1)\log q$ bits, and the size of the broadcast is $[(m + j + 1)t + (m + 1)]\log q$ bits.

We stress that the main difference of our protocol, compared to the one given in [37, 32], lies in a different implementation of the self-healing property: In [37, 32] any user, belonging to G, from B_r and B_s can recover all keys K_j, for every $r \leq j \leq s$. In our protocol, it is not difficult to see that user U_i, from B_r and B_s, can recover K_j for *any* session $j \leq s$ in which U_i is member of the group. Moreover, notice that the scheme given in [37, 32] satisfies the following condition : Every user $U_i \in G$, from a single broadcast B_j, can compute only information about K_j. Formally, it holds that:

$$H(\mathbf{K}_1, \ldots, \mathbf{K}_{j-1}, \mathbf{K}_j, \mathbf{K}_{j+1}, \ldots, \mathbf{K}_m | \mathbf{S}_i, \mathbf{B}_j) = H(\mathbf{K}_1, \ldots, \mathbf{K}_{j-1}, \mathbf{K}_{j+1}, \ldots, \mathbf{K}_m). \tag{7}$$

In our scheme, such a strong condition is not satisfied because an authorized user can get from B_j *partial information about previous keys of sessions in which he belongs to the group.* Such a possibility is not a problem in terms of security.

7 Key-recovery from a single Broadcast

In this section we propose another key-recovery method. This method has the following advantage, compared to the mechanism introduced in [37]: in those schemes, the users can only recover the keys belonging to sessions "sandwiched" between two sessions in which the user gets the broadcast messages. In particular, if the user loses the first message B_1 and gets message B_j for a certain session j, then he has no way for recovering the lost session keys. With the following scheme, a user can recover all lost session keys (for sessions in which he belongs to the group) by using *only the current* broadcast message. From a formal point of view, the key-recovery property we require, which replaces $1.(a), 3.(a)$ in Definition 3.1, is the following:

1. *For any $\ell \leq s \leq m$, and for any user $U_i \in G_\ell$, who is not revoked before session s, the key K_ℓ is determined by $\mathbf{B}_s$ and $\mathbf{S}_i$.* Formally, it holds that:

$$H(\mathbf{K}_\ell | \mathbf{B}_s, \mathbf{S}_i) = 0. \tag{8}$$

The following scheme, which slightly modifies the previous one, describes a scheme resilient to collusion attacks, with key-recovery and with revocation capabilities.

Scheme 3.

Assume that $G_1 = \{U_1, \ldots, U_n\}$.

Set-up
The group manager GM:

- chooses, independently and uniformly at random, m polynomials of degree t, say $s_1(x), \ldots, s_m(x) \in F_q[x]$, and m session keys, $K_1, \ldots, K_m \in F_q$
- defines, for each $j = 1, \ldots, m$, the value $z_j = K_j + s_j(0)$
- sends in a private way, for $i = 1, \ldots, n$, to user U_i as personal key the sequence of values $S_i = < s_1(i), \ldots, s_m(i) >$.

Broadcast

Let $R_j \subseteq G_{j-1}$ denote the set of users revoked in session j, and let $R = R_j \cup R_{j-1} \cup \ldots \cup R_2$, such that $|R| \leq t$. The group manager GM:

- chooses a set of indices (different from 0) $W = \{\omega_1^j, \ldots, \omega_t^j\}$, such that the indices of the users in R, denoted by the set I_R, are contained in W, i.e., $I_R \subseteq W$, but $W \cap I_{G_j} = \emptyset$, where I_{G_j} represents the set of indices of the users in G_j.
- broadcasts in session $j \in \{1, \ldots, m\}$ a message B_j given by the concatenation $B_j^1 || B_j^2$ of the sequences B_j^1 and B_j^2, where:

$$B_j^1 = < z_j, z_{j-1}, \ldots, z_1 >$$

and

$$B_j^2 = < \omega_1^j, \ldots, \omega_t^j, s_j(\omega_1^j), \ldots, s_j(\omega_t^j) > || B_{j-1}^2$$

denoting by B_0^2 the empty sequence.

Key Computation in Session j

User U_i:

- recovers $s_j(0)$ applying Lagrange's formula to $\{(\omega_\ell^j, s_j(\omega_\ell^j))\}_{\ell=1,\ldots,t}$ and $(i, s_j(i))$
- computes K_j evaluating $z_j - s_j(0)$

Along the same lines of the proof given for Scheme 2 the above scheme can be shown to be correct and secure. In particular, Condition (8) is satisfied because, from any broadcast B_s user U_i, belonging to the group G_ℓ, can compute all keys K_ℓ, for $\ell \leq s$. Indeed, once the user has recovered the polynomial z_ℓ from B_s, then he needs exactly t points, plus the point provided by his own personal key, in order to compute $s_\ell(0)$. But these collections of points are computable by using B_s^2.

In the above scheme we allow from a single broadcast message B_s to recover K_ℓ for *any* $\ell \leq s$, if the user was member of the group during those sessions. In terms of memory storage and communication complexity, also this construction requires that user U_i stores a personal key S_i of size $|S_i| = (m - j + 1) \log q$, which is optimal with respect to Theorem 5.2, and, for $j \geq 1$, it has broadcast size $|B_j| = j \log q + 2tj \log q$.

8 Long-Lived Schemes

The schemes presented in the above sections enable the group manager GM to establish a session key with the users for m different sessions. Along the same lines of [37], applying a standard cryptographic technique, we can set up computationally secure *long-lived* protocols, i.e., protocols where the number of sessions is not fixed, by using a public generator g of a cyclic subgroup $H \subseteq F_q^*$ of prime order p, and by moving all computations to the exponent. However, as we will point out later on, the computationally secure long-lived scheme given in [37] (i.e. construction 5) has two problems. We modify the scheme to solve one of them, while the other seems to be an interesting open problem.

The key idea, on which the long lived scheme is based, is to do interpolation in the exponents. Since, as we have seen before, Lagrange's interpolation formula for a polynomial $P(x)$ from $t+1$ values at points $x_0, \ldots, x_t$ different from 0 says that we can compute

$$P(0) = \sum_{i=0}^{t} \lambda_i P(x_i)$$

where the $\lambda_i = \prod_{j \neq i} \frac{x_j}{x_j - x_i}$ are the Lagrange's coefficients and depend on the points x_i, then we can also compute

$$g^{P(0)} = g^{\sum_{i=0}^{t} \lambda_i P(x_i)} = \prod_{i=0}^{t} g^{\lambda_i P(x_i)}. \qquad (9)$$

Interpolation in the exponents was first used by Feldman in [18] in order to enable each participant in a secret sharing scheme to verify his own share, received from a possibly dishonest dealer, with no loss in security. Such a property was guaranteed by the difficulty of computing the discrete log in certain finite multiplicative groups. Since then, interpolation in the exponents has been applied in several papers. In the following construction it is used to allow users *to evolve* their personal keys from one set of m sessions to multiple sets of m sessions. This is accomplished through the broadcast of random values from the group manager, which are used by the group members for computing, in each new set of m sessions, a new instance of their own personal keys.

More precisely, the long-lived scheme is implemented by using repeatedly SCHEME 3, performing the computation in the exponents (similarly, we can make long-lived also SCHEME 2). The group manager, for the α-th set of m sessions, defines the session keys as $g^{K_{\alpha,1}}, \ldots, g^{K_{\alpha,m}}$, and broadcasts some random values $g^{v_{\alpha,1}}, \ldots, g^{v_{\alpha,m}}$. Each user U_i uses his/her sequence of values $< s_1(i), \ldots, s_m(i) >$ for computing $< g^{v_{\alpha,1} s_1(i)}, \ldots, g^{v_{\alpha,m} s_m(i)} >$ which can be seen as the instance of the personal key for the α-th run of the long-lived scheme. Then, every user in the communication group recovers the session key $g^{K_{\alpha,j}}$ from $z_{\alpha,j} = g^{K_{\alpha,j} + v_{\alpha,j} s_j(0)}$ and $\omega^j_{\alpha,1}, \ldots, \omega^j_{\alpha,t}, g^{v_{\alpha,j} s_j(\omega^j_{\alpha,1})}, \ldots, g^{v_{\alpha,j} s_j(\omega^j_{\alpha,t})}$, available from the broadcast B_j, and by using his/her own value $g^{v_{\alpha,j} s_j(i)}$ for computing the value $g^{v_{\alpha,j} s_j(0)}$ and, then, $g^{K_{\alpha,j}} = z_{\alpha,j} / g^{v_{\alpha,j} s_j(0)}$.

As we will see later, the security of such an extended scheme relies on the difficulty of solving the decisional Diffie-Hellman problem (DDH, for short) in H, which is a well-known assumption used in cryptography.

Let t be a positive integer, let g be a generator of a cyclic subgroup $H \subseteq F_q^*$ of prime order p in which the DDH assumption holds.

SCHEME 4.

Assume that $G_1 = \{U_1, \ldots, U_n\}$.

Set-up

The group manager GM:

- chooses, independently and uniformly at random, m polynomials of degree t in both variables, say $s_1(x), \ldots, s_m(x)$ with coefficients in F_p.
- sends in a private way, for $i = 1, \ldots, n$, to user U_i the sequence of values $S_i = < s_1(i), \ldots, s_m(i) >$.

Computation of fresh values and keys for the α-th set of m sessions

The group manager GM:

- chooses, uniformly at random, integers $v_{\alpha,1}, \ldots, v_{\alpha,m} \in F_p$ and computes the sequence $UP_\alpha = < g^{v_{\alpha,1}}, \ldots, g^{v_{\alpha,m}} >$
- chooses, uniformly at random, m values, $K_{\alpha,1}, \ldots, K_{\alpha,m} \in F_p$ and, for each $j = 1, \ldots, m$, he defines $z_{\alpha,j} = g^{K_{\alpha,j} + v_{\alpha,j} s_j(0)}$

When the group manager needs to send a new session key, then he applies the following steps:

Scheme 4.

Broadcast during the α-th set of m sessions

Let $R_j \subseteq G_{j-1}$ denote the set of users revoked in session j, and let $R = R_j \cup R_{j-1} \cup \ldots \cup R_2$, such that $|R| \leq t$. The group manager GM:

- chooses a set of indices (different from 0) $W = \{\omega^j_{\alpha,1}, \ldots, \omega^j_{\alpha,t}\}$, such that the indices of the users in R, denoted by the set I_R, are contained in W, i.e., $I_R \subseteq W$, but $W \cap I_{G_j} = \emptyset$, where I_{G_j} represents the set of indices of the users in G_j.
- broadcasts in session $j \in \{1, \ldots, m\}$ a message B_j given by the concatenation $B^1_j || B^2_j$ of the sequences B^1_j and B^2_j, where:

$$B^1_j = UP_\alpha || < z_{\alpha,j}, z_{\alpha,j-1}, \ldots, z_{\alpha,1} >,$$

and

$$B^2_j = < \omega^j_{\alpha,1}, \ldots, \omega^j_{\alpha,t}, g^{v_{\alpha,j} s_j(\omega^j_{\alpha,1})}, \ldots, g^{v_{\alpha,j} s_j(\omega^j_{\alpha,t})} > || B^2_{j-1}$$

denoting by B^2_0 the empty sequence.

Key Computation during the α-th set of m sessions

User U_i in session j:

- recovers $g^{v_{\alpha,j} s_j(0)}$ interpolating $\{(\omega^j_{\alpha,\ell}, g^{v_{\alpha,j} s_j(\omega^j_{\alpha,\ell})})\}_{\ell=1,\ldots,t}$ and $(i, g^{v_{\alpha,j} s_j(i)})$
- gets the session key $g^{K_{\alpha,j}}$ computing $z_{\alpha,j}/g^{v_{\alpha,j} s_j(0)}$.

Key Computation. It is easy to see by simple algebra that condition (8) is satisfied *in each set* of m sessions. From the broadcast messages and his own personal key, a group member can compute the session keys she is entitled to. We could also extend the scheme for enabling key-recovery over *more than one set* of m sessions, by sending during the α-th set of m sessions also the sequences $UP_\alpha, UP_{\alpha-1}, \ldots, UP_1$ and the broadcast messages of previous sets of m sessions. However, the size of the broadcast will be too large. An interesting open problem is to find more efficient constructions. A first improvement could be, instead of generating uniformly at random, integers $v_{\alpha,1}, \ldots, v_{\alpha,m} \in F_p$ and computing $UP_\alpha =< g^{v_{\alpha,1}}, \ldots, g^{v_{\alpha,m}} >$ to use only one random value r as a seed for a pseudo-random generator, in order to derive the other values.

A Weakness in Construction 5 of [37]. Notice that Scheme 4 is different from Construction 5 of [37] in an important aspect: we are sending with *each* broadcast the values that are used for computing the instances of the personal key in each new set α of m sessions. In [37], these values are sent by means of a *single* broadcast message at the beginning of the α-th set of m sessions. But since we are assuming that the network is not reliable, if some user does not receive such a message, then she has no way for computing her personal key. It follows that such a user gets out from the communication group.

Join. Notice that a second problem associated with Construction 5 of [37] and our scheme lies in the *join* operation in presence of new users. A straightforward extension of a scheme for m sessions does not work. If a new user gets an identifier r and the sequence $< s_1(r), \ldots, s_m(r) >$, then from the broadcast messages of previous sessions, she can recover session keys associated to groups where she does not belong to. On the other hand, if the user joins the group in session j and gets the sequence $< s_j(r), \ldots, s_m(r) >$, then she cannot use the personal key during the first $j-1$ sessions of subsequent sets of m sessions and she can still recover previous session keys. It seems difficult to slightly modify Construction 5 of [37] and SCHEME 4 in order to enable a secure join.

Computationally Secure Schemes. In order to show that the above construction is secure, we have to provide a definition of what we mean by "computationally secure". Indeed, if the key computation condition and the key-recovery condition (the self-healing property, resp., if we extend SCHEME 2) can still be stated in terms of information theory, the security conditions given by Definition 3.1 do not apply to this new scenario.

Notice that a broadcast message contains, in a certain sense, an encryption of the value of the session key. Such an encryption can be "opened" only by group members. In our scheme, for example, the encryption is represented by $z_{\alpha,j}$ and it can be opened by computing and removing the term $g^{v_{\alpha,j} s_j(0)}$. Notice that this is essentially an El Gamal encryption [17].

Informally speaking, we say that the scheme is computationally secure with respect to revoked users, if the following condition is satisfied: for any session j and for any set of revoked users of size less than or equal to t, it is computationally infeasible to distinguish with non-negligible probability between the session key at time j and a random value, given their View, which consists in their own personal keys and broadcast messages before, on, and after time j.

Security of SCHEME 4. We can show that SCHEME 4 is computationally secure with respect to revoked users assuming the difficulty of solving the DDH problem in a large group of prime order. Loosely speaking, the DDH assumption says that, given a cyclic group H and a generator g, it is computationally infeasible to establish if a certain triple $< g^a, g^b, y >$ is of the form $< g^a, g^b, g^{ab} >$ or $< g^a, g^b, g^c >$, where a, b, and c are chosen uniformly at random in $\{0, \ldots, |H|-1\}$. For details the reader is referred to [9].

The idea is to divide the proof in two steps: first we consider the case where $m = 1$ (i.e., the basic scheme is used for providing a single session key), and then we extend the proof to the case $m > 1$.

When $m = 1$, the broadcast message B_j is:

$$< g^{v_\alpha}, g^{K_\alpha + v_\alpha s(0)}, \omega_{\alpha,1}, \ldots, \omega_{\alpha,t}, g^{v_\alpha s(\omega_{\alpha,1})}, \ldots, g^{v_\alpha s(\omega_{\alpha,t})} >,$$

where we have used v_α, $\omega_{\alpha,1}, \ldots, \omega_{\alpha,t}$ and $s(\cdot)$ instead of $v_{\alpha,1}$, $\omega^1_{\alpha,1}, \ldots, \omega^1_{\alpha,t}$ and $s_1(\cdot)$ to simplify the notation. A coalition of t users, say $U_1, \ldots, U_t$, revoked at the α-th iteration of the scheme has the following View:

- the sequence of values $s(1), \ldots, s(t)$
- $\alpha - 1$ tuples of $(2t+4)$ elements, say

  $$< g^{v_\beta}, g^{K_\beta + v_\beta s(0)}, \omega_{\beta,1}, \ldots, \omega_{\beta,t}, g^{v_\beta s(\omega_{\beta,1})}, \ldots, g^{v_\beta s(\omega_{\beta,t})}, g^{K_\beta}, g^{v_\beta s(0)} >,$$

  where $\beta = 1, \ldots, \alpha - 1$, and for some $j = 1, \ldots, t$, we might have $\omega_{\beta,j} = i$ (i.e., some users among $U_1, \ldots, U_t$ have already been revoked before the α-th set of sessions). The $(2t+4)$-tuples represent information that $U_1, \ldots, U_t$ have received (and computed) when they were members of the communication group.
- a $(2t+2)$-tuple for the α-th set given by

  $$< g^{v_\alpha}, g^{K_\alpha + v_\alpha s(0)}, 1, \ldots, t, g^{v_\alpha s(1)}, \ldots, g^{v_\alpha s(t)} >$$

- ϕ tuples of $(2t+2)$ elements, say

$$< g^{v_\ell}, g^{K_\ell + v_\ell s(0)}, 1, \ldots, t, g^{v_\ell s(1)}, \ldots, g^{v_\ell s(t)} >$$

where $\ell > \alpha$, corresponding to future broadcast messages in which the users are kept revoked.

The coalition is successful in breaking the scheme if there exists an efficient algorithm A that, given the View, with non-negligible probability, can determine whether a challenge value γ is g^{K_α} or it is a random element in H.

We show that such an algorithm A can be used to construct an efficient algorithm A' for solving the DDH problem.

Notice that, for each value $g^{K_\alpha + v_\alpha s(0)}$, given g^{v_α}, g^{v_β}, and $g^{v_\beta s(0)}$, the value of g^{K_α} is *uniquely determined.* Hence, given the View, the value g^{K_α} is uniquely determined.

The idea of the proof is the following: given the challenge triple $< g^a, g^b, y >$ for the DDH problem, chosen uniformly at random, A' first generates a View, i.e., a set of tuples to be given in input to A, which have *exactly the same distribution* of tuples produced by a real execution of SCHEME 4, but are constructed in such a way that the output of A on them gives also an answer to the DDH problem. Then, A is run on this View, and its output is taken as the output of A'.

More precisely, A' constructs the input for A as follows: it chooses uniformly at random values $s(1)$, …, $s(t)$ in F_p, and computes $g^{s(1)}, \ldots, g^{s(t)}$. These values are used in constructing the α-th tuple, corresponding to the α-th set, and the tuples associated with future sets, i.e., the $(\alpha+1)$-th, $(\alpha+2)$-th, …. Moreover, *some* of them are also used for constructing tuples associated with previous sets of sessions. Indeed, in constructing previous (to the α-th) tuples, some values are associated with users in $\{U_1, \ldots, U_t\}$ who have already been revoked at that time, but some other values do not correspond to users in the communication group, and need to be used in order to construct the broadcast message. Notice that these values must be consistent with the polynomial $s(x)$ on which $s(1), \ldots, s(t)$ belong to.

The idea is to choose these values all on the polynomial $s(x)$ interpolated by $s(1), \ldots, s(t)$ and $s(0)$, where $s(0) = b$, the exponent of g^b in the DDH challenge triple $< g^a, g^b, y >$. Such an operation is possible by using Lagrange's interpolation formula in the exponents. Indeed, it is not difficult to see that, given $(1, g^{s(1)}), \ldots, (t, g^{s(t)}), (0, g^{s(0)})$, where $s(0) = b$, equation (9) can be used to construct pairs of values $(\omega_{\beta,j}, g^{s(\omega_{\beta,j})})$, where $\omega_{\beta,j}$ is chosen uniformly at random in $F_p \setminus \{0, 1, \ldots, t\}$, and $g^{s(\omega_{\beta,j})}$ is obtained by computing

$$g^{s(\omega_{\beta,j})} = g^{\sum_{i=0}^{t} \lambda_i s(i)} = \prod_{i=0}^{t} g^{\lambda_i s(i)}, \tag{10}$$

where the Lagrange's coefficients are in this case given by $\lambda_i = \prod_{r \neq i} \frac{\omega_{\beta,j} - r}{i - r}$.

Let us fix a revocation schedule, i.e., the users the group manager revokes and the corresponding revocation times, as well as the set of $\omega_{\beta,j}$ values. Then, A' proceeds as follows:

- chooses, uniformly at random, $\alpha+\phi$ values of $v_1, \ldots, v_{\alpha+\phi}$, and $\alpha+\phi$ values of $K_1, \ldots, K_{\alpha+\phi}$, for computing session keys $g^{K_1}, \ldots, g^{K_{\alpha+\phi}}$
- constructs $\alpha - 1$ tuples of the form

$$< g^{v_\beta}, g^{K_\beta + v_\beta s(0)}, \omega_{\beta,1}, \ldots, \omega_{\beta,t}, g^{v_\beta s(\omega_{\beta,1})}, \ldots, g^{v_\beta s(\omega_{\beta,t})}, g^{K_\beta}, g^{v_\beta s(0)} >,$$

where $\beta = 1, \ldots, \alpha-1$. Notice that, if in the q-th tuple, for $1 \leq q \leq \alpha - 1$, the value $\omega_{\beta,j} = i$, for some $i = 1, \ldots, t$, then $g^{s(i)}$ is already available. Otherwise, it is computed according to (10). Notice that all the tuples can be efficiently computed, since g is the public generator of the group, g^b is the second element of the triple $< g^a, g^b, y >$, and all other terms can be obtained by simple exponentiations.

- constructs the tuple

$$< g^a, g^{K_\alpha + as(0)}, 1, \ldots, t, g^{as(1)}, \ldots, g^{as(t)} >$$

where g^a is the first term of the triple $< g^a, g^b, y >$

- constructs other ϕ tuples

$$< g^{v_\beta}, g^{K_\beta + v_\beta s(0)}, 1, \ldots, t, g^{v_\beta s(1)}, \ldots, g^{v_\beta s(t)} >$$

where $\beta = \alpha + 1, \ldots, \alpha + \phi$. Again, all elements of the tuple can be constructed.

Observe that, by construction, the above $\mathsf{View}^{A'}$ *has the same distribution of the real* View. Then, A' constructs the challenge $\gamma = g^{K_\alpha + as(0)} \cdot y^{-1}$, where y is the third term of the triple $< g^a, g^b, y >$.

Finally, the algorithm A' gives in input to A the View and the challenge γ. Notice that:

> *If $y = g^{ab}$, then $\gamma = g^{K_\alpha}$. On the other hand, if y is a random value, then γ is a random value.*

Therefore, If A outputs that γ is the session key g^{K_α}, then A' outputs that $< g^a, g^b, y >$ is of the form $< g^a, g^b, g^{ab} >$. On the other hand, if A outputs that γ is a random value, then A' outputs that $< g^a, g^b, y >$ is a random triple $< g^a, g^b, g^c >$.

It follows that A' solves the DDH problem with the same non-negligible probability with which A distinguishes a session key from a random value. Hence, if the DDH assumption holds, then Scheme 4 is secure with respect to revoked users.

To extend the proof to the case where $m > 1$, note that the polynomials $s_1(x), \ldots, s_m(x)$ are chosen independently and uniformly at random. In the general case, the coalition $U_1, \ldots, U_t$, revoked at the j-th session of the α-th set of m sessions, has the following view:

- m sequences of values $< s_1(1), \ldots, s_m(1) >, \ldots, < s_1(t), \ldots, s_m(t) >$
- $m \cdot (\alpha - 1) + (j - 1)$ tuples of $(2t + 4)$ elements
- a (2t+2)-tuple for the j-th session in the α-th set
- ϕ tuples of $(2t + 2)$ elements associated with the other $m - j$ sessions of the α-th set and future sets of m sessions

whose structure is identical to the structure of the tuples considered for the case in which $m = 1$. Assuming that there exists an efficient algorithm A which distinguishes with non-negligible probability a session key from a random value in the challenge, then we can construct again an algorithm A' which solves the DDH problem. In such a case, A' constructs the input for A by using $(m - 1)(\alpha + \phi)$ tuples produced by a real execution of Scheme 4, and constructs $\alpha + \phi$ tuples, which correspond to the j-th sessions of the $\alpha + \phi$ sets of m sessions, exactly according to the strategy we have used for the case $m = 1$. It is easy to see that all tuples given in input to A have the same distribution of real ones. Hence, A' solves DDH with the same probability with which A breaks Scheme 4.

Performance. It is easy to see that the size of the personal key of each user is equal to $m \log q$ bits, while the broadcast size is equal to $m \log q + j \log q + 2tj \log q$ bits. However, notice that q must be big enough so that DDH is difficult in H (see [28] for choosing the parameter).

9 Conclusions and Open Problems

In this paper we have analysed key distribution schemes with key-recovery capabilities, enabling groups of users to establish a common key for secure communication over an unreliable network.

We have analysed some constructions given in [37], showing an attack that can be applied to the basic construction, in order to point out the threats underlying the design of such schemes, and a weakness in the long-lived construction, for which some members can be excluded from the communication group in presence of packet loss.

Then, we have proposed a new self-healing key distribution scheme, which is optimal in terms of user memory storage and quite efficient in terms of communication complexity. Finally, we have slightly modified the model, in order to extend the self-healing model, and we have proposed a scheme which enables a user to recover from a single broadcast message all keys associated with sessions in which he is member of the communication group.

The self-healing approach is a new and suitable method for key distribution. Many applications can benefit from efficient and secure implementations. Further research could be done in order to clearly identify the attacks that might be implemented in such models, the constraints affecting available schemes, and to design more efficient and flexible schemes.

Towards this goal, in [6], it is shown that the collusion resistance condition, introduced in the model of [37], is impossible to achieve.

Acknowledgements

We would like to thank German Saez and an anonymous referee for helpful comments and suggestions.

References

[1] J. Anzai, N. Matsuzaki, and T. Matsumoto, A Quick Group Key Distribution Scheme with Entity Revocation, *Advances in Cryptology - Asiacrypt '99, Lecture Notes in Computer Science*, Vol. 1716, pp. 333-347.

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

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

[4] C. Blundo, P. D'Arco, and M. Listo, A New Self-healing Key Distribution Scheme, *Proceedings of the IEEE Symposium on Computers and Communications (ISCC 2003)*, pp. 803 - 808, 2003.

[5] C. Blundo, P. D'Arco, and M. Listo, A Flaw in a Self-Healing Key Distribution Scheme, *Proceedings of the 2003 Information Theory Workshop (ITW 2003)*, pp. 163-166, 2003.

[6] C. Blundo, P. D'Arco, and A. De Santis, A General Approach to Self-Healing Key Distribution, manuscript.

[7] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, Perfectly-Secure Key Distribution for Dynamic Conferences, *Information and Computation*, Vol. 146, no.1, pp. 1-23, 1998 .

[8] C. Blundo, Luiz A. Frota Mattos, and D. R. Stinson, Generalised Beimel-Chor Schemes for Broadcast Encryption and Interactive Key Distribution, *Theoretical Computer Science*, Vol. 200, pp. 313–334, 1998.

[9] D. Boneh, The Decision Diffie-Hellman Problem, *Proceedings of the Third Algorithmic Number Theory Symposium*, Lecture Notes in Computer Science, Vol. 1423, pp. 48–63, 1998.

[10] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, Issue in Multicast Security: A Taxonomy and Efficient Constructions, *Infocom '99*, pp. 708–716, 1999.

[11] R. Canetti, T. Malkin, and K. Nissim, Efficient Communication-Storage Tradeoffs for Multicast Encryption, *Advances in Cryptology - Eurocrypt '99, Lecture Notes in Computer Science*, Vol. 1592, pp. 459–474, 1999.

[12] B. Chor, A. Fiat, M. Naor and B. Pinkas, Traitor Tracing, *IEEE Transactions on Information Theory*, Vol. 46, No. 3, pp. 893–910, May 2000.

[13] T. M. Cover and J. A. Thomas, Elements of Information Theory, *John Wiley & Sons*, 1991.

[14] P. D'Arco and D. R. Stinson, Fault Tolerant and Distributed Broadcast Encryption, *Proceedings of the Cryptographers' Track RSA Conference 2003 (CT-RSA 2003), Lecture Notes in Computer Science*, Vol. 2612, pp. 262–279, 2003.

[15] G. Di Crescenzo and O. Kornievskaia, Efficient Multicast Encryption Schemes, *Security in Communication Network (SCN02), Lecture Notes in Computer Science*, Vol. 2576, pp. 119–132, 2003.

[16] C. Dwork, J. Lotspiech, and M. Naor, Digital Signets: Self-Enforcing Protection of Digital Information, *Proceedings of the* 28-*th Symposium on the Theory of Computation*, pp. 489–498, 1996.

[17] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory*, N. 31, pp. 469-472, 1985.

[18] P. Feldman, A Practical Scheme for Non-Interactive Secret Sharing, *Proceedings of the 28-th IEEE Symposium on Foundations of Computer Science*, pp. 427–437, 1987.

[19] A. Fiat and M. Naor, Broadcast Encryption, *Proceedings of Crypto '93, Lecture Notes in Computer Science*, Vol. 773, pp. 480-491, 1994.

[20] A. Fiat and T. Tessa, Dynamic Traitor Tracing, *Journal of Cryptology*, Vol. 14, pp. 211–223, 2001.

[21] E. Gafni, J. Staddon, and Y. L. Yin, Efficient Methods for Integrating Traceability and Broadcast Encryption, *Advances in Cryptology - Crypto '99, Lecture Notes in Computer Science*, Vol. 1666, p. 372–387, 1999.

[22] J. Garay, J. Staddon, and A. Wool, Long-Lived Broadcast Encryption, *Advances in Cryptology - Crypto 2000, Lecture Notes in Computer Science*, Vol. 1880, pp. 333–352, 2000.

[23] D. Halevy and A. Shamir, The LSD Broadcast Encryption Scheme, *Advances in Cryptology - Crypto '02, Lecture Notes in Computer Science*, Vol. 2442, pp. 47-60, 2002.

[24] A. Kiayias and M. Yung, Traitor Tracing with Constant Transmission Rate, *Advances in Cryptology - Eurocrypt '02, Lecture Notes in Computer Science*, Vol. 2332, pp. 450-465, 2002.

[25] A. Kiayias and M. Yung, Self Protecting Pirates and Black-Box Traitor Tracing, *Advances in Cryptology - Crypto '01, Lecture Notes in Computer Science*, Vol.2139, pp. 63-79, 2001.

[26] R. Kumar, S. Rajagopalan, and A. Sahai, Coding Constructions for Blacklisting Problems without Computational Assumptions, *Advances in Cryptology - Crypto '99, Lecture Notes in Computer Science*, Vol. 1666, pp. 609–623, 1999.

[27] H. Kurnio, R. Safani-Naini, and H. Wang, A Secure Re-keying Scheme with Key Recovery Property, *ACISP 2002, Lecture Notes in Computer Science*, Vol. 2384, pp. 40–55, 2002.

[28] A. K. Lenstra and E. R. Verheul, Selecting Cryptographic Key Sizes, *Journal of Cryptology*, Vol. 14, N. 4, pp. 255 - 293, 2001.

[29] M. Luby and J. Staddon, Combinatorial Bounds for Broadcast Encryption, *Advances in Cryptology - Eurocrypt '98, Lecture Notes in Computer Science*, Vol. 1403, pp. 512–526, 1998.

[30] D. Naor, M. Naor, and J. Lotspiech, Revocation and Tracing Schemes for Stateless Receivers, *Advances in Cryptology - Crypto '01, Lecture Notes in Computer Science*, Vol. 2139, pp. 41–62, 2001.

[31] M. Naor and B. Pinkas, Efficient Trace and Revoke Schemes, *Financial Cryptography 2000, Lecture Notes in Computer Science*, Vol. 1962, pp. 1–21, 2000.

[32] D. Liu, P. Ning, and K. Sun, Efficient Self-Healing Key Distribution with Revocation Capability, *Proceedings of the* 10*-th ACM Conference on Computer and Communications Security*, October 27-31, 2003, Washington, DC, USA.

[33] A. Perrig, D. Song, and J. D. Tygar, ELK, a new Protocol for Efficient Large-Group Key Distribution, *Proceedings of the IEEE Symposium on Security and Privacy (2000).*

[34] B. Pfitzmann, Trials of Traced Traitors, Information Hiding, *Lecture Notes in Computer Science*, Vol. 1174, pp. 49-64, 1996.

[35] R. Safavi-Naini and H. Wang, New Constructions for Multicast Re-Keying Schemes Using Perfect Hash Families, 7*th ACM Conference on Computer and Communication Security*, ACM Press, pp. 228–234, 2000.

[36] R. Safavi-Naini and Y. Wang, Sequential Traitor Tracing, *Lecture Notes in Computer Science*, Vol. 1880, p. 316–332, 2000.

[37] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin and D. Dean, Self-Healing Key Distribution with Revocation, *IEEE Symposium on Security and Privacy*, May 12-15, 2002, Berkeley, California.

[38] J. N. Staddon, D.R. Stinson and R. Wei, Combinatorial properties of frameproof and traceability codes, *IEEE Transactions on Information Theory*, Vol. 47, pp. 1042-1049, 2001.

[39] D. R. Stinson, On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption, *Designs, Codes and Cryptography*, Vol. 12, pp. 215–243, 1997.

[40] D. R. Stinson and T. van Trung, Some New Results on Key Distribution Patterns and Broadcast Encryption, *Designs, Codes and Cryptography*, Vol. 15, pp. 261–279, 1998.

[41] D. R. Stinson and R. Wei, Key preassigned traceability schemes for broadcast encryption, *Proceedings of SAC'98, Lecture Notes in Computer Science*, Vol. 1556, pp. 144-156, 1999.

[42] D. R. Stinson and R. Wei, Combinatorial properties and constructions of traceability schemes and frameproof codes, *SIAM Journal on Discrete Mathematics*, Vol. 11, pp. 41–53, 1998.

[43] D. R. Stinson and R. Wei, An Application of Ramp Schemes to Broadcast Encryption, *Information Processing Letters*, Vol. 69, pp. 131–135, 1999.

[44] D. M. Wallner, E. J. Harder, and R. C. Agee, Key Management for Multicast: Issues and Architectures, *Internet Draft available from http://www.ietf.org/rfc/rfc2627.txt*

[45] C. Wong, and S. Lam, Keystone: A Group Key Management Service, *Proceedings of the International Conference on Telecommunications, ICT 2000.*