A New Self-Healing Key Distribution Scheme

Carlo Blundo, Paolo D’Arco, and Massimiliano Listo
Dipartimento di Informatica ed Applicazioni
Universita di Salerno, 84081 Baronissi (SA), Italy
e-mail: {carblu, paodar}@dia.unisa.it, maslis@tiscalinet.it

Abstract

A self-healing key distribution scheme enables a group of users to establish a group key over an unreliable channel. In such a protocol, a group manager, to distribute a session key to each member of the group, broadcasts packets along the channel. If some packet gets lost, then users are still capable of recovering the group key using the received packets, without requesting additional transmission from the group manager. A user must be member both before and after the session in which a particular key is sent and lost, in order to be recovered through “self-healing”. In this paper we propose a new technique to do self-healing, and we provide a secure and efficient scheme.

1. Introduction

How to distribute session keys for secure communication to groups of users of a network over unreliable channels, 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 [19], an interesting approach to deal with this scenario has been proposed. A self-healing key distribution scheme [19] enables a group of users to establish a group key over an unreliable channel. In such a protocol, a group manager, to distribute a session key to each member of the group, broadcasts packets along the channel. If some packet gets lost, then users are still capable of recovering the group key using the received packets, without requesting additional transmission from the group manager. The only requirement is that a user must be member both before and after the session in which a particular key is sent and lost, in order to be recovered through “self-healing”. In this paper we propose a new technique to do self-healing, and we provide a secure and efficient scheme.

Our Contribution. In this paper we propose a new mechanism for implementing self-healing key distribution, and we provide a secure and efficient construction. According to the self-healing approach [19], a user from two different broadcast messages, say $B_{j_1}$ and $B_{j_2}$, can recover all keys $K_j$, for every $j_1 \leq j \leq j_2$. However, the model does not say anything about $K_{j_1}, \ldots, K_{j-1}$. The protocols proposed in [19] do not enable recovering such keys; while, we allow from $B_{j_1}$ and $B_{j_2}$ the recovering of $K_j$ for any $j \leq j_2$. This approach enables us to design a secure and efficient self-healing key distribution scheme.

Previous work. Broadcast Encryption is one of the closest area to the subject of this paper. Originated in [2], and formally defined in [10], it has been extensively studied (e.g., [3, 12, 21]), and it has grown up in different directions: mainly, re-keying schemes for dynamic groups of users (see, [23, 4, 5] to name a few), and broadcast schemes with tracing capability for dishonest users [6, 9, 11, 22, 20]. Moreover, several papers have addressed the special case of users revocation from a privileged subset [14, 1, 17, 16, 13]. However, all the above papers assume that the underlying network is reliable. The authors of [18] and [24], 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 are appended to the packets. The schemes given in [14], by accurately choosing the values of the parameters, can provide resistance to packet loss as well. Recently, in [19, 15] the problem 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. Finally, in [8] also this problem is considered. The paper generalises several known constructions in order to gain resistance to packet loss.
crypt old fragments, even if recoverable from the scheme, are pretty much useless, due to the nature of the application.

2. Background

In this section we briefly recall some basic notions of Information Theory [7]. Indeed, in the following section, we will use the entropy function to state the properties that self-healing key distribution schemes have to satisfy.

Let \( X \) be a random variable taking values on a set \( X \), according to a probability distribution \( \{P_X(x)\}_{x \in X} \). The entropy of \( X \), denoted by \( H(X) \), is defined as

\[
H(X) = - \sum_{x \in X} P_X(x) \log P_X(x),
\]

where the logarithm is relative to the base 2. The entropy satisfies \( 0 \leq H(X) \leq \log |X| \), where \( H(X) = 0 \) if and only if there exists \( x_0 \in X \) such that \( P_X(X = x_0) = 1 \); whereas, \( H(X) = \log |X| \) if and only if \( P_X(X = x) = 1/|X| \) for all \( x \in X \). The entropy of a random variable is usually interpreted as

- a measure of the “distance” of the random variable from a random variable uniformly distributed
- a measure of the “amount of information” given on average by the outcome of an experiment described by the random variable

Given two random variables \( X \) and \( Y \), taking values on sets \( X \) and \( Y \), respectively, according to a probability distribution \( \{P_{X,Y}(x,y)\}_{x \in X, y \in Y} \) on their Cartesian product, the conditional entropy \( H(X|Y) \) is defined as

\[
H(X|Y) = - \sum_{y \in Y} \sum_{x \in X} P_{X,Y}(x,y) \log P_{X|Y}(x|y).
\]

Simple algebra shows that

\[
H(X|Y) \geq 0
\]

with equality if and only if \( X \) is a function of \( Y \). The conditional entropy is interpreted as a measure of the amount of information that \( X \) still has, once given \( Y \).

The mutual information between \( X \) and \( Y \) is given by

\[
I(X;Y) = H(X) - H(X|Y).
\]

Since,

\[
I(X;Y) = I(Y;X) \quad \text{and} \quad I(X;Y) \geq 0,
\]

it is easy to see that

\[
H(X) \geq H(X|Y),
\]

3. The Model

We start by describing the setting we will deal with. Let \( GM \) be a group manager, and let \( U_1, \ldots, U_n \) be \( n \) users of the network. Each user \( U_i \) stores a personal key, \( S_i \), which can be seen as a subset of elements of a certain field \( F_q \), where \( q > n \). Individual personal keys can be related. All the operations of the protocols take place in \( F_q \). For \( j = 1, \ldots, m \), the session key \( K_j \) is sent to the group members through a broadcast, \( B_j \), from the group manager. The session key \( K_j \), for all \( U_i \) belonging to the group \( G \), is determined by \( B_j \) and \( S_i \), i.e., every user of the group, once received the broadcast \( B_j \), by means of his personal key, computes the session key \( K_j \). Session keys are generated by \( GM \) independently and according to the uniform probability distribution. Denoting by \( S_i, B_j, K_j \) the random variables associated with the above elements, and using the entropy function, we state in a rigorous way, the properties that a self-healing key distribution scheme must satisfy. Indeed, the entropy function enables modelling the protocol in a compact, elegant and clear form.

Let \( i \in \{1, \ldots, n\} \) and \( j \in \{1, \ldots, m\} \). \( D \) is a self-healing session key distribution scheme if the following conditions are true:

**Reconstruction.**

1. For every user \( U_i \in G \), the session key \( K_j \) is determined by \( S_i \) and \( B_j \). Formally, it holds that:

\[
H(K_j|S_i, B_j) = 0.
\]

2. For any \( j = 1, \ldots, m \), users \( U_1, \ldots, U_n \) do not learn any information about \( K_j \) from the broadcasts or personal keys alone. Formally, it holds that:

\[
H(K_j|B_1, \ldots, B_m) = H(K_j|S_1, \ldots, S_n) = H(K_j).
\]

Conditions (1) and (2) state that every user in \( G \), from the broadcast and his own personal key, recovers the current
session key; while, personal keys and broadcasts alone, do not give any information about any session key.

**Independence of Future Session Keys.**

1. All $U_i \in G$, without broadcasts $B_j$ for $j = m-r+1, \ldots, m$, cannot compute any information about keys $K_{m-r+1}, \ldots, K_m$. Formally, it holds that:

$$H(K_{m-r+1}, \ldots, K_m|S_1, \ldots, S_m, B_1, \ldots, B_{m-r}) = H(K_{m-r+1}, \ldots, K_m).$$

The above condition states that future broadcasts are needed to recover future session keys.

**Collusion Resistance.**

1. For any subset $F = \{U_{t_1}, \ldots, U_{t_r}\} \subseteq \{U_1, \ldots, U_n\}$, such that $|F| \leq t$ and $U_i \notin F$, the users in $F$ cannot determine anything about $S_i$. Formally, it holds that:

$$H(S_i|S_F, B_1, \ldots, B_m) = H(S_i),$$

where $S_F$ denotes $S_{t_1}, \ldots, S_{t_r}$.

The above condition establishes that a collusion of $t$ users does not give information about the personal key of another user.

**Self-Healing Property.**

1. Every $U_i \in G$, from broadcasts $B_r$ and $B_s$, where $1 \leq r < s \leq m$, can recover all keys $K_j$, for $j = r, \ldots, s$. Formally, it holds that:

$$H(K_r, \ldots, K_s|S_i, B_r, B_s) = 0.$$

The above condition characterizes the self-healing property: any two broadcasts are enough to recover all lost session keys, sent between session $r$ and session $s$.

**User Revocation.**

1. Given any subset $R = \{U_{t_1}, \ldots, U_{t_r}\} \subseteq \{U_1, \ldots, U_n\}$, where $|R| \leq t$, GM can generate a broadcast $B_j$ such that, for all $U_i \notin R$, the user $U_i$ can recover $K_j$ but the revoked users cannot. Formally, it holds that:

$$H(K_j|B_j, S_R) = H(K_j),$$

where $S_R$ denotes $S_{t_1}, \ldots, S_{t_r}$.

The above condition deals with the remove operations: the group manager GM is able to revoke users from the group.

**Forward-Backward Security.**

1. Let $g_1, g_2$ be integers such that $g_1 + g_2 = t$. A coalition of users $G_1 = \{U_{t_1}, \ldots, U_{t_{g_1}}\}$ of size $g_1$, removed before session $j_1$, and a coalition of users $G_2 = \{U_{p_1}, \ldots, U_{p_{g_2}}\}$ of size $g_2$, who join the scheme after session $j_2$, does not get any information about keys $K_j$, for any $j_1 < j < j_2$ Formally, it holds that:

$$H(K_j|B_1, \ldots, B_m, S_{G_1}, S_{G_2}) = H(K_j),$$

where $S_{G_1}$ denotes $S_{t_1}, \ldots, S_{t_{g_1}}$ and $S_{G_2}$ denotes $S_{p_1}, \ldots, S_{p_{g_2}}$.

The above condition states that the scheme is robust against collusion attacks, performed by groups of users composed of revoked users and new users.

Notice that the above model is a slightly simplified version⁠¹ of the one given in [19].

### 4. 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:

**Lemma 4.1** Let $X, Y, Z,$ and $W$ be four random variables. If $H(X|Y, W) = 0$ and $H(X|Z, W) = H(X)$, then $H(Y|Z) \geq H(X)$.

**Proof.** Notice that $I(Y; X|Z, W)$ can be written as:


Since from (1) $H(Y|ZW) \geq 0$, from the hypothesis $H(X|ZW) = H(X)$, and from (5) it holds that $H(X|ZW) \leq H(X|YW) = 0$, we get, applying again (5), that $H(Y|Z) \geq H(Y|ZW) \geq H(X)$.

Hence, the result holds.

Using the above lemma, we can show the following theorem (which gives the same result proven for the model in [19]):

**Theorem 4.2** In any self-healing key distribution scheme, for any $U_i \in G$, $H(S_i) \geq m \log q$.

**Proof.** Since $H(K_1, \ldots, K_m|B_1B_mB_S) = 0$ and $H(K_1, \ldots, K_m|B_mB_S) = H(K_1, \ldots, K_m)$, from Lemma 4.1 we get that $H(S_i) \geq H(K_1, \ldots, K_m)$.

From the independence of $K_1, \ldots, K_m$, it holds that

---

¹We do not use an intermediate random variable $Z$ in stating the conditions that a scheme has to satisfy and, as we will point out later on, we slightly weaken security requirement.
$\text{We have } H(K_1, \ldots, K_m) = H(K_1) + \ldots + H(K_m) = mH(K) = m \log q. \hfill \blacksquare$

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

Using close techniques, it is not difficult to show a weak lower bound on the size of the broadcast.

**Theorem 4.3** In any self-healing key distribution scheme, for any $j = 2, \ldots, m$,

$$H(B_j) \geq (j-1)\log q.$$ 

By applying a similar argument it is possible to show also that $H(B_1) \geq H(K) = \log q$. Hence, for any $j = 1, \ldots, m$, broadcast $B_j$ must be at least $j \log q$ bits long.

5. Constructions

In this section we describe some self-healing schemes. The first one does not support collusion attacks and user revocation, but these features can be easily built upon this basic protocol. The novelty of the following protocols, compared to the ones given in [19], lies in a different self-healing technique.

**Construction 1.** A self-healing session key distribution scheme not resilient to collusion attacks and without revocation capability.

**Set-up**

The group manager GM chooses $m$ values, say $h_1, \ldots, h_m$, and $m$ session keys, $K_1, \ldots, K_m \in F_q$, all at random. Then, for each $j = 1, \ldots, m$, he defines $z_j = h_j + K_j$. For $i = 1, \ldots, n$, user $U_i$ stores the personal key $S_i = \{h_1, \ldots, h_m\}$.

**Broadcast**

The group manager GM, in session $j \in \{3, \ldots, m\}$, broadcasts

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

The first two broadcasts are $B_1 = z_1$ and $B_2 = z_2$.

Notice that a new user $U_i$, can join the group at the $j$-th session: The group manager GM gives him an identifier in $F_q$ and sends him $S_i = \{h_j, \ldots, h_m\}$ as a personal key.

**Theorem 5.1** Construction 1 is a self-healing key distribution scheme not collusion resistant and without revocation capability.

**Proof (Sketch).** It is not difficult to see that, apart the conditions defining collusion resistance and user revocation, not supported by the above protocol, Construction 1 realizes a self-healing key distribution scheme. Indeed, about the reconstruction property, for all $i \in \{1, \ldots, n\}$, user $U_i$ recovers $K_j$ from the broadcast $B_j$, by evaluating $z_j - h_j$; while, either the broadcasts or the personal keys alone do not give any information about any key. Actually, notice that $K_j$ is protected in $B_j$ by $h_j$, and, intuitively, even if an adversary puts together two or more different pieces of $B_j$, say $p_{j-2} = z_1 + z_{j-2}$ and $p_{j-1} = z_1 + z_{j-1}$, and compute some function, like $p_{j-1} - p_{j-2} = z_j - z_{j-1}$, the difference $K_{j-2} - K_{j-1}$ is still protected by $h_{j-2} - h_{j-1}$, which is a random value. About the independence of future keys from the current available information, notice that without future broadcasts, there is no way to get information about new keys. On the other hand, the self-healing property holds since another broadcast message after $B_j$, say $B_{j+1}$, enables the user to compute $z_1 = (z_1 + z_j) - z_j$, by means of which the user can subsequently recover $z_{j+1}, \ldots, z_{j-1}$ and, hence, the whole sequence of session keys, $K_j, \ldots, K_r$. A detailed proof will be given in the journal version of the paper. \hfill \blacksquare

The above construction shows that Theorems 4.2 and 4.3 are tight in the special case in which collusion attacks are not supported and revocation operations are not provided.

Applying the same ideas presented in [19], we can enhance the above construction in order to be resilient to collusion attacks, and for providing revocation operations. More precisely, to gain resilience against collusion attacks, it is enough for the GM to generate $m$ polynomials $h_1(x), \ldots, h_m(x) \in F_q[x]$ of degree $t$, instead of $m$ values $h_1, \ldots, h_m \in F_q$. In such a way, the personal key of user $U_i$ becomes $S_i = \{h_1(i), \ldots, h_m(i)\}$, and $z_j$ is not a simple constant but a polynomial defined by $z_j(x) = K_j + h_j(x)$. User $U_i$ recovers $K_j$ by computing $z_j(i) - h_j(i)$. However, our construction has shorter broadcast size, compared to the protocol given in [19]: Indeed, $|B_j| = j t \log q$, instead of $m t \log q$.

We can use bivariate $t$-degree (in both variables) polynomials for providing also user revocation. Let us consider the following construction:

**Construction 2.** A self-healing session key distribution scheme resilient to collusion attacks and with revocation capability.

**Set-up**

Let $t > 0$, and let $N \in F_q$ such that $N \not\in \{1, \ldots, n\}$. The group manager GM chooses $m$ bivariate polynomials, of degree $t$ in both variables, say $s_1(x, y), \ldots, s_m(x, y) \in F_q[x,y]$, and $m$ session keys, $K_1, \ldots, K_m \in F_q$, all at random. Then, for each $j = 1, \ldots, m$, he defines $z_j(y) = \ldots$
Indeed, the values of each personal key are points of the broadcast messages and the private keys, enough points to recover any $K_j$.

In terms of memory storage and communication complexity, our construction requires to user $U_i$ to store a personal key $S_i$ of size $|S_i| = m$ (which is optimal with respect to Theorem 4.2), and broadcast size $|B_j| = j^t \log q + j^t \log q$. Hence, there is a substantial improvement compared to [19], where $|S_i| = m^2$ and $|B_j| = m^t \log q + m^t \log q$.

We stress that the main difference of our protocol, compared to the one given in [19], lies in a different implementation of the self-healing property: In [19] any user, belonging to $G$, from $B_{j_1}$ and $B_{j_2}$ can recover all keys $K_j$, for every $j_1 \leq j \leq j_2$. In our protocol, we allow from $B_{j_1}$ and $B_{j_2}$ to recover $K_j$ for any $j \leq j_2$.

Notice that the scheme given in [19] 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: $H(K_1, \ldots, K_{j-1}, K_j, K_{j+1}, \ldots, K_m | S_i, B_j) = H(K_1, \ldots, K_{j-1}, K_{j+1}, \ldots, K_m)$.

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. We believe that such a relaxation is not a problem in terms of security.

The above protocol enables GM to establish a session key with the users for $m$ different sessions. Along the same line of [19], we could set up a computationally secure long-lived protocol, by using a public generator $g$ of $F_q^*$, and by moving all the computation to the exponent. We skip the details from this abstract and refer the reader to [19] for details.

6. Conclusions

In this paper we have proposed a new mechanism for implementing self-healing key distribution. Moreover, we have described a secure and efficient construction, which is
optimal in terms of memory storage, and improve the communication complexity of the constructions given in [19]. Further research could be done in order to improve and meet (if possible), the lower bound on the broadcast size in such schemes. New techniques to implement self-healing key distribution, apart the one given in [19] and the one we have just described, represent another interesting target for researchers, due to the suitability of this approach to key distribution.

References


