Efficient Self-healing Key Distribution with Revocation for Wireless Sensor Networks Using One Way Key Chains

Ratna Dutta[1], Ee-Chien Chang[2], and Sourav Mukhopadhyay[2]

[1] Computing Division
Systems and Security Department
Institute for Infocomm Research
21, Heng Mui Keng Terrace, Singapore 119613
dratna@i2r.a-star.edu.sg
[2] School of Computing
National University of Singapore
3 Science Drive 2, Singapore 117543
{changec,sourav}@comp.nus.edu.sg

Abstract. Security of group communication for large mobile wireless sensor network hinges on efficient key distribution and key management mechanism. As the wireless medium is characterized by its lossy nature, reliable communication cannot be assumed in the key distribution schemes. Therefore, self-healing is a good property for key distribution in wireless applications. The main idea of self-healing key distribution scheme is that even if during a certain session some broadcast messages are lost due to network faults, the users are capable of recovering lost session keys on their own, without requesting additional transmission from the group manager. The only requirement for a user to recover the lost session keys, is its membership in the group both before and after the sessions in which the broadcast packets containing the keys are sent. Self-healing approach of key distribution is stateless in the sense that a user who has been off-line for some period is able to recover the lost session keys immediately after coming back on-line. In this paper, we propose two constructions for scalable self-healing key distribution with t revocation capability. The novelty of our constructions are that we apply a different and more efficient self-healing mechanism compared to the ones in the literature using one-way key chain. The main improvements that our proposed schemes achieve over previous approaches are

(a) communication bandwidth reduces from $O((tj + j - t - 1) \log q)$ to $O((t + 1) \log q)$, and
(b) computation costs for our first and second constructions reduce from $O(2tj + j)$ to $O(2t + 1)$ and $O(2(t^2 + t))$ respectively,

where m is the maximum number of sessions, j is the current session number, t is the maximum number of compromised group members that may collude and q is a large prime number. We achieve this result without any increase in the storage complexity. The schemes are scalable to very large groups in highly mobile, volatile and hostile network. We

J. Katz and M. Yung (Eds.): ACNS 2007, LNCS 4521, pp. 385–400, 2007.

prove in an appropriate security framework that our constructions are computationally secure and achieve both forward secrecy and backward secrecy.

Keywords: sensor network, session key distribution, self-healing, revocation, computational security.

1 Introduction

Secure group communication relies on secure and robust distribution of group keys. A single symmetric key known only to the group members can effectively protect a multicast group. However, only legitimate users should have access to the group communication in order to achieve privacy. Thus the group key (session key) must be updated each time when new users join or old users leave the group and securely redistributed to the existing members of the group. This is referred to as group rekeying. The newly joint users should not be able to derive the previous group keys, even if they are able to derive future group keys with subsequently distributed keying information. Similarly, the revoked users should not be able to derive the future session keys, even if they are able to compute the previous session keys with previously distributed keying information. If a group is rekeyed on each membership change, the frequency of rekeying becomes the primary bottleneck as the size of the group grows and/or the rate of membership change increases. Therefore, scalable group rekeying is an important and challenging problem to be addressed in order to support secure multicast communication for dynamic groups, where typical systems are large: tens of millions of users. How to distribute and update session key efficiently over an unreliable channel is an interesting research topic.

Self-Healing Key Distribution: In this paper, we address *self-healing key distribution scheme with revocation* [22] that deals with the problem of distributing session keys for secure communication to a dynamic group of users over an unreliable, lossy network in a manner that is resistant to packet lost and collusion attacks. The main concept of self-healing key distribution schemes is that users, in a large and dynamic group communication over an unreliable network, can recover lost session keys on their own, even if lost some previous key distribution messages, without requesting additional transmissions from the group manager. This reduces network traffic and risk of user exposure through traffic analysis and also decreases the work load on the group manager. The key idea of self-healing key distribution schemes is to broadcast information that is useful only for trusted members. Combined with its pre-distributed secrets, this broadcast information enables a trusted member to reconstruct a shared key. On the contrary, a revoked member is unable to infer useful information from the broadcast. The only requirement that a user must satisfy to recover the lost keys through self-healing, is its membership in the group both before and after the sessions in which the broadcast packet containing the key is sent. A user who has been off-line for some period is able to recover the lost session keys immediately after coming back on-line. Thus self-healing approach of key distribution is stateless. The scheme is said to have t-revocation capability if the key distribution mechanism cannot be broken by any coalition of up to t users.

Our Contribution: This paper focuses on designing computationally secure and efficient key distribution schemes with self-healing property and revocation capability for large and dynamic groups over insecure wireless networks. We propose two new constructions for self-healing key distributions adopting a new self-healing technique. The novelty of our self-healing approach is that it uses one-way key chain that is more efficient compared to the self-healing techniques used in the previous schemes [4, 12, 14, 22]. This yields an improved secret-sharing based self-healing key distribution scheme (Construction 2) compared to the secret-sharing based scheme in [4]. We obtain further improvement (Construction 1) using polynomial based revocation which is more efficient compared to all the previous scheme. The main attraction of this paper is that our constructions have significant improvements in terms of both communication and computation overhead without any increase in the storage complexity. Table 1 summarizes the comparison of our schemes with the previous approaches (m being the maximum number of sessions, j stands for the current session number and q is a prime large enough to accommodate a cryptographic key).

Table 1. Comparison among different self-healing key distribution schemes in j-th session

Schemes	Storage Overhead	Communication Overhead	Computation Overhead
Construction 3 of [22]	$(m-j+1)^2 \log q$	$(mt^2+2mt+m+t)\log q$	$2mt^2+3mt-t$
Scheme 3 of [14]	$2(m-j+1)\log q$	$[(m+j+1)t+(m+1)]\log q$	$mt+t+2tj+j$
Scheme 2 of [4]	$(m-j+1)\log q$	$(2tj+j)\log q$	$2j(t^2+t)$
Construction 1 of [12]	$(m-j+1)\log q$	$(tj+j-t-1)\log q$	$2tj+j$
Our Construction 1	$(m-j+1)\log q$	$(t+1)\log q$	$2t+1$
Our Construction 2	$(m-j+1)\log q$	$(t+1)\log q$	$2(t^2+t)$

We emphasize that each user in both the proposed constructions requires $(m-j+1)\log q$ memory and size of the broadcast message at the j-th session is $(t+1)\log q$ with computation costs $2t+1$ and $2(t^2+t)$ respectively. Our key distribution schemes are scalable to very large groups in highly mobile, volatile and hostile wireless network as the communication and computation overhead does not depend on the size of the group, instead they depend on the number of compromised group members that may collude together. We have shown in an appropriate security model that our proposed constructions are computationally secure and achieve both forward secrecy and backward secrecy.

Related Works: Broadcast encryption is a closely related area which has received much attention from both the network and cryptography community. Efficient key distribution and key management mechanisms are at the core of this. The area of broadcast encryption was formally defined by Fiat and Naor [10] after the work of Berkovits [1] and has been extensively studied since then. A number of approaches have been proposed and has grown up in different directions: rekeying schemes for dynamic groups, broadcast schemes with tracing capability, users revocation from a predefined subset of users *etc.* A few of them are [2, 3, 5, 6, 7, 8, 10, 11, 13, 15, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27]. However, the underlying networks are assumed to be reliable in all the above works. Self-healing key distribution with revocation was first introduced by Staddon *et al.* in [22]. They provide formal definitions and security notions that were later generalized by Liu *et al.* [14] and Blundo *et al.* [4]. The constructions given in [22] suffers from high storage and communication overhead. Liu *et al.* [14] introduced a novel personal key distribution scheme and combining it with the self-healing technique in [22], they proposed a new construction that improves the storage and communication overhead greatly. Blundo *et al.* [4] showed an attack to the first construction in [22] and developed a new self-healing technique different from [22] under a slightly modified framework. More recently, Hong *et al.* [12] proposed self-healing key distribution constructions having less storage and communication complexity. Recently, Dutta and Mukhopadhyay [9] proposed a new storage efficient self-healing key distribution scheme.

Applications: The spectrum of applicability of self-healing key distribution is quite large. Self-healing key distribution is a potential candidate to establish session keys for secure communication to large and dynamic groups in highly mobile, volatile and hostile wireless network, where frequent membership changes may be necessary and ability to revoke users during certain exchanges is desirable. In such situations the session keys need to be used for a short time-period or need to be updated frequently. Mobile wireless ad hoc networks have wide applications in military operations, rescue missions and scientific explorations, where there are usually no network infrastructure support and the adversary may intercept, modify, and/or partially interrupt the communication. In such applications, security becomes a critical concern. The traditional approaches for key distribution and group re-keying used for reliable network, are not suitable for large and dynamic wireless networks because of the lossy nature of wireless medium. Therefore, self-healing is a good property for key distribution in wireless mobile and ad hoc networks, where the nodes/devices are powered by batteries and have the unique feature of moving in and out of range frequently. Hence expensive computations like the ones required by public key cryptography are not suitable for such networks. For example, military networks consist of mobile devices carried by soldiers, automatic weapons, sensing devices *etc.* and there could be a need in a battle field for a rapid revocation of devices caught by the enemy. Also there might be situations where some users are not constantly on-line or experience burst packet losses. It can rejoin the group once the power is on again. All these aspects can take great advantage from self-healing key distribution schemes with revocation capability. Self-healing key distribution schemes have also found applicable in broadcast communication over low-cost channels, pay-per-view

TV, information service delivering sensitive content/information to authorized recipients and several other Internet-related settings.

Organization: The rest of the paper is organized as follows. Section 2 presents notations to be used in the paper and our security model. Section 3 describes the details of our constructions. We provide a proof of security of our proposed schemes in Section 4. Section 5 focuses on the performance analysis of the schemes and their comparison with the previous works. Finally, we conclude in Section 6.

2 Preliminaries

This section briefly define our security model for self-healing key distribution.

Table 2. Notations

$\mathcal{U}$	: set of all users in the networks
U_i	: i-th user
GM	: group manager
n	: total number of users in the network
m	: total number of sessions
t	: the maximum number of compromised user
F_q	: a field of order q
S_i	: personal secret of user U_i
SK_j	: session key generated by the GM in session j
$\mathcal{B}_j$	: broadcast message by the GM during session j
$Z_{i,j}$	: the information learned by U_i through $\mathcal{B}_j$ and S_i
R_j	: the set of all revoked users in session j
$\mathcal{H}$	: a cryptographically secure one-way function
S^F	: forward key seed generated by the GM
S^B	: backward key seed generated by the GM
K_i^F	: i-th forward key in the forward key chain
K_i^B	: i-th backward key in the backward key chain

2.1 Our Security Model

We now state the following definitions that are aimed to computational security for session key distribution adopting the security model of [14, 22].

Definition 2.1 *(Session Key Distribution with privacy [22]) Let* $t, i \in \{1, \ldots, n\}$ *and* $j \in \{1, \ldots, m\}$.

1. $\mathcal{D}$ *is a session key distribution with privacy if*

   (a) for any user U_i*, the session key* SK_j *is efficiently determined from* $\mathcal{B}_j$ *and* S_i.

   (b) for any set $R \subseteq \mathcal{U}$, $|R| \leq t$*, and* $U_i \notin R$*, it is computationally infeasible for users in* R *to determine the personal key* S_i.

(c) *what users $U_1, \ldots, U_n$ learn from $\mathcal{B}_j$ cannot be determined from broadcasts or personal keys alone.* i.e. *if we consider separately either the set of m broadcasts $\{\mathcal{B}_1, \ldots, \mathcal{B}_m\}$ or the set of n personal keys $\{S_1, \ldots, S_n\}$, then it is computationally infeasible to compute session key SK_j (or other useful information) from either set.*

2. *$\mathcal{D}$ has t-revocation capability if given any $R \subseteq \mathcal{U}$, where $|R| \leq t$, the group manager GM can generate a broadcast $\mathcal{B}_j$, such that for all $U_i \notin R$, U_i can efficiently recover the session key SK_j, but the revoked users cannot.* i.e. *it is computationally infeasible to compute SK_j from $\mathcal{B}_j$ and $\{S_l\}_{U_l \in R}$.*
3. *$\mathcal{D}$ is self-healing if the following is true for any j, $1 \leq j_1 < j < j_2 \leq m$: For any user U_i who is a member in sessions j_1 and j_2, the key SK_j is efficiently determined by the set $\{Z_{i,j_1}, Z_{i,j_2}\}$.*

Definition 2.2 *(t-wise forward and backward secrecy [14]) Let $t, i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.*

1. *A key distribution scheme $\mathcal{D}$ guarantees t-wise forward secrecy if for any set $R \subseteq \mathcal{U}$, where $|R| \leq t$, and all $U_l \in R$ are revoked before session j, it is computationally infeasible for the members in R together to get any information about SK_j, even with the knowledge of group keys $\mathsf{SK}_1, \ldots, \mathsf{SK}_{j-1}$ before session j.*
2. *A session key distribution $\mathcal{D}$ guarantees t-wise backward secrecy if for any set $J \subseteq \mathcal{U}$, where $|J| \leq t$, and all $U_l \in J$ join after session j, it is computationally infeasible for the members in J together to get any information about K_j, even with the knowledge of group keys $\mathsf{SK}_{j+1}, \ldots, \mathsf{SK}_m$ after session j.*

3 Our Constructions

In this section, we present two constructions for self-healing key distribution with revocation capability. We use revocation polynomial in our first construction and apply secret sharing in our second construction. Unlike previous approaches, we adopt a different technique to perform self-healing which is more efficient from both communication and computation point of view compared to the previous techniques.

We consider a setting in which there is a group manager (GM) and n users $\mathcal{U} = \{U_1, \ldots, U_n\}$. All of our operations take place in a finite field, F_q, where q is a large prime number ($q > n$). In our setting, we never allow a revoked user to rejoin the group in a later session. Let $\mathcal{H} : F_q \longrightarrow F_q$ be a cryptographically secure one-way function.

3.1 Construction 1: Revocation Using Polynomial

- *Setup*: The group manager randomly picks two initial key seeds, the forward key seed $S^F \in F_q$ and the backward key seed $S^B \in F_q$. It repeatedly applies (in the pre-processing time) the one-way function $\mathcal{H}$ on S^B and computes the one-way key chain of length m:

$$K_i^B = \mathcal{H}(K_{i-1}^B) = \mathcal{H}^{i-1}(S^B)$$

for $1 \leq i \leq m$. The j-th session key is computed as

$$\mathsf{SK}_j = K_j^F + K_{m-j+1}^B,$$

where $K_j^F = \mathcal{H}^{j-1}(S^F)$. The group manager chooses independently and uniformly at random m t-degree polynomials $f_1(x), \ldots, f_m(x) \in F_q[x]$, $t < m, n$. Each user U_i, for $1 \leq i \leq n$, receives its personal secret keys corresponding to the m sessions $S_i = \{f_1(i), \ldots, f_m(i)\}$ and the forward key seed S_F from the group manager via the secure communication channel between them.

- *Broadcast*: Let $R_j = \{U_{l_1}, \ldots, U_{l_{w_j}}\}$ be the set of all revoked users for sessions in and before j such that $|R_j| = w_j \leq t$. In the j-th session the group manager locates the backward key K_{m-j+1}^B in the backward key chain and computes the polynomials

$$r_j(x) = (x - l_1) \cdots (x - l_{w_j}),$$

$$h_j(x) = K_{m-j+1}^B r_j(x) + f_j(x).$$

The polynomial $r_j(x)$ is called the revocation polynomial in session j and the polynomial $f_j(x)$ plays the role of masking polynomial in session j. The group manager broadcasts the following message $\mathcal{B}_j$:

$$\mathcal{B}_j = R_j \cup \{h_j(x)\}.$$

- *Session Key Recovery*: When a non-revoked user U_i receives the j-th session key distribution message $\mathcal{B}_j$, it evaluates the polynomial $r_j(x)$ at point i and recovers

$$K_{m-j+1}^B = \frac{h_j(i) - f_j(i)}{r_j(i)}.$$

Finally, U_i computes the j-th forward key $K_j^F = \mathcal{H}^{j-1}(S^F)$ and evaluates the current session key

$$\mathsf{SK}_j = K_j^F + K_{m-j+1}^B.$$

- *Add Group Members*: When the group manager adds a new group member starting from session j, it picks an unused identity $v \in F_q$, computes the personal secret keys corresponding to the current and future sessions $S_v = \{f_j(v), f_{j+1}(v), \ldots, f_m(v)\}$ and gives $\{v, S_v, K_j^F\}$ to this new group member via the secure communication channel between them.
- *Re-initialization:* The system fails when all m sessions are exhausted, or when number of revoked users becomes more than t. At this phase, re-initialization is required and a new setup is executed.

Complexity:

Storage overhead: Storage complexity of personal key for each user is $m \log q$ bits. The group members that join later need to store less data. Foe example, the personal key for a user joining in j-th session occupies $(m - j + 1) \log q$ bits memory space.

Communication overhead: Communication bandwidth for key management is $(t+1)\log q$ bits. Here we ignore the communication overhead for the set of identities of revoked users, as these identities of revoked users can be picked from a small finite field [12].

3.2 Construction 2: Revocation Using Secret Sharing

- *Setup*: Let t be a positive integer. The group manager GM chooses independently and uniformly at random m polynomials $f_1(x), \ldots, f_m(x) \in F_q[x]$, each of degree t. The group manager randomly picks two initial key seeds, the forward key seed $S^F \in F_q$ and the backward key seed $S^B \in F_q$. It repeatedly applies (in the pre-processing time) the one-way function $\mathcal{H}$ on S^B and computes the one-way backward key chain of length m:

$$K_i^B = \mathcal{H}(K_{i-1}^B) = \mathcal{H}^{i-1}(S^B) \text{ for } 1 \leq i \leq m.$$

  The j-th session key is computed as

$$\mathsf{SK}_j = K_j^F + K_{m-j+1}^B,$$

  where $K_j^F = \mathcal{H}^{j-1}(S^F)$.

  Each user U_i, for $1 \leq i \leq n$, receives its personal secret keys corresponding to the m sessions $S_i = \{f_1(i), \ldots, f_m(i)\}$ and the forward key seed S_F from the group manager via the secure communication channel between them.

- *Broadcast*: Let R_j be the set of all revoked users for sessions in and before j such that $|R_j| \leq t$ and G_j be the set of all non-revoked users in session j. In the j-th session the GM first chooses a set of indices (different from 0) $W_j = \{x_{1,j}, \ldots, x_{t,j}\}$ such that $I_{R_j} \subseteq W_j$, but $W_j \cap I_{G_j} = \emptyset$, where I_{R_j} represents the indices of the users in R_j, I_{G_j} denotes the set of indices of users in G_j and $\emptyset$ is the empty set. The GM then computes $Z_j = K_{m-j+1}^B + f_j(0)$ and broadcasts the following message $\mathcal{B}_j$:

$$\mathcal{B}_j = \{x_{1,j}, \ldots, x_{t,j}; f_j(x_{1,j}), \ldots, f_j(x_{t,j}); Z_j\}.$$

- *Session Key Recovery and Message Recovery*: When a non-revoked user U_i receives the j-th session key distribution message $\mathcal{B}_j$, it interpolates $\{(x_{l,j}, f_j(x_{l,j}))\}_{l=1,\ldots,t}$ and $(i, f_j(i))$ to recover $f_j(0)$ by Lagrange's interpolation formula as follows:

$$f_j(0) = \sum_{l=0}^{t} \Lambda_l f_j(x_{l,j}),$$

  where

$$\Lambda_l = \prod_{\substack{k=0 \\ k \neq l}}^{t} \frac{-x_{k,j}}{x_{l,j} - x_{k,j}}$$

  with $x_{0,j} = i$. Then U_i recovers the key K_{m-j+1}^B as

$$K_{m-j+1}^B = Z_j - f_j(0).$$

Finally, U_i computes the j-th forward key $K_j^F = \mathcal{H}^{j-1}(S^F)$ and evaluates the current session key

$$\mathsf{SK}_j = K_j^F + K_{m-j+1}^B.$$

- *Add Group Members*: When the group manager adds a new group member starting from session j, it picks an unused identity $v \in F_q$, computes the personal secret keys corresponding to the current and future sessions $S_v = \{f_j(v), f_{j+1}(v), \ldots, f_m(v)\}$ and gives $\{v, S_v, K_j^F\}$ to this new group member via the secure communication channel between them.
- *Re-initialization:* The system fails when all m sessions are exhausted, or when number of revoked users becomes more than t. At this phase, re-initialization is required and a new setup is executed.

Complexity:

Storage overhead: Storage complexity of personal key for each user is $m \log q$ bits. The group members that join later need to store less data. Foe example, the personal key for a user joining in j-th session occupies $(m - j + 1) \log q$ bits memory space.

Communication overhead: Communication bandwidth for key management is $(t + 1) \log q$ bits. Here we ignore the communication overhead for the broadcast of points $x_{l,j}$ for $l = 1, \ldots, t$, as these identities can be picked from a small finite field.

3.3 Self-healing

We now explain our self-healing mechanism in the above constructions: Let U_i be a group member that receives session key distribution messages $\mathcal{B}_{j_1}$ and $\mathcal{B}_{j_2}$ in sessions j_1 and j_2 respectively, where $1 \le j_1 \le j_2$, but not the session key distribution message $\mathcal{B}_j$ for session j, where $j_1 < j < j_2$. User U_i can still recover all the lost session keys K_j for $j_1 < j < j_2$ as follows:

(a) U_i recovers from the broadcast message $\mathcal{B}_{j_2}$ in session j_2, the backward key $K_{m-j_2+1}^B$ and repeatedly apply the one-way function $\mathcal{H}$ on this and computes the backward keys K_{m-j+1}^B for all j, $j_1 \le j < j_2$.

(b) U_i computes the forward keys K_j^F for all j, $j_1 \le j \le j_2$ by repeatedly applying $\mathcal{H}$ on the forward seed S^F or on the forward key $K_{j_1}^F$ of the j_1-th session.

(c) U_i then recovers all the session keys $\mathsf{SK}_j = K_j^F + K_{m-j+1}^B$, for $j_1 \le j \le j_2$.

Note that a user revoked in session j cannot compute the backward keys $K_{m-j_1+1}^B$ for $j_1 > j$, although it can compute the forward keys $K_{j_1}^F$. As a result, revoked users cannot compute the subsequent session keys SK_{j_1} for $j_1 > j$, as desired.

Similarly, a user U_i joined in session j cannot compute the forward keys $K_{j_2}^F$ for $j_2 < j$ as U_i knows only the j-th forward key K_j^F, not the initial forward seed value S^F, although it can compute the backward keys $K^B_{m-j_2+1}$ for $j_2 < j$. This forbids U_i to compute the previous session keys as desired.

4 Security Analysis

In this section we show that our Constructions realizes self-healing key distribution schemes with revocation capability. More precisely, we can prove Theorem 4.1 and Theorem 4.2 which state the security result of Construction 1 and Construction 2 respectively in our security model described in Section 2.1

Theorem 4.1 *Construction 1 is secure, self-healing session key distribution scheme with privacy, t-revocation capability with respect to Definition 2.1 and achieve t-wise forward and backward secrecy with respect to Definition 2.2.*

Proof: Our goal is security against coalition of size at least t. We will show that the Construction 1 is computationally secure with respect to revoked users assuming the difficulty of inverting one-way function, *i.e.* for any session j it is computationally infeasible for any set of revoked users of size at most t before and on session j to compute with non-negligible probability the session key SK_j, given the View consisting of personal keys of revoked users, broadcast messages before, on and after session j and session keys of revoked users before session j.

Consider a coalition of t revoked users, say $U_1, U_2, \ldots U_t$, who are revoked on or before the j-th session. The revoked users are not entitled to know the j-th session key SK_j. We can model this coalition of t users as a polynomial-time algorithm $\mathcal{A}'$ that takes View as input and outputs its guess for SK_j.We say that $\mathcal{A}'$ is successful in breaking the construction if it has a non-negligible advantage in determining the session key SK_j. Then using $\mathcal{A}'$, we can construct a polynomial-time algorithm $\mathcal{A}$ for inverting one-way function $\mathcal{H}$ and have the following claim:

Claim: $\mathcal{A}$ inverts one-way function $\mathcal{H}$ with non-negligible probability if $\mathcal{A}'$ is successful.

Proof: Given any instance $y = \mathcal{H}(x)$ of one-way function $\mathcal{H}$, $\mathcal{A}$ first generates an instance View for $\mathcal{A}'$ as follows: $\mathcal{A}$ randomly selects a forward key seed $S^F \in F_q$ and constructs the following backward key chain by repeatedly applying $\mathcal{H}$ on y:

$$K^B_1 = y, K^B_2 = \mathcal{H}(y), K^B_3 = \mathcal{H}^2(y), \ldots, K^B_j = \mathcal{H}^{j-1}(y), \ldots, K^B_m = \mathcal{H}^{m-1}(y).$$

$\mathcal{A}$ computes the j-th forward key $K^F_j = \mathcal{H}^{j-1}(S^F)$ and sets the j-th session key

$$\mathsf{SK}_j = K^F_j + K^B_{m-j+1}.$$

$\mathcal{A}$ chooses at random m polynomials $f_1(x), \ldots, f_m(x) \in F_q[x]$, each of degree $t < n$. Each user U_i, for $1 \leq i \leq n$, receives its personal secret keys corresponding to the m sessions $S_i = \{f_1(i), \ldots, f_m(i)\}$ and the forward key seed S_F from $\mathcal{A}$ via the secure communication channel between them. For $1 \leq j \leq m$, $\mathcal{A}$ computes broadcast message $\mathcal{B}_j$ as:

$$\mathcal{B}_j = R_j \cup \{h_j(x)\}$$

where $R_j = \{U_1, \ldots, U_t\}$ is the set of revoked users for sessions in and before j such that $R_{j-1} \subseteq R_j$ for $2 \leq j \leq m$, and

$$h_j(x) = K^B_{m-j+1} r_j(x) + f_j(x), \text{ with } r_j(x) = (x-1)\cdots(x-t).$$

Then $\mathcal{A}$ sets View as

$$\mathsf{View} = \begin{Bmatrix} f_j(1), \ldots, f_j(t) \text{ for } j = 1, \ldots, m, \\ \mathcal{B}_j \text{ for } j = 1, \ldots, m, \\ S^F, \\ \mathsf{SK}_1, \ldots, \mathsf{SK}_{j-1} \end{Bmatrix}$$

$\mathcal{A}$ gives View to $\mathcal{A}'$, which in turn selects $X \in F_q$ randomly, sets the j-th session key to be $\mathsf{SK}'_j = K^F_j + X$ and returns SK'_j to $\mathcal{A}$. $\mathcal{A}$ checks whether $\mathsf{SK}'_j = \mathsf{SK}_j$. If not, $\mathcal{A}$ chooses a random $x' \in F_q$ and outputs x'.

$\mathcal{A}'$ can compute the j-th forward key $K^F_j = \mathcal{H}(S^F)$ as it knows S^F from View for $j = 1, \ldots, m$. Note that from View, $\mathcal{A}'$ knows at most t points on the t-degree polynomial $f_j(x)$ and at most $j-1$ session keys $\mathsf{SK}_1, \ldots, \mathsf{SK}_{j-1}$. Consequently $\mathcal{A}'$ has knowledge of at most $j-1$ backward keys $K^B_m, \ldots, K^B_{m-j+2}$. Observe that $\mathsf{SK}'_j = \mathsf{SK}_j$ provided $\mathcal{A}'$ knows the backward key K^B_{m-j+1}. This occurs if either of the following two holds:

(a) $\mathcal{A}'$ is able to compute the t-degree polynomial $f_j(x)$ from View and consequently can recover the backward key K^B_{m-j+1} as follows:

$$K^B_{m-j+1} = \frac{h_j(i) - f_j(i)}{r_j(i)},$$

where $i \neq 1, \ldots, t$. Note that $r_j(i) = 0$ for $i = 1, \ldots, t$.

From View, $\mathcal{A}'$ knows only t-points on the t-degree polynomial $f_j(x)$ and will not be able to compute $f_j(x)$. Consequently, $\mathcal{A}'$ will not be able to recover K_{m-j+1} from $\mathcal{B}_j$ as described in (a) above.

(b) $\mathcal{A}'$ is able to choose $X \in F_q$ so that the following relations hold:

$$K^B_m = \mathcal{H}^{j-1}(X), K^B_{m-1} = \mathcal{H}^{j-2}(X), \ldots, K^B_{m-j+2} = \mathcal{H}(X)$$

This occurs with a non-negligible probability only if $\mathcal{A}$ is able to invert the one-way function $\mathcal{H}$. In that case, $\mathcal{A}$ returns $x = \mathcal{H}^{-1}(y)$.

The above arguments show that if $\mathcal{A}'$ is successful in breaking the security of Construction 1, then $\mathcal{A}$ is able to invert the one-way function. □
(of claim)

Hence Construction 1 is computationally secure under the hardness of inverting one-way function. We will now show that Construction 1 satisfies all the conditions required by Definition 2.1.

1) (a) Session key efficiently recovered by a non-revoked user U_i is described in the third step of our Construction 1.

(b) For any set $R \subseteq \mathcal{U}$, $|R| \leq t$, and any non-revoked user $U_i \notin R$, we show that the coalition R knows nothing about the personal secret S_i of U_i. For any session j, U_i's personal secret $S_i = f_j(i)$ is a point over a t-degree polynomial $f_j(x)$. Since the coalition R gets at most t points over the t-degree polynomial $f_j(x)$, it is computationally infeasible for coalition R to learn $f_j(i)$ for $U_i \notin R$.

(c) The j-th session key $\mathsf{SK}_j = K_j^F + K_{m-j+1}^B$, where $K_j^F = \mathcal{H}(K_{j-1}^F) = \mathcal{H}^{j-1}(S^F)$, $K_j^B = \mathcal{H}(K_{j-1}^B) = \mathcal{H}^{j-1}(S^B)$, S^F is the forward seed value given to all initial group members and S^B is the secret backward seed value. Thus $\mathsf{SK_j}$ is independent of the personal secret $S_i = f_j(i)$ for $i = 1, \ldots, n$. So the personal secret keys alone do not give any information about any session key. Since the initial backward seed S^B is chosen randomly, the backward key K_{m-j+1}^B and consequently the session key SK_j is random as long as S^B, $K_1^B, K_2^B, \ldots, K_{m-j+2}^B$ are not get revealed. This in turn implies that the broadcast messages alone cannot leak any information about the session keys. So it is computationally infeasible to determine $Z_{i,j}$ from only personal key S_i or broadcast message $\mathcal{B}_j$.

2) (t-revocation property) Let R be a collection of t-revoked users collude in session j. It is impossible for coalition R to learn the j-th session key SK_j because knowledge of SK_j implies the knowledge of either the backward key K_{m-j+1}^B or the knowledge of the personal secret $f_j(i)$ of user $U_i \notin R$. The coalition R knows the points $\{f_j(i) : U_i \in R\}$. The size of the coalition R is at most t. Consequently, the colluding users only have at most t-points on the polynomial $f_j(x)$. But degree of the polynomial $f_j(x)$ is t. Hence the coalition R cannot recover $f_j(x)$, which in turn makes K_{m-j+1}^B appears random to R. Therefore, SK_j is completely safe to R from computation point of view.

3) (Self-healing property) From the third step of our Construction 1, any user U_i that is a member in sessions j_1 and j_2 $(1 \leq j_1 < j_2)$, can recover the backward key $K_{m-j_2+1}^B$ and hence can obtain the sequence of backward keys $K_{m-j_1}^B, \ldots, K_{m-j_2+2}^B$ by repeatedly applying $\mathcal{H}$ on $K_{m-j_2+1}^B$. User U_i also holds the forward key $K_{j_1}^F = \mathcal{H}^{j_1-1}(S^F)$ of the j_1-th session and hence can obtain the sequence of forward keys $K_{j_1+1}^F, \ldots, K_{j_2-1}^F$ by repeatedly applying $\mathcal{H}$ on $K_{j_1}^F$. Hence, as shown in Section 3.3, user U_i can efficiently recover all missed session keys.

We will show the Construction 1 satisfies all the conditions required by Definition 2.2.

1) (t-wise forward secrecy) Let $R \subseteq \mathcal{U}$, where $|R| \leq t$ and all user $U_l \in R$ are revoked before the current session j. The coalition R can not get any information about the current session key SK_j even with the knowledge of group keys before session j. This is because of the fact that in order to know SK_j, $U_l \in R$ needs to know at least $t+1$ points on the polynomial $f_j(x)$. Since

size of the coalition R is at most t, the coalition R has at most t personal secrets $f_j(i)$, *i.e.* gets t points on the polynomial $f_j(x)$. But at least $t+1$ points are needed on the polynomial $f_j(x)$ to recover the current session key SK_j for any user $U_l \in R$. Besides, because of the one-way property of $\mathcal{H}$, it is computationally infeasible to compute $K^B_{j_1}$ from $K^B_{j_2}$ for $j_1 < j_2$. The users in R might know the sequence of backward keys $K^B_m, \ldots, K^B_{m-j+2}$, but cannot compute K^B_{m-j+1} and consequently SK_j from this sequence. Hence the Construction 1 is t-wise forward secure.

2) (t-wise backward secrecy) Let $J \subseteq \mathcal{U}$, where $|J| \leq t$ and all user $U_l \in J$ join after the current session j. The coalition J can not get any information about any previous session key SK_{j_1} for $j_1 \leq j$ even with the knowledge of group keys after session j. This is because of the fact that in order to know SK_{j_1}, $U_l \in J$ requires the knowledge of j_1-th forward key $K^F_{j_1} = \mathcal{H}(K^F_{j_1-1}) = \mathcal{H}^{j_1-1}(S^F)$. Now when a new member U_v joins the group starting from session $j+1$, the GM gives $(j+1)$-th forward key K^F_{j+1} instead of the initial forward key seed S^F, together with the values $f_{j+1}(v), \ldots, f_m(v)$. Note that $K^F_{j+1} = \mathcal{H}(K^F_j)$. Hence it is computationally infeasible for the newly joint member to trace back for previous forward keys $K^F_{j_1}$ for $j_1 \leq j$ because of the one-way property of the function $\mathcal{H}$. Consequently, our protocol is t-wise backward secure. In fact, this backward secrecy is independent of t. □

A similar result holds for our Construction 2 and we can prove Theorem 4.2 stated below following the same line of proving Theorem 4.1.

Theorem 4.2 *Construction 2 is secure, self-healing session key distribution scheme with privacy, t-revocation capability with respect to Definition 2.1 and achieves t-wise forward and backward secrecy with respect to Definition 2.2.*

5 Performance Analysis

Comparison of storage overhead, communication complexity and computation cost of each user (not the GM) in our constructions with the existing self-healing session key distribution schemes is provided in Table 1 (see Introduction). It is demonstrated in Table 1 that our proposed constructions are more efficient than the previous schemes. In particular, our Construction 1 is the most efficient key distribution scheme with self-healing and revocation property among all the previous approaches. In one hand our constructions reduce the communication complexity (bandwidth) to $O(t)$, whereas optimal communication complexity achieved by the previous schemes is $O(tj)$ at the j-th session. Achieving less computation cost is on the other side of the coin. For a user U_i at the j-th session, the computation cost is incurred by recovering all previous session keys upto the j-th session (worst case) by self-healing mechanism. The backward key used at the j-th session in our Construction 1 is $K^B_{m-j+1} = \frac{h_j(i)-f_j(i)}{r_j(i)}$.

Consequently, user U_i needs to computes two points $h_j(i), r_j(i)$ on the polynomials $h_j(x)$ and $r_j(x)$ which require at most $2t$ multiplication operations. Since division can be regarded as multiplication, total number of multiplication operations required to get K^B_{m-j+1} is $2t+1$. After obtaining K^B_{m-j+1}, user U_i can easily compute $K^B_{m-j+2}, K^B_{m-j+3}, \ldots, K^B_{m-1}, K^B_m$ by applying the one-way function $\mathcal{H}$ each time. Then U_i is able to compute all previous session keys $\mathsf{SK}_{j_1} = K^F_{j_1} + K^B_{m-j_1+1}$ for all $1 \leq j_1 \leq j$. Thus the computation cost for each user is $2t+1$, whereas the computation complexity of the revocation polynomial based scheme in [12] is $j(2t+1)$. Similarly, for Construction 2 the computation complexity is $2\{(t+1)^2 - (t+1)\} = 2(t^2+t)$ which is the number of multiplication operations needed to recover a t degree polynomial by using Lagrange formulation. Thus the communication complexity and computation cost in our constructions do not increase as the number of session grows. These are the most prominent improvement of our schemes over the previous works. The storage overhead of each user for personal key in both our self-healing key distribution schemes is $O((m-j+1)\log q)$, which is same as that of [4, 12].

Remark and Future Work: Our security model excludes the following property of self-healing key distribution unlike the security model provided by [14, 22]: Let $1 \leq j_1 < j < j_2 \leq m$. For any disjoint subsets $L_1, L_2 \subset \mathcal{U}$, where $|L_1 \cup L_2| \leq t$, no information about the session key SK_j, $j_1 < j < j_2$ can be obtained by the coalition $L_1 \cup L_2$, where the set L_1 is a coalition of users removed before session j_1 and the set L_2 is a coalition of users joined from session j_2. Our protocol does not satisfy this property as illustrated by the following simple example: Let $L_1 = \{U_3\}, L_2 = \{U_6\}$ and $j_1 = 2, j_2 = 5$. The above property states that U_3 and U_6 jointly should not be able to know SK_j, $j = 3, 4$. But U_3 knows K^F_2 and U_6 knows K^B_{m-5+1}. Consequently, U_2 can compute K^F_3, K^F_4 and U_6 can compute K^B_{m-4+1}, K^B_{m-3+1}. Hence, U_3 and U_6 together can compute $\mathsf{SK}_j = K^F_j + K^B_{m-j+1}, j = 3, 4$. As a future work we are interested to incorporate this property in our scheme.

6 Conclusion

In this paper, we develop and analyze two efficient computationally secure self-healing key distribution schemes with revocation capability, enabling a very large and dynamic group of users to establish a common key for secure communication over an insecure wireless network. We introduce a novel self-healing mechanism for session key-recovery on possible packet lost in the lossy environment using one-way key chain. Our proposed key distribution mechanism significantly improves the communication and computation costs over the previous approaches without any increase in the storage complexity. The schemes are properly analyzed in an appropriate security model to prove that they are computationally secure and achieve both forward secrecy and backward secrecy.

References

[1] S. Berkovit. *How to Broadcast a Secret.* Advances in Cryptology, Eurocrypt'91, LNCS 547, pp. 536-541, Springer-Verlag, 1991.

[2] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung. *Perfectly-Secure Key Distribution for Dynamic Conferences.* In Crypto'92, LNCS 740, pp. 471-486, Springer-Verlag, 1993.

[3] C. Blundo, L. F. Mattos, D. Stinson. *Trade-offs between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution.* Advances in Cryptology, Crypto'96, LNCS 1109, pp. 387-400, Springer-Verlag, 1996.

[4] C. Blundo, P. D'Arco, A. Santis, M. Listo. *Design of Self-healing Key Distribution Schemes.* Design Codes and Cryptology, N. 32, pp. 15-44, 2004.

[5] C. Blundo, P. D'Arco, A. Santis, M. Listo. *Definitions and Bounds for Self-healing Key Distribution.* Fuzzy Sets and Systems, 31st International Colloquium on Automata, Languages and Programming ICALP 04, LNCS 3142, pp. 234-245, Springer-Verlag, 2004.

[6] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas. *Multicast Security: A Taxonomy and Some Efficient Constructions.* In IEEE INFOCOMM'99, 1999.

[7] R. Canetti, T. Malkin, K. Nissim. *Efficient Communication-Storage Tradeoffs for Multicast Encryption.* Advances of Cryptology - Eurocrypt'99, LNCS 1592, pp. 459-474, Springer-Verlag, 1999.

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

[9] R. Dutta, S. Mukhopadhyay. *Improved Self-Healing Key Distribution with Revocation in Wireless Sensor Network.* In the proceeding of the IEEE Wireless Communications and Networking Conference (WCNC 2007), will be held in Hong Kong, China, 2007 (to appear).

[10] A. Fiat, M. Naor. *Broadcast Encryption.* In Crypto'93, LNCS 773, pp. 480-491, Springer-Verlag, 1994.

[11] L. Gong. *New Protocols for Third-Party-Based Authentication and Secure Broadcast.* In Proceedings of ACM CCS, 1994.

[12] D. Hong, J. Kang. *An Efficient Key Distribution Scheme with Self-healing Property.* IEEE Communication Letters'05, Vol. 9, pp. 759-761, 2005.

[13] M. Just, E. Kranakis, D. Krizanc, P. van Oorschot. *On Key Distribution via True Broadcasting.* In Proceedings of ACM CCS, 1994.

[14] D. Liu, P. Ning, K. Sun. *Efficient Self-healing Key Distribution with Revocation Capability.* Proceedings of the 10th ACM CCS'03, pp. 27-31, 2003.

[15] D. McGrew, A. Sherman. *Key Establishment in large dynamic groups using one-way function trees.* TIS Report No. 0755, 1998.

[16] S. More, M. Malkin, J. Staddon. *Sliding-window Self-healing Key Distribution with Revocation.* ACM Workshop on Survivable and Self-regenerative Systems'03, pp. 82-90, 2003.

[17] D. Naor, M. Naor, J. Lotspiech. *Revocation and Tracing Schemes for Stateless Users.* Advances of Cryptology - Crypto'01, LNCS 2139, pp. 41-62, Springer-Verlag, 2001.

[18] A. Perrig, D. Song, J. D. Tygar. *ELK, a New Protocol for Efficient Large-Group Key Distribution.* Proceedings of IEEE Symposium on Security and Privacy'01, pp. 247-262, 2001.

[19] G. Saez. *On Threshold Self-healing Key Distribution Schemes.* Cryptography and Coding'04, LNCS 3796, pp. 340-354, Springer-Verlag, 2004.

[20] R. Safavi-Naini, H. Wang. *New Constructions of Secure Multicast Re-Keying Schemes using Perfect Hash Families.* In Proceedings of ACM CCS'00, pp. 228-234, 2000.

[21] S. Setia, S. Koussih, S. Jajodia. *Korons: A Scalable Group Re-Keying Approach for Secure Multicast.* In Proceedings of IEEE Symp. on Security and Privacy.

[22] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, D. Dean. *Self-healing key distribution with Revocation.* Proceedings of IEEE Symposium on Security and Privacy'02, pp. 224-240, 2002.

[23] D. R. Stinson. *On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption.* Designs, Codes and Cryptology, vol. 12, pp.215-243, 1997.

[24] D. R. Stinson, T. van Trung. *Some New Results on Key Distribution Patterns and Broadcast Encryption.* Designs, Codes and Cryptography, vol. 14, pp. 261-279, 1998.

[25] C. Wong, M. Gouda, S. Lam. *Secure Group Communications using Key Graphs.* In Proceedings of ACM SIGCOMM'98, pp. 68-79, 1998.

[26] S. Xu. *On the security of group communication schemes.* In the Journal of Computer Security, Volume 15, Number 1, 2007, pp. 129 - 169, 2007.

[27] Y. R. Yang, X. S. Li, X. B. Zhang, S. S. Lam. *Reliable Group Re-Keying: A Performance Analysis.* In ACM SIGCOMM'01, pp. 27-38, 2001.