Generalized Self-healing Key Distribution Using Vector Space Access Structure

Ratna Dutta[1], Sourav Mukhopadhyay[2], Amitabha Das[2], and Sabu Emmanuel[2]

[1] Cryptography & Security Department
Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore - 119613
ratna.dutta@gmail.com
[2] School of Computer Engineering
Nanyang Technological University
N4-B2c-06, Nanyang Avenue, Singapore - 639798
{msourav,ASADAS,ASEmmanuel}@ntu.edu.sg

Abstract. We propose and analyze a generalized self-healing key distribution using vector space access structure in order to reach more flexible performance of the scheme. Our self-healing technique enables better performance gain over previous approaches in terms of storage, communication and computation complexity. We provide rigorous treatment of security of our scheme in an appropriate security framework and show it is computationally secure and achieves forward and backward secrecy.

Keywords: key distribution, self-healing, wireless network, access structure, computational security, forward and backward secrecy.

1 Introduction

Self-healing key distribution 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.

A. Das et al. (Eds.): NETWORKING 2008, LNCS 4982, pp. 612–623, 2008.

Our Contribution. We design a computationally secure and efficient generalized self-healing key distribution scheme for large and dynamic groups over insecure wireless networks. We consider general monotone decreasing access structure for the family of subsets of users that can be revoked instead of threshold one. More precisely, we use vector space access structure, which allows us to obtain a family of more flexible self-healing key distribution schemes with more flexible performances. Our self-healing mechanism uses one-way key chain that is more efficient compared to the self-healing techniques used in the previous schemes [1, 3, 4, 5, 7, 6]. Our construction is general in the sense that it depens on a particular public mapping ϕ and for different choices of ϕ we obtain different self-healing key diatribution schemes. The broadcast message length also depends on this particular function ϕ. Additionally, our construction does not require to send the history of revoked subsets of users in order to perform self-healing, yielding significant reduction in the communication cost. The main attraction of this paper is that our general construction has significant performance gain in terms of storage, communication and computation overhead. A special case of our family of self-healing key distribution is one that considers Shamir's (t, n)-threshold secret sharing. We emphasize that each user in this special self-healing key distribution scheme 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 cost $2(t^2+t)$. Here m is the maximum number of sessions, j is the current session number and q is a prime large enough to accommodate a cryptographic key. 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.

2 Preliminaries

2.1 Secret Sharing Schemes

In this section we define secret sharing schemes which play an important role in distributed cryptography.

Definition 2.1 (Access Structure). *Let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be a set of participants. A collection $\Gamma \subseteq 2^{\mathcal{U}}$ is monotone if $B \in \Gamma$ and $B \subseteq C \subseteq \mathcal{U}$ imply $C \in \Gamma$. An access structure is a monotone collection Γ of non-empty subsets of $\mathcal{U}$.* i.e., *$\Gamma \subseteq 2^{\mathcal{U}} \backslash \{\emptyset\}$. The sets in Γ are called the authorized sets. A set B is called minimal set of Γ if $B \in \Gamma$, and for every $C \subset B$, $C \neq B$, it holds that $C \notin \Gamma$. The set of minimal authorized subsets of Γ is denoted by Γ_0 and is called the basis of Γ. Since Γ consists of all subsets of $\mathcal{U}$ that are supersets of a subset in the basis Γ_0, Γ is determined uniquely as a function of Γ_0. More formally, we have $\Gamma = \{C \subseteq \mathcal{U} : B \subseteq C, B \in \Gamma_0\}$. We say that Γ is the closure of Γ_0 and write $\Gamma = cl(\Gamma_0)$. The family of non-authorized subsets $\overline{\Gamma} = 2^{\mathcal{U}} \backslash \Gamma$*

is monotone decreasing, that is, if $C \in \overline{\Gamma}$ *and* $B \subseteq C \subseteq \mathcal{U}$, *then* $B \in \overline{\Gamma}$. *The family of non-authorized subsets* $\overline{\Gamma}$ *is determined by the collection of maximal non-authorized subsets* $\overline{\Gamma}_0$.

In case of a (t, n)-threshold access structure, the basis consists of all subsets of (exactly) t participants. *i.e.* $\Gamma = \{B \subseteq \mathcal{U} : |B| \geq t\}$ and $\Gamma_0 = \{B \subseteq \mathcal{U} : |B| = t\}$.

Definition 2.2 (Secret Sharing). *Let* $\mathcal{K}$ *be a finite set of secrets, where* $\mathcal{K} \geq 2$. *An* n*-party secret sharing scheme* Π *with secret domain* $\mathcal{K}$ *is a randomized mapping from* $\mathcal{K}$ *to a set of* n*-tuples* $\mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_n$, *where* $\mathcal{S}_i$ *is called the share domain of* $U_i \in \mathcal{U}$. *A dealer* $D \notin \mathcal{U}$ *distributes a secret* $K \in \mathcal{K}$ *according to* Π *by first sampling a vector of shares* $(s_1, \ldots, s_n)$ *from* $\Pi(K)$, *and then privately communicating each share* s_i *to the party* U_i. *We say that* Π *realizes an access structure* $\Gamma \subseteq 2^{\mathcal{U}}$ *if the following two requirements hold:*

Correctness: *The secret* K *can be reconstructed by any authorized subset of parties. That is, for any subset* $B \in \Gamma$ *(where* $B = \{U_{i_1}, \ldots, U_{i_{|B|}}\}$*), there exists a reconstruction function* $\mathsf{Rec}_B : \mathcal{S}_{i_1} \times \ldots \times S_{i_{|B|}} \rightarrow \mathcal{K}$ *such that for every* $K \in \mathcal{K}$, $\mathsf{Prob}[\mathsf{Rec}_B(\Pi(K)_B) = K] = 1$, *where* $\Pi(K)_B$ *denotes the restriction of* $\Pi(K)$ *to its* B*-entries.*

Privacy: *Every unauthorized subset cannot learn anything about the secret (in the information theoretic sense) from their shares. Formally, for any subset* $C \notin \Gamma$, *for every two secrets* $K_1, K_2 \in \mathcal{K}$, *and for every possible shares* $\langle s_i \rangle_{U_i \in C}$, $\mathsf{Prob}[\Pi(K_1)_C = \langle s_i \rangle_{U_i \in C}] = \mathsf{Prob}[\Pi(K_2)_C = \langle s_i \rangle_{U_i \in C}]$.

The above correctness and privacy requirements capture the strict notion of perfect secret sharing, which is the one most commonly referred in the secret sharing literature.

Definition 2.3 (Vector Space Access Structure). *Suppose* Γ *is an access structure, and let* $(Z_q)^l$ *denote the vector space of all* l*-tuples over* Z_q, *where* q *is prime and* $l \geq 2$. *Suppose there exists a function* $\Phi : \mathcal{U} \cup \{D\} \rightarrow (Z_q)^l$ *which satisfies the property:* $B \in \Gamma$ *if and only if the vector* $\Phi(D)$ *can be expressed as a linear combination of the vectors in the set* $\{\Phi(U_i) : U_i \in B\}$. *An access structure* Γ *is said to be a vector space access structure if it can be defined in the above way.*

We now present vector space secret sharing scheme that was introduced by Brickell [2].

- *Initialization*: For $1 \leq i \leq n$, D gives the vector $\Phi(U_i) \in (Z_q)^l$ to U_i. These vectors are public.
- *Share Distribution*:
  1. Suppose D wants to share a key $K \in Z_q$. D secretly chooses (independently at random) $l - 1$ elements $a_2, \ldots, a_l$ from Z_q.
  2. For $1 \leq i \leq n$, D computes $s_i = v.\Phi(U_i)$, where $v = (K, a_2, \ldots, a_l) \in (Z_q)^l$.
  3. For $1 \leq i \leq n$, D gives the share s_i to U_i.

– *Key Recovery*: Let B be an authorized subset, $B \in \Gamma$. Then

$$\Phi(D) = \sum_{\{i:U_i \in B\}} \Lambda_i \Phi(U_i)$$

for some $\Lambda_i \in Z_q$. In order to recover the secret K, the participants of B pool their shares and computes

$$\sum_{\{i:U_i \in B\}} \Lambda_i s_i = v.\left(\sum_{\{i:U_i \in B\}} \Lambda_i \Phi(U_i)\right) = v.\Phi(D) = K \bmod q.$$

Thus when an authorized subset of participants $B \in \Gamma$ pool their shares, they can determine the value K. On the other hand, one can show that if an unauthorized subset $B \notin \Gamma$ pool their shares, they can determine nothing about the value of K (see [2] for proof).

2.2 Our Security Model

We now state the following definitions that are aimed to computational security for session key distribution adopting the security model of [5, 7].

Let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be the universe of the network. We assume the availability of a broadcast unreliable channel and there is a group manager GM who sets up and performs join and revoke operations to maintain a communication group, which is a dynamic subset of users of $\mathcal{U}$. Let m be the maximum number of sessions, and $\mathcal{R} \subset 2^{\mathcal{U}}$ be a monotone decreasing access structure of subsets of users that can be revoked by the group manager GM. Let $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m\}$ and $G_j \in \mathcal{U}$ be the group established by the group manager GM in session j.

Definition 2.4. *(Session Key Distribution with privacy [7])*

1. *$\mathcal{D}$ is a session key distribution with privacy if*
   (a) *for any user $U_i \in G_j$, the session key SK_j is efficiently determined from $\mathcal{B}_j$ and S_i.*
   (b) *for any set $R_j \subseteq \mathcal{U}$, where $R_j \in \mathcal{R}$ and $U_i \notin R_j$, it is computationally infeasible for users in R_j 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 $\mathcal{R}$-revocation capability if given any $R_j \subseteq \mathcal{U}$, where $R_j \in \mathcal{R}$, the group manager GM can generate a broadcast $\mathcal{B}_j$, such that for all $U_i \notin R_j$, 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 $\{S_l\}_{U_l \in R_j}$.*

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 $\{Z_{i,j_1}, Z_{i,j_2}\}$. In other words, every user $U_i \in G_{j_1}$, who has not been revoked after session j_1 and before session j_2, can recover all session keys SK_j for $j = j_1, \ldots, j_2$, from the broadcasts $\mathcal{B}_{j_1}$ and $\mathcal{B}_{j_2}$ where $1 \leq j_1 < j_2 \leq m$.*

Definition 2.5. *($\mathcal{R}$-wise forward and backward secrecy [5])*

1. *A key distribution scheme $\mathcal{D}$ guarantees $\mathcal{R}$-wise forward secrecy if for any set $R_j \subseteq \mathcal{U}$, where $R_j \in \mathcal{R}$, and all $U_s \in R_j$ are revoked before session j, it is computationally infeasible for the members in R_j 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 $\mathcal{R}$-wise backward secrecy if for any set $J_j \subseteq \mathcal{R}$, where $J_j \in \mathcal{U}$, and all $U_s \in J_j$ join after session j, it is computationally infeasible for the members in J_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 General Construction

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, $\mathsf{GF}(q)$, 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 $\mathcal{H} : \mathsf{GF}(q) \rightarrow \mathsf{GF}(q)$ be a cryptographically secure one-way function. The life of the system is divided in sessions $j = 1, 2, \ldots, m$. The communication group in session j is denoted by $G_j \subset \mathcal{U}$. We consider a linear secret sharing scheme realizing some access structure Γ over the set $\mathcal{U}$. For simplicity, suppose there exists a public function $\Phi : \mathcal{U} \cup \{\mathsf{GM}\} \rightarrow \mathsf{GF}(q)^l$ satisfying the property $\Phi(\mathsf{GM}) \in \langle \Phi(U_i) : U_i \in B \rangle \Leftrightarrow B \in \Gamma$, where l is a positive integer. In other words, the vector $\Phi(\mathsf{GM})$ can be expressed as a linear combination of the vectors in the set $\{\Phi(U_i) : U_i \in B\}$ if and only if B is an authorized subset. Then Φ defines Γ as a *vector space access structure*.

– *Setup*: Let $G_1 \in \mathcal{U}$. The group manager GM chooses independently and uniformly at random m vectors $v_1, v_2, \ldots, v_m \in \mathsf{GF}(q)^l$. The group manager randomly picks two initial key seeds, the forward key seed $S^F \in \mathsf{GF}(q)$ and the backward key seed $S^B \in \mathsf{GF}(q)$. It repeatedly applies (in the preprocessing time) the one-way function $\mathcal{H}$ on S^B and computes the one-way backward key chain of length m: $K_i^B = \mathcal{H}(K_{i-1}^B) = \mathcal{H}^{i-1}(S^B)$ 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 $U_i \in G_1$ receives its personal secret keys corresponding to the m sessions $S_i = (v_1.\Phi(U_i), \ldots, v_m.\Phi(U_i)) \in \mathsf{GF}(q)^m$ and the forward key seed S^F from the group manager via the secure communication channel between them. Here the operation "." is the inner product modulo q.

- *Broadcast*: Let R_j be the set of all revoked users for sessions in and before j such that $R_j \notin \Gamma$ and G_j be the set of all non-revoked users in session j. In the j-th session the GM first chooses a subset of users $W_j \subset \mathcal{U} \backslash G_j$ with minimal cardinality such that $W_j \cup R_j \in \overline{\Gamma}_0$. The GM then computes $Z_j = K^B_{m-j+1} + v_j.\Phi(\mathsf{GM})$ and broadcasts the message $\mathcal{B}_j = \{(U_k, v_j.\Phi(U_k)) : U_k \in W_j \cup R_j\} \cup \{Z_j\}$.
- *Session Key Recovery*: When a non-revoked user U_i receives the j-th session key distribution message $\mathcal{B}_j$, it recovers $v_j.\Phi(\mathsf{GM})$ as follows: Since $W_j \cup R_j \in \overline{\Gamma}_0$ is the maximal non-authorized subset with minimum cardinality having the property $W_j \in \mathcal{U} \backslash G_j$, the set $B = W_j \cup R_j \cup \{U_i\} \in \Gamma$. Thus B is an authorized subset, and one can write $\Phi(\mathsf{GM}) = \sum_{\{k:U_k \in B\}} \Lambda_k \Phi(U_k)$ for some $\Lambda_k \in \mathsf{GF}(\mathsf{q})$. Hence U_i knows Λ_k and $v_j.\Phi(U_k)$ for all $k \in B$ and can compute $\sum_{\{k:U_k \in B\}} \Lambda_k (v_j.\Phi(U_k)) = v_j.\left(\sum_{\{k:U_k \in B\}} \Lambda_k \Phi(U_k)\right) = v_j.\Phi(\mathsf{GM})$
  Then U_i recovers the key K^B_{m-j+1} as $K^B_{m-j+1} = Z_j - v_j.\Phi(\mathsf{GM})$. Finally, U_i computes the j-th forward key $K^F_j = \mathcal{H}^{j-1}(S^F)$ and evaluates the current session key $\mathsf{SK}_j = K^F_j + K^B_{m-j+1}$. A user U_k who either does not know its private information $v_j.\Phi(U_k)$ or who is a revoked user in R_j, *i.e.* $U_k \in W_j \cup R_j$, cannot compute $v_j.\Phi(\mathsf{GM})$ because U_k only knows values broadcast in the message $\mathcal{B}_j$ corresponding to an unauthorized subset of the secret sharing scheme. Consequently, U_k cannot recover the backward key K^B_{m-j+1} and hence the j-th session key SK_j.
- *Add Group Members*: When the group manager adds a new group member starting from session j, it picks an unused identity $v \in \mathsf{GF}(q)$, computes the personal secret keys corresponding to the current and future sessions $S_v = (v_j.\Phi(U_v), \ldots, v_m.\Phi(U_v)) \in \mathsf{GF}(q)^{m-j+1}$ and gives $\{v, S_v, K^F_j\}$ to this new group member via the secure communication channel between them.
- *Re-initialization:* The system fails when all m sessions are exhausted, or the set of revoked users for sessions in and before the current session becomes an element of Γ. At this phase, re-initialization is required and a new setup is executed.

3.1 Complexity

Storage overhead: For simplicity, we use vector space secret sharing. Then storage complexity of personal key for each user is $m \log q$ bits. The group members that join later need to store less data. For example, the personal key for a user joining at the j-th session occupies $(m-j+1) \log q$ bits memory space. If we use a more general linear secret sharing scheme in which a participant U_i is associated with $m_i \geq 1$ vectors, then its personal secret key consists of m_i vectors and hence is of size $mm_i \log q$ bits.

Communication overhead: The communication bandwidth for key management at the j-th session is $(t_j + 1) \log q$ bits, where $t_j = |W_j \cup R_j|$, $R_j \notin \Gamma$ is the set of all revoked users for sessions in and before j and $W_j \subset \mathcal{U} \backslash G_j$ with minimum cardinality such that $W_j \cup R_j \in \overline{\Gamma}_0$. Here we ignore the communication overhead for the broadcast of user identities U_i for $U_i \in W_j \cup R_j$, as

these identities can be picked from a small finite field. In particular, if our scheme is obtained from Shamir's (t, n)-threshold secret sharing scheme that realizes access structure defined by $\Gamma = \{A \subseteq \mathcal{U} : |A| \geq t\}$ by means of polynomial interpolation, then communication bandwidth for key management is $(t+1)\log q$ bits.

Computation overhead: The computation complexity is $2(t_j^2 + t_j)$, where $t_j = |W_j \cup R_j|$, $R_j \notin \Gamma$ is the set of all revoked users for sessions in and before j and $W_j \subset \mathcal{U} \backslash G_j$ with minimum cardinality such that $W_j \cup R_j \in \overline{\Gamma}_0$. This is the number of multiplication operations needed to recover $\Phi(\mathsf{GM})$ by using equation (1). Considering Shamir's (t, n)-threshold secret sharing scheme, the computation cost for key management is $2(t^2+t)$, which is essentially the number of multiplication operations needed to recover a t-degree polynomial by using Lagrange's interpolation formula.

3.2 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 $1 \leq j_1 \leq j_2$, 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 $K^B_{m-j_2+1}$ and repeatedly apply the one-way function $\mathcal{H}$ on this and computes the backward keys K^B_{m-j+1} for all j, $j_1 \leq j < j_2$.
(b) U_i computes the forward keys K^F_j for all j, $j_1 \leq j \leq j_2$ by repeatedly applying $\mathcal{H}$ on the forward seed S^F or on the forward key $K^F_{j_1}$ of the j_1-th session.
(c) U_i then recovers all the session keys $\mathsf{SK}_j = K^F_j + K^B_{m-j+1}$, for $j_1 \leq j \leq j_2$.

Note that a user revoked in session j cannot compute the backward keys $K^B_{m-j_1+1}$ for $j_1 > j$, although it can compute the forward keys $K^F_{j_1}$. 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 $K^F_{j_2}$ for $j_2 < j$ as U_i knows only the j-th forward key K^F_j, 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

Theorem 4.1. *Our construction is secure, self-healing session key distribution scheme with privacy, $\mathcal{R}$-revocation capability with respect to Definition 2.4 and achieve $\mathcal{R}$-wise forward and backward secrecy with respect to Definition 2.5.*

Proof: Our goal is security against coalition of users from $\mathcal{R}$. We will show that our construction is computationally secure with respect to revoked users under the difficulty of inverting one-way function, *i.e.* for any session j it is computationally infeasible for any set of revoked users from $\mathcal{R}$ 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 revoked users from $\mathcal{R}$, say $R_j \in \mathcal{R}$, 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 users from $\mathcal{R}$ 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 $S^F \in \mathsf{GF}(q)$ and constructs the following backward key chain by repeatedly applying $\mathcal{H}$ on y:

$$K_1^B = y, K_2^B = \mathcal{H}(y), \ldots, K_j^B = \mathcal{H}^{j-1}(y), \ldots, K_m^B = \mathcal{H}^{m-1}(y).$$

$\mathcal{A}$ computes the j-th forward key $K_j^F = \mathcal{H}^{j-1}(S^F)$ and sets the j-th session key $\mathsf{SK}_j = K_j^F + K_{m-j+1}^B$. $\mathcal{A}$ chooses at random m vectors $v_1, \ldots, v_m \in \mathsf{GF}(q)^l$. Each user $U_i \in \mathcal{U}$ receives its personal secret keys corresponding to the m sessions $S_i = (v_1.\Phi(U_i), \ldots, v_m.\Phi(U_i)) \in \mathsf{GF}(q)^m$ and the forward key seed S^F from $\mathcal{A}$ via the secure communication channel between them. In this setting, $\Gamma = 2^{\mathcal{U}} \backslash \mathcal{R}$ is a monotone increasing access structure of authorized users over $\mathcal{U}$. Γ is determined by the family of *minimal qualified subsets*, Γ_0, which is called the basis of Γ. Now $R_j \in \mathcal{R}$ implies $R_j \notin \Gamma$.

Let G_j be the set of all non-revoked users in session j. At the j-th session, $\mathcal{A}$ chooses a subset of users $W_j \subset \mathcal{U} \backslash G_j$ with minimal cardinality such that $W_j \cup R_j \in \overline{\Gamma}_0$. $\mathcal{A}$ then computes broadcast message $\mathcal{B}_j$ for $j = 1, \ldots, m$ as:

$$\mathcal{B}_j = \{(U_k, v_j.\Phi(U_k)) : U_k \in W_j \cup R_j\} \cup \{Z_j\},$$

where $Z_j = K_{m-j+1}^B + v_j.\Phi(\mathsf{GM})$. Then $\mathcal{A}$ sets View as

$$\mathsf{View} = \left\{ \begin{array}{l} v_s.\Phi(U_k) \text{ for all } U_k \in R_j \text{ and } s = 1, \ldots, m; \\ \mathcal{B}_j \text{ for } j = 1, \ldots, m; \\ S^F; \\ \mathsf{SK}_1, \ldots, \mathsf{SK}_{j-1} \end{array} \right\}$$

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

$\mathcal{A}'$ can compute the j-th forward key $K_j^F = \mathcal{H}(S^F)$ as it knows S^F from View for $j = 1, \ldots, m$. Note that from View, $\mathcal{A}'$ knows $\{v_j.\Phi(U_k) : U_k \in W_j \cup R_j, 1 \leq j \leq m\} \cup \{v_s.\Phi(U_k) : U_k \in R_j, 1 \leq s \leq m\}$ 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 $K_m^B, \ldots, K_{m-j+2}^B$. Observe that $\mathsf{SK}'_j = \mathsf{SK}_j$ provided $\mathcal{A}'$ knows the backward key K_{m-j+1}^B. This occurs if either of the following two holds:

(a) $\mathcal{A}'$ is able to compute the $v_j.\Phi(\mathsf{GM})$ from View and consequently can recover the backward key K_{m-j+1}^B as follows:$K_{m-j+1}^B = Z_j - v_j.\Phi(\mathsf{GM})$.
From View, $\mathcal{A}'$ knows $\{v_j.\Phi(U_k) : U_k \in W_j \cup R_j, 1 \leq j \leq m\} \cup \{v_s.\Phi(U_k) : U_k \in R_j, 1 \leq s \leq m\}$, where $W_j \subset \mathcal{U} \backslash G_j$ has minimal cardinality with $W_j \cup R_j \in \overline{\Gamma}_0$ and will not be able to compute $v_j.\Phi(\mathsf{GM})$ by the property of Φ. Observe that $v_j.\Phi(\mathsf{GM})$ is linear combination of $\{v_j.\Phi(U_k) : U_k \in B\}$ if and only if $B \in \Gamma$. Consequently, $\mathcal{A}'$ will not be able to recover K_{m-j+1}^B from $\mathcal{B}_j$ as described in (a) above.

(b) $\mathcal{A}'$ is able to choose $X \in \mathsf{GF}(q)$ so that the following relations hold: $K_m^B = \mathcal{H}^{j-1}(X), K_{m-1}^B = \mathcal{H}^{j-2}(X), \ldots, K_{m-j+2}^B = \mathcal{H}(X)$. 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 our construction, then $\mathcal{A}$ is able to invert the one-way function. □
(of claim)

Hence our construction is computationally secure under the hardness of inverting one-way function. We will now show that our construction satisfies all the conditions required by Definition 2.4.

1) (a) Session key efficiently recovered by a non-revoked user U_i is described in the third step of our construction.
(b) For any set $R_j \subseteq \mathcal{U}$, $R_j \in \mathcal{R}$, and any non-revoked user $U_i \notin R_j$, we show that the coalition R_j knows nothing about the personal secret $S_i = (v_1.\Phi(U_i), \ldots, v_j.\Phi(U_i), \ldots, v_m.\Phi(U_i))$ of U_i. For any session j, U_i uses $v_j.\Phi(U_i)$ as its personal secret. Since the coalition $R_j \notin \Gamma$, the values $\{v_s.\Phi(U_k) : U_k \in R_j, 1 \leq s \leq m\}$ is not enough to compute $v_j.\Phi(U_i)$ by the property of Φ. So it is computationally infeasible for coalition R_j to learn $v_j.\Phi(U_i)$ for $U_i \notin R_j$.
(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 secrets $S_1, \ldots, S_m$ where
$S_i = (v_1.\Phi(U_i), \ldots, v_j.\Phi(U_i), \ldots, v_m.\Phi(U_i))$ 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^B_{m-j+1} and consequently the session key SK_j is random as long as S^B, $K^B_1, K^B_2, \ldots, K^B_{m-j+2}$ 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) ($\mathcal{R}$-revocation property) Let $R_j \subseteq \mathcal{U}$, where $R_j \in \mathcal{R}$, collude in session j. It is impossible for coalition R_j to learn the j-th session key SK_j because the knowledge of SK_j implies the knowledge of either the backward key K^B_{m-j+1} or the knowledge of the personal secret $v_j.\Phi(U_i)$ of user $U_i \notin R_j$. The coalition R_j knows the set $\{v_s.\Phi(U_k) : U_k \in R_j, 1 \leq s \leq m\}$, which is not enough to compute $v_j.\Phi(U_i)$ by the property of Φ. Hence the coalition R_j cannot recover $v_j.\Phi(U_i)$, which in turn makes K^B_{m-j+1} appears random to all users in R_j. Therefore, SK_j is completely safe to R_j from computation point of view.
3) (Self-healing property) From the third step of our construction, 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 $K^B_{m-j_2+1}$ and hence can obtain the sequence of backward keys $K^B_{m-j_1}, \ldots, K^B_{m-j_2+2}$ by repeatedly applying $\mathcal{H}$ on $K^B_{m-j_2+1}$. User U_i also holds the forward key $K^F_{j_1} = \mathcal{H}^{j_1-1}(S^F)$ of the j_1-th session and hence can obtain the sequence of forward keys $K^F_{j_1+1}, \ldots, K^F_{j_2-1}$ by repeatedly applying $\mathcal{H}$ on $K^F_{j_1}$. Hence, as shown in Section 3.2, user U_i can efficiently recover all missed session keys.

We will show that our construction satisfies all the conditions required by Definition 2.5.

1) ($\mathcal{R}$-wise forward secrecy) Let $R_j \subseteq \mathcal{U}$, where $R \in \mathcal{R}$ and all user $U_s \in R_j$ are revoked before the current session j. The coalition R_j 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, any user $U_s \in R_j$ needs to know either $v_j.\Phi(\mathsf{GM})$ or K^B_{m-j+1}. Determining $v_j.\Phi(\mathsf{GM})$ requires knowledge of values $\{v_j.\Phi(U_k) : U_k \in B$ for some $B \in \Gamma\}$. But the coalition R_j knows only the values $\{v_j.\Phi(U_s) : U_s \in R_j\}$ which is insufficient as $R_j \notin \Gamma$. Hence R_j is unable to compute SK_j.

   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_j 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 our construction is $\mathcal{R}$-wise forward secure.
2) ($\mathcal{R}$-wise backward secrecy) Let $J_j \subseteq \mathcal{U}$, where $J_j \in \mathcal{R}$ and all user $U_s \in J_j$ join after the current session j. The coalition J_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}, any user $U_s \in J_j$ 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 $S_v = (v_j.\Phi(U_v), \ldots, v_m.\Phi(U_v)) \in \mathsf{GF}(q)^{m-j+1}$. 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 $\mathcal{R}$-wise backward secure. In fact, this backward secrecy is independent of $\mathcal{R}$. □

5 Performance Analysis

The existing work to deal with self-healing key distribution using monotone decreasing family of revoked subset of users instead of monotone decreasing threshold structure is by Saez [6]. In this section, we discuss the comparison of storage overhead, communication complexity and computation cost of each user (not the GM) in our construction with [6]. In contrast to the family of the self-healing key distribution schemes proposed in [6], our general construction uses a different self-healing approach which is more efficient in terms of computation and communication without any further trade-off in storage, yielding more flexible self-healing key distribution scheme that can provide better properties. Unlike [6], the length of the broadcast message in our scheme does not depend on the history of revoked subsets of users to perform self-healing. This feature provides significant reduction in the communication cost, which is one of the main improvement of our scheme over the previous works. For simplicity, we compare a special case of our construction with the other similar schemes considering Shamir's (t, n)-threshold secret sharing.

In one hand our construction reduces the communication complexity (bandwidth) to $O(t)$, whereas optimal communication complexity achieved by the previous schemes [1, 6, 7] 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 is $K^B_{m-j+1} = Z_j - v_j.\Phi(\mathsf{GM})$. Thus computation complexity for each user 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. 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 communication complexity and computation cost in this special construction do not increase as the number of session grows. These are the most prominent improvements of our scheme over the previous secret sharing based self-healing key distributions [1, 6, 7]. The storage requirement in our scheme comes from *Setup* phase and after receiving the session key distribution message. The storage overhead of each user for personal key is $O((m - j + 1) \log q)$, which is same as that of [1, 6, 7].

If we consider a secret sharing scheme realizing a specific bipartite access structure defined in the set of users, the previous self-healing mechanisms allow

to improve the efficiency of revocations of a small number of users, say less than j, for some positive integer $j \leq t-1$, t is the threshold on the number of revoked users. This is because of the fact that in all the previous self-healing key distribution schemes, a part of the broadcast message of every session contains a history of revoked subsets of users in order to perform self-healing. This part of broadcast message has a proportional amount of information to $t-1$ in all the previous self-healing key distribution schemes, despite only two or three users must be revoked. We overcome this overhead on broadcast message length in our general construction since our self-healing mechanism does not need to send any such history.

6 Conclusion

This paper presents an efficient computationally secure generalized self-healing key distribution scheme 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. Our proposed key distribution mechanism reduces communication and computation costs over the previous approaches, without any additional increase in the storage complexity compared to the previous works, and is scalable to very large groups in highly mobile, volatile, and hostile wireless network. Our scheme is properly analyzed in an appropriate security model to prove that it is computationally secure and achieves both forward secrecy and backward secrecy.

References

[1] Blundo, C., D'Arco, P., Santis, A., Listo, M.: Design of Self-healing Key Distribution Schemes. Design Codes and Cryptology 32, 15–44 (2004)
[2] Brickell, E.F.: Some Ideal Secret Sharing Schemes. Journal of Combinatorial Mathematics and Combinatorial Computing 9, 105–113 (1983)
[3] Dutta, R., Mukhopadhyay, S.: Improved Self-Healing Key Distribution with Revocation in Wireless Sensor Network. In: Proceedings of WCNC 2007 - Networking, IEEE Computer Society Press, Los Alamitos (2007)
[4] Hong, D., Kang, J.: An Efficient Key Distribution Scheme with Self-healing Property. IEEE Communication Letters 2005 9, 759–761 (2005)
[5] Liu, D., Ning, P., Sun, K.: Efficient Self-healing Key Distribution with Revocation Capability. In: Proceedings of the 10th ACM CCS 2003, pp. 27–31 (2003)
[6] Saez, G.: On Threshold Self-healing Key Distribution Schemes. In: Smart, N.P. (ed.) Cryptography and Coding 2005. LNCS, vol. 3796, pp. 340–354. Springer, Heidelberg (2005)
[7] Staddon, J., Miner, S., Franklin, M., Balfanz, D., Malkin, M., Dean, D.: Self-healing key distribution with Revocation. In: Proceedings of IEEE Symposium on Security and Privacy 2002, pp. 224–240 (2002)