Enhanced Access Polynomial Based Self-healing Key Distribution

Ratna Dutta[1], Sourav Mukhopadhyay[2], and Tom Dowling[1]

[1] Claude Shannon Institute, Computer Science Department, NUI Maynooth, Co. Kildare, Ireland
{rdutta,tdowling}@cs.nuim.ie

[2] School of Electronic Engineering, Dublin City University, Dublin 9, Ireland
msourav@eeng.dcu.ie

Abstract. A fundamental concern of any secure group communication system is that of key management. Wireless environments create new key management problems and requirements to solve these problems. One such core requirement in these emerging networks is that of self-healing. In systems where users can be offline and miss updates self healing allows a user to recover lost keys and get back into the secure communication without putting extra burden on the group manager. Clearly self healing must be only available to authorized users and this creates more challenges in that we must ensure unauthorized or revoked users cannot, themselves or by means of collusion, avail of self healing. To this end we enhance the one-way key chain based self-healing key distribution of Dutta *et al.* by introducing a collusion resistance property between the revoked users and the newly joined users. Our scheme is based on the concept of access polynomials. These can be loosely thought of as white lists of authorized users as opposed to the more widely used revocation polynomials or black lists of revoked users. We also allow each user a pre-arranged life cycle distributed by the group manager. Our scheme provides better efficiency in terms of storage, and the communication and computation costs do not increase as the number of sessions grows as compared to most current schemes. We analyze our scheme in an appropriate security model and prove that the proposed scheme is computationally secure and not only achieving forward and backward secrecy, but also resisting collusion between the new joined users and the revoked users. Unlike most existing schemes the new scheme allows temporary revocation. Also unlike existing schemes, our construction does not collapse if the number of revoked users crosses a threshold value. This feature increases resilience against revocation based denial of service (DOS) attacks and thus improves availability of communication channel.

Keywords: session key distribution, self-healing, computational security, forward and backward secrecy.

1 Introduction

In a large and dynamic group communication over an unreliable wireless network, self-healing means that authorized users can recover the missing session keys by

Q. Gu, W. Zang, and M. Yu (Eds.): SEWCN 2009, LNICST 42, pp. 13–24, 2010.

themselves, without requesting additional transmission from the group manager. This reduces network traffic, the risk of user exposure to traffic analysis, and the work load on the group manager.

Self-healing property is being widely used for various applications. For example, mission critical applications such as in military, content sensitive internet applications such as broadcast transmission, pay per-view TV, and information distribution services.

The idea of self-healing key distribution was proposed by Staddon *et al.* [9]. Following it, a number of self-healing techniques have been proposed. The hash chain based schemes [3,4] are computationally secure and are highly efficient compared to the existing unconditionally secure schemes [2,6,8,12]. However, these hash chain based constructions have the fatal defect of not being collusion resistant in the sense that the collusion between new joined users and the revoked users are able to recover all the session keys which they are not entitled to.

Our contribution: In this paper, we provide a solution to the problem of resisting the collusion attack in the one-way hash chain based self-healing key distributions introduced by Dutta *et al.* [3,4], coupling it with the pre-arranged life cycle based approach of Tian *et al.* [10] that uses the same self-healing mechanism introduced in Dutta *et al.* [3,4]. However, we use the concept of access polynomial instead of revocation polynomial in our construction. For scalability of business it is often necessary to design more innovative and flexible business strategies in certain business models that allow contractual subscription or rental, such as subscription of mobile connection or TV channel for a pre-defined period. The subscribers are not allowed to revoke before their contract periods (life cycles) are over. Our scheme fits into such business strategies. Our construction is flexible and robust in the sense that there is no restriction on the number of revoked users, any number of users can leave/join the group and a user can join/leave as many times as she wishes. Consequently, the availability of communication channel is increased and revocation based denial of service (DOS) attacks are reduced. As compared to most existing schemes, our scheme provides better efficiency in terms of storage, and the communication and computation costs do not increase as the number of session grows, rather they increase as the number of authorized users in a session grows. While most of the existing schemes collapse when the number of revoked users crosses a threshold value, say t, our scheme is unaffected by this limitation. Moreover, if the number of authorized users remains less than t, the communication and computation cost in our scheme are significantly less than that in the existing schemes together with less storage overhead. These are the most important features of our construction. The proposed scheme is proven to be computationally secure and achieve forward and backward secrecy together with resisting collusion between the newly joined users and the revoked users. The security analysis is in an appropriate security model.

2 Preliminaries

2.1 Notational Convention

The following notations are used throughout the paper.

$\mathcal{U}$	: 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
Auth_j	: the set of all authorized users in session j
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

2.2 Our Security Model

We state the following definitions that are aimed to computational security for session key distribution adopting the security model of [7,9].

Definition 1 *(Session Key Distribution with privacy [9]). Let $i \in \{1, \ldots, n\}$ 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 $R \subseteq \mathcal{U}$ of revoked users and $U_i \notin R$, it is computationally infeasible for users in R to determine the personal key S_i.

(c) 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 for users $U_1, \ldots, U_n$ to compute session key SK_j (or other useful information) from either set. Information from both the sets is required in order to compute SK_j or any useful information.

2) $\mathcal{D}$ has revocation capability if given any $R \subseteq \mathcal{U}$ of users revoked in and before session j, 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, $1 \le j_1 < j < j_2 \le m$:

(a) 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 $\{Z_{i,j_1}, Z_{i,j_2}\}$.

(b) Let $1 \le j_1 < j < j_2 \le m$. For any disjoint subsets $L_1, L_2 \subset \mathcal{U}$, where the set L_1 is a coalition of users removed before and in session j_1 and the set L_2 is a coalition of users joined since session j_2, the set $\{Z_{l,j}\}_{U_l \in L_1, 1 \le j \le j_1} \cup \{Z_{l,j}\}_{U_l \in L_2, j_2 \le j \le m}$ cannot determine the session key SK_j, $j_1 < j < j_2$. i.e. *SK_j can not be obtained by the coalition $L_1 \cup L_2$. This is collusion resistance property for self-healing.*

Definition 2 *(Forward and backward secrecy [7]). Let $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$.*

1) A key distribution scheme $\mathcal{D}$ guarantees forward secrecy if for any set $R \subseteq \mathcal{U}$ of users revoked in and 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 backward secrecy if for any set $J \subseteq \mathcal{U}$ of users joined after session j, it is computationally infeasible for the members in J together to get any information about SK_j, even with the knowledge of group keys $\mathsf{SK}_{j+1}, \ldots, \mathsf{SK}_m$ after session j.

3 Proposed Scheme

For our construction, we consider a setting in which there is a group manager (GM) and n users $\mathcal{U} = \{U_1, \ldots, U_n\}$. All operations take place in a finite field, F_q, where q is a large prime number $(q > n)$. In our setting, we 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. See [5] for a formal definition of one-way function. We use Cryptographically Secure Pseudo random bit Generators (CSPRBG) in our construction. An example of CSPRBGs include the RSA PBG [1].

3.1 Key Distribution

• *Setup*: The group manager randomly picks an initial backward key seed $S^B \in F_q$. It repeatedly applies the one-way function $\mathcal{H}$ to compute the one-way key chain of length m: $K_i^B = \mathcal{H}(K_{i-1}^B) = \mathcal{H}^{i-1}(S^B)$ for $1 \le i \le m$. The GM also selects at random n numbers $\alpha_1, \ldots, \alpha_n \in F_q$ and m numbers $\beta_1, \ldots, \beta_m \in F_q$ by running a CSPRBG which is cryptographically secure. The j-th session key is computed as $\mathsf{SK}_j = \beta_j + K_{m-j+1}^B$. Unlike the existing self-healing key distribution schemes, our setting allows a revoked user to rejoin the group in a later session with a new identity. However, we make the following restriction on the life cycle of each user as determined by the GM. Each user U_i is first assigned a pre-arranged life cycle (s_i, t_i), where $1 \le s_i < t_i \le m$, by the GM. *i.e.* U_i is involved in $k_i = t_i - s_i + 1$ many sessions and is not allowed to revoke before session t_i. However U_i may go off-line during its life cycle due to power failure. Self-healing is needed at this point. Each user U_i, for $1 \le i \le n$, receives its personal secret keys corresponding to the $k_i = t_i - s_i + 1$ sessions $S_i = \{\alpha_i; \beta_{s_i}, \ldots, \beta_{t_i}\}$ from the group manager via the secure communication channel between them.

• *Broadcast*: Let Auth_j be the set of all authorized (active) users in session j. In the j-th session, the group manager randomly chooses a blind value $\theta_j \in F_q$, $\theta_j \notin \{\alpha_1, \ldots, \alpha_n\}$, locates the backward key K_{m-j+1}^B in the backward key chain and computes the polynomials: $A_j(x) = 1 + (x - \theta_j) \prod_{\{l : U_l \in \mathsf{Auth}_j\}} (x - \alpha_l)$, $h_j(x) = K_{m-j+1}^B A_j(x)$. The polynomial $A_j(x)$ is called the *access polynomial* in session j. The factor $(x - \theta_j)$ is a blinding term and $\theta_j \in F_q$ is randomly selected for each session j and is different from $\alpha_1, \ldots, \alpha_n \in F_q$. The purpose of $(x - \theta_j)$ is to make $A_j(x)$ different for different session j even they contain the same α's of authorized users. Note that $A_j(\alpha_i) = 1$ for $U_i \in \mathsf{Auth}_j$. However, it is random for an unauthorized user. The group manager broadcasts the following message $\mathcal{B}_j = \{h_j(x)\}$.

• *Session Key Recovery*: When an authorized (non-revoked) user $U_i \in \mathsf{Auth}_j$ receives the j-th session key distribution message $\mathcal{B}_j$, it recovers $K^B_{m-j+1} = h_j(\alpha_i)$ as $A_j(\alpha_i) = 1$. Finally, $U_i \in \mathsf{Auth}_j$ evaluates the current session key $\mathsf{SK}_j = \beta_j + K^B_{m-j+1}$. An unauthorized user cannot construct the polynomial $A_j(x)$ as it does not know the α-values of the set of authorized users Auth_j in session j and the blind value θ_j used in session j.

• *Add Group Members*: When a new user wants to join the communication group starting from session j, the user gets in touch with the GM. The GM in turn picks an unused identity $v \in F_q$, selects a new $\alpha_v \in F_q$, assigns a life cycle (s_v, t_v) to the new user with $s_v = j$, computes the personal secret keys corresponding to $k_v = t_v - s_v + 1$ sessions $S_v = \{\alpha_v; \beta_{s_v}, \ldots, \beta_{t_v}\}$ and gives S_v to this new group member via the secure communication channel between them.

Complexity

- *Storage overhead:* Storage complexity of personal key for user U_i with life cycle (s_i, t_i) is $(t_i - s_i + 2)\log q$ bits.
- *Communication overhead:* Communication bandwidth for key management at the j-th session is $(|\mathsf{Auth}_j| + 2)\log q$ bits, where Auth_j is the set of authorized users in session j.
- *Computation overhead:* The computation cost for key management at the j-th session is $(|\mathsf{Auth}_j| + 1)$, which is the number of multiplication operations needed to find a point on a $|\mathsf{Auth}_j| + 1$-degree polynomial.

3.2 Self-healing

We now explain our self-healing mechanism for the construction. 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 desired by Definition 1 $3(a)$ using the following steps.

- 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, $j_1 \le j < j_2$.
- U_i then recovers all the session keys $\mathsf{SK}_j = \beta_j + K^B_{m-j+1}$, for $j_1 \le j \le j_2$.

Note that a user U_i revoked in session j cannot compute the backward keys $K^B_{m-j_1+1}$ for $j_1 > j$. Moreover, since a user is not allowed to revoke before the end of its life cycle, U_i revoked in j-th session means its life cycle completes at the j-th session. Consequently, U_i does not have β_{j_1} for $j_1 > j$. As a result, revoked users cannot compute the subsequent session keys SK_{j_1} for $j_1 > j$, as desired. This is forward secrecy.

Similarly, a user U_i joined in session j does not have β_{j_2} for $j_2 < j$, 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. This is backward secrecy.

Now we will show that our construction can resist collusion required by Definition 1 3(b). Let $1 \leq j_1 < j < j_2 \leq m$. For any disjoint subsets $L_1, L_2 \subset \mathcal{U}$, let the set L_1 is a coalition of users removed before and in session j_1 and the set L_2 is a coalition of users joined from session j_2. Then no information about the session key SK_j, $j_1 < j < j_2$ can be obtained by the coalition $L_1 \cup L_2$. Our construction satisfies this property as illustrated below: Secret information held by users in $L_1 \cup L_2$ and broadcasts in all the sessions do not get any information about SK_j for $j_1 < j < j_2$. This is true because in the worst case, the coalition knows $S_i = \{\alpha_i; \beta_1, \ldots, \beta_{j_1-1}\}$ for $U_i \in L_1$, $S_i = \{\alpha_i; \beta_{j_2}, \ldots, \beta_m\}$ for $U_i \in L_2$, and $\mathcal{B}_1, \ldots, \mathcal{B}_m$. For each session j, $j_1 < j \leq j_2 - 1$, the coalition can get backward key K^B_{m-j+1} from L_2. However the session key SK_j is computed from the backward key K^B_{m-j+1} and a random number β_j. The coalition $L_1 \cup L_2$ cannot obtain the random numbers β_j for $j_1 < j < j_2$. Consequently, all the guess for SK_j with $j_1 < j < j_2$ are equi-probable.

4 Security Analysis

Theorem 3. *Our construction is secure, self-healing session key distribution scheme with privacy, revocation capability with respect to Definition 1 in our security model as described in Section 2.2 and achieves forward and backward secrecy with respect to Definition 2 in the model.*

Proof: Our goal is security against coalition of any size. We will show that our construction 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 before and in 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, in and after session j and session keys of revoked users before session j.

Consider a coalition R_j of users revoked in 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 R_j 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: Assuming a cryptographically secure CSPRBG, $\mathcal{A}$ 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 generates n distinct numbers $\alpha_1, \ldots, \alpha_n \in F_q$ and m distinct numbers $\beta_1, \ldots, \beta_m \in F_q$ by running a cryptographically secure CSPRBG and constructs the following backward key chain by repeatedly applying $\mathcal{H}$ on y: $K_1^B = y, K_2^B = \mathcal{H}(y), K_3^B = \mathcal{H}^2(y), \ldots, K_j^B = \mathcal{H}^{j-1}(y), \ldots, K_m^B = \mathcal{H}^{m-1}(y)$. $\mathcal{A}$ computes the j-th session key $\mathsf{SK}_j = \beta_j + K_{m-j+1}^B$. For $1 \leq i \leq n$, each user $U_i \in \mathcal{U}$ with life cycle, say (s_i, t_i), $1 \leq s_i < t_i \leq m$ (which is assigned to U_i by $\mathcal{A}$), receives its personal secret keys corresponding to the k_i sessions $S_i = \{\alpha_i; \beta_{s_i}, \ldots, \beta_{t_i}\} \in F_q^{k_i+1}$ from $\mathcal{A}$ via the secure communication channel between them.

Let Auth_j be the set of all authorized (active) users in session j. In the j-th session, $\mathcal{A}$ randomly chooses a blind value $\theta_j \in F_q$, $\theta_j \notin \{\alpha_1, \ldots, \alpha_n\}$ and computes the access polynomial

$$A_j(x) = 1 + (x - \theta_j) \prod_{\{l : U_l \in \mathsf{Auth}_j\}} (x - \alpha_l)$$

and the polynomial $h_j(x) = K_{m-j+1}^B A_j(x)$. For $1 \leq j \leq m$, $\mathcal{A}$ computes broadcast message $\mathcal{B}_j$ as: $\mathcal{B}_j = \{h_j(x)\}$. Then $\mathcal{A}$ sets View as

$$\mathsf{View} = \left\{ \begin{array}{l} \alpha_k \text{ for all } U_k \in R_j; \\ \mathcal{B}_j \text{ for } j = 1, \ldots, m; \\ \beta_1, \ldots, \beta_{j-1}; \\ \mathsf{SK}_1, \ldots, \mathsf{SK}_{j-1} \end{array} \right\}$$

$\mathcal{A}$ gives View to $\mathcal{A}'$, which in turn selects $X, \beta_j' \in F_q$ randomly, sets the j-th session key to be $\mathsf{SK}_j' = \beta_j' + X$ 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'.

Note that the access polynomial $A_j(x)$ at the j-th session is not publicly computable from the broadcast message $\mathcal{B}_j = \{h_j(x)\}$ as:

- The set of authorized users is not transmitted publicly during broadcast.
- α-values of authorized users are used in $A_j(x)$ which are parts of secret of authorized users.
- A blinding factor $(x - \theta_j)$ is used in $A_j(x)$ where $\theta_j \in F_q$ is randomly chosen for each session j and is different from α-values of users. Thus $A_j(x)$ is different for different sessions j even if the same α-values of authorized users are used.
- $\mathcal{A}_j(\alpha_i) = 1$ for $U_i \in \mathsf{Auth}_j$ and $\mathcal{A}_j(\alpha_i)$ is random for $U_i \notin \mathsf{Auth}_j$.
- Computing α_i for $U_i \in \mathsf{Auth}_j$ is infeasible from the set $\{\alpha_k : U_k \notin \mathsf{Auth}_j\}$ as we assume that the CSPRBG used to generate these α-values is cryptographically secure.
- an adversary or a coalition R_j of users revoked in and before session j cannot construct the polynomial $A_j(x)$ as it does not know the α-values of the authorized users Auth_j in session j and the blind value θ_j used in session j.

From View, $\mathcal{A}'$ knows only α_k for all $U_k \in R_j$, $\beta_1, \ldots, \beta_{j-1}$ 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

(i) the guess β'_j of $\mathcal{A}'$ for β_j is correct; and
(ii) $\mathcal{A}'$ knows the backward key K^B_{m-j+1}.

The condition (i) occurs if either of the following two holds:

- $\mathcal{A}'$ is able to choose $\beta'_j \in F_q$ so that $\beta'_j = \beta_j$, the probability of which is $1/q$ (negligible for large q).
- $\mathcal{A}'$ is able to generate β_j from View. Note that from View, $\mathcal{A}'$ knows $\beta_1, \ldots, \beta_{j-1} \in F_q$. Observe that $\beta_1, \ldots, \beta_{j-1}$ are generated by a cryptographically secure CSPRBG. Thus if $\mathcal{A}'$ is able to generate β_j from the known random numbers $\beta_1, \ldots, \beta_{j-1}$, then the CSPRBG is insecure, leading to a contradiction.

The condition (ii) occurs if either of the following two holds:

- $\mathcal{A}'$ is able to compute the access polynomial $A_j(x)$ (or $A_j(\alpha_k)$ for some $U_k \in R_j$) from View and consequently can recover the backward key $K^B_{m-j+1} = h_j(x)/A_j(x)$. From View, $\mathcal{A}'$ knows α_k for all $U_k \in R_j$ and with this knowledge it is computationally infeasible for $\mathcal{A}'$ (or coalition R_j) to learn α_i for $U_i \in \mathsf{Auth}_j$ under the security of CSPRBG. Moreover, $\mathcal{A}'$ will not be able to compute $A_j(x)$ as mentioned earlier. Consequently, $\mathcal{A}'$ will not be able to recover K^B_{m-j+1} from $\mathcal{B}_j$.
- $\mathcal{A}'$ 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 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 and the security of the CSPRBG. This is forward secrecy. We can also prove the computational security for backward secrecy of our construction using the similar arguments as above considering a coalition of new joined users. The only difference in the proof is that this coalition of new users joined in and after session j knows all the backward keys, but they do not know $\beta_1, \ldots, \beta_{j-1}$ and consequently are unable to compute the past session keys they were unauthorized to.

We will now show that our construction satisfies all the conditions required by Definition 1.

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 $R_j \subseteq \mathcal{U}$ of users revoked in and before session j, and any non-revoked user $U_i \notin R_j$, we show that the coalition R_j knows nothing about the personal secret $S_i = (\alpha_i; \beta_{s_i}, \ldots, \beta_j, \ldots, \beta_{t_i})$ of U_i with life cycle (s_i, t_i), $1 \leq s_i \leq t_i \leq m$. For any session j, U_i uses α_i and β_j as its personal secret. The coalition R_j may at most learn $\beta_1, \ldots, \beta_{j-1}$ and the probability of R_j to guess β_j is negligible under the cryptographic security of CSPRBG. Similarly, it is computationally infeasible for coalition R_j to learn α_i for $U_i \in \mathsf{Auth}_j$ from the set $\{\alpha_k : U_k \in R_j\}$ under the security of CSPRBG.

(c) The session key SK_j for the j-th session is computed from two parts: backward key K^B_{m-j+1} and random number β_j. Note that β_j is part of personal key of an unauthorized user $U_i \in \mathsf{Auth}_j$ that U_i receives from GM before or when U_i joins the group and $K^B_{m-j+1} = h_j(\alpha_i)/A_j(\alpha_i)$ is recovered by U_i from the broadcast message $\mathcal{B}_j$. Note that $A_j(\alpha_i) = 1$ for $U_i \in \mathsf{Auth}_j$ and is random for $U_i \notin \mathsf{Auth}_j$. 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) (Revocation property) Let $R_j \subseteq \mathcal{U}$ be a set of users revoked in and before session j who 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 the backward key K^B_{m-j+1}, and the knowledge of the personal secret α_i, β_j of user $U_i \in \mathsf{Auth}_j$. The coalition R_j knows the set $\{\alpha_k : U_k \in R_j\}$. The coalition R_j cannot compute α_i for $U_i \in \mathsf{Auth}_j$ from the set $\{\alpha_k : U_k \in R_j\}$ by the security of CSPRBG. Moreover, $A_j(x)$ is not publicly computable as discussed earlier. This in turn makes K^B_{m-j+1} appears random to all users in R_j. Moreover the coalition knows at most $\beta_1, \ldots, \beta_{j-1}$ and guessing β_j is negligible under the security of CSPRBG. Therefore, SK_j is completely safe from R_j in computation point of view.

3) (a) (Self-healing property) As shown in Section 3.2, user U_i can efficiently recover all missed session keys.

(b) We can prove using similar arguments as the proof of claim that our construction is computationally secure for resisting coalition under the assumption that the CSPRBG is cryptographically secure. We omit the proof here due to space constraint which will be avalible in the full version of the paper.

We now show that our construction satisfies all the conditions required by Definition 2.

1) (Forward secrecy) Let $R_j \subseteq \mathcal{U}$ 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 α_i for all $U_i \in \mathsf{Auth}_j$, K^B_{m-j+1} and β_j. Determining α_i for $U_i \in \mathsf{Auth}_j$ from the set $\{\alpha_k : U_k \in R_j\}$ is infeasible by the security of CSPRBG. 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

Table 1. Comparison among different self-healing key distribution schemes in j-th session ($k_i = t_i - s_i + 1$, where (s_i, t_i) is the life cycle assigned to user U_i by the GM; Auth_j is the set of authorized users in the j-th session; T_j is a threshold on the number of revoked users which depend on the monotone decreasing access structure; and t is the maximum number of revoked users)

Schemes	Storage Overhead	Communication Overhead	Computation Overhead
Construction 3 of [9]	$(m-j+1)^2 \log q$	$(mt^2 + 2mt + m + t) \log q$	$2mt^2 + 3mt - t$
Scheme 3 of [7]	$2(m-j+1) \log q$	$[(m+j+1)t + (m+1)] \log q$	$mt + t + 2tj + j$
Scheme 2 of [2]	$(m-j+1) \log q$	$(2tj + j) \log q$	$2j(t^2 + t)$
Construction 1 of [6]	$(m-j+1) \log q$	$(tj + j - t - 1) \log q$	$2tj + j$
Construction 1 of [3]	$(m-j+2) \log q$	$(t+1) \log q$	$2t + 1$
Construction 2 of [3]	$(m-j+2) \log q$	$(t+1) \log q$	$2(t^2 + t)$
Construction of [4]	$(m-j+2) \log q$	$(T_j + 1) \log q$	$2(T_j^2 + T_j)$
Construction of [10]	$(2k_i + 1) \log q$	$(T_j + 1) \log q$	$2(T_j^2 + T_j)$
Our Construction	$(k_i + 1) \log q$	$(\lvert \mathsf{Auth}_j \rvert + 2) \log q$	$\lvert \mathsf{Auth}_j \rvert + 1$

construction is forward secure. Moreover the coalition knows at most $\beta_1, \ldots, \beta_{j-1}$ and guessing β_j is negligible under the security of CSPRBG.
2) (Backward secrecy) Let $J_j \subseteq \mathcal{U}$ 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 β_{j_1}. Now when a new member U_v joins the group starting from session $j+1$, the GM gives U_v at most $\beta_{j+1}, \ldots, \beta_m$, together with the value α_v. Hence it is computationally infeasible for the newly joint member to trace back for previous β_{j_1} under the security of CSPRBG for $j_1 \leq j$. Consequently, our protocol is backward secure.

5 Performance Analysis

Table 1 shows comparisons of different self-healing schemes in terms of storage, communication and computation. We use the one-way key chain based approach of self-healing mechanism introduced in [3,4] which yields computationally secure and efficient scheme as no history of revoked users are sent during broadcast.

The most prominent improvement of our scheme over the previous self-healing key distributions [2,6,7,9] is that the communication complexity and computation cost in our construction does not increase as the number of session grows, but as the number of authorized users in a session grows.

As mentioned earlier, our construction is based on [3,4]. However we have the following enhancements:

(a) No forward key chain is used in our construction unlike [3,4].

(b) We make use of access polynomial instead of revocation polynomial. Access polynomial is computable only by authorized users, whereas revocation polynomial is publicly computable.

(c) Contrary to [3,4], each U_i in our construction is pre-assigned a life cycle (s_i, t_i) by the GM following the work of [10]. Thus user U_i can participate in $k_i = t_i - s_i + 1$ sessions and can not revoke before session t_i is over.

(d) In contrast to [3,4], we have been able to resist collusion attack in our construction by using pre-selected random numbers $\beta_1, \ldots, \beta_m$ (fixed) as part of users' secret keys. A user U_i with life cycle (s_i, t_i) is given only $k_i = t_i - s_i + 1$ values $\beta_{s_i}, \ldots, \beta_{t_i}$ and a value α_i as part of its secret key by the GM via a secure communication channel between them at the initial setup. As compared to [3,4], we get less storage for our scheme. The communication and computation costs at the j-th session for our scheme are linear to Auth_j, where Auth_j is the set of authorized users in session j. Our scheme has less computation and communication overhead as compared to [3] as long as $\mathsf{Auth}_j < t$ where t is a threshold on the number of revoked users in [3].

(e) The new scheme allows temporary revocation. Unlike previous self-healing key distribution schemes, revoked users may join at later sessions with new identities without violating any security and can get only the keys of the sessions it was in. Thus our scheme is more flexible as there is no restriction on the number of revoked users. Any number of users can leave/join the group and a user can join/leave as many times as it wishes. Most of the previous schemes constrained the number of revoked users to the threshold t. If more than t users are revoked, the security of the constructions cannot be guaranteed. Our scheme overcomes this limitation and thus more practical as it increases reliability of communication channel.

(f) **Denial of service attacks:** Availability is of critical business importance from an information security and business perspective. By availability we mean that a system is working and any attack that prevents the system working is known as a denial of service (DOS) attack. DOS attacks are not interested in breaking encryption or recovering keys, just in reducing availability. DOS attack scenarios are discussed in [11]. Use of the revocation polynomial in self healing systems actually facilitates a DOS attack. The attacker in this case colludes with others to increase the number of revoked users above the threshold t thus stopping the system. Using the access polynomial approach is resilient against this attack as it does not care how many users are revoked.

We adapt the similar approach as [10] to achieve resistance to collusion attacks and the ability of revoked users to rejoin the group. However, in contrast to [10], we done away with forward hash key chains. Consequently, our scheme is more efficient than [10] in terms of both storage and computation cost. Moreover, if $|\mathsf{Auth}_j| < T_j$, the communication cost in our scheme at the j-th session is less than that in [10].

6 Conclusion

We have enhanced an existing one-way key chain based self-healing key distribution by fixing the problem of collusion attack between the revoked users and the newly joined users. We have used the concept of access polynomial and assigned a pre-arranged life cycle on each user. Our scheme is robust and efficient as compared to the most previous schemes. It does not collapse as the number of revoked users exceeds a threshold value, which increases the availability of communication channel by reducing revocation based denial of service (DOS) attacks. Our scheme does not forbid revoked users from rejoining in later sessions unlike the existing self-healing key distribution schemes. This again has commercial advantages. The proposed scheme has been proven to be computationally secure and resists collusion between new joined users and revoked users together with forward and backward secrecy in an appropriate security model. Such security properties greatly increase confidence in a system.

References

1. Alexi, Chor, Goldreich, Schnorr: RSA Rabin Bits are $1/2+1/\mathsf{poly}(logn)$ secure. In: Proceedings of the IEEE 25th Annual Symposium on Foundations of Computer Science, pp. 449–557 (1984)
2. Blundo, C., D'Arco, P., Santis, A., Listo, M.: Design of Self-healing Key Distribution Schemes. Design Codes and Cryptology 32, 15–44 (2004)
3. Dutta, R., Chang, E.-C., Mukhopadhyay, S.: Efficient Self-Healing Key Distributions with Revocation for Wireless Network using One Way Key Chains. In: Katz, J., Yung, M. (eds.) ACNS 2007. LNCS, vol. 4521, pp. 385–400. Springer, Heidelberg (2007)
4. Dutta, R., Mukhopadhyay, S., Das, A., Emmanuel, S.: Generalized Self-Healing Key Distribution using Vector Space Access Structure. In: Das, A., Pung, H.K., Lee, F.B.S., Wong, L.W.C. (eds.) NETWORKING 2008. LNCS, vol. 4982, pp. 612–623. Springer, Heidelberg (2008)
5. Goldreich, O.: Foundations of Cryptography: Basic Tools. Cambridge University Press, Cambridge (2001)
6. Hong, D., Kang, J.: An Efficient Key Distribution Scheme with Self-healing Property. IEEE Communication Letters 2005, 9, 759–761 (2005)
7. 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)
8. 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)
9. 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)
10. Tian, B., Han, S., Dillon, T.-S., Das, S.: A Self-Healing Key Distribution Scheme Based on Vector Space Secret Sharing and One Way Hash Chains. In: Proceedings of IEEE WoWMoM 2008 (2008)
11. Tipton, H.: Official (ISC)2- Guide to The CISSP-CBK, 1st edn. Auerbach Publications (2006)
12. Zou, X.K., Dai, Y.S.: A Robust and Stateless Self-Healing Group Key Management Scheme. In: ICCT 2006, vol. 28, pp. 455–459 (2006)