Trade-Off between Collusion Resistance and User Life Cycle in Self-Healing Key Distributions with t-Revocation

Ratna Dutta
Claude Shannon Institute
NUI, Maynooth
Co. Kildare, IRELAND
ratna.dutta@nuim.ie

Sourav Mukhopadhyay
School of Electronic Engineering
Dublin City University
Dublin 9, IRELAND
msourav@eeng.dcu.ie

Tom Dowling
Claude Shannon Institute
NUI, Maynooth
Co. Kildare, IRELAND
tdowling@cs.nuim.ie

***Abstract*— We solve the problem of resisting the collusion attack in the one-way hash chain based self-healing key distributions introduced by Dutta et al., coupling it with the pre-arranged life cycle based approach of Tian et al. that uses the same self-healing mechanism introduced in Dutta et al. Highly efficient schemes are developed compared to the existing works with the trade-off in pre-arranged life cycles on users by the group manager and a slight increase in the storage overhead. For scalability of business it is often necessary to design more innovation 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 schemes fit into such business environment. The proposed schemes are proven to be computationally secure and resist collusion between new joined users and revoked users together with forward and backward secrecy. The security proof is in an appropriate security model. Moreover, our schemes do not forbid revoked users from rejoining in later sessions unlike the existing self-healing key distribution schemes.**

***Index Terms*— session key distribution, self-healing, revocation, wireless networks, access structure, computational security, forward and backward secrecy.**

I. Introduction

Efficient key distribution and key management over reliable channel has attracted much research interest and many classic and efficient schemes have been proposed [1], [4]. However these schemes are more appropriate for wired networks rather than wireless networks. Staddon *et al.* [15] introduced the self-healing key distribution which addresses the problem of how to distribute session keys over an unreliable channel. Since then, self-healing key distribution has received much attention.

The central 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 they have lost some previous key distribution messages, without requesting additional transmissions from the group manager. This reduces network traffic and the risk of user exposure through traffic analysis and also decreases the work load on the group manager. 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. Self-healing key distribution can find applications for many settings in wireless network, military oriented applications, rescue missions, scientific explorations, broadcast transmissions and various Internet services, where session keys are used only for short time periods.

Following the pioneering works by Staddon *et al.* [15], a number of self-healing key distribution approaches are proposed [15], [9], [2], [3], [7], [11], [12], [13] to achieve unconditional security in formal generalized model with improved efficiency. Further improvements in efficiency are obtained by relaxing the security slightly - from unconditional to computational [8], [5], [10]. The schemes [12], [13], [17] are based on vector space access structure instead of Shamir's [14] secret sharing. The hash chain based schemes [8], [5], [10] are computationally secure and are highly efficient compared to the existing unconditionally secure schemes. 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. Among the collusion resistance self-healing key distribution schemes [2], [3], [12], [13], [16], [17], only [17] is hash chain based and uses the same self-healing mechanism as introduced in [5]. We address the problem of achieving collusion resistance for [5] following the approach of [17], with better efficiency gains in computation and storage. Similar to [17], we assume that a user can not choose the session for its revoked on its own. Rather the user is assigned a pre-arranged life cycle by the group manager and is forcefully revoked once its life cycle finishes. With this trade-off and a slight increase in the storage as compared to [5], we achieve the following:

(a) Our schemes resist collusion between the new joined users and the revoked users (unlike [5]), besides keeping forward and backward secrecy.

(b) Our schemes allow revoked users to join in later sessions with new identities, while this rejoining is prohibited for all the existing hash chain based self-healing key distribution schemes including [5] (except [17]).

(c) Storage and computation overheads in our scheme are less than in [17] as we do not use forward key chain.

978-1-4244-4457-1/09/$25.00 ©2009 IEEE

Self-healing key distribution is an ideal candidate to establish session keys in large and dynamic wireless networks in which session keys can be used only for a short time period due to frequent membership change. Thus assigning each user a pre-arranged life cycle by the group manager and not allowing the user to revoke before its life cycle completes, has natural appeal in many applications. Several innovative business models allow contractual subscription or rental by the service provider for the scalability of business and do not allow the user to revoke before his contract is terminated. Our schemes are suitable for such applications. Moreover, rejoining of revoked users can be done in our schemes at later session with new identities without compromising security, unlike the existing self-healing schemes.

II. Preliminaries

A. Key Distribution and Self-Healing

Consider the following scenario for pay-per-view TV channel. Suppose $\{U_1, \ldots, U_n\}$ is a dynamically changing group of users (clients) and GM $\notin \{U_1, \ldots, U_n\}$ is the group manager (the cable operator). The problem is how the GM can securely communicate with its dynamically changing group of clients over an insecure broadcast channel, so that only authorized clients (who pay) may view the content broadcast by the GM. The GM encrypts the content using a session key. We need a mechanism of distributing this session key in such a way that only the authorized users can recover this session key and decrypt the encrypted content. This mechanism is referred to as the key distribution problem. Our goal is to minimize the overhead for this key distribution keeping the following few issues in mind: (a) group-rekeying is needed on each membership change; (b) depending on specific nature of applications, we can adopt periodic group-rekeying; (c) efficient and secure revocation as well as joining mechanisms are required for dynamic groups.

On top of this, U_i may be off-line for some time due to power failure and may need to recover lost session keys immediately after going on-line again. The Self-healing property enables qualified users to recover lost session keys on their own, without requesting additional transmission from the GM.

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
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 and before session j
$\mathcal{H}$	: a cryptographically secure one-way function
S^B	: backward key seed generated by the GM
K_i^B	: i-th backward key in the backward key chain

B. One-Way Functions

Our constructions for self-healing key distribution are based on the practical intractability of one-way functions. Informally speaking, a one-way function $f : A \rightarrow B$ satisfies the following two properties where A and B are two finite sets: (a) f is easy to compute; and (b) f is hard to invert, i.e., it is difficult to get x from $f(x)$. See [6] for a formal definition of one-way function.

C. Our Security Model

We now state the following definitions that are aimed to computational security for session key distribution adopting the security model of [9], [15].

Definition 2.1: (Session Key Distribution with privacy [15]) Let $i, t \in \{1, \ldots, n\}$ 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, $1 \leq j_1 < j < j_2 \leq m$:

(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 $1 \leq j_1 < j < j_2 \leq m$. For any disjoint subsets $L_1, L_2 \subset \mathcal{U}$ where L_1 is a coalition of users removed before session j_1 and L_2 is a coalition of users joined from session j_2, and $|L_1 \cup L_2| \leq t$, the set $\{Z_{l,j}\}_{U_l \in L_1, 1 \leq j \leq j_1} \cup \{Z_{l,j}\}_{U_l \in L_2, j_2 \leq j \leq m}$ cannot determine the session key SK_j, $j_1 < j < j_2$. *i.e.* SK_j can not be obtained by the coalition $L_1 \cup L_2$.

Definition 2.2: (t-wise forward and backward secrecy [9]) 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 SK_j, even with the knowledge of group keys $\mathsf{SK}_{j+1}, \ldots, \mathsf{SK}_m$ after session j.

III. Self-Healing Key Distributions with t-revocation

In this section, we present three efficient constructions for self-healing key distribution with t-revocation capability and refer them as $\mathsf{SHKD}-1$, $\mathsf{SHKD}-2$ and $\mathsf{SHKD}-3$ respectively. $\mathsf{SHKD}-1$ uses Shamir's (t,n)-threshold secret sharing scheme For $\mathsf{SHKD}-2$ and $\mathsf{SHKD}-3$, we adopt the same self-healing technique and make use of a revocation polynomial for efficient revocation instead of using secret sharing schemes.

For all the following constructions, 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.

A. Key Distribution $\mathsf{SHKD}-1$

• *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 GM randomly picks an initial 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:

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

The GM also selects at random m numbers $\beta_1,\ldots,\beta_m \in F_q$. 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 prearranged 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 get 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 sessions $S_i = \{f_{s_i}(i),\ldots,f_{t_i}(i);\beta_{s_i},\ldots,\beta_{t_i}\}$ 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 $|R_j| \le t$ 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 evaluates the current session key

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

A user U_k who either does not know its private information $(f_j(k);\beta_j)$ or who is a revoked user in R_j, *i.e.* $U_k \in W_j \cup R_j$, cannot compute $f_j(0)$ because U_k knows insufficient number of points to interpolate the polynomial $f_j(x)$ from the broadcast message $\mathcal{B}_j$. Consequently, U_k cannot recover the backward key K_{m-j+1}^B and hence the j-th session key SK_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 $v \in F_q$, 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 $S_v = \{f_{s_v}(v),\ldots,f_{t_v}(v);\beta_{s_v},\ldots,\beta_{t_v}\}$ 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 $2(t_i - s_i + 1)\log q$ bits.

– *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.

– *Computation overhead:* 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.

B. Key Distribution $\mathsf{SHKD}-2$

• *Setup*: The group manager randomly picks an initial 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 $1 \le i \le m$. The GM also selects at random m numbers $\beta_1,\ldots,\beta_m \in F_q$. The j-th session key is computed as $\mathsf{SK}_j = \beta_j + K^B_{m-j+1}$. 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 is first assigned a prearranged life cycle (s_i, t_i), where $1 \le s_i < t_i \le m$, by the GM. 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 $S_i = \{f_{s_i}(i), \ldots, f_{t_i}(i); \beta_{s_i}, \ldots, \beta_{t_i}\}$ 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 $|R_j| = w_j \le t$. In the j-th session the group manager locates the backward key K^B_{m-j+1} in the backward key chain and computes the polynomials

$$r_j(x) = (x - l_1) \cdots (x - l_{w_j}),$$
$$h_j(x) = K^B_{m-j+1} r_j(x) + f_j(x).$$

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

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

Note that from the broadcast message $\mathcal{B}_j$, one gets R_j, thereby gets the indices of all revoked users, and consequently can easily compute the revocation polynomial. Finally, U_i evaluates the current session key

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

• *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 $v \in F_q$, 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 $S_v = \{f_{s_v}(v), \ldots, f_{t_v}(v); \beta_{s_v}, \ldots, \beta_{t_v}\}$ 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 $2(t_i - s_i + 1)\log q$ bits.

– *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.

– *Computation overhead:* The computation cost for key management is $2(t + 1)$, which is the number of multiplication operations needed to find two points on two t-degree polynomials.

C. *Key Distribution* $\mathsf{SHKD} - 3$

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

$$K^B_i = \mathcal{H}(K^B_{i-1}) = \mathcal{H}^{i-1}(S^B)$$

for $1 \le i \le m$. The GM also selects at random m numbers $\beta_1, \ldots, \beta_m \in F_q$. The j-th session key is computed as $\mathsf{SK}_j = \beta_j + K^B_{m-j+1}$. The group manager chooses independently and uniformly at random a polynomial $f(x, y) = a_{0,0} + a_{1,0}x + a_{0,1}y + \cdots + a_{t,t}x^t y^t$ from $F_q[x, y]$, where t is a positive integer, $t < m, n$. Each user U_i is first assigned a prearranged life cycle (s_i, t_i), where $1 \le s_i < t_i \le m$, by the GM. 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 $S_i = \{f(i, y); \beta_{s_i}, \ldots, \beta_{t_i}\}$ 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 $|R_j| = w_j \le t$. In the j-th session the group manager locates the forward key K^F_j in the forward key chain and the backward key K^B_{m-j+1} in the backward key chain and computes the polynomials

$$r_j(x) = (x - l_1) \cdots (x - l_{w_j}),$$
$$h_j(x) = K^B_{m-j+1} r_j(x) + f(x, \beta_j).$$

The polynomial $r_j(x)$ is called the revocation polynomial in session j and the polynomial $f(x, \beta_j)$ 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

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

Finally, U_i computes the current session key

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

Note that $r_j(x)$ is computable from the broadcast message $\mathcal{B}_j$ which carries the information of revoked uses' IDs and β_j is transmitted to the non-revoked user U_i during the Setup phase.

• *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 $v \in F_q$, assigns a life cycle (s_v, t_v) to the new user with $s_v = j$, computes the personal secret keys $S_v = \{f(v, y); \beta_{s_v}, \ldots, \beta_{t_v}\}$ 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 $(t_i - s_i + t + 2) \log q$ bits.

– *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.

–*Computation overhead:* The computation cost for key management is $2(t + 1)$, which is the number of multiplication operations needed to find two points on two t-degree polynomials.

D. Self-Healing

We now explain our self-healing mechanism for the construction $\mathsf{SHKD} - 1$. The self healing for the constructions $\mathsf{SHKD} - 2$ and $\mathsf{SHKD} - 3$ can be performed in a similar way. 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 desired by Definition 2.1 $3(a)$ using the following steps.

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

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]. 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 2.1 $3(b)$. 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 constructions satisfy this property as illustrated below for the scheme $\mathsf{SHKD} - 1$. The similar arguments hold for the other two constructions. 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 $j_1 \leq j < j_2$. This is true because in the worst case, the coalition knows $S_i = \{f_1(i), \ldots, f_{j_1-1}(i); \beta_1, \ldots, \beta_{j_1-1}\}$ for $U_i \in L_1$, $S_i = \{f_{j_2}(i), \ldots, f_m(i); \beta_{j_2}, \ldots, \beta_m\}$ for $U_i \in L_2$, and $\mathcal{B}_1, \ldots, \mathcal{B}_m$. For each session j, $j_1 \leq j < j_2$, 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 $j_1 \leq j < j_2$. Consequently, all the guess for SK_j with $j_1 \leq j < j_2$ are equi-probable.

IV. Security Analysis

The following theorems state that our constructions realize self-healing key distribution schemes with revocation capability. The proofs follow the argument in [5]. We omit the proofs here due to space constraints. They are available in the full version of the paper.

Theorem 4.1: Construction $\mathsf{SHKD} - 1$ is secure, self-healing session key distribution scheme with privacy, t-revocation capability with respect to Definition 2.1 in our security model as described in Section D and achieves t-wise forward and backward secrecy with respect to Definition 2.2 in the model.

Theorem 4.2: Construction $\mathsf{SHKD} - 2$ is secure, self-healing session key distribution scheme with privacy, t-revocation capability with respect to Definition 2.1 in the model and achieve t-wise forward and backward secrecy with respect to Definition 2.2 in the model.

Theorem 4.3: Construction $\mathsf{SHKD}{-}3$ is secure, self-healing session key distribution scheme with privacy, t-revocation capability with respect to Definition 2.1 in the model and achieves t-wise forward and backward secrecy with respect to Definition 2.2 in the model.

V. Performance Analysis

Comparison of storage overhead, communication complexity and computation cost of each user (not the GM) in our constructions $\mathsf{SHKD} - 1$, $\mathsf{SHKD} - 2$, $\mathsf{SHKD} - 3$ with the existing self-healing session key distribution schemes is provided in Table I. 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. On the other hand, we achieve less computation cost. 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 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 self-healing key distributions [2], [7], [9], [15].

Our constructions are based on [5] with the following subtle difference:

TABLE I

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

Schemes	Storage Overhead	Communication Overhead	Computation Overhead
Construction 3 of [15]	$(m-j+1)^2 \log q$	$(mt^2 + 2mt + m + t) \log q$	$2mt^2 + 3mt - t$
Scheme 3 of [9]	$2(m-j+1) \log q$	$[(m+j+1)t + (m+1)] \log q$	$mt + t + 2tj + j$
Scheme 2 of [2]	$(m-j+1) \log q$	$(2tj + j) \log q$	$2j(t^2 + t)$
Construction 1 of [7]	$(m-j+1) \log q$	$(tj + j - t - 1) \log q$	$2tj + j$
Construction 1 of [5]	$(m-j+1) \log q$	$(t+1) \log q$	$2t + 1$
Construction 2 of [5]	$(m-j+1) \log q$	$(t+1) \log q$	$2(t^2 + t)$
$\mathsf{SHKD}-1$	$2k_i \log q$	$(t+1) \log q$	$2(t^2 + t)$
$\mathsf{SHKD}-2$	$2k_i \log q$	$(t+1) \log q$	$2t + 1$
$\mathsf{SHKD}-3$	$(k_i + t + 1) \log q$	$(t+1) \log q$	$2t + 1$

(a) No forward key chain is used.

(b) Each user U_i is pre-assigned a life cycle (s_i, t_i) by the GM following the work by [17]. This means user U_i can participate in $k_i = t_i - s_i + 1$ sessions and can not revoke before session t_i is over.

(c) In contrast to [5], we have been able to resist collusion attack in our constructions by using pre-selected random numbers $\beta_1, \ldots, \beta_m$ (fixed) as part of users' secret keys apart from values on polynomials. A user U_i with life cycle (s_i, t_i) is given only $k_i = t_i - s_i + 1$ values $f_{s_i}(i), \ldots, f_{t_i}(i)$ and the additional values $\beta_{s_i}, \ldots, \beta_{t_i}$ as part of its secret key by the GM via a secure communication channel between them at the initial setup. As compared to [5], we get increased storage for $\mathsf{SHKD}-1$ and $\mathsf{SHKD}-2$ if $k_i > \frac{m-j+1}{2}$, and for $\mathsf{SHKD}-3$ if $k_i > m-j+1-(t+1)$. The communication and computation costs for all our schemes are the same as in [5].

(d) Revoked users may join at later sessions with new identities without violating any security.

We adapt the similar approach as [17] to achieve resistance to collusion attacks and the ability of revoked users to rejoin the group. However, in contrast to [17], we done away with forward hash key chains. Consequently, our schemes are more efficient than [17] in terms of both storage and computation cost.

VI. CONCLUSION

We introduced three efficient self-healing key distribution schemes with t-revocation capability. Our proposed key distribution mechanism reduces storage, communication and computation costs over the previous approaches, and is scalable to very large groups in highly mobile, volatile and hostile wireless network. Our schemes are properly analyzed in an appropriate security model and are proven to be computationally secure and achieve both forward and backward secrecy. Also our schemes can resist collusion between the new joined user and the revoked users, with a trade-off in the storage overhead and assigning pre-determined life cycle to each user by the group manager. These trade-offs are often allowed in certain business models such as rental or subscription. Our setup therefore allows each user to choose its joining session at its will, but the session for its revocation is pre-selected by the group manager. However, unlike the existing self-healing key distribution schemes, rejoining of revoked users is permitted in our schemes at a later session with new identities.

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

[3] C. Blundo, P. D'Arco, A. Santis, M. Listo. *Definitions and Bounds for Self-healing Key Distribution.* Proceedings of the 31st ICALP 04, LNCS 3142, pp. 234-245, Springer-Verlag, 2004.

[4] 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.

[5] R. Dutta, E-C. Chang, S. Mukhopadhyay. *Efficient Self-Healing Key Distributions with Revocation for Wireless Network using One Way Key Chains.* Proceedings of the 5th ACNS 2007, LNCS, Springer-Verlag, 2007.

[6] O. Goldreich. *Foundations of Cryptography: Basic Tools.* Cambridge University Press, Cambridge, 2001.

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

[8] Y. Jiang, C. Lin, M. Shi and X. Shen. *Self-healing Group Key Distribution with Time-limited Node Revocation for Wireless Sensor Networks.* Ad Hoc Networks, Vol. 5, No. 1, pp. 14-23, 2007.

[9] 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.

[10] F. Kausar, S. Hassian, J. H. Park, and A. Masood. *Secure Group Communication with Self-Healing and Rekeying in Wireless Sensor Networks.* Proceedings of the Third International Conference, MSN 2007, pp. 737-748, 2007.

[11] 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.

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

[13] G. Saez. *Self-healing Key Distribution Schemes with Sponsorization.* IFIP International Federation for Information Processing'05, LNCS 3677, pp. 22-31, Springer-Verlag, 2005.

[14] A. Shamir. *How to Share a Secret.* Communications of the ACM, Vol. 22, No. 11, pp. 612-613, 1979.

[15] 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.

[16] B. Tian and M. He. *Self-Healing Key Distribution Scheme with Novel Properties.* International Journal of Network Security, Vol. 7, No. 2, pp. 147-152, 2008.

[17] B. Tian, S. Han, T-S Dillon, and S. Das. *A Self-Healing Key Distribution Scheme Based on Vector Space Secret Sharing and One Way Hash Chains.* Proceedings of IEEE WoWMoM 2008.