Efficient Self-Healing Group Key Distribution with Revocation Capability*

Donggang Liu
Cyber Defense Laboratory
Department of Computer Science
NC State University
Raleigh, NC 27695-8207
dliu@unity.ncsu.edu

Peng Ning
Cyber Defense Laboratory
Department of Computer Science
NC State University
Raleigh, NC 27695-8207
ning@csc.ncsu.edu

Kun Sun
Cyber Defense Laboratory
Department of Computer Science
NC State University
Raleigh, NC 27695-8207
ksun3@unity.ncsu.edu

ABSTRACT

This paper presents group key distribution techniques for large and dynamic groups over unreliable channels. The techniques proposed here are based on the self-healing key distribution methods (with revocation capability) recently developed by Staddon et al. [27]. By introducing a novel personal key distribution technique, this paper reduces (1) the communication overhead of personal key share distribution from $O(t^2 \log q)$ to $O(t \log q)$, (2) the communication overhead of self-healing key distribution with t-revocation capability from $O((mt^2 + tm) \log q)$ to $O(mt \log q)$, and (3) the storage overhead of the self-healing key distribution with t-revocation capability at each group member from $O(m^2 \log q)$ to $O(m \log q)$, where t is the maximum number of colluding group members, m is the number of sessions, and q is a prime number that is large enough to accommodate a cryptographic key. All these results are achieved without sacrificing the unconditional security of key distribution. In addition, this paper presents two techniques that allow trade-off between the broadcast size and the recoverability of lost session keys. These two methods further reduce the broadcast message size in situations where there are frequent but short-term disruptions of communication and where there are long-term but infrequent disruptions of communication, respectively.

Categories and Subject Descriptors

C.2.0 [**Computer-communication networks**]: General–security and protection

General Terms

Design, Security

*This work is partially supported by NCSU Center for Advanced Computing & Communication. The authors would like to thank the anonymous reviewers for their valuable comments.

Keywords

key management, group key distribution, self-healing

1. INTRODUCTION

Wireless networks, especially mobile wireless ad hoc networks, are ideal candidates for communications in applications such as military operations, rescue missions, and scientific explorations, where there is usually no network infrastructure support. In situations where there are adversaries who may want to intercept and/or interrupt the communication, security becomes one of the top concerns. In particular, it is critical to make sure that the adversaries cannot access or interrupt the wireless communication, and even if they do, it is possible to recover from such compromises quickly.

A common way to ensure communication security is to encrypt and authenticate the messages. In mobile wireless networks, a sender may broadcast encrypted and/or authenticated messages to his/her team members, and only nodes with valid keys can have access to and/or verify these messages. The remaining challenge is how to distribute the cryptographic keys to valid nodes.

Theoretically, all techniques developed for secure group communications in traditional networks (e.g., LKH [32, 33]) can be used for key distribution in mobile wireless networks. However, some unique features of mobile wireless networks introduce new problems that have not been fully considered. First, nodes in mobile wireless networks may move in and out of range frequently, and sometimes be completely separate from the network. Moreover, the adversary may intentionally disrupt the wireless communication using various methods. Thus, techniques without fault tolerant features, or those that use error correction codes in traditional ways (e.g., Keystone [34]) cannot fully address this problem, especially in large, dynamic wireless networks (e.g., military networks consisting of mobile devices carried by soldiers, automatic weapons, sensing devices, etc.). Second, devices in mobile wireless networks are typically powered by batteries. It will reduce the lifetime of the batteries, and thus the availability of wireless devices, to adopt some power-consuming techniques such as public key cryptography. Thus, not all of the existing techniques are suitable for large and dynamic wireless networks.

Among existing group key distribution techniques, two methods are potential candidates for large mobile wireless networks: self-healing key distribution [27] and stateless key distribution [20]. Self-healing key distribution allows group members to recover lost session keys, while stateless group key distribution permits group members to get up-to-date session keys (without recovering lost keys) even if they miss some previous key distribution messages.

In this paper, we develop novel self-healing group key distribution schemes for large and dynamic groups over unreliable channels based on the techniques proposed in [27], aiming at addressing group key distribution in highly mobile, volatile and hostile wireless networks. By introducing a novel personal key distribution technique, we reduce (1) the communication overhead of personal key share distribution from $O(t^2 \log q)$ to $O(t \log q)$, (2) the communication overhead of self-healing key distribution with t-revocation capability from $O((mt^2 + tm) \log q)$ to $O(mt \log q)$, and (3) the storage overhead of the self-healing key distribution with t-revocation capability at each group member from $O(m^2 \log q)$ to $O(m \log q)$, where t is the maximum number of colluding group members, m is the number of sessions, and q is a prime number that is large enough to accommodate a cryptographic key. All these results are achieved without sacrificing the unconditional security of key distribution. In addition, we develop two techniques that allow trade-off between the broadcast size and the recoverability of lost session keys. These two methods address the situations where there are frequent but short-term disruptions of communication and where there are long-term but infrequent disruptions of communication, respectively.

The proposed key distribution schemes have several advantages, which make these schemes very attractive for large mobile wireless networks. First, the proposed techniques are self-healing. A wireless node can recover lost keys even if it is separated from the network when the keys are distributed. Second, the proposed techniques do not require heavy computation, and wireless nodes can get or recover keys by passively listening to broadcast key distribution messages. This is particularly important to devices in mobile wireless networks, which are typically powered by batteries. Reducing the computation and active communication can significantly reduce the power consumption and prolong the lifetime of wireless devices. Third, the proposed techniques distribute keys via true broadcast, conforming to the broadcast nature of wireless networks. Finally, the proposed techniques are scalable to very large groups. The processing, communication, and storage overheads do not depend on the size of the group, but on the number of compromised group members that may collude together.

Our contributions in this paper are as follows. The first, and most fundamental contribution is the novel personal key distribution scheme that allows efficient distribution of different key shares to different group members via a broadcast channel. Second, based on this technique, we develop an efficient self-healing key distribution scheme that requires less storage and communication overhead than those in [27]. Third, we further develop two ways to trade off the self-healing capability with broadcast size, allowing less communication overhead in bandwidth constrained applications.

The rest of this paper is organized as follows. Section 2 presents our model as well as notations to be used in this paper. Section 3 gives the details of our approaches. Section 4 reviews existing techniques related to group key distribution. Section 5 concludes this paper and points out some future directions.

2. OUR MODEL

Communication Model. To focus on the key distribution problem, we adopt a simplified group communication model. We assume that communication entities in a wireless network form groups to control access to broadcast messages. There may be more than one group with certain relationships between them (e.g., members of the captain group are also members of the soldier group). Without loss of generality, we will focus on the case of one group unless it is necessary to discuss multiple groups. The lifetime of a wireless network is partitioned into time intervals called *sessions*. The duration of sessions may be fixed or dynamic due to the change of group membership. There is one or several *group managers* that are responsible for distributing *group (session) keys* to a large number of authorized *group members*. Only group members with valid group keys can broadcast authenticated messages to other group members and access encrypted broadcast messages. A *sender* may transmit a broadcast message to the other group members (i.e., *receivers*) directly, or indirectly through network components (e.g., wireless routers) or other group members.

Mobile wireless networks are usually highly volatile. Wireless nodes may move in and out of range frequently, and there is usually no infrastructure support to guarantee reliable delivery of messages. Thus, we do *not* assume reliable communication in our system; a message sent to a group *may* or *may not* reach all the group members.

Threat Model. We assume an adversary may passively listen to, or actively insert, intercept and modify, or drop broadcast messages. Our goal is to ensure the group manager can distribute group keys to group members as long as the group members can get *some* of the broadcast messages. Certainly, our approach won't work if the adversary completely jams the communication channel. We assume there are other means to defeat signal jamming (e.g., spread spectrum). Moreover, we consider the possibility that the adversary may compromise one or more group members (e.g., by capturing and analyzing the devices). Our goal is to ensure that once detected, such group members will be revoked from the group, and the adversary has to compromise more than t devices to defeat our approach, where t is a system parameter.

Notations. We assume each group member is uniquely identified by an ID number i, where $i \in \{1, ..., n\}$ and n is the largest ID number, and denote the group member as U_i. All of our operations take place in a finite field F_q, where q is a sufficiently large prime number. Each group member U_i stores a personal secret $S_i \subseteq F_q$, which represents all information the group member may use to recover the session keys. We use $H(\cdot)$ to denote the entropy function of information theory [9]. We use K_j to denote the session key that the group manager distributes to the group members in session j, and k_i to denote the personal key of group member U_i. Note that to enable the group manager to revoke U_i when necessary, we cannot allow k_i to be computed only from S_i. Instead, k_i must also depend on information distributed by the group manager.

The group manager distributes the session key among the group via a broadcast message. We use $\mathcal{B}_j$ to denote the broadcast message, called the *session key distribution message*, that the group manager uses to distribute the group session key during session j. We use $z_{i,j}$ to denote what the group member U_i learns from its own personal secret S_i and $\mathcal{B}_j$. We use R_j to denote the set of revoked group members in session j, which contains all of the revoked members since the beginning of session key distribution. We reserve the letter t to represent the number of compromised group members. We would like to develop techniques that are resistant to adversaries who are able to compromise t group members (or, equivalently, the coalition of up to t revoked group members).

Goals. Our general goal is to develop efficient and unconditionally secure key distribution schemes for mobile wireless networks. The resulting techniques should be able to tolerate the volatile nature of mobile wireless networks as well as compromise of past group members. We are particularly interested in practical solutions that can be deployed in the current or next generation wireless networks.

To further clarify our goals and facilitate the later presentation, we give the following definitions.

DEFINITION 1. (Personal Key Distribution [27]) *Let* $t, i \in \{1, ..., n\}$. *In a* personal key distribution scheme $\mathcal{D}$, *the group manager seeks to establish a new key* $k_i \in F_q$ *with each group member* U_i *through a broadcast message* $\mathcal{B}$.

1. $\mathcal{D}$ *is a personal key distribution scheme if*
   (a) *for any group member* U_i, k_i *is determined by* S_i *and* $\mathcal{B}$ *(i.e.,* $H(k_i|\mathcal{B}, S_i) = 0$*),*
   (b) *for any set* $B \subseteq \{U_1, ..., U_n\}$, $|B| \leq t$, *and any* $U_i \notin B$, *the members in* B *are not able to learn anything about* S_i *(i.e.,* $H(k_i, S_i|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}) = H(k_i, S_i)$*), and*
   (c) *no information on* $\{k_i\}_{i \in \{1,...,n\}}$ *is learned from either the broadcast or the personal secrets alone (i.e.,* $H(k_1, ..., k_n|\mathcal{B}) = H(k_1, ..., k_n) = H(k_1, ..., k_n|S_1, ..., S_n)$*).*
2. $\mathcal{D}$ *has* t-revocation capability *if given any* $R \subseteq \{U_1, ..., U_n\}$ *such that* $|R| \leq t$, *the group manager can generate a broadcast* $\mathcal{B}$, *such that for all* $U_i \notin R$, U_i *can recover* k_i *(i.e.,* $H(k_i|\mathcal{B}, S_i) = 0$*), but the revoked group members cannot recover any of the keys (i.e.,* $H(k_1, ..., k_n|\mathcal{B}, \{S_{i'}\}_{U_{i'} \in R}) = H(k_1, ..., k_n)$*).*

DEFINITION 2. (Session Key Distribution with b-bit privacy) *Let* $t, i \in \{1, ..., n\}$ *and* $j \in \{1, ..., m\}$.

1. $\mathcal{D}$ *is a key distribution scheme with* b*-bit privacy if*
   (a) *for any member* U_i, K_j *is determined by* $z_{i,j}$, *which in turn is determined by* $\mathcal{B}_j$ *and* S_i *(i.e.,* $H(K_j|z_{i,j}) = 0$ *and* $H(z_{i,j}|\mathcal{B}_j, S_i) = 0$*),*
   (b) *for any* $B \subseteq \{U_1, ..., U_n\}$, $|B| \leq t$, *and* $U_i \notin B$, *the uncertainty of the members in* B *to determine* S_i *is at least* b *bits (i.e.,* $H(S_i|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m) \geq b$*), and*
   (c) *what members* $U_1, ..., U_n$ *learn from* $\mathcal{B}_j$ *can't be determined from the broadcasts or personal keys alone (i.e.,* $H(z_{i,j}|\mathcal{B}_1, ..., \mathcal{B}_m) = H(z_{i,j}) = H(z_{i,j}|S_1, ..., S_n)$*).*
2. $\mathcal{D}$ *has* t*-revocation capability if given any* $R \subseteq \{U_1, ..., U_n\}$, *where* $|R| \leq t$, *the group manager can generate a broadcast* $\mathcal{B}_j$, *such that for all* $U_i \notin R$, U_i *can recover* K_j *(i.e.,* $H(K_j|\mathcal{B}_j, S_i) = 0$*), but the revoked members cannot (i.e.,* $H(K_j|\mathcal{B}_j, \{S_{i'}\}_{U_{i'} \in R}) = H(K_j)$*).*
3. $\mathcal{D}$ *is self-healing if the following are true for any* $1 \leq j_1 < j < j_2 \leq m$*:*
   (a) *For any* U_i *who is a member in sessions* j_1 *and* j_2, K_j *is determined by the set,* $\{z_{i,j_1}, z_{i,j_2}\}$ *(i.e.,* $H(K_j|z_{i,j_1}, z_{i,j_2}) = 0$*).*
   (b) *For any disjoint subsets* $B, C \subset \{U_1, ..., U_n\}$, *where* $|B \cup C| \leq t$, *the set* $\{z_{i',j}\}_{U_{i'} \in B, 1 \leq j \leq j_1} \cup \{z_{i',j}\}_{U_{i'} \in C, m \geq j \geq j_2}$ *contains no information on the key* K_j *(i.e.,* $H(K_j|\{z_{i',j}\}_{U_{i'} \in B, 1 \leq j \leq j_1} \cup \{z_{i',j}\}_{U_{i'} \in C, m \geq j \geq j_2}) = H(K_j)$*).*

Our Definition 2 is a generalization of the notion of session key distribution in [27]. The difference lies in item 1(b). Both definitions are aimed at unconditional security. However, session key distribution in [27] requires that any coalition of at most t valid group members cannot get *any* information about another member's personal secret, while Definition 2 in our paper requires that the uncertainty of such a coalition to determine another member's personal secret is at least b bits. In other words, session key distribution in [27] doesn't allow any information leakage, while our Definition 2 allows certain information leakage as long as the uncertainty of the secret is at least b bits.

As a side note, we found that Construction 3 in [27] doesn't meet their criteria of session key distribution as claimed in their Theorem 1. Assume U_i is the member that the coalition wants to attack. Though it is shown in [27] that the coalition of at most t group members cannot get any information of U_i's share on each individual polynomial, the uncertainty of the personal secret S_i, which consists of a point on each of m^2 such polynomials, decreases when the coalition receives the broadcast messages. This is because the session key distributed to U_i for each session remains constant in multiple broadcast messages, and the coalition can get the sum of this key and a point on a polynomial for multiple polynomials. As a result, the uncertainty of all the related shares in S_i is determined by the uncertainty of this session key. Nevertheless, Construction 3 in [27] still meets the criteria specified in our Definition 2 with at least $m \log q$-bit privacy.

Security properties of a group key management system have been considered in the past [21, 29]. These security properties consist of (1) *group key secrecy*, which guarantees that it is at least computationally infeasible for an adversary to discover any group key, (2) *forward secrecy*, which guarantees that a passive adversary who knows a contiguous subset of old group keys cannot discover subsequent group keys, (3) *backward secrecy*, which guarantees that a passive adversary who knows a contiguous subset of group keys cannot discover preceding group keys, and (4) *key independence*, which is the combination of forward and backward secrecy.

These security properties have been studied for group key management systems such as CLIQUES [28] and ELK [21]. However, they are not sufficient in our framework, since each group member also has access to some secret information (i.e., S_i for U_i), which is used to compute the group keys. In particular, forward secrecy doesn't imply that the adversary cannot discover the subsequent group keys if he/she further has the secret information only known to some past group members, and backward secrecy doesn't guarantee that the adversary cannot discover the preceding group keys if he/she is further provided the secret information only known to some new group members. To clarify these requirements, we introduce the notions of t*-wise forward and backward secrecy*.

DEFINITION 3. (t-wise forward and backward secrecy) *Let* $t, i \in \{1, ..., n\}$ *and* $j \in \{1, ..., m\}$.

- *A key distribution scheme guarantees* t-wise forward secrecy *if for any set* $R \subseteq \{U_1, ..., U_n\}$, *where* $|R| \leq t$, *and all* $r \in R$ *are revoked before session* j, *the members in* R *together cannot get any information about* K_j, *even with the knowledge of group keys before session* j *(i.e.,* $H(K_j|\mathcal{B}_1, ..., \mathcal{B}_m, \{S_i\}_{U_i \in R}, K_1, ..., K_{j-1}) = H(K_j)$*).*
- *A key distribution scheme guarantees* t-wise backward secrecy *if for any set* $R \subseteq \{U_1, ..., U_n\}$, *where* $|R| \leq t$, *and all* $r \in R$ *join after session* j, *the members in* R *together cannot get any information about* K_j, *even with the knowledge of group keys after session* j *(i.e.,* $H(K_j|\mathcal{B}_1, ..., \mathcal{B}_m, \{S_i\}_{U_i \in R}, K_{j+1}, ..., K_m) = H(K_j)$*).*

Note that t-wise forward (backward) secrecy implies forward (backward) secrecy. Thus, ensuring t-wise forward and backward secrecy guarantees forward and backward secrecy, key independence, and group key secrecy. Moreover, it is easy to see that t-wise forward secrecy also implies t-revocation capability.

3. EFFICIENT SESSION KEY DISTRIBUTION WITH REVOCATION

In this section, we present our techniques for self-healing key distribution with revocation capability. Our techniques start with a novel personal key distribution scheme, in which the communication complexity is only $O(t \log q)$ to provide t-revocation capability. We then apply this technique to develop an efficient key distribution scheme in Section 3.2, and then reduce its storage requirement in Section 3.3. To further reduce the broadcast message size, we propose two kinds of trade-offs between the self-healing capability and broadcast message size in Section 3.4. The security of these schemes is guaranteed through a number of theorems. For space reasons, we only present the proof of Theorem 2. The proofs of the other theorems can be found in the full version of this paper [16].

One limitation of these schemes is that self-healing key distribution is restricted to m sessions. However, we note that the technique that extends the lifetime of the methods in [27] is also applicable to ours. Due to space reasons, we do not discuss it in this paper.

3.1 Personal Key Share Distribution

The purpose of personal key share distribution is to distribute keys to select group members so that each of the select (or non-revoked) group members shares a distinct personal key with the group manager, but the other (revoked) group members (as well as the adversary) cannot get any information of the keys. In our approach, the group manager broadcasts a message, and all the select group members derive their keys from the message.

Our approach chooses a random t-degree polynomial $f(x)$ from $F_q[x]$, and select $f(i)$ to be the personal key share for each group member U_i. The group manager constructs a single broadcast polynomial $w(x)$ such that for a select group member U_i, $f(i)$ can be recovered from the knowledge of $w(x)$ and the personal secret S_i, but for any revoked group member $U_{i'}$, $f(i')$ cannot be determined from $w(x)$ and $S_{i'}$.

Specifically, we construct $w(x)$ from $f(x)$ with the help of a *revocation polynomial* $g(x)$ and a *masking polynomial* $h(x)$ by computing $w(x) = g(x)f(x) + h(x)$. The revocation polynomial $g(x)$ is constructed in such a way that for any select group member U_i, $g(i) \neq 0$, but for any revoked group member $U_{i'}$, $g(i') = 0$. Each group member U_v has its own personal secret $S_v = \{h(v)\}$, which may be distributed by the group manager during setup via the secure communication channel between each group member and the group manager. Thus, for any select group member U_i, new personal key $f(i)$ can be computed by $f(i) = \frac{w(i)-h(i)}{g(i)}$, but for any revoked group member $U_{i'}$, new personal key cannot be computed because $g(i') = 0$. This scheme has the properties of unconditional security and t-revocation capability, which are guaranteed by Theorem 1.

SCHEME 1. *Personal key distribution with t-revocation capability. The purpose of this scheme is to distribute distinct shares of a target t-degree polynomial, $f(x)$, to non-revoked group members.*

1. Setup: *The group manager randomly picks a $2t$-degree masking polynomial, $h(x) = h_0 + h_1 x + ... + h_{2t}x^{2t}$, from $F_q[x]$. Each group member U_i gets the personal secret, $S_i = \{h(i)\}$, from the group manager via the secure communication channel between them.*
2. Broadcast: *Given a set of revoked group members, $R = \{r_1, r_2, ..., r_w\}, |R| \leq t$, the group manager distributes the shares of t-degree polynomial $f(x)$ to non-revoked group members via the following broadcast message:*
$\mathcal{B} = \{R\} \cup \{w(x) = g(x)f(x) + h(x)\}$, *where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2)...(x - r_w)$.*
3. Personal key recovery: *If any non-revoked group member U_i receives such a broadcast message, it evaluates the polynomial $w(x)$ at point i and gets $w(i) = g(i)f(i) + h(i)$. Because U_i knows $h(i)$ and $g(i) \neq 0$, it can compute the new personal key $f(i) = \frac{w(i)-h(i)}{g(i)}$.*

In Scheme 1, each non-revoked group member U_i can only recover its own personal share $f(i)$, since computing the personal key of another non-revoked member U_j requires the knowledge of the personal secret $\{h(j)\}$. The coalition of no more than t revoked members has no way to determine any share on $f(x)$, because no matter what $f(x)$ is, for any revoked group member $U_{i'}$, we have $h(i') = w(i')$, which implies that any $f(x)$ is possible from the knowledge of the coalition of the revoked group members.

It is noted that the degrees of $g(x)$, $f(x)$ and $h(x)$ are w, t and $2t$, respectively. If $w < t$, after the broadcast of $w(x)$, we actually disclose $h_{2t}, h_{2t-1}, ..., h_{t+w+1}$ to anybody who receives the broadcast message. Fortunately, this information disclosure does not give the coalition of no more than t revoked members any information that they are not entitled to. This is guaranteed by Theorem 1. In fact, $t + w$ degree is enough for the masking polynomial $h(x)$. However, at the setup stage, the group manager does not know the exact number of revoked group members in a particular session. Thus, a practical way to address this problem is to choose the degree of $h(x)$ as $2t$.

THEOREM 1. *Scheme 1 is an unconditionally secure personal key distribution scheme with t-revocation capability.*

In the setup stage, each group member U_i needs to store its ID i and one share of the masking polynomial $h(i)$. Thus, the storage requirement in each group member is $O(\log q)$. The broadcast message consists of a set of no more than t IDs and one $2t$ degree polynomial. Thus, the communication overhead for Scheme 1 is $O(t \log q)$. This is a significant improvement over the scheme in [27], in which the communication complexity is $O(t^2 \log q)$.

3.2 Self-Healing Key Distribution with Revocation Capability

The technique in Scheme 1 is an efficient scheme to distribute personal key shares to select group members. Here we further extend it to enable the group manager to distribute group session keys to select group members, at the same time allowing group members to recover lost session keys for previous sessions. This technique combines the technique in Scheme 1 with the self-healing method in [27].

Intuitively, the group manager randomly splits each group session key K_j into two t-degree polynomials, $p_j(x)$ and $q_j(x)$, such that $K_j = p_j(x) + q_j(x)$. The group manager then distributes shares $p_j(i)$ and $q_j(i)$ to each select group member U_i (via broadcast). This allows a group member that has both $p_j(i)$ and $q_j(i)$ to recover K_j by $K_j = p_j(i) + q_j(i)$. Thus, assuming there are m sessions, we can build $(m + 1)$ broadcast polynomials in session j to distribute the shares of $\{p_1(x), ..., p_j(x), q_j(x), ..., q_m(x)\}$ to all select group members. If a valid U_i receives the broadcast message, it can recover $\{p_1(i), ..., p_j(i), q_j(i), ..., q_m(i)\}$ and compute session key $K_j = p_j(i) + q_j(i)$. But the revoked group members get nothing about the corresponding keys from this broadcast message. Furthermore, if a select group member U_i receives session key distribution messages in sessions j_1 and j_2, where $j_1 < j_2$, but not the session key distribution message for session j, where $j_1 < j < j_2$, it can still recover the lost session key K_j by first recovering $p_j(i)$ and $q_j(i)$ from the broadcast messages in sessions j_2 and j_1, respectively, and then computing $K_j = p_j(i) + q_j(i)$.

SCHEME 2. *Self-healing session key distribution scheme with t-revocation capability.*

1. Setup*: The group manager randomly picks $m \cdot (m+1)$ $2t$-degree masking polynomials from $F_q[x]$, which are denoted as $\{h_{i,j}(x)\}_{i=1,...,m,j=1,...,m+1}$. Each U_v gets its personal secret, $S_v = \{h_{i,j}(v)\}_{i=1,...,m,j=1,...,m+1}$, from the group manager via the secure communication channel between them. The group manager also picks m random session keys, $\{K_i\}_{i=1,...,m} \subset F_q$ and m random t-degree polynomials $p_1(x), ..., p_m(x)$ from $F_q[x]$. For each $p_i(x)$, the group manager constructs $q_i(x) = K_i - p_i(x)$.*

2. Broadcast*: In the j^{th} session key distribution, given a set of revoked member IDs, $R_j = \{r_1, r_2, ..., r_{w_j}\}$, $|R_j| = w_j \leq t$, the group manager broadcasts the following message:*

$$\begin{aligned}\mathcal{B}_j = &\{R_j\}\\ &\cup \{\mathcal{P}_{j,i}(x) = g_j(x)p_i(x) + h_{j,i}(x)\}_{i=1,...,j}\\ &\cup \{\mathcal{Q}_{j,i}(x) = g_j(x)q_i(x) + h_{j,i+1}(x)\}_{i=j,...,m},\end{aligned}$$

where $g_j(x) = (x - r_1)(x - r_2)...(x - r_{w_j})$.

3. Session key and shares recovery*: When a non-revoked group member U_v receives the j^{th} session key distribution message, it evaluates the polynomials $\{\mathcal{P}_{j,i}(x)\}_{i=1,...,j}$ and $\{\mathcal{Q}_{j,i}(x)\}_{i=j,...,m}$ at point v, recovers the shares $\{p_1(v), ..., p_j(v)\}$ and $\{q_j(v), ..., q_m(v)\}$, and computes the current session key by $K_j = p_j(v) + q_j(v)$. Then it stores all the items in $\{p_1(v), ..., p_{j-1}(v), K_j, q_{j+1}(v), ..., q_m(v)\}$ that it doesn't have.*

4. Add group members*: When the group manager wants to add a member starting from session j, it picks an unused ID $v \in F_q$, computes all $\{h_{i,k}(v)\}_{i=j,...,m,k=j,...,m+1}$, and gives $\{v, \{h_{i,k}(v)\}_{i=j,...,m,k=j,...,m+1}\}$ to this group member via the secure communication channel between them.*

A requirement of Scheme 2 is that the sets of revoked group members must change monotonically. That is, $R_{j_1} \subseteq R_{j_2}$ for $1 \leq j_1 \leq j_2 \leq m$. Otherwise, a group member that is revoked in session j and rejoins the group in a later session can recover the key for session j, due to the self-healing capability of Scheme 2. This requirement also applies to the later schemes. Scheme 2 has the properties of unconditional security, self-healing, t-revocation capability, t-wise forward secrecy and t-wise backward secrecy, as shown in Theorems 2 and 3.

THEOREM 2. *Scheme 2 is an unconditionally secure, self-healing session key distribution scheme with $m \log q$-bit privacy and t-revocation capability.*

PROOF. We need to prove that Scheme 2 satisfies all the conditions listed in Definition 2.

1. (a) Session key recovery is described in step 3 of Scheme 2. Thus, $H(K_j|\mathcal{B}_j, S_i) = H(K_j|z_{i,j}) = 0$.

   (b) For any $B \subseteq \{U_1, ..., U_n\}$, $|B| \leq t$, and any non-revoked member $U_v \notin B$, we show that the coalition of B knows nothing about S_v. First, we have $\{h_{j,i}(v) = \mathcal{P}_{j,i}(v) - g_j(v)p_i(v)\}_{i \leq j}$, $\{h_{j,i+1}(v) = \mathcal{Q}_{j,i}(v) - g_j(v)q_i(v)\}_{i \geq j}$, $\{p_i(v) + q_i(v) = K_i\}_{i=1,...,m}$. Since all $P_{j,i}(v)$, $Q_{j,i}(v)$, K_i and $g_j(v)$ are known values after the broadcast of all $\{\mathcal{B}_1, ..., \mathcal{B}_m\}$, we have

$$\begin{aligned}&H(S_v|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m)\\ &= H(\{h_{j,i}(v)\}_{j=1,...,m,i=1,...,m+1}|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m)\\ &= H(\{p_i(v), q_i(v)\}_{i=1,...,m}|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m)\\ &= H(\{p_i(v)\}_{i=1,...,m}|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m)\end{aligned}$$

   Second, we randomly pick all $\{p'_i(v)\}_{i=1,...,m}$. Because the coalition of B knows at most t points on each $\{p_i(x)\}_{i=1,...,m}$, we can construct $\{p'_i(x)\}_{i=1,...,m}$ based on Lagrange interpolation on these points. Thus, we construct $\{q'_i(x) = K_i - p'_i(x)\}_{i=1,...,m}$, $\{h'_{j,i}(x) = \mathcal{P}_{j,i}(x) - g_j(x)p'_i(x)\}_{i \leq j}$ and $\{h'_{j,i+1}(x) = \mathcal{Q}_{j,i}(x) - g_j(x)q'_i(x)\}_{i \geq j}$. We can easily verify that the following constraints, which are all the knowledge that the coalition of B knows.

   (i) $\{p'_i(x) + q'_i(x) = K_i\}_{i=1,...,m}$

   (ii) $\{g_j(x)p'_i(x) + h'_{j,i}(x) = \mathcal{P}_{j,i}(x)\}_{i \leq j}$

   (iii) $\{g_j(x)q'_i(x) + h'_{j,i+1}(x) = \mathcal{Q}_{j,i}(x)\}_{i \geq j}$

   (iv) $\forall U_{i'} \in B$, $\{h'_{j,i}(i') = h_{j,i}(i')\}_{j=1,...,m,i=1,...,m+1}$.

   Since $\{p'_i(v)\}_{i=1,...,m}$ are picked randomly, we have

$$\begin{aligned}&H(\{p_i(v)\}_{i=1,...,m}|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m)\\ &= H(\{p_i(v)\}_{i=1,...,m}).\end{aligned}$$

   Thus, $H(S_v|\{S_{i'}\}_{U_{i'} \in B}, \mathcal{B}_1, ..., \mathcal{B}_m) = H(\{p_i(v)\}_{i=1,...,m}) = m \log q$.

   (c) Since $\{p_i(x)\}_{i=1,...,m}$ and $\{h_{j,i}(x)\}_{1 \leq i \leq m, 1 \leq j \leq m+1}$ are all randomly picked, $z_{i,j} = \{p_1(i), ..., p_j(i), q_j(i), ..., q_m(i)\}$ cannot be determined only by broadcast messages or personal keys. It follows that $H(z_{i,j}|\mathcal{B}_1, ..., \mathcal{B}_m) = H(z_{i,j}) = H(z_{i,j}|S_1, ..., S_n)$.

2. Assume a collection R of t revoked group members collude. The coalition of R knows at most t points on $q_j(x)$ and nothing on $p_j(x)$ before the broadcast of $\mathcal{B}_j$. Based on Lagrange interpolation, we randomly construct a polynomial $q'_j(x)$ from these t points. Then we randomly pick K'_j, and construct $p'_j(x) = K'_j - q'_j(x)$ and $h'_{j,j}(x) = \mathcal{P}_{j,j}(x) - g_j(x)p'_j(x)$. After the broadcast of $\mathcal{B}_j$, we can verify that $g_j(x)p'_j(x) + h'_{j,j}(x) = \mathcal{P}_{j,j}(x)$. Moreover, for any $U_{i'} \in R$, $q'_j(i') = q_j(i')$ (from the construction of $q'_j(x)$). Since $g_j(i') = 0$, $h'_{j,j}(i') = \mathcal{P}_{j,j}(i') - g_j(i')p'_j(x) = \mathcal{P}_{j,j}(i') = h_{j,j}(i')$. In addition, since K'_j is randomly chosen, any value is possible from what the coalition knows about K_j. Thus, $H(K_j|\mathcal{B}_1, ..., \mathcal{B}_j, \{S_{i'}\}_{U_{i'} \in R}) = H(K_j)$.

3. (a) From step 3 of Scheme 2, for any U_i that is a member in sessions j_1 and j_2 ($1 \leq j_1 < j < j_2 \leq m$), U_i can recover $\{p_1(i), ..., p_{j_1}(i), q_{j_1}(i), ..., q_j(i), ..., q_m(i)\}$ and $\{p_1(i), ..., p_j(i), ..., p_{j_2}(i), q_{j_2}(i), ..., q_m(i)\}$, and recover K_j by computing $K_j = p_j(i) + q_j(i)$. Thus, $H(K_j|z_{i,j_1}, z_{i,j_2}) = 0$.

   (b) For any disjoint subsets $B, C \subset \{U_1, ..., U_n\}$, where $|B \cup C| \leq t$ and $1 \leq j_1 < j < j_2 \leq m$, $\{z_{i',j}\}_{U_{i'} \in B, 1 \leq j \leq j_1}$

contains $\{q_j(i)\}_{U_i \in B}$, and the set $\{z_{i',j}\}_{U_{i'} \in C, m \geq j \geq j_2}$ contains $\{p_j(i)\}_{U_i \in C}$. Thus, for session j, the coalition $B \cup C$ knows at most $|B|$ points on $q_j(x)$ and $|C|$ points on $p_j(x)$. Because $p_j(x), q_j(x)$ are two t-degree polynomials and $|B \cup C| \leq t$, the coalition of $B \cup C$ cannot recover K_j. That is, $H(K_j|\{z_{i',j}\}_{U_{i'} \in B, 1 \leq j \leq j_1} \cup \{z_{i',j}\}_{U_{i'} \in C, m \geq j \geq j_2}) = H(K_j)$. □

THEOREM 3. *Scheme 2 has the properties of t-wise forward secrecy and t-wise backward secrecy.*

The storage requirement in Scheme 2 comes from two parts. First, at the setup step, each group member is required to store the personal secret, which occupies $m(m+1)\log q$ memory space. (Note that the group members that join later need to store less data.) Second, after receiving the session key distribution message in session j, each group member U_v need store the session key K_j and $\{q'_j(v)\}_{j' \in \{j+1,...,m\}}$. The latter is necessary to recover future lost session keys. This takes at most $m \log q$ memory space. Hence, the total storage overhead in each group member is at most $m(m+2)\log q$.

The broadcast message in step 2 consists of the set of IDs of all revoked group members and $(m+1)$ $2t$-degree polynomials. Since we only require the uniqueness of the ID of a particular group member, the member IDs can be picked from a much smaller finite set than F_q. Further considering that the number of revoked IDs will never be greater than t, we can omit the overhead for storing or broadcasting the revoked member IDs. Thus, the broadcast message size can be simplified to $(m+1)(2t+1)\log q$, which almost reaches the lower bound $max\{t^2 \log q, mt \log q\}$ presented in [27].

3.3 Reducing Storage Requirement

In Scheme 2, the storage overhead in each group member is $O(m^2 \log q)$. The majority of this storage overhead comes from the personal secret that each group member has to keep, which is determined by the number of masking polynomials.

By carefully evaluating the broadcast messages in scheme 2, we note that each $p_i(x)$ is masked by different masking polynomials (i.e., $\{h_{j,i}(x)\}_{j=i,...,m}$) in different sessions. Though having multiple masking polynomials seems to make it more difficult to attack, it does not contribute to the security of this scheme.

Indeed, having one masking polynomial for each $p_i(x)$ is sufficient to protect $p_i(x)$ and its shares in our scheme. In Scheme 2, the purpose of the broadcast polynomial $g_j(x)p_i(x) + h_{j,i}(x)$ is to make sure that all non-revoked members in session j can recover one share on $p_i(x)$, but all revoked members cannot. Consider a given $p_i(x)$. The members who are valid in session i but revoked *after* session i are expected to compute their shares on $p_i(x)$. (Even if such revoked members may lose the broadcast message in session i, they can still recover the corresponding key and shares if they somehow get a copy of that message later.) Therefore, it is unnecessary to protect the same $p_i(x)$ multiple times with different masking polynomials. In other words, once a broadcast polynomial $g_i(x)p_i(x) + h_{i,i}(x)$ is constructed in session i, the group manager may reuse it for the remaining sessions. This implies that we need only one masking polynomial for each $p_i(x)$. As a result, the total number of masking polynomials for $\{p_i(x)\}_{i=1,...,m}$, and thus the number of personal shares that each group member has to keep are both reduced.

Similarly, the number of masking polynomials for each $q_i(x)$ can also be reduced. First, in Scheme 2, the members that join in or before session i are expected to compute all their shares on $q_i(x), ..., q_m(x)$. Thus, we can reuse the masking polynomials as discussed earlier. Second, it is easier to prevent later added group members from accessing shares of earlier $q_i(x)$, since the group manager already knows which group members to deal with. In particular, the group manager doesn't need to use any revoking polynomial, but just need to keep the shares of the masking polynomials for $\{p_i(x)\}_{i=1,...,j}$ away from the group members added after session j. Thus, the broadcast polynomial in Scheme 2, $\{g_j(x)q_i(x) + h_{j,i+1}(x)\}_{i=j,...,m}$, can be replaced with $\{q_i(x) + f_i(x)\}_{i=j,...,m}$, where $\{f_i(x)\}_{i=j,...,m}\}$ is a set of random t-degree polynomials.

Based on the above discussion, we propose Scheme 3 to reduce the storage requirement in each member from $O(m^2 \log q)$ in Scheme 2 to $O(m \log q)$.

SCHEME 3. *Improved self-healing session key distribution scheme with t-revocation capability.*

1. Setup*: The group manager randomly picks m $2t$-degree masking polynomials, $\{h_i(x)\}_{i=1,...,m}$, and m t-degree polynomials, $\{f_i(x)\}_{i=1,...,m}$, from $F_q[x]$. Each U_v gets its personal secret, $S_v = \{h_i(v), f_i(v)\}_{i=1,...,m}$, from the group manager via the secure communication channel. The group manager also picks m random session keys, $\{K_i\}_{i=1,...,m} \subset F_q$ and m random t-degree polynomials $p_1(x), ..., p_m(x)$ from $F_q[x]$. For each $p_i(x)$, the group manager constructs $q_i(x) = K_i - p_i(x)$.*

2. Broadcast*: In the j^{th} session key distribution, given the sets of revoked member IDs for sessions in and before session j, $R_i = \{r_1, r_2, ..., r_{w_i}\}_{i=1,...,j}$, where $|R_i| = w_i \leq t$ for $i = 1, ..., j$, the group manager broadcasts the following message:*

$$\mathcal{B}_j = \{R_i\}_{i=1,...,j} \cup \{\mathcal{P}_i(x) = g_i(x)p_i(x) + h_i(x)\}_{i=1,...,j} \cup \{\mathcal{Q}_i(x) = q_i(x) + f_i(x)\}_{i=j,...,m},$$

where $g_i(x) = (x - r_1)(x - r_2)...(x - r_{w_i}), 1 \leq i \leq j$.

3. Session key and shares recovery*: When a non-revoked group member U_v receives the j^{th} session key distribution message, it evaluates $\{\mathcal{P}_i(x)\}_{i=1,...,j}$ and $\{\mathcal{Q}_i(x)\}_{i=j,...,m}$ at point v, recovers the shares $\{p_1(v), ..., p_j(v)\}$ as well as $\{q_j(v), ..., q_m(v)\}$, and then computes the current session key $K_j = p_j(v) + q_j(v)$. It finally stores the items in $\{p_1(v), ..., p_{j-1}(v), K_j, q_{j+1}(v), ..., q_m(v)\}$ that it does not have.*

4. Add group members*: When the group manager adds a group member starting from session j, it picks an unused ID $v \in F_q$, computes all $\{h_i(v)\}_{i=j,...,m}$ and $\{f_i(v)\}_{i=j,...,m}$, and gives $\{v, \{h_i(v)\}_{i=j,...,m}, \{f_i(v)\}_{i=j,...,m}\}$ to this group member via the secure communication channel between them.*

Though Scheme 3 requires less storage than Scheme 2, it still retains the nice security properties such as unconditional security and t-wise forward and backward secrecy, as shown in Theorems 4 and 5.

THEOREM 4. *Scheme 3 is an unconditionally secure, self-healing session key distribution scheme with $m \log q$-bit privacy and t-revocation capability.*

THEOREM 5. *Scheme 3 has the properties of t-wise forward secrecy and t-wise backward secrecy.*

During the setup stage, each group member needs to store one share of each of the masking polynomials, which totally occupy $2m \log q$ space. Moreover, in order to recover from message loss, each member needs to store one share (out of the two shares) of each session key, or the session key itself if it has both shares, which totally require $m \log q$ space. Hence, the overall storage overhead in each member is at most $3m \log q$, which is much less than $m(m+2)\log q$ in Scheme 2.

The broadcast message in session j consists of j revocation sets $\{R_i\}_{i=1,...,j}$ and $m+1$ polynomials. Since $R_1 \subseteq R_2 \subseteq, ..., \subseteq R_m$ and $|R_m| \leq t$, we can use a one-dimensional array with j elements to indicate the number of revoked members in each session. In other words, we can represent all $\{R_i\}_{i=1,...,j}$ by R_j and this array. In addition, the member IDs can be picked from a small finite field. Therefore, we can ignore the communication overhead for the broadcast of all those revocation sets here. Thus, the broadcast size in session j is $((m+j+1)t+m+1)\log q$, which is a little smaller than that in Scheme 2. The reason is that the degree of polynomials $\{\mathcal{Q}_j(x)\}_{j=1,...,m}$ is reduced from $2t$ to t. The largest broadcast size (when $j=m$) is $((2m+1)t+m+1)\log q$.

As we discussed earlier, in Scheme 3, if a revoked group member doesn't receive a broadcast message before it is revoked, it may recover the corresponding session key by receiving broadcast messages after it is revoked. This doesn't introduce security problem, since the revoked member is entitled to that information. However, such a revoked member cannot do the same thing in Scheme 2 unless it gets the lost broadcast message, because different masking polynomials are used in different sessions. This is the difference between Scheme 2 and Scheme 3.

3.4 Trading Off Self-healing Capability for Less Broadcast size

In our previous schemes, each key distribution message contains redundant information for all the other $m-1$ sessions. However, in certain situations, having redundant information for all the sessions may be unnecessary and consume too much bandwidth. For example, when there are only short term communication failures, which are never longer than a fraction of the m sessions, it is only necessary to include redundant information to prepare for the maximum number of such sessions. As another example, when there are relatively long term but infrequent communication failures, always preparing for such failures may generate more-than-necessary overhead.

In this subsection, we study two possible ways to further reduce the broadcast message size based on the above observation. Our first technique is targeted at possibly frequent but short term communication failures. We assume that after a group member receives a broadcast key distribution message, it takes no more than $l-1$ sessions for it to receive another one, where $l-1 << m$. The basic approach is to introduce a "sliding window"[1] so that only redundant information for the sessions that fall into this window is broadcasted. The key distribution message in each session includes the recovery information on the current session key and shares of the previous and the future $l-1$ session keys. The valid member can recover any lost key in the sessions between two successfully received key distribution messages.

Obviously, with the "sliding window" technique, we cannot ensure the same self-healing property as in our previous schemes. In the following, we extend the notion of self-healing to l-session self-healing to clarify the capability of the new scheme.

DEFINITION 4. *(l-session self-healing) Let $t, i \in \{1, ..., n\}$ and $j, l \in \{1, ..., m\}$. $\mathcal{D}$ is l-session* self-healing *if*

(a) *for any session j, where $max(j-l+1, 1) \leq j_1 < j < j_2 \leq min(j+l-1, m)$, and any U_i who is a member in sessions j_1 and j_2, K_j is determined by the set, $\{z_{i,j_1}, z_{i,j_2}\}$ (i.e., $H(K_j|z_{i,j_1}, z_{i,j_2}) = 0$), and*

(b) *for any session j, where $1 \leq j_1 < j < j_2 \leq m$, and any disjoint subsets $B, C \subset \{U_1, ..., U_n\}$ where $|B \cup C| \leq t$, the set $\{z_{i',j}\}_{U_{i'} \in B, 1 \leq j \leq j_1} \cup \{z_{i',j}\}_{U_{i'} \in C, m \geq j \geq j_2}$ contains no information on K_j (i.e., $H(K_j|\{z_{i',j}\}_{U_{i'} \in B,\ 1 \leq j \leq j_1} \cup \{z_{i',j}\}_{U_{i'} \in C,\ m \geq j \geq j_2}) = H(K_j)$).*

Based on the above discussion, we develop the following scheme to trade off self-healing capability with broadcast size.

SCHEME 4. *Session key distribution with t-revocation capability for short term communication failures. The* setup *and* adding group members *steps are the same as Scheme 3.*

- Broadcast*: In the j^{th} session key distribution, given the sets of revoked member IDs for sessions in and before session j, $R_i = \{r_1, r_2, ..., r_{w_i}\}_{i=max(j-l+1,1),...,j}$, where $|R_i| = w_i \leq t$ for $i = max(j-l+1, 1), ..., j$, the group manager broadcasts the following message:*

$$\mathcal{B}_j = \{R_i\}_{i=max(j-l+1,1),...,j} \cup \{\mathcal{P}_i(x) = g_i(x)p_i(x) + h_i(x)\}_{i=max(j-l+1,1),...,j} \cup \{\mathcal{Q}_i(x) = q_i(x) + f_i(x)\}_{i=j,...,min(j+l-1,m)}$$

where $g_i(x) = (x-r_1)(x-r_2)...(x-r_{w_i}), max(j-l+1, 1) \leq i \leq j$.

- Session key and shares recovery*: When a non-revoked group member U_v receives the j^{th} key distribution message, it first evaluates the polynomials $\{\mathcal{P}_i(x)\}_{i=max(j-l+1,1),...,j}$ and $\{\mathcal{Q}_i(x)\}_{i=j,...,min(j+l-1,m)}$ at point v, then recovers the shares $\{p_{max(j-l+1,1)}(v), ..., p_j(v)\}$ as well as $\{q_j(v), ..., q_{min(j+l-1,m)}(v)\}$, and computes the current session key $K_j = p_j(v) + q_j(v)$. Finally, the member U_v stores the items in $\{p_{max(j-l+1,1)}(v), ..., p_{j-1}(v), K_j, q_{j+1}(v), ..., q_{min(j+l-1,m)}(v)\}$ that it does not have.*

THEOREM 6. *Scheme 4 is an unconditionally secure, l-session self-healing session key distribution scheme with $m \log q$-bit privacy and t-revocation capability, t-wise forward and backward secrecy.*

In Scheme 4, the size of personal secret in each member is at most $2m \log q$. In addition, it needs additional $(2l-1)\log q$ memory space to store the session key and shares. Therefore, the total storage overhead is at most $(2m+2l-1)\log q$. The broadcast message consists of l $2t$-degree polynomials and l t-degree polynomials, which occupies $l(3t+2)\log q$ in the communication bandwidth.

Our second technique is aimed at situations where there are relatively long term but infrequent communication failures. Specifically, we assume that each group member can receive at least d consecutive broadcast key distribution messages, and after a group member receives a broadcast key distribution message, it takes no more than $(l-1)d$ sessions for it to receive another one.

Intuitively, the second technique is to selectively include the same amount of redundant information from a large "window" of sessions (i.e., $2(l-1)d+1$ instead of $2l-1$ sessions) in each key distribution message. Specifically, the group manager picks one from every d consecutive sessions in a particular window of sessions and includes key shares for those selected sessions in the key

[1] The term "sliding window" was also mentioned in [27]. However, no specific technique has been presented there.

distribution message. In other words, the recovery information for a particular session key is evenly distributed among a large number of sessions. Given the window size $2(l-1)d+1$, the key distribution message for session j will contain key shares for sessions $j-(l-1)d, j-(l-2)d, ..., j-d$ and $j+d, j+2d, ..., j+(l-1)d$. Thus, any d consecutive session key distribution messages contain shares of the previous and the future $(l-1)d$ sessions. A group member may not find the necessary information to recover a particular session key in one key distribution message; however, it is guaranteed to find one in the next $d-1$ key distribution messages. In general, this idea is to trade off the key recovery delay with the number of recoverable sessions.

Scheme 4 can be viewed as a special case of this technique (when $d=1$). To clarify the self-healing capability of this new technique, we generalize Definition 4 into the following notion of (l,d) self-healing.

DEFINITION 5. *((l,d) self-healing) Let* $t, i \in \{1, ..., n\}$ *and* $j, l, d \in \{1, ..., m\}$. $\mathcal{D}$ *is* (l,d) self-healing *if*

(a) for any session j, where $max(j-(l-1)\cdot d, 1) \le j - j_1 \cdot d < j < j + j_2 \cdot d \le min(j+(l-1)\cdot d, m)$, and any U_i who is a member in sessions $j - j_1 \cdot d$ and $j + j_2 \cdot d$, K_j is determined by the set, $\{z_{i,j-j_1\cdot d}, z_{i,j+j_2\cdot d}\}$ (i.e., $H(K_j|z_{i,j-j_1\cdot d}, z_{i,j+j_2\cdot d}) = 0$), and

(b) for any session j, where $1 \le j_1 < j < j_2 \le m$, and any disjoint subsets $B, C \subset \{U_1, ..., U_n\}$ where $|B \cup C| \le t$, the set $\{z_{i',j}\}_{U_{i'} \in B, 1 \le j \le j_1} \cup \{z_{i',j}\}_{U_{i'} \in C, m \ge j \ge j_2}$ contains no information on K_j (i.e., $H(K_j|\{z_{i',j}\}_{U_{i'} \in B,\ 1 \le j \le j_1} \cup \{z_{i',j}\}_{U_{i'} \in C,\ m \ge j \ge j_2}) = H(K_j)$).

The scheme built on the above idea is a natural generalization of Scheme 4.

SCHEME 5. *Session key distribution with t-revocation capability for long term but infrequent communication failures. The* setup *and* adding group members *steps are the same as in Scheme 3.*

- Broadcast*: Let $G_j^p = \{j - i \cdot d\}_{0 \le i < min(j/d, l)}$, and $G_j^q = \{j + i \cdot d\}_{0 \le i < min((m-j)/d, l)}$. In the j^{th} session key distribution, given the sets of revoked member IDs for sessions in and before session j, $R_i = \{r_1, r_2, ..., r_{w_i}\}_{i \in G_j^p}$, where $|R_i| = w_i \le t$ for $i \in G_j^p$, the group manager broadcasts the following message:*

$$\mathcal{B}_j = \{R_i\}_{i \in G_j^p} \cup \{\mathcal{P}_i(x) = g_i(x)p_i(x) + h_i(x)\}_{i \in G_j^p} \cup \{\mathcal{Q}_i(x) = q_i(x) + f_i(x)\}_{i \in G_j^q}$$

  where $g_i(x) = (x - r_1)(x - r_2)...(x - r_{w_i})$, $i \in G_j^p$.

- Session key and shares recovery*: When a non-revoked group member U_v receives the j^{th} session key distribution message, it evaluates $\{\mathcal{P}_i(x)\}_{i \in G_j^p}$ and $\{\mathcal{Q}_i(x)\}_{i \in G_j^q}$ at point v, recovers the shares $\{p_i(v)\}_{i \in G_j^p}$ and $\{q_i(v)\}_{i \in G_j^q}$, and then computes the current session key $K_j = p_j(v) + q_j(v)$. It finally stores the items in $\{p_i(v)\}_{i \in G_j^p}$ and $\{q_i(v)\}_{i \in G_j^q}$ that it does not have.*

THEOREM 7. *Scheme 5 is an unconditionally secure, (l,d) self-healing session key distribution scheme with $m \log q$-bit privacy and t-revocation capability, t-wise forward and backward secrecy.*

From the broadcast step in Scheme 5, it is obvious that the communication overhead of this generalized scheme is the same as Scheme 4. Since the group member needs to buffer the key and shares of $2(l-1)d+1$ consecutive sessions, the total storage overhead is $(2m + 2(l-1)d + 1)\log q$.

Generally, the above two extensions (Scheme 4 and Scheme 5) allow small key distribution messages, which are independent of the total number of sessions. The choice of window size depends mainly on the network environment. Thus, it is possible to have a large number of sessions and still have a reasonable broadcast message size and self-healing capability. Nevertheless, the storage overhead in each member still limits the total number of sessions.

A special case of these two scheme is to let $m = t$, and have the group manager update session keys if and only if at least one compromised member is detected. On the one hand, it is possible to cover a long network lifetime. On the other hand, the compromised member can be revoked immediately. This customization may be suitable for the applications that cannot afford a large number of sessions, but still want to cover a long period of time.

3.5 Comparison with Previous Self-Healing Methods

In this subsection, we give a simple comparison between Scheme 3 and Constructions 3 and 4 presented in [27]. Since Schemes 4 and 5 are mainly about trade offs between self-healing capability and broadcast message size, we do not include them here. Note that the technique used in the long-lived construction (Construction 5) in [27] is also applicable to our schemes. Thus, we do not consider it here either.

Table 1 summarizes the comparison between these three self-healing key distribution methods. We use C_3 to denote Construction 3 in [27], which is the basic unconditionally secure self-healing scheme with t-revocation capability, and C_4 to denote Construction 4 in [27], which is the less broadcast size variant of C_3. Note that C_4 reduces the broadcast size by sacrificing the unconditional security property of C_3 (for computational security). In contrast, Scheme 3 proposed in this paper reduces the communication and storage overhead without sacrificing any security property. From Table 1, it is easy to see that our scheme has less communication and storage overhead than both constructions in [27]. Figure 1 further shows the possible values for m and t given a maximum of 64KB packet size[2]. Obviously, our scheme allows more sessions and can deal with more colluding users under the same condition.

4. RELATED WORK

Early approaches to group key management (e.g., Group Key Management Protocol (GKMP) [12]) rely on a group controller, which shares a pairwise key with each group member and distributes group keys to group members on a one-to-one basis. These approaches cannot scale to large groups.

To address the scalability problem, Iolus organizes the multicast group into a hierarchy of subgroups to form a virtual secure multicast group [18]. The group hierarchy can be used for both group communication and distribution of group keys. Wallner et al. [32] and Wong et al. [33] independently discovered the Logical Key Hierarchy (LKH) (or Key Graph) approach. In this approach, individual and auxiliary keys are organized into a hierarchy, where each group member is assigned to a leaf and holds all the keys from its leaf to the root. The root key is shared by all group members and

[2]The values for C_3 and C_4 are slightly larger than those given in Figure 3 in [27]; we compute the values purely from the formula given in Table 1 for the purpose of fair comparison.

Table 1: Comparison between different self-healing key distribution schemes.

	C_3	C_4	Scheme 3
Communication overhead	$(mt^2 + 2mt + m) \log q$	$(3mt + t^2 + 2m + t) \log q$	$(2mt + m + t + 1) \log q$
Storage overhead	$(m^2 + m) \log q$	$(m^2 + m) \log q$	$3m \log q$
Self-healing	Yes	Yes	Yes
Security	unconditional	computational	unconditional
Revocation capability	Yes	Yes	Yes

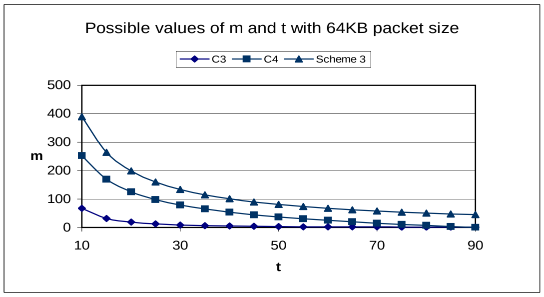

Figure 1: Possible values of m and t for different self-healing key distribution schemes, which are the areas under the corresponding lines. Assume that q is a 64-bit integer. C_4 can only guarantee computational security, while the other two can guarantee unconditional security.

thus used as the group key. A rekey operation in LKH requires $2 \log_2 n$ messages, where n is the number of group members.

A number of techniques have been proposed to improve the LKH approach. Canetti et al. reduce the number of rekey messages to $\log_2 n$ using a pseudo-random generator [7]. Keystone uses Forward Error Correction to reduce message loss, and employs unicast-based re-synchronization to help group members recover lost keys [34]. Periodic (or batch) rekey was proposed to reduce the rekey cost for groups with frequent joins and leaves [15,25,35,36]. Moreover, several issues about scalable and reliable distribution of group keys have been thoroughly studied, including how to determine where to add, delete or update keys in a key tree (for individual or batch rekey) [15, 19, 35, 36] and how to efficiently place encrypted keys in multicast rekey packets [35, 36]. A few other variations of LKH were also proposed, including associating keys with each level in the key hierarchy (instead of each node) [8], combining a-ary LKH+ (i.e., key tree with degree a) with unicast-based rekey to trade-off between communication and storage cost [22], decentralized management of group keys [23], One-way Function Trees (OFT) [1], and ELK which inserts key verification information into data packets to help recover lost group keys [21].

The above methods need at least $O(\log n)$ computation and communication to remove a member. In contrast, MARKS only requires constant computation by distributing seeds of group keys with Binary Hash Tree (BHT) and its variations [6]. However, MARKS only works if the duration that a member stays in the group is known when the member joins the group. In [2], Banerjee and Bhattacharjee proposed to organize group members into different levels of clusters, in which the cluster head can communicate with cluster members via both unicast and multicast. By limiting the size of each cluster and isolating the changes to the related clusters, this approach incurs constant processing, communication and storage overhead for single member joins or leaves, and logarithmic overhead for batch joins and/or leaves [2].

Group key distribution is closely related to broadcast encryption studied in the cryptography community. An overview of early results can be found in [30]. Berkovits presented a way to broadcast a secret to a predetermined set of receivers using secret sharing technique [3]. Fiat and Naor developed broadcast encryption schemes resilient to one bad member, and then proposed approaches to building high resilient schemes from low resilient ones based on Perfect Hash Families (PHF) [10]. Safavi-Naini and Wang applied PHF to construct group rekey schemes directly [24]. Blundo et al. developed a family of one-time broadcast encryption schemes based on the key predistribution scheme in [4], and then extended them to allow interactive group key distribution [5]. Trade off between storage and communication requirements as well as their lower bounds in the proposed schemes are also studied in [5] and [17]. Stinson and van Trung continued the work in [5] and presented new constructions of key predistribution and broadcast encryption schemes [31]. Just et al. studied group key distribution via broadcast encryption and derived a lower bound on the broadcast message size using information theoretic techniques [13]. Kumar et al. proposed two schemes that can revoke up to t group members with storage overhead $O(t \log n)$, and communication overhead $O(t \log n)$ and $O(t^2)$, respectively, where n is the group size [14]. Naor et al. developed a subset-difference based bulk rekey method, which requires $\log^2 n$ keys being stored at members and $2t$ communication overhead [20]. Gong proposed a method to securely broadcast different keys to different group members [11]. However, Gong's method results in a broadcast message linear to the group size, while with our method the size of the broadcast message is linear to the maximum number of colluding users, but independent of the group size.

Our work in this paper is based on the self-healing key distribution approach (with revocation capability) in [27]. The technique in [27] uses secret sharing [26] based on two dimensional polynomials to distribute group keys, enabling group members to recover lost session group keys as long as they have received one broadcast rekey message before and one after the above session. Compared with the approaches discussed earlier, an advantage of both [27] and our techniques is that the computation, communication, and storage overheads required to revoke group members and achieve self-healing capability are independent of the group size, and thus are suitable for very large groups. However, our techniques also improve over those in [27] as discussed in Section 3, and thus are able to deal with coalition of more evicted group members.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented several group key distribution schemes for very large and dynamic groups over unreliable channels. By introducing a novel personal key distribution technique, we developed several efficient unconditionally secure and self-healing group key distribution schemes that significantly improve over the previous approaches. In addition, we developed two techniques that allow trade-offs between the broadcast message size and the recoverability of lost session keys, which can further reduce the broadcast message size in situations where there are frequent but short-term disruptions of communication and where there are long-term but infrequent disruptions of communication, respectively. We have developed an API implementation to facilitate the deployment of the proposed techniques [16].

Our future work includes development of a model that characterizes failures in large and highly mobile wireless networks and further investigation of the performance of the proposed schemes in this model. In addition, we would like to seek more efficient ways to perform the initial key distribution for the proposed schemes.

6. REFERENCES

[1] D. Balenson, D. McGrew, and A. Sherman. Key management for large dynamic groups: One-way function trees and amortized initialization. Internet Draft, draft-balenson-groupkeymgmt-oft-00.txt, February 2000.

[2] S. Banerjee and B. Bhattacharjee. Scalable secure group communication over ip mulitcast. In *Proc. of Int'l Conf. on Network Protocols*, November 2001.

[3] S. Berkovit. How to broadcast a secret. In *EUROCRYPT '91, LNCS 547*, pages 536–541, 1991.

[4] C. Blundo, A. De Santis, Amir Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. In *CRYPTO '92, LNCS 740*, pages 471–486, 1993.

[5] C. Blundo, L. Mattos, and D. R. Stinson. Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution. In *CRYPTO '96, LNCS 1109*, 1996.

[6] B. Briscoe. MARKS: Zero side-effect multicast key management using arbitrarily revealed key sequences. In *Proc. of 1st Int'l Workshop on Networked Group Communication*, 1999.

[7] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. Multicast security: A taxonomy and some efficient constructions. In *IEEE INFOCOMM '99*, 1999.

[8] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, and D. Saha. Key management for secure internet multicast using boolean function minimization techniques. In *IEEE INFOCOMM'99*, volume 2, pages 689–698, 1999.

[9] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991.

[10] A. Fiat and M. Naor. Broadcast encryption. In *CRYPTO '93, LNCS 773*, pages 480–491, 1994.

[11] L. Gong. New protocols for third-party-based authentication and secure broadcast. In *Proc. of the 2nd ACM Conf. on Computer and Communications Security*, 1994.

[12] H. Harney and C. Muckenhirn. Group key management protocol (GKMP) architecture. IETF Request for Comments, RFC 2094, July 1997.

[13] M. Just, E. Kranakis, D. Krizanc, and P. van Oorschot. On key distribution via true broadcasting. In *Proc. of ACM Conf. on Computer and Communications Security*, 1994.

[14] R. Kumar, S. Rajagopalan, and A. Sahai. Coding constructions for blacklisting problems without computational assumptions. In *CRYPTO '99, LNCS 1666*, pages 609–623, 1999.

[15] X. S. Li, Y. R. Yang, M. Gouda, and S. S. Lam. Batch rekeying for secure group communications. In *Proc. of 10th Int'l World Wide Web Conf.*, May 2001.

[16] D. Liu, P. Ning, and K. Sun. Efficient and self-healing group key distribution with revocation capability. Technical Report TR-2003-03, Department of Computer Science, North Carolina State University, 2003.

[17] M. Luby and J. Staddon. Combinatorial bounds for broadcast encryption. In *EUROCRYPT '98, LNCS 1403*, 1998.

[18] S. Mittra. Iolus: A framework for scalable secure multicasting. In *ACM SIGCOMM '97*, pages 277–288, 1997.

[19] M. J. Moyer, J. R. Rao, and P. Rohatgi. Maintaining balanced key trees for secure multicast. Internet Draft, draft-irtf-smug-key-tree-balance-00.txt, June 1999.

[20] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *CRYPTO '01, LNCS 2139*, pages 41–62, 2001.

[21] A. Perrig, D. Song, and J.D. Tygar. ELK, a new protocol for efficient large-group key distribution. In *Proc. of IEEE Symp. on Security and Privacy*, pages 247–262, 2001.

[22] T. Malkin R. Canetti and K. Nissim. Efficient communications-storage tradeoffs for multicast encryption. In *EUROCRYPT '99, LNCS 1592*, pages 459–474, 1999.

[23] O. Rodeh, K. Birman, and D. Dolev. Optimized group rekey for group communication systems. In *Proc. of Network and Distributed Systems Security Symp.*, 2000.

[24] R. Safavi-Naini and H. Wang. New constructions of secure multicast re-keying schemes using perfect hash families. In *Proc. of the 7th ACM Conf. on Computer and Communications Security*, pages 228–234, 2000.

[25] S. Setia, S. Koussih, and S. Jajodia. Kronos: A scalable group re-keying approach for secure multicast. In *Proc. of IEEE Symp. on Security and Privacy*, pages 215–228, 2000.

[26] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[27] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean. Self-healing key distribution with revocation. In *Proc. of 2002 IEEE Symp. on Security and Privacy*, pages 224–240, 2002.

[28] M. Steiner, G. Tsudik, and M. Waidner. CLIQUES: A new approach to group key agreement. In *Proc. of the Int'l Conf. on Distributed Computing Systems*, pages 380–387, 1998.

[29] M. Steiner, G. Tsudik, and M. Waidner. Key agreement in dynamic peer groups. *IEEE Trans. on Parallel and Distributed Systems*, 11(8):769–780, August 2000.

[30] D. R. Stinson. On some methods for unconditionally secure key distribution and broadcast encryption. *Designs, Codes and Cryptology*, 12:215–243, 1997.

[31] D. R. Stinson and T. van Trung. Some new results on key distribution patterns and broadcast encryption. *Designs, Codes and Cryptography*, 14:261–279, 1998.

[32] D. Wallner, E. Harder, and R. Agee. Key management for multicast: Issues and architectures. IETF Request For Comments, RFC 2627, June 1999.

[33] C. K. Wong, M. G. Gouda, and S. S. Lam. Secure group communications using key graphs. In *ACM SIGCOMM '98*, pages 68–79, 1998.

[34] C. K. Wong and S. S. Lam. Keystone: A group key management service. In *Int'l Conf. on Telecomm.*, 2000.

[35] Y. R. Yang, X. S. Li, X. B. Zhang, and Simon S. Lam. Reliable group rekeying: A performance analysis. In *ACM SIGCOMM '01*, pages 27–38, 2001.

[36] X. B. Zhang, S. S. Lam, D. Lee, and Y. R. Yang. Protocol design for scalable and reliable group rekeying. In *Proc. SPIE Conf. on Scalability and Traffic Control in IP Networks*, August 2001.