A Self-Healing and Mutual-Healing Key Distribution Scheme Using Bilinear Pairings for Wireless Networks

Biming Tian, Song Han, and Tharam S. Dillon
DEBI Institute Curtin University of Technology, Australia
{biming.tian, song.han, tharam.dillon }@cbs.curtin.edu.au

Abstract

Self-healing key distribution mechanism can be used to establish session keys within a large and dynamic groups of users over an unreliable network. Mutual-healing key distribution mechanism deals with some deficiency existed in self-healing key distribution mechanism. It is a complementarity to the self-healing mechanism. In this paper, a self-healing key distribution scheme using bilinear pairings is proposed. As far as we know, it is the first self-healing and mutual-healing key distribution scheme based on bilinear pairings. The contributions of this paper are as follows : firstly, the scheme is collusion-free for any coalition of non-authorized users. Secondly, the private key has nothing to do with the number of revoked users and can be reused as long as it is not disclosed. Thirdly, the storage overhead for each user is a constant. In addition, we present technique details on how to realize mutual-healing.

1. Introduction

In recent years, many schemes on distributing session keys for large group communication have been proposed. For large dynamic group communication, membership changes frequently. In order to keep the security of communication, the session key has to be updated on each adding or revoking user. Many existing schemes focused on different key updating mechanism. For example, OFT (One-way Function Tree) [1]-[2] schemes devote to reduce the size of the rekeying message. Broadcast encryption addresses the problem of sending encrypted messages to a large user group so that the encrypted messages can only be decrypted by a dynamic changing privileged subset [3]-[5]. All these literatures supposed that underlying networks are reliable. However, how to distribute session keys for unreliable wireless networks, in a manner that is resistant to packet loss, is an issue that has not been addressed deeply.

Packet loss happens frequently in an unreliable network. The key distribution broadcast for a particular session might never reach some users. A naive solution is requesting retransmission. On the one hand, both requesting and retransmission messages would incur more communication overhead. In some large communication group, such individual interactions place a heavy burden on the group manager. On the other hand, users may reveal their current location by sending messages in some high security environments. Self-healing key distribution schemes enable large and dynamic group users to establish group keys over an unreliable network for secure communication. The main property of the scheme is that, even if at the beginning of certain sessions some broadcast packets get lost, group users are still capable of recovering the session key for those sessions simply by using the broadcasts they have received at a previous session and the packets they will receive at a subsequent one. In this scheme, users do not need to send any requesting message to the group manager and do not need to update their personal keys. This noninteractive key distribution scheme reduces the network traffic, decreases the work load on the group manager, and lowers the risk of user exposure through traffic analysis. Therefore, self-healing key distribution schemes are desirable for both efficiency and security reasons.

Group communications over low-cost channels in different fields can benefit from self-healing key distribution mechanism, especially for those settings in which session keys need to be used for a short time-period, due to frequent adding or deleting users. For example, military-oriented applications as well as Internet application [7], such as broadcast transmissions, pay-per-view TV. In addition, self-healing key distribution scheme may be useful in commercial content distribution applications or electronic services in which the contents are highly sensitive.

In this paper, we will propose a self-healing key distribution scheme using bilinear pairings. The main contribution of this paper is highlighted by following properties:

- It is the first time to deal with self-healing key distribution scheme using bilinear pairings-based cryptology.
- The scheme is collusion-free for any coalition of non-

978-0-7695-3492-3/08 $25.00 © 2008 IEEE
DOI 10.1109/EUC.2008.36

authorized users.

- The private key has nothing to do with the number of revoked users and can be reused as long as it is not disclosed.
- The storage overhead for user is a constant.
- It is the first time to realize mutual-healing technique.

The rest of the paper is organized as follows: In Section 2, we present an overview of earlier works in the area of self-healing key distribution. In Section 3, we briefly introduce the preliminaries to be used in the design of self-healing key distribution protocol. In Section 4, we give system parameters firstly and present a concrete construction secondly. In Section 5, we analysis the security of the proposed scheme and make an efficiency comparison with previous protocols. We explore the technique to realize the mutual-healing mechanism in Section 6 and conclude the paper in the last section.

2. Related Works

2.1. Self-healing key distribution schemes

The first Self-healing key distribution with revocation scheme was introduced by Staddon et al. in [7]. They presented formal definitions, lower bounds on the resources as well as some constructions. However, the constructions given in this paper suffer from high storage overhead and communication overhead. Since then, self-healing key distribution has been one of the hot research topics. Liu et al. generalized the definitions in [7] and gave some constructions in [8]. The scheme reduces communication overhead and storage overhead by introducing a novel personal key distribution technique. Furthermore, they developed two techniques that allow trade-off between the broadcast message size and the recoverability of lost session keys.

Blundo et al. in [9] showed an attack that can be applied to the first construction in [7], developed a new mechanism under a slightly modified framework, and proposed another key-recovery scheme which enables an authorized user to recover all lost session keys by using only the current broadcast message. More et al. in [10] used a sliding window to address three problems in [7]. The three problems were inconsistent robustness, high overhead and expensive maintenance costs. Dutta et al. developed a new self-healing key distribution scheme in [13]. The scheme has significant improvement in terms of both storage overhead and communication overhead.

Sáez in [11] and [12] considered applying vector space secret sharing instead of Shamir's secret sharing schemes to design self-healing key distribution scheme. He made use of general monotone decreasing structures for the family of subsets of users that can be revoked instead of a threshold one. Recently, Jiang et al proposed an efficient self-healing group key scheme with time-limited node revocation based on DDHC (Dual Directional Hash Chains) in [15]. The performance of the proposed scheme was evaluated by both theory analysis and experiment data. The results showed that the scheme made a good balance between performance and security.

In general, three cryptographic primitives were used to design self-healing key distribution schemes. They are polynomial secret sharing, vector space secret sharing, and hash function. Polynomial secret sharing is the most common technique used to realize self-healing key distribution. It was used in the pioneering paper [7] and was followed by several subsequent works. However, the maximum number of revoked users is constraint to the degree of the polynomial. Vector space secret sharing based self-healing key distribution schemes consider a monotone decreasing family of revoked subset of users instead of a threshold structure. This general case makes the self-healing scheme more flexible and suitable for practical application. Both forward and backward securities can be guaranteed by DDHC-based self-healing key distribution schemes. However, the feature of resisting collusion of revoked nodes and new joined nodes can not be assured, due to the properties of one-way hash function.

2.2. Mutual-healing

Muhammad et al. in [14] considered incorporating the self-healing feature to SD(Subset Difference) method, which was first proposed by Naor et al. in [6]. Some optimization techniques that can be used to reduce the overhead caused by the self-healing capability were proposed in the paper. At last, the idea of mutual-healing was discussed. One motivation behind mutual-healing was that, if a user has missed more than a fixed number of broadcast messages, it does not have to keep on waiting. Instead it can get assistance from its neighbors. Similarly, if a user misses the last broadcast message, it can not recover the last session key by performing self-healing. For this condition, it can look for assistance from its neighboring users too. This paper only talked about the feasibility of mutual-healing without exploring the technical details.

2.3. Identity-based cryptography and bilinear pairings

In 1984, Shamir [16] introduced the notion of identity-based cryptography to alleviate many of the problems inherent with managing certificates. Since than, many ID-based cryptographic schemes have been proposed using bilinear pairings. In 2001, Boneh and Franklin [17] proposed the first practical identity-based encryption scheme. Inspired by the idea of [17], Du et al. proposed a broadcast encryption scheme for key distribution in [18]. In this paper, we extend the broadcast encryption scheme for key distribution to a self-healing key distribution scheme.

3. Preliminaries

In this section, we briefly describe the bilinear pairing, BDH (Bilinear Diffie-Hellman) assumption and ID-based PKI (Public Key Infrastructure).

3.1. Bilinear pairings and BDH assumption

Let G_1 and G_2 be two cyclic groups of order q for some large prime q. G_1 is a cyclic additive group and G_2 is a cyclic multiplicative group. We assume that the discrete logarithm problems in both G_1 and G_2 are hard. Let $e : G_1 \times G_1 \rightarrow G_2$ be a pairing which satisfies the following conditions:

- Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, for $\forall\ P, Q \in G_1$ and $\forall\ a, b \in \mathbb{Z}_q^*$;
- Non-degeneracy: there exists $P \in G_1$ and $Q \in G_1$, such that $e(P, Q) \neq 1$; That is, for any point $P, Q \in G_1$, $e(P, Q) = 1$ iff $P = O$.
- Computability: there exists an efficient algorithm to compute $e(P, Q)$ for any $P, Q \in G_1$.

BDH PARAMETER GENERATOR. A BDH parameter generator $\mathcal{IG}$ is a probabilistic algorithm that takes a security parameter $0 < k \in \mathbb{Z}$, runs in polynomial time, and output the description of two groups G_1 and G_2 of the same order q and the description of an admissible bilinear map $e : G_1 \times G_1 \rightarrow G_2$.

BDH PROBLEM. Given $\langle P, aP, bP, cP\rangle$ for some $a, b, c \in \mathbb{Z}_q^*$, computes $e(P, P)^{abc} \in G_2$.

BDH ASSUMPTION. There is no polynomial time algorithm to solve the BDH problem.

3.2. ID-based public key infrastructure

DLP(Discrete Logarithm Problem). Given two group elements P and Q, to find an integer $n \in \mathbb{Z}_q^*$, such that $Q = nP$ when such an integer exists.

ID-based public key infrastructure involves a trusted KGC (Key Generation Center)and users. Users' private keys are calculated by KGC and send to the user via a secure channel. The basic operations consist of **Set up** and **Private Key Extraction**. When we use bilinear pairings to construct ID-based private/public keys, the operations can be implemented as follows: KGC runs BDH parameter generator to generate two groups G_1, G_2 and a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$. It chooses an arbitrary generator $P \in G_1$ and defines two cryptographic hash functions: $H_1 : \{0,1\}^* \rightarrow G_1$, $H_2 : G_2 \rightarrow \{0,1\}^*$.

- **Set Up:** KGC chooses a random number $s \in \mathbb{Z}_q^*$ and set $P_{pub} = sP$. Then KGC publishes system parameters $params = \{G_1, G_2, q, P, P_{pub}, H_1, H_2\}$, and keeps s as master-key, which is only known by him.
- **Private Key Extraction:** A user submits its identity to KGC. KGC computes the user's public key $Q_{ID} = H_1(ID)$ and private key $S_{ID} = sQ_{ID}$, then privately returns $S_{ID} = sQ_{ID}$ to the user.

4. The Proposed Scheme

In this section, we introduce system parameters firstly and present a concrete construction secondly.

4.1. System parameters

Let $U = \{u_1, \ldots, u_n\}$ be the finite universe of users. Each user u_i has a unique identifier ID_i. A broadcast unreliable channel is available, and time is defined by a global clock. GM (Group Manager) sets up and manages, by means of joining and revoking operations, a communication group which is a dynamic subset of U. m denotes the number of sessions. Let $G_j \subseteq U$ be the communication group established by the group manager in session j. Each node is preloaded with a public/private key pair (Q_i, S_i) before deployment. The public/private key pair is used to recover the session keys as long as user U_i is not removed by GM from the group. Let $R_j \subseteq G_{j-1}$ denotes the set of revoked group users in session j and $J_j \subset U \setminus G_{j-1}$ denotes the set of users who joins the group in session j with $R_j \cap J_j = \phi$. Hence, $G_j = (G_{j-1} \cup J_j) \setminus R_j$ for $j \geq 2$ and by definition $G_1 = U$. Moreover, for $j \in \{1, \ldots, m\}$, the session key K_j is randomly chosen by GM and according to the uniform distribution. For any non-revoked user $u_i \in G_j$, the j-th session key K_j is determined by the broadcast message B_j and the personal public/private key pair (Q_i, S_i).

4.2. The construction

Different from the previous unconditional security schemes, the self-healing key distribution scheme proposed in this paper is a computationally secure scheme. The self-healing key distribution scheme includes several procedures. We will introduce them one by one.

SETUP. GM obtains both public system parameters and all the public keys of possible users from the ID-based PKI.

GM chooses m session keys $K_1, \cdots, K_m$ from $\mathbb{Z}_q^*$. The session keys are independent to each other and according to the uniform distribution.

BROADCAST. Suppose $|G_j|$ denotes the number of users in session j. For each session $1 \leq j \leq m$, according to the session group G_j, GM computes $Q_{V_1} = \sum_{i=1}^{n} Q_i$ and a $(|G_j|-1) \times |G_j|$ matrix which is defined as follows:

$$\begin{pmatrix} a_2 \\ a_3 \\ \vdots \\ a_{|G_j|} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & 0 & 0 & \dots & 1 \end{pmatrix}$$

Let $a_i^{'}$ represents the transpose of a_i. GM also constructs $|G_j|-1$ auxiliary keys

$$Q_{V_i} = (Q_1, Q_2, \dots, Q_{|G_j|}) \times a_i^{'}, \quad 2 \leq i \leq |G_j|,$$

which means $Q_{V_2} = Q_1 + Q_2$, $Q_{V_3} = Q_1 + Q_3$, $\dots$, $Q_{V_{|G_j|}} = Q_1 + Q_{|G_j|}$. The broadcast message is then formed by computing, for a random $r_j \in \mathbb{Z}_q^*$,

$$U_1 = r_j P, \quad U_i = r_j Q_{V_i} (2 \leq i \leq |G_j|),$$

$$V_j = K_j \oplus H_2(e(P_{pub}, r_j Q_{V_1}))$$

Let $z_j = (U_i(1 \leq i \leq |G_j|), V_j)$. GM broadcasts the ciphertext to the set of users G_j. The ciphertext for the j-th broadcast is in the following form: $B_j = \{z_1, \dots, z_j\}$.

KEY RECOVERY. When a user $u_i \in G_j$ receives the broadcast message B_j, it sets a vector $a_1 = (0, \dots, 0, 1, 0, \dots, 0)$ with $|G_j|$ elements, and only the i-th element is 1. Then A_j is a $|G_j| \times |G_j|$ matrix

$$A_j = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{|G_j|} \end{pmatrix}.$$

The user u_i can solve the following system of equations using Cramer's Rule or other algebraic methods.

$$(x_1, x_2, \dots, x_{|G_j|}) \times A_j = (1, 1, \dots, 1).$$

With $(x_1, x_2, \dots, x_{|G_j|})$, u_i gets

$$(x_1, x_2, \dots, x_{|G_j|}) \times \begin{pmatrix} Q_i \\ Q_{V_2} \\ \vdots \\ Q_{V_{|G_j|}} \end{pmatrix} = Q_{V_1}.$$

In order to decrypt the ciphertext, u_i needs to compute $e(P_{pub}, rQ_{V_1})$. With the knowledge of the private key S_i, it can do via:

$$\begin{aligned} & e(P_{pub}, r_j Q_{V_1}) \\ & \quad = e(P_{pub}, r_j(x_1 Q_i + x_2 Q_{V_2} + \dots + x_{|G_j|} Q_{V_{|G_j|}})) \\ & \quad = e(P_{pub}, r_j x_1 Q_i) \cdot e(P_{pub}, r_j(x_2 Q_{V_2} + \dots + x_{|G_j|} Q_{V_{|G_j|}})) \\ & \quad = e(r_j P, x_1 s Q_i) \cdot e(P_{pub}, x_2 r_j Q_{V_2} + \dots + x_{|G_j|} r_j Q_{V_{|G_j|}}) \\ & \quad = e(U_1, x_1 S_i) \cdot e(P_{pub}, x_2 U_2 + \dots + x_{|G_j|} U_{|G_j|}). \end{aligned}$$

Then, u_i can recover the session key

$$K_j = V_j \oplus H_2(e(U_1, x_1 S_i) \cdot e(P_{pub}, \sum_{i=2}^{|G_j|} x_i U_i)).$$

SELF-HEALING. Without loss of generality, suppose u_i lost the broadcast message for a session $t < j$. As far as it belongs to the session group G_t, it picks up the polynomial z_t from the broadcast message B_j and forms the $|G_t| \times |G_t|$ matrix A_t as operations in the procedure of KEY RECOVERY. Then u_i solves the following system of equations.

$$(x_1, x_2, \dots, x_{|G_t|}) \times A_t = (1, 1, \dots, 1).$$

With $(x_1, x_2, \dots, x_{|G_t|})$, u_i gets

$$(x_1, x_2, \dots, x_{|G_t|}) \times \begin{pmatrix} Q_i \\ Q_{V_2} \\ \vdots \\ Q_{V_{|G_t|}} \end{pmatrix} = Q_{V_1}.$$

After that, with the knowledge of its private key S_i, u_i computes $e(P_{pub}, r_t Q_{V_1})$ as follows:

$$\begin{aligned} & e(P_{pub}, r_t Q_{V_1}) \\ & \quad = e(P_{pub}, r_t(x_1 Q_i + x_2 Q_{V_2} + \dots + x_{|G_t|} Q_{V_{|G_t|}})) \\ & \quad = e(P_{pub}, r_t x_1 Q_i) \cdot e(P_{pub}, r_t(x_2 Q_{V_2} + \dots + x_{|G_t|} Q_{V_{|G_t|}})) \\ & \quad = e(r_t P, x_1 s Q_i) \cdot e(P_{pub}, x_2 r_t Q_{V_2} + \dots + x_{|G_t|} r_t Q_{V_{|G_t|}}) \\ & \quad = e(U_1, x_1 S_i) \cdot e(P_{pub}, x_2 U_2 + \dots + x_{|G_t|} U_{|G_t|}). \end{aligned}$$

Finally, u_i recovers the lost session key

$$K_t = V_t \oplus H_2(e(U_1, x_1 S_i) \cdot e(P_{pub}, \sum_{i=2}^{|G_t|} x_i U_i))$$

If more than one broadcast messages get lost, the operations of session key recovery are the same as aforementioned.

ADD AND REVOKE USER. If a new user u_{new} apply for joining the session j, GM checks the validity of its identity firstly. If it is an authorized user, in the procedure of

BROADCAST, GM constructs a new $(|G_j|-1) \times |G_j|$ matrix and computes new $Q_{V_i}(1 \leq i \leq |G_j|)$ which should includes Q_{new}.

If a user u_{rov} is revoked in session j, what GM should do is constructing a new $(|G_j|-1) \times |G_j|$ matrix and computes new $Q_{V_i}(1 \leq i \leq |G_j|)$ which should excludes Q_{rov}.

The adding and revoking operations are very efficient in our scheme. For the condition that more than one user joining or revoking, the operations are the same as aforementioned.

5. Analysis of Performance

In this section, we analyze the security of the proposed scheme firstly. Then we analyze the efficiency of the proposed scheme in terms of storage overhead, computation overhead, and communication overhead.

5.1. Analysis of security

According to the construction described before, we can say that our construction realizes self-healing key distribution scheme. Here we show that our scheme is collusion-free for any coalition of non-authorized users. In our scheme, if one user wants to obtain session key, it should compute $e(U_1, x_1 S_i)$ in the procedure of KEY RECOVERY. Therefore, only the authorized users can recover the session key. In addition, due to the difficulty of solving DLP, any coalition of non-authorized users cannot derive the private keys of authorized users from their public keys. Furthermore, session keys are independent to each other and according to the uniform distribution, which subsumes forward security and backward security.

In previous secret sharing-based self-healing key distribution schemes, the personal key can be reused on the condition that less than a threshold number of users are revoked. In this paper, personal key pair has nothing to do with the number of revoked users and can be reused as long as it is not disclosed. In addition, our scheme enables a user to recover from a single broadcast message all keys associated with sessions in which it belongs to the session group.

5.2. Analysis of efficiency

Different from the existing papers, we take advantage of a bilinear pairings-based broadcast encryption to design a self-healing key distribution scheme. As far as we know, it is the first self-healing key distribution scheme using bilinear pairings. In this section, we analyze the efficiency of the scheme.

In terms of storage overhead, each user only stores its public/private key pair and GM's public key P_{pub}. Therefore, the storage overhead for end users is a constant. Furthermore, the private key has nothing to do with the number of revoked users and can be reused as long as it is not disclosed. In addition, our scheme enables a user to recover from a single broadcast message all keys associated with sessions in which it belongs to the session group.

Generally speaking, GM takes up more resources than end users and thus can perform more complex computation. We elaborate on analyzing the computation overhead at users. In the j-th session, all the computations in the procedure of KEY RECOVERY are as follows: (1) Solving a set of linear equations with $|G_j|$ variables; (2) $|G_j|$ scalar multiplications in the cyclic additive group G_1; (3) $|G_j|$-1 additions in the cyclic additive group G_1; (4) Two pairings computation; (5) One hashing computation; (6) One XOR operation. Generally speaking, the computation of the pairing is the most time-consuming in pairings-based cryptosystems. Although there have been many papers talking about the complexity of pairings and how to speed up the computation of bilinear pairings ([23] and [24]), the computation overhead of bilinear pairings are still larger than the scalar multiplication, let alone other types of computation. Therefore, the main computation overhead of the scheme comes from (4).

The communication overhead comes from broadcast messages $B_j = \{z_1, \ldots, z_j\}$. z_j is composed of $U_i(1 \leq i \leq |G_j|)$ and V_j. Therefore, the size of z_j increases in direct proportion to the number of $|G_j|$. The length of broadcast is $\sum_{i=1}^{j} |G_i| logq + jlogq$. $logq$ is the size of session key.

6. The Possibility of Mutual-Healing

More et al. in [10] pointed out that the protocol in [7] suffered from inconsistent robustness. Subsequently, they used a sliding window to make error recovery consistently robust: after the initial SETUP procedure, any lost key can be recovered as long as two sufficiently close broadcast messages–one before it and one after it–are received. Similar technique was taken in [19]. The minimum size of the window can be dynamic adjusted according to the condition of networks. Both [10] and [19] guaranteed that authorized users can recover window size session keys as long as they receive corresponding broadcast messages. However, how to recover the session key if the last broadcast message gets lost or more than sliding window number broadcast messages get lost is never addressed clearly. Clearly, it is impossible to make users completely self-healing according to existing self-healing key distribution mechanism. In view of some concrete applications, such as live and pay-per-view TV, have strictly requirement of freshness. They'd better lose only a limited number of broadcast messages. It is meaningful to detect counterpart measures to deal with the aforementioned issues. In this section, we reconsider the idea of mutual-healing between neighboring users in wireless communication networks and present a practical technique to realize it.

The idea of mutual-healing was proposed in [14] by Muhammad et al. without exploring technical details. That is, if a user has missed more than a fixed number broadcast messages or the last broadcast message, it can get assistance from its neighboring nodes. The neighboring users in the same session group cooperate with each other forwarding broadcast messages which their neighboring users miss. Thus the robustness of self-healing key distribution scheme is achieved. It was claimed in [14] that there are two requirements for mutual-healing: the authentication of the requesting user and the authorization of the requested session key. On the one hand, in order to avoid attacks on their limited resource, effective authentication of requesting user must be developed to identify misbehaving users. On the other hand, as messages are broadcasted in the form of plain-text, anyone in this communication networks can receive them. We argue that the second authentication is unnecessary. Instead, the neighboring user only needs to forward the broadcast message which corresponds to the requested session key. If the requesting user is authorized for the session, it would be able to recover the session key. Otherwise, even unauthorized users receive the broadcast message, they can not recover the session key.

6.1. The mutual-healing scheme

Many wireless networks have an intrinsic property that nodes are stationary. Therefore, we can bound LBKs (the Location-Based Keys) of users to both their identities and geographic locations rather merely their identities or locations as in conventional schemes. Based on their LBKs, two neighboring users can perform node-to-node neighborhood authentication. In order to reduce communication overhead, we restrict that mutual-healing only happens between one-hop neighboring nodes. There are three steps in a mutual-healing scheme, we will introduce them one by one.

RANGE-BASED LOCATION. There are many methods to localize users after deployment. We adopt the first method introduced in [20]. This step may complete within several minutes after the deployment of networks. We assume that a group of mobile robots are dispatched to sweep across the whole network field along preplanned routes. Mobile robots have GPS (Global Positioning System) capabilities, as well as more powerful computation and communication capacities than ordinary users. The leading robot equipped with the a master secret k. To localize a user, say A, mobile robots run the secure range-based localization protocol given in [21] or [22] to measure their respective absolute distance to user A and co-determine l_A, the location of A. Subsequently, the leading robot calculates $LK_A = kH(ID_A \parallel l_A)$, and sends $\langle \{LK_A \parallel l_A\}_{Q_A}, h_{Q_A}(LK_A \parallel l_A)\rangle$ to A. Henceforth, $\{M\}_k$ means encrypting message M with key k, and $h_k(M)$ refers to the MIC (Message Integrity Code) of message M under key k.

Upon receipt of the message, user A first uses its private key to decrypt LK_A and l_A and then regenerates the MIC. If the result matches with what the robot sent, A saves LK_A and l_A for subsequent use. Following this process, all the nodes can be furnished with their respective locations and LBKs. After that, mobile robots leave the sensor field and the leading robot should securely erase k from its memory. During subsequent network operations, node addition may be necessary to maintain good network connectivity. The localization of new nodes can be done in the same manner.

MUTUAL-HEALING REQUEST. During the procedures of self-healing key distribution, if a user has missed more than a fixed number broadcast messages or the last broadcast message, it looks for assistance from its neighboring nodes.

- $A \rightarrow * : ID_A, l_A, j$;
- $B \rightarrow A : ID_B, l_B, (B_j)_{K_{BA}}$;

Suppose user A wishes to receive broadcast message B_j, A locally broadcasts an authentication request including its identity ID_A, location l_A and the sequence of the expect broadcast message j. Upon receipt of such a request, user B first needs to ascertain that the claimed location l_A is in its transmission range by verifying if the Euclidean distance $\parallel l_A - l_B \parallel \leq \mathcal{R}$, where $\mathcal{R}$ is the radius of communication range of each user. This check is the baseline defense against the attack that adversaries surreptitiously tunnel authentication messages between B and a virtually non-neighboring node. Without the location check, B and that victim will falsely believe that they are neighbors.

If the inequality does not hold, user B simply discards the request. Otherwise, B calculates a shared key as $K_{BA} = e(LK_B, H(ID_A \parallel l_A))$. It then unicasts a reply to node A including its ID ID_B, location l_B key distribution broadcast and encrypted message $(B_j)_{K_{BA}}$.

VERIFICATION. Upon receiving the reply, user A also first checks if the inequality $\parallel l_A - l_B \parallel \leq \mathcal{R}$ holds. If so, it proceeds to derive a shared key as $K_{AB} = e(LK_A, H(ID_B \parallel l_B)) = K_{BA}$ between it and user B whereby to decrypt the message $(B_j)_{K_{BA}}$ and get the broadcast message B_j. If the inequality doesn't hold, user A discards the message received from user B. Using received broadcast message B_j, the authorized user A can recover the lost session keys. By this way, mutual-healing key distribution scheme is fulfilled.

6.2. The security of the proposed mutual-healing scheme

It is general assumption that adversaries do not launch active and explicit pinpoint attack on users during deployment and initialization which usually dose not last too long. According to [20], this assumption in the step of Range-based Location is reasonable in that mobile robots are much fewer than ordinary nodes and can be equipped with tamper-proof hardware and putting them under super monitor.

A false requesting user might send an authentication request with a forged location within user B's range. Since the false requesting user does not hold the LBK corresponding to the forged location, even it deceive B into believing it is in B's range, it can not get any useful information from B. There are some false requesting users who might mount DoS (Denial of Service) attack by continuously sending bogus mutual-healing requests to allure legitimate nodes into endless processing of such messages. Because the number of neighbors of any node is limited in reality, abnormally many mutual-healing requests are highly likely an indicator of malicious attacks. If this situation happens, user B will discard the requesting message and stop assistance. By the same reason, if a requesting node receives too many replies, it only decrypts a fixed number messages in order to save its limited resource and avoids exhausting-resource attack. Furthermore, we assume that there are efficient mechanisms available for authorized nodes to report such an abnormality to the sink.

7. Conclusion

A self-healing key distribution scheme using bilinear pairings is proposed in this paper. The new scheme achieves good features. Firstly, the storage overhead for user is a constant. Secondly, the scheme is collusion-free for any coalition of non-authorized users, private key has nothing to do with the number of revoked users and can be reused only if it is not disclosed. While in secret sharing-based self-healing key distribution schemes, the personal key can be reused on the condition that less than threshold number users are revoked. Furthermore, session keys are independent to each other and according to uniform distribution and thus keeps forward security and backward security. In addition, our scheme enables a user to recover from a single broadcast message all keys associated with sessions in which he belongs to the session group. Finally, we present technique to perform mutual-healing between neighboring users. As far as we know, it's the first time to explore technical details of mutual-healing.

References

[1] D. Balenson, D. McGrew, and A. Sherman, Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization, *IETF Internet draft (work in progress)*, Aug. 2000.

[2] W. Ku and S. Chen, An Improved Key Management Scheme for Large Dynamic Groups Using One-Way Function Trees, *International Conference on Parallel Processing Workshops*'**03**, pp.391-396, Oct. 2003.

[3] A. Fiat and T. Tessa, Dynamic Traitor Tracing, *Journal of Cryptology*, 14:211-223, 2001.

[4] D. Halevy and A. Shamir, The LSD Broadcast Encryption Scheme, *Advances in Cryptology-Crypto*'**02**, LNCS, 2442:47-60, 2002.

[5] M. Naor and B. Pinkas, Efficient Trace and Revoke Schemes, *Financial Cryptography*'**2000**, LNCS, 1962:1-21, 2000.

[6] D. Naor, M. Naor, and J. Lotspiech, Revocation and Tracing Schemes for Stateless Receivers, *Advances in Cryptology-Crypto*'**01**, LNCS, 2139:41-62, 2001.

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

[8] D. Liu, P. Ning, and K. Sun, Efficient Self-healing Key Distribution with Revocation Capability, *Proceeding of the* **10***th ACM*, 2003.

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

[10] S. M. More, M. Malkin, J. Staddon, and D. Balfanz, Sliding Window Self-healing Key Distribution with Revocation, *ACM Workshop on Survivable and Self-Regenerative Systems*, pp.82-90, 2003.

[11] G. Sáez, On Threshold Self-healing Key Distribution Schemes, *Cryptography and Coding*, LNCS, 3796:340-354, 2005.

[12] G. Sáez, Self-healing Key Distribution Schemes with Sponsorization, *International Federation for Information Processing, IFIP*'**05**, LNCS, 3677:22-31, 2005.

[13] R. Dutta and S. Mukhopadhyay, Improved Self-healing Key Distribution with Revocation in Wireless Sensor Network, *Wireless Communications and Networking Conference*, pp.2963-2968, 2007.

[14] J. B. Muhammad and and M. Ali, Self-healing Group Key Distribution, *International Journal of Network Security*,1(2):110-117, 2005.

[15] 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 5*, pp.14-23, 2007.

[16] A. Shamir, Identity-Based Cryptosystems and Signature Scheme, *Proceedings of Crypto*'**84**, pp.47-53, 1984.

[17] D. Boneh and M. Franklin, Identity Based Encryption From the Weil Pairing, *Advanced in Cryptology-CRYPTO*'**01**, pp.213-229, 2001.

[18] X. Du, Y. Wang, J. Ge, and Y. Wang, An ID-Based Broadcast Encryption Scheme for Key Distribution, *IEEE Transactions on Broadcasting*, 51(2):264-266, Jun. 2005.

[19] X. Zou and Y. Dai, A Robust and Stateless Self-Healing Group Key Management Scheme, *International Conference on Communication Technology, ICCT*'**06**, pp.1-4, Nov. 2006.

[20] Y. Zhang, W. Liu, W. Lou, and Y. Fang, Location-based Compromise-Tolerant Security Mechanisms for Wireless Sensor Networks, *IEEE Journal on Selected Areas in Communications*, 24(2):247-260, Feb. 2006.

[21] S. Capkun and J.-P. Hubaux, Secure Positioning of Wireless Devices with Application to Sensor Networks, *Proceedings of IEEE INFOCOM*, Miami, Florida, pp.19171928, Mar. 2005.

[22] Y. Zhang, W. Liu, and Y. Fang, Secure Localization in Wireless Sensor Networks, *Proceedings of IEEE MILCOM*, 2005.

[23] P. S. L. M. Barreto, H. Y. Kim, B. Lynn, and M. Scott, Efficient Algorithms for Pairing-Based Cryptosystems, *Advances in Cryptology-Crypto*'**02**, LNCS, 2442:354-368, 2002.

[24] S. D. Galbraith, K. Harrison, and D. Soldera, Implementing the Tate Pairing, *Proceedings of the* **5***th International Symposium on Algorithmic Number Theory*, LNCS, 2369:324-337, 2002.