



A mutual-healing key distribution scheme in wireless sensor networks

Biming Tian^a, Song Han^a, Jiankun Hu^{b,*}, Tharam Dillon^a

^a DEBI Institute, Curtin University, Perth 6845, Australia

^b School of Computer Science and IT, RMIT University, Melbourne 3001, Australia

ARTICLE INFO

Article history:

Received 22 January 2010

Received in revised form

1 September 2010

Accepted 5 September 2010

Keywords:

Sensor network

Security

Self-healing

Key distribution

ABSTRACT

How to establish secure session keys is one of the central tasks for wireless sensor network communications. General key distribution schemes for traditional computer networks could not be directly shifted to wireless sensor network environments as broadcast messages may be lost due to sensor network internal factors or external attacks. Self-healing key distribution schemes, therefore, have been proposed to address packet loss issues since 2002. The essential issue that self-healing key distribution mechanism addressed is the fixed-number of broadcast messages (excluding the last broadcast message) loss. In other words, a node could not recover its new session keys if a node has missed more than a fixed number broadcast messages or the last broadcast message in a self-healing key distribution scheme for wireless sensor networks. This paper aims to address this emerged issue and provide a new key distribution scheme: mutual-healing key distribution scheme for wireless sensor networks. This mutual-healing key distribution can enable a node in a wireless sensor network to recover its new session key although its last broadcast message was lost. A formal definition for mutual-healing key distribution will also be proposed in this paper. The proposed mutual-healing key distribution scheme is based on bilinear pairings. The scheme is collusion-free for any coalition of non-authorized nodes. Each node's private key has nothing to do with the number of revoked nodes and can be reused as long as it is not disclosed. The storage overhead for each node is a constant.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Key management including key distribution and key update is significant for maintaining private communications in any dynamic networks (Balenson et al., 1999; Ku and Chen, 2003; Staddon et al., 2002; Liu et al., 2003; Jiang et al., 2007; Han et al., 2007; Han et al., 2009; Hong and Kang, 2005; Hu and Han, 2009; Hu et al., 2010; Eltoweissy et al., 2004a; Lu et al., 2006; Sufi et al., 2010; Xi et al., 2010). Wireless sensor networks, especially mobile ad hoc networks, as one of dynamic networks, therefore rely on a secure key management. This is because membership frequently changes in large dynamic group communications of wireless sensor networks. In order to keep the security of communications, the session key has to be updated on each membership change. Therefore, one of the important questions is how to distribute and update session keys in a secure and efficient way for large dynamic wireless sensor networks. In recent years, many schemes on distributing session keys for large group communication have been proposed. These existing schemes focused on different key updating mechanism. For example, LKH (Logical Key Hierarchy)-based schemes (Wong et al., 2000) and OFT (One-way Function

Tree) based schemes (Balenson et al., 1999; Ku and Chen, 2003) devote to reduce the size of the rekeying message. Broadcast encryption addresses the problem of sending encrypted messages to a large node group so that the encrypted messages can only be decrypted by a dynamic changing privileged subset (Fiat and Tessa, 2001; Halevy and Shamir, 2002; Naor and Pinkas, 2000). EBS (Exclusion Basis System) based approach was proposed in Eltoweissy et al. (2004b), and then be put into use for sensor networks in Eltoweissy et al. (2004a) and Moharrum et al. (2006). The members store less number of keys than LKH tree for the multicast group of the same size. 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 wireless sensor networks. The key distribution broadcast for a particular session might never reach some nodes. A naive solution is requesting retransmission. On the one hand, both requesting and re-transmission messages would incur more communication overhead. In a very large communication group, such individual interactions place a heavy burden on the group manager. On the other hand, nodes may reveal their current locations by sending messages in some high security environments. All these issues can be addressed by self-healing key distribution schemes (Staddon et al., 2002; Liu et al., 2003; Blundo et al., 2004; More et al., 2003;

* Corresponding author. Tel.: +61 3 99259793; fax: +61 3 9662 1617.

E-mail addresses: biming.tian@cbs.curtin.edu.au (B. Tian), song.han@cbs.curtin.edu.au (S. Han), jjiankun@cs.rmit.edu.au (J. Hu), haram.dillon@cbs.curtin.edu.au (T. Dillon).

Sáez, 2005a, b; Dutta and Mukhopadhyay, 2007; Muhammad and Ali, 2005; Jiang et al., 2007; Zou and Dai, 2006; Dutta et al., 2007; Han et al., 2009; Hong and Kang, 2005; Kausar et al., 2007). Self-healing key distribution enables large and dynamic group nodes to establish group keys over an unreliable network for secure communications. The main property of self-healing key distribution schemes is that, even if at the beginning of certain sessions some broadcast packets get lost, group nodes 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 kind of scheme, nodes do not need to send any requesting message to the group manager and do not need to update their personal keys. In this regard, self-healing key distribution schemes are noninteractive ones which can hence reduce the network traffic, decrease the workload on the group manager, and lower the risk of node exposure through traffic analysis. Therefore, self-healing key distribution schemes are desirable for both efficiency and security reasons in wireless sensor networks.

An emerged scenario is a node in a wireless sensor network may miss its last broadcast message. This node is therefore not able to recover its session key for the last session in a self-healing key distribution scheme. This is because self-healing key distribution needs the broadcast messages the node received in the previous sessions and those in the subsequent sessions (Suppose the current session is the one where broadcast messages are lost). To tackle this scenario, mutual-healing key distribution mechanism will be introduced. Mutual-healing key distribution can help a node who missed the last broadcast message to recover the session key for this last session.

Possible applications: Group communications over low-cost channels in different fields can benefit from mutual-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 nodes. For example, video conference for commercial content distribution or electronic services in which the contents are highly sensitive. Self-healing key distribution schemes are also good levers in military-oriented operations, scientific explorations and rescue missions. The nodes in these environment are powered by batteries. They may experience short-term off-line and rejoin the group once the power is on again. Also the nodes may move in and out of communication range frequently and experience burst packet losses.

Contribution. The contribution of this paper are the following. First of all we define a security model of computationally secure self-healing key distribution scheme. This definition outperforms those in Staddon et al. (2002) and Liu et al. (2003) in two aspects. The first one is that there is no threshold on the number of revoked nodes. The scheme is collusion-free for any coalition of non-authorized nodes. The second one is that a node can recover from a single broadcast message all keys associated with sessions in which it belongs to the session group. These two properties add flexibility to self-healing key distribution scheme. We propose a self-healing key distribution scheme using bilinear pairings. The scheme is highlighted by several desirable features. Firstly, it is collusion-free for any coalition of non-authorized nodes. Secondly, the private key has nothing to do with the number of revoked nodes and can be reused as long as it is not disclosed. Thirdly, the storage overhead for node is a constant. Subsequently, we discuss the requirement, state formal definition, and develop technical details of mutual-healing mechanism. It is the first time to realize mutual-healing key distribution technique.

Organization. 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, mutual-healing, and Identity-based cryptography. In Section 3, we briefly introduce the preliminaries

to be used in the design of mutual-healing key distribution protocol. In Section 4, we give system parameters, security model and a formal definition. In Section 5, a concrete construction for mutual-healing key distribution is proposed. In Section 6, we provide security analysis and efficiency comparison. We conclude this paper and point out future research in Section 7.

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. (2002). 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. Subsequent works focus on improving performance or adding some new properties which make the self-healing key distribution mechanism more flexible or more robust. Liu et al. generalized the definitions in Staddon et al. (2002) and gave some constructions in Liu et al. (2003). The scheme reduces communication overhead and storage overhead by introducing a novel personal key distribution technique. Blundo et al. (2004) showed an attack that can be applied to the first construction in Staddon et al. (2002), developed a new mechanism under a slightly modified framework. More et al. (2003) used a sliding window to address three problems in Staddon et al. (2002). The three problems were inconsistent robustness, high overhead and expensive maintenance costs. Dutta et al. developed a new self-healing key distribution scheme in Dutta and Mukhopadhyay (2007). The scheme has significant improvement in terms of both storage overhead and communication overhead. The schemes (Staddon et al., 2002; Liu et al., 2003; Blundo et al., 2004; More et al., 2003; Dutta and Mukhopadhyay, 2007) are based on Shamir's secret sharing. They have a common property: the maximum number of revoked nodes is constraint to the degree of the polynomial. Sáez (2005a,b) 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 nodes that can be revoked instead of a threshold one. All of the schemes (Staddon et al., 2002; Liu et al., 2003; Blundo et al., 2004; More et al., 2003; Sáez, 2005a, b; Dutta and Mukhopadhyay, 2007) are unconditionally secure.

Computationally secure self-healing key distribution schemes emerged recently. They achieved good properties at the cost of slight relax security requirements. Jiang et al. proposed an efficient self-healing group key scheme with time-limited node revocation based on DDHC (dual directional hash chains) in Jiang et al. (2007). 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. The scheme in Dutta et al. (2007) also based on hash function. It reduced communication overhead and computation overhead greatly without any increase in the storage overhead.

In general, according to the security level, the existing self-healing key distribution schemes can be classified as unconditionally secure self-healing key distribution schemes and computationally secure self-healing key distribution scheme. The former has more strict security while the latter is more flexible and efficient. According to the cryptographic primitives, the schemes can be classified as polynomial secret sharing based schemes, vector space secret sharing based schemes, and hash

function based schemes. Polynomial secret sharing is the most common technique used to realize self-healing key distribution. It was used in the pioneering paper (Staddon et al., 2002) and was followed by several subsequent works. However, the maximum number of revoked nodes 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 nodes 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 hash function based self-healing key distribution schemes. However, the feature of resisting collusion of revoked nodes and new joined nodes cannot be assured, due to the properties of one-way hash function.

2.2. Motivation for mutual-healing key distribution

We cannot deny the novel property of the self-healing idea. However, some improvements are still necessary for the original scheme (Staddon et al., 2002). For example, it may allow an authorized subset of nodes in the group to sponsor a new node without the help of the group manager. Sáez (2005b) considered the feature that a coalition of nodes sponsor a node outside the group for one session. This feature added a dynamic character to the self-healing key distribution scheme. There are still other schemes (Hong and Kang, 2005; Kausar et al., 2007) which focus on improving the efficiency.

More et al. (2003) pointed out that the protocol in Staddon et al. (2002) suffered from inconsistent robustness. The so-called inconsistent robustness is that some session keys cannot be recovered if the corresponding broadcast messages are lost, no matter how many other update messages are received. For example, if the broadcast message for the last session gets lost, nodes cannot recover the last session key by themeless even they receive all the other broadcast messages. Subsequently, they used a sliding window to make error recovery consistently robust. That is, after the initial SET-UP procedure, any lost key can be recovered as long as two sufficiently close broadcast messages (one before it and the other after it) are received. Similar technique was used in Zou and Dai (2006). The size of the window can be dynamic adjusted according to the condition of networks. Both More et al. (2003) and Zou and Dai (2006) guaranteed that authorized nodes 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 has never been addressed clearly. Obviously, it is impossible to make nodes 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. The customers would better lose only a limited number of broadcast messages. In the group communication, the last session usually is of great importance. The authorized nodes would never like to miss it. Therefore it is significant to detect counterpart measures to deal with the aforementioned issues.

Muhammad and Ali (2005) considered incorporating the self-healing feature to SD(Subset Difference) method, which was first proposed by Naor et al. (2001). 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 node 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 node

misses the last broadcast message, it cannot recover the last session key by performing self-healing. To provide a counter-measure for this situation, it can look for assistance from its neighboring nodes too. However, the paper (Muhammad and Ali, 2005) only talked about the feasibility of mutual-healing without exploring any technical detail.

2.3. Identity-based cryptography

In identity-based cryptography, the public key of a user is some unique information about the identity of the user. Identity-based schemes can allow any party to generate a public key from a known identity value such as an ASCII string. A trusted third party, called the private key generator (PKG), generates the corresponding private keys. Therefore, any party without using certificates can verify the public key of a user. This eliminates the need for a public key distribution infrastructure.

Shamir (1984) proposed the first identity-based cryptography to alleviate many of the problems inherent with managing certificates in 1984. Boneh and Franklin (2001) proposed the first practical identity-based encryption scheme in 2001. Since then, many ID-based cryptographic schemes have been proposed using bilinear pairings. Inspired by the idea of Boneh and Franklin (2001), Du et al. proposed a broadcast encryption scheme for key distribution in Du et al. (2005). By extending the broadcast encryption scheme, we will proposed a mutual-healing key distribution scheme for wireless sensor networks.

3. Preliminaries

In this section, we briefly describe bilinear pairings, 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 a 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 PKI involves a trusted KGC(key generation center) and nodes. Nodes' private keys are calculated by KGC and send to the node 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 node submits its identity to KGC. KGC computes the node's public key $Q_{ID} = H_1(ID)$ and private key $S_{ID} = sQ_{ID}$, then privately returns $S_{ID} = sQ_{ID}$ to the node.

4. Security model and definition for mutual-healing key distribution

In this section, we present system parameters, security model and formal definition for mutual-healing key distribution, and a new key distribution scheme.

4.1. System parameters

Let $U = \{u_1, \dots, u_n\}$ be the finite universe of nodes. Each node 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 adding 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 node u_i is not removed by GM from the group. Let $R_j \subseteq G_{j-1}$ denote the set of revoked group nodes in session j and $J_j \subset U \setminus G_{j-1}$ denote the set of nodes who join the group in session j with $R_j \cap J_j = \emptyset$. Hence, $G_j = (G_{j-1} \cup J_j) \setminus R_j$ for $j \geq 2$ and by definition $G_1 = U$. Moreover, for $j \in \{1, \dots, m\}$, the session key K_j is randomly chosen by GM and according to the uniform distribution. For any nonrevoked node $u_i \in G_j$, 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. Mutual-healing key distribution security model

The idea of mutual-healing was discussed in Muhammad and Ali (2005) without exploring any technical detail. Just as self-healing means that nodes are capable of recovering lost group keys on their own, mutual-healing implies that nodes help each other in recovering some lost group keys. The central concept of mutual-healing is that if a node has missed more than a fixed number broadcast messages or the last broadcast message, it can get assistance from its neighboring nodes. The neighboring nodes in the same session group cooperate with each other forwarding broadcast messages which their neighboring nodes miss. By this way the robustness of self-healing key distribution scheme is achieved.

It was claimed in Muhammad and Ali (2005) that two requirements are necessary for mutual-healing. They are the authentication on the requesting node and the authorization on the requested session key. On the one hand, in order to avoid attacks on their limited resource, effective authentication on the

requesting node must be developed to identify misbehaving nodes. On the other hand, any entities in this communication networks can access broadcast messages because broadcast messages are broadcasted in plain-text within the communication group. We argue that the second authentication is unnecessary. Instead, the neighboring nodes only need to forward the broadcast message which corresponds to the requested session key. If the requesting node is authorized for the session, it would be able to recover the session key. Otherwise, even unauthorized nodes receive the broadcast message from their neighbors, they can not recover the session key.

To further clarify our design goal and facilitate understanding of readers, according but not constraint to the security model of Dutta et al. (2007), we define the mutual-healing key distribution security model from four aspects. Definition 4.1 defines self-healing key distribution scheme with revocation capability. Definition 4.2 defines mutual-healing key distribution scheme Definition 4.3 defines forward secrecy and backward secrecy. Definition 4.4 defines resisting collusion properties.

Definition 4.1. Let $U = \{u_1, \dots, u_n\}$ and $j \in \{1, \dots, m\}$.

1. The scheme is a session key distribution with privacy if
 - (a) for any node u_i , the session key K_j is efficiently determined from B_j and the personal public/private key pair (Q_i, S_i) .
 - (b) for any set $R \subseteq U$ and $u_i \notin R$, it is computationally infeasible for node in R to determine the personal private key S_i .
 - (c) what node u_1, \dots, u_n learn from B_j cannot be determined from broadcasts or personal key pairs alone. That is, if we consider separately either the set of m broadcasts $\{B_1, \dots, B_m\}$ or the set of n personal key pairs $\{(Q_1, S_1), \dots, (Q_n, S_n)\}$, then it is computationally infeasible to compute session key K_j from either set.
2. The scheme has revocation capability. Particularly, there is no upper limitation of revocation. For each session j and $R_j \subseteq U$, GM can generate a broadcast message B_j such that for all $u_i \notin R_j$ can efficiently recover the session key K_j , but the revoked nodes in R_j cannot even knowing all the information broadcast in sessions $1, \dots, j$.
3. The scheme is self-healing if the following is true for any $r, 1 \leq r < j \leq m$: For any node $u_i \in G_r$ who is also a member in session j , the session key K_r is efficiently determined by (Q_i, S_i) and B_j .

Definition 4.2. Let $U = \{u_1, \dots, u_n\}$ and $j \in \{1, \dots, m\}$. The scheme is mutual-healing if the following is true:

1. For any node $u_i \in U$, if it misses more than a fixed number of broadcast messages or the last broadcast message, it can generate and broadcast effective requesting message to its neighbors.
2. For any u_i 's neighboring node u_j , it can verify whether u_i is its qualified neighboring node or a malicious one. If u_i is a qualified neighboring node and u_j is an authorized node for the requested broadcast, u_j generates and sends responsive message to u_i .
3. The requester u_i can verify whether the responder u_j is its neighbor or not. If u_j is its neighbor, u_i can decrypts the responsive message and thus get the requested broadcast message.

Definition 4.3. Let $U = \{u_1, \dots, u_n\}$ and $j \in \{1, \dots, m\}$. The scheme guarantees both forward security and backward security if

1. for any set $R \subseteq U$, and all $u_i \in R$ are revoked before session j , it is computationally infeasible for the nodes in R together to get

- any information about K_j , even with the knowledge of group keys K_1, \dots, K_{j-1} before session j .
2. for any set $J \subseteq U$, and all $u_i \in J$ join after session j , it is computationally infeasible for the nodes in J together to get any information about K_j , even with the knowledge of group keys K_{j+1}, \dots, K_m after session j .

Definition 4.4. Let $B \subset R_r \cup \dots \cup R_2$ be a coalition of nodes who are revoked from the group before session r and let $C \subset J_s \cup \dots \cup J_m$ be a coalition of nodes who join the group from session s with $r < s$.

The scheme is collusion-free for any coalition of non-authorized nodes if the coalition $B \cup C$ does not get any information about session keys K_j , for any $r \leq j < s$.

5. The proposed mutual-healing key distribution scheme

In this section, we will present a mutual-healing key distribution scheme for wireless sensor networks. The proposed mutual-healing key distribution scheme has the following procedures:

- (1) SYSTEM SET-UP;
- (2) BROADCAST;
- (3) KEY RECOVERY;
- (4) SELF-HEALING;
- (5) MUTUAL-HEALING; and
- (6) ADDING AND REVOKING NODE.

We need to point out the scheme is computationally secure scheme. All parameters and symbols used in this section have been defined in Sections 3 and 4. Otherwise, they will be defined in this section.

5.1. System set-up

GM obtains both public system parameters and all the public keys of possible nodes from the ID-based PKI. GM chooses m session keys K_1, \dots, K_m from \mathbb{Z}_q^* . The session keys are independent to each other and according to the uniform distribution. We also use the system parameters proposed in Section 3.

5.2. Broadcast

Suppose $|G_j|$ denotes the number of nodes 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:

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

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 $r_j \in \mathbb{Z}_q^*$,

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

where P is a generator for a BDH group G_1 as shown in Section 3.2.

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

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

5.3. Key recovery

When a node $u_i \in G_j$ receives the broadcast message B_j , it sets a vector $a_i = (0, \dots, 0, 1, 0, \dots, 0)$ 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 node 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 $e(P_{pub}, r Q_{V_1})$. With the knowledge of the private key S_i , it can do via:

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

Then, u_i can recover the session key

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

5.4. 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:

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

Finally, u_i recovers the lost session key

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

One may wonder how u_i got V_t if the t -th broadcast message was lost. In fact, u_i could get the broadcast message B_j . Then u_i retrieved the polynomial z_j from B_j . Finally, u_i could obtain V_t from z_j . If more than one broadcast messages get lost, the operations of session key recovery are the same as aforementioned.

5.5. Mutual-healing

Here we consider the mutual-healing between neighboring nodes in wireless communication networks and present a practical technique to realize it.

Many wireless networks have an intrinsic property that nodes are stationary. Therefore, we can bound LBKs (the location-based keys) of nodes to both their identities and geographic locations rather merely their identities or locations as in conventional schemes. Based on their LBKs, two neighboring nodes 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. In order to realize mutual-healing capability, The scheme has to execute range-based location operation after SET-UP procedure.

There are many methods to localize nodes. We adopt the first method in Zhang et al. (2006). 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) capability, as well as more powerful computation and communication capacities than ordinary nodes have. The leading robot equipped with the a master secret k . To localize a node, say u_i , mobile robots run the secure range-based localization protocol given in Capkun and Hubaux (2005) or Zhang et al. (2005) to measure their respective absolute distance to node u_i and co-determine l_i , the location of u_i . Subsequently, the leading robot calculates $LK_i = kH(ID_i \| l_i)$, and sends $\langle \{LK_i \| l_i\}_{Q_i}, h_{Q_i}(LK_i \| l_i) \rangle$ to u_i . $\{M\}_k$ 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, node u_i first uses its private key to decrypt LK_i and l_i and then regenerates the MIC. If the result matches with what the robot sent, u_i saves LK_i and l_i 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 have to securely erase k from its memory. During subsequent network operations, Adding nodes may be necessary in order to maintain good network connectivity. The localization of new nodes can be done in the same manner.

It is general supposed that adversaries do not launch active and explicit pinpoint attack on nodes during deployment and initialization which usually dose not last too long. According to Zhang et al. (2006), this assumption in range-based location operation 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.

During the procedures of self-healing key distribution, if a node has missed more than a fixed number broadcast messages or the last broadcast message, it looks for assistance from its neighboring nodes. The realization of mutual-healing includes three steps. We will introduce them one by one.

5.5.1. Mutual-healing request

Suppose node u_i wishes to receive broadcast message B_t , u_i locally broadcasts an authentication request including its identity ID_i , location l_i and the sequence number of the expected broadcast message t .

5.5.2. Mutual-healing response

Upon receipt of a request, the neighboring node u_j first needs to ascertain that the claimed location l_i is within its one-hop communication range by verifying if the Euclidean distance $\|l_i - l_j\| \leq \mathcal{R}$, where \mathcal{R} is one-hop communication distance.

If the inequality does not hold, node u_j simply discards the request. Otherwise, u_j calculates a shared key as $K_{ji} = e(LK_j, H(ID_i \| l_i))$. Then it unicasts a reply to node u_i including its identity ID_j , location l_j and encrypted broadcast message $(B_t)_{K_{ji}}$.

- $u_i \rightarrow * : ID_i, l_i, t;$
- $u_j \rightarrow u_i : ID_j, l_j, (B_t)_{K_{ji}};$

5.5.3. Verification

Upon receiving the response, node u_i also first checks if the inequality $\|l_i - l_j\| \leq \mathcal{R}$ holds. If the inequality does not hold, u_i directly discards the message received from u_j . Otherwise, u_i proceeds to derive a shared key as $K_{ij} = e(LK_i, H(ID_j \| l_j)) = K_{ji}$ between it and the node u_j whereby to decrypt the message $(B_t)_{K_{ji}}$ and get the broadcast message B_t . Using the broadcast message B_t and its public/private key pair, the authorized node u_i can recover the lost session keys.

5.5.4. Adding and revoking node

If a new node u_{new} applies for joining the session j , GM checks the validity of its identity firstly. If it is an authorized node, 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 include Q_{new} .

If a node u_{rov} is revoked from the session j , what GM should do is constructing a new $(|G_j| - 1) \times |G_j|$ matrix and computing new $Q_{V_i} (1 \leq i \leq |G_j|)$ which should exclude Q_{rov} .

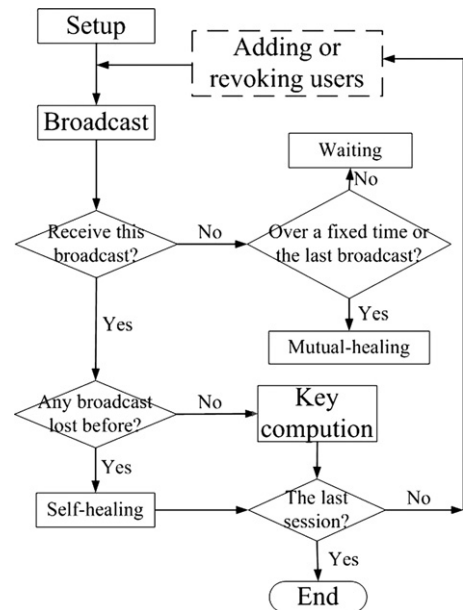


Fig. 1. The process of the mutual-healing key distribution scheme, where the panes are operation which must be executed in each round, the dashed panes represent the operations which may not be executed in every round.

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

Figure 1 shows all the procedures involved in a mutual-healing key distribution scheme.

6. Security and efficiency

Security analysis for the proposed mutual-healing key distribution scheme will be provided in this section. Performance discussion for the scheme will also be presented.

6.1. Security of the proposed scheme

In this subsection, we analyze the security of the proposed scheme. More precisely, we show that our construction satisfies all the security requirements in our security model described in Section 4.2.

6.1.1. Property 1: Self-healing key distribution

We show our construction satisfies the security requirements described in Definition 4.1.

1. The scheme is a session key distribution scheme.

(a) Any node $u_i \in G_j$ can recover the session key K_j from B_j and the personal public/private key pair (Q_i, S_i) . This is because: When a node $u_i \in G_j$ receives the broadcast message B_j , it sets a vector $c_i = (0, \dots, 0, 1, 0, \dots, 0)$ with $|G_j|$ elements, and only the i -th element is 1. Define a new matrix C_j using c_i with $i = 1, 2, \dots, |G_j|$. Then C_j is a $|G_j| \times |G_j|$ matrix

$$C_j = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_{|G_j|} \end{pmatrix}.$$

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

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

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 , u_i can do so via:

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

Then, u_i can recover the session key K_j by

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

(b) Any coalition $R \subseteq U$ of non-authorized nodes cannot derive the private key S_i of any authorized node $u_i \notin R$. This is

because each node is preloaded with a public/private key pair (Q_i, S_i) before deployment. The key pairs are computed by an ID-based PKI which can realize public and private keys without certificate management (Han et al., 2004). Because it is infeasible to solve the discrete logarithm problem $S_{ID} = sQ_{ID}$, any coalition $R \subseteq U$ of non-authorized nodes cannot derive the private key S_i of any authorized node $u_i \notin R$.

(c) It is computationally infeasible to compute session key K_j from either broadcast messages or personal public/private key pairs. This is because the j -th session key is computed from V_j and $e(P_{pub}, r_j Q_{V_1})$. On the one hand, V_j is taken from the broadcast message while

$$e(P_{pub}, r_j Q_{V_1}) = e(U_1, x_1 S_i) \cdot e \left(P_{pub}, \sum_{i=2}^{|G_j|} x_i U_i \right).$$

Therefore, only the authorized nodes who holds corresponding private key S_i can recover the session key. They cannot get any session key only from the broadcast messages. On the other hand, the personal public/private key pairs of nodes are computed by the ID-based PKI. The session keys are chosen by the GM. They are independent of one another. Therefore, The coalition of nodes cannot get any session key only by their public/private key pairs $\{(Q_1, S_1), \dots, (Q_n, S_n)\}$.

2. There is no threshold for the revocation in the proposed scheme. This means any number of nodes who are compromised or malicious can be revoked and although they work together they cannot work out the private key of any nonrevoked authorized node. In our scheme, the broadcast messages are computationally related to the authorized nodes' public/private key pairs. On the one hand, only nodes who hold the corresponding private keys can recover the session keys from the masked broadcast messages. On the other hand, even all the revoked unauthorized node work together they cannot get the private key of any nonrevoked authorized node due to the difficulty of solving the discrete logarithm problem.

3. The proposed scheme has the self-healing capability. It can also enable a node to recover from a single broadcast message all keys associated with sessions in which it belongs to the session group. It is a stronger self-healing key distribution scheme.

Without loss of generality, suppose $u_i \in G_t$ lost the broadcast message for a session t where $t < j$ and . It selects polynomial z_t from the broadcast message B_j and constructs a $|G_t| \times |G_t|$ matrix A_t as below:

$$A_t = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{|G_t|} \end{pmatrix},$$

where $a_k = (0, 0, \dots, 1, 0, \dots, 0)$ (note: the k th bit is 1 and all other bits are 0) and $k = 1, 2, \dots, |G_t|$.

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:

$$\begin{aligned} e(P_{pub}, r_t Q_{V_1}) &= e(P_{pub}, r_t(x_1 Q_i + x_2 Q_{V_2} + \dots + x_{|G_1|} Q_{V_{|G_1|}})) \\ &= e(P_{pub}, r_t x_1 Q_i) \cdot e(P_{pub}, r_t(x_2 Q_{V_2} + \dots + x_{|G_1|} Q_{V_{|G_1|}})) \\ &= e(r_t P, x_1 S_i) \cdot e(P_{pub}, x_2 r_t Q_{V_2} + \dots + x_{|G_1|} r_t Q_{V_{|G_1|}}) \\ &= e(U_1, x_1 S_i) \cdot e(P_{pub}, x_2 U_2 + \dots + x_{|G_1|} U_{|G_1|}). \end{aligned}$$

Finally, u_i recovers the lost session key

$$K_t = V_t \oplus H_2 \left(e(U_1, x_1 S_i) \cdot e \left(P_{pub}, \sum_{k=2}^{|G_1|} x_k U_k \right) \right).$$

6.1.2. Property 2: Forward security and backward security

We show our construction satisfies forward security and backward security described in Definition 4.2.

In this scheme, due to the special construction of broadcast messages, only the current authorized nodes can recover the session keys by using their private keys. As described before, any coalition of non-authorized nodes cannot derive the private key of any authorized node. Furthermore, session keys are independent to each other and according to the uniform distribution. All of these features imply forward security and backward security of our scheme.

6.1.3. Property 3: Collusion-free property

We show our construction is collusion-free for the new joined nodes and the revoked nodes described in Definition 4.3.

Our scheme is collusion-free for any coalition of non-authorized nodes, including the revoked nodes and new joined nodes. In our scheme, if one node 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 nodes can recover the session key. In addition, due to the difficulty of solving DLP, any coalition of non-authorized nodes cannot derive the private keys of authorized nodes from their public keys.

6.1.4. Property 4: Mutual-healing key distribution

We show that we securely realize mutual-healing property between neighboring nodes. More specifically, the realization satisfies the security requirements described in Definition 4.4.

1. It can be seen in the step of MUTUAL-HEALING REQUEST that any node $u_i \in U$, if it misses more than a fixed number of broadcast messages or the last broadcast message, it can generate and broadcast the requesting message to its neighbors. The broadcast message is composed of its identity ID_i , location l_i and the sequence number of the expected broadcast message t .
2. A false requesting node might send a request with a forged location within node u_j 's range. Since the false requesting node does not hold the LBK corresponding to the forged location, even it deceive u_j into believing it is in u_j 's range, it cannot recover the session key from the broadcast message that u_j sends. Therefore, it get any useful information from u_j . There are some false requesting nodes who might mount DoS (denial of service) attack by continuously sending bogus mutual-healing requests to allure legitimate nodes into endless verifying of such messages. Because the number of neighbors of any node is limited in reality, abnormally many mutual-healing requests are highly like an indicator of malicious attacks. If this situation happens, node u_j will discard the requesting message and stop assistance.
3. Upon receiving the reply, node u_i also first checks whether u_j is its neighbor. This check is the baseline defense against the

attack that adversaries surreptitiously tunnel authentication messages between u_i and a virtually non-neighboring node. Without the location check, u_i will falsely believe that the broadcast messages come from its neighbors. If the first check is true, then u_i checks whether the message received from u_j is effective. u_i can recover the requested broadcast message if the second check is true.

If a requester receives too many replies, it only decrypts a fixed number messages in order to save its limited resource and avoid exhausting-resource attack. Furthermore, we assume that there are efficient mechanisms available for authorized nodes to report such an abnormality to the sink.

6.2. The analysis of efficiency

Different from the existing papers, we take advantage of a bilinear pairings-based broadcast encryption to design the self-healing procedures in the proposed mutual-healing key distribution scheme. In this section, we first analyze the efficiency for the self-healing procedures followed by the one for mutual-healing procedures.

In terms of storage overhead, each node only stores its public/private key pair and GM's public key P_{pub} . Therefore, the storage overhead for end nodes is a constant. Furthermore, the private key has nothing to do with the number of revoked nodes and can be reused as long as it is not disclosed. In addition, our scheme enables a node 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 nodes and thus can perform more complex computation. We elaborate on analyzing the computation overhead at nodes. 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 (Barreto et al., 2002; Galbraith et al., 2002), 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 $B_j = \{z_1, \dots, z_j\}$. 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 $\sum_{i=1}^j |G_i| \log q + j \log q$. $\log q$ is the size of session key.

The mutual-healing procedures achieves all the security requirements and is efficient. Each node only has to store another pair of LBK and its location in order to achieve mutual-healing property. Therefore, the storage overhead is still a constant. In terms of communication overhead, only two interactions are involved. In the first step, the broadcast message is composed of the requester's identity, location and the sequence number of the expected broadcast message. Furthermore, the message is broadcasted within one-hop communication range. The communication overhead of this step is very small. The second interaction may include one or several unicasts. The unicast message is composed of the responder's identity, location, and encrypted the broadcast message. Therefore, the size of these unicast messages is a constant. The computation overhead at the responder includes generating a shared key and encrypting the requested message

and the computation overhead at the requester includes generating a shared key and decrypting the requested message. Because mutual-healing happens not very often, the computation overhead would not consume too much resource of the involved nodes.

7. Conclusion and future research

A mutual-healing key distribution scheme using bilinear pairings is proposed in this paper. Security model and formal definition for mutual-healing key distribution were discussed. The proposed new scheme achieves several desirable features. The storage overhead for each node is a constant. The scheme is collusion-free for any coalition of non-authorized nodes. Each authorized node's private key has nothing to do with the number of revoked nodes 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 nodes are revoked. In addition, our scheme enables a node to recover from a single broadcast message all keys associated with sessions in which it belongs to the session group.

The proposed mutual-healing scheme relies on identity and location-based keys. This implies that the proposed scheme can only be used over the wireless networks where nodes are stable. It is not trivial to realize mutual-healing in mobile wireless networks. In fact, the mutual-healing mechanism is more useful in mobile wireless networks because mobile wireless networks have lower network connectivity than stable wireless networks. Therefore, it is significant to investigate new methods to realize the mutual-healing feature in mobile wireless networks.

Acknowledgement

The work is financially supported by the Australia Research Council (ARC) Projects with project IDs LP100200693, LP100200538, LP1000100404 and DP0985838.

References

- Balenson D, McGrew D, Sherman A. Key management for large dynamic groups: one-way function trees and amortized initialization, IRTF SMUG Meeting, March 15, 1999.
- Barreto PSLM, Kim HY, Lynn B, Scott M. Efficient algorithms for pairing-based cryptosystems. In: *Advances in cryptology—crypto'02*. Lecture notes in computer science, vol. 2442; 2002. p. 354–68.
- Blundo C, D'Arco P, Santis A, Listo M. Design of self-healing key distribution schemes. *Design Codes and Cryptography* 2004;32:15–44.
- Boneh D, Franklin M. Identity based encryption from the weil pairing. In: *Advanced in cryptology—CRYPTO'01*; 2001. p. 213–29.
- Capkun S, Hubaux J-P. Secure positioning of wireless devices with application to sensor networks. In: *Proceedings of IEEE INFOCOM*. Miami, Florida; March 2005. p. 1917–28.
- Du X, Wang Y, Ge J, Wang Y. An ID-based broadcast encryption scheme for key distribution. *IEEE Transactions on Broadcasting* 2005;51(2):264–6.
- Dutta R, Mukhopadhyay S. Improved self-healing key distribution with revocation in wireless sensor network. In: *Wireless communications and networking conference*; 2007. p. 2963–8.
- Dutta R, Chang EC, Mukhopadhyay S. Efficient self-healing key distribution with revocation for wireless sensor networks using one way key chains. In: *ACNS 2007*, Lecture notes in computer science, vol. 4521; 2007. p. 385–400.
- Eltoweissy M, Younis M, Ghumman K. Lightweight key management for wireless sensor networks. In: *IEEE international conference on performance, computing, and communications*; April 2004. p. 813–8.
- Eltoweissy M, Heydari M, Morales L, Sudborough H. Combinatorial optimization for group key management. *Journal of Network and System Management* 2004b;12(1):33–50.
- Fiat A, Tessa T. Dynamic traitor tracing. *Journal of Cryptology* 2001;14:211–23.
- Galbraith SD, Harrison K, Soldera D. Implementing the Tate pairing. In: *Proceedings of the 5th international symposium on algorithmic number theory*. Lecture notes in computer science, vol. 2369; 2002. p. 324–37.
- Halevy D, Shamir A. The LSD broadcast encryption scheme. In: *Advances in cryptology—crypto'02*. Lecture notes in computer science, vol. 2442; 2002. p. 47–60.
- Han F, Hu J, Yu X, Wang Y. Fingerprint images encryption via multi-scroll chaotic attractors. *Applied Mathematics and Computation* 2007;185(2):931–9.
- Han S, Tian B, He M, Chang E. Efficient threshold self-healing key distribution with sponsorship for infrastructureless wireless networks. *IEEE Transactions on Wireless Communications* 2009;8(4).
- Han S, Wang J, Liu W. An efficient identity-based group signature scheme over elliptic curves. In: *European conference on universal multiservice and networks*. Lecture notes in computer science, vol. 3262. Springer-Verlag; 2004. p. 417–29.
- Hong D, Kang JS. An efficient key distribution scheme with self-healing property. *IEEE Communication Letters* 2005;9(8).
- Hu J, Chen HH, Hou TW. A hybrid public key infrastructure solution (HPKI) for HIPAA privacy/security regulations. *Computer Standards & Interfaces* 2010;32(5–6):274–80.
- Hu J, Han F. A pixel-based scrambling scheme for digital medical images protection. *Journal of Network and Computer Applications* 2009;32(4):788–94.
- Jiang Y, Lin C, Shi M, Shen X. Self-healing group key distribution with time-limited node revocation for wireless sensor networks. *Ad hoc Networks* 2007;5:14–23.
- Kausar F, Hussain S, Park JH, Masood A. Secure group communication with self-healing and rekeying in wireless sensor networks. In: *Proceedings of the third international conference, MSN 2007*; 2007. p. 737–48.
- Ku W, Chen S. An improved key management scheme for large dynamic groups using one-way function trees. In: *International conference on parallel processing workshops '03*. October 2003. p. 391–6.
- Liu D, Ning P, Sun K. Efficient self-healing key distribution with revocation capability. In: *Proceeding of the 10th ACM conference on computer, communications and security*; 2003.
- Lu K, Qian Y, Hu J. A framework for distributed key management schemes in heterogeneous wireless sensor networks. In: *The 25th IEEE international conference on performance, computing, and communications, USA*; April 2006. p. 513–9.
- Moharrum M, Eltoweissy M, Mukkamala R. Dynamic combinatorial key management scheme for sensor networks. *Wireless Communications and Mobile Computing* 2006;6(7):1017–35.
- More SM, Malkin M, Staddon J, Balfanz D. Sliding window self-healing key distribution with revocation. In: *ACM workshop on survivable and self-regenerative systems*; 2003. p. 82–90.
- Muhammad JB, Ali M. Self-healing group key distribution. *International Journal of Network Security* 2005;1(2):110–7.
- Naor D, Naor M, Lotspiech J. Revocation and tracing schemes for stateless receivers. In: *Advances in cryptology—crypto'01*. Lecture notes in computer science, vol. 2139; 2001. p. 41–62.
- Naor M, Pinkas B. Efficient trace and revoke schemes. In: *Financial cryptography'2000*. Lecture notes in computer science, vol. 1962; 2000. p. 1–21.
- Sáez G. On threshold self-healing key distribution schemes. In: *Cryptography and coding*, Lecture notes in computer science, vol. 3796; 2005a. p. 340–54.
- Sáez G. Self-healing key distribution schemes with sponsorship. In: *International federation for information processing, IFIP'05*, Lecture notes in computer science, vol. 3677; 2005b. p. 22–31.
- Shamir A. Identity-based cryptosystems and signature scheme. *Proceedings of Crypto 1984*;84:47–53.
- Staddon J, Miner S, Franklin M, Balfanz D, Malkin M, Dean D. Self-healing key distribution with revocation. In: *Proceedings of IEEE symposium on security and privacy*; 2002. p. 224–40.
- Sufi F, Han F, Khalil I, Hu J. A chaos based encryption technique to protect ECG packets for time critical telecardiology applications. *Journal of Security and Communication Networks* (2010), doi:10.1002/sec.226.
- Wong CK, Gouda MG, Lan SS. Secure group communication using key graphs. *IEEE/ACM Transactions on Networking* 2000;8(1):16–30.
- Xi K, Ahmad T, Han F, Hu J. A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. *Journal of Security and Communication Networks* (2010), doi:10.1002/sec.225.
- Zhang Y, Liu W, Fang Y. Secure localization in wireless sensor networks. In: *Proceedings of IEEE MILCOM*; 2005.
- Zhang Y, Liu W, Lou W, Fang Y. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 2006;24(2):247–60.
- Zou X, Dai Y. Robust and stateless self-healing group key management scheme. *International Conference on Communication Technology ICCT '06 2006*:1–4.