Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 821486, 7 pages
doi:10.1155/2012/821486

Research Article

Self-Healing Key-Distribution Scheme with Collusion Attack Resistance Based on One-Way Key Chains and Secret Sharing in Wireless Sensor Networks

Dong Jiao,[1] Mingchu Li,[1] Yan Yu,[2] and Jinping Ou[3]

[1] *School of Software Technology, Dalian University of Technology, Dalian 116621, China*
[2] *School of Electronic Science and Technology, Dalian University of Technology, Dalian 116024, China*
[3] *School of Civil Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Yan Yu, yuyan@dlut.edu.cn

Received 14 June 2012; Accepted 21 August 2012

Academic Editor: Leonardo B. Oliveira

In wireless sensor networks, self-healing key-distribution schemes are used to ensure that, even if the message packets that are broadcast in some sessions get lost, the group nodes can still recover the lost session keys simply by using their personal secret keys and broadcast messages that have been received without requesting additional transmissions from the group manager. These schemes reduce network traffic, decrease the group manager's workload, and lower the risk of node exposure through traffic analysis. However, most existing schemes have many deficiencies, such as high overhead for storage and communication and collusion attacks. In this paper, we have proposed a modified, self-healing, key-distribution scheme based on one-way key chains and secret sharing. Our scheme has the properties of constant storage, lower communication overhead, long lifespan, forward secrecy, backward secrecy, and resistance to collusion attacks.

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes with limited power, storage, computation, and communication capabilities. WSNs have wide applications in military operations and scientific exploration [1, 2] in which there may be inadequate support by the infrastructure of the network, allowing adversaries to potentially intercept, modify, or partially interrupt communication. In such applications, security is a critical concern. In addition, in some deployment scenarios, sensor nodes must operate under adversarial conditions. Therefore, determining how to distribute group session keys for secure communication to a large dynamic group over an unreliable network is a serious issue. In WSNs, packet loss occurs frequently. Messages that are broadcast by the group manager (base station) might never reach some authorized nodes (sensor nodes). So, it is important to guarantee the reliable transmission of information for updating the group's session keys to the authorized nodes. An easy solution is requesting retransmission, but requesting retransmission increases the overhead associated with communication incurs a high risk of revealing the nodes' physical locations, which is not acceptable in some high-security environments.

A self-healing, key-distribution scheme is proposed to solve the problem described above. The main concept of self-healing, key-distribution schemes is that, even if the message packets that are broadcast in some sessions get lost, the group nodes can still recover the lost session keys simply by using their personal secret keys and broadcast messages that have been received without requesting additional transmissions from the group manager. These schemes reduce network traffic, decrease the group manager's workload, and lower the risk of node exposure through traffic analysis. Figure 1 shows network topology in a key distribution scheme under adversarial conditions.

In 2002, Staddon et al. [3] proposed the first self-healing, key-distribution scheme with revocation using secret sharing [4]. However, Staddon et al.'s schemes incur high overhead for storage and communication. Later, several other schemes

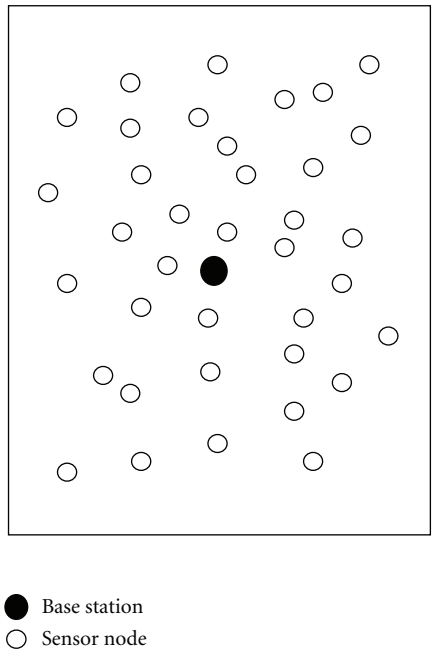

Figure 1: Network topology in a key-distribution scheme.

were proposed [5–9] based on Staddon et al.'s schemes. Liu et al. (2003) generalized the definitions and security notions and proposed a new scheme that significantly decreased the overhead for communication by introducing a novel, personal key-distribution [5]. Blundo et al. [10] showed that the first scheme in [3] is insecure. An adversary could recover the group's session key with just broadcast messages. In [11], Dutta et al. proposed two self-healing, key-distribution schemes with revocation that were secure, but they did not consider collusion attacks. In [12], Dutta et al. proposed a new self-healing key-distribution scheme with a constant storage overhead by using only one secret polynomial. But Xu and He's scheme [13] and Du and He's scheme [14] showed that the scheme in [12] was insecure. Any user can recover the manager's secret polynomial, which should not been known by any user. Xu and He (2009) proposed two schemes in [13], one of which improved the scheme in [12] by using an access polynomial instead of the revocation polynomial with the other, which was based on the scheme in [11], still using an access polynomial. But neither of the two schemes proposed in [13] considered collusion attacks between the revoked user and the newly-joined user. In [14], Du and He proposed a new self-healing, key-distribution scheme with revocation and resistance to collusion attacks. However, Bao and Zhang (2011) showed that the scheme in [14] was vulnerable to collusion attacks [15]. A revoked user and a newly-joined user easily could recover the session keys that they should not know. However, Bao and Zhang (2011) used m secret polynomials for m sessions and an access polynomial in the broadcast phase, which resulted in an excessive communication overhead.

In this paper, we propose a self-healing key-distribution scheme for WSNs based on one-way key chains and secret sharing. In our scheme, only one secret polynomial is used in all sessions, and modified access polynomials are used, which produces a lower communication overhead. Also, our scheme can resist collusion attacks between a newly-joined user and a revoked user.

The rest of the paper is organized as follows. In Section 2, the security model is presented and Bao and Zhang's scheme [15] is reviewed briefly. In Section 3, our modified, self-healing, key-distribution scheme is proposed. Then, we discuss the security and performance of our scheme in Section 4. Our conclusions are presented in Section 5.

2. Preliminaries

In this section, we briefly introduce Bao and Zhang's scheme [15] and the security definitions. The following notations will be used in the rest of the paper.

U is the set of all users (sensor nodes) in wireless sensor networks.

U_i is the user i in U.

u_i is the identity of U_i.

GM is the group manager (base station).

n is the total number of users in U.

m is the total number of sessions.

t is the maximum number of compromised users in all sessions.

p is a large prime modulus, where $2^{799} < p < 2^{800}$.

q is a large prime divisor of $p-1$, where $2^{159} < q < 2^{160}$ and $q^2 \mid (p-1)$.

$\{g_i\}_{i=1}^{m}$ are m generators with order q in GF(p).

$f(x) \in F_q[x]$ is the secret polynomial of degree t generated by GM.

S_i is the personal secret of user U_i.

B_j is the broadcast message generated by GM for session j.

β_j is the self-healing key generated by GM for session j.

K_j is the session key in session j generated by GM.

K_0 is the initial key seed generated by GM.

R_j is the set of all revoked users in and before session j.

G_j is the set of nonrevoked users in session j.

H_1, H_2 are two cryptographically secure, one-way functions, and $H_1 : \{0,1\}^* \to F_p, H_2 : \{0,1\}^* \to F_p$.

$D_k(\cdot)$ is a symmetric decryption function.

$E_k(\cdot)$ is a symmetric encryption function.

2.1. Security Model

Definition 1 (self-healing key-distribution with t-revocation capability [11]). A key-distribution scheme is a self-healing, key-distribution scheme with t-revocation capability if the following conditions are true.

(a) For any nonrevoked user U_i in session j, the group session key K_j is efficiently determined by the broadcast message B_j and the personal secret S_i.

(b) The group session key K_j cannot be determined by what the non-revoked users learn from B_j or their own personal secret alone.

(c) t-revocation capability: for each session j, let R_j denote a set of revoked users in and before session j, where $|R_j| \leq t$, the group manager can generate a broadcast message B_j such that all the revoked users in R_j cannot recover the group session key K_j.

(d) Self-healing property: any U_i who joins in or before session j_1 and is not revoked before session j_2 ($1 \leq j_1 < j_2$) can recover all the keys $K_j (j_1 \leq j \leq j_2)$ by the broadcast messages B_{j_1}, B_{j_2}, and the personal secret S_i.

Definition 2 (t-wise forward secrecy [11]). Let $R_j \subseteq U$ denote a set of all revoked users in and before session j, where $|R_j| \leq t$. A key-distribution scheme guarantees forward secrecy if the members in R_j together cannot get any information about K_j, even with the knowledge of group session keys before session j.

Definition 3 (t-wise backward secrecy [11]). Let $J_j \subseteq U$ denote a set of users who join the group after session j, where $|J_j| \leq t$. A key-distribution scheme guarantees backward secrecy if the members in J_j together cannot get any information about K_j, even with the knowledge of group session keys after session j.

Definition 4 (resistance to the collusion attack [16]). Let $R \subseteq U$ denote a set of all revoked users in and before session j_1 and let $J \subseteq U$ denote a set of users who join the group after session j_2, where $1 \leq j_1 < j_2$ and $|R \cup J| \leq t$. A key-distribution scheme with resistance to collusion attacks means that, even if all users in R and J cooperate, they cannot get any information about keys K_j, for all $j_1 < j < j_2$.

2.2. Review of Bao and Zhang's Scheme. In [15], Bao and Zhang proposed an improved key-distribution scheme for [14] that included resistance to collusion attacks. The scheme is divided into the four phases described below.

Phase 1: Setup. First, the GM randomly chooses m polynomials $f_1(x), \ldots, f_m(x) \in F_p[x]$, each of degree t.

Second, the GM randomly chooses numbers $\alpha_1, \ldots, \alpha_m \in F_p$ for each session.

Third, the GM chooses a random secret value $t_i \in F_p$ for user U_i and the t_i values are different from each other. Then, the GM sends the personal secret $S_i = \{t_i, \alpha_{j'}, f_{j'}(t_i)\}$ to user U_i in a secure manner. (The term j' denotes the session number when the user joins the group and $\alpha_{j'} \in \{\alpha_1, \ldots, \alpha_m\}$.)

Then, the GM randomly chooses a prime, initial key seed $K_0 \in F_p$, which is kept secret and m numbers $\{\beta_j\}_{j=1}^{m} \in F_p$ as the self-healing keys.

The GM computes a key seed and corresponding key chain for each session using two one-way hash functions H_1, H_2 and m numbers $\{\beta_j\}_{j=1}^{m}$. For $1 \leq j \leq m$, the key seed of session j is computed as shown:

$$K_j^0 = H_1\left(K_{j-1}, \beta_j\right). \tag{1}$$

And the key chain of session j of length j is computed as shown:

$$K_j^{j-1} = H_2\left(K_j^{j-2}\right) = H_2^{j-1}\left(K_j^0\right), \tag{2}$$

where $H_2^i()$ means applying the hash operation i times. Then, $\{K_j^0, K_j^1, \ldots, K_j^{j-1}\}$ is the key chain of session j, and the group session key in session j is $K_j = K_j^{j-1}$.

Phase 2: Broadcast. Let $\mathcal{U}_{\text{act}_j} = \{U_{\text{act}_1}, \ldots, U_{\text{act}_{a_j}}\}$ be the set of all active users for session j, where a_j is the number of active users in session j. Let $\mathcal{T}_{\text{act}_j} = \{t_{\text{act}_1}, \ldots, t_{\text{act}_{a_j}}\}$ be the set of all active users' secret values in session j. Then, the GM generates $\{G_j^1, G_j^2, \ldots, G_j^j\}$ of size j as a masking key sequence for session j by applying XOR on both $\alpha_{j'}$, and every key forms the key chain of session j, where

$$G_j^{j'} = K_j^{j'-1} \oplus \alpha_{j'}. \tag{3}$$

In session j, the GM broadcasts the following message:

$$\begin{aligned} B_j = &\left\{Z_j^{j'}(x) = A_j^{j'}(x) G_j^{j'} + f_{j'}(x)\right\}_{j'=1}^{j} \\ &\cup \left\{E_{K_j^0}(\beta_1), E_{K_j^1}(\beta_2), \ldots, E_{K_j^{j-1}}\left(\beta_j\right)\right\}, \end{aligned} \tag{4}$$

where $A_j^{j'}(x) = (S_j^{j'} \cdot x - T_j) \prod_{i=1}^{a_{j'}} (x - t_{\text{act}_i}) + 1$ is an access polynomial. When an active user U_{act_i} receives the broadcast message B_j of session j, U_{act_i} can evaluate $A_j^{j'}(t_{\text{act}_i}) = 1$ by using its secret value t_{act_i}, where j' denotes that U_{act_i} has joined the group in session j'. However, a revoked user can only evaluate a random value.

Phase 3: Group Session Key and Self-Healing Key Recovery. When a nonrevoked user U_i in session j, who joins in the group in session j', receives the broadcast message B_j of session j, U_i can recover the group session key K_j as follows.

First, U_i computes $G_j^{j'} = z_j^{j'}(t_i) - f_{j'}(t_i)$ from (4), where $f_{j'}(t_i) \in s_i$ and $A_j^{j'}(t_i) = 1$. Then, U_i evaluates $K_j^{j'-1} = G_j^{j'} \oplus \alpha_{j'}$ from (3), where $\alpha_{j'}$ is secret value of U_i.

Then U_i can compute all the future keys $\{K_j^{j'}, K_j^{j'+1}, \ldots, K_j^{j-1}\}$ in the key chain of session j by using the one-way hash function $H_2()$. The group session key of session j is $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$.

Then, U_i can decrypt $\{E_{K_j^{j'-1}}(\beta_{j'}), E_{K_j^{j'}}(\beta_{j'+1}),\ldots, E_{K_j^{j-1}}(\beta_j)\}$ by using the corresponding keys $\{K_j^{j'-1}, K_j^{j'},\ldots, K_j^{j-1}\}$ to get the corresponding self-healing keys $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$.

However, a revoked user can recover neither the group session key nor the self-healing keys of session j, since $A_j^{j'}(t_i)$ is a random number for any user $U_i \in R_j$.

Phase 4: Add Group Members. If a new user wants to join the group in session j, the GM chooses a never-used identity $v \in \{1,2,\ldots,n\}$ for U_v. Then, the GM selects a random secret value $t_v \in F_p$ and sends the personal secret key $S_v = \{t_v, \alpha_j, f_j(t_v)\}$ to U_v using RSA algorithm.

3. The Proposed Scheme

In this section, we propose an improved version of Bao and Zhang's scheme [15] using secret sharing. In our scheme, we use only one secret polynomial and modified access polynomials, which lower the communication overhead. Our scheme is divided into four phases, as follows.

Phase 1: Initiation. First, the GM creates a polynomial $f(x) \in F_q[x]$ of degree t as the secret polynomial. Then, the GM chooses $\{g_i\}_{i=1}^m$ as m generators with order q in GF(p) and $\{\alpha_i\}_{i=1}^m$ for each session.

Second, the GM selects a unique identity $u_i \in F_q$ for user U_i and sends $S_i = \{u_i, \alpha_{j'}, f(u_i) \bmod q\}$ to user U_i for $i = 1,\ldots,n$ as personal secret keys via a secure communication channel, where j' denotes the session number when the user joined the group. For example, user U_r, who joins the group in session 1, will receive $S_r = \{u_r, \alpha_1, f(u_r) \bmod q\}$.

Then, the GM randomly chooses a prime initial key seed $K_0 \in F_p$, which is kept secret, and m numbers $\{\beta_j\}_{j=1}^m \in F_p$ as the self-healing keys.

In our scheme, as in Du-He's scheme [14], we still use key chains. The GM computes a key seed and corresponding key chain for each session using two one-way hash functions H_1, H_2 and m numbers $\{\beta_j\}_{j=1}^m$. For $1 \le j \le m$, the key seed of session j is computed by (1): $K_j^0 = H_1(K_{j-1}, \beta_j)$.

And the key chain of session j of length j is computed by (2): $K_j^{j-1} = H_2(K_j^{j-2}) = H_2^{j-1}(K_j^0)$, where $H_2^i()$ means applying the hash operation i times. Then, $\{K_j^0, K_j^1,\ldots, K_j^{j-1}\}$ is the key chain of session j and the group session key in session j is $K_j = K_j^{j-1}$.

Phase 2: Broadcast. Assume that $R_j \subseteq U$ and $|R_j| \le t$ are the sets of all revoked users in and before session j, respectively. Let G_j be the set of all nonrevoked users in session j. In session j, the GM chooses a set of nonzero indices $X_j = \{x_{j,1}, x_{j,2},\ldots, x_{j,t}\}$ such that $I_{R_j} \subseteq X_j$, but $W_j \cap I_{G_j} = \emptyset$, where I_{R_j} denotes the set of indices of the users in R_j, and I_{G_j} represents the indices of users in G_j. Let $\mathcal{U}_j^{j'} = \{u_{j',1}, u_{j',2},\ldots, u_{j',n_{j'}}\}$ be the set of indices of the users who join the group in session j' and are still active in session j, where $n_{j'}$ is the number of users of the set and $1 \le j' \le j$ and $G_j = \bigcup_{j'=1}^{j} \mathcal{U}_j^{j'}$. Then, the GM computes a sequence $\{Z_j^{j'}\}_{j'=1}^{j}$ using the key chain of session j as shown:

$$Z_j^{j'}(x) = \left(K_j^{j'-1} \oplus \alpha_{j'}\right) + g_j{}^{f(0)} A_j^{j'}(x) \bmod p, \tag{5}$$

where $A_j^{j'}(x) = (Y_j^{j'} x - T_j)\prod_{i=1}^{n_{j'}}(x - u_{j',i}) + 1$ is a modified-access polynomial. $Y_j^{j'}$ and T_j are randomly selected by the GM in F_p, such that $Y_j^{j'}/T_j$ is different from all users' indices. When an active user U_i receives the broadcast message B_j of session j, U_i can evaluate $A_j^{j'}(u_i) = 1$ by using its secret identity value u_i, where j' denotes that U_i joined the group in session j'. However, a revoked user or an active user who does not join in the group in session j' only can evaluate a random value.

Then, the GM broadcasts the following message B_j:

$$\begin{aligned} B_j = g_j \cup \left\{x_{j,i}, g_j^{f(x_{j,i})}\right\}_{i=1}^{t} \cup \left\{Z_j^{j'}(x)\right\}_{j'=1}^{j} \\ \cup \left\{E_{K_j^0}(\beta_1), E_{K_j^1}(\beta_2),\ldots, E_{K_j^{j-1}}\left(\beta_j\right)\right\}. \end{aligned} \tag{6}$$

Phase 3: Group Session Key Recovery and Self-Healing Key Recovery. When a non-revoked user U_i, who joins the group in session j', receives the broadcast message B_j of session j, he or she can recover $g_j^{f(0)}$ by Lagrange's interpolation using B_j and her or his personal secret keys as following:

$$g_j^{f(0)} = \prod_{l=0}^{t}\left(g_j^{f(x_{j,l})}\right)^{w_l} \bmod p, \tag{7}$$

where

$$w_l = \prod_{\substack{k=0 \\ k \ne l}}^{t} \frac{-x_{j,k}}{x_{j,l} - x_{j,k}}. \tag{8}$$

With $x_{j,0} = u_i$, user U_i can recover $g_j^{f(0)}$, then he or she can recover $K_j^{j'-1}$ by (5) with $A_j^{j'}(x) = 1$, as follows:

$$K_j^{j'-1} = \left(Z_j^{j'}(u_i) - g_j^{f(0)} \bmod p\right) \oplus \alpha_{j'}, \tag{9}$$

where j' denotes the session number when U_i joined the group, and $\alpha_{j'}$ is the secret of user U_i distributed by the GM when he or she joins the group in session j'.

Then, U_i computes the group session key of session j as $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$.

User U_i also can recover the self-healing key $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$ using $K_j^{j'-1}$ and B_j. First, U_i computes all the keys $\{K_j^{j'}, K_j^{j'+1},\ldots, K_j^{j-1}\}$ in the key chain of session j by using the one-way hash function $H_2()$. Then, U_i can decrypt $\{E_{K_j^{j'-1}}(\beta_{j'}), E_{K_j^{j'}}(\beta_{j'+1}),\ldots, E_{K_j^{j-1}}(\beta_j)\}$ by using the keys $\{K_j^{j'-1}, K_j^{j'},\ldots, K_j^{j-1}\}$ to get $\{\beta_{j'}, \beta_{j'+1},\ldots,\beta_j\}$. Then,

the user with session key $K_{j'}$ can recover all session keys between session j' to j based on (1) and (2).

A user who was revoked in session j cannot recover the current group session key or the self-healing key even with the B_j, since he or she cannot recover $g_j^{f(0)}$ based on Lagrange's interpolation.

Phase 4: New User Added. If a user U_x wishes to be added to the group in session j, GM chooses a unique and never-used identity u_x for U_x and sends the secret $S_x = \{u_x, \alpha_j, f(u_x) \bmod q\}$ to U_x using the RSA algorithm.

4. Security and Performance Analyses

In this section, we show that our proposed scheme has self-healing property, forward security, backward security, and resistance to collusion attacks. Compared with Bao and Zhang's scheme [15], our scheme has lower communication overhead.

4.1. Self-Healing Property. Assume that U_i, who join the group in session j', are active in session j_1 and session j_2, where $1 \le j' \le j_1 \le j_2$. And U_i receive session-key broadcast messages B_{j_1} and B_{j_2} but lose the session key broadcast message B_j, where $j_1 < j < j_2$. Users U_i can still recover all the lost session keys K_j for $j_1 < j < j_2$ as follows.

(1) When the broadcast message B_{j_1} is received, U_i can recover $g_{j_1}^{f(0)}$ using their personal secrets by (7) and (8). Then, because U_i are active users in session j_1, U_i can recover $K_{j_1}^{j'-1}$ by (5) and (9), where $A_{j_1}^{j'}(u_i) = 1$. Then, U_i compute the group session key of session j as $K_{j_1} = K_{j_1}^{j_1-1} = H_2^{j_1-j'}(K_{j_1}^{j'-1})$.

(2) When the broadcast message B_{j_2} is received, U_i can recover $g_{j_2}^{f(0)}$ using their personal secrets by (7) and (8). Then, because U_i are active users in session j_2, U_i can recover $K_{j_2}^{j'-1}$ by (5) and (9), where $A_{j_2}^{j'}(u_i) = 1$. U_i compute all the keys $\{K_{j_2}^{j'}, K_{j_2}^{j'+1}, \ldots, K_{j_2}^{j_2-1}\}$ in the key chain of session j_2 by using the one-way hash function $H_2()$. Then, U_i can recover $\{\beta_{j'}, \beta_{j'+1}, \ldots, \beta_{j_1}, \ldots, \beta_{j_2}\}$ using the keys $\{K_{j_2}^{j'-1}, K_{j_2}^{j'}, \ldots, K_{j_2}^{j_2-1}\}$ by decryption $\{E_{K_{j_2}^{j'-1}}(\beta_{j'}), E_{K_{j_2}^{j'}}(\beta_{j'+1}), \ldots, E_{K_{j_2}^{j_2-1}}(\beta_{j_2})\}$.

(3) With K_{j_1} and $\{\beta_{j_1}, \ldots, \beta_{j_2}\}$, U_i can recover all session keys K_j for $j_1 < j < j_2$ by (1) and (2).

Therefore, our scheme achieves the self-healing property.

4.2. Forward Secrecy. Let $R_j \subseteq U$ and $|R_j| \le t$ be the set of all revoked users in and before session j, respectively. Then, we show that the coalition R_j cannot get any information about the current session key K_j, even with the previous group session keys before session j. To recover the session key $K_j = K_j^{j-1} = H_2^{j-j'}(K_j^{j'-1})$, user $U_i \in R_j$ must recover $K_j^{j'-1} = (Z_j^{j'}(u_i) - g_j^{f(0)} A_j^{j'}(u_i)) \oplus \alpha_{j'}$ by (5), where j' denotes the session number when U_i joined the group. But for revoked users U_i, $A_j^{j'}(u_i)$ is a random value that is not known by U_i. Moreover, U_i cannot recover $g_j^{f(0)}$ even with all of the information of all revoked users, because, according to Lagrange's interpolation, to recover $g_j^{f(0)}$, U_i must know at least $(t+1)$ number pairs, such as $(x_i, g_j^{f(x_i)})$, where $(x_i, f(x_i))$ is a point on $f(x)$. Since the size of the coalition R_j is, at most, t, the coalition R_j cannot recover $g_j^{f(0)}$. In [17], Harn showed that U_i may be able to recover $g_j^{f(0)}$ with the previous $g_{j_1}^{f(0)}$ and $g_{j_2}^{f(0)}$ when $g_j = g_{j_1} g_{j_2}$. But in our scheme, the probability is 2^{-160}, which is extremely low and can be almost neglected. After all, the coalition R_j cannot get any information about the current session key K_j.

The above analysis shows that our scheme is forward secure.

4.3. Backward Secrecy. Let $J_j \subseteq U$, where $|J_j| \le t$, be the set of all users who join the group after session j. We will show that the coalition J_j cannot get any information about any previous session key K_{j_1} for $j_1 \le j$, even with the knowledge of group keys after session j.

Users in J_j can get only the session keys $\{K_{j+1}, K_{j+2}, \ldots\}$ and self-healing keys $\{\beta_{j+1}, \beta_{j+2}, \ldots\}$. Without loss of generality, one can get $K_{j+1} = H_2^j(K_{j+1}^0)$ and $K_{j+1}^0 = H_1(K_j, \beta_{j+1})$ by (1) and (2), where $H_1(\cdot)$ and $H_2(\cdot)$ are two one-way hash functions. It is computationally infeasible for any user in J_j to compute any previous session key K_{j_1} with keys $\{K_{j+1}, K_{j+2}, \ldots\}$ and self-healing keys $\{\beta_{j+1}, \beta_{j+2}, \ldots\}$ for $j_1 \le j$.

However, users in J_j could attempt to recover the previous session keys by their personal secret keys and the previous broadcast messages. However, by (5) and (6), it is evident that the previous broadcast messages do not have the equations for users in J_j. So users in J_j cannot recover the previous session keys.

The above analysis shows that our scheme is backward secure.

4.4. Resistance to Collusion Attack. Let $R_{j_1} \subseteq U$ be the set of all revoked users in and before session $j_1 + 1$ and let $J_{j_2} \subseteq U$ be the set of all users who join the group from session j_2. We will show that collusion of R_{j_1} and J_{j_2} cannot recover any session key K_j $(j_1 < j < j_2)$ with their personal secret keys and the broadcast message B_{j_1} and B_{j_2}.

To recover session key K_j $(j_1 < j < j_2)$, $R_{j_1} \cup J_{j_2}$ must recover the self-healing keys $\beta_{j_1+1}, \beta_{j_1+2}, \ldots, \beta_{j_2-1}$. Without loss of generality, assume that U_a joins the group in session j_1 and that U_b joins the group in session j_2. For the equation, $Z_{j_2}^{j_1}(x) = (K_{j_2}^{j_1-1} \oplus \alpha_{j_1}) + g_{j_2}^{f(0)} A_{j_2}^{j_1}(x)$, since user U_a, who joined the group in session j_1, is not active in session j_2, $A_{j_2}^{j_1}(u_a)$ is a random number. Then, U_a cannot recover $K_{j_2}^{j_1-1}$ even with α_{j_1} and $g_{j_2}^{f(0)}$ provided by user U_b. Therefore, users in $R_{j_1} \cup J_{j_2}$

Table 1: Comparison among different schemes.

Schemes	Storage overhead	Communication overhead	Long lifespan	Forward secrecy	Backward secrecy	Collusion attack resistance
Staddon et al.'s scheme 3 [3]	$(m-j-1)^2 \log p$	$(mt^2+2mt+m+t)\log p$	No	Yes	Yes	Yes
Liu et al.'s scheme 3 [5]	$(m-j+1)\log p$	$(2tj+j)\log p$	No	Yes	Yes	Yes
Dutta et al.'s scheme [11]	$3\log p$	$(t+1+j)\log p$	Yes	No	No	No
Xu and He's scheme 1 [13]	$4\log p$	$\left(\max\left\{t+j+1, a_j+j+2\right\}\right)\log p$	Yes	Yes	Yes	No
Du and He's scheme [14]	$(m-j+2)\log p$	$[(t+1)j+j]\log p$	No	Yes	Yes	Yes
Bao and Zhang's scheme [15]	$3\log p$	$\left(\max\left\{a_j+2, t+1\right\}\cdot j+j\right)\log p$	Yes	Yes	Yes	Yes
Our scheme	$3\log p$	$\left(a_j+3j+2t+1\right)\log p$	Yes	Yes	Yes	Yes

cannot recover the self-healing keys $\beta_{j_1+1}, \beta_{j_1+2}, \ldots, \beta_{j_2-1}$ and session key K_j $(j_1 < j < j_2)$.

The above analysis shows that our scheme can resist collusion attacks.

4.5. Constant Storage Overhead and Lower Communication Overhead. Our scheme has a constant storage overhead, which comes only from the user's personal secret keys $\{u_i, \alpha_{j'}, f(u_i) \bmod q\}$. So, the storage overhead is $(3\log p)$ bits.

In our scheme, we use only one secret polynomial and modified access polynomials, which lower the communication overhead. The communication overhead is $(a_j + 3j + 2t + 1)\log p$, where t is the maximum number of revoked users, and a_j is the number of active users in session j. Table 1 shows the comparison among the different schemes.

5. Conclusions

In this paper, we proposed a modified and an improved version of Bao and Zhang's scheme. Our scheme uses only one secret polynomial and modified access polynomials, which achieve a lower communication overhead. In addition, our scheme has the properties of constant storage, long lifespan, forward secrecy, backward secrecy, and resistance to collusion attacks. And, compared with the previous schemes, our proposed scheme is an efficient and secure, self-healing, key-distribution scheme for WSNs.

Acknowledgments

Financial supports for this study provided by grant from National Natural Science Foundation of China (Project nos. 51108060, 50921001, 90815022), National Key Technology Research and Development Program during the Twelfth Five-Year Plan Period (Project no. 2011BAK02B01), and the Fundamental Research Funds for the Central Universities (Project no. DUT12JR13) are gratefully acknowledged.

References

[1] Y. Yu, J. Ou, J. Zhang, C. Zhang, and L. Li, "Development of wireless MEMS inclination sensor system for swing monitoring of large-scale hook structures," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 4, pp. 1072–1078, 2009.

[2] Y. Yu and J. Ou, "Wireless collection and data fusion method of strain signal in civil engineering structures," *Sensor Review*, vol. 29, no. 1, pp. 63–69, 2009.

[3] J. Staddon, S. Miner, M. Franklin, D. Balfanz, M. Malkin, and D. Dean, "Self-healing key distribution with revocation," in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 241–257, May 2002.

[4] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[5] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability," in *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)*, pp. 231–240, New York, NY, USA, October 2003.

[6] S. M. More, M. Malkin, J. Staddon, and D. Balfanz, "Sliding-window self-healing key distribution," in *Proceedings of the ACM Workshop on Survivable and Self-Regenerative Systems (In Association with 10th ACM Conference on Computer Communications Security)*, pp. 82–90, October 2003.

[7] C. Blundo, P. D'Arco, A. Santis, and M. Listo, "Definitions and bounds for self-healing key distribution," in *31st International Colloquium on Automata, Languages, and Programming (ICALP 2004)*, J. Díaz, J. Karhumaki, A. Lepistö, and D. Sannella, Eds., vol. 3142 of *Lecture Notes in Computer Science*, pp. 234–246, Springer, NewYork, NY, USA, 2004.

[8] D. Hong and J. S. Kang, "An efficient key distribution scheme with self-healing property," *IEEE Communications Letters*, vol. 9, no. 8, pp. 759–761, 2005.

[9] T. Biming and H. Mingxing, "A Self-healing key distribution scheme with novel properties," *International Journal of Network Security*, vol. 7, no. 1, pp. 115–120, 2008.

[10] C. Blundo, P. D'Arco, and M. Listo, "A flaw in a self-healing key distribution scheme," in *Proceedings of the Information Theory Workshop*, pp. 163–166, Paris, France, 2003.

[11] R. Dutta, E. Chang, and S. Mukhopadhyay, "Efficient self-healing key distribution with revocation for wireless sensor networks using one way hash chains," in *Proceedings of the 5th*

International Conference on Applied Cryptography and Network Security (ACNS'07), J. Katz and M. Yung, Eds., vol. 4521 of *Lecture Notes in Computer Science*, pp. 385–400, Springer, Heidelberg, Germany, 2007.

[12] R. Dutta, Y. D. Wu, and S. Mukhopadhyay, "Constant storage self-healing key distribution with revocation in wireless sensor network," in *Proceedings of the IEEE International Conference on Communications (ICC'07)*, pp. 1323–1328, Glasgow, UK, June 2007.

[13] Q. Y. Xu and M. X. He, "Improved constant storage self-healing key distribution with revocation in wireless sensor network," in *Information Security Applications (WISA 2008)*, vol. 5379 of *Lecture Notes in Computer Science*, pp. 41–55, Springer, Heidelberg, Germany, 2009.

[14] W. Du and M. X. He, "Self-healing key distribution with revocation and resistance to the collusion attack in wireless sensor networks," in *Provable Security (ProvSec 2008)*, vol. 5324 of *Lecture Notes in Computer Science*, pp. 345–359, Springer, Heidelberg, Germany, 2008.

[15] K. H. Bao and Z. F. Zhang, "Collusion attack on a self-healing key distribution with revocation in wireless sensor networks," in *Information Security Applications (WISA 2010)*, vol. 6513 of *Lecture Notes in Computer Science*, pp. 221–233, Springer, Heidelberg, Germany, 2011.

[16] C. Blundo, P. D'Arco, A. de Santis, and M. Listo, "Design of self-healing key distribution schemes," *Designs, Codes, and Cryptography*, vol. 32, no. 1–3, pp. 15–44, 2004.

[17] L. Harn, "Efficient sharing (broadcasting) of multiple secrets," *IEE Proceedings: Computers and Digital Techniques*, vol. 142, no. 3, pp. 237–240, 1995.