Research Article
A Group Key Distribution Scheme for Wireless Sensor Networks in the Internet of Things Scenario

Hong Yu,¹ Jingsha He,² Ting Zhang,¹ and Peng Xiao¹

¹ College of Computer Science and Technology, Beijing University of Technology, Beijing 100124, China
² School of Software Engineering, Beijing University of Technology, Beijing 100124, China

Correspondence should be addressed to Hong Yu, yuhong_0826@emails.bjut.edu.cn

Received 5 July 2012; Revised 2 October 2012; Accepted 5 October 2012

1. Introduction

As an indispensable part of the Internet of Things (IoT), wireless sensor networks (WSNs) need to be completely integrated into the Internet. When an Internet user communicates with multiple sensor nodes in WSNs, secure group key management becomes necessary. However, most current group key management schemes developed for WSNs do not consider the Internet scenario while traditional group key management in the Internet is deemed to be not suitable for WSNs due to the resource constraint characteristics of WSNs. In this paper, we propose a group key distribution scheme for WSNs in the IoT scenario in which we organize sensor nodes into groups in a hierarchical structure. In the upper wired layer, an end-to-end secure communication protocol is used to distribute group keys for subgroups to the trusted head nodes and the head nodes then distribute the group keys through underlying tree-based topology and wireless multicast to minimize energy consumption. We also perform some quantitative analyses as well as experiments to show that our proposed scheme is secure and has t-revocation capability. The total cost of distributing and rekeying the group keys is also analyzed and compared to that in some other comparable schemes.
Secure group communication requires secure and robust distribution or negotiation of group keys. A single symmetric key known only to the group in which the authorized user in the Internet and the multiple targeted sensor nodes in WSNs are the members can effectively protect communication for multicast group. Current group key management schemes in WSNs that belong primarily to the broadcast fashion by making use of wireless channels and transmission ranges cannot be directly applied to the IoT context since the user is usually located in different physical locations, even in different networks. Meanwhile, traditional group key management protocols developed for the Internet, such as IP multicast [12] that needs the support of multicast routers or application layer multicast [13] that needs the support of endhosts, are deemed to be inadequate for WSNs due to the characteristics of limited energy, storage, network bandwidth, communication capabilities in most sensor nodes, as well as multihop characteristics of WSNs, that is, each sensor node can act both as a router and as an end host.

In this paper, we propose a group key distribution scheme for WSNs in the IoT scenario. In our proposed scheme, the sensor nodes in WSNs are organized into groups in a hierarchical structure. In the upper wired layer, an end-to-end secure communication protocol is used to distribute group keys for the subgroups to trusted head nodes. In the lower wireless layer, the head nodes distribute the group keys through underlying tree-based topology and wireless multicast to minimize energy consumption of the sensor nodes. The main contributions of this paper can be summarized as follows.

1. We analyze the need for group key management for WSNs in the IoT scenario and propose a secure and efficient solution to overcome the limitations of existing mechanisms in the IoT scenario.
2. We design a hierarchical group key distribution scheme without requiring any preshared keys between the user and the sensor nodes. The scheme has the t-revocation capability and can minimize communication cost of the WSNs.
3. We demonstrate how our scheme can perform as a secure and efficient countermeasure against some attacks towards WSNs among which the cooperative compromised attack is analyzed emphatically, which is a capability that is absent in most existing mechanisms.
4. We conduct mathematical analysis on our scheme as well as performance comparison between our scheme and the Topological Key Hierarchy (TKH) scheme which is considered to be the most efficient mechanism for group key management in traditional WSNs. The comparison results show that our scheme outperforms the TKH scheme for group key distribution and group key rekeying when a small number of nodes are deleted.

The remainder of this paper is organized as follows. Section 2, we review some related work on group key distribution, group key negotiation, the self-healing theory, and the TKH scheme with which we will compare our scheme in the analysis. In Section 3, we describe our proposed group key distribution scheme, which includes assumptions, initialization, group key distribution, in upper wired layer, group key distribution and rekeying in lower wireless layer. In Section 4, we analyze our proposed scheme in terms of security and performance and compare it to TKH. Finally, in Section 5, we conclude this paper in which we also discuss some future work.

2. Related Work

In both the Internet and the WSNs, group key management can be classified into group key distribution (centralized model) and group key negotiation (distributed model).

2.1. Centralized Model. Group key distribution schemes (e.g., Group Key Management Protocol (GKMP) [14]) usually rely on a group controller which shares a pairwise key with each member of the group and distributes group keys to group members on a point-to-point basis. This approach cannot scale to large groups, which generates O(N) rekeying messages with a network of size N. In Logical Key Hierarchy (LKH) [15], individual and auxiliary keys are organized into a hierarchy and each group member is assigned to a leaf and holds all the keys from its leaf to the root. The root key as the group key is shared by all the group members. This scheme can reduce the number of rekeying messages to O(log N). One-Way Function Tree (OFT) [16] improves LKH by reducing the number of rekeying messages from (2log N) to (log N) in the binary key tree by using the local key computations. However, since in the multihop WSNs each sensor node can act both as a router and as an end host, rekeying messages generated from the logical key tree may be forwarded through one or more intermediate nodes to reach their final destination nodes, incurring heavy communication overheads. The Topological Key Hierarchy (TKH) [17] lowers the cost of rekeying messages by generating a key tree based on the underlying topology information of WSNs to minimize communication cost.

For group communication for WSNs in the IoT scenario, the authorized user in the Internet can act as the group controller, but all of the schemes mentioned above cannot be directly applied due to the following reasons. Firstly, the rekeying messages generated from the user must be transmitted over both the wired and wireless links, which may incur noticeable delay. Secondly, all of the centralized schemes rely on pairwise keys between the user and each sensor node, thus the user must authenticate and negotiate a shared secret key with each sensor node. Any group key distribution scheme should support end-to-end security communication that will make the cost of communication and computation grow linearly with the number of group members.

2.2. Distributed Model. In distributed group key negotiation, all group members are treated equally. Hence, group keys
should be negotiated among all group members through Diffie-Hellman (DH) key exchange or based on secret sharing theory to ensure fairness. In the CLIQUE scheme [18], group members can deliver their DH seeds orderly through insecure channels and the last member get all the DH seeds to compute the group key and then multicast the received DH seeds to other members so that all members can get the group key, thus generating $O(N^2)$ key messages and incurring $O(N^2)$ computation cost. Secret sharing theory [19] can enhance the robustness of group key generation in which each group member is issued a seed of the group key securely and any group member must collect $M(M < N)$ secret seeds from a subset of the group members to recover the group key. Therefore, an attacker who captures less than $M$ members cannot recover the group key. This approach causes frequent interactions and incurs high computation cost, and the exchange of secret seeds must be protected using shared keys between the peers.

In the IoT scenario, the above group key negotiation schemes are not suitable for WSNs since the cost of communication and computation is more than that of group key distribution schemes. Moreover, the reasons for the infeasibility of group key distribution schemes are also exist. Hierarchical group key management, for example, the Lotus scheme [20], is a tradeoff of the above two models in which group members are divided into many subgroups each of which has an independent group key. Thus, rekeying can be executed within the corresponding subgroup. Therefore, techniques based on hierarchical group are more suitable for WSNs in IoT scenario. In our proposed scheme, we separate the group key management into two layers. In the upper wired link layer, the Internet user distributes the group key to the head nodes in a WSN by using the end-to-end secure communication protocol. In the lower wireless link layer, each head node distributes the group key using self-healing theory [21] and TKH structure [17], which will be introduced in the next two subsections.

2.3. Self-Healing Theory. To be self-healing with the $t$-recovery capability, the group manager constructs and broadcasts a $t$-degree masking polynomial $\omega(x) = f(x)K + h(x)$ in which $h(x)$ is a $t$-degree shielded polynomial. For any normal group member $i$, $h(i)$ is preloaded, $f(x) = (x - r_1)(x - r_2)\cdots(x - r_t)$ is a $t$-degree revocation polynomial and $R = \{r_1, r_2, \ldots, r_t\}$ is the set of all deleted group members, $|R| \leq t$. For any normal group member $i$, it evaluates the polynomial at point $i$ and gets $(i) = f(i)K + h(i)$. Because $i$ knows $h(i)$ and $f(i) \neq 0$, it can compute $K = (\omega(i) - h(i))/f(i)$. Since the coalition $R$ gets at most $t$ points over the $t$-degree polynomial $h(x)$, it is computationally infeasible for coalition $R$ to learn $h(i)$ for $i \not\in R$.

2.4. TKH Group Rekeying Policy. In the TKH scheme that is applied to WSNs, the nodes in the same subtree (ST) share the same tree key (TK). ST is a tree with nodes below each subroot node, and the subroot nodes are direct neighbors of a sink. The nodes sharing the same parent node in a tree, that is, the sibling nodes, share the same sibling key (SK). Every node shares its own individual key (IK) with the sink. The group key (GK) is used to encrypt all data traffic within a group. TKH offers an advantage that the depth of the key tree is bounded to “4” regardless of the size of the network. Therefore, each node is only required to save a maximum of four keys, which is highly suitable for storage-limited sensor nodes.

TKH takes the advantage of the wireless multicast. Since a message transmission can be heard by multiple neighbors, sibling nodes can efficiently receive a message by a single transmission from their parent. An example is shown in Figure 1.

When node 3 in ST1 is revoked, the rekeying messages for ST2 and ST3 are $\{GK\}'_{TK}$ and $\{GK\}'_{TK}$, respectively. For ST1, the rekeying messages ($m$) and the corresponding communication cost ($C$) are

\[
\begin{align*}
m_{S-\{1\}} &= \{GK', TK\}'_{IK_1}, \\
C_{S-\{1\}} &= e_{tx} + e_{rx}, \\
m_{S-\{2,4\}} &= \{GK', TK\}'_{SK_2}, \\
C_{S-\{2,4\}} &= 2e_{tx} + 3e_{rx}, \\
m_{S-\{5,6\}} &= \{GK', TK\}'_{SK_2'}, \\
C_{S-\{5,6\}} &= 3e_{tx} + 4e_{rx}, \\
m_{S-\{7,8\}} &= \{GK', TK\}'_{SK_3}, \\
C_{S-\{7,8\}} &= 3e_{tx} + 4e_{rx},
\end{align*}
\]

where $e_{tx}, e_{rx}$ are the energy consumption of transmitting, receiving one bit respectively. Note that the sibling sets that share SK2 and SK3 are slightly changed. However, TKH does not update SK2 and SK3 since none of the nodes sharing them are deleted. By maintaining the link from node 7 to SK2 in the key tree, the sink can update both SK2 and SK3 later when node 7 is deleted. Thus, the total rekeying cost of ST1 is

\[
\text{TRC}_{ST1} = 2|m| \times (C_{S-\{1\}} + C_{S-\{2,4\}} + C_{S-\{5,6\}} + C_{S-\{7,8\}}) \\
+ |m| \times (C_{S-\{2\}} + C_{S-\{4\}} + C_{S-\{7\}} + C_{S-\{8\}}) \\
= |m|(25e_{tx} + 31e_{rx}).
\]

A new node should select a parent node to join the network and the existing nodes can change the corresponding GK, TK, and SK by using the preshared one-way function (Formula (3)) and the sink unicasts, $\{GK', TK', SK'\}'_{IK_{new}}$ to the newly added node:

\[
GK' = F(GK), \quad TK' = F(TK), \quad SK' = F(SK).
\]
Our method is motivated by the above analysis that the TKH scheme is suitable for WSNs due to taking use of the underlying sensor network topology to decrease the forwarding through intermediate nodes and the one-hop wireless multicast to save energy. However, in our proposed scheme, we further reduce the forwarding from intermediate nodes through the only once hop-by-hop wireless multicast along with the underlying topology. The legitimate sensor node in the group can recover the GK from the received information without the shared IK with the head node. Moreover, the adversary has to compromise at least $t$ group nodes instead of only one to get the new GK, which is known as $t$-revocation property.

3. The Proposed Scheme

Our proposed scheme organizes sensor nodes in a WSN into groups in a hierarchical structure as shown in Figure 2. An end-to-end secure communication protocol is used to distribute group key to the head nodes in subgroups in the upper wired link layer, while the head nodes distribute group keys to subgroup members in the lower wireless link layer through the underlying tree-based topology and by the means of wireless multicast.

3.1. The Assumption. We assume that the edge router is deployed for a WSN by the service provider (SP) and the
edge router is credible and has unlimited resources in terms of energy, computation, and storage.

3.2. Initialization. An SP should deploy and manage the WSN to provide services to the Internet users. The SP randomly picks a t-degree shielded polynomial $h(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j$ from $F_p(x, y)$, where $p$ is a prime number that is large enough to accommodate a cryptographic key, $a_{ij}$ is a polynomial coefficient for the WSN which is provided to authorized Internet users, $x$ is the current node, and $y$ is the current head node. Each node in the WSN $N_r (r = 1, 2, \ldots, n)$ is issued a personal key $S_r = h(N_r, y)$ and the one-way function $F$.

After deployment, the secure bootstrapping process in the WSN could be used (referring to [22]) to establish pairwise keys between neighboring nodes and trust paths to the edge router so as to form a multihop cluster-tree hierarchical topology, which can be considered as the routing tree in our scheme. The edge router knows the whole network topology, and each node knows its descendants as well as parent nodes.

3.3. The Wired Link Layer. An authorized Internet user $U$ can generate a group key $GK$ for the set $C$ of sensor nodes he/she want to communicate with, where $C = \{N_1, N_2, \ldots, N_c\}$, $|C| = c, c \leq n$. When $U$ launches a group request attached $C$ to the WSN, the edge router replies with a set of head nodes $H = \{N_1, N_2, \ldots, N_h\}$ ($|H| = h < c$) by checking the topology, choosing the nodes which have the minimum hop count to the edge router in each ST and ensuring that all nodes in the set $C$ could be included in all the STs of set $H$.

User $U$ distributes $GK$, $TK$ and $a_{ij}$ along with the set $C$ to all the nodes in $H$ through established secure channels (referring to [10] for establishing such secure channels).

3.4. The Wireless Link Layer. Suppose that the number of STs in the WSN is $h$. We take the head node $N_g (N_g \in H)$, that is, the subtree $ST_g$ as an example to describe the process executed in the wireless link layer.

3.4.1. Group Key Distribution. The head node $N_g$ constructs a $t$-degree shielded polynomial $h(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} x^i y^j$ using $a_{ij}$ received from user $U$ and computes $h(x, N_g)$. According to $C$ and the descendant nodes, head node $N_g$ will prepare three sets:

$M$: the malicious nodes in the WSN that have been deleted from the WSN and are not included in the routing tree;

$I$: the invited nodes in $ST_g$;

$N$: the normal nodes in $ST_g$ that are not invited by $U$ but can correctly and honestly execute the routing protocol.

$N_g$ sends the message $B$ along with the routing tree using wireless multicast fashion:

$$B = R \cup \{ \omega(x) = f(x)TK_g + h(x,N_g) \} \cup ET_{K_g}(GK), \tag{4}$$

where $f(x) = (x - N_r)(x - N_{r'}), \ldots, (x - N_{r''})$, a $t$-degree revocation polynomial, $R = M \cup N = \{N_r, N_{r'}, \ldots, N_{r''}\}$, $|R| = \omega_j \leq t$, and $TK_g$ is the key received from user $U$.

After receiving $B$, every remaining node $N_r (N_r \in I)$ will evaluate $f(x)$ and $\omega(x)$ at the point $x = N_r$, and $S_r = h(N_r, y)$ at the point $y = N_g$ to get $TK_g$:

$$TK_g = \left[ \frac{\omega(N_r) - h(N_r, N_g)}{f(N_r)} \right]. \tag{5}$$

where $f(N_r) \neq 0$. Then $N_r$ decrypts $GK$ with the computed $TK_g$.

3.4.2. Node Addition. The authorized Internet user $U$ can add a new sensor node $N_{new}$ to the set $C$. The edge router will inform $U$ to which ST $N_{new}$ should belong. $N_{new}$ can be either in a new ST or a node in an existing ST.

In the former case, taking $N_{new}^1$ in Figure 3 for example, $U$ distributes $GK' = F(GK)$, $TK_{new}$, and $a_{ij}$ along with the set $C'$ to $N_{new}^1$ through secure channels as described in Section 3.3. All nodes in $C$ will locally compute $GK' = F(GK)$ by using the pre-shared one-way function $F$.

In the latter case, assuming that $N_{new} \in ST_g$, all nodes in $C$ will locally compute $GK' = F(GK)$ and all nodes in $ST_g$ will locally compute $TK' = F(TK)$. $U$ informs $N_g$ of the new set $C'$, and $N_g$ checks the location of $N_{new}$ in the routing tree. If the immediate parent node of $N_{new}$ is in $C$, that is $N_{new}^2$ in Figure 3, $N_{new}^2$ will receive $\{GK', TK'\}$ encrypted using the pairwise key shared between $N_{new}^2$ and node $3$. Otherwise, that is $N_{new}^3$ in Figure 3, node $1$ will unicast $R \cup \{ \omega(x) = f(x)TK'_g + h(x,N_g) \} \cup ET_{K'_g}(GK')$ along the routing tree to $N_{new}^3$ which can then compute $TK'_g$ and decrypt $GK'$ as described in Section 3.4.1.
3.4.3. Node Deletion. The authorized Internet user \( U \) deletes a sensor node \( N_{\text{rev}} \) from the set \( C \). The edge router will inform \( U \) of to which ST \( N_{\text{rev}} \) belongs. We classify the node deletion event into the cases of ordinary node deletion and head node deletion.

In the former case, assuming that \( N_{\text{rev}} \) is an ordinary node, \( N_{\text{rev}} \in ST_g \), if \( N_{\text{rev}} \) is a leaf node, that is node 5 in Figure 4, the routing tree is not affected but if \( N_{\text{rev}} \) is a nonleaf node, that is node 4 in Figure 4, the network topology might be affected. Then, the routing tree should be repaired using a method in which a nonleaf ordinary node is replaced by one of its siblings in the routing tree [22]. \( U \) will then distribute the new GK and \( R' \) (\( R' = R \cup \{N_{\text{rev}}\}, |R'| \leq t \)) to all the head nodes in \( H \) and the new TK'_{g} to \( N_{g} \). The head nodes in the other STs except \( ST_g \) will multicast \( \{GK'_{TK} \cup R' \} \) along with the routing tree. Every remaining node \( N_{r} \) (\( N_{r} \in I \) and \( N_{r} \notin R \)) will get TK'_{S} and GK' as described in Section 3.4.1.

In the latter case, assuming that \( U \) will delete node 1 in Figure 4, such a nonleaf head node can be replaced by one of its children which also has children in the routing tree, that is node 3 in Figure 4. The edge router will check the network topology and inform \( U \) of the new head node 3. \( U \) will implement the procedure as described in Section 3.3, node 3 will implement the procedure as described in Section 3.4.1, and all the other STs will multicast \( \{GK'_{TK} \cup R' \} \).

The procedure of node addition and node deletion is illustrated in Figure 5.

3.4.4. Reduction of the Size of Set \( R \). We can see that the multicast information \( B \) carries \( \omega \) IDs in set \( R \), which increases the packet length and therefore energy consumption for communication. This will become more serious since the size of \( R \) will increase dramatically when the number of revoked nodes increases. In the following, we use the bloom filter technique to reduce the size of set \( R \).

Bloom filter is a well-known data structure that can be used for efficient membership checking. Using the method, we can find whether an element belongs to a predefined set. A bloom filter consists of a set \( S = \{S_1, S_2, \ldots, S_n\} \), a string of \( m \) bits, and \( k \)-independent hash functions \( h_1, h_2, \ldots, h_k \) [23].

Each hash function \( h_i(i = 1, 2, \ldots, k) \) maps an element \( S_j(j = 1, 2, \ldots, n) \) uniformly to range \([0, 1, \ldots, m - 1] \), each of which corresponding to a bit in the \( m \)-bit string. The \( m \)-bit string is initially set to 0. For the element \( S_j \in S \), we can obtain its \( k \) hash values \( h_i(S_j)(i = 1, 2, \ldots, k) \). Thus, the bits corresponding to these values are set to 1 in the string. There may be more than one of the values mapped to the same bit in the string. In order to find whether element \( S_j' \in S \), the \( k \) bits \( h_i(S_j')(i = 1, 2, \ldots, k) \) are checked. If all the bits are 1, then \( S_j' \in S \). Otherwise, at least one of the bits is 0, \( S_j' \notin S \).

Bloom filter may yield false positives, that is, although an element is not in \( S \), its hash values \( h_i(S_j) \) are collectively marked by other elements in \( S \). If the hash is uniformly random over the \( m \) values, the probability that a bit is 0 after all the \( n \) elements are hashed and their bits marked is \((1 - (1/m)^k)^n \approx e^{-kn/m}\).

Therefore, the probability for a false positive is:

\[
\left(1 - \left(1 - \frac{1}{m}\right)^k\right)^n \approx \left(1 - e^{-kn/m}\right)^k. \tag{6}
\]

An example is shown in Figure 6 in which we assume that \( c = 128 \) and \( t = [0.1c] = 12 \). Hence, the length of node’s ID is \(|p| = \log c = 7 \) bits, and, consequently, the size of \( R \) is 84 bits. We should let \( m < t|p| = 84 \) in order to reduce the size of \( R \) and let \( m > kt = 12k \) in order to minimize the probability of false positive. Therefore, we use the bloom filter technique with \( k = 3 \) hash functions, which will map 12 IDs to an \( m = 63 \) bit string. As a result, the total size of \( R \) can be reduced by 25% and the probability of false positive is
about 8%. Fortunately, however, false positive has little effect on our scheme. On one hand, the illegitimate node that is mistakenly regarded as a legitimate node calculates $f(x) = 0$. On the other hand, the legitimate node which has been mistakenly revoked may also receive $B$ and calculate $TK$ due to communication in a multicast fashion.

4. Analysis

We analyze and show that our proposed scheme can provide confidentiality, forward secrecy, backward secrecy, and $t$-revocation capability for group keys. We also analyze the performance of our scheme and compare it with TKH
scheme in terms of the cost of storage, computation, and communication.

4.1. Security Analysis. (1) Group key confidentiality: any sensor node out of the group defined by the Internet user cannot get the group key.

In our scheme, GK and TK are generated by the authorized user. In the wired link layer, GK and TK are protected by the end-to-end secure channel and transmitted to the authenticated head nodes. In the wireless link layer, firstly, the head node is responsible for distributing GK and TK to other nodes in its ST. Every node must compute TK from \( \omega(x), h(x,N_g) \) and \( f(x) \) in order to decrypt GK. However, no attacker has the correct personal key \( S_x = h(x,N_g) \), and any normal node which is not invited by \( U \) may calculate \( f(x) = 0 \), neither of which can generate the correct TK. Secondly, when a new node is added, the rekeying GK and TK are either locally computed by using the one-way function \( F \) after encryption using the pairwise key shared between the added node and its parent.

(2) Forward secrecy: any sensor node cannot get the group key after its deletion.

In our scheme, all the deleted nodes are added to the revocation set \( R \) and known by every node in the WSN. The head node constructs the new multicast message \( B' \) using the revocation polynomial \( f'(x) \), but the deleted nodes compute \( f'(x) = 0 \) so that it cannot get the new GK and TK.

(3) Backward secrecy: any sensor node added by the Internet user cannot get the group key before it is actually added.

In our scheme, the newly added node gets the TK\(_{new} \) and GK\(^{'} = F(GK) \) from U using an end-to-end secure channel, gets TK\(^{'} = F(TK) \) and GK\(^{'} = F(GK) \) from its immediate parent node, or computes TK\(^{'} \) and decrypts GK\(^{'} \) from \( B' \) with its personal key. Because \( F \) is a one-way function, it is not possible to derive TK and GK from \( F(TK) \) and \( F(GK) \).

(4) \( t \)-revocation capability: the coalition \( R \) cannot get any information about the current GK.

In our scheme, in order to know TK\(_{new} \), the coalition \( R \) needs to know \( h(x,N_g) \) and at least \( (t+1) \) points on the polynomial \( h(x,N_g) \). Since the size of the coalition \( R \) is at most \( t \), the coalition \( R \) has at most \( t \) pieces of personal secrets \( S_x = h(N_x,y) \), that is, \( t \) points on the polynomial \( h(x,N_g) \). But at least \( (t+1) \) points are needed on the polynomial \( h(x,N_g) \) to recover the current TK\(_{new} \) for any node in \( R \).

4.2. Performance Analysis. For any group key management schemes for WSNs, even in the IoT scenario, storage, computation, and communication overhead as well as energy consumption of the sensor node are among the issues mostly concerned about. We therefore conduct performance analysis by comparing our proposed scheme with TKH in the following three aspects: storage, computation, communication. In the wireless link layer, (1) the WSN in our scheme is bootstrapped based on multihop hierarchical tree topology, (2) the group key distribution makes use of the underlying network topology with the consideration of ST-based group organization, and (3) wireless multicast advantage is taken to replace multicasting mechanisms, which are the same as TKH.

According to our scheme and TKH scheme, we do not consider the node addition event since the topology change and the corresponding rekeying cost is negligible.

In our analysis, we use the \( a\beta\gamma \)-tree model in which \( a \) is the number of STs, \( \beta \) is the number of sibling sets in each ST and \( y \) is the number of nodes in each sibling set. So, \( |C| = c = a(\beta y + 1) \), and each ST has \( (\beta y + 1) \) nodes.

4.2.1. Storage Overhead. In our scheme, every node in WSN is preloaded with a personal key \( S_x = h(N_x,y) \), which is a \( t \)-degree shielded polynomial, the two keys GK and TK, and \( (y + 1) \) pairwise keys shared with its neighboring nodes. Therefore, the storage space required is \( (t+1)\lg p + (y+3)\lg q \) assuming that the length of the node's ID is \( p \) and the length of the key is \( \lg q \). In the TKH scheme, every ordinary node must store four keys: GK, TK, SK, IK, and the pairwise keys shared with every other node in the ST. Thus, the storage overhead is \( (\beta y + 4)\lg q \). In addition, every head node must store \( \beta \) SKs, making the storage overhead become \( (\beta y + \beta + 2)\lg q \).

4.2.2. Computation Overhead. In our scheme, for group key distribution, the head node must compute a point \( h(x,N_g) \) on the polynomial \( h(x,y) = \sum_{j=0}^{t} \sum_{i=0}^{\gamma} a_{ij}x^iy^j \) which requires at most \( t \) multiplication operations and carry out one encryption operation. While the ordinary node must compute \( f(x), \omega(x) \) at the point \( x = N_i \) and \( S_x = h(N_x,y) \) at the point \( y = N_q \). Since division can be regarded as multiplication, the total number of multiplication operations required to get TK\(_{new} \) is \( 3t + 1 \). Meanwhile, the ordinary
node must carry out one decryption operation. In the TKH scheme, the group key distribution needs $\beta y$ encryption operations for a head node and one decryption operation for each ordinary node. In addition, TKH needs $\beta y$ pairwise key establishing operations between the head node and other ordinary nodes.

4.2.3. Communication Overhead. We define communication overhead as the total cost which reflects both the number of messages and the cost of message transmission.

1. Total cost of distributing group key (TDC): in our scheme, the head node must multicast message $B(B = R \cup \{w(x) f(x)_{\text{TK}_{2}} + h(x,N_{2})\} \cup E_{\text{TR}}(GK))$ which is transmitted along the tree in every hop. Thus, the total cost of distributing group key is

$$TDC = \alpha(\beta \cdot e_{tx} + \beta y \cdot e_{rx}) \cdot |B|,$$

where $|B| = (\alpha w + t + 1) lg p + lg q$.

In the TKH scheme, the head node takes the responsibility of unicasting GK, TK and SK to every ordinary node encrypted using the pairwise key between the head node and an ordinary node. So the total cost of distributing group key is:

$$TDC_{\text{TKH}} = \alpha \cdot \beta y \cdot \text{avg}(1, \beta) \cdot (e_{tx} + e_{rx}) \cdot |\text{keys}|,$$

where $\text{avg}(1, n) = (1 + 2 + \cdots + n)/n = (n + 1)/2$ and $|\text{keys}| = 3 lg q$.

2. The total cost of rekeying group key (TRC): we only consider the node deletion event since the rekeying cost of node addition is negligible. When $l$ nodes are deleted, including $v$ head nodes and $(l - v)$ nonhead nodes, assuming that $w$ STs have no deleting nodes, then the total cost of rekeying key is

$$\text{TRC} = [((\alpha - w) \cdot \beta \cdot \gamma \cdot e_{tx} + ((\alpha - w) \cdot \beta \cdot \gamma \cdot \gamma \cdot l \cdot e_{rx}) \cdot |B'| + (\beta \cdot e_{tx} + \beta y \cdot e_{rx}) \cdot w \cdot |B''|],$$

where $|B'| = (l + t + 1) lg p + lg q$ and $|B''| = l \cdot lg p + lg q$.

Exp($w$) calculates the average number of ST which has no deleting nodes when $l$ nodes are deleted:

$$\text{Exp} (w) = \sum_{i=a-\min(l, a)}^{a-\lceil \beta y + 1 \rceil} \text{Pr} \{w = i\} \cdot i.$$

In the above equation,

$$\text{Pr} \{w = i\} = \frac{C_{\alpha - l} \cdot N(\alpha - l, \beta y + 1)}{C_{\alpha}},$$

where $\text{Pr} \{w = i\}$ is the probability of $w = i$, $N(\alpha - l, \beta y + 1)$ is the number of ways in which $l$ nodes are to be deleted from the $(\alpha - i)$ STs, at least one has to come from each $(\alpha - i)$ ST that has $(\beta y + 1)$ nodes. $N(\alpha - l, \beta y + 1)$ can be calculated by using the recursive procedure and the result is as follows:

$$N(\alpha - l, \beta y + 1) = \sum_{k=1}^{l-(\alpha - i) + 1} C_{\beta y + 1}^k \cdot N(\alpha - i - 1, \beta y + 1).$$

In the TKH scheme, the total cost of rekeying key is $\text{TRC}_{\text{TKH}}$ which is also expressed by the function with $\alpha, \beta, \gamma, v$, and $l$ [17].

3. Comparison between our scheme and the TKH scheme: we set the total number in set $C(c)$ to be 32, 64, 128, 256, 512, 1024 by designing $(\alpha, \beta, \gamma)$ as $(2, 5, 3), (4, 5, 3), (2, 7, 9), (4, 7, 9), (8, 7, 9),$ and $(16, 7, 9)$, respectively. Let $q = 2^{128}$, $t = 0.1c$, and $|p| = lg c$. Each node can be identified by using lg c bits. From the characteristics of the CC2420 transceiver used in the Crossbow’s MICAZ and Telos B sensor nodes [24], the unit communication cost is $c_{tx} = 0.209 \mu f$ and $c_{rx} = 0.226 \mu f$.

Figure 7 shows the total cost of group key distribution for $(2, 5, 3), (4, 5, 3), (2, 7, 9), (4, 7, 9), (8, 7, 9),$ and $(16, 7, 9)$. We can see from Figure 3 that the total cost of group key distribution in our scheme is lower than that in TKH and the difference widens as the size of the network increases. This is because message $B$ is transmitted only once in every hop along the hierarchical routing tree by using the wireless multicast advantage in our scheme. In TKH, however, key
messages are different for each node, which are transmitted according to the structure of the TKH tree. Moreover, the length of $B (|B| = (\omega_j + t + 1)\lg p + \lg q)$ is shorter than keys ($|\text{keys}| = 3\lg q$).

Figure 8 shows the total cost of rekeying group key when $l(1 \leq l \leq \lfloor 0.1c \rfloor)$ nodes are revoked for (2,5,3), (4,5,3), (2,7,9), (4,7,9), and (8,7,9) in our scheme. We can see that the total cost of rekeying group key increases as the total number of nodes increases. However, for each $(\alpha, \beta, \gamma)$, it increases slightly when $l$ increases. This is because $B$ must be transmitted by more nodes when the total number of nodes increases, which is the main source of the total cost of rekeying group key. However, when $l$ increases in the same $(\alpha, \beta, \gamma)$, even though the cost on multicasting $B$ in one ST decreases, the number of STs in which TK should be revoked increases, which leads to the slight increase in the total cost of rekeying group key.

The total cost of rekeying group key in both schemes is shown in Figure 9 when (a) one node is deleted and (b) $t = \lfloor 0.1c \rfloor$ nodes are deleted for (2,5,3), (4,5,3), (2,7,9), (4,7,9), and (8,7,9). When one node is deleted (i.e., Figure 9(a)), in the ST where the deleted node existed, the total cost in our scheme is smaller than that in the TKH scheme when the total number of nodes is smaller than about 400 while, in all other STs, the cost is almost the same in both schemes. This is because when the total number of nodes is more than about 400, the value $t$ should be much larger to ensure $t$-revocation capabilities, causing $|B|$ to also become larger and longer than $|\text{keys}|$ in the TKH scheme. Thus, the cost of multicasting is higher than that in the TKH scheme when the total number of nodes is more than about 300.

In conclusion, our scheme outperforms the TKH scheme for group key distribution and group key rekeying when fewer numbers of nodes are deleted but it may become less advantageous in group key rekeying as the size of the network increases and when a large number of nodes are deleted.

4.3. Experiment. We set up a real experimental environment in which the 34 Crossbow’s MICAZ motes that are used as the sensor nodes each has 8-bit ATmegaL 128L clocked at about 7.37-MHz microcontroller and complies with the IEEE 802.15.4 standards with data transmission rate of 250 kbps. As depicted in Figure 10, we only show half of the sensor nodes that are deployed on the fourth and the fifth floor in our lab (the other half are deployed on the second and the third floors in the same fashion). The Stargate NetBridge gateway (Base 0 of the light blue) has an Intel IXP420 XScale processor running at 266 MHz. The MIB 600 attached to the gateway can connect to a wired Ethernet and the 802.15.4. The Internet user uses a Pentium IV machine clocked at about 2.1 GHz CPU with data transmission rate of 100 mbps. Through the IP-enabled router, the remote Internet user can randomly choose $c = 32$ sensor nodes as the elements in the set $C$ with which the user communicates. We set the unit communication cost to be the same as that in the analysis, that is, $e_{tc} = 0.209 \mu J$ and $e_{tc} = 0.226 \mu J$. We obtain the total cost of group key distribution (TDC) and the total cost of rekeying group keys (TRC) when $l = 1$ and $l = \lfloor 0.1c \rfloor = 3$ by executing the experiment 10 times and the experimental results are shown in Figure 11.

We can see from Figure 11 that the average TDC in our proposed scheme is 1.33 mJ, lower than 1.78 mJ in the TKH scheme. However, both of them are higher than the theoretical values of 1.3 mJ and 1.7 mJ, respectively. Meanwhile, the average TRC in our proposed scheme is 1.28 mJ and 1.35 mJ when $l = 1$ and $l = 3$, respectively, while the average TRC in the TKH scheme is 2.88 mJ and 3.08 mJ when $l = 1$ and $l = 3$, respectively, both of which also higher than the theoretical values. The reason is probably that messages could be retransmitted due to wireless channel errors, resulting in additional multicasting cost. Therefore, we consider the experiment results to be consistent with the theoretical analysis.

5. Conclusion

In this paper, we proposed a group key distribution scheme for WSNs in IoT scenario in which the sensor nodes are organized in two logic layers in a hierarchical structure. In the upper wired link layer, an end-to-end secure communication protocol is used to distribute group key to the head nodes in the subgroups. In the lower wireless link layer, each head node distributes the group key by using the underlying tree-based topology and wireless multicast advantage to minimize energy consumption. An analysis on the proposed scheme showed that our proposed scheme can achieve $t$-revocation security. We also performed some analyses to compare our
scheme with the TKH scheme in terms of the cost of storage, computation, and communication. The analyses showed that our scheme outperforms the TKH scheme in terms of total cost of communication for distributing group key and rekeying group key when fewer nodes are deleted but less advantageous when the size of the network increases and when a large number of nodes are deleted in rekeying group key. In the future, we will conduct more experiments to verify the results in real applications and further improve the performance in large-scale WSNs when a large number of nodes are deleted.

Acknowledgments

The work in this paper has been supported in part by National Natural Science Foundation of China (Grant no. 61272500) and in part by Beijing Education Commission Science and Technology Fund (Grant no. KM201010005027).

References

[1] N. Kushalnagar, G. Montenegro, and C. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” IETF RFC 4919, 2007.

[2] G. Montenegro, N. Kushalnagar, and J. Hui, “Transmission of IPv6 Packets over IEEE 802.15.4 Networks,” IETF RFC 4944, 2007.

[3] J. Granjal, R. Silva, E. Monteiro, J. S. Silva, and F. Boavida, “Why is ipsec a viable option for wireless sensor networks,” in Proceedings of the 5th IEEE International Conference on Mobile
International Journal of Distributed Sensor Networks

Ad-Hoc and Sensor Systems (MASS ’08), pp. 802–807, October 2008.

[4] J. Granjal, E. Monteiro, and J. S. Silva, “A secure interconnection model for IPv6 enabled Wireless Sensor Networks,” in Proceedings of the IFIP Wireless Days (WD ’10), pp. 1–6, October 2010.

[5] J. Granjal, E. Monteiro, and J. Sá Silva, “Enabling network-layer security on IPv6 wireless sensor networks,” in Proceedings of the 53rd IEEE Global Communications Conference (GLOBECOM ’10), pp. 1–6, December 2010.

[6] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, “Securing communication in 6LoWPAN with compressed IPsec,” in Proceedings of the International Conference on Distributed Computing in Sensor Systems, pp. 1–8, 2011.

[7] S. Raza, T. Voigt, and U. Roedig, “6LoWPAN Extension for IPsec,” http://www.iab.org/wp-content/IAB-uploads/2011/03/Raza.pdf.

[8] W. Jung, S. Hong, M. Ha, Y. J. Kim, and D. Kim, “SSL-based lightweight security of ip-based wireless sensor networks,” in Proceedings of the International Conference on Advanced Networking and Applications Workshops (WAINA ’09), pp. 1112–1117, May 2009.

[9] R. Mzid, M. Boujelben, H. Youssef, and M. Abid, “Adapting TLS handshake protocol for heterogeneous IP-based WSN using identity based cryptography,” in Proceedings of the International Conference on Wireless and Ubiquitous Systems, pp. 1–8, 2010.

[10] H. Yu, J. He, T. Zhang, P. Xiao, and Y. Zhang, “Enabling End-to-End Secure Communication between Wireless Sensor Networks and the Internet,” World Wide Web Journal.

[11] R. Riaz, K. H. Kim, and H. F. Ahmed, “Security analysis survey and framework design for IP connected LoWPANs,” in Proceedings of the International Symposium on Autonomous Decentralized Systems (ISADS ’09), pp. 29–34, March 2009.

[12] C. Diot, B. N. Levine, B. Lyles, H. Kasse, and D. Balasescu, “Deployment issues for the IP multicast service and architecture,” IEEE Network, vol. 14, no. 1, pp. 78–88, 2000.

[13] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, “A survey of application-layer multicast protocols,” IEEE Communications Surveys and Tutorials, vol. 9, no. 3, pp. 58–74, 2007.

[14] H. Harney and C. Muckenhirn, “Group Key Management Protocol (GKMP) Architecture,” IETF RFC 2094, 1997.

[15] D. Wailner, E. Harder, and R. Agee, “Key Management for Multicast: Issues and Architectures,” IETF RFC 2627, 1997.

[16] G. Horng, “Cryptanalysis of a key management scheme for secure multicast communications,” IEICE Transactions on Communications, vol. 85, no. 5, pp. 1050–1051, 2002.

[17] J. H. Son, J. S. Lee, and S. W. Seo, “Topological key hierarchy for energy-efficient group key management in wireless sensor networks,” Wireless Personal Communications, vol. 52, no. 2, pp. 359–382, 2010.

[18] M. Setiner, G. Taudik, and M. Waidnet, “Cliques: a new approach to group key agreement,” in Proceedings of the 18th International Conference on Distributed Computing Systems, pp. 380–387, 1998.

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

[20] S. Mittra, “Iolus: a framework for scalable secure multicast,” ACM SIGCOMM Computer Communication Review, vol. 27, no. 4, pp. 277–288, 1997.

[21] 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, June 2007.

[22] H. Yu and J. He, “Trust-based mutual authentication for bootstrapping in 6LoWPAN,” Journal of Communications, vol. 7, no. 8, pp. 634–642, 2012.

[23] BLOOM BH, “Space/time trade-offs in hash coding with allowable errors,” Communications of the ACM, vol. 13, no. 7, pp. 422–426, 1970.

[24] Texas Instruments Inc., “Single-Chip 2.4GHz IEEE 802.15.4 Compliant and ZigBee (TM) Ready RF Transceiver,” http://www.ti.com/lit/ds/symlink/cc2420.pdf.