9-7-2007

Group Rekeying Schemes for Secure Group Communication in Wireless Sensor Networks

Yong Wang

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Yuyan Xue

University of Nebraska–Lincoln, Computer Science and Engineering
Technical Report TR-UNL-CSE-2007-0018
Issued Sept. 7, 2007

Group Rekeying Schemes for Secure Group Communication in Wireless Sensor Networks

Yong Wang, Byrav Ramamurthy, and Yuyan Xue

***Abstract*— Wireless sensor networks are promising solutions for many applications. However, wireless sensor nodes suffer from many constraints such as low computation capability, small memory, limited energy resources, and so on. *Grouping* is an important technique to localize computation and reduce communication overhead in wireless sensor networks. In this paper, we use grouping to refer to the process of combining a set of sensor nodes with similar properties. We propose four centralized group rekeying (CGK) schemes for secure group communication in sensor networks. The lifetime of a group is divided into three phases, i.e., group formation, group maintenance, and group dissolution. We demonstrate how to set up the group and establish the group key in each phase. We further analyze and evaluate the performance of the proposed schemes in different scenarios.**

***Index Terms*— Wireless sensor network, secure group communication, clustering, grouping**

I. Introduction

Wireless Sensor Networks (WSNs) are used in many applications in military, environmental and health related areas. However, nodes in a WSN suffer from many constraints such as low computation capability, small memory, limited energy resources, and so on. Grouping is an important technique to localize computation and reduce communication overhead in WSNs.

The most common method of grouping is clustering. The essential operation in sensor node clustering is to select a set of cluster heads among the sensors in the network, and cluster the rest of the nodes with these heads [1]. Cluster heads are responsible for coordination among the nodes within their clusters, and communication with each other and/or with external observers on behalf of their clusters. Many routing protocols and key management protocols have been proposed using the clustering technique [2], [3].

However, grouping goes far beyond clustering. In this paper, we use grouping to refer to the process of combining a set of sensors with similar properties. The essential operation in sensor node grouping is to dynamically combine a set of sensors based on the observed events. The result of the grouping is a group. Unlike clustering focusing on the whole sensor network, grouping is only involved with sensors in a small region. Without additional clarifications, the term grouping in this paper refers to the local combination of a set of sensor nodes.

There are many similarities between clustering and grouping, for example:

- Sensors in a cluster or a group are usually geographically close to each other.
- Both clustering and grouping are used to localize computation and reduce communication overhead.
- A cluster usually has a cluster head and a group may have a group controller.

However, important differences exist between clustering and grouping. The main differences are listed below:

- Clustering is a global concept while grouping usually focuses on a small region. When clustering is used in a sensor network, the whole sensor network is divided into clusters. However, grouping usually involves with a relatively small number of sensors. These sensors are combined together based on the defined properties.
- Clustering and grouping could be adopted separately or together. They do not depend on each other. Grouping can be carried out with clustering or without.
- When clustering and grouping are both used to organize a sensor network, a group could be a part of a cluster, or even the union of several clusters.
- Clusters are decided by the partition algorithm adopted in the sensor networks. There is no relation between the clusters and the observed events. However, groups are usually activated by events. A group is set up and dissolved on the fly.

Security is an important research area in sensor networks [4]. In this paper, we focus on secure group communication (SGC) [5]. Secure group communication in sensor networks refers to a scenario in which sensors in a group can send and receive messages to/from group members in a way that outsiders are unable to glean any information even when they are able to intercept the messages. Secure group communication depends on the *group key* to protect the messages. The security requirements of group communication include authentication, confidentiality, integration, freshness etc. [5]. In addition, secure group communication also requires forward secrecy and backward secrecy [5]:

- Forward secrecy: A sensor should not be able to read any future messages after it leaves the group.
- Backward secrecy: A joining sensor should not be able to read any previously transmitted message.

The obvious benefit of secure group communication to WSNs is that outside nodes are unable to obtain any messages transmitted to the group. Secure group communication is also attractive for in-network processing and data aggregation [6]. The group controller can be used to aggregate similar data to reduce the communication overhead in a sensor network. Further, recent research has also revealed that the group key can be used for filtering false data injected in a sensor network

[7], [8]. The basic idea using the group keys to filter the injected false data is as follows: Sensor nodes are divided into multiple groups. Nodes in the same group share a group key and the base station knows all the group keys. Each data report sent to the base station is attached with multiple Message Authentication Codes (MACs) and each MAC is generated by a node that detects the same event. When the data report is forwarded along the way to the the base station, an en-route node may use its group key to verify the MACs probabilistically and drop those with invalid MACs at earliest points. The base station further filters out remaining false reports that escape the en-route filtering.

Although a few papers [9], [10] discussed secure group communication in sensor networks in literature, the problem has not been well studied. Previous works on secure group communication either consider the whole sensor network as a group or define the immediate neighboring nodes around a sensor as a group. However, grouping is more general than these two cases.

Our contributions in this paper are as follows: We formally define the grouping and secure group communication problem in WSNs. We differentiate between the concepts of clustering and grouping. We propose four centralized group rekeying schemes for secure group communication in WSNs and further evaluate their performances in various group settings.

The paper is organized as follows: Section II discusses the related work. Section III introduces grouping and its properties. Section IV presents our proposed centralized group rekeying schemes, followed by the security and performance analysis in Section V, the simulation and results in Section VI. Section VII concludes the paper.

II. Related work

The secure group communication problem has been extensively studied in the context of secure multicast in wired or wireless networks. Many centralized solutions and a few distributed solutions have been proposed. However, most of them are not suitable for WSNs. For example, the centralized schemes proposed in [11], [12] assume a key tree is maintained in the central controller. However, none of them considers the management overhead of such a key tree structure in the central controller, which is important in sensor networks due to the constraints on the sensor nodes. The distributed schemes, such as [13], [14], use exponential operations to generate and update the group key, which are also unbearable in WSNs. Since most of the distributed schemes for secure group communication requires excessive computation to generate the group key, most of proposed secure group communication schemes for WSNs are centralized scheme. In this paper, we focus on centralized group communication schemes.

A few papers [9], [15], [10], [16] address the secure group communication problem in the content of sensor networks. However, most of the work consider the whole sensor network as a group or define the immediate neighboring nodes around a sensor as a group. For example, in [15], Pietro *et al.* proposed a centralized group rekeying scheme based on logical key tree hierarchy (LKH) for WSNs. The whole sensor network is considered as a group and the base station is regarded as the central controller in the group. In [16], Zhang *et al.* proposed a group rekeying scheme for filtering false data in sensor networks. In their scheme, the group is defined as the immediate neighboring nodes around a sensor.

In [6], Huang *et al.* proposed a hierarchical secure group communication scheme for wireless sensor networks. Their scheme allows multiple groups existing in the sensor network. Each group has a group controller and shares a group key. A sensor node can query the group information from the base station and decide which group to join. The base station is the only node which can initiate and construct the group-based tree.

In this paper, we consider a more general situation. A set of sensor nodes form a group and one of the sensor nodes takes the role of the group controller. The group controller starts to initialize and construct the group. We consider the situations in which the sensor nodes in the group might be separated by multiple hops. Such groups are attractive in many applications. For example, many applications in WSNs are event-based and the focus in these applications is an event area or area of interest [17]. This area becomes a hotspot in the sensor network and all the sensor nodes within this area form a group.

In the following sections we present our proposed centralized group rekeying schemes. We use the following notation for the remainder of this paper:

- A, B are principals such as communicating nodes.
- ID_A denotes the sensor identifier of node A.
- $e(A,T)$ is a set of events observed by sensor A in time period T.
- $K_{A,B}$ denotes the secret pairwise key shared between A and B.
- M_K is the encryption of message M with key K.
- $MAC(K,M)$ denotes the computation of the message authentication code of message M with key K.
- $A \longrightarrow B$ denotes A unicasts a message to B.
- $A \longrightarrow *$ denotes A broadcasts a message to its neighbors.

III. Grouping and its properties

As we discussed before, grouping refers to the process of combining a set of sensors with similar properties. These properties usually refer to the events observed by the sensors. A group can be defined by many aspects. For example, all photo sensors activated in the last one minute form a group; the temperature sensors with temperature more than $100°\mathrm{C}$ form a group. Without loss of generality, we define a group *G* as a set of sensors A in region R which observe an event *E* satisfying criteria C in a period of time *T*:

$$G = \{A | E \in e(A,T) \ \ and\ C\ and\ A\ in\ R\} \tag{1}$$

where $C = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ and $C_1, \cdots, C_n$ are criteria describing the properties of the event, R is a rectangle area which is defined by its left top and right bottom points, $R = [(x_1, y_1), (x_2, y_2)]$. Each event is identified by a unique *event id (eid)*.

Let $eid = 1000$ represent the temperate event and t represent the sensing value. The temperature sensors with

temperature more than $100°\mathrm{C}$ in the area of (50, 50) to (100, 100) can be defined as:

$$\begin{aligned} G = \{A \mid eid &= 1000\ and \\ t &> 100\ and \\ A &\ in\ [(50,50),(100,100)]\} \end{aligned}$$

The lifetime of a group can be divided into three phases, i.e., *group formation*, *group maintenance*, and *group dissolution*.

In the *group formation* phase, the sensor nodes which satisfy the defined criteria form a group. The process of group formation is usually triggered by a special node, which is called a *group controller*. The group controller can be decided by the *controller selection process*. A simple way to decide a group controller is as follows: when an event E occurs in the field, the sensor detecting this event and having the strongest signal stands out as the group controller. The group formation phase is ended with all the group members receiving the group key. Then, the group maintenance phase begins.

The *group maintenance phase* is divided into sessions. The duration of sessions Δ_s can be fixed or dynamic depending on the applications. The group controller is responsible for distributing the group key to the sensor nodes during each session. When new sensors join a group or existing members leave the group, the group membership must be updated. In addition, when a compromised group member is detected, the compromised group member must also be removed from the group.

In the *group dissolution* phase, the sensor nodes in the group are not bound together anymore. If a group member does not receive the group key update messages in a period of time τ ($\tau > \Delta_s$), the key materials become obsolete and can be released.

IV. Group rekeying schemes for SGC in WSNs

In this section, we present four centralized group rekeying (CGK) schemes for secure group communication in WSNs. In Scheme 1 and Scheme 2, the group is initialized and set up by a sensor node in the group. In Scheme 3 and Scheme 4, the base station initializes and sets up the group.

A. Security model

We assume that there is a *secure channel* between the sensor node and the base station. By a *secure channel*, we mean a channel that offers confidentiality, data authentication, integrity, and freshness. The key materials to build the secure channel can be set up by the key management protocols described in [3], [18]. Our proposed schemes depend on some intrusion detection techniques [19], [20] to be used. If a sensor node in the group is compromised, the sensor node must be removed from the group and from the network [18], [21], [22]. We also assume that the base station is well protected and cannot be compromised. In the remainder of the section, we present our proposed schemes.

B. Scheme 1-unicasting

Scheme 1 is based on Blundo's theory [23]. A key tree is set up during the group formation process and the group key is distributed to the group members through unicasting (Figure 1). Scheme 1 consists of five message types, i.e., Interesting, Join Request, Leave Request, Forced Leaving, and Group Key. These messages are described below:

- Interesting, the message requesting expression of interest in a particular event.
- Join Request, a sensor wants to join the group.
- Leave Request, a group member wants to leave the group.
- Forced Leaving, the message contains the compromised sensor identifiers to be removed from the group.
- Group Key, the message to distribute the group key.

The complete process of the group formation phase in Scheme 1 is described below.

1) Setup: Before sensor nodes are distributed, the setup server randomly generates a bivariate *t*-degree polynomial $f(x,y) = \sum_{i,j=0}^{t} a_{ij}x^i y^j$ over a finite field F_q where *q* is a prime number that is large enough to accommodate a cryptographic key such that it has the property that $f(x,y) = f(y,x)$. For each sensor *i*, the setup server computes a polynomial share of $f(x,y)$, that is, $f(i,y)$, and loads the single-variate polynomial $f(i,y)$ to the sensor *i*. For any two sensor nodes *i* and *j*, node *i* can compute the common key $f(i,j)$ by evaluating $f(i,y)$ at point *j*, and node *j* can compute the same key $f(j,i) = f(i,j)$ by evaluating $f(j,y)$ at point *i*.
2) Broadcast interest: Once the group controller is identified, it first obtains a group identifier gid from the base station and then generates a random key K_G as the group key. Subsequently, the group controller broadcasts an Interesting message requesting expression of interest in a particular event E to its neighboring nodes. The Interesting message is flooded to the neighboring nodes which are reachable in at most *L* hops (global broadcasting is not necessary) (Figure 1∼(a)):

   $$I \longrightarrow * : ID_I|gid|E$$

   During the period when the Interesting message is broadcasted, an uplink pointer (from the sensor node to the group controller) is kept in each sensor node.
3) Join: All the receivers observing the same event E send a Join Request to the group controller *I* (Figure 1∼(b)):

   $$A \longrightarrow I : ID_A|gid|E, MAC(K_{AI}, ID_A|gid|E)$$

   where K_{AI} is the pairwise key shared by the group controller I with the sensor *A*. Since each sensor node keeps an uplink pointer to the group controller, the Join Request can be sent to the group controller along the uplink path. Meanwhile, during the period when the Join Request is transmitted, each en-route node also sets up a route table including the downlink information to the sensor nodes. Each item in the route table includes two fields: $\langle destination \rangle$, $\langle nexthop \rangle$. The group controller

also keeps track of the routing information for all group members.

4) Group key distribution: Once the group controller authenticates the join request, the group controller unicasts a Group Key message which includes the group key K_G encrypted by the pairwise key to the sensor A (Figure 1∼(c)):

$$I \longrightarrow A : \{K_G\}_{K_{AI}}$$

Since routing paths have been set up during the period when the Interesting message and Join Request messages are transmitted, the group controller can unicast the group key to each group member along the routing path.

In the group key maintenance phase, the group controller keeps track of the join and leave requests in the group. If a node receiving the Interesting message wants to join the group, it will send a Join Request to the group controller and the group controller can send the group key to the node as described in steps 3 and 4. If the node fails to receive the group key in time period Δ_s (it may happen due to the hardware failure of an en-route node), the sensor node can start a process called probing to reconstruct the key tree structure (Figure 1∼(d)). In the process of probing, the Join Request is broadcasted locally (at most L hops) among its neighboring nodes until it attains the group controller to repair the downlink path. Then, the group controller can send the group key to the sensor node.

In case a sensor node leaves the group, there are three situations [6]:

- Active leaving: a node notifies other group members of leaving before it moves or runs out of battery power.
- Passive leaving: a node fails silently due to hardware failure and does not notify other group members.
- Forced leaving: a node might be compromised and is forced to leave the group by the group controller.

In all three cases, if the leaving node is a leaf node in the key tree (a leaf node can identify itself according to the routing table), the key tree is not affected due to the leave of the sensor node. In case of active leaving, the leaving node sends a Leave Request to the group controller and the group controller can update the group key after receiving the request. If a sensor node fails silently due to hardware failure, it cannot participate the group after the group key is updated on the next session. In case a sensor node is compromised and forced to leave the group, the group controller will update the group key of non-compromised sensor nodes using unicasting messages as described in step 4.

If the leaving node is a parent node, the key tree needs to be reconstructed due to the leave of the sensor node. In case of active leaving, the leaving node sends a Leave Request to its parent node and each of its child node. The parent node updates its routing table (remove the leaving node entries) when it receives the Leave Request. Since each of the child node no longer has a parent node, the child node needs to start the probing process and rejoin the group as described before. In case of forced leaving, the group controller needs to update the group key of the remaining sensor nodes in the group. Since the child nodes of the leaving sensor lost their links to the group controller, the child nodes need to send Join Request messages to the group controller and rejoin the group. If a sensor node fails to receive a Group Key message in time period Δ_s due to the passive leaving of its parent nodes, the sensor node might be isolated from the key tree and needs to rejoin the group again.

Let $|G| = n$. Scheme 1 requires one local broadcast in the group formation phase. The group controller may receive n join requests and needs to send the group key to n members. Thus, to set up the group key among n members, it requires $2n$ unicasts and one local broadcast. To update the group key, it requires n unicasts of messages.

Note that Scheme 1 requires n unicasts of messages to update the group key which may cause heavy traffic in the area when the group size is large. Further, Scheme 1 has also to maintain the key tree whenever a sensor node joins or leaves the group. The management overhead of the key tree may get worse when the join and leave operations become frequent. We propose Scheme 2 which uses local broadcast to replace the unicasts to reduces the communication overhead when updating the group key.

C. Scheme 2-broadcasting

Scheme 2 is based on Blundo's theory [23] and the personal key share distribution scheme [24]. The key tree is used only for secret share distribution in the group formation process. After that, the group key is distributed through local broadcasting and there is no need to maintain the key tree structure. Scheme 2 consists six message types, i.e., Interesting, Join Request, Leave Request, Group Key, and Secret Share. The first five message types are the same with those of Scheme 1. The Secret Share message is used to distribute a personal secret in Scheme 2.

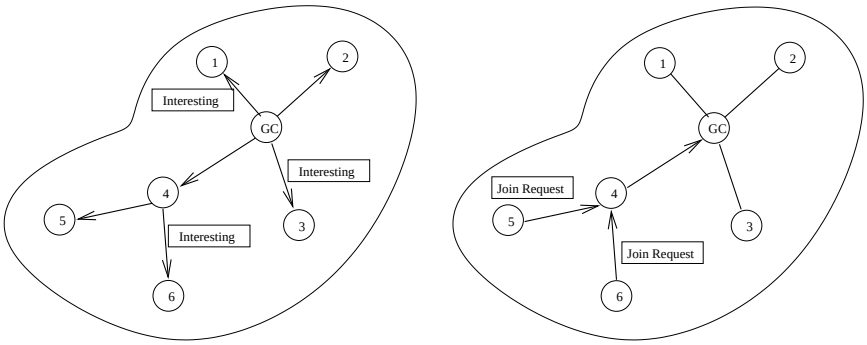
The group formation phase in Scheme 2 is described below. The setup, broadcast interest and join steps (1, 2, and 3) are the same as in Scheme 1 and are omitted.

4) Secret share distribution: The group controller randomly picks a $2t$-degree masking polynomial, $h(x) = h_0 + h_1x + \cdots + h_{2t}x^{2t}$, over F_q. Each group member A_i gets the personal secret, $S_i = h(i)$, from the group controller via the Secret Share message:

$$I \longrightarrow A_i : \{S_i\}_{K_{AI}}$$

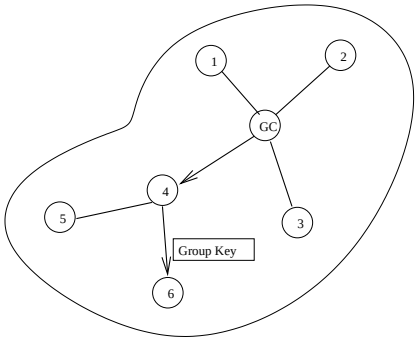
5) Distinct share broadcast: Given a set of IDs of revoked group members, $R = \{r_1, r_2, \cdots, r_w\}$, $w \leq t$, the group controller randomly picks a t-degree polynomial $p(x)$ and constructs $q(x) = K_G - p(x)$. Then, the group controller distributes the shares of the t-degree polynomials $p(x)$ and $q(x)$ to non-revoked sensors using the Group Key message which is broadcasted in the group:

$$\begin{aligned} B &= \{R\} \\ &\cup \{P(x) = g(x)p(x) + h(x)\} \\ &\cup \{Q(x) = g(x)q(x) + h(x)\} \end{aligned}$$

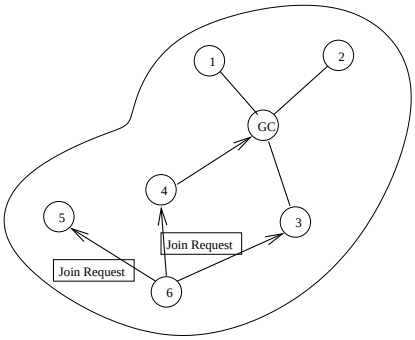


(a) Group controller (GC) floods out Interesting

(b) Interested nodes 5 and 6 send Join Request to group controller



(c) Group controller sends a Group Key to node 6



(d) Node 6 starts a process of probing

Fig. 1. Scheme 1 uses unicasting to distribute the group key.

where the revocation polynomial $g(x)$ is constructed as $g(x) = (x - r_1)(x - r_2) \cdots (x - r_w)$.

6) Group key recovery: If any non-revoked sensor node A_i receives such a broadcast message, it evaluates the polynomial $P(x)$ and $Q(x)$ at point i and gets $P(i) = g(i)p(i) + h(i)$ and $Q(i) = g(i)q(i) + h(i)$. Because A_i knows $h(i)$ and $g(i) \neq 0$, it can compute $p(i) = \frac{P(i)-h(i)}{g(i)}$ and $q(i) = \frac{Q(i)-h(i)}{g_j(i)}$. A_i can finally compute the new group key $K_G = p(i) + q(i)$. The revoked sensors cannot recover the group key because $g(x) = 0$.

In the group maintenance phase, the group controller repeats step 5 and 6 to distribute the group key during each session. If a sensor node wants to join the group, the sensor node sends a Join Request to the group controller. When the group controller receives the Join Request, it needs to send the secret share to the sensor node as described in step 4. Then, the sensor node can reveal the group key on the next session. If the sensor node fails to receive the group key share in time period Δ_s, the sensor node needs to start the probing process to find a path to the group controller.

Since the distribution of group key does not depend on the key tree in Scheme 2, the activate leaving and the passive leaving have no effect on the key tree. The leaving node does not need to notify the group controller. In case of forced leaving, the sensor identifier is added to the revocation list R in step 5 and thus the revoked sensor node cannot recover the group key and cannot participate the group communication.

Scheme 2 requires $2n$ unicasts and two local broadcasts to set up the group key among n members. However, to update the group key, it only requires one broadcast message. Since the distribution of the group key does not depend on the key tree, Scheme 2 greatly reduces the management overhead of the group.

Note that Scheme 1 and Scheme 2 are vulnerable to the flooding attack from insider attackers. Since each sensor node can potentially be a group controller, a sensor node must be capable of sending flooding messages. Although the flooding message is limited only in a local area (limited by the max-hops a flood message can traverse), an adversary can compromise a sensor node and abuse the power to start the flooding attack. Although we can adopt a rate control scheme [6] to relieve the flooding attack, it does not really solve the problem. In addition, the lack of efficient local broadcast authentication schemes in literature might also be a problem in Scheme 1 and Scheme 2. To protect the sensor network against the flooding attack and the impersonate of the group controller, we propose Scheme 3 and Scheme 4.

D. Scheme 3-overlapping

Scheme 3 is also based on the personal key share distribution scheme [24] but the base station is the actual group controller of the group. The Scheme 3 is inspired by the revocation scheme proposed in [22]. In [22], Wang *et al.* proposed a scheme utilizing a globally distributed session key to facilitate key revocation scheme. The session key can also be used to facilitate secure group communication. The group key can be generated using a function $MAC(K, M)$ over two secrets, a group key share K_s, and a session key K_j.

The lifetime of a WSN is partitioned into $sessions$. A session key is distribute to the sensor network during each session. We first describe the group formation process and the distribution of group key share K_s assuming a session key is used. Then, we describe the session key K_j distribution process.

1) Broadcast interesting: The base station broadcast an Interesting message to the sensor network:

$$B \longrightarrow * : gid|E$$

The authenticity of the broadcast message is ensured by broadcast authentication schemes such as $\mu Tesla$ [25] and its extensions [26], [27]. No one can impersonate a base station and broadcast an authenticated message.

2) Join: All the receivers observing the same event E send a Join Request to the base station:

$$A \longrightarrow B : ID_A|gid|E, MAC(K_{A,B}, ID_A|gid|E)$$

where $K_{A,B}$ is the pairwise key shared by the sensor node A with the base station B. The pairwise key can be set up using the Blundo's theory as described in Scheme 1.

3) Group key share distribution: Once the base station authenticate the join request, the base station unicasts the group key share K_s to the sensor A using Secret Share message:

$$B \longrightarrow A : \{K_s\}_{K_{A,B}}$$

along the routing paths set up during the transmission of the Interesting and the Join Request messages (similar process as described in Scheme 1).

4) Group key recover: Let K_j be the current session key. Each group member can calculate the shared group key K_G:

$$K_G = MAC(K_j, K_s)$$

The session key distribution is the same with the scheme in [22]. We briefly describe it below. The session key distribution scheme can be divided into three phases, viz., setup, broadcast, and session key recovery.

1) Setup: The setup server randomly picks m $2t$-degree masking polynomial, $h_j(x) = h_{j,0} + h_{j,1}x + \cdots + h_{j,2t}x^{2t}, j \in \{1, 2, \cdots, m\}$, over a finite field F_q where q is a sufficiently large prime number. For each sensor node A_i, the setup server loads the personal secrets, $\{h_1(i), h_2(i), \cdots, h_m(i)\}$, to the node A. The setup server also loads the polynomial, $h_j(x)$, to the base station. For each session key K_j, the setup server randomly picks a t-degree polynomial $p_j(x)$ and constructs $q_j(x) = K_j - p_j(x)$.
2) Broadcast: Given a set of revoked group members, $R = \{r_1, r_2, \cdots, r_w\}$, $w \leq t$ in session j, the base station distributes the shares of t-degree polynomial $p_j(x)$ and $q_j(x)$ to non-revoked sensors via the following broadcast message:

$$\begin{aligned} B &= \{R\} \\ &\cup \{P_j(x) = g_j(x)p_j(x) + h_j(x)\} \\ &\cup \{Q_j(x) = g_j(x)q_j(x) + h_j(x)\} \end{aligned}$$

where the revocation polynomial $g_j(x)$ is constructed as $g_j(x) = (x - r_1)(x - r_2)\cdots(x - r_w)$.

3) Session key recovery: If any non-revoked sensor node A_i receives such a broadcast message, it evaluates the polynomial $P_j(x)$ and $Q_j(x)$ at point i and gets $P_j(i) = g_j(i)p_j(i) + h_j(i)$ and $Q_j(i) = g_j(i)q_j(i) + h_j(i)$. Because A_i knows $h_j(i)$ and $g_j(i) \neq 0$, it can compute $p_j(i) = \frac{P_j(i) - h_j(i)}{g_j(i)}$ and $q_j(i) = \frac{Q_j(i) - h_j(i)}{g_j(i)}$. A_i finally can compute the new session key $K_j = p_j(i) + q_j(i)$.

In the group maintenance phase, if a sensor node wants to join the group, the senor node sends a Join Request to the base station. Then, the base station can send the group key share to the sensor node and thus the sensor node can recover the group key. In case of active leaving and passive leaving, the leaving node does not need to notify the group controller. When a sensor is forced to leave the group, there are two options to remove the sensor node from the group: updating the group key share or update the session key. Since the group key share can be updated only by unicasting, the solution is not efficient. However, the session key can be updated using a broadcast message in one round, thus, it is more efficient than updating the group key share.

Scheme 3 requires $2n$ unicasts and one broadcast message to set up the group among n members. Unlike Scheme 1 and 2, the unicasting messages are transmitted over the whole sensor network instead of a local area. Scheme 3 may take longer time to set up the group. To update the group key, Scheme 3 requires one broadcast message.

E. Scheme 4-preloading

Scheme 4 is a specialization of Scheme 3. Note that the group formation phase in Scheme 3 may take a long time due to the use of unicasting to distribute the group key share. If we know the group membership during the pre-distribution stage, we can load a group key share on the sensor nodes before the sensor network is deployed. Thus, the group formation phase in Scheme 4 can be simplified as the distribution of the session key only. As shown before, the session key distribution is efficient and fast. To add a new sensor node in the group, the setup server needs to load the same group key share to the sensor node. Then, the join and the leave can be operated as in Scheme 3.

Note that the solution of preloading a single mission key on the sensor nodes in the group does not work. Once a sensor node is compromised, the single mission key is exposed and cannot ensure the security of the group communication. In Scheme 4, although an adversary may compromise the sensor node to steal the group key share, the adversary cannot participate the group communication until it reveals the session key.

By preloading a group key share in sensor nodes before the sensor network is deployed, Scheme 4 greatly reduces the

group formation time. Scheme 4 requires only one broadcast message to set up the group and update the group key.

V. Security and performance analysis

A. Security analysis

In Scheme 1 and Scheme 2, the group controller is a sensor node. Thus, the sensor node must be granted local broadcast capability to start a group. However, the capability can be abused by an adversary to start inside flooding attacks. Further, since the group controller needs to broadcast a message to its neighboring nodes to request expression of interest, the authenticity of the local broadcast message must be ensured to avoid the impersonate of the group controller. However, most of proposed broadcast authentication scheme such as [25], [26], [27] are used for authenticating broadcast messages from the base station and cannot be used for local broadcast. Without an efficient local broadcast authentication scheme, the sensor nodes must be trustworthy in Scheme 1 and Scheme 2. In case a sensor node in the group is compromised, the adversary can know the group keys which it possesses. If the compromised sensor node is the group controller, the compromised sensor node may start revocation attacks to remove non-revoked sensor nodes from the group.

In Scheme 3 and Scheme 4, the base station is the group controller to initialize and start the group. Thus, it is not necessary to grant the local broadcast capability to sensor nodes. The authenticity of the broadcast message can be ensured by schemes such as [25], [26], [27]. Since the base station is trustworthy, no adversary can impersonate the base station and start the revocation attack either. In case a sensor node in the group is compromised, the adversary may steal the group key and eavesdrop the communication until the compromised sensor node is removed from the network.

B. Performance analysis

We use Blundo's theory [23] to set up the pairwise keys in Scheme 1-4. To set up the pairwise key, the sensor node needs to evaluate the polynomial value at point (i, j). The polynomial evaluation is efficient in computation and thus the additional computation overhead for calculating the pairwise key is almost negligible. To use Blundo's theory, each sensor node *i* needs to store a *t*-degree polynomial $f(i, x)$, which occupies $(t + 1) \log q$ storage space. In Schemes 2-4, the personal key share scheme is used. Each sensor node needs one additional storage unit for the personal secret and the group controller needs $(2t + 1) \log q$ storage unit for the masking polynomial. The communication overhead of the proposed schemes is compared in Table I.

Since Scheme 1 depends on a key tree to distribute the group key, Scheme 1 must maintain the integration of the key tree when sensors join or leave the group. In Schemes 2-4, the distribution of the group key does not depend on the key tree. Thus, it is more easier to handle the join and leave operations in Schemes 2-4.

VI. Simulation and results

The performance of the proposed schemes was evaluated in SENSIM [28], a component-based discrete-event simulator for sensor networks. Each sensor node in SENSIM consists of six components, i.e., app, net, mac, phy, event generator, and battery. The proposed schemes are implemented in the network component independently. In the simulation, all the packets sent to the MAC layer are guaranteed to be received at the receivers. Thus, no packet collisions are considered and the performance evaluated in the simulation are under ideal conditions.

TABLE II
Characteristic data for the Mica2 sensor platform.

Field	Value
Effective data rate	19.2kbps
Transmit power	36mW
Receive power	14.4mW
Idle power	14.4mW
Sleep	0.015mW
Transition power	28.8mW
Transition time	800μs

TABLE III
Group size and the max-hops in the simulation.

L (max-hops)	1	2	3	4	5	6	7	8
Group size	16	38	70	126	206	284	389	503

We consider all the schemes operating on a finite field F_q, where q is a 56-bit integer. The polynomial degree t in Blundo's theory is set to $t = 4$. We use the simulator parameters that represent the Mica2 Mote radio characteristics. These parameters are shown in Table II.

We assume that 1000 nodes are uniformly dispersed in a field with dimension $2000m \times 2000m$. The base station is located at $(2000, 2000)$ and the group controller in Scheme 1 and Scheme 2 is set at $(1088, 1151)$. The evaluation metrics include the group formation time, the group key update time, the energy consumption in group controller, and the energy consumption in group member nodes. The group formation time is the time duration from the group controller broadcasting the interest message till all the group members receive the first group key. The group key update time is the time period when the group controller updates the group key of all group members. It does not include the time when a sensor node joins or leaves the group.

We test the four schemes for different group sizes. The group size is decided by a maximum count (max-hops) along the routes in which the interest message is forwarded and we assume that all sensor nodes which hear the message become group members. For each group size, we run the simulation ten times and the average value is measured. Table III shows the group size and the max-hops in our simulation. The same group of sensor nodes is used for all the four schemes in each test scenario.

Table IV shows the group formation time as the number of max-hops increases. It shows that Scheme 2 and Scheme 3

TABLE I

COMPARISON.

		Scheme 1		Scheme 2		Scheme 3		Scheme 4	
	message	nums	size	nums	size	nums	size	nums	size
Group formation	unicast	$2n$	$O(\log q)$	$2n$	$O(\log q)$	$2n$	$O(\log q)$	0	n/a
	broadcast	1	$O(\log q)$	2	$O(t \log q)$	2	$O(t \log q)$	1	$O(t \log q)$
Group key update	unicast	n	$O(\log q)$	0	n/a	0	n/a	0	n/a
	broadcast	0	n/a	1	$O(t \log q)$	1	$O(t \log q)$	1	$O(t \log q)$

TABLE IV

GROUP FORMATION TIME (SECONDS). GROUP FORMATION IS TIME CONSUMING IN SCHEMES 1-3. BY PRELOADING A GROUP KEY SHARE IN SENOR NODES, SCHEME 4 GREATLY REDUCES THE GROUP FORMATION TIME.

L (max-hops)	1	2	3	4	5	6	7	8
Scheme 1	10.15	26.71	39.92	73.13	112.27	158.40	203.79	272.61
Scheme 2	10.94	28.35	42.53	75.98	115.96	162.82	208.82	278.27
Scheme 3	29.83	46.30	61.06	85.48	128.87	174.93	233.22	277.13
Scheme 4	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62

TABLE V

GROUP KEY UPDATE TIME (SCCONDS). SCHEME 3 AND SCHEME 4 ARE MUCH BETTER WHEN UPDATING THE GROUP KEY.

L (max-hops)	1	2	3	4	5	6	7	8
Scheme 1	8.10	19.81	37.35	63.88	107.25	144.27	197.55	253.99
Scheme 2	0.61	1.58	2.17	3.19	3.55	4.05	4.69	5.36
Scheme 3	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62
Scheme 4	0.62	0.62	0.62	0.62	0.62	0.62	0.62	0.62

require more time to set up the group than Scheme 1. Although the group formation phase is similar in Scheme 1 and Scheme 3, Scheme 3 takes longer because the whole sensor network is involved in the group formation phase. Due to the transmission of additional key materials in Scheme 2, Scheme 2 needs more time than Scheme 1 to set up the group. Scheme 2 is even worse than Scheme 3 when the max-hops is greater than eight. By preloading a group key share in sensor nodes, Scheme 4 can greatly reduce the group formation time. The group formation time in Scheme 4 is equal to the broadcast message transmission time in the network. Further, we notice that it takes a long time ($> 1min$) for Schemes 1-3 to initialize the group when the number of max-hops is greater than three. It indicates that the number of max-hops on routes which the interest messages are allowed to traverse should be less than four. Table V shows the group key update time in the group maintenance phase. By using the broadcasting instead of the unicasting to distribute the group key, Schemes 2-4 are much better than Scheme 1. Scheme 3 and Scheme 4 use the same group key update process and thus have the same group key update time. Scheme 2 consumes more energy than Schemes 3 and 4 because it requires more communication rounds (L) to flood the message.

Figure 2 shows the average group controller energy consumption in the group formation and the group key update phrases. Since the base station takes the role of the group controller in Schemes 3 and 4, the energy consumption of Scheme 3 and Scheme 4 is not shown in the figures. As the figures indicate, although Scheme 1 requires less energy for the group controller to set up the group, the group controller in Scheme 1 consumes much more energy to update the group key. Because the group key is updated at regular time intervals, Scheme 1 may cause the group controller to deplete its energy much faster than Scheme 2.

Tables VI and VII show the average group member energy consumption in the group formation and group key update phases. As the tables show, Scheme 1 is slightly better than Schemes 2-3 in the group formation phase but Schemes 2-3 are far better than Scheme 1 in the group key update phase. Due to the preloading of the group key share in sensor nodes, Scheme 4 performs the best in both the group formation and the group key update phases.

Figure 3 shows the energy distribution among group members when the number of max-hops is three. As the figures show, Scheme 1 may cause the energy to be distributed unevenly in the group formation phase. However, the energy is distributed more evenly in Schemes 2 and 3 in both the group formation and group update phases. Scheme 4 has no such issues due to the use of broadcasting messages.

To summarize, with respect to the group formation time and the energy consumption in the group controller and the group member sensor nodes, Scheme 1 is slightly better than Schemes 2-3 in the group formation phase; however, Schemes 2-3 are far better than Scheme 1 in the group key update phase. In the simulation, we use a simple flooding protocols to set up the route path in Scheme 3. In consideration that the routing paths might be set up before the group formation phase and the setup of the routing paths can also benefit the data acquisition in the network, Scheme 3 is a better solution for secure group communication in sensor networks. As the simulation results show, if we can group the sensor nodes together before the sensor network is deployed, Scheme 4 is the best selection for secure group communication in sensor networks. Note that the performance of Scheme 1 represents a general category of secure group communication schemes using a key tree structure.

TABLE VI

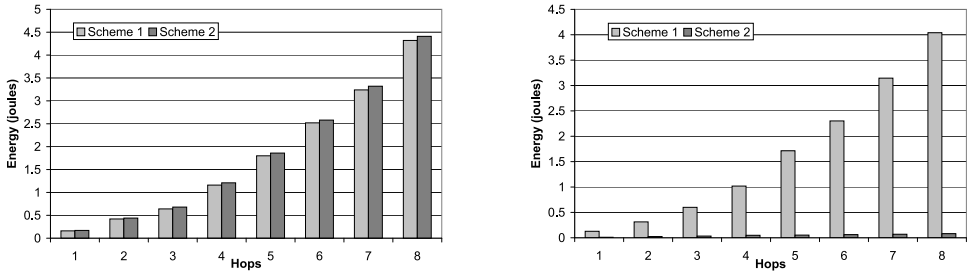
AVERAGE GROUP MEMBER ENERGY CONSUMPTION (JOULES): GROUP FORMATION PHASE.

L (max-hops)	1	2	3	4	5	6	7	8
Scheme 1	0.15	0.39	0.58	1.06	1.63	2.29	2.95	3.94
Scheme 2	0.16	0.41	0.62	1.10	1.68	2.36	3.02	4.02
Scheme 3	0.42	0.66	0.86	1.23	1.85	2.52	2.35	4.00
Scheme 4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01

TABLE VII

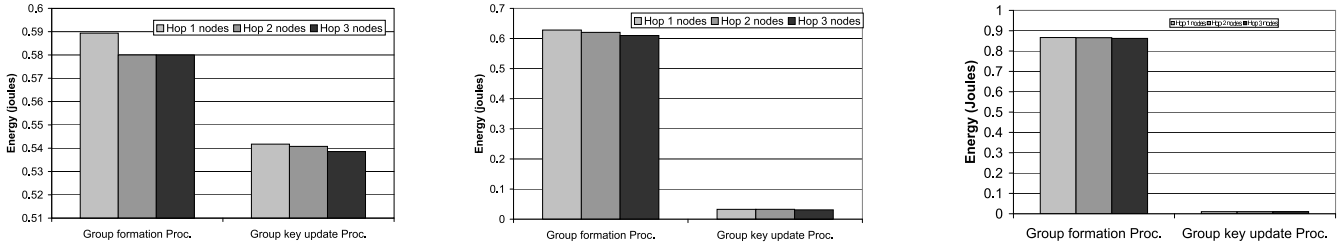
AVERAGE GROUP MEMBER ENERGY CONSUMPTION (JOULES): GROUP KEY UPDATE PHASE.

L (max-hops)	1	2	3	4	5	6	7	8
Scheme 1	0.12	0.29	0.54	0.92	1.55	2.08	2.85	3.67
Scheme 2	0.01	0.02	0.03	0.05	0.05	0.06	0.07	0.08
Scheme 3	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Scheme 4	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01



(a) Group formation phase. (b) Group key update phase.

Fig. 2. Average group controller energy consumption (Joules). Scheme 1 requires less energy for the group controller to set up the group, however, the group controller in Scheme 1 consumes much more energy to update the group key.



(a) Scheme 1 (b) Scheme 2 (c) Scheme 3

Fig. 3. Average group member energy distribution by hops (Joules). Scheme 1 may cause the energy to be distributed unevenly in the group formation phase.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed four centralized group rekeying (CGK) schemes for secure group communication in WSNs. As the simulation and analysis show, if the group membership can be decided before the sensor network is deployed, Scheme 4 is the best selection for secure group communication in sensor networks. If the group membership is decided dynamically after the sensor network is deployed, Scheme 3 is a better option. Due to the lack of the efficient local broadcast authentication schemes in sensor networks, Schemes 1 and 2 are vulnerable to the flooding attack from inside attackers.

As the simulation shows, by distributing a session key to the whole sensor network, the group key update phases is efficient. However, the group formation phase may take a long time $(> 2mins)$ when the number of max-hops is great than four. The group formation phase needs to be improved when the group membership is decided dynamically. Further, efficient local broadcast authentication schemes are also needed for

local broadcast in the sensor networks.

ACKNOWLEDGEMENTS

This work is partially supported by NSF Grant No. CCR-0311577.

REFERENCES

[1] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach," in *Proceedings of IEEE INFOCOM*, 2004, pp. 629–640.

[2] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of 33rd Annual Hawaii International Conference on System Sciences*, January 2000.

[3] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 62–72.

[4] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 2, 2006.

[5] X. Zou, B. Ramamurthy, and S. S. Magliveras, *Secure Group Communications Over Data Networks*. Springer, 2005.

[6] J.-H. Huang, J. Buckingham, and R. Han, "A level key infrastructure for secure and efficient group communication in wireless sensor network," in *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 249–260.

[7] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," in *Proceedings of IEEE INFOCOM*, 2004.

[8] S. Zhu, S. Setia, S. Jajodia, and P. Ning, "An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, USA, May 2004, pp. 259–271.

[9] N. Thepvilojanapong, Y. Tobe, and K. Sezaki, "A proposal of secure group communication for wireless sensor networks," in *The 23th Computer Security (CSEC) Group Meeting, IPSJ*, Tokyo, Japan, Dec. 2003, pp. 47–52.

[10] M. Tubaishat, J. Yin, B. Panja, and S. Madria, "A secure hierarchical model for sensor network," *SIGMOD Rec.*, vol. 33, no. 1, pp. 7–13, 2004.

[11] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, pp. 16–30, 2000.

[12] D. Balenson, D. McGrew, and A. Sherman, "Key management for large dynamic groups: One-way function trees and amortized initialization," IETF Internet draft, August 2000.

[13] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman key distribution extended to group communication," in *CCS '96: Proceedings of the 3rd ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 1996, pp. 31–37.

[14] Y. Kim, A. Perrig, and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in *CCS '00: Proceedings of the 7th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2000, pp. 235–244.

[15] R. D. Pietro, L. V. Mancini, Y. W. Law, S. Etalle, and P. J. M. Havinga, "LKHW: A directed Diffusion-Based secure multicast scheme for wireless sensor networks," in *ICPPW '03: Proceedings of the 32nd International Conference on Parallel Processing Workshops*. IEEE Computer Society Press, 2003, pp. 397–406.

[16] W. Zhang and G. Cao, "Group rekeying for filtering false data in sensor networks: a predistribution and local collaboration-based approach," in *Proceedings of IEEE INFOCOM*, March 13-17 2005.

[17] Y. Xue, B. Ramamurthy, and Y. Wang, "Providing loss tolerant reliable data transport services on wireless sensor networks," 2007, under review.

[18] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *CCS '02: Proceedings of the 9th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2002, pp. 41–47.

[19] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 255–265.

[20] G. Wang, W. Zhang, C. Cao, and T. L. Porta, "On supporting distributed collaboration in sensor networks," in *Proceedings of MILCOM*, 2003.

[21] H. Chan, V. Gligor, A. Perrig, and G. Muralidharan, "On the distribution and revocation of cryptographic keys in sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, no. 3, pp. 233–247, July-Sept. 2005.

[22] Y. Wang, B. Ramamurthy, and X. Zou, "KeyRev: An efficient key revocation scheme for wireless sensor networks," in *ICC '07: Proceedings of IEEE International Conference on Communications*, Glasgow, Scotland, U.K., June 2007.

[23] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*. London, UK: Springer-Verlag, 1993, pp. 471–486.

[24] D. Liu, P. Ning, and K. Sun, "Efficient self-healing group key distribution with revocation capability," in *CCS '03: Proceedings of the 10th ACM conference on Computer and communications security*. New York, NY, USA: ACM Press, 2003, pp. 231–240.

[25] B. Przydatek, D. Song, and A. Perrig, "SIA: secure information aggregation in sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*. New York, NY, USA: ACM Press, 2003, pp. 255–265.

[26] D. Liu and P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," in *Proceedings of the 10th Annual Network and Distributed System Security Symposium*, San Diego, CA, USA, February 2003, pp. 263–276.

[27] D. Liu and P. Ning, "Multi-level μTESLA: Broadcast authentication for distributed sensor networks," *Trans. on Embedded Computing Sys.*, vol. 3, no. 4, pp. 800–836, 2004.

[28] Y. Wang and B. Ramamurthy, "SENSIM: SEnsor Network SIMulator (Version 0.1)," August 2006. [Online]. Available: http://cse.unl.edu/∼ywang/sensim.htm