A Proposal of Secure Group Communication for Wireless Sensor Networks

Niwat Thepvilojanapong[†], Yoshito Tobe[‡], Kaoru Sezaki[†]

[†]Institute of Industrial Science, University of Tokyo
[‡]Graduate School of Engineering, Tokyo Denki University
E-mail: [†]{niwat, sezaki}@iis.u-tokyo.ac.jp, [‡]yoshito@unl.im.dendai.ac.jp

Abstract *We propose a secure group communication protocol for wireless sensor networks. Sensor network we consider consists of a large number of wireless sensors randomly distributed in an interesting area and a small number of base stations connected to wired network. To exploit inherent characteristic of wireless communication, sensor nodes are classified into three types according to capability to communicate with the base station. Logical hierarchy is then created for bidirectional communication between base station and sensor nodes, i.e., gathering and disseminating data. Based on a combination of symmetric key encryption and public-key encryption algorithm, both base station and sensor nodes can confidentially communicate and authenticate communicating party. We also propose a method to manage tree and keys when members join or leave from the group to achieve robustness of protocol.*

1 Introduction

Sensor networks have emerged as a fundamentally new tool for monitoring environments. Sensor nodes coordinate amongst themselves to achieve a larger sensing task both in urban environments and in inhospitable terrain. Integrated low-power sensing devices will permit remote object monitoring and tracking in many different contexts: in the field (vehicles, equipment, personnel), the office building (projectors, furniture, books, people), the hospital ward (syringes, bandages) and the factory floor (motors, small robotic devices). Many literatures focus on routing, querying, rate control, including optimizations like aggregation and compression. However, there is little consideration in security issue.

The process of monitoring tasks in sensor networks can be divided into three phases. The first phase is to gather data by sensing. The purpose of emerging sensor technology is to gather environmental data, *e.g.*, temperature, light intensity, humidity, velocity, and so on. Communication in this phase can be considered as *concast*, *i.e.*, a large number of sensor nodes transmit sensed data to the base station. The next phase is to process gathered data. After receiving sensed data from sensor nodes, base stations analyze such data, *e.g.*, summarizing sensed data, finding average value in a specific area, detecting abnormality, and so on. The last phase is to send feedback to sensor nodes. The example of feedback is the frequency of sensing (once a minute, once an hour, once a day, *etc.*), the type of sensed information (only temperature or both temperature and humidity, *etc.*), upgrading software, and querying. Almost of communications in this phase are *multicast*, but it may be considered as *unicast* which is a rare case (query to a specified sensor node). Sensor networks are useful for monitoring and tracking in many different contexts. The example of ongoing work is habitat monitoring on Great Duck Island [1].

To motivate the challenges in designing sensor networks, consider the following scenarios. Several thousand sensors are deployed in remote terrain. The sensors coordinate to establish a communication network, monitor specified task, report monitored data periodically or on-demand, re-organize upon sensor failure. When additional sensors are added or old sensors fail, the sensors re-organize themselves to take advantage of the added system resources. The solutions in these challenges were described in many literature including routing protocol, energy minimization, *etc.* However, adversaries may invade into monitoring terrain and deploy same type of sensors to do malicious actions by trying to access the network, capture the packet, impersonate legitimate sensors, and attack the system (DoS attack, *etc.*). Therefore security is another challenge issue in sensor network. It can be achieved by introducing conventional cryptographic algorithms (*e.g.*, symmetric-key, public-key encryption algorithm) into sensor networks. Sensors, however, are easily stolen because they are small sensing devices deployed in the wide area without defense system. Adversaries can analyze stole sensors in order to achieve cryptographic keys and use such keys to access the communication of target network. This kind of vulnerability should be considered when designing protocol for sensor networks.

The remainder of the paper is organized as follows. Section 2 describes notation used in this paper. Section 3 proposes architecture for secure group communicaiton. Section 4 presents a scheme for building tree used in multicasting and concasting which are described in Section 5 and 6 respectively. Related work is discussed in Section 7. Finally, we conclude this paper and describe future works in Section 8.

2 Notation

Public and private key used in our protocol are keys according to public-key encryption (asymmetric encryption) scheme, *e.g.*, RSA, Rabin, ElGamal, McEliece, Knapsack public-key encryption, and so on. Public key is not announced to public as signature scheme; it is kept confidential. Symmetric key is a key according to symmetric-key encryption scheme, *e.g.*, DES, AES. We use the following notations to describe our protocol especially cryptographic operations in this paper.

$K_{sym,grp}$ is shared symmetric key for individual group.
$K_{pub,bs}$ is base station's public key.
$K_{pri,bs}$ is base station's private key.
$K_{sym,sn}$ is individual sensor node's symmetric key.
$E(K, M)$ is the encryption of message M with key K.
$X|Y$ denotes the concatenation of X and Y.
IV is initialization vector which is block of random data. A timestamp can be used in our protocol.

3 Architecture

We consider sensor networks consist of a large number of wireless sensors randomly distributed in an interesting area and a small number of base stations (BSs) or sink machines connected to wired network. These sensors have limited processing power, storage, bandwidth, and energy, while base stations have powerful resource to collect sensed data from sensors, process data, and transmit raw data or processed data to central machine which is controlled by user. User can remotely access base stations via central machine. Any base station can act as a central machine or dedicated machine may be used in order to distribute load. Moreover, central machine can change between each base station anytime. This means that user can use nearest base station as central machine.

Generally, sensor nodes have omni-directional antennas and use RF to communicate. All wireless network transmissions are inherently broadcast and a host may be able configure its network interface into promiscuous receive mode. Our protocol exploits this property to minimize the energy consumption. On the other hand, we also consider security issues mentioned in Section 1.

Sensor nodes are classified into three types according to capability in communicating with the base station (Figure 1).

- **One-hop to base station (OHB) node.** OHB nodes are sensor nodes which can send packet to the base station within one hop.
- **One-hop to sensor (OHS) node.** OHS nodes are sensor nodes which can receive packet from the base station within one hop but they cannot send packet directly to the base station (within one hop) because of shorter communication range compared with base station.
- **Out of range (OFR) node.** OFR nodes are sensor nodes which are outside communication range of the base station.

Before sensor nodes are deployed in the field by any method (*e.g.*, deployed by hand, thrown from an aircraft), every sensor node receives $K_{sym,sn}$, $K_{sym,grp}$, and $K_{pub,bs}$ in advance for doing cryptography.

3.1 Node Classification

First, every sensor node is set as *OFR node* before deployment. Next, base station broadcasts HELLO packet with group ID which is defined in advance. The timer t_{cf} is also set and sent along with this packet. This timer is used for determining the period of classification phase. Every packet has same format: $\langle$*type_id*, ID_{src}, ID_{dst}, ID_{grp}, $E(K, IV \mid$ *type_id* $\mid ID_{src} \mid ID_{dst} \mid ID_{grp} \mid$ *seq* $\mid$ *len* $\mid$ *data*$)\rangle$. The *type_id* field is type of packet such as HELLO (1X), REPLY (2X), NOTICE (3X), and so on. ID_{src}, ID_{dst}, and ID_{grp} are source ID, destination ID, and group ID respectively. The *seq* field is sequence number of packet. The *len* field is total length of message which will be encrypted by K.

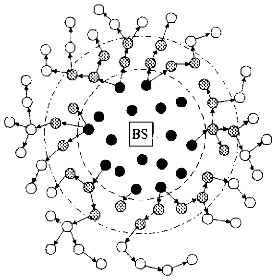

Figure 1: Nodes are classified as OHB nodes (black), OHS nodes (grey), and OFR nodes (white). Then tree is securely created for group communication.

Sensor nodes determine action after receiving packet from *type_id* field which is revealed (unencrypted). For instance, they immediately drop packets which came from other communication group (ID_{grp}). These revealed fields do not lead to vulnerability like substituting message or replay attack because the message is encrypted including *seq* field.

According to Table 1, base station encrypts HELLO packet (*type_id* = 11) by group key ($K_{sym,grp}$). Recipients use group key received in advance for decrypting and send REPLY packet (*type_id* = 21) encrypted by their own key ($K_{sym,sn}$) to the base station. Then they change from OFR nodes to *OHS nodes*. However, some replies may not arrive at the base station because of shorter communication range of sensor nodes compared to the base station (Figure 2).

After receiving REPLY packet, base station uses key corresponds to revealed source ID to decrypt the message. Base station uses a list of key-ID pair as *access control list* (ACL) for each group. If decryption succeeds and decrypted fields match with revealed fields, base station accepts that node as a group member.

Then, base station announces the list of *OHB nodes* by broadcasting MSG packet (*type_id* = 41) encrypted with its private key.

Because of unstable in wireless communication, packets may lose and node classification will be wrong. There is probability that OHB node can be OHS node (missed list of OHB nodes), OHB node can be OFR node (missed both HELLO packet and list of OHB nodes), and OHS node can be OFR node (missed both HELLO packet and list of OHB nodes). Even though such problems occur, group communication can still perform with degraded efficiency because

Table 1: Types of packet. Basically, the format of packet is $\langle$*type_id*, ID_{src}, ID_{dst}, ID_{grp}, $E(K, IV \mid$ *type_id* $\mid ID_{src} \mid ID_{dst} \mid ID_{grp} \mid seq \mid len \mid data)\rangle$.

Type	ID	Description	Encrypting key	*data* field
HELLO	11	Base station invites members.	$K_{sym,grp}$	NULL
	12	Leaf node invites unattached nodes; group member looks for alive leaf nodes.	$K_{sym,grp}$	NULL
	13	Joining node looks for parent node.	$K_{sym,grp}$	NULL
REPLY	21	Joining node declares to join the group.	$K_{sym,sn}$	NULL
	22	Joining node declares to be a child node.	$K_{sym,grp}$	$E(K_{sym,sn}, IV \vert ID_{bs})$
	23	Group member declares to be parent node.	$K_{sym,grp}$	NULL
NOTICE	31	Parent node informs joining node of child status.	$K_{sym,grp}$	$E(K_{sym,sn}, IV \vert ID_{sn})$
	32	Joining node informs parent node of its joining.	$K_{sym,grp}$	$E(K_{sym,sn}, IV \vert ID_{bs})$
MSG	41	Base station announces the list of OHB nodes.	$K_{pri,bs}$	OHB node IDs
	42	Node sends the message.	$K_{sym,grp}$	sensed data, command, *etc.*
	43	Base station privately communicates with sensor.	$K_{sym,sn}$	command, *etc.*

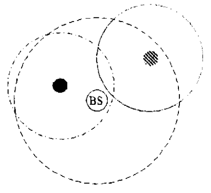

Figure 2: Base station has longer communication range than sensor nodes.

of increasing packets. However, based on promiscuous receive mode, sensor nodes will recover to correct type automatically. At the later time, if OFR nodes directly receive packet from the base station, it will change to OHS nodes. If base station hear packet from OHS node, it will inform that node to change to OHB node.

4 Establishing the Tree

After the classification of nodes completed (*i.e.*, timer t_{cf} is over, where t_{cf} is the timer that indicates the end of node classification phase), every OHB node broadcasts HELLO packet (*type_id* = 12) encrypted with group key to find child nodes. The recipients decrypt the packet and can verify whether packet was sent from legitimate nodes in the group. Any sensor node except OHB node which receives this packet will send REPLY packet (*type_id* = 22) back to the sender in order to inform that they will be child nodes of that sender, *i.e.*, they will be leaf nodes of current tree where root nodes are OHB nodes. Sensor nodes may receive many HELLO packets but they reply to only first arrived HELLO packet and ignore other incoming packets. Another metric to be used to choose the parent node is the strength of received power. They will ignore packet whose power is below the threshold.

To prove the identity of new joining nodes, they encrypt arbitrary message (for example, ID_{bs}) with their individual key and attach in the *data* field as *authentication tag*. Parent node encrypts child node ID and encrypted message with its individual key. Then it sends *challenge* message, $\langle E(K_{sym,parent}, IV \mid ID_{child} \mid E(K_{sym,child}, IV \mid ID_{bs}))\rangle$, to the base station by using MSG packet (*type_id* = 42). Base station decrypts the message with a key corresponds to source ID field ($K_{sym,parent}$), then it uses a key corresponds to node ID specified in decrypted message ($K_{sym,child}$). If the decryption passes, joining node is a legitimate sensor node. Base station sends the result to parent node by using MSG packet (*type_id* = 42) including node IDs along the path until parent node for forwarding purpose. The result is encrypted with destination node's key, thereby, the *data* field is $\langle ID_1 \mid ID_2 \mid \ldots \mid ID_n \mid E(K_{sym,sn}, \text{result})\rangle$. Before forwarding the packet, sensor node deletes its ID from the list. If the authentication passes, parent node accepts joining node as a child node, otherwise it expels that joining node. However, parent nodes notify the acceptance of new child nodes before authentication message arrives by using NOTICE packet (*type_id* = 31). They expel new joined nodes after receiving a notice of failed authentication.

Let t_{tr} be the timer that indicates the period for creating tree in each level. Sensor nodes set timer t_{tr} after receiving HELLO packet (*type_id* = 12). When this timer expired and they got parent node, they (current leaf nodes) look for new joining nodes by broadcasting HELLO packet (*type_id* = 12). Sensor nodes which do not attach to the tree yet reply to the first arrived HELLO packet and procedures are done as mentioned above until no any reply arrives at the current leaf nodes. Thereby, we get completed trees for group communication which is rooted at many OHB nodes (Figure 1).

Base station know logical topology of trees by multicasting topology request packet (*type_id* = 42). Every sensor node replies with the list of their child nodes. After base station received reply from every sensor node, it creates logical topology and knows at which level sensor nodes are. This information is useful for tree maintenance. Base station can check the status of sensor nodes and tree structure by periodically sending topology request packet and frequently updating the database.

5 Multicast for Sensor Networks

As stated in Section 1, Almost of communications in data dissemination phase are multicast communication. Our multicast protocol exploits inherent characteristic of wireless communication in order to minimize energy consumption. The communication uses group key for reinforcing security, *i.e.*, base station uses MSG packet (*type_id* = 42). It encrypts *data* field with its private key, encrypts entire message with group key, and multicasts by setting group ID as destination ID. OHB nodes and OHS nodes can receive multicast packet directly. Only OHS nodes which have OFR nodes as their child nodes multicast received packet along the tree. OFR nodes also multicast received packet to their child nodes. They multicast the original message only if the decrypted fields match revealed fields, otherwise the packet is dropped. If they receive duplicate packet (same sequence number), they also drop the redundant packet. Proposed multicast protocol effectively minimizes energy consumption, especially OHB nodes and OHS nodes as shown in Figure 3 (using tree created in Figure 1).

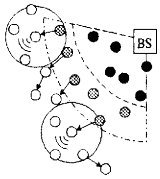

Figure 3: Multicast communication. Only some sensor nodes forward received packet to their child nodes.

5.1 Members Join and Leave

Joining means user deploys new sensor into the current network. Before deploying new sensor, user puts current group key and necessary keys ($K_{sym,sn}$, $K_{pub,bs}$) along with the sensor. Newly deployed sensor must find parent node in order to access group communication. After deploying, it broadcasts HELLO packet (*type_id* = 13) which is encrypted with group key. Any node in the tree (including base station) which receives HELLO packet replies with REPLY packet (*type_id* = 23). As usual, a node which replies first will be chosen as parent node. The priority of becoming parent node is OHB node, OHS node, and OFR node respectively. If joining node receives a reply from base station, parent node is not necessary in this case and it does not send reply to any node. It will be root node and only one node in the tree. If joining node does not receive any REPLY packet, this means that no other nodes are in their communication range or nodes in their communication range do not attach to the tree yet. In this case it wait for incoming packet, *i.e.*, any type of HELLO packet.

Joining node sends NOTICE packet (*type_id* = 32) to inform chosen parent. The procedures in proving identity of new joining node are same as in Section 4 when new sensor node joins the communication group. Joining node attaches challenge message ($E(K_{sym,sn}, ID_{bs})$) to the NOTICE packet. Parent node then sends the challenge message to the base station and waits for a result.

Another issue in joining phase is node classification (OHB, OHS, or OFR node). Joining node is OFR node by default. Based on communication in joining procedures, if it receives a reply from the base station, it will become OHB node. It means that its HELLO packet arrives the base station and we can infer that base station is in its communication range. Unless it receives a reply from the base station, it still is OFR node. There is no problem in communication for OFR node which should be OHS node, since it has attached to the tree. Therefore, it can do concast by sending packet to parent node (Section 6) and receive multicast packet from its parent node. The classification of OHS node and OFR node is used for multicasting, not for concasting. When base station multicasts the packet, if it hears multicast packet, it will become OHS node. After base station multicasted the packet, parent node will ask children whether they received the packet. If they do not receive the packet, they still are OFR nodes.

On the other hand, *leaving* means sensor node lost communication with other sensor nodes. For instance, the battery depleted, sensor node broke down, malfunctioned or was stolen. It is not necessary to update group key in this case. Since, left sensor node cannot access current communication any more for all of above mentioned causes of loss. In fact, left sensor node is legitimate sensor nodes deployed by user. Therefore, messages in the future are not confidential for it.

Here, the problem is how to maintain broken tree when sensor node left the network. We propose simple scheme to deal with this problem. First of all, we have to detect broken link. As natural characteristic of sensor networks, sensor node periodically transmits sensed data to the base station. Therefore, if parent node does not receive packet for a while from any child node, we infer that such node has left from the network. Another method is to detect broken link by using the underlying MAC-sublayer protocol.[1] The last method can be done by the base station. Only one left node causes losses of data from itself until the leaf nodes. Therefore, base station perceives that many nodes left from the group. However, after comparing left node IDs with known logical topology (Section 4), it know a real left node which is a cause of other left nodes, *i.e.*, a root node of losses. We call other left nodes, which are alive and really working, *temporarily left nodes* (TL nodes).

Tree maintenance is divided into two cases depending on first-level TL node whether it is OHS node or OFR node.[2] In the former case, base station can directly communicate with the first-level TL node as shown in Figure 4 and 5. First, base station informs first-level TL nodes to find new parent by using MSG packet (*type_id* = 43) encrypted with those nodes' individual key. The following steps are same as when new sensor node joins the group. In addition, they do not choose their current children as new parent. Second- and next-level TL nodes do noth-

[1] Our protocol is independent of MAC-sublayer protocol.

[2] OHB node cannot be TL node since it must be root node of the tree, it cannot be child node.

ing, they still attach to the first-level TL node. If all first-level TL nodes get new parent node, tree maintenance finishes. However, some first-level TL nodes may not have new parent in the case of sparse sensor networks. Such nodes choose one of their children as parent and inform it of new relation. They also inform other children to find new parent. Then, the second-level TL nodes which receive notice from their former parent find new parent by using same procedures. This step is repeatedly done until tree maintenance finishes. Node B in Figure 4 is perceived as a left node. Therefore, its children until leaf nodes cannot communicate with base station. Base station directly informs node C and D to find new parent. Node C receives reply from node E and G, while node D receives reply from only node F. Hence, Node C chooses to attach to node E and node D chooses node F as new parent. Figure 5 shows the case that first-level TL node does not find any new parent. Node A is a left node, then node B and C must find new parent. Node C gets node D as new parent. Node B cannot find new parent within its communication range. Consequently, it informs node E and F to find new parent and it chooses node E as its new parent. Node F broadcasts HELLO packet (*type_id* = 13) and receives reply from node H and I, then it chooses node H as its new parent. Node E does the same thing and gets node G as its new parent. Then, tree maintenance finished and group communication can perform again.

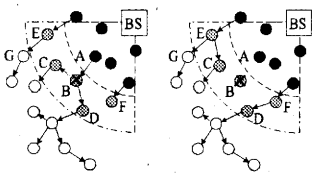

Figure 4: Node B left from the network. First-level TL nodes are Node C and D which are OHS nodes. Node C and D got node E and F as new parent respectively.

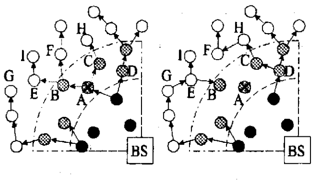

Figure 5: First-level TL nodes (Node B and C) must find new parent. Node C got node D as new parent node but no any node is in communication range of node B. Therefore, node B chooses one of its children as parent.

In the event that first-level TL node is OFR node, base station cannot directly communicate with it. Consequently base station cannot inform such node to find new parent. In this case, base station informs *surrounding nodes* which are nodes being in the same-level, one-upper-level, and one-lower-level compared with the first-level TL nodes to find new child. It uses MSG packet (*type_id* = 43) to inform OHB and OHS nodes. For OFR nodes, it adds list of nodes along the path from itself into MSG packet (*type_id* = 42) and uses it for informing. Surrounding nodes do the same procedures as when leaf node looks for children. If base station does not receive proof message $(E(K_{sym,sn}, ID_{bs}))$, it assumes that first-level TL node does not resurrect yet. Base station moves target down one level (second-level TL nodes) and informs surrounding nodes compared with second-level TL nodes to find new parent. Base station repeats this step until the leaf nodes or tree successfully resurrects. If all of TL nodes do not resurrect, it means that they have no neighbor node within their communication range or they may be real left nodes too (more than one left node in this case). The example is shown in Figure 6. Node B is a left node, then node E and F which are OFR nodes must find new parent. Surrounding nodes (node J and H as same-level, node K and C as one-upper-level, and node G as one-lower-level node compared with node E and F) broadcast HELLO packet to find new child by setting destination address to be node E and F.[3] Node E received HELLO packet from node G and node F received HELLO packet from node H. They reply to be child of those nodes, therefore, tree maintenance finishes.

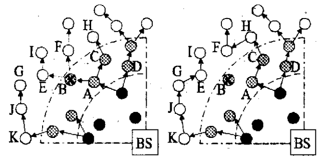

Figure 6: First-level TL nodes are node E and F which are OFR nodes. Surrounding nodes (node J, H, K, C, and G) try to recover node E and F.

To prevent physical attack, left node's ID and key are deleted from ACL immediately. This means that adversary who takes sensor away and analyzes it to find keys cannot use that individual key to join group communication. We hope that base station will be aware of such left node before it comes back with malicious action or code.

5.2 Key Update

We can increase the level of security by updating group key frequently. Yang *et al.* showed that periodic batch rekeying give more reliable for conventional multicast [2]. The simplest updating scheme is using hash function (*i.e.*, MD5, SHA-1) to derive new key. However, we propose to use forward security scheme for updating the key because information in the past is still secret though current key is broken [3], [4].

There are two methods for distributing new derived key. First, base station can derive new group keys and distribute them by using secure multicast proposed in Section 5 (using current group key for encryption). Another method is

[3] If they do not set destination address, *i.e.*, just broadcast, they may findnew deployed nodes which do not attach to the tree yet.

to insert algorithm into every sensor which requires additional memory resources and processing power but sensor node can derive key by itself. Time synchronization is required in this case, *i.e.*, every sensor should use same group key at the same time. However, this is not a serious issue because if decryption fails it can try again with new key or old key and adjust their clock. Moreover, there are many works proposed techniques to deal with this problem. We propose to use the former for distributing keysbecause time synchronization is not required. Sensor can change to new key immediately after receiving it.

6 Concast for Sensor Networks

Concast uses communication tree in reverse direction compared with multicast to carry sensed data until the base station (Figure 1). Tree creation in OHS zone is used for concasting because OHS node cannot send packet to base station within one hop. Sensor node uses MSG packet (*type_id* = 42) for concasting. It encrypts sensed data with its individual key, places it in *data* field, and uses group key for encrypting entire message again.

Recipient along the path decrypts with group key in order to prove packet header before forwarding. If the proof succeeds, it forwards the original packet to its parent, otherwise it drops the packet. Thereby, even if adversary has group key, he cannot forge sensed data because he does not have individual key of sensor node. After base station received the packet, it verifies the packet with group key and corresponding individual key respectively.

Since the main cause of energy consumption in sensor networks is communication cost (compared with computation cost), we can decrease energy consumption by minimizing the number of forwarding messages. Sensor does not transmit sensed data immediately, but it waits for a while to collect sensed data from lower-level sensors. Let T be sensing period, *i.e.*, sensor nodes sense data every T period. Let DF denote *delay factor*, where $DF \geq 0$. In this case, sensor will wait for $T \times DF$ period before transmitting sensed data. If $DF = 0$, sensor nodes transmit sensed data immediately after sensing. If we set DF to high value, base station must wait for a long period of time and the freshness of data will decrease. This is trade-off between energy minimization and data freshness. However, if the amounts of data fill up vacant space in *data* field, sensor will immediately transmit regardless of delay factor.

7 Related Work

Deering and Cheriton invented the idea of IP multicast as a way to efficiently provide data to a group [5]. There are no restrictions on who may join a group or send data to a group. Our protocol restricts group members to only sensors deployed by user. Both Wong *et al.* [6] and Wallner *et al.* [7] use a logical hierarchy of key-encrypting keys to efficiently update group key on each join or leave. In contrast, re-keying is not necessary for our protocol because the communication in the past is not confidential for legitimate deployed sensor. We add newly deployed sensor's individual key into base station's ACL for authenticating purpose on each join, and delete such key from ACL on each leave. Concast are multipoint to point packet gathering approaches that try to reduce the network traffic by merging small packets travelling towards a common destination [8]. Implementing concast requires concast-capable routers to recongize concast datagrams and apply merge processing hop-by-hop. In our protocol, every sensor node can simply do concasting by only forwarding packet to parent node according to the tree shared with multicast. Aggregation and data compression can reduce communication cost in concasting [9], [10]. Directed diffusion is data-centric, *i.e.*, data generated by sensor nodes is named by attribute-value pairs [10]. A node requests data by sending interests for named data. Directed diffusion is query-style communication which is obviously different from our work.

8 Conclusion

We have proposed a secure group communication protocol for wireless sensor networks which differ significantly from existing networks both wired networks and ad hoc networks. Communication is done through hierarchical topology providing energy efficiency. A combination of symmetric key encryption and public-key encryption algorith provides security composed of confidentiality, authentication, and integrity. Improving robustness to loss and simulating proposed protocol are our future works.

References

[1] A. Mainwaring and et al., "Wireless sensor networks for habitat monitoring," in *International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Sept. 2002.

[2] Y. Yang and et al., "Reliable group rekeying: A performance analysis," in *SIGCOMM'01*, Aug. 2001.

[3] M. Bellare and B. Yee, "Forward-security in private-key cryptography," in *Topics in Cryptology - CT-RSA'03 (LNCS 2612)*, Apr. 2003.

[4] R. Canetti, S. Halevi, and J. Katz, "A forward-secure public-key encryption scheme," in *Advances in Cryptology - EUROCRYPT'03 (LNCS 2656)*, Apr. 2003.

[5] S. Deering and D. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Trans. Computer Systems*, May 1990.

[6] C. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," in *SIGCOMM'98*, Sept. 1998.

[7] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: Issues and architectures," RFC 2627, June 1999.

[8] K. Calvert and et al., "Concast: Design and implementation of a new network service," in *International Conference on Network Protocols (ICNP'99)*, Nov. 1999.

[9] D. Petrovic and et al., "Data funneling: Routing with aggregation and compression for wireless sensor networks," in *International Workshop on Sensor Network Protocols and Applications (SNPA'03)*, May 2003.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *MOBICOM'00*, Aug. 2000.