

SECURING GROUP COMMUNICATION IN WIRELESS SENSOR NETWORKS

Niwat Thepvilojanapong[†], Yoshito Tobe[‡], Kaoru Sezaki^{††}

[†]Department of Information and Communication Engineering, University of Tokyo

[‡]Department of Information Systems and Multimedia Design, Tokyo Denki University

^{††} Center for Spatial Information Science, University of Tokyo

[†]wat@mcl.iis.u-tokyo.ac.jp, [‡]yoshito@unl.im.dendai.ac.jp, ^{††}sezaki@iis.u-tokyo.ac.jp

ABSTRACT

In this paper, we present a secure group communication protocol for multi-hop, wireless sensor networks. Our protocol which is based on hierarchical tree works well on bidirectional communications between the base stations and sensor nodes. Tree construction is initiated by the base station which broadcasts solicit packet to discover child nodes. Sensor node receiving this packet decides an appropriate parent to which it will attach, it then broadcasts solicit packet to discover child nodes in the next level of the tree. This process is performed on every node. Consequently, hierarchical tree is rapidly created without flooding of any routing information packets. Based on a combination of symmetric key encryption and public-key encryption algorithm, both base stations and sensor nodes can securely discover route and confidentially communicate and authenticate communicating parties. We also propose a method to manage tree and keys when members join or leave from the group to achieve robustness of the protocol.

1. INTRODUCTION

Recent advances in MEMS-based sensor technology and low-power RF design have enabled the development of relatively inexpensive and low-power wireless sensors. A great number of such sensors can coordinate amongst themselves to achieve a larger sensing task both in urban environments and in inhospitable terrain. For instance, we may use such wireless sensor networks for environmental and habitat monitoring, tracking system, failure detection, intrusion detection, etc. Generally, the process of sensing tasks in sensor networks can be divided into three phases. The first phase is to gather data by sensing modules. Communicating pattern in this phase can be considered as *anycast* [7], i.e., each sensor node transmits sensed data to one of the base stations (or sink nodes). The next phase is processing gathered data at the base station which is not considered in this paper. In the last phase, the base stations *multicast* [4] commands, data, query, or image software to the sensor nodes. Many research literatures have focused on routing, querying,

rate control, and optimizations such as aggregation and compression. However, there is little consideration in security which is presented in this paper.

To motivate the challenges in designing a secure communication protocol, we show scenarios which usually happen in any sensing application. A large number of sensors (over one thousand nodes, for example) are deployed in remote terrain. The sensors coordinate to establish a communication network, monitor specified task, and report sensed data periodically or on-demand. When additional sensors are deployed or existing sensors fail, the sensors re-organize themselves to achieve efficient and robust sensing tasks. Moreover, adversaries may invade into monitoring terrain and deploy wireless nodes to do malicious actions by trying to access the network, capture the packets, impersonate legitimate sensors, etc. To prevent such attacks, we can employ conventional cryptographic algorithms (e.g., symmetric-key, public-key encryption algorithm) into sensor networks. Sensors, however, are easily stolen because they are tiny sensing devices deployed in the wide area without any defense system. Adversaries can analyze captured sensors in order to achieve cryptographic keys. This kind of vulnerability should also be considered when designing a secure protocol for sensor networks.

The remainder of the paper is organized as follows. Section 2 describes problem statement of our work. Section 3 presents architecture for securing group communication in sensor networks. Related work is discussed in Section 4. Finally, we conclude this paper in Section 5.

2. PROBLEM STATEMENT

2.1. Problem Setting

We consider network composed of a small number of base stations and a numerous number of wireless sensors randomly distributed in an interesting area. These sensors have limited processing power, storage, bandwidth, and energy, while the base stations have powerful resources in performing any tasks or communicating with the sensors. We assume that sensors are not mobile nodes, i.e., all nodes are fixed for the duration of their lifetime, however, sensor network we consider has dynamic characteristics such that new nodes

may be added at any time or battery of the node is depleted with time. In particular, sensor nodes have omni-directional antennas and use RF to communicate. All wireless network transmissions are inherently broadcast and a node may be able to configure its network interface into promiscuous receive mode.

2.2. Notation

Public and private key used in our protocol are keys according to any public-key encryption (asymmetric encryption) scheme. Public key in our scheme is not announced to public as signature scheme, in contrast, it is kept confidential. Symmetric key is a key according to any symmetric-key encryption scheme. The examples of implementing both cryptographic algorithms in sensor node can be found in [2, 3]. We use the following notations to describe our protocol, especially cryptographic operations in the paper.

K_{pub} is base station's public key.

K_{pri} is base station's private key.

K_{grp} is shared symmetric key for each group.

K_{sn} is individual sensor node's symmetric key.

$E(K, M)$ is the encryption of message M with the key K .

$X|Y$ denotes the concatenation of X and Y .

IV is initialization vector which is a block of random data. A timestamp can be used in our scheme.

3. ARCHITECTURE

Before sensor nodes are deployed in the working area, they are pre-loaded with K_{sn} , K_{grp} , and K_{pub} which are necessary for participating in the group. Base station's ID or any number defined by the user may be used as a group ID.

The format of packet used in our protocol is as follows: $\langle type, ID_{src}, ID_{dst}, ID_{grp}, E(K, IV | type | ID_{src} | ID_{dst} | ID_{grp} | seq | len | data) \rangle$. The *type* field is packet types. The ID_{src} , ID_{dst} , and ID_{grp} are source ID, destination ID, and group ID respectively. The *seq* field is a sequence number of packet. The *len* field is total length of packet and *data* field is used for carrying data. Each packet type is defined for individual purpose which will be discussed later. After receiving a packet, sensor node determines its action from *exposed tuple* (unencrypted tuple) consisting of *type*, ID_{src} , ID_{dst} , and ID_{grp} field. For instance, it immediately drops the packet of other communication groups which is not its interests by deciding from ID_{grp} field. The exposed tuple does not lead to vulnerability like substituting message or replay attack because it is also included in *secret tuple* encrypted with key K . Thereby, the adversaries cannot fake packets and legitimate nodes can detect such attacks from *seq* field in secret tuple.

3.1. Tree Construction

The base station initiates tree construction by broadcasting two *child request* (CREQ) packets separated

with interval T_i . Using two broadcast packets increases the reliability of protocol because broadcast packet is prone to lose and no retransmission mechanism supports. This CREQ packet is encrypted with group key (K_{grp}). *Nonmember node*, a node which does not attach to the tree yet, determines its parent by choosing a node whose received CREQ packet has arrived first as a parent or waiting for T_{creq} seconds in order to collect a number of candidates and choose a node whose defined metric is the best one (highest received power strength, highest remaining energy, etc.). We propose to use the latter method, i.e., nonmember node waits for a short period of time and then choose the best parent. It then sends a *child reply* (CREP) packet encrypted with group key to selected parent so as to inform that it will be a child node or a leaf node of current tree. *Member node* which is an internal or leaf node drops CREQ packet immediately.

Nonmember node proves its identity by encrypting arbitrary message (for example, ID_{bs}) with its individual key as an *authentication tag* and placing it in the *data* field of CREP packet. Parent node, in turn, encrypts child node ID and authentication tag with its individual key, and places it in the *data* field of *verify request* (VREQ) packet whose secret tuple is encrypted with group key. Thereby, the secret tuple of VREQ packet is $\langle E(K_{grp}, IV | E(K_{parent}, IV | ID_{child} | E(K_{child}, IV | ID_{bs}))) \rangle$. This packet is forwarded to the base station which decrypts it with group key, a key corresponding to source ID field (K_{parent}), and a key corresponding to nonmember node ID specified in decrypted message (K_{child}) respectively, in order to verify the identity of nonmember node. Each base station maintains a list of key-ID pairs called *access control list* (ACL) for authenticating purpose. If the decryption is successful, nonmember node is a legitimate sensor node. Base station sends the result of verification to parent node by using *verify reply* (VREP) packet including node IDs along the path until parent node in *data* field for forwarding purpose (source routing). The result is encrypted with parent node's individual key, thereby, the *data* field is $\langle ID_1 | ID_2 | \dots | ID_n | E(K_{parent}, result) \rangle$. Before forwarding the packet, each sensor node deletes its ID from routing list. If the authentication succeeds, parent node accepts nonmember node as a child node by using *child acceptance* (CACP) packet encrypted with the group key; otherwise it expels that node. However, parent node may notify the acceptance before VREP packet arrives in order to accelerate tree construction process. It can expel malicious node later after receiving a notice of failed authentication. In this case, the delay of VREP packet should be low enough that attacker cannot perform any malicious action.

Nonmember node waits CACP packet for a period of T_{caccp} seconds and if CACP packet does not arrive within this period, it issues second CREP packet. If T_{caccp} period has passed again and CACP packet still does not arrive, it sends third CREP packet as a last reply and chooses a new parent in the next round of T_{caccp} period. After receiving CACP packet from the

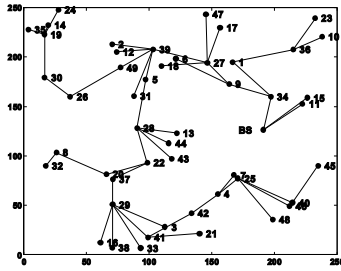


Fig. 1. Hierarchical tree created in the simulation.

parent, child node does the same process as its parent by broadcasting CREQ packet to discover its child nodes in the next level of the tree. These procedures are performed on every node throughout the network. The simulation result of tree construction according to this process by *ns-2* [1] simulation tool is shown in Figure 1.

3.2. Joining Process

Joining in sensor networks means user deploys new sensor nodes in the current network. Newly deployed node must find a parent for communicating purpose by broadcasting *parent request* (PREQ) packet encrypted with the group key. Any members of the tree that hear this packet reply by unicasting CREQ packet to joining node. Then, the processes follow tree construction phase, *i.e.*, joining node sends CREP packet to selected parent and waits for CACP packet as a confirmation of their relation. If joining node does not receive any CREQ packet, it can infer that no any node is within its radio coverage or all of its neighboring nodes do not attach to the tree yet. In this case, it waits for incoming CREQ packets after one of its neighbors has attached to the tree. We can also limit the number of child nodes on each parent in order to distribute loads and prolong network lifetime.

3.3. Leaving Process

Leaving means a group member has lost communication with all of its neighboring nodes due to numerous reasons. For instance, battery of the node is depleted with time or the node can be damaged due to harsh environment or by the enemy. If the left node is a leaf node of the tree, there is nothing to do with. However, we must reconstruct the tree if an internal node has left from communication group. Before explaining the processes in reconstructing the tree, we give the ideas on how to detect the left node. The first method is periodical announcement which is widely used by many routing protocols. Since our protocol is based on hierarchical tree, only parent node periodically broadcasts HELLO packet to assure child nodes of its existence. If child node does not receive HELLO packet for a certain number of times (three times, for example), it infers that its parent has left from the network. To decrease traffic load as well as energy consumption,

parent node suppresses this HELLO packet when there is multicast data for forwarding. The second method is adopted from a common characteristic of sensor networks that each node periodically transmits its sensed data to the base station. If parent node does not receive the packets from any child node for a while, it infers that such child nodes have completely left from the network. Alternatively, if the child node successively receives multicast packets from other nodes which are not its parent, it can also presume that it has become orphaned node. Note that multicast packets are always broadcasted (see Section 3.4). The last method can be done by relying on the underlying MAC-sublayer protocol. If the acknowledgement on MAC-sublayer does not arrive after a number of attempts (three times as usual) when trying to send data to the other party, it infers that the other party has left from the network. Each method has its own advantages and disadvantage. For instance, the first method consumes much energy but it is the most reliable method. We propose to use the MAC based approach which is less complex to detect left nodes and saves energy in our implementation.

When knowing the absence of parent, child node immediately switches to a new parent by choosing the most appropriate one from candidate list (which is created in tree construction phase) and sending CREP packet to selected parent. If there is no any parent in the list, it broadcasts PREQ packet as if it is a newly deployed node. The later procedures follow those described in Section 3.2. However, its child and grandchild nodes will not reply to this PREQ packet to prevent routing loop. Every node that attaches to the orphaned node does nothing because they do not know the absence of their grandparent. They can still forward packets to orphaned node as usual, and orphaned node keeps received packets in its buffer for sending later. Also, orphaned node can still forward heard multicast packets to child nodes as usual. In the worst case, orphaned node does not have any parent in the list and no any response to PREQ packet, it sends *parent query* (PQRY) packet to its child nodes asking whether they have any candidate for parent. Child nodes reply with *parent reply* (PREP) packet containing such information. Both packets are also encrypted with the group key. Then, orphaned node randomly chooses a child node that has at least one candidate parent as its new parent by sending CREP packet to inform new relation, and that child will switch to a new parent chosen from its list. If all of its child nodes do not have any candidate parent, orphaned node randomly chooses one child node as a new parent by sending CREP packet as usual and let this selected child node find a new parent by broadcasting PREQ packet. Note that the last scenario is very rare case that may occur in highly sparse networks.

3.4. Data Communication and Discussions

Anycast and multicast communication can be performed on same hierarchical tree. For anycast communication, each node encrypts sensed data with its

individual key, places it in *data* field, then encrypts secret tuple with group key. If the node receives anycast packets from its child nodes, it appends them in the *data* field. If it does not attach to the tree yet, it keeps such data in the buffer and send them later. Recipients along the path decrypt the packets with group key before forwarding for authentication purpose. If the decryption succeeds, they forward the original packet to their parent; otherwise they drop the packet immediately. Even if the adversary has group key, he cannot forge sensed data because he does not have individual key of any sensor node. After the base station receives the packet, it verifies the packet with group key and corresponding individual key respectively.

To achieve energy efficiency, the node propagates multicast packets by broadcasting. Data are propagated by every parent node upon receiving new multicast packets, both packets directly received from parent node and packets indirectly heard from other nodes. Note that the latter case is not overheard packets because we use broadcast communication in our approach. The redundant propagation of a multicast packet multiple times by a single node is prevented because each node records the source ID, group ID, and sequence number of received packet. It suddenly discards packets with same sequence number it has received.

Re-keying is not necessary in our protocol because the communication in the past is not confidential for legitimate deployed sensor node. User just adds newly deployed sensor's individual key into ACL on each join, and delete such key-ID pair from ACL on each leave. Unlike many existing protocols that use hop count as a routing metric, the node in our protocol locally decides the next hop based on the knowledge of the existing of neighborhood which is secured by using symmetric key and public-key algorithm that prevents the proposed architecture from all of known attacks such as sybil attack, wormhole attack, sinkhole attack, selective forwarding attack, HELLO broadcast attack, etc. [6].

4. RELATED WORK

Deering and Cheriton invented the idea of IP multicast as a way to efficiently provide data to a group [4]. 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 the user. Both Wong *et al.* [10] and Wallner *et al.* [9] 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 in our protocol, thereby, the number of keys the node carries decrease greatly.

Aggregation and data compression can reduce communication cost in anycasting as proposed in [8], [5]. Directed diffusion is data-centric scheme, *i.e.*, data generated by sensor nodes is named by attribute-value pairs [5]. A node requests data by sending interests for named data. Directed diffusion is query-style communication which is obviously different from our work.

5. CONCLUSION

We have proposed a secure group communication protocol for multi-hop, wireless sensor networks while maintaining a constant amount of local state and making only local decisions. Since the state required on each node is very low and independent of both network size and network density, our protocol is highly scalable. Moreover, anycast and multicast communication can be effectively done through the same hierarchical tree. Furthermore, a combination of symmetric key encryption and public-key encryption algorithm provides required security composed of confidentiality, authentication, and integrity for all of communications in the network.

6. REFERENCES

- [1] "Network simulator - ns-2," <http://www.isi.edu/nsnam/ns/>.
- [2] "TinyPK project," <http://www.is.bbn.com/projects/lws-nest/>.
- [3] "TinySec: Link layer encryption for tiny devices," <http://www.cs.berkeley.edu/nks/tinysec/>.
- [4] S. Deering and D. Cheriton, "Multicast routing in datagram internetworks and extended lans," *ACM Trans. Computer Systems*, vol. 49, no. 1-4, pp. 85-111, May 1990.
- [5] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *Proc. of International Conference on Mobile Computing and Networking (MOBICOM)*, pp. 56-67, Aug. 2000.
- [6] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pp. 113-127, May 2003.
- [7] C. Partridge, T. Mendez, and W. Milliken, "Host anycasting service," IETF, RFC 1546, Nov. 1993.
- [8] D. Petrovic, R. C. Shah, K. Ramchandran, and J. Rabaey, "Data funneling: Routing with aggregation and compression for wireless sensor networks," in *Proc. of the 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pp. 156-162, May 2003.
- [9] D. M. Wallner, E. J. Harder, and R. C. Agee, "Key management for multicast: Issues and architectures," IETF, RFC 2627, June 1999.
- [10] C. K. Wong, M. G. Gouda, and S. S. Lam, "Secure group communications using key graphs," in *Proc. of ACM SIGCOMM*, pp. 68-79, Sept. 1998.