An Energy-efficient Security Node-based Key Management Protocol for WSN

Bi Jiana
Department of Information Science and Technology
Bohai University
Jinzhou, China
bijiana@yahoo.com.cn

E Xu
Department of Information Science and Technology
Bohai University
Jinzhou, China
exu21@163.com

***Abstract*—A security node-based key management protocol is proposed for cluster-based sensor networks. Member nodes and cluster heads are responsible for data collection and transmission. Security nodes are responsible for key management. Security nodes restrain key management function of cluster heads, and reduce damage of captured cluster heads. Generation of security nodes and different kinds of keys is described. Performance analysis and simulation show that the proposed key management protocol consumes less energy, and its delay time of key generation is short. At the same time, the protocol can provide more collaborative authentication security for keys. It has strong resilience against node capture, and can support large scale network.**

Keywords- key management; wireless sensor networks; security; cluster

I. INTRODUCTION

Key management is the most difficult part of network security [1]. Attacks that aim for keys are easier than attacks that decipher key algorithms [2]. Excellent key management protocols can effectively increase network security and network resistance against attacks [3]. Symmetric and asymmetric encryption algorithms all involve key management. As wireless sensor networks(WSN) face particular security threats, traditional key management schemes are no longer fit for WSN. There should be new and special key management schemes for WSN. Study of this problem has been paid more attention by researchers [4-6]. In this paper, we propose a security node-based key management protocol(SNKM) for cluster-based WSN. In SNKM, we design several different kinds of keys. According to different kinds of data packets, nodes can select different keys for encryption and authentication. To improve security of cluster heads and reduce energy consumption of establishing clusters, we introduce security nodes and related schemes. Security nodes are responsible for key management in their clusters.

II. SECURITY NODE-BASED KEY MANAGEMENT

A. System Suppositions

We suppose that sink has enough energy. Sensor nodes are static, and they have the same communication and computation capability. Every data packet is designed as 36 bytes. Every node has enough space to store key information. Before deployed, nodes do not know their neighbors. New deployed nodes have different data storage structure from old ones. Our key management protocol works before rooting protocol. We suppose that before the time of T_{test} when pre-shared key of node is canceled, information of node can not be obtained by attackers. After $T_{\min}$ seconds, information of captured node will be obtained. Sink is absolutely safe, and it can not be captured.

B. Protocol Description

We orient our key management protocol to cluster-based WSN. According to different security levels of nodes, we adopt different security schemes and different kinds of keys. Cluster head plays an important role in WSN, and its security must be ensured. In SNKM, we let security nodes monitor cluster head. Once abnormal behaviors of a cluster head are found, election of a new head will be initiated immediately.

1) Security Node

Generation of security node is described in algorithm 1.

Algorithm 1. Generation of security nodes.

Input: n member nodes, pseudo random function f, random threshold T, proportion value P

Output: security nodes

```
begin
for (every member node i)
    compute f(i);
    if ( f(i)>T )
       set the node to be candidate security node;
    endif
endfor
if (current security nodes ratio> P )
          cancel part of candidate security nodes;
          else return all candidate security nodes;
endif
end
```

At first, according to a random function, node computes a random number. If the random number is more than T, the node can be a candidate security node. Then according to proportion value P, candidate security nodes are selected as security nodes. Proportion of security nodes is dependent on applications. Once member node detects abnormal behaviors of its neighbor nodes, it will report to security node. Given that general cluster-based rooting protocol often initiates a new cluster head election by all nodes in every round, this wastes much energy and communication resources. In our key management protocol, we let security nodes designate

Published by Atlantis Press, Paris, France.

cluster head by turn. A particular security node designates a new cluster head of each round. In different rounds, different security nodes need to maintain information tables to record remaining energy of every neighbor node and its ID. And then according to principle of energy priority, one security node periodically selects a new cluster head among its neighbor nodes in every round. Algorithm 1 needs to compute random value of every node, so its time complexity is $O(N)$. Our protocol only sets less security nodes, so communication complexity of selecting candidate security nodes as security nodes is less than $O(N)$.

2) Node Key

K_u is a key between node and sink. It is allotted by sink in advance. It is used by node to encrypt information that needs to be sent to sink.

3) Pair-wise Key

In our system, before deployed, nodes do not know their neighbors. When node u is added to networks, it immediately starts a timer to find neighbor nodes. At the same time, it broadcasts a hello packet containing its ID, and waits information from neighbors. A neighbor node v which receives the hello packet will return an ACK packet, and encrypt information with public key(K_g). Generation of pair-wise key is described in algorithm 2.

$$u \rightarrow *: u \quad (1)$$

$$v \rightarrow u: v, MAC(k_g, u \mid v) \quad (2)$$

When the node u receives ACK packet from neighbor node v, it computes pair-wise key(K_{uv}) between them. f is a pseudo random function. $v \mid n_c$ is joint of node v's ID and a random number. Then node u encrypts $v \mid n_c$ with public key(K_g), and returns to node v.

$$K_{uv} = f_{Ku}(v \mid n_c) \quad (3)$$

$$u \rightarrow v: (K_{uv})_{K_g} \quad (4)$$

Algorithm 2. Generation of pair-wise key.

Input: n nodes, public key K_g, pseudo random function f

Output: pair-wise keys of neighbor nodes
begin
for (every node u)
broadcast a hello packet;
set a TIMER and wait;
if (receive ACK packet from neighbor node v within TIMER)
obtain ID of node v;
encrypt f (ID of node v) with K_g and send it to node v;
endif
endfor
end

Algorithm 2 needs nodes to communicate with their neighbor nodes to obtain shared pair-wise keys. Because the scheme is easy, the time complexity of algorithm 2 is $O(N)$.

4) Cluster Key

In our key management protocol, cluster key is negotiated by security nodes. At first, every security node generates a random key(K_u^c), and sends it to other security nodes with a timestamp. Security nodes compare the timestamps, and take the random key with the minimum timestamp as new cluster key. Then the new cluster key is sent to cluster head. When cluster head receives the same cluster key information from more than one security nodes, it confirms the new cluster key, and then encrypts the cluster key with pair-wise keys to inform all member nodes. Generation of cluster key is described in algorithm 3.

Algorithm 3. Generation of cluster key.

Input: cluster head, n member nodes, m security nodes, pair-wise keys of security nodes, pair-wise keys between cluster head and security nodes, pair-wise keys between cluster head and member nodes

Output: cluster key
begin
for (every security node)
generate a random key and a timestamp;
encrypt them to other security nodes with pair-wise keys;
if (its timestamp is the least)
encrypt its random key and send to cluster head with pair-wise key;
else encrypt the random key with the least timestamp it receives and send to cluster head with pair-wise key;
endif
endfor
if (cluster head receives more than one same random keys from security nodes)
encrypt the random key with pair-wise key and send to every member node;
endif
end

In algorithm 3, negotiation of cluster key between security nodes uses less time and communication. Transmission of cluster key between cluster and member nodes is realized by one-on-one encryption. So he scheme ensures transmission security of cluster key. The time complexity of algorithm 3 is $O(N)$.

5) Public Key

Public key is used by sink to encrypt broadcast information, and it needs to be regularly updated. We borrow minds from μTESLA which is a classic broadcast authentication protocol. The protocol uses unidirectional pseudo random function to generate authentication key chain. In addition, in our key management protocol, general member nodes do not receive broadcast information. Broadcast information is only received by security nodes.

Published by Atlantis Press, Paris, France.

Encryption and authentication of public information is described in algorithm 4.

Algorithm 4. Encryption and authentication of public information.

Input: broadcast information of sink, pair-wise keys between cluster head and security nodes, cluster key

Output: reply of broadcast information

```
begin
sink broadcasts information, and authenticates it with
uTESLA;
if (security nodes deem that the information needs to be
    known by all member nodes)
     encrypt the information to cluster head with pair-
     wise key;
     cluster head broadcasts the information to all
     member nodes with cluster key;
     for (every member node)
          select a security node randomly to validate the
          information;
          if (the information is correct)
               return right reply;
               else report to security nodes;
          endif
     endfor
endif
end
```

In algorithm 4, broadcast information is only processed by security nodes, and not by all nodes. In a few cases, member nodes need to receive broadcast information. In these cases, member nodes only communicate with cluster head and security nodes. The time complexity of algorithm 4 is $O(1)$.

III. PERFORMANCE EVALUATION

A. *Security Connectivity Probability*

In our key management protocol, at key pre-distribution stage, nodes establish shared pair-wise keys with their neighbor nodes. So security connectivity probability of SNKM is 1.

B. *Key Safety*

In cluster-based sensor networks, cluster heads are easily captured. SNKM restricts some rights of cluster heads, and lets security nodes monitor them and manage keys. We use random number and node key to generate cluster key. Even if attackers monitor all communication packets, they can only obtain node key, and they can not obtain cluster key and pair-wise keys.

C. *Capture Attack Resistance of Node*

Sensor nodes are deployed in open areas, so they are possible to be captured by attackers. Given easy design of nodes, attackers can use reverse engineering techniques to obtain key information. Effective key management protocol should avoid leaking shared key information. SNKM is oriented to cluster-based sensor networks. Captured nodes may only leak their pair-wise keys with neighbor nodes, current cluster key and public key. Thus attackers can only obtain cluster broadcast information of the captured node, and obtain network broadcast information encrypted by public key. Unidirectional key chain ensures security of broadcast communication. So obtaining of public key information is not very significant. Furthermore, SNKM can restrain information leakage diffusion of captured nodes. In sensor networks protected by SNKM, when there is broadcast information, nodes will consult security nodes to verify the reliability of the information. Just because of security nodes, capture attack resistance ability of nodes rises effectively.

D. *Network Expansibility*

Due to the limitation of transmission radius, in cluster-based sensor networks, number of nodes in the same cluster should be a reasonable value. Too many nodes in the same cluster will let routing information increase rapidly. And this may eventually leads that storage space of the node depletes and energy consumption increases. So when sensor networks expand, communication should be completed by establishing more clusters. Our key management protocol is oriented to cluster-based senor networks. Communication in the same cluster is encrypted by cluster key. Because of this, SNKM has excellent network expansibility.

IV. SIMULATION ANALYSIS

Energy consumption is an important standard of security protocol. Due to introduction of security nodes, generation of cluster keys needs security nodes to negotiate, and this will consume energy. We compare energy consumption of SNKM with other two cluster-based key management protocols which are LEKM [7] and LKM [8]. Fig. 1 shows their differences. We randomly select 50 nodes, and compare their energy consumption. Simulation shows that energy consumption of SNKM is less than LEKM and LKM. This is mainly because that the cluster head election scheme of SNKM effectively reduces broadcast information.

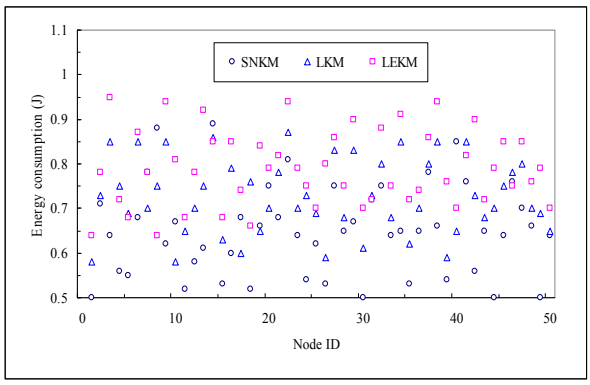

Figure 1. Energy consumption of node in key consultation

V. CONCLUSION

In the paper, we propose a security node-base key management protocol. Member nodes and cluster heads are responsible for data collection and transmission. Security nodes are responsible for key management. Security nodes restrain key management function of cluster heads, and reduce damage of captured cluster heads. We describe generation of security nodes, cluster heads, node key, pair-wise key, cluster key and public key. Performance analysis and simulation show that the proposed key management protocol consumes less energy, and delay time of key generation is short. The protocol can provide more collaborative authentication security for key. It has strong resilience against node capture, and can support large scale network.

ACKNOWLEDGMENT

The paper is supported by the National Natural Science Foundation of China(No. 61203002), the Project of Liaoning Province Office of Education(No. L2012396, No. L2012397, No. L2012400), and Postdoctoral Science Foundation of China(No. 2012M520158).

REFERENCES

[1] L. Eschenauer and G. Virgil, "A Key Management Scheme for Distributed Sensor Networks," Proc. the 9th ACM Conference on Computer and Communication Security, Nov. 2002, pp. 41-47.

[2] D. Sanchez and H. Baldus, "Deterministic Pair-wise Key Pre-distribution Scheme for Mobile Sensor Networks," Proc. the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks, Dec. 2005, pp. 277-288.

[3] W. Du and J. Deng, "A Pair-wise Key Pre-Distribution Scheme for Wireless Sensor Networks," Proc. the 10th ACM Conference on Computer and Communication Security, Sep. 2003, pp. 42-51.

[4] Wu Liang and Cao Xiaomei, "An Efficient Broadcast Key Management Policy in Wireless Sensor Networks," Journal of Electronics and Information Technology, vol. 32, Jan. 2010, pp. 1480-1484.

[5] D. Liu and P. Ning, "Establishing Pair-wise Keys in Distributed Sensor Networks," Proc. the 10th ACM Conference on Computer and Communications Security, Oct. 2003, pp. 128-134.

[6] Yang Geng and Wang Jiangtao, "A Key Establish Scheme for WSN Based on IBE and Diffie-Hellman Algorithms," Chinese Journal of Electronics, vol. 35, Jan. 2007, pp. 180-184.

[7] G. Jolly and M. Kuscu, "A Low-Energy Key Management Protocol for Wireless Sensor Networks," Proc. the 8th IEEE Symposium on Computer and Communications, Sep. 2003, pp. 335-340.

[8] M. Eltoweissy and M. Youis, "Lightweight Key Management for Wireless Sensor Networks," Proc. IEEE International Conference on Performance Computing and Communication, Mar. 2004, pp. 813-818.

Published by Atlantis Press, Paris, France.