

2012 International Workshop on Information and Electronics Engineering (IWIEE)

## A Replication Detection scheme for Sensor Networks

Xiaoming Wang, Yaohuan Liao<sup>a\*</sup>

*Department of Computer Science, Jinan University, Guangzhou 510632, China*

---

### Abstract

In this paper, we propose a replication detection scheme based on cluster-based structure. Our scheme employs the authenticated claim, neighbor proof and travel-time as the basic means of constructing replication detection procedure in order to avoid the need for public key cryptography and reduce storage overhead. Our scheme can effectively detect replica nodes and prevent replica nodes from adding sensor networks. Moreover, our scheme has no use for periodical replica detection and greatly decreases communication overhead. The analysis of security and performance shows that our scheme not only can resist node replication attack, but also has low computation overhead and smaller storage overhead.

© 2011 Published by Elsevier Ltd.

*Keywords:* Sensor networks; Replication detection; Authenticated claim; Neighbor proof; Travel-time

---

### 1. Introduction

So far, some replication detection schemes have proposed. In 2005, Parno et al.[1] proposed first the node replication attacks. They proposed Randomized Multicast and Line-Selected Multicast protocols. In Randomized Multicast protocol, each node broadcasts a location claim to its neighbors. Then each neighbor selects some random locations within the network and forwards the location claim with a probability to the nodes closest to chosen locations by using geographic routing. According to Birthday Paradox, at least one witness node is likely to receive conflicting location claims when replicated nodes exist in the network. In order to reduce the communication costs and increase the probability of detection, they proposed Line-Selected Multicast protocol. Besides storing location claims in randomly selected witness nodes, the intermediate nodes for forwarding location claims can also be witness nodes. In 2009,

---

\* Corresponding author. Tel.: 8602085221457; fax: 8602085221457.  
E-mail address: [wmsq@eyou.com](mailto:wmsq@eyou.com).

Ho et al. [2] proposed a distributed replication detection scheme to identify and revoke replica nodes. In their scheme, nodes are expected to be in their home zone and are marked by their neighbors as trusted if it is the case. The nodes of place their home zones have to prove their legitimacy by requesting their neighbors to forward their location claims to their home zones for conflicting location claim detection. Because the assumption of the deployment knowledge is not often reasonable and general, the heavy dependence on it makes the scheme undesirable. Furthermore, the scheme also uses the expensive public key cryptography for location claim generation and verification. In 2010, Fu et al. [3] proposed a novel approach against node replication attacks by key pre-distribution. Every node just could establish pairwise key with its location and time binding called *LTB*. In their scheme, every node just could establish pairwise key with its neighbors in the location that are assigned to nodes. Therefore, replica nodes can't establish shared key with its neighbors unless they are deployed in the location where duplicated nodes lie. In this way, sensor nodes in the networks need not periodically detect replica nodes. However, we found that *LTB* has following disadvantages (1) If an attacker captures a node *SN* and deploys the replica nodes of *SN* in the communication range of *SN*'s neighbors, *LTB* can't stop these replica nodes into networks. (2) When a legal node is first deployed in networks, *LTB* can't stop it establishing shared key with replica nodes. (3) *LTB* needs time synchronization mechanisms since it uses time  $T_{min}$  to decide whether establish shared key with its neighbors.

In this paper, we propose a scheme against node replication attacks based on cluster-based structure. Our scheme employs the authenticated claim, neighbor proof and travel-time as the basic means of constructing replication detection procedure in order to avoid the need for public key cryptography and reduce storage overhead. Our scheme can effectively detect replica nodes and prevent replica nodes from adding sensor networks. Moreover, our scheme has no use for periodical replica detection and greatly decreases communication overhead. The analysis of security and performance shows that our scheme not only can resist node replication attack, but also has low computation overhead and smaller storage overhead.

The rest of the paper is organized as follows. The Assumptions and model is described in section 2. In section 3, a replication detection scheme for sensor network is presented. In section 4, the security and properties of the proposed scheme are analyzed. Finally, the concluding remarks are given.

## 2. Assumptions and model

In this section, we first present the underlying assumptions for our schemes and then describe the attacker model.

### 2.1. Network assumptions

Our scheme uses the same network model as reference [4], that is, the sensor network consists of base station (BS), cluster heads (CH) and sensor nodes (*S*). Sensor nodes only can secretly communicate with its cluster head (CH) directly and can't communicate with each other directly, Sensor nodes only forward message from its neighbor nodes. Cluster heads can secretly communicate with base station directly and can't communicate with each other directly. We study the replication detection in a two-dimensional static sensor network where the locations of sensor nodes do not change after deployment. We assume sensor nodes are grouped together and nodes' location of deployment can be estimated in advance as in [5] described.

### 2.2. Attacker model

We assume that the attacker can identify and compromise a substantial fraction of the nodes in a small area. He will subsequently make replicas of one or more of these nodes and attempt to distribute them

throughout the network. We define an effective range of a node is the communication range of the node's neighbors. For example, a node  $S_1$ 's neighbors are the nodes  $S_2, S_3, S_4, S_5$ , then the four dashed circle zones are the effective rang of the node  $S_1$  and other zones is the non-effective rang of the node  $S_1$  as Fig.1 shows.

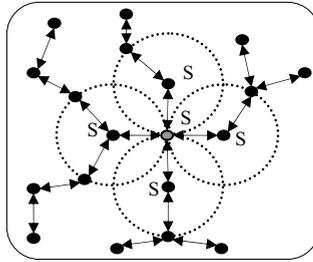


Fig.1 effective rang of the node  $S_1$

Our scheme would discuss two replication attacks as following:

- (1) The replica nodes of a node are deployed in non-effective range of the node.
- (2) The replica nodes of a node are deployed in effective rang of the node.

### 3. A replication detection scheme

Our scheme employs the authenticated claim, neighbor proof and travel-time as the basic means of constructing replica node detection procedure in order to avoid the need for public key cryptography and reduce storage overhead. Our scheme consists of following sections.

#### 3.1. Set up

BS chooses a finite field  $F_q$ , where  $q$  is a large odd prime of at least 160 bits, a secure one-way hash function  $H(\cdot)$  and a  $t$ -degree trivariate symmetric polynomial

$$f(x, y, z) = \sum_{i=0}^{i=t} \sum_{j=0}^{j=t} \sum_{k=0}^{k=t} a_{i,j,k} x^i y^j z^k \text{ mod } p$$

Where  $f(x, y, z)$  has a symmetric property such as  $f(x, y, z) = f(x, z, y)$  and  $a_{i,j,k}$ -s are the coefficients of  $f(x, y, z)$ ,  $t$  is a positive integer.

Before deployment, each sensor node  $S_i$  can be preloaded with relevant knowledge such as a deployment coordinates  $(x_i, y_i)$ , a location hash value  $LH_i = H(x_i || y_i)$ , a node secret key  $k_i$  and a cluster secret key  $K$ , a node identifier  $ID_i$ , a cluster identifier  $GID$ , a hash function  $H(\cdot)$ . Where  $k_i = f(GID, ID_i | LH_i, ID_{CH} | LH_{CH})$ . Besides, every node maintains a revoked nodes list. To establish share keys, prior to deployment, CH also is loaded a polynomial share  $f(GID, ID_{CH} | LH_{CH}, z)$ , a cluster secret key  $K$  and a hash function  $H(\cdot)$ .

#### 3.2. Replication detection

##### 3.2.1. Replication detection in non-effective range

In this section, we present the replication detection scheme when the replica nodes of one node are deployed in non-effective range of the node. Our scheme uses authenticated claim and neighbor proof to resist node replication attacks. Replica detection consists of three phases as described below.

##### (1) Authenticated claim

When a node needs to communicate with its cluster head CH, it is asked for an identifier and location authenticated claim with secret pair-wise share key between the node and cluster head CH. Suppose that a sensor node  $S_i$  (a member of the cluster  $GID$ ) needs to communicate with its cluster head CH, the node  $S_i$  first generates authenticated claim  $z_i = H(ID_i, LH_i, k_i)$  and broadcasts  $\{z_i, ID_i, E_K\{x_i, y_i\}\}$ . Where  $E_K\{\cdot\}$  denote encryption operation with a key  $K$  using symmetrical encryption algorithm such as AES.

##### (2) Neighbor proof

In our scheme, each node forwards the authenticated claim and only provides neighbor proof for its neighbor nodes as well as ignores all messages from the replica nodes. Suppose that a sensor node  $S_j$  receives an authenticated claim from node  $S_i$ . Node  $S_j$  first checks whether its revoked nodes list contains node  $S_i$  or not. If its revoked nodes list doesn't contain  $S_i$ , node  $S_j$  decrypts  $E_K\{x_i, y_i\}$  and gets  $(x_i, y_i)$ , then checks whether the distance between the deployment points of node  $S_i$  and node  $S_j$  is smaller than a pre-defined system threshold distance  $R$ . i.e.  $\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \leq R$ . If it is not, then node  $S_j$  thinks node  $S_i$  is not its neighbor and only forwards the authenticated claim. Otherwise, node  $S_j$  computes  $LH_i = H(x_i/y_i)$  and provides a neighbor proof  $v_j = H(ID_i, LH_i, k_j)$  for node  $S_i$ . Node  $S_j$  forwards the authenticated claim and neighbor proof  $(z_i, v_j, ID_i, LH_i, ID_j, E_K\{x_i, y_i\}, LH_i)$  to CH.

If its revoked nodes list contains node  $S_i$ , then node  $S_i$  is a replica node and node  $S_j$  will ignore all messages from the node  $S_i$ . Intuitively, if a node is confirmed to be a replica node and is revoked from cluster, then the replica node will be isolated and unable to send messages.

### (3) Detection and revocation

On receiving the authenticated claim from node  $S_i$ , CH first checks whether there is a neighbor proof. If there is, CH verifies whether the neighbor proof is valid or not, that is, CH computes secret pair-wise share key  $k_j = f(GID, ID_{CH}/LH_{CH}, ID_j/LH_j)$  between CH and node  $S_j$ , and verifies  $v_j = H(ID_i, LH_i, k_j)$ . If it holds, CH believes the neighbor proof is valid and computes the secret share key  $k_i = f(GID, ID_{CH}/LH_{CH}, ID_i/LH_i)$  between CH and node  $S_i$ , and verifies authenticated claim  $z_i = H(ID_i, LH_i, k_i)$ . If it holds, then CH believes the authenticated claim is valid and the node  $S_i$  is a legal node. CH accepts node  $S_i$ 's communication request.

If the authenticated claim is not valid, or there is no a neighbor proof, CH thinks that the node  $S_i$  is a replica node and refuses the node  $S_i$ 's communication request. Furthermore, CH broadcasts a node revocation message to node  $S_i$ 's neighbor nodes. The node  $S_i$ 's neighbor nodes receive the revocation message and add the node  $S_i$  into the revoked nodes list. Therefore the replica node will be isolated and unable to send messages.

#### 3.2.2. Replication detection in effective range

By above detection, we can only detect the replica nodes deployed in non-effective range, but not detect the replica nodes deployed in effective range. If the replica nodes are deployed in effective range, we could detect the replica nodes based on travel-time. For simplicity, we only consider a replica node  $C_{S_1}$  of node  $S_1$ , and  $C_{S_1}$  is deployed in effective range as Fig. 2 shows.

In order to detect replica nodes in effective range, each node maintains a timetable about the travel-time  $TT$  of its neighbor nodes, which is transmission time that a data packet is transmitted from its neighbor node to itself. For example, node  $S_1$ 's neighbors are the nodes  $S_2, S_3, S_4, S_5$ , then node  $S_1$  maintains a timetable as shows in Table 1. Where  $F$  denotes the state of a node.  $F=0$  denotes a legal node;  $F=1$  denotes a replica node.

When a replica node  $C_{S_1}$  sends a message to its neighbor node  $S_2$ , the replica node  $C_{S_1}$  must also send the time  $T_1$  of sending the message. Node  $S_2$  first computes the travel-time  $\bar{t}_1 = |T - T_1|$ , where  $T$  is the time of receiving message from replica node  $C_{S_1}$ . Then node  $S_2$  seeks for the timetable to obtain the travel-time  $TT_1$  according to node's  $ID$ , and checks whether the time difference between the travel-time  $TT_1$  and  $\bar{t}_1$  is smaller than a pre-defined system threshold time  $t$ . i.e.  $|TT_1 - \bar{t}_1| \leq t$ . If it not so, node  $S_2$  asks the replica node  $C_{S_1}$  to send a test packet to node  $S_2$ , node  $S_2$  checks again the travel-time. If still  $|TT_1 - \bar{t}_1| > t$ , then node  $S_2$  considers that  $S_1$  is compromised and  $C_{S_1}$  is a replica node. Therefore, the replica node  $C_{S_1}$  can be detected according travel-time unless the replica node  $C_{S_1}$  is deployed on the circle where  $S_1$  lies or the difference between  $S_1$ 's travel-time and it's replica node  $C_{S_1}$ 's travel-time is smaller than a threshold time  $t$ . It is noted that there is little benefit to the attacker of having a replica node in the smaller range as another compromised node [2].

### 4. Security and Overhead analysis

#### 4.1. Security analysis

By above replica nodes detection, our scheme can prevent replica nodes from communicating with CH. If the replica nodes of the node are deployed in non-effective range of the node, its neighbor nodes can detect the replica nodes by checking the authenticated claim and neighbor proof. For simplicity, suppose that the adversary has already compromised node  $S_1$  and placed a replica node  $C_{S_1}$  of  $S_1$  in the in non-effective range of  $S_1$  as Fig. 3 shows.

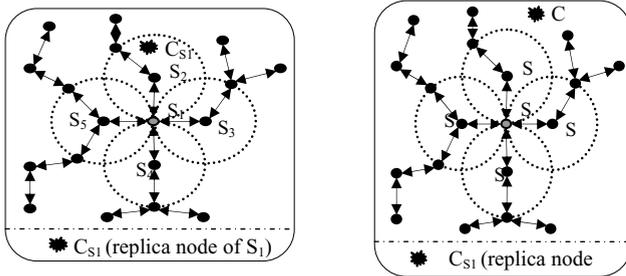


Fig. 2  $C_{S_1}$  in the effective range of  $S_1$  Fig 3  $C_{S_1}$  in the non-effective range of  $S_1$

From our scheme description, we see that the replica node  $C_{S_1}$  can only send correct identity  $ID_1$  and  $(x_1, y_1)$ ,  $C_{S_1}$  can pass the authenticated claim, that is, CH can confirm whether identity  $ID$ , location  $LH_1=(x_1/y_1)$ , and secret key  $k$  is correct or not by verifying  $z_1=H(ID_1, LH_1, k_1)$ . However, if the replica node  $C_{S_1}$  sends correct  $(x, y)$ , then its neighbor node  $S_j$  can find the replica node  $C_{S_1}$  is not its neighbor by

checking the distance between the  $C_{S_1}$  and node  $S_j$ , that is, node  $S_j$  checks  $\sqrt{(x_j - x_1)^2 + (y_j - y_1)^2} \leq R$ .

Because the replica node  $C_{S_1}$  of  $S_1$  is deployed in non-effective rang of  $S_1$ , the distance between the  $C_{S_1}$  and node  $S_j$  does not satisfy the above equation. Therefore, the neighbor node  $S_j$  can find that the replica node  $C_{S_1}$  is not its neighbor and does not provide neighbor proof for the replica node  $C_{S_1}$ . Without the neighbor proof, CH believes that the node  $C_{S_1}$  is a replica node.

If the replica node  $C_{S_1}$  of the node  $S_1$  is deployed in effective range of  $S_1$ , its neighbor nodes cannot detect the replica node  $C_{S_1}$  by checking the authenticated claim and neighbor proof. However, its neighbor nodes can detect the replica node  $C_{S_1}$  by the travel-time unless the replica node  $C_{S_1}$  is deployed on the circle where  $S_1$  lies or the difference between  $S_1$ 's travel-time and it's replica node  $C_{S_1}$ 's travel-time is smaller than a threshold time  $t$ . It is noted that there is little benefit to the attacker of having a replica node in the smaller range as another compromised node. Therefore, our scheme can detect replica nodes.

(2) From the above description, we can see that its neighbour nodes couldn't detect it when the replica node  $C_{S_1}$  of the node  $S_1$  is deployed on the circle where  $S_1$  lies. But this probability is small. We will make some three assumptions before we analyze this case. First, we assume that there are  $n$  replica nodes and they are deployed in the effective range of the node  $S_1$ . Second, we consider sensor node as a circle whose radius is  $r$  and its communication range is  $R$ . Finally, we could assume that the replica nodes are deployed in network by the same probability. According to the assumption, we could estimate that the number of locating on the circle is  $\pi R/r$  and the number of locating in the effective range is  $R^2/r^2$ . The probability that a replica node is deployed on circle is  $P_c = \pi r/R$ , and thus, the probability that  $n$  replica nodes are all

deployed on circle is  $\prod_{i=1}^{i=n} \frac{\pi \cdot r}{R}$ . So the probability of detecting replica nodes is  $P_c = 1 - \prod_{i=1}^{i=n} \frac{\pi \cdot r}{R}$ .

Table.1 timetable

ID	TT	F
$ID_2$	$TT_2$	0
$ID_3$	$TT_3$	0
$ID_4$	$TT_4$	0
$ID_5$	$TT_5$	0

## 4.2. Overhead analysis

From the above description, we can clearly see that our scheme does not introduce any significant communication, computation, or storage overhead. Each node only needs to check whether the distance between the deployment points two is smaller than a pre-defined system wide threshold distance  $R$  and can immediately determine whether to provides the neighbour proof. Each node only needs to perform a hash computation in order to establish communication with CH. Therefore the efficiency of our scheme is high. The storage overhead only includes a key of constant size and a neighbour timetable for each node, a share polynomial for cluster head. Therefore, the storage overhead of our scheme is low.

## 5. Conclusion

In this paper, we proposed a replication detection scheme for sensor networks that takes advantage of the authenticated claim and neighbour proof and travel-time to resist node replication attacks. Our scheme could stop the replica nodes to communicate with CH by binding the key material with  $ID$  and deployment location. In this way, it is very convenient to detect replica nodes for us because replica nodes are limited to a smaller zone. Moreover, our scheme is that communication overhead is every low because nodes needn't periodically detect the replica nodes.

## Acknowledgements

This work was supported in part by National Natural Science Foundation of China under Grant (61070164); Science and Technology Planning Project of Guangdong Province, China (2010B010600025; 2010A032000002); Natural Science Foundation of Guangdong Province, China (815106 3201000022).

## References

- [1] B. Parno, A. Perrig, V.D. Gligor. Distributed Detection of Node Replication Attacks in Sensor Networks. Proceedings of IEEE Symposium on Security and Privacy, 2005, 49-63.
- [2] J.W. Ho, D. Liu, M. Wright et al. Distributed detection of replica node attacks with group deployment knowledge in wireless sensor networks. Elsevier B.V : Ad Hoc Networks, 2009, 7(8) : 1476-1488.
- [3] F. Fu, W. Qi. Key predistribution with location and time binding: novel approach against node replication attacks in wireless sensor networks. Journal on Communications, 2010, 4(31) : 16-25.
- [4] M. Bechler, H.J. Hof, D. Kraft, F. Pählke, L.Wolf. A Cluster-Based Security Architecture for Ad Hoc Networks. Twenty-third Annual Joint Conference of the IEEE computer and Communications Societies. vol.4, 2004, pp. 2393 – 2403.
- [5] W. Heinzelman, A. Chandrakasan, H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, 2000, 2 : 1-10.
- [6] Y. Zhou, Y. Fang. A Two-Layer Key Establishment Scheme for Wireless Sensor Networks. IEEE Transactions on mobile computing, 2007: 6(9): 471–486.
- [7] Blundo C, De Santis A, Herzberg A et al. Perfectly-Secure Key Distribution for Dynamic Conferences[C]//Advances in Cryptology- Crypto'92. Berlin: Springer- Verlag, 1992: 471–486.