LETTER

A Resilient and Efficient Replication Attack Detection Scheme for Wireless Sensor Networks*

Chano KIM[†a], ***Student Member***, **Seungjae SHIN**[†b], **Chanil PARK**[†c], ***and*** **Hyunsoo YOON**[†d], ***Nonmembers***

SUMMARY In a large-scale sensor network, replicated hostile nodes may be used for harsh inner attacks. To detect replicas, this paper presents a distributed, deterministic, and efficient approach robust to node compromise attacks without incurring significant resource overheads.

key words: *wireless sensor network, replication attack, node compromise attack*

1. Introduction

Since sensors may be deployed in hostile environments for many large-scale wireless sensor network (WSN) applications, they are vulnerable to physical attacks. In a replication attack, which is considered to be a cost-effective physical attack, after capturing at least one sensor node, an adversary clones several nodes and surreptitiously returns replicas to the network. Using these replicated nodes, if undetected, an adversary can eavesdrop on secure links between legitimate nodes, misroute packets, and insert false sensing data since it rendered the whole authentication mechanism useless [1]. Therefore, when designing WSN security, it is important to detect cloned nodes. However, due to the limited resources of sensors and unknown network topology prior to deployment, it is difficult to distinguish between an original and a replica among all the operating nodes.

To cope with replication attacks, several protocols have been proposed on the basis of the witness-based strategy in the literature [2]–[4]; however, these protocols not only incur large resource consumption, but also fail to focus on the impacts of compromised nodes. Moreover, they require each sensor to perform public key cryptographic operations, which are not allowed in resource constrained sensor devices.

In this paper, we propose a distributed and deterministic replication attack detection protocol in static WSNs, with more resilience against a large number of compromised nodes and without incurring significant resource overheads. In addition, since network operators enable verification operations to direct nodes to specific locations, the proposed scheme can be adapted easily to various network conditions.

The rest of this paper is organized as follows: Sect. 2 introduces related work, Sect. 3 describes the protocol, Sect. 4 provides security and performance analysis, Sect. 5 evaluates our approach through simulation, and Sect. 6 concludes the paper.

2. Related Work

The first distributed approach of detecting replication attacks in WSNs was proposed by Parno et al. [2]. Since their approaches are based on the randomly selected witness nodes, the detection rate is relatively low even though there are large communication and storage overheads.

Bo Zhu et al. [4] proposed LM (Localized Multicast). In this scheme, each node sends a location claim message to a predetermined cell which is grouped in a geographically separated region. Upon arriving at a cell, this message is broadcasted and stored probabilistically at the witness nodes within the cell. Therefore, the detection rate and the communication overhead are tightly related to the number of nodes and the fraction of witness nodes, which store the location claim message in a cell. However, this scheme is not robust when all nodes within a predetermined cell are compromised.

Mauro et al. [3] proposed RED (Randomized, Efficient, and Distributed Protocol), which requires all sensors to be synchronized with a Base Station (BS) and have the same random value at a certain time through periodic broadcast beacons. To determine a witness point randomly, each node uses the pseudo random function, which takes in the *id* of a node and the current random value. Though this scheme greatly reduced the communication and storage overheads, it is difficult for all nodes to obtain the same value in large networks, and this remains the single point of failure of the BS.

3. Protocol Description

3.1 Assumption

We consider a randomly distributed WSN consisting of N static sensor nodes. The deployment region is a square area denoted by A, where the length of each side of the sensing area is q. Therefore,every node can be deployed at a spe-

Manuscript received August 19, 2008.

Manuscript revised January 20, 2009.

[†] The authors are with Korea Advanced Institute of Science and Technology (KAIST), Korea.

[*] This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy (MKE) in Korea as a result of UCN's subproject 09C1-T1-20S.

a) E-mail: cokim@nslab.kaist.ac.kr
b) E-mail: sjshin@nslab.kaist.ac.kr
c) E-mail: chanil@nslab.kaist.ac.kr
d) E-mail: hyoon@nslab.kaist.ac.kr

DOI: 10.1587/transinf.E92.D.1479

cific geographic coordinate (x_i, y_i), where $0 \leq x_i, y_i \leq q$, for $i = 1, \cdots, N$. We also assume each sensor node has a unique integer-valued *id*, and the BS is a trusted data collection center equipped with sufficient computation and storage. Moreover, every sensor node shares symmetric pairwise keys with the BS and its neighboring nodes to secure communication links [6]. For this, each sensor discovers its neighbors within its radio range which is a circle with radius R centered at its deployed location, and shares a pairwise key with every other sensor.

In our model, an attacker can perform the following actions: (1) intercept or tamper with messages (except its physical location information), (2) capture a limited number of nodes and control them, and (3) deploy replicated sensors, which can collude themselves with each other.

3.2 Protocol Description

A replica detection round is started when either a new node joins the existing network or may be requested by higher layers (i.e., application layer).

Initialization: Before deployment, a BS associates a particular location coordinate (hereafter referred to as the verification point,*vp*) with each node's *id* using geographic hash function F. A *vp* is the target location coordinate in the network where each sensor node will be verified, and it can be predetermined by a network operator to a certain extent with experience. For example, if area-fairness is considered to share the burden of replica detection overheads evenly among each node, every geographic coordinate in the sensing field has the potential to be chosen as a *vp*. Let the network area A be a rectangular of $q \times q$ square area, where q is an integer. If $h_x : \{0, 1\}^* \rightarrow Z_q$ and $h_y : \{0, 1\}^* \rightarrow Z_q$ are two uniformly distributed hash functions, the geographic hash function F can be defined as follows:

$F : S \rightarrow A$ such that $F(S_i) = (h_x(S_i), h_y(S_i))$ where S_i is an arbitrarily chosen node *id*. As a result, S_i can be mapped to $(h_x(S_i), h_y(S_i))$ as its vp_i.

Witness node discovery phase: In this step, the replicas with same *id* but different deployment locations are detected. For example, in Fig. 1, let S_u and $S_u^{'}$ be the original and replicated nodes, respectively. When a detection round for S_u is started, S_u generates the location claim message, which format is $[S_u, l_u, vp_u, H(S_u \parallel l_u)]$, where l_u is the geographic coordinate of S_u, H is a one-way hash function, and $\|$ denotes concatenation. After generating this message, it selects the next nodes among all the neighboring nodes with probability p_f and sends the location claim message to them. Upon receiving a given location claim, a neighbor of S_u checks the plausibility of l_u based on the communication range and its deployed location. After the neighbor verifies the validity of the received message using F and H, it stores $< S_u, l_u >$ entry in its buffer and checks the inconsistency of S_u with the same *id*, but different deployment location. Then, the location claim message is forwarded towards the closest node to the vp_u (referred as the candidate node: S_u^c), over multiple hops using the cheapest routing (i.e., GPSR [7]). The location claim message forwarding can be denoted by $S_u \rightarrow S_u^{1c} \rightarrow \cdots \rightarrow S_u^{ic} \rightarrow S_u^c$, in which i refers to the sequence of the intermediate forwarding nodes. Similarly, a location claim from S'_u is forwarded like $S_u^{'} \rightarrow S_u^{1c'} \rightarrow \cdots \rightarrow S_u^{jc'} \rightarrow S_u^c$.

Whenever every intermediate node along the routing path receives the location claim message of S_u, it stores $< S_u, l_u >$ entry in its temporary buffer until a collision takes place. Then, it checks whether it has received the location message from S_u within the same time frame. If l_u of S_u does not match, which means the nodes S_u are present with two non-coherent locations, that collision-detecting node becomes a witness node of S_u and triggers a replica revocation message for S_u. Therefore, even if a collision does not occur at the intermediate nodes, if replicas exist, they could be detected at S_u^c in the end.

Node revocation phase: In this step, a BS floods the revocation node lists after checking out the revocation request message received from the witness nodes. If a collision occurs at S_u^w, a replica revocation should be performed to prevent the cloned nodes from participating in network activities. To do this, S_u^w generates a revocation request message, $[S_u^w, l_w, S_u, l_u, l'_u]$ and delivers it over a multi-hop to the BS. Once a BS receives this revocation request message, it checks whether the revocation request message is correctly encrypted by S_w using a pairwise key shared with S_w. If the key is correct, a BS floods a list of replica nodes including S_u through the network. If the key fails, which means that an attacker sent the forged replica revocation message, the BS regards S_u^w has been compromised.

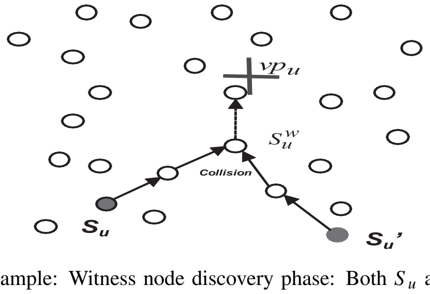

Fig. 1 Example: Witness node discovery phase: Both S_u and S'_u send a location message to the same S_u^w and then detect collision during the routing process.

4. Analysis

4.1 Security Analysis

In the variety of approaches for detecting replication attacks, the most important metric is the replica detection rate. Hence, we use the probability of detecting a replica P_r. We also use a network density d, which is the average number of neighbor nodes within the communication range of a node, and the probability p_f that each sending node decides how many neighbors to forward the location claim message toward the *vp*. In our scheme, the value of p_f can be provided to each node as a system parameter. Basically, p_f should be guaranteed to be greater than $\frac{1}{d}$. That is, at least one neighbor of each node should be selected as a location claim message forwarding node. Additionally, p_f can be regulated. For example, under normal circumstance without any attack effects on the network, p_f may be small, which reduces the number of communication messages, but increases the probability that the location claim message will not reach the witness node. By contrast, in a vulnerable environment, p_f is large, which provides an early detection of replicas and increases the energy consumption of each node to communicate and compute. If the location claim message is successfully forwarded to the candidate node, it is guaranteed that P_r is 100%. Since the probability that any candidate node may receive a location claim message is $1 - (1 - p_f)^d$, P_r is

$$P_r = (1 - (1 - p_f)^d)^2 \tag{1}$$

The above formula tells us that P_r increases with increasing p_f and d.

There are two security considerations for our scheme. First, since the adversary can predict candidate nodes as well, he may precisely survert them through denial of service (DoS) attacks or something similar. However, unless the majority of the nodes are compromised or the network is separated, the closest node to a certain location always exists in the network since a geographic coordinate is assigned as a destination of location claim message in our proposed scheme. As a result, even if the original closest node to the *vp* fails, the next closest node to the *vp* is selected as a candidate node. Unless two nodes with the same *id* are placed in the separated area as the compromised areas increase, our scheme guarantees that P_r is 100%.

Second, we consider that an suspicious intermediate node may modify the *vp* and mis-route the location claim messages to the wrong location. Note that all replicas, including an original node, have a distinct *vp*. Therefore, since every intermediate node checks the *vp* using both public and deterministic geographic hash function F and the plausibility of the previous forwarding node's location information, this threat can be prevented.

4.2 Performance Analysis

Since communication is the most energy-consuming operation in the WSN [5], the total number of hops of the location claim messages should be reduced as much as possible. Intuitively, the cost of communication overheads depends on the distance between a node and its corresponding *vp*. Table 1 shows comparisons of the asymptotic overheads of the witness node based detection protocols. In our scheme, both memory space and communication are required in most $O(\sqrt{N})$ operations on each device for a single detection round. Moreover, our scheme does not require the public key operation to generate or verify the signature. At first glance, the total resource consumption in the RED [3] is effective, except for the computation. However, RED [3] needs the help of a trusted third party to broadcast the random number periodically.

Table 1 Comparisons of the witness node based detection protocols. (s: the number of nodes within a cell, ω: the fraction of witness nodes within a cell)

Schemes	Number of messages	Number of storing nodes	Signature verification
RED [3]	$O(\sqrt{N})$	$O(1)$	$O(1)$
LM [4]	$O(\sqrt{N} + s)$	$O(\omega)$	$O(s)$
Ourscheme	$O(\sqrt{N})$	$O(\sqrt{N})$	·

5. Simulation Results

In this section, we demonstrate the security and performance of our proposed approach compared with RED [3], and LM [4]. First, to assess the robustness against node compromise attack, we measured P_r by increasing the number of compromised nodes, which are intelligently selected in each scheme: In RED [3], the nodes are located in the center of the network and are highly likely to be intermediate nodes on every communication path. In LM [4], the nodes are within a predetermined cell, and in our scheme, the nodes are within the vicinity of the *vp*. Also, the more replicas there are, the higher the successful detection rate is and, as a result, we insert only one replica node while leaving an original node as it is.

Simply put, we consider uniform network topology in which nodes are randomly distributed within a 500 × 500. For both RED [3] and our scheme, we select the center point (250, 250) as *vp* for each of the trials and perform it until a collision is detected. Next, an original node is placed (30, 30) and the replica is placed at various positions according to the simulation requirements. The simulation ends when at least one collision takes place, and each simulation result is obtained from the average of ten independent runs. For maintaining the consistency in simulation, as depicted by the authors, the number of cells and the fraction of witness nodes in a cell are ten and 10% respectively for LM [4].

Figure 2 presents the successful replica detection rate. Even though the compromised areas increase, RED [3] and our proposed scheme guarantee that P_r is 100% unless the network is separated into multiple disconnected subnetworks. That is, Though a large number of nodes (50%) are disabled, and they can achieve a high success rate for detecting replicas. Even if the network is partitioned, if two related nodes are placed in each network component, the network can detect the replica in our proposed scheme. By contrast, if all the nodes within a cell are compromised, the replica detection for the nodes whose destination is that cell fails completely in LM [4].

Next, to evaluate the communication overhead, we measured the average number of total hops of a location claim message per one detection round by increasing the relative distance between the original one and its replica. For this measurement, the relative distance between two replicas

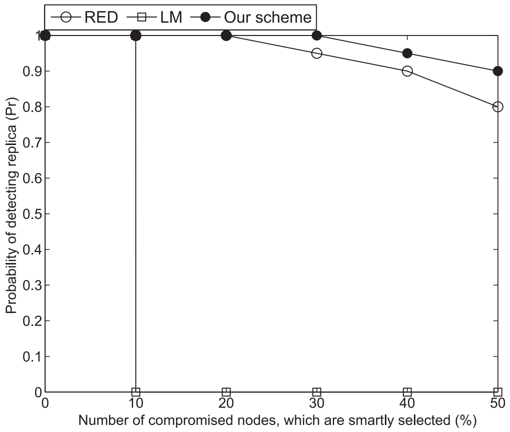

Fig. 2 Detection rate when the carefully selected nodes are compromised. (d = 20)

d_r is defined as

$$d_r = \frac{dist(S_u, S_u^{'})}{dist_{max}}, \tag{2}$$

where $dist_{max} = \sqrt{2 \cdot A}$. Note that early replica detection results in a significant improvement in performance with a larger network. However, in RED [3], since the witness nodes are selected randomly from the whole network, even if the replicated nodes are placed next to the original node, the location claim messages may be forwarded to the witness nodes through the same forwarding path. In a large scale sensor network, this can consume a large amount of energy and shorten the network lifetime.

Our approach can, however, detect replicas without introducing a significant amount of traversing of location claim messages, as the distance between two replicas is reduced. As shown in Fig. 3, the shorter the distance between two related nodes, the higher the probability a pair of nodes will intersect on the routing path. Since the sensors are deployed and operated in spatially distributed groups, the attacker inserts the replicas around the original node. Therefore, our scheme provides a high level of detection rate without increasing the energy consumption.

Next, to evaluate the storage cost, we defined it as the number of nodes storing the location claims. Figure 4 shows the number of nodes for storing location claims while increasing the number of nodes. In the RED [3], only the sender and the witness node store the location claim, regardless of the network size. However, in LM [4], the amount of required storage increases linearly as the network size increases. In the proposed scheme, the cost of storage is almost constant. This is because the next forwarding nodes are greedily selected in the routing algorithm. As shown in Fig. 4, the proposed scheme incurs only an almost constant amount of storage overhead.

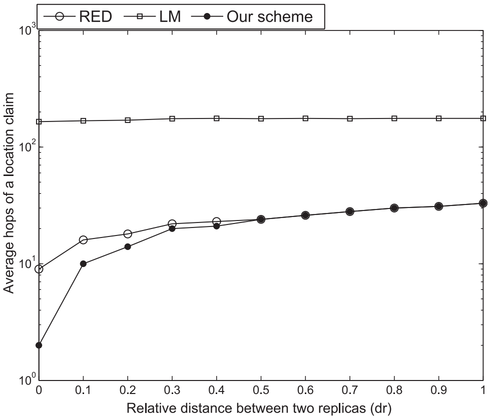

Fig. 3 Average number of hops of a location claim. (d = 20)

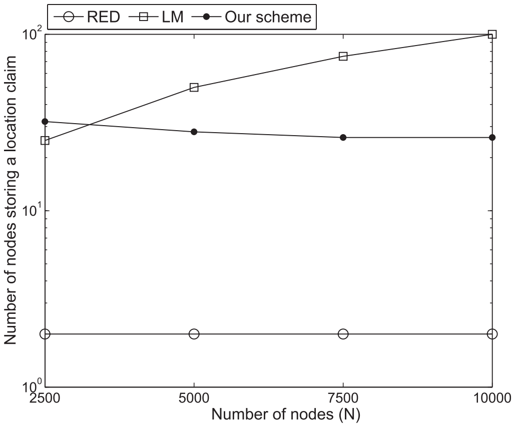

Fig. 4 Average number of nodes storing a location claim. (P_r = 1)

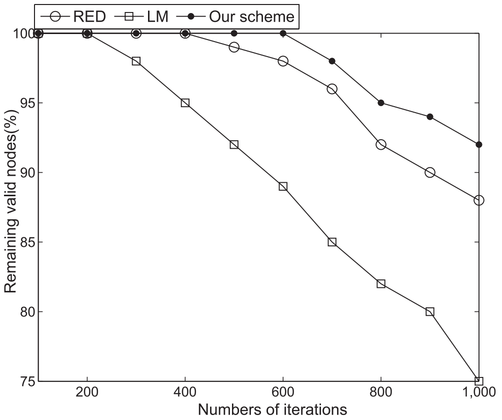

Fig. 5 Remaining number of valid nodes while increasing the number of detection iterations. (d = 20)

From the perspective of energy consumption for calculating cryptographic primitives, a public key signature requires extensive computations, causing the protocol to be become extremely slow. To measure the energy consumption, we used an energy model in [5] with the same message length of 32 bytes, a total available node power of 324,000 mJ, 45.0 mJ for signing, 15.10 mJ for sending a message [5], 7.17 mJ for receiving a message, and 3.4 mJ for symmetric (en/de) cryption operation.

Especially in RED [3], whenever every node receives a random number from the BS, it should check its correctness with BS's public key to prevent spoofing attacks. As shown in Fig. 5, after sufficient iterations (1,000), about 8%, 12%, and 25% of nodes have been depleted in each scheme, respectively. Therefore, our approach provides better sustainability than other related schemes.

6. Conclusion

In this paper, we addressed the node replication attacks in WSNs. We proposed a distributed, efficient, and resilient scheme against a large number of compromised nodes while minimizing resource consumption. Moreover, since it is important to predetermine the locations for some stringent military applications rather than choosing them randomly, our protocol provides more scalability to network operators than previous works.

References

[1] A. Becher, Z. Benenson, and M. Dornseif, "Tampering with motes: Real-world physical attacks on wireless sensor networks," Security in Pervasive Computing (SPC 2006), pp.104–118, 2006.

[2] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," Proc. 2005 IEEE Symposium on Security and Privacy (S&P'05), pp.49–63, 2005.

[3] M. Conti, R. Di Pietro, and L.V. Mancini, "A randomized, efficient, and distributed protocol for the detection of node replication attacks in wireless sensor networks," Proc. 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc'07), pp.80–89, 2007.

[4] B. Zhu, V.G.K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," Computer Security Applications Conference (ACSAC 2007), pp.257–267, 2007.

[5] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, "Energy analysis of public key cryptography for wireless sensor networks," Proc. Third IEEE International Conference on Pervasive Computing and Communications (PERCOM'05), pp.324–328, 2005.

[6] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," Proc. 10th ACM Conference on Computer and Communication Security (CCS 2003), pp.62–72, 2003.

[7] B. Karp and H.T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," Proc. 6th ACM International Conference on Mobile Computing and Networking, (MobiCom 2000), pp.243–254, 2000.