Mobile Sensor Network Resilient Against Node Replication Attacks

Chia-Mu Yu[§†], Chun-Shien Lu[§*], and Sy-Yen Kuo[†]
[§]Institute of Information Science, Academia Sinica, Taipei, Taiwan 115, ROC
[†]Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan 106, ROC

***Abstract*—By launching the *node replication attack*, the adversary can place the replicas of captured sensor nodes back into the sensor networks in order to eavesdrop the transmitted messages or compromise the functionality of the network. Although defending against node replication attacks demands immediate attention, only a few solutions were proposed. Most of the existing distributed protocols adopt the *witness finding* strategy, which selects a set of sensor nodes somewhere as the witnesses, to detect the replicas. However, the energy consumption of the witness finding strategy is remarkably high and even gets worse in mobile networks. In addition, the location information is necessary for each node if the witness finding strategy is applied. In this paper, a novel protocol, called eXtremely Efficient Detection (XED), is proposed to resist against node replication attacks in mobile sensor networks. The advantages of XED include (1) only constant communication cost is required for replica detection; (2) the location information of sensor nodes is not required. Performance analyses and comparison with known methods are provided to demonstrate the effectiveness of our protocol.**

I. Node Replication Attack

Usually, the sensor networks are unattended and the sensor nodes are not equipped with the tamper-resistance hardware so that the adversary can capture one sensor node, fabricate many replicas having the same identity (ID) from the captured node, and then place these replicas back into the strategic positions in the network for further malicious activities. This is a so-called *node replication attack*. Since the credentials of replicas are all the clones from the captured nodes, the replicas can be considered as legitimate members of the network, which make detection difficult. From the security point of view, the node replication attack is considerably harmful to the networks, because having legitimate keys, the replicas controlled by the adversary can easily launch the insider attacks, without easily being detected.

Based on the assumption that a sensor node, when attempting to join the network, must broadcast a signed location claim to its neighbors, most of the existing distributed detection protocols [2], [5], [6] adopt the *witness finding* strategy, in which each node finds a set of sensor nodes somewhere as the *witnesses* for checking whether there are the same IDs used at different locations, to detect the replicas. A naive detection approach is *Broadcasting* [5], in which each node floods its ID. It is obvious that broadcasting incurs tremendous communication cost. In Deterministic Multicast (DM) [5], the witnesses are determined for each node by using a public-known hash function. In [5], both *Randomized Multicast* (RM) and *Line-Selected Multicast* (LSM) were proposed to determine the witnesses randomly. In the Single Deterministic Cells (SDC) and Parallel Multiple Probabilistic Cells (P-MPC) approaches [6], a set of witness nodes located in the vicinity are chosen for each node by using a public-known hash function. Based on the assumption that there is a very efficient way to broadcast a pseudorandom number to all of the sensor nodes periodically, RED [2] also adopts the witness finding strategy to detect the node replication attacks but with less communication cost. Nevertheless, the reduction of communication cost relies on the involvement of the special centralized broadcasting devices such as satellites and Unmanned Aerial Vehicles (UAVs), which are not always realistic. Some other protocols exploit the characteristics, such as the increased key usage for certain keys used by replicas [1] and the non-emptiness of two exclusive subsets of sensor nodes' IDs [3], to detect the existence of replicas. However, we find that the common weakness of the existing protocols in detecting node replication attacks is that a large amount of communication cost is still unavoidable.

*Contact Author: Dr. C. S. Lu (lcs@iis.sinica.edu.tw).

II. The eXtremely Efficient Detection (XED) Protocol

The witness finding strategy exploits the fact that one sensor node cannot appear at different locations. Unfortunately in mobile networks, the sensor nodes have the possibility of appearing at different locations at different time. The witness finding strategy can adapt to the mobile environments if time-stamp is associated with each location claim. In addition, setting a fixed time window t in advance and performing the witness finding for every t units of time can also keep witness finding strategy feasible in mobile networks. However, the former additionally requires accurate time synchronization while the latter has the drawback that routing the message to the witness nodes in the mobile networks incurs even higher communication cost. In view of these, the eXtremely Efficient Detection (XED) protocol is proposed to address the node replication attacks in mobile networks by adopting a proposed strategy, *remember and challenge*. Moreover, the sensor nodes do not need to be aware of their respective locations when XED is performed. It should be noted that the location information, however, is necessary for all detection protocols.

978-1-4244-1777-3/08/$25.00 ©2008 IEEE

A. Assumptions

Let the sensor network consist of $n+q$ sensor nodes, n of which, $s_1, \ldots, s_n$, are genuine, and q of which are counterfeit and have the same ID, s_A. In XED, the sensor nodes are assumed to have mobility. In the beginning, the sensor nodes are uniformly deployed. After deployment, sensor nodes can move according to some mobility models such as the random waypoint model. All of the replicas under the adversary's control are assumed to not simultaneously communicate and collaborate with each other. We will further discuss in Sec. III how to cope with the case when the replicas can communicate with each other.

B. Algorithm: remember and challenge strategy

The idea behind XED is motivated from the observation that for the networks without replicas, if a sensor node s_i meets the other sensor node s_j at earlier time and s_i sends a random number r to s_j at that time, then when s_i and s_j meet again, s_i can ascertain whether this is the node s_j met before by requesting the random number r. Based on this observation, a "remember and challenge strategy" is proposed and described as follows.

Once two sensor nodes, s_i and s_j, are within the communication ranges of each other, they first, respectively, generate random numbers $r_{s_i \to s_j}$ and $r_{s_j \to s_i}$ of b bits, where $r_{s_i \to s_j}, r_{s_j \to s_i} \in \{0, \ldots, 2^b - 1\}$ and b is an integer, and then they exchange their generated random numbers. They also use a table to record the node ID, the generated random number, and the received random number in their respective memory. Note that for the pair of two nodes met before, the above procedure is also performed such that the random number stored in the memory is replaced by the newly received random number.

Consider the case illustrated in Fig. 1 that the sensor node s_i meets another sensor node s_j. If s_i never meets s_j before, they exchange random numbers. Otherwise, the sensor node s_i requests the sensor node s_j for the random number $r_{s_i \to s_j}$ exchanged at easier time. For the sensor node s_i, if the sensor node s_j cannot replies or replies a number which does not match the number in s_i's memory, s_i announces the detection of a replica, as shown in Fig. 2. Note that when the replicas meet the genuine nodes, the replicas can always pretend that they meet for the first time. However, if the genuine nodes have a record showing that they ever met at earlier time, the replicas are also detected.

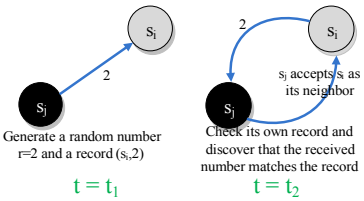

Fig. 1. The operations between two genuine nodes in XED at time t_1 and t_2. (Gray and black nodes are genuine.)

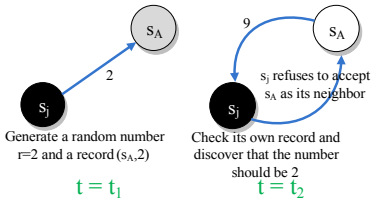

Fig. 2. Replica detection in XED at time t_1 and t_2. (Gray and white nodes are different replicas and black node is the genuine node.)

C. Security Analysis

We can know from Sec. II-B that the genuine sensor node s_i is aware of the existence of the replicas when the node s_j it met before cannot reply $r_{u \to v}$ correctly. Thus, two factors relevant to the security of our protocol need to be analyzed. First, since only if s_i meets two different replicas, then the replicas have possibility to be detected, we need to calculate the expected number of moves that s_i requires to meet two different replicas. This value determines how long will one of the replicas be detected by the node s_i. Second, since the replicas can try to fool the genuine node by guessing and then replying a random number, we need to calculate the probability that a genuine node is fooled by the replicas.

We make the following assumption to simplify the security analysis for XED. Though sensor nodes have mobility, the model we consider to simplify the mobility of sensor nodes is to assume that sensor nodes are deployed uniformly at the end of each time window. In addition, the average number of neighbors, d, for each node is the same. Based on this assumption, we can calculate the expected number of moves, X, that s_i requires to meet two different replicas as follows. The probability that the node s_i meets one of two replicas can be calculated as $\frac{q \cdot \binom{n+q-1}{d-1}}{\binom{n+q}{d}}$, while the probability that the node s_i meets the other can be computed as $\frac{(q-1) \cdot \binom{n+q-1}{d-1}}{\binom{n+q}{d}}$. Thus, the expected number of moves, $E[X]$, that s_i requires to meet two different replicas is calculated as:

$$E[X] = \frac{\binom{n+q}{d}}{q \cdot \binom{n+q-1}{d-1}} + \frac{\binom{n+q}{d}}{(q-1) \cdot \binom{n+q-1}{d-1}}. \quad (1)$$

The computational and simulation results are shown in Fig. 3 to validate our formulation. Note that each simulation result is the average of 100 runs. Since the replica can try to fool the genuine node by guessing a number, each node is, thus, possible to be deceived with certain probability for every $E[X]$ moves of sensor nodes. Since a replica can successfully deceive a genuine node with the probability $\frac{1}{2^b}$, this implies that each node will be deceived with the probability $\frac{1}{2^b}$ for every $E[X]$ moves. Since there are n genuine sensor nodes in the network and every genuine sensor node meets two different replicas for every $E[X]$ moves, it is obvious that the replicas successfully deceive the detection of the network with probability $(\frac{1}{2^b})^n$ for every $E[X]$ moves.

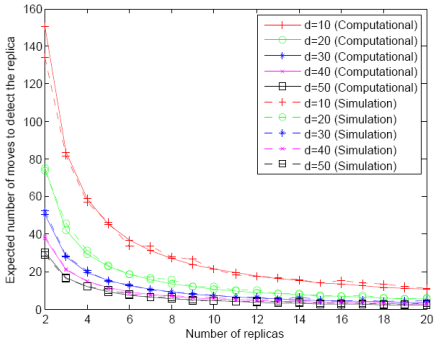

Fig. 3. The relationship between the expected moves in detecting replicas and the number of replicas ($n = 1000$).

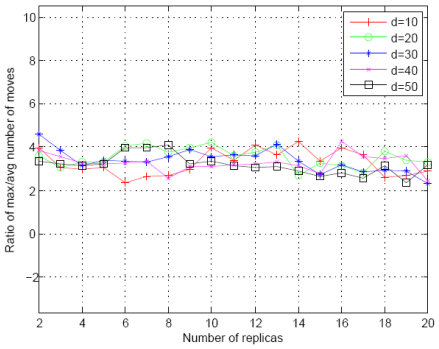

Fig. 4. The ratio of maximum/average number of sensor node moves ($n = 1000$).

D. Efficiency Analysis

Since the expected number of moves for a genuine sensor node needed to discover the existence of replicas in XED is $E[X]$, the memory overhead for each node should be at least $d \cdot E[X]$, which enables the random numbers sent to the replica met before to be stored. However, exact $d \cdot E[X]$ memory overhead is not sufficient for genuine nodes to detect node replication. This is because although a genuine node may meet different replicas with moves fewer than $d \cdot E[X]$, it is often possible for a genuine node to encounter different replicas with moves more than $d \cdot E[X]$. Our simulation result, shown in Fig. 4, reveals that $4 \cdot d \cdot E[X]$ moves is the upper bound of moves needed for a genuine node to encounter different replicas in most cases. In other words, if a genuine node has the record of random numbers sent to the other nodes within $4 \cdot d \cdot E[X]$ moves, then it is sufficient for this genuine node to check the consistency of random numbers if the same nodes are encountered again, resulting in $O(4 \cdot d \cdot E[X])$ memory overhead.

In the literature, the communication cost of a detection protocol is evaluated by considering the number of message exchanges in a single round of detection. In terms of the communication cost incurred by a single detection, our XED protocol only requires constant communication cost for exchanging the random number, since each node in XED is capable of detecting replicas per move. This unique feature contrasts with other protocols that need to mobilize the whole network for replica detection. A comparison between the XED protocol and the existing detection protocols in term of communication cost is shown in Table I.

TABLE I
COMMUNICATION COST COMPARISON FOR REPLICA DETECTION

Schemes	Comm. Cost
RKP [1]	$O(n \log n)$
RED [2]	$O(n\sqrt{n})$
SET [3]	$O(n)$
CP [4]	$O(n\sqrt{n})$
Broadcast [5]	$O(n^2)$
DM [5]	$O(n\sqrt{n})$
RM [5]	$O(n^2)$
LSM [5]	$O(n\sqrt{n})$
SDC [6]	$O(n\sqrt{n})$
P-MPC [6]	$O(n\sqrt{n})$
XED (this paper)	$O(1)$

III. DISCUSSIONS

In this paper, we propose an eXtremely Efficient Detection (XED) protocol based on the remember and challenge strategy for detecting node replication attacks in mobile networks. A unique feature of XED is that each node is capable of detecting replicas per move, which contrasts sharply with other protocols that need to mobilize the whole network for replica detection. Our protocol outperforms the existing detection protocols in two aspects: (1) only constant communication cost is required per detection; (2) the location information of sensor nodes is unnecessary.

As mentioned in Sec. II-A, the ability of replicas is somewhat weakened in this paper. Nevertheless, if this restriction is released to allow communication between replicas, XED still works except that the time for replica detection is postponed. In addition, it may be possible that one genuine node considers another genuine node as counterfeit due to the limited memory for storing the received random number. This problem can be properly amended by taking advantage of the cooperation between the one-hop neighbors.

REFERENCES

[1] R. Brooks, P. Y. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the Detection of Clones in Sensor Networks Using Random Key Predistribution," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 37, no. 6, pp. 1246-1258, Nov. 2007.

[2] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei, "A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks." *ACM MobiHoc*, 2007.

[3] H. Choi, S. Zhu, T. F. La Porta, "SET: Detecting node clones in Sensor Networks," *IEEE/ICST Securecomm*, 2007.

[4] L. Eschenauer and V. Gligor, "A Key-management Scheme for Distributed sensor networks," *ACM CCS*, 2002.

[5] B. Parno, A. Perrig, and V. Gligor, "Distributed Detection of Node Replication Attacks in Sensor Networks,", *IEEE S&P*, 2005.

[6] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks," *ACSAC*, 2007.