International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 8, Issue 10 (October 2013), PP. 26-31

Efficient and Effective Detection of Node Replication Attacks in Mobile Sensor Networks

S.Dhanalakshmi[1], S.Kaliraj[2], Dr.J.Vellingiri[3]
[1]*M.E-Student,Department of CSE, Kongunadu College of Engineering and Technology,Tamilnadu,India.*
[2]*Asst.Professor,Department of CSE, , Kongunadu College of Engineering and Technology,Tamilnadu,India.*
[3]*Associate Professor, Department of CSE, Kongunadu College of Engineering and Technology, Tamilnadu,India.*

Abstract:- Wireless sensor networks are often deployed in hostile environments, where an adversary can physically capture some of the nodes. Once a node is captured, the attacker can re-program it and replicate the node in a large number of replicas, thus easily taking over the network. The detection of node replication attacks in a wire- less sensor network is therefore a fundamental problem. Compared to the extensive exploration on the defense against node replication attacks in static networks, only a few solutions in mobile networks have been presented. Moreover, while most of the existing schemes in static networks rely on the witness-finding strategy, which cannot be applied to mobile networks, the velocity-exceeding strategy used in existing schemes in mobile networks incurs efficiency and security problems. In this paper Localized algorithms are proposed to resist node replication attacks in mobile sensor networks. The Merits of proposed algorithms are, it can effectively detect the node replication in localized manner. These algorithms are, also avoid network-wide synchronization and network-wide revocation.

Keywords:- Replication attack, security, wireless sensor networks, localized detection.

I. INTRODUCTION

Node Replication attack

Sensor networks, which are composed of a number of sensor nodes with limited resources, have been demonstrated to be useful in applications, such as environment monitoring and object tracking. As sensor networks could be deployed in a hostile region to perform critical missions, the sensor networks are unattended and the sensor nodes normally are not equipped with tamper-resistant hardware. This allows a situation where the adversary can compromise one sensor node, fabricate many replicas having the same identity (ID) from the captured node, and place these replicas back into strategic positions in the network for further malicious activities. This is a so-called *node replication attack*. Since the credentials of replicas are all clones of the captured nodes, the replicas can be considered as legitimate members of the network, making detection difficult. From the security point of view, the node replication attack is extremely harmful to networks because replicas, having keys, can easily launch insider attacks, without easily being detected.

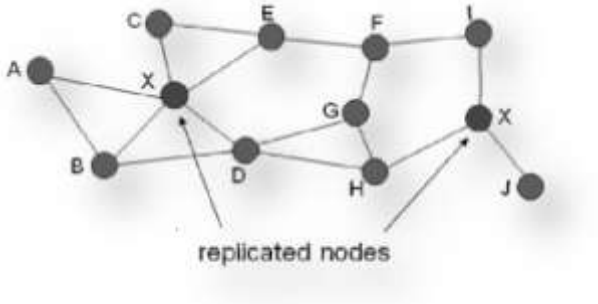

Fig1. Replication Attack

Recently, due to advances in robotics, mobile sensor net-works have become feasible and applicable. Nevertheless, although the problem of node replication detection in static net- works has been extensively studied, only a few schemes have been proposed for mobile sensor networks. Even worse, as indicated in , the techniques used in detecting replicas in static environments are not useful in identifying replicas in mobile environments. With the consideration of nodes' mobility and the distributed nature of

sensor networks, it is desirable, but very challenging, to have efficient and effective distributed algorithms for detecting replicas in mobile sensor networks.

II. RELATED WORK

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 proto- cols [4], adopt the *witness-finding* strategy to detect the replicas. In particular, the general procedure of applying wit- ness-finding to detect the replicas can be stated as follows. After collecting the signed location claim(v ,L(v),sig(L(v))) for each neighbor v of the node u, where L(v)and sig(.) denote the location of v and the digital signature function , respectively , u sends the signed location claims to a properly selected subset of nodes, which are witnesses. When there are replicas in the network, the witnesses, according to the received location claims, have possibility to find a node ID with two distant locations, which implies that the node ID is being used by replicas. Afterward, the detected replicas can be excluded using, for example, network-wide revocation. The detection algorithms proposed in [4], all belong to this category. For example, RM and LSM, were pro- posed in to determine the witnesses randomly. The difference between RM and LSM is that the witness nodes that find the conflicting location in the former are primarily affected by the number of witness nodes and the ones in the latter are primarily affected by the forwarding traces of location claims. SDC and P-MPC can be thought of as the cell versions of RM and LSM.

The preliminary version of this paper presents the first distributed detection algorithm for mobile networks based on a simple challenge-and-response strategy. Nevertheless, its detection effectiveness is vulnerable to the collusive replicas. Thus, propose exploitation of the mobility pattern to detect the collusive replicas. Unfortunately, their storage requirement is linearly dependent on the network size and is not scalable.Ho *et al.* [10] propose a centralized detection algorithm for mobile sensor networks using Sequential Probability Ratio Test (SPRT). Intuitively, by having each node send the location of each encountered node, the base station can check if there is a node appearing at two distant locations with a velocity exceeding the predefined limit. If such a node exists, it is very likely to be a replica. Nevertheless, practically there could be some errors in the node speed measurement, leading to either false positives or false negatives. To avoid the above false judgement, the method in [10] checks whether the estimated speed of a specific node can conclude that is a replica with the aid of SPRT. Essentially, SPRT is a specific sequential hypothesis test with null and alternative hypotheses. The purpose of SPRT is to determine which hypothesis should be accepted with the consideration of a sequence of observations. In the case of replica detection, null and alternative hypotheses correspond to "the node is not a replica" and "the node is a replica," respectively. The BS using SPRT continuously receives a stream of the estimated speeds of a specific node. Based on the decision principle of SPRT, the BS can make an accurate decision on whether the node under consideration is a replica even though some of the measured speeds are erroneous. The effectiveness of the method in [10], however, relies on the involvement of the base station, easily incurring the problems of single-point failure and fast en- ergy depletion of the sensor nodes around the base station.

Challenge In Detecting Replicas In Mobile Environments

The witness-finding strategy exploits the fact that one sensor node cannot appear at different locations, but, unfortunately, the sensor nodes in mobile sensor networks have the possibility of appearing at different locations at different times, so the above schemes cannot be directly applied to mobile sensor networks. Slight modification of these schemes can be helpful for applicability to mobile sensor networks. For instance, the witness-finding strategy can adapt to mobile environments if a timestamp is associated with each location claim. In addition, setting a fixed time window in advance and performing the wit- ness-finding strategy for every units of time can also keep wit- ness-finding feasible in mobile sensor networks. Nevertheless, accurate time synchronization among all the nodes in the net- work is necessary. Moreover, when witness-finding is applied to mobile sensor networks, routing the message to the witnesses incurs even higher communication cost.

After identifying the replicas, a message used to revoke the replicas, possibly issued by the base station or the witness that detects the replicas, is usually flooded throughout the network. Nevertheless, network-wide broadcast is highly energy-consuming and, therefore, should be avoided in the protocol design.
Time synchronization is needed by almost all detection algorithms [4], [10], Nevertheless, it is still a challenging task to synchronize the time of nodes in the network, even though loose time synchronization is sufficient for the detection purpose. Hence, as we know that time synchronization algorithms currently need to be performed periodically to synchronize the time of each node in the network, thereby incurring tremendous overhead, it would be desirable to remove this requirement.

Witness-finding could be categorized as a strategy of cooperative detection; sensor nodes collaborate in certain ways to determine which ones are the replicas. In this regard, the effective- ness of witness-finding could

be reduced when a large number of sensor nodes have been compromised , because the compromised nodes can block the message issued by the nodes near the replicas. Hence, the witness nodes cannot discover the existence of replicas. To cope with this issue, localized algorithms could enhance the resilience against node compromise.

In spite of the effectiveness in detecting replicas, all of the schemes adopting witness-finding have the common drawback that the detection period cannot be determined. In other words, the replica detection algorithm can be triggered to identify the replicas only after the network anomaly has been noticed by the network planner. Therefore, a detection algorithm that can always automatically detect the replica is desirable.

Since the existing algorithms are built upon several other requirements, we have found that the common weakness of the existing protocols in detecting node replication attacks is that a large amount of communication cost is still unavoidable.

III. CONTRIBUTION

To detect the node replicas in mobile sensor networks, two localized algorithms, XED and EDD, are proposed. The techniques developed in our solutions, challenge-and-response and encounter-number, are fundamentally different from the others. Our algorithms possess the following advantages.

- Localized Detection: XED and EDD can resist node replication attacks in a localized fashion. Note that, compared to the distributed algorithm, which only requires that nodes perform the task without the intervention of the base station, the localized algorithm is a particular type of distributed algorithm. Each node in the localized algorithm can communicate with only its one-hop neighbors. This characteristic is helpful in reducing the communication overhead significantly and enhancing the resilience against node compromise.
- Efficiency and Effectiveness: The XED and EDD algorithms can identify replicas with high detection accuracy. Notably, the storage, communication, and computation overhead of EDD are all only O(1).
- Network-Wide Revocation Avoidance: The revocation of the replicas can be performed by each node without flooding the entire network with the revocation messages.
- Time Synchronization Avoidance: The time of nodes in the network does not need to be synchronized.

IV. SYSTEM MODEL

Network Model

Assume that the sensor network consists of sensor nodes with IDs,$\{1,\ldots.n\}$. The communication is assumed to be symmetric. In addition, each node is assumed to periodically broad cast a beacon containing its ID to its neighbors. This is usually required in various applications, for example, object tracking. The time is divided into time intervals, each of which has the same length T . Nonetheless, the time among sensor nodes does not need to be synchronized. The sensor nodes have mobility and move according to the Random Way Point (RWP) model which is commonly used in modeling the mobility of *ad hoc* and sensor networks . Each node is assumed to be able to be aware of its geographic position. In this model, each node randomly chooses a destination point (waypoint) in the sensing field, and moves toward it with velocity v , randomly selected from a predefined interval [v min ,v max] . After reaching the destination point, the node remains static for a random time and then starts moving again according to the same rule. To simplify the analysis, we assume each node has d neighbors on average per move. Finally, we follow the conventional assumption in prior works that the network utilizes an identity-based public key system so signature generation and verification are feasible. In general, the models used in this paper are the same as the ones in prior works.

SecurityModel

In our methods, sensor nodes are not tamper-resistant. In other words, the corresponding security credentials can be accessed after sensor nodes are physically compromised. Sensor nodes could be compromised by the adversary immediately after sensor deployment .The adversary has all of the legitimate credentials from the compromised nodes. After that, the adversary deploys two or more nodes with the same ID; i.e., replicas, into the net- work. Replicas can communicate and collude with each other in order to avoid replica detection in EDD . For example, replicas can share their credentials and can selectively be silent for a certain time if required after the collusion. Owing to the use of the digital signature function , the replicas cannot create a new ID as the nodes being not compromised before, because it is too difficult for the adversary to have the corresponding security credentials. Since the focus of this paper is on the node replication attack, despite many security issues on sensor networks such as key management , replay attack , worm hole attack [9], Sybil attack , se- cure query , etc., we assume that they can be well handled.

V. THE PROPOSED METHODS

In this section, our proposed algorithms, eXtremely Efficient Detection (XED) and Efficient Distributed Detection (EDD), for replica detection in mobile networks .

XED

The idea behind XED is motivated by the observation that, if a sensor node u meets another sensor node v at an earlier time and u sends a random number to v at that time, then, when u and v meet again, u can ascertain whether this is the node met before by requesting the random number. Note that, in XED, we assume that the replicas cannot collude with each other but this assumption will be removed in our next solution. In addition, all of the exchanged messages should be signed un- less specifically noted.

Specifically, the XED scheme is composed of two steps: an offline step and an online step. The former is executed before sensor deployment while the latter is executed by each node after deployment.

offline Step. A security parameter b and a cryptographic hash function h() are stored in each node. Additionally, two arrays Lr(u), and Ls(u) , of length n , which keep the received random numbers and the materials used to check the legitimacy of received random numbers, respectively, along with a set B(u) rep- resenting the nodes having been blacklisted by u , are stored in each node u . Lr(u), and Ls(u) are initialized to be zero-vectors. is initialized to be empty.

Online step. If u encounters v for the first time , u randomly generate $\alpha \in [1,2b-1]$,computes $h(\alpha)$,sends $h(\alpha)$, to v, and stores $Ls(u)[v]= \alpha$. Note that it encounters v for first time if $Ls(u)[v]=0$.

The same procedure applied for node v.the pseudo code of the online step of XED can be found in fig1.

Algorithm: XED-On-line-Step
// this algorithm is performed by the node u at each time t
// $v_1, \ldots, v_d$ are the neighbors of u
// $\{v_1, \ldots, v_d\} \notin \mathcal{B}^{(u)}$
1: send $\mathcal{L}_r^{(u)}[v_1], \ldots, \mathcal{L}_r^{(u)}[v_d]$ to $v_1, \ldots, v_d$, respectively
2: receive $\mathcal{L}_r^{(v_1)}[u], \ldots, \mathcal{L}_r^{(v_d)}[u]$
3: **for** $\kappa = 1$ to d
4: **if** $h(\mathcal{L}_s^{(u)}[v_\kappa]) = \mathcal{L}_r^{(v_\kappa)}[u]$
5: choose $\alpha \in [1, 2^b - 1]$ and set $\mathcal{L}_s^{(u)}[v_\kappa] = \alpha$
6: calculate $h(\alpha)$ and send $h(\alpha)$ to v_κ
7: **else**
8: set $\mathcal{B}^{(u)} = \mathcal{B}^{(u)} \cup \{v_\kappa\}$

Fig:2 online step of the XED scheme

Note that B(u) could be different for different nodes. This can be attributed to the fact that each node detects the replica by itself and will detect the replica at different time. Nonetheless, we can guarantee that the replica will be blacklisted by all nodes eventually.

EDD

Algorithmic Description of EDD: The idea behind EDD is motivated by the following observations. The maximum number of times Y1, that node u encounters a specific node V, should be limited with high probability during a fixed period of time, while the minimum number of times Y2 that encounters replica with same ID v, should be larger than a threshold during the same period of time. According to these observations, if each node can discriminate between these two cases, it has the ability to identify the replicas. Different from XED, EDD assumes that the replicas can collude with each other. In addition, all of the exchanged messages should be signed unless specifically noted.

Particularly, the EDD scheme is composed of two steps: an offline step and an online step. The offline step is performed before sensor deployment. . The goal is to calculate the parameters, including the length T of the time interval and the threshold used for discrimination between the genuine nodes and the replicas. On the other hand, the online step will be performed by each node at each move.

Offline Step. The offline step of EDD is shown in Fig. 2. The array L(u) of length n1,s1 is used to store the number of encounters with every other node in a given time interval, while set B(u) contains the IDs having been considered by u as replica. Let u1 and u2 be the expected number of encounters with the

genuine nodes and replicas, respectively .Let σ 1and σ2.
Here, an intrinsic assumption for the calculation of Y1 and Y2(in fig2) is that the random variables representing the number of encounters with genuine nodes and replicas are Gaussian distributed.

Algorithm: EDD-Off-line-Step
1: set $T = 1$ and $\mathcal{B}^{(u)} = \emptyset$, $u \in [1, n]$
2: set $\mathcal{L}^{(u)}[i] = 0$, $1 \leq i \leq n$, $u \in [1, n]$
3: **repeat**
4: $T = T + 1$,
5: calculate μ_1, μ_2, σ_1^2, and σ_2^2
6: set $Y_1 = \mu_1 + 3\sigma_1$ and $Y_2 = \mu_2 - 3\sigma_2$
7: **until** $Y_1 < Y_2$
8: set $\psi = \frac{Y_2 - Y_1}{2}$

Fig. 3. Offline step of the EDD scheme.

Note that, in some cases, the setting of Y1>Y2 is acceptable because the network planner would like to make a trade-off between the detection time and detection accuracy
Online Step. The online step of the EDD scheme is shown in Fig. 4. Each node locally maintains a counter t to record the elapsed time after the beginning of each time interval . After time T units is reached i.e t>T.

Algorithm: EDD-On-line-Step
// this algorithm is performed by node u at each time t
// $v_1, \ldots, v_d$ are the neighbors of u
// $\{v_1, \ldots, v_d\} \notin \mathcal{B}^{(u)}$
1: broadcast beacon b_u //$b_u = \langle u \rangle$ contains the ID of u
2: **if** $t \neq t_0$
3: receive beacons $b_{v_1}, \ldots, b_{v_d}$
4: **for** $\kappa = 1$ to d
5: $\mathcal{L}^{(u)}[v_\kappa] = \mathcal{L}^{(u)}[v_\kappa] + 1$
6: **if** $\mathcal{L}^{(u)}[v_\kappa] > \psi$ **then** set $\mathcal{B}^{(u)} = \mathcal{B}^{(u)} \cup \{v_\kappa\}$
7: **else** //$t = t_0$
8: set $\mathcal{L}^{(u)}[s_\kappa] = 0, \kappa = 1, \ldots, n$

Fig. 4. Online step of the EDD scheme

.

Finally, since the effectiveness of EDD relies on the fact that each node faithfully and periodically broadcasts its ID, a strategy called *selective silence* could be taken by the replicas to compromise the detection capability of EDD.

Both approaches are able to contain selected silence. They differ in the sense that the passive approach is purely localized and takes relatively longer time to find the replica with selected silence, while the active approach requires the cooperation among sensor nodes but can immediately detect the replica with selected silence.

VI. DISCUSSION

Since all of the existing detection algorithms for mobile net- works rely on the verification of sensors' locations at different times, time synchronization is essential. Otherwise, the detection accuracy could significantly drop because the genuine node (the replica) may be falsely regarded as the replica (genuine node). Thus, that XED and EDD do not need time synchronization among nodes to achieve detection works as a distinguished feature of our methods.

One characteristic that deserves to be mentioned is that the solutions for static networks provide a detection algorithm that "can detect the replicas" without mentioning "when the network owner should apply the detection algorithm." The drawback is that the network owner has to be aware of the existence of the replicas. Afterward, the network owner resorts to the detection algorithms to identify the replicas. In contrast, our proposed algorithms automatically detect the replica anytime and any- where.
In the algorithms adopting the witness-finding strategy, the spatial distribution of witness nodes is

usually an evaluation metric of the underlying detection algorithms. Ideally, it is uniformly distributed over the sensing region. Nevertheless, this evaluation metric is specific for the algorithms adopting the wit- ness-finding strategy due to the need of witness nodes in their methods, and is not required in our proposed algorithms.

VII. CONCLUSION

Two replica detection algorithms , are proposed. for mobile sensor networks, XED and EDD, Although XED is not resilient against collusive replicas, its detection framework, , *challenge-and-response*, is considered novel as compared with the existing algorithm. Different from XED,EDD can resilient against collusive replicas, its detection framework , so it can achieves unique characteristics, including network-wide time synchronization avoidance and network-wide revocation avoidance, in the detection of node replication attacks.

REFERENCES

[1]. G. Cormode and S. Muthukrishnan, "An improved data stream summary the count-min sketch and its applications," *J. Algorithms*, vol.55, no. 1, pp. 56–75, 2005.

[2]. M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "A randomized, efficient, and distributed protocol for the detection of node replication at- tacks in wireless sensor networks," in *Proc. ACM Int. Symp. Mobile AdHoc Networking and Computing (MobiHoc)*, Montreal, Canada, 2007, pp. 80–89.

[3]. M. Conti, R. D. Pietro, and A. Spognardi, "Wireless sensor replica detection in mobile environment," in *Proc. Int. Conf. Distributed Computing and Networking (ICDCN)*, Hong Kong, China, 2012, pp.249–264.

[4]. J. Ho, M. Wright, and S. K. Das, "Fast detection of replica node attacks in mobile sensor networks using sequential analysis," in *Proc. IEEE Int. Conf. Computer Communications (INFOCOM)*, Brazil, 2009, pp.773–1781.

[5]. D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Comput.*, pp. 153–181, 1996.

[6]. T. Karagiannis, J. L. Boudec, and M. Vojnovic, "Power law and expo- nential decay of inter contact times between mobile devices," in *Proc.ACM Int. Conf. Mobile Computing and Networking (MobiCom)*, Mon- treal, Canada, 2007, pp. 183–194.

[7]. M. Luk, G. Mezzour, A. Perrig, and V. Gligor, "MiniSec: A secure sensor network communication architecture," in *Proc. Int. Conf. Infor- mation Processing in Sensor Networks (IPSN)*, Cambridge, MA, USA,2007.

[8]. B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node repli- cation attacks in sensor networks," in *Proc. IEEE Symp. Security and Privacy (S&P)*, Oakland, CA, USA, 2005, pp. 49–63.

[9]. K. Xing and X. Cheng, "From time domain to space domain: Detecting replica attacks in mobile ad hoc networks," in *Proc. IEEE Int. Conf.Computer Communications (INFOCOM)*, San Diego, CA, USA, 2010, pp. 1–9.

[10]. M. Zhang, V. Khanapure, S. Chen, and X. Xiao, "Memory efficient protocols for detecting node replication attacks in wireless sensor net-works," in *Proc. IEEE Int. Conf. Network Protocols (ICNP)*, Princeton, NJ, USA, 2009, pp. 284–293.