Volume 4, Issue 2, February 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

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

Ms Sidhhi Raut [1] , Prof Vrunda Bhusari[2]

Department of Computer Science and Engineering

Bhivarabai Sawant Institute of Technology & Research (BSIOTR), 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 wireless 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:-

A new set of security challenges arises in sensor networks due to the fact that current sensor nodes lack hardware support for tamper-resistance and are often deployed in unattended environments where they are vulnerable to capture and compromise by an adversary. A serious consequence of node compromise is that once an adversary has obtained the credentials of a sensor node, it can surreptitiously insert replicas of that node at strategic locations within the network. These replicas can be used to launch a variety of insidious and hard-to-detect attacks on the sensor application and the underlying networking protocols. In a centralized approach for detecting node replication, when a new node joins the network, it broadcasts a signed message (referred to as a location claim) containing its location and identity to its neighbors. One or more of its neighbors then forward this location claim to a central trusted party (e.g., the base station). With location information for all the nodes in the network, the central party can easily detect any pair of nodes with the same identity but at different locations. Hence, a distributed solution is desirable. Distributed approaches for detecting node replications are based on location information for a node being stored at one or more witness nodes in the network. When a new node joins the network, its location claim is forwarded to the corresponding witness nodes. If any witness receives two different location claims for the same node identity (ID), it will have detected the existence of replica and can take appropriate actions to revoke the node's credentials. The basic challenge for any distributed protocol detecting node replicas is to minimize communication and per node memory costs while ensuring that the adversary cannot defeat the protocol.

II. 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.

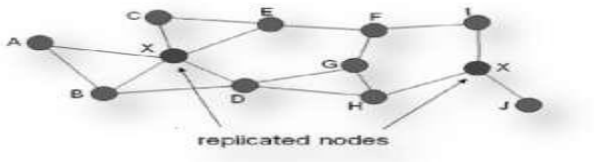


Fig 1 Node Replication Attack

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 witness-finding strategy for every units of time can also keep witness-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 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 effectiveness 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. Methodology

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. 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 neighbours. 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.

THE PROPOSED METHODS

In this section, we are studied 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 Fig 2.

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

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. 3. 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 fig3) 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.

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

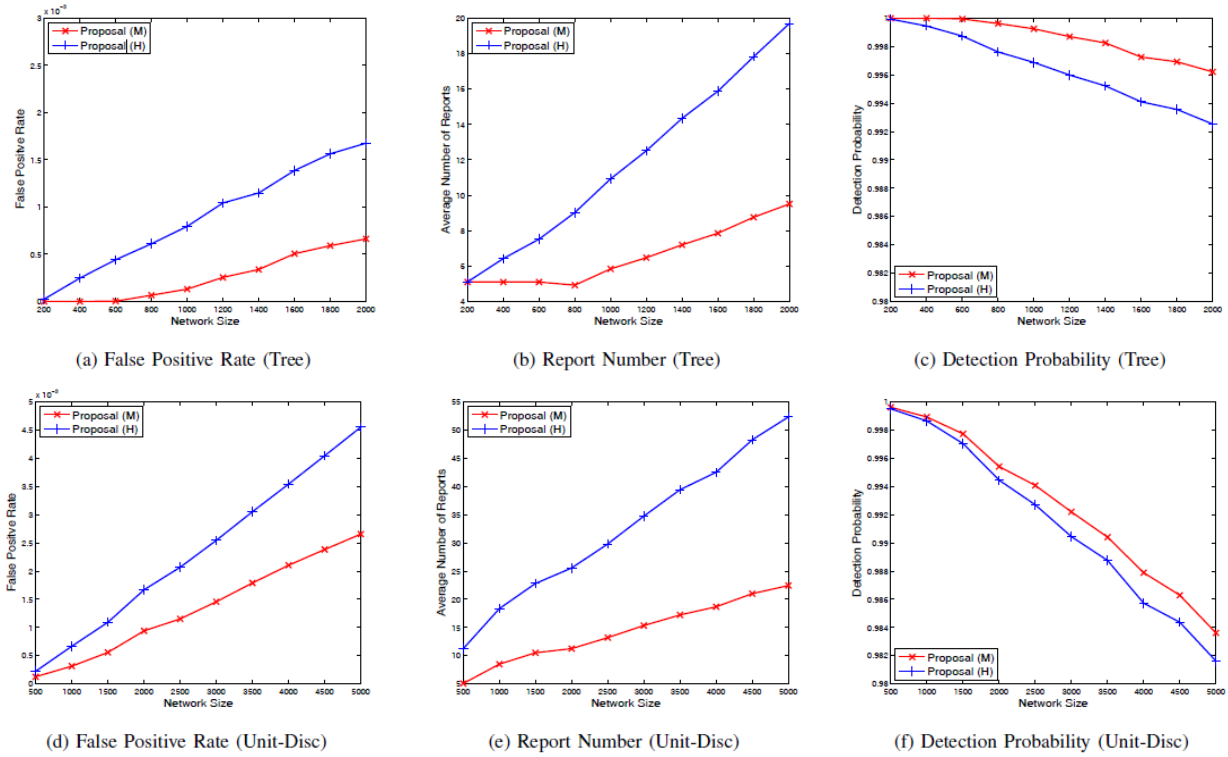
The effectiveness of EDD relies on the fact that each node faithfully and periodically broadcasts its ID, a strategy called selective silence could be taken by the replicas to compromise the detection capability of EDD. Both approaches are able to contain selected silence. They differ in the sense that 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.

SECURE BLOOM FILTER

Bloom filter, conceived by Bloom [6], is a simple, space-efficient probabilistic data structure that succinctly represents a set in order to support membership queries. Due to its distinguished space advantages and excellent distributed properties, bloom filter has been widely used in numerous areas, such as web cache sharing [4], distributed storage system [5]. Typically, a Bloom filter is implemented as a bit-array of m bits associated with h different hash functions, each of which maps a set element to one of the m array positions in a uniformly random manner. All bits in an initial Bloom filter are set to 0, standing for an empty set. To insert an element u into a set represented by a Bloom filter BF, h array positions are calculated by hash functions on u and the bits at those positions in BF are set to 1. Correspondingly, when it is required to check the membership of an element v within the Bloom filter BF, supplying v to hash functions outputs h array positions if any of the bits at the h positions is 0, then element v does not belong to the set; otherwise, the element is claimed to be a member of the set. It is clear that there is no false negative in the Bloom filter membership verification—an element which tests negative within a Bloom filter definitely is not a legitimate member of the set. On the other hand, Bloom filter may yield false positive a member outside the set passing the membership verification on the Bloom filter.

SIMULATION DESIGN

Network Scenarios: We run simulations in two network scenarios. The first one is an abstract hierarchical network modeled as a Tree rooted on the base station, which fairly emulates many wireless sensor networks. We generate a tree topology in a way that the number of a non-leaf node's children is uniformly selected from four, five, and six. In our experiments, the network size for the Tree Topology increases from 200 to 2000 at a step of 200.The other one is Unit-Disc Graph, in which nodes are uniformly deployed in a 500_500 square and nodes follow the standard unit-disc bidirectional communication model. We run simulations under the Unit-Disc Graph for network sizes ranging from 500 to 5000 with a space of 500 between. In order to imitate a dense sensor network and maintain consistence, for networks of different sizes, we adjust the node communication range such that the average node degree keeps the approximate 20. This kind of wireless sensor network simulation scenario models random deployed sensor networks and has been widely used in many literatures, such as [3].



(a) False Positive Rate (Tree) (b) Report Number (Tree) (c) Detection Probability (Tree)

(d) False Positive Rate (Unit-Disc) (e) Report Number (Unit-Disc) (f) Detection Probability (Unit-Disc)

Fig 2 Simulation Results on Protocols Communication Performances

IV. Experimental Results

In figures of this paper, "Proposal", "Native", "(M)", and "(H)" respectively stand for our proposed data aggregation protocol, the native scheme, mild workload, and heavy workload. The protocols communication performances along with their comparison are depicted in Figure 1. For general sensor network protocols, the communication cost is often represented by average transmission bits per node (Fig. 1a and Fig. 1d). For a data aggregation scheme, the energy

consumption of aggregators and the balance among them are critical performance metrics; so we also measure the means (Fig. 1b and Fig. 1e) and the standard deviations (Fig. 1c and Fig. 1f) of bits sent by aggregators (intermediate nodes for the native scheme). The simulation results clearly demonstrate that for all four test cases, our protocol remarkably outweighs the native scheme on all three metrics; not only the protocol reduces the overall energy consumption and reserves previous energy of aggregators, but also it achieves spleen did balance among aggregators, which is highly desired for data aggregation scheme. Moreover, as those measurements for our protocol gently grow with n, it proves that the proposed protocol scales to network size well, which is quite appreciated for large-size sensor networks[8].

The simulation results regarding the protocol's security are illuminated in Fig. 2. Specifically, Fig. 2a and Fig 2d depict the protocol false positive rates; Fig. 2b and Fig 2e show the average report numbers; Fig. 2c and Fig. 2f exhibit the detection probabilities. The false positive rates of our protocol are fairly small, compared to typical Bloom filter applications. Lastly, the high detection probabilities, all of which are greater than 98% in the simulations, exemplify the strong security of our protocol.

V. Conclusion:-

In this work on Two replica detection algorithms , are studied 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] T. Karagiannis, J. L. Boudec, and M. Vojnovic, "Power law and exponential 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.

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

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

[4] 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.

[5] C. Bettstetter, H. Hartenstein, and X. P. Costa. Stochastic Properties of the Random Waypoint Mobility Model. *Wireless Networks*, vol. 10, no. 5, pp. 555-567, 2004.

[6] 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.

[7] M. Conti, R. D. Pietro, L. V. Mancini, and A. Mei. Emergent properties: detection of the node-capture attack in obile wireless sensor networks. In *ACM WiSec*, 2008.

[8] Zhijun Li, Guang Gong On Data Aggregation with Secure Bloom Filter in Wireless Sensor Networks

