International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 03 | March - 2023 Impact Factor: 7.185 ISSN: 2582-3930

CLONE NODE DETECTION AND RECTIFICATION OF WIRELESS SENSOR NETWORK

Narayanasamy S [1], Suryavarshini .S[2] Sureshkumar S[3] M.P Revathi[4] S.Harthy Ruby Priya[5]

Assistant Professor[1&3,5], Department of Computer Science and Engineering, J.J College of Engineering and Technology, Trichy, India

Professor[4], Department of Computer Science and Engineering, J.J College of Engineering and Technology, Trichy, India

PG Student[2], Department of Computer Science and Engineering, Sri Krishna College of Engineering and Technology, Coimbatore, India

Abstract-A wireless sensor network is a collection of nodes organized into a cooperative network. Each node consists of processing capability, multiple types of memory (program, data and flash memories), a RF transceiver, a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. Sensor nodes that are deployed in hostile environments are vulnerable to capture and compromise. An adversary may obtain private information from these sensors, clone and intelligently deploy them in the network to launch a variety of insider attacks. The defenses against clone attacks are not only very few, but also suffer from selective interruption of detection and high overhead (computation and memory).A few distributed solutions to address this fundamental problem are not satisfactory. First, they are energy and memory demanding to be used in the WSN-resource constrained environment. Further, they are vulnerable to the certain adversary models. Hence we make the solutions of this work in threefold. First, we analyze the desirable properties of a distributed mechanism for the detection of node replication attacks. Second, we show that the known solutions for this problem do not completely meet our requirements. Third, we propose a new self-healing, distributed hash table DHT-Based Protocol for the detection of node replication attacks, and we show that it satisfies the introduced requirements. Finally, extensive simulations show that our protocol is highly efficient in communication, memory, and computation.

Keywords: distributed hash table, tamper-proof, Replication attack

2. INTRODUCTION

The challenges in the hierarchy of detecting the relevant quantities, monitoring and collecting the data, assessing and evaluating the information, formulating meaningful user displays, and performing decision-making and alarm functions are enormous. These information's requiDHT by the smart environments is provided by Distributed Wireless Sensor Networks, which are responsible for sensing as well as for the initial stage of the processing hierarchy. Wireless Network Sensors is a mechanism which can facilitate large scale and real time data processing in a complex environment. A wireless network sensor consists of

large number of sensors which are interconnected to each other and all of them will communicate with the Base station. Sensor nodes have limited processing capability, storage and energy and bandwidth when you compare them to traditional desktop computers. The sensors have low power and less coverage. Wireless Sensor Networks finds enormous applications in Military **as** in detecting landmines in battle field; locating enemy location to name few**,** Homeland Security**,** which is responsible for security of United States, uses it for purpose of detecting any invasion of security. Sensors are use to protect vital Governmental Institution and public places in general from foreign elements. In Healthcare of Detecting tumor, cancerous growths in body are some of prime example of how sensors are used in field of healthcare. An environmental application involves keeping a check on climate change, global warming etc., Of Agriculture, Plantation, vegetation growth is for which sensors are used to accumulate data.

Due to their wide operating nature, they are often unattended, hence prone to different kinds of novel attacks. For instance, an adversary could eavesdrop all network communications; Thus leading to a variety of malicious activities. In general, countermeasures against node clone can be categorized into three categories: prevention schemes that inherently forbid cloned nodes to join network, centralized detection in which there exists a central, powerful party responsible for receiving reports and making judgments of node clone, and distributed detection where all nodes cooperatively process information and detect node clone in a distributed manner.

3. RELATED WORK

A. Prevention

A proposed the use of location-based keys to defend against several attacks, one of which is node clone attack. The identity-based cryptography is used in their protocol such that nodes' private keys are bounded by both their identities and locations. Once nodes are deployed, some trusted mobile agents travel around the sensor network and issue the location-based keys to sensor nodes. Since those location-based keys cannot be used in nodes at other locations, node clone attack is inherently frustrated. By similar arguments, we review key distribution protocols for sensor networks, and it can be claimed that some of them prevent node clone as well. For example, in schemes based on initial trust which assume that it takes adversaries a certain amount of time to compromise nodes after their deployment, valid keys only can be established during that safety period, and henceforth compromising nodes will not grant adversaries extra advantages, including the ability to cloned nodes. Those prevention schemes might be useful on particular applications, but their assumptions as trusted mobile agents and initial trust are too strong to be applicable in general cases.

B. Centralized Detection

In a simplest centralized detection approach, each node sends a list of its neighbor nodes and their locations to a base station, which then can find cloned nodes. The SET protocol [8] manages to reduce the communication cost of the approach above by constructing exclusive subsets such that each node belongs to one and only one disjointed subset, and the subset nodes information is reported to the base station by a subset leader. However, in order to prevent malicious nodes, an authenticated subset covering protocol has to be

performed, which considerably increases the communication burden and complicates the detection procedure. Brooks et al. [9] proposed a clone detection protocol in the context of random key pre distribution [10]. Technically, it is detecting compromised keys rather than cloned nodes. The basic idea is that the keys employed in random key pre distribution scheme should follow a certain pattern, and those keys whose usage exceeds a threshold can be thought to be suspicious. In the protocol, every node reports its keys to a base station, and then the base station performs an abnormality-based intrusion-detection-like statistical analysis to catch cloned keys. A common concern for this kind of approach is its high false negative and positive rates. Furthermore, the authors do not address how to assure malicious nodes to honestly report their keys, which is critical to the protocol effectiveness. As pointed out in [1], centralized approaches are prone to single point of failure, and the nodes surrounding the base station suffer an undue communication burden that may shorten the network's life expectancy. In general, a distributed, balanced detection scheme is more desirable.

C. Distributed Detection

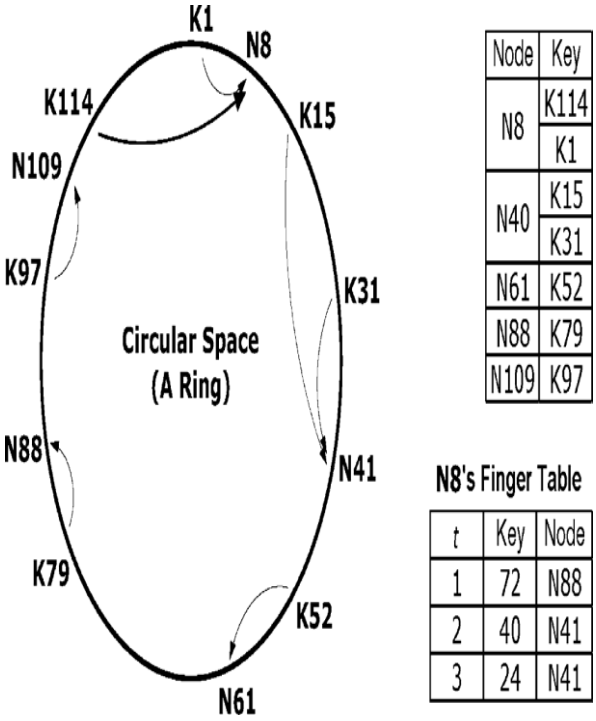
The straightforward node-to-network broadcasting [1] is a quite practical way to distributed detect the node clone, in which every node collects all of its neighbors identities along with their locations and broadcasts to the network. The main problem in this approach is its extremely high communication overhead. provided two probabilistic detection protocols in a completely distributed, balanced manner. Randomized multicast scheme distributes node location information to randomly selected nodes as inspectors, exploiting the birthday paradox to detect cloned nodes, while line-selected multicast scheme uses the topology of the network to improve detection—that is, in addition to inspector nodes, the nodes along the multicast path check the node clone as well. Unfortunately, to obtain acceptable detection probability, nodes have to buffer a great many of messages. Moreover, the communication cost in the randomized multicast is similar to that in the node-to-node broadcasting. For the procedure of choosing random inspectors, both schemes imply that every node is aware of all other nodes' existence, which is a very strong assumption for large-scale sensor networks and thus limits their applicability. Based on the geographic hash table, which maps a key into a geographical coordination, Zhu et al. [7] and Conti et al. [6] proposed several clone detection schemes. Their approaches rely on the nodes' knowledge of the general deployed geography of sensor networks. This prerequisite may hold in some circumstances,but cannot be guaranteed generally. Table I compares those distributed detection protocols along with our two proposystemS terms of requirements, communication cost, memory consumption, and detection level.

4. SYSTEM MODEL

4.1 DHT-BASED DETECTION PROTOCOL

The principle of our first distributed detection protocol is to make use of the DHT mechanism to form a decentralized caching and checking system that can effectively detect cloned nodes. Essentially, DHT enables sensor nodes to distributed construct an overlay network upon a physical sensor network and provides an efficient key-based routing within the overlay network. A message associated with a key will be transmitted through the overlay network to reach a destination node that is solely determined by the key; the source node does not need to specify or know which node a message's destination is—the DHT

key-based routing takes care of transportation details by the message's key. More importantly messages with a same key will be stored in one destination node. Those facts build the foundation for our first detection protocol. Before diving into the detection protocol, we briefly introduce DHT techniques. In principle, a distributed hash table is a decentralized distributed system that provides a key-based lookup service similar to a hash table : (key, record) pairs are stored in the DHT, and any participating node can efficiently store an retrieve records associated with specific keys. By design, DHT distributes responsibility of maintaining the mapping from keys to records among nodes in an efficient and balanced way, which allows DHT to scale to extremely large networks and be suitable to



Node	Key
N8	K114
	K1
N40	K15
	K31
N61	K52
N88	K79
N109	K97

N8's Finger Table

t	Key	Node
1	72	N88
2	40	N41
3	24	N41

Distributed Hash Table

balanced way, which allows DHT to scale to extremely large networks and be suitable to serve as a facility of distributed node clone detection. There are several different types of DHT proposals, such as CAN [14], Chord [15], and Pastry [16]. Generally, CAN least efficiency than others in terms of communication cost and scalability, and it is rarely employed in real systems. By contrast, Chord is widely used, and we choose Chord as a DHT implementation demonstrate our protocol. However, our protocol can easily migrate to build upon Pastry and present similar security and performance results.

Algorithm 1: handle a messagein the DHT-based detection, where is the current node'sChord coordinate, is the first node on the ring that succeeds key is the next th successor,
Output: NIL if the message arrives at its destination; otherwise, it is the ID of the next node that receives the message in the Chord overlay network
1:then
2: if then has reached destination
3: act as an inspector, see Algorithm 2
4: return NIL
5: for to do
6: if then destination is in the
next Chord hop
7: act as an inspector, see
Algorithm 2
8: return
9: for to do for normal DHT routing process
10: if then
11: return
12: return

Algorithm 2: Inspect a message to check for clone detection in the DHT-based detection protocol
1: verify the signature of
2: if found in cache table then
3: if has two distinct locations found clone, become a witness
4: broadcast the evidence
5: else
6: buffer into cache table

Architecture model: In our simulations, we randomly deploy 100 nodes with in a 1000m×1000m square. The transmission range is set to 50m. A tested our protocols in a standard network topology.

Location Update: This protocol is modified from RWS. This has been developed mainly to DHT reduce the memory cost of RWS protocol. This employs a table of values at each node to record the trace of the random walks. Each witness node will create a new entry in its table for every new location claim.

Witness Selection: A select increasing subareas of the network, and for each subarea, we count the number of witnesses present in the area after a run of the Detection protocol. Each subarea from the center of the unit square toward the external border provides an increment of five percent of the total area.

Attack Detection: The section we propose DHT (Distributed Hash table), a new protocol for the detection of clone attacks. DHT is similar in principle to the Randomized Multicast protocol, but with witnesses chosen pseudo randomly based on a network-wide seed.

Performance Evaluation: The average probability of detection is compared between RWS (indicated by DHT line) and MRWS (indicated by blue line) with different number of bytes to DHT , numbers of walk steps and different number of witness nodes MRWS trade increased communication overhead for stronger security properties.

5. PERFORMANCE EVALUATION

The average probability of detection is compared between RWS (indicated by DHT line) and MRWS (indicated by blue line) with different number of bytes to DHT, numbers of walk steps and different number of witness nodes MRWS trade increased

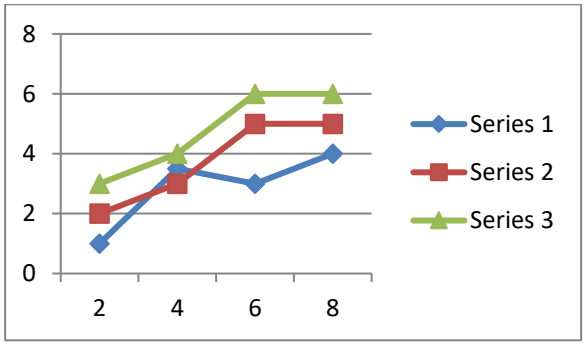


Figure 1 Clone Node Detection probability testing

communication overhead for stronger security properties.

6. CONCLUSION

The paper presents a few basic requirements an ideal protocol for distributed detection of node replicas. In particular the preliminary notion of ID-obliviousness and area-obliviousness that convey a measure of the quality of the node replicas detection protocol; that is, its resilience to a smart adversary. Moreover, it indicates that the overhead of such a protocol should be not only small, but also evenly distributed among the nodes, both in computation and memory. Further, it introduces new adversary threat models. However, a major

contribution of this paper is the proposal of a self-healing, randomized, efficient, and distributed protocol to detect node replication attacks. The analytical comparison of DHT with the state of the art solution (LSM) and proved that the overhead introduced by DHT is low and almost evenly balanced among the nodes; DHT is both ID-oblivious and area oblivious; furthermore, DHT outperforms LSM in terms of efficiency and effectiveness. Extensive simulations confirm these results. Lastly, also in the presence of compromised nodes, we can analytically show that DHT is more resilient in its detection capabilities than LSM.

REFERENCE

[1] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication
attacks in sensor networks," in Proc. IEEE Symp. Security Privacy, 2005, pp. 49–63.

[2] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica,"Looking up data in P2P systems," Commun. ACM, vol. 46, no. 2, pp. 43–48, 2003.

[3] Y. Zhang,W. Liu,W. Lou, andY. Fang, "Location-based compromisetolerant
security mechanisms for wireless sensor networks," IEEE J.Sel. Areas Commun., vol. 24, no. 2, pp. 247–260, Feb. 2006.

[4] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient security mechanisms for large-scale distributed sensor networks," in Proc. 10th ACM CCS, Washington, DC, 2003, pp. 62–72.

[5] R. Anderson, H. Chan, and A. Perrig, "Key infection: Smart trust for smart dust," in Proc. 12th IEEE ICNP, 2004, pp. 206–215.

[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 inwireless sensor networks," in Proc. 8thACMMobiHoc,Montreal, QC, Canada, 2007, pp. 80–89.

[7] B. Zhu, V. G. K. Addada, S. Setia, S. Jajodia, and S. Roy, "Efficient distributed detection of node replication attacks in sensor networks," in Proc. 23rd ACSAC, 2007, pp. 257–267.

[8] H. Choi, S. Zhu, and T. F. La Porta, "SET: Detecting node clones in sensor networks," in Proc. 3rd SecureComm, 2007, pp. 341–350.

[9] 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 Trans. Syst.s, Man, Cybern. C,Appl. Rev., vol. 37, no. 6, pp. 1246–1258, Nov. 2007.

[10] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed
sensor networks," in Proc. 9th ACM Conf. Comput. Commun. Security, Washington, DC, 2002, pp. 41–47.

[11] A. Shamir, "Identity-based cryptosystems and signature schemes," in Proc. CRYPTO, 1984, LNCS 196, pp. 47–53.