IJMETMR
ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Detection of Node Clone in Wireless Sensor Networks

Amaresh Chavali
P.G. Scholar,
Dept. of CSE,
Narasaraopet Engineering College,
Narasaraopet.

Lakshmi Natdh
Assistant Professor,
Dept. of CSE,
Narasaraopet Engineering College,
Narasaraopet.

Abstract:

Wireless sensor networks are vulnerable to the node clone, and several distributed protocols have been proposed to detect this attack. However, they require too strong assumptions to be practical for large-scale, randomly deployed sensor networks. In this paper, we propose two novel node clone detection protocols with different tradeoffs on network conditions and performance. The first one is based on a distributed hash table (DHT), by which a fully decentralized, key-based caching and checking system is constructed to catch cloned nodes effectively. The protocol performance on efficient storage consumption and high security level is theoretically deducted through a probability model, and the resulting equations, with necessary adjustments for real application, are supported by the simulations. Although the DHT-based protocol incurs similar communication cost as previous approaches, it may be considered a little high for some scenarios. To address this concern, our second distributed detection protocol, named randomly directed exploration, presents good communication performance for dense sensor networks, by a probabilistic directed forwarding technique along with random initial direction and border determination. The simulation results uphold the protocol design and show its efficiency on communication overhead and satisfactory detection probability.

INTRODUCTION

WIRELESS sensor networks (WSNs) have gained a great deal of attention in the past decade due to their wide range of application areas and formidable design challenges. In general, wireless sensor networks consist of hundreds and thousands of low-cost, resource-constrained, distributed sensor nodes, which usually scatter in the surveillance area randomly, working without attendance. If the operation environment is hostile, security mechanisms against adversaries should be taken into consideration. Among many physical attacks to sensor networks, the node clone is a serious and dangerous one [1]. Because of production expense limitation, sensor nodes are generally short of tamper-resistance hardware components; thus, an adversary can capture a few nodes, extract code and all secret credentials, and use those materials to clone many nodes out of off-the-shelf sensor hardware. Those cloned nodes that seem legitimate can freely join the sensor network and then significantly enlarge the adversary’s capacities to manipulate the network maliciously. For example, those vicious nodes occupy strategic positions and cooperatively corrupt the collected information. With a large number of cloned nodes under command, the adversary may even gain control of the whole network. Furthermore, the node clone will exacerbate most of inside attacks against sensor networks.

In this paper, we present two novel, practical node clone detection protocols with different tradeoffs on network conditions and performance. The first proposal is based on a distributed hash table (DHT) [2], by which a fully decentralized, key-based caching and checking system is constructed to catch cloned nodes. The protocol’s performance on memory consumption and a critical security metric are theoretically deducted through a probability model, and the resulting equations, with necessary adjustment for real application, are supported by the simulations. In accordance with our analysis, the comprehensive simulation results show that the DHT-based protocol can detect node clone with high security level and holds strong resistance against adversary’s attacks. Our second protocol, named randomly directed exploration, is intended to provide highly efficient communication performance with adequate detection probability for dense sensor networks. In the protocol, initially nodes send claiming messages containing a neighbor-list along with a maximum hop limit to randomly selected neighbors; then, the subsequent message transmission is regulated by a probabilistic directed technique to approximately maintain a line property through the network as well as to incur sufficient randomness for better performance on communication and resilience against adversary. In addition, border determination mechanism is employed to further reduce communication payload. During forwarding, intermediate nodes explore claiming messages for node clone detection. By design, this protocol consumes almost minimal memory, and the simulations show that it outperforms all other detection protocols in terms of communication cost, while the detection probability is satisfactory.

PREVENTION

Zhang et al. [3] 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 [4], [5]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.

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 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.

DISTRIBUTED DETECTION

The straightforward node-to-network broadcasting is a quite practical way to distributively 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. Parno et al.[1] 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.

NETWORK MODEL

We consider a large-scale, homogeneous sensor network consisting of resource-constrained sensor nodes. Analogous to previous distributed detection approaches; we assume that an identity-based public-key cryptography facility [11] is available in the sensor network. Prior to deployment, each legitimate node is allocated a unique ID and a corresponding private key by a trusted third party. The public key of a node is its ID, which is the essence of an identity-based cryptosystem. Consequently, no node can lie to others about its identity. Moreover, anyone is able to verify messages signed by a node using the identity- based key. Let and denote the public and private keys of node , respectively, and represent the signature of signed by node . We also assume that every sensor node can determine its geographic location and current relative time via a secure localization protocol and a secure time synchronization scheme, respectively. A number of those mechanisms have been proposed, which can be referred to in [12]. We do not specify the particular selections of secure localization and time synchronization schemes for our protocols since they are comparatively irrelevant to our proposals. There may or may not be a powerful base station in our modeled network, but there should exist a trusted role named initiator that is responsible for initiating a distributed detection procedure. Otherwise, an adversary can readily launch a denial-of-service (DoS) attack to the system by repeatedly mobilizing the sensor network to conduct the clone detection protocol and exhausting nodes energy.

GENERAL DETECTION GUIDELINES

Relying on the identity-based cryptography, secure localization, and secure time synchronization used in our network model, node clone in sensor networks can be determined by the occurrence of nodes with same ID appearing on reasonably distant locations at a designated time. Specifically, at the beginning time of a round of detection that is specified by the initiator, the information regarding the ID and location of every node is claimed by its neighbors for the clone detection. In this sense, the neighbors of a node are its observers. Subsequently, some nodes will be selected as inspectors to examine claiming messages for the purpose of clone detection. If an inspector successfully finds a clone, it becomes a witness, which will broadcast necessary evidence to inform all connected nodes revoking the cloned nodes. While the initiator is presumably trusted, the other roles (observer, inspector, and witness) might be compromised by the adversary and behavior maliciously. The four roles in our protocols are summarized .

PERFORMANCE METRICS

The following metrics are used to measure a protocol's performance And evaluate its practicability.

• Detection probability and security level: As a primary security requirement, a practical detection scheme should detect the occurrence of the attack with high probability. Thus, the detection probability is the most important security metric for a probabilistic clone detection scheme. On the other hand, if a detection protocol is deterministic in the sense that cloned nodes are always caught by witnesses, and it is also a fully symmetric approach in which nodes are equally likely to become witnesses prior to a round of detection procedure, we will use the number of witnesses to evaluate the security level because more witnesses improve protocol resilience against the adversary's potential attacks to witnesses.

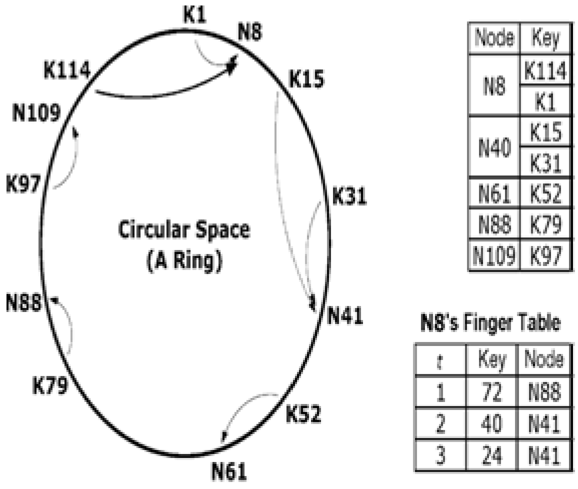
• Communication cost: Communication cost is always a crucial performance metric for sensor network protocols because usually energy is the most valuable resource for nodes, and message transmission consumes at least one order of magnitude power than any of the other operations [13]. For simplicity, we use the average number of messages sent per node to represent a protocol's communication cost.

• Storage consumption: Ordinary, low-cost sensor nodes are only equipped with a limited amount of memory; thus, any schemes requiring high storage will be considered as impractical.

• Balance: In a homogeneous sensor network, schemes are supported to consume the energy and memory in a balanced fashion. It should be avoided to create hot nodes that would be buffer-overflowed or die away quickly.

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 distributive 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. As a beginning of a round of DHT-based clone detection, the b initiator broadcasts the action message including a random seed. Then, every observer constructs a claiming message for each neighbor node, which is referred to as an examinee of the observer and the message, and sends the message with probability independently. The introduction of the claiming probability is intended to reduce the communication overwork in case of a high-node-degree network. In the protocol, a message's DHT key that determines its routing and destination is the hash value of concatenation of the seed and the examinee ID. By means of the DHT mechanism, a claiming message will eventually be transmitted to a deterministic destination node,whichwill cache the ID-location pair and check for node clone detection, acting as an inspector. In addition, some intermediate nodes also behave as inspectors to improve resilience against the adversary in an efficient way.



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

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

hash value of the node's MAC address. All nodes divide the ring into segments by their Chord points. Likewise, the key of a record is the result of the hash function. Every node is responsible for one segment that ends at the node's Chord point, and all records whose keys fall into that segment will be transmitted to and stored in that node. As the kernel of efficient key-based routing, every node maintains a finger table of size to facilitate a binary-tree search. Specifically, the finger table for a node with Chord coordinate contains information of nodes that are respectively responsible for holding the keys: for . If two nodes are within the ring-segments distance, they are each other's predecessor and successor by the order of their coordinates, with respect to predefined . In theory, a Chord node only needs to know its direct predecessor and finger table. To improve resilience against network churn and enhance routing efficiency, every node additionally maintains a successor table, containing its successors. Typical values of are between 10 and 20.

PERFORMANCE ANALYSIS OF DHT-BASED PROTOCOL

For the DHT-based detection protocol, we use the following specific measurements to evaluate its performance:

- average number of transmitted messages, repr esenting the protocol's communication cost;
- average size of node cache tables, standing for the protocol's storage consumption;
- average number of witnesses, serving as the protocol's security level because the detection protocol is deterministic and symmetric.

For simplicity, we hereby assume that all nodes, including compromised ones, abide by the detection protocol. Later in Section VI, we will see that the malicious behaviors such as discarding claiming messages only slightly affect those performance measurements. In this detection protocol, claiming messages associated with a same examinee's ID will be transported to one destination node. Because there are examinees and potential destinations, and due to the good pseudo-randomness of the Chord system, on average, every node stores one record in its cache table associated with one examinee's ID as its destination, regardless of the number of claiming messages per examinee.

SIMULATIONS FOR DHT-BASED PROTOCOL

We implement the DHT-based detection protocol and run Simulations to evaluate performance compre hensively on the OMNeT++ framework [17]. We design the simulations in two network scenarios. The first is an abstract network following a random graph model. By definition, a random graph is a graph that is generated by starting with a set of vertices and adding edges between them at random. The other one is a practical unit-disk graph, in which nodes are uniformly deployed in a square and follow the standard unit-disk bidirectional communication model. In our simu lations, node communication ranges are dynamically adjusted such that the average node degree approximates The average size of cache tables for integrity nodes and the average witness number are illuminated in Fig. 2(c) and (d), respectively, which clearly indicates that those two performance cloned nodes, and the claiming probability increases from 10% to 100%. Since network scenarios do not affect the results on the two performance metrics, we run the simulation only on random graph. For the purpose of comparison, we test the performance for both cases

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

that cloned nodes drop messages and comply with protocol (no-dropping).

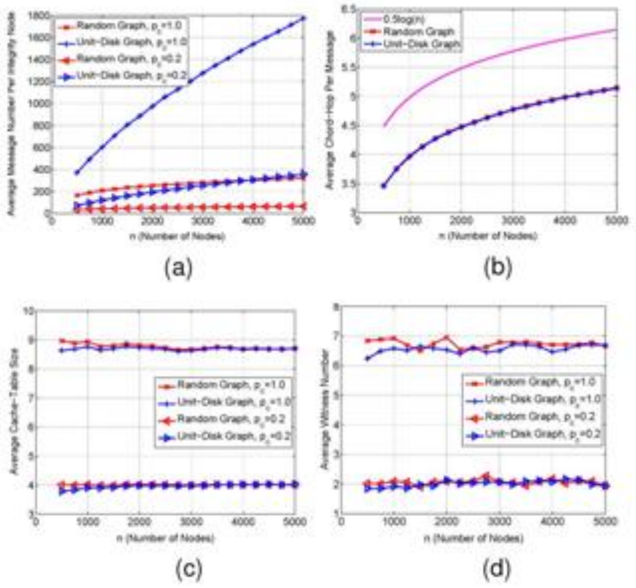


Fig. 2. Simulation results of the DHT-based detection on varying network sizes, where there are two cloned nodes, finger table size , successors table size g=16, and node d=20 degree ; no message-discarding. (a) Communication cost. (b) Chord-hop per message. (c) Storage consumption. (d) Security level.

Results on Different Numbers of Cloned Nodes We develop the second simulation to evaluate the protocol's performance on the different numbers of cloned nodes. We run simulations with one network size , and the cloned node number increases from 2 to 100. We test each case with 10 runs, and for each run we repeat 200 rounds of node detection, in each of which a seed is randomly generated and nodes are randomly chosen as clones. Fig. 3 depicts the simulation results about the average size of cache tables for integrity nodes and the average number of witnesses, which support our security arguments in Section IV-C. In particular, we can see that the protocol shows strong resilience against message-discarding by cloned nodes. Even if there are 10% nodes that maliciously discard messages, the number of witnesses is pretty high. In fact, the more cloned nodes, the less the size of cache tables for integrity nodes as storage consumption and the more witnesses as security level.

Therefore, we really only need to consider the boundary case of for performance measurements. In Fig. 3, when there are more than 1% cloned nodes, the simulation results for random graph and unit-disk graph are evidently distinct. This is because the message-dropping by malicious nodes affects the performance to a different extent. As implied in Fig. 2(a), the average transmission hop of claiming messages in unit-disk graph is greater than that in random graph, then messages are more likely to be dropped by cloned nodes in unit-disk graph.

EXPERIMENTAL RESULTS FOR RANDOMLY DIRECTED EXPLORATION

We implement the randomly directed exploration protocol on the same simulation framework as the previous protocol. Since the randomly directed exploration protocol relies on a local network topology, the random graph model cannot server for its simulations. Instead, we take the unit-disk graph as the sole network scenario. We choose a constant node degree and select as the priority range of the protocol. As a result, there are an average 2.5 neighbors in the priority zone of a node

CONCLUSION

Sensor nodes lack tamper-resistant hardware and are subject to the node clone attack. In this paper, we present two distributed detection protocols: One is based on a distributed hash table, which forms a Chord overlay network and provides the key-based routing, caching, and checking facilities for clone detection, and the other uses probabilistic directed technique to achieve efficient communication overhead for satisfactory detection probability. While the DHT-based protocol provides high security level for all kinds of sensor networks by one deterministic witness and additional memory-efficient, probabilistic witnesses, the randomly directed exploration presents outstanding communication performance and minimal storage consumption for dense sensor networks.

REFERENCES

[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 in wireless sensor networks," in Proc. 8thACM Mobi Hoc,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.