Volume 5, Issue 9, September 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Clone Node Detection in Wireless Sensor Network Using DHT and RDE

V. Monica
U.G Student,
Department of C.S.E
KMMITS, A.P., India

C. Govardhan
Asst. Professor,
Department of C.S.E
KMMITS, A.P., India

Abstract: ***WSN consists of distributed independent sensors to find physical or environmental conditions, such as temperature, sound, vibration, pressure to pass their data to each other and through the network to achieve common goal. Hardware of sensor node can be modified easily and it is expensive, so most Wireless Sensor Networks are consist of undefended sensor nodes. WSN's are generally deployed in an environments where An adversary or attacker can easily attack, analyze .To Detect the cloned node efficiently DHT Distributed Hash Table was proposed .DHT is Expensive for some scenarios.. To find clone nodes in such scenarios, our second distributed detection protocol, named Randomly Directed Exploration RDE, wasproposed which works efficiently in highly dense networks, by a probabilistic directed forwarding technique along with random initial direction and border determination.***

Key Words: Wireless Sensor network, Clones, DHT, RDE.

I. INTRODUCTION

A Wireless Sensor Network (WSN) is a collection of sensors with resources limitedly that collaborate eachother to achieve a common goal. WSNs can be deployed in dense environments to various applications . Due to their nature, they are often ununderfended, hence prone to different kinds of attacks. For instance, an adversary attack could view all internal network communications secretly; further, an adversary could capture nodes acquiring all the information stored therein—sensors are commonly assumed to be easily manipulated. Therefore, an attacker or adversary may replicate cloned sensors and deploy them in the network to launch a variety of malicious activities. This attack is referred to as the clone attack. Since a clone has rightful information (code and cryptographic material), it may participate in the network operations in the same way as a non compromised node;hence, cloned nodes can launch a variety of attacks.

The first protocol a distributed hash table (DHT) [2], by which a fully non centralized, key-basedcaching and checking system is constructed to catch clones. second protocol, named Randomly Directed Exploration (RDE), is intended to provide performance oriented efficient communication with adequate detection probability for dense sensor networks. In this protocol, initially nodes send claiming messages containing a neighbor-list along with a maximum hop limit to selected random nodes; then, the subsequent message transmission is regulated by a probabilistic directed technique to approximately maintain a incur sufficient randomness for better performance on communication and resilience against adversary.

II. RELATED WORK

2.1 Prevention

In [3] proposed the use of location-based keys to defend against several attacks, which include clone node attack.The cryptography based on identity is used in their protocol such that nodes' private keys are bounded by location as well as identity their . 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 protocolsfor 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 node.

2.2 Centralized Detection

In a simple detection approach, each node sends a list of its neighbor nodes and their locations to a base station, where then can find cloned nodes. The SET protocol [8] decreases the communication cost of the approach by constructing exclusive subsets such that each node belongs to one disjointed subset, and the subset nodes information was sent to the base station by a subset leader. However, in order to prevent malicious nodes, an covering protocol of authenticated subset was performed, which considerably increases the communication burden and complicates the detection procedure.

2.3 Distributed Detection

The node-to-network broadcasting [1] is a very practical way to distributed clone node detection , in which every node gathers all of its neighbors locations with identities and informs to the network. The major problem in this approach is its extremely high communication problems. 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, to identify cloned nodes. On Other hand line-selected multicast scheme uses the network topology to increase detection—that is, in addition to the inspector nodes, the nodes along the multicast path check the node clones as well. Moreover, the communication cost in the randomized multicast is similar to that in the node-to-node broadcasting

ADVERSARY MODEL

In this paper we consider a threat model as a sensor nodes are deployed in a hostile environment in which node is easy to capture and take control by an adversary. The limitation of adversary is that he can only compromise a very limited number of sensor nodes, and t adversary uses the captive compromised nodes to clone other genuine nodes and installs the cloned nodes in places that are decided intellectually . In addition, we assume each cloned node has at least one neighbor that remains as it is i.e intact. The adversary wants to hide the existence of clone. In our settings, this enemy is allowed to interfere with a detection protocol as follows. Initially , the cloned nodes may skip the regular detection procedures. Next , the nodes controlled by the adversary may duplicate, delete, or manipulate claiming messages that they forward. Finally, the adversary can capture some nodes , but it may take time, and the total number of nodes that an adversary can compromise is less. Node , that are not controlled by adversary will be defined as integrity nodes. The adversary may also try to abuse a detection protocol to frame innocent nodes as cloned such that they will be expelled from the network. This is called framing attack, and solutions should be provided to address this issue.

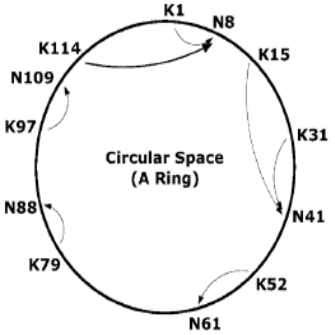
III. DHT-BASED DETECTION PROTOCOL

The main principle of this distributed detection protocol is to make use of the DHT mechanism to form a caching in decentralized way and and also checking system that detect cloned nodes effectively . DHT enables sensor nodes to constructdistributively an overlay other network upon a physical sensor networks and to provide an key-based routing in the overlay network. A message along with a key will be sent by using the overlay network to reach a destination node that is solely identify by the key; the source node does not need to mention which node is a message's destination is—the DHT key-based routing takes care of transportation details by the message's key.In DHT all the messages with a same key will be stored in single destination node. As a beginning of a round of DHT-based clone detection, the initiator transmits the action message including a random seed. Then, every observer constructs a claiming message that for each neighbor node, which is called as an examinee of the observer and the message, and sends the message with Pc probability independently. The introduction of the claiming probability is Pc whose intention is to decrease communication extra work in case of a network of high-node-degree . In this protocol, a message's DHT key which determines that its routing and destination determines 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,which will store the ID-location pair and check for node clone detection, acting as an inspector. In addition, some intermediate nodes will also behave as inspectors to improve resilience against the adversary in an efficient way.

3.1 Distributed Hash Table

Here we introduce DHT techniques. In principle, a distributed hash table is a non centralized distributed system which provides a key-based lookup service : (key, record) pairs are stored in the DHT, and any active node can store and retrieve records associated with their specific keys. DHT distributes responsibility of maintaining the mapping keys to records in a balanced way which is every efficient, in which it allows DHT to scale to extremelylarge networks and be suitable for distributed node clone detection.

The technical core of Chord [13] is to form a massive virtual ring in which every node is located at one point, owning a segment of the periphery. To achieve randomness on output, a hash function is used to map an arbitrary input into a -bit space, which can be declared a as a ring. Each node which is assigned with a Chord coordinate upon joining the network.



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

Fig. 1. Chord network example, where the key space is 7-bit ($b = 7$), seven records with different keys are stored in five nodes, and the successor table size $g = 2$. For node $N8$, its direct predecessor is $N109$, and its two successors are $N41$ and $N61$.

A node's Chord point's coordinate is the hash value of the node's MAC address. All nodes partition 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 Chord point of nodes , and all the records whose keys fall into that segment will be transmitted to and stored in the concerned node.

A demonstrative example of a Chord system with small parameters is given in Fig. 1. In this system, if node N8 wants to query a record with key K97 , it first looks up its successor table. Since K97 is not in (109, 61], namely (direct predecessor,the last successor], node N8 proceeds with the finger table andfinds that the next forwarding node should be N88 because 97 Є72: the first item in finger table corresponding to ,109: direct predecessor). When N88 receives this query aboutK97 , by checking its successor table with two nodes of N109 and N8, it determines the destination should be N109 , as 97 Є(88: itself, 109: the first successor]. When node N109 gets thequery, it knows itself be the destination because 97 Є (88: directpredecessor, 109: itself].

3.2 Protocol Details

Prerequistely all nodes build cooperatively a Chord overlayed network over the sensor network. Cloned node maynot participate in this procedure, but it does not give the many advantage of avoiding detection. The overlay network development is independent of node clone detection. As a result, Each direct predecessor and successor information is passed to each node in the Chord ring. And also each node stores or caches information of its consecutive successors in the corresponding *successorstable*. Many Chord systems uses this kind of storing mechanism to decrease the communication cost and enhance systems robustness. More importantly in our protocol, the facility of the successors table contributes to the economical selection of inspectors. Each detection round consists of three stages.

Stage 1: Initialization

To activate all nodes starting a new round of node clone detection, the initiator node uses a authentication scheme of broadcasting to release an *action message* including a monotonously increasing nonce, a random round seed, and an action time. The main purpose of nonce is to prevent adversaries from launching a Denial of Service attack by repeating broadcasting action messages

$$M_{\mathrm{ACT}} = \mathrm{nonce},\ \mathrm{seed},\ \mathrm{time}, \{\mathrm{nonce} \parallel \mathrm{seed} \parallel \mathrm{time}\}_{K^{-1}_{\mathrm{initiator}}}.$$

Stage 2: Claiming neighbors information

After receiving an action message, Each node verifies if the message nonce is greater than last nonce and if the message signature is valid. If both pass, the node updates the nonce and stores the seed. At the designated action time, the node operates as an observer that generates a claiming message for each neighbor (examinee) and transmits the message through the overlay network with respect to the claiming probability . The claiming message by observer for examinee is constructed by

$$M_{\alpha 4 \beta} = \mathrm{id}_\beta, L_\beta, \mathrm{id}_\alpha, L_\alpha, \{\mathrm{id}_\beta \parallel L_\beta \parallel \mathrm{id}_\alpha \parallel L_\alpha \parallel \mathrm{nonce}\}_{K^{-1}_\alpha}$$

where are locations L of α and β, respectively.

Stage 3: Processing claiming messages

A claiming message will be passed to its destination node via several Chord intermediate nodes. Only the source node, Chord intermediate nodes, and the destination nodes in the overlay network layer are necessary to process a message, whereas other nodes along the path simply pass the message to temporary targets. Algorithm 1 for handling a message is the kernel of our DHT-based detection protocol. If message has arrived at its destination. the algorithm returns NIL.

Algorithm 1: dht_handlemessage($M_{\alpha 4 \beta}$) : handle a message in the DHT-base detection, where y is the current node;s Chord coordiant, finger[i] is the first node on the ring that succeeds key (($y+2^{b-i}$)mod 2^b), i[1,t], successors[j] is the next jth successor, j [1,g]

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: key H(seed||id $_\beta$)
2. if key (predecessor, y] then { has reached destination}
3: inspect($M_{\alpha 4 \beta}$){act as an inspector,See Algorithm 2}
4: return NIL
5: for i=1 to g do
6: if key (y,successor[i]) then { destination is in the next Chord hop}
7: inspect ($M_{\alpha 4 \beta}$) {act as an inspector see Algorithm 2}
8: return successors[i]
9: for j=1 to t do {normal DHT routing process}
10: if key [y+ 2^{b-i}) mod 2^b,y) then
11: return finger[j]
12: return successors[g]

Algorithm 2: inspect ($M_{\alpha 4 \beta}$) :Inspect a message to check for clone detection in the DHT based detection protocol

1: verify the signature of $M_{\alpha 4 \beta}$
2: if id $_\beta$ found in cache table then

3: if id_β has two distinct location { foundclone, become witness}
4: broadcast the evidence
5: else
6: buffer ($M_{\alpha 4\,\beta}$) into cache table

IV. RANDOMLY DIRECTED EXPLORATION

In The DHT-based cloned nodes will be caught by one deterministic witness plus several probabilistic witnesses, which is the specialization of this DHT. . Certain sensor networks are prone to energy high consumption a Chord overlap network fails in such networks because of considerable communication cost,. To Address this disadvantage, The randomly directed exploration (RDE) was proposed, which decreases storage expenses withmode clone detection capabilities. Each node only necessary to know and store all neighbors IDs and locations in neighbor list.. For DHT and RDE, each node developing s a claiming message with signed version of its neighbor-list, and then tries to deliver its message to others which will compare with its own neighbor-list to find clone node. For a more dense network, broadcasting will drive all neighbors of cloned nodes to find the attack, but in fact one witness is sufficient that successfully identifies the clone node then informs the entire network would sufficient for the detection purpose.

To achieve, Initially , a claiming message needs to provide maximal hop limit, and initially it is sent to a random neighbor node. Then, the message subsequent transmission will roughly maintain a line. The line transmission property helps a message go through the network as fast as possible from a locally optimal perspective. In addition, we introduce border determination mechanism to significantly reduce communication cost. It is possible Only because each node knows its neighbors locations.

Algorithm 3: rde_processmessage (M_α): An intermediate node process a message

1:verify the signature of M_α
2: compare its own neighbor-list with neighbor-list in M_α
3: if found clone then
4: broadcast the evidence;
5: ttl←*ttl*-1
6: if ttl0 then
7: discard M_α
8: else
9: nextnode<= getnextnode($M_{\alpha)}$ {see Algorithm 4}
10: if nextnode=NIL then
11: discard M_α
12: else
13: forward

A. Protocol Description

One round of clone detection is still activated by the initiator. Subsequently, at the designated action time, each nodecreates its own neighbor-list including the neighbors IDs and locations, which constitutes the sole storage consumption of the protocol. Then, it, as an observer for all its neighbors, starts to generate a claiming message containing its own ID, location, and its neighbor-list. The claiming message by node is constructed by

$$M_\alpha = \mathrm{ttl}, \mathrm{id}_\alpha, L_\alpha, \mathrm{NeighborList}_\alpha, \{\mathrm{id}_\alpha \| L_\alpha \| \mathrm{NeighborList}_\alpha \| \mathrm{nonce}\}_{K_\alpha^{-1}}$$

where is time to live *ttl* (a.k.a. message maximum hop). Since *ttl* will be altered by intermediate nodes during transmission, it should not be authenticated. The observer will deliver the claiming message times

Algorithm 4: To determine getNextNode (Mα) the next node that receives the message that receives the message

1: determine ideal angle, target zone, and priority zone
2: **if** no neighbors within the target zone **then**
3: **return** NIL
4: **if** no neighbors within the priority zone **then**
5.nextnode<= the node closest to ideal angle
6.else
7.**nextnode<=** a probabilistic node in the priorityzone, with respect to its probability proportional
Toangle distance from priority zone border
8.returnnextnode

Essentially, Algorithm 4 contains the following three mechanisms.

• **Deterministic directed transmission**: The ideal direction can be measured if node receives a claim message from its predecessor node. In order to achieve the best effect of line transmission, the next destinationnode should be node which is nearest to the ideal direction.

• **Network border determination** :By considering network shape to decrease the communication cost. With many sensor network applications, there exist outside borders of network due to its physical constrains. Upon reaching its some border in the network, the claim message can immediately discarded. In determining the border parameter *target range* is utilised along with ideal direction to determine a *target zone*. When no neighbor is found in this zone, the current node will conclude that the message has reached a border, and then discard..

• **Probabilistic directed transmission** :In the probabilistic directed transmission, parameter *priority range*along with the ideal direction is used to specify a priority zone, in which the next node will be selected. When nonodes are located in that zone, the deterministic directed candidate within the target zone will be selected as the nextnode. If there are several nodes in the priority zone, their selection probabilities are proportional to their angle distances to priority zone border.

V. CONCLUSION

Manu sensor nodes lacks hardware which can't be manipulated and hence they are subjected to the node clone attack. In this paper we proposes: One is based on a distributed hash table, which forms a Chord overlay network which provides the key which is based on routing, storing , and identifying clone detection, and the other technique uses probabilistic directed to achieve decreased communication overhead for satisfactory detection probability. In future DHT can be enhanced and same can be used in dense sensor networks instead of using RDE.

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 attacksinwireless 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.

[12] R. Poovendran, C. Wang, and S. Roy, *Secure Localization and Time Synchronization for Wireless Sensor and Ad Hoc Networks*. New York: Springer-Verlag, 2007.

[13] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker,"A scalable content-addressable network," in *Proc. SIGCOMM*, San Diego, CA, 2001, pp. 161–172

