International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 02 - Issue 03 || March 2017 || PP. 22-28

CLONE DETECTION USING CHORD ALGORITHM IN WIRELESS SENSOR NETWORK

HEMAVATHI.Y[1], KEERTHIGA.M[1], JEYAMOHAN .H[2]
[1]*U. G. STUDENTS, DEPT OF CSE, ALPHA COLLEGE OF ENGINEERING, CHENNAI,*
[2]*ASST PROF.DEPARTMENT OF CSE, ALPHA COLLEGE OF ENGINEERING, CHENNAI*

Abstract: In this paper, we propose an energy-efficient location-aware clone detection protocol in densely deployed WSNs, which can guarantee successful clone attack detection and maintain satisfactory network lifetime. Specifically, we exploit the location information of sensors networks.we use 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) in which Chord algorithm is used to detect the cloned node, every node is assigned with the unique key, before it transmits the data it has to give its key which would be verified by the witness node. The second one is based on the Distributed Detection Protocol which is same as DHT, but it is easy and cheaper implementation. Here every node only needs to know the neighbor-list containing all neighbor IDs and its locations.

Keywords: detect clone using chord algorithm and DHT protocol

I. INTRODUCTION

The objective of A wireless sensor network is a collection of nodes organized into a cooperative network. Each node consists of processing capability (one or more microcontrollers, CPUs or DSP chips), may contain multiple types of memory (program, data and flash memories), have a RF transceiver (usually with a single Omni-directional antenna), have a power source (e.g., batteries and solar cells), and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad hoc fashion. Systems of 1000s or even 10,000 nodes are anticipated. Such systems can revolutionize the way we live and work. Currently, wireless sensor networks are beginning to be deployed at an accelerated pace. It is not reasonable to expect that in 10-15 years that the world will be covered with wireless sensor networks with access to them via the Internet. This can be considered as the Internet becoming a physical network. This new technology is exciting with unlimited potential for numerous application areas including environmental, medical, military, transportation, entertainment, crisis management, homeland defense, and smart spaces.

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. 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 allsecret credentials, and use those materials to clone many nodes out of off-the-shelf sensor hardware.

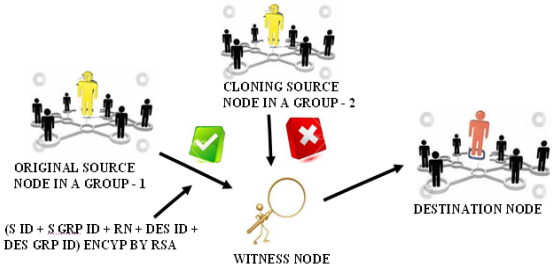

Fig. 1. Architecture

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.

II. RELATED WORK

The Wireless sensor networks are vulnerable to the node clone, and several distributed protocols have been proposed to detect this attack. So they require too strong assumptions to be practical for large-scale, randomly deployed sensor networks. 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. We use 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) in which Chord algorithm is used to detect the cloned node, every node is assigned with the unique key, before it transmits the data it has to give its key which would be verified by the witness node. If same key is given by another Node, then the witness node identifies the cloned Node. The second one is based on the Distributed Detection Protocol which is same as DHT, but it is easy and cheaper implementation. Here every node only needs to know the neighbor-list containing all neighbor IDs and its locations. So, that can detect node clone with high security level and holds strong resistance against adversary's attacks.

III. SYSTEM MODEL AND PROBLEM STATEMENT

In this work, we are implementing RDE protocol, by location based nodes identification, where every region/location will have a group leader. The Group leader will generate a random number with time stamp to the available nodes in that location. Witness nodes verify the random number and time stamp to detect the cloned node. The message is also encrypted for security purpose.

A. NETWORK CONSTRUCTION

In this module, we are going to connect the network. Here each node connected to the neighboring node and it is independently deployed in network area in this we give ID Number to each node in network. Here many nodes are interconnected and monitored by admin node.

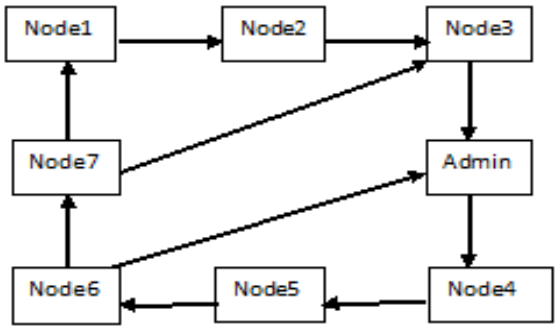

Fig. 2. Network construction

B. WITNESS NODE DISTRIBUTION

A major issue in designing a protocol to detect clone attacks is the selection of the witnesses. We will call '**Witness'** as a node that detects the existence of a node in two different locations within the same protocol run. If the adversary knows the future witnesses before the detection protocol executes, the adversary could subvert these nodes so that the attack goes undetected.

Here, we have identified two kinds of predictions:

1. ID-based prediction
2. Location-based prediction.

We say that a protocol for replica detection is ID-oblivious if the protocol does not provide any information on the ID of the sensors that will be the witnesses of the clone attack during the next protocol run. Similarly, a protocol is area-oblivious if probability does not depend on the geographical position of node in the

network. Clearly, when a protocol is neither ID-oblivious nor area-oblivious, then a smart adversary can have good chances of succeeding, since it is able to use this information to subvert the nodes that, most probably, will be the witness.

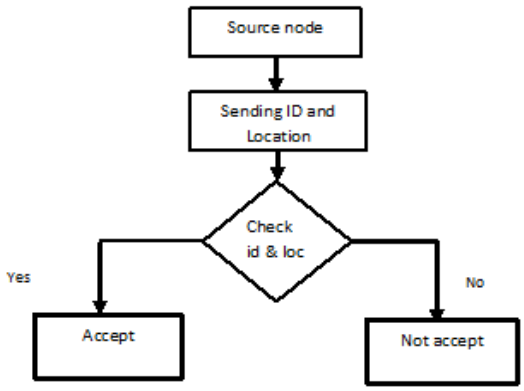

Fig. 3.Witness node distribution

C. VERIFICATION OF RANDOM NUMBER

Random Key pre-distribution security scheme is implemented in the sensor network. That is, each node is assigned a number randomly with Time Stamp from Group Leader. Then the Group Leader will transmit Random Number (Encrypted with RSA algorithm) which was generated with respect to that Time Stamp to the Witness node.

Witness node will now check the Random number which is generated with the User information. If both the data are matched, then the Witness node will confirm that this node is Genuine.

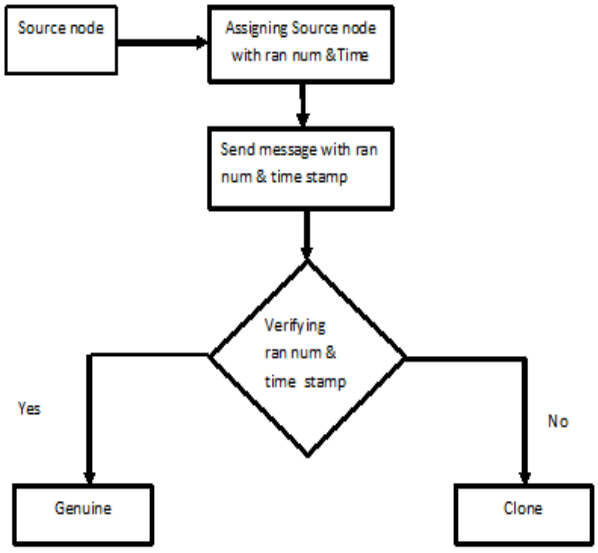

Fig.4.VERIFICATION OF RANDOM NUMBER

D. VERIFICATION OF USER ID

Each node is assigned an ID as individual once it is registered into the network and also an ID for the whole group (i.e.) Location ID is generated for each and every Location. That Node ID and Location ID are also appended with 1 (Encrypted with RSA algorithm). Then the Witness node will now check the node ID + Location ID which is generated with the UserInformation.If both the data are matched, then the Witness node will confirm that this node with that Location is Genuine.

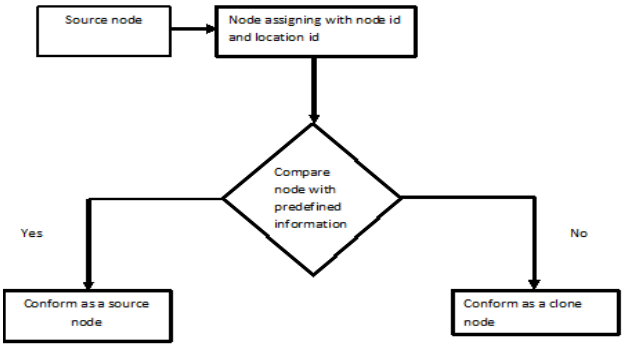

Fig. 5.VERIFICATION OF USER ID

E. CLONING DETECTION AND DATA TRANSFER

Only the Witness node confirms the Sender node, the data is send to the Destination, which is Genuine. If user specified information and the internal information are varied, then the Witness node will identify that Cloning or some Mal practice has occurred and the Packets are discarded by the witness node.

IV. PROPOSED METHODOLOGY

We use 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) in which Chord algorithm is used to detect the cloned node, every node is assigned with the unique key, before it transmits the data it has to give its key which would be verified by the witness node. If same key is given by another Node, then the witness node identifies the cloned Node.

The second one is based on the Distributed Detection Protocol which is same as DHT, but it is easy and cheaper implementation.

Here every node only needs to know the neighbor-list containing all neighbor IDs and its locations. So, that can detect node clone with high security level and holds strong resistance against adversary's attacks.

V. EXISTING ALGORITHMS

A. CHORD ALGORITHM

Chord is a protocol and algorithm for a peer-to-peer distributed hash table.A distributed hash table stores key-value pairs by assigning keys to different computers (known as "nodes"). A node will store the values for all the keys for which it is responsible. Chord specifies how keys are assigned to nodes, and how a node can discover the value for a given key by first locating the node responsible for that key.the process done by chord algorithm are Construction,Localization of nodes,Node joins and stabilization,Failure of nodes.

B. FINGER TABLE

Each node n maintains a routing table with up tomentries (which is in fact the number of bits in identifiers), called finger table. To avoid the linear search above, Chord implements a faster search method by requiring each node to keep a finger table containing up to m entries. Queries are passed around the circle via successor pointers. Requires traversing all Nodes to find the appropriate mapping.

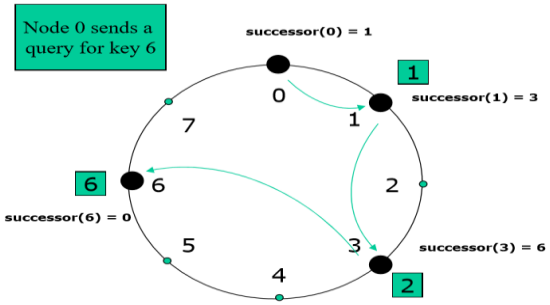

Fig 6. Finger Table

C. DISTRIBUTED HASH TABLES

A global view of data distributed among many nodes. Mapping nodes and data items into a common address space. Each DHT node manages a small number of references to other nodes. Queries are routed via a small number of nodes to the target node. Load for retrieving items should be balanced equally among all nodes. Robust against random failure and attacks. Provides a definitive answer about results.

DHT distributes responsibility for maintaining the mapping from keys to records among the nodes in an efficient, balanced way, which allows DHT to scale to extremely large networks and suitable for an infrastructure of distributed node clone detection. we choose Chord to demonstrate our protocol. Our protocol can easily migrate to be Pastry-based and get similar security and performance results. The principle behind Chord [17] is to form a massive virtual ring in which every node occupies one point, owning a segment of the periphery. A hash function is used to map an arbitrary input into an m-bit space, which can be conceived as a ring. Each node is assigned with a Chord position upon joining the network.For example, if the Chord positions of nodes α, β, andγ are 21, 96, and 182 (for m = 8) respectively, then β is responsible for keys 22–96, γ for keys 97–182, and α for keys 183–21.

ALGORITHM 1

```
// ask node n to find the successor of id
n. find_successor(id)
  //Yes, that should be a closing square bracket to match the opening parenthesis.
  //It is a half closed interval.
  if (id ∈ (n, successor] )
    return successor;
  else
    // forward the query around the circle
    n0 = closest_preceding_node(id);
  return n0. find successor(id);
 // search the local table for the highest predecessor of id
n. closest_preceding_node(id)
  for i = m down to 1
    if (finger[i]∈(n, id))
      return finger[i];
  return n;
```

The pseudocode to stabilize the chord ring/circle after node joins and departures is as follows:

```
// create a new Chord ring.
n.create()
  predecessor = nil;
  successor = n;
// join a Chord ring containing node n'.
n.join(n')
  predecessor = nil;
  successor = n'.find_successor(n);
// called periodically. n asks the successor
// about its predecessor, verifies if n's immediate
// successor is consistent, and tells the successor about n
n.stabilize()
  x = successor. Predecessor;
  if (x∈(n, successor))
    successor = x;
successor. Notify(n);
// n' thinks it might be our predecessor.
n.notify(n')
  if (predecessor is nil or n'∈(predecessor, n))
    predecessor = n';
// called periodically. refreshes finger table entries.
// next stores the index of the finger to fix
n.fix_fingers()
  next = next + 1;
  if (next > m)
    next = 1;

finger[next] = find_successor(n+        );
// called periodically. checks whether predecessor has failed.
n.check_predecessor()
  if (predecessor has failed)
    predecessor = nil;
```

V. PERFOMANCE ANALYSIS

A.Storage Cost and Number of Witnesses

Suppose all nodes, including clone ones, abide by the detection protocol, we analyze the average size of cacheable and the average number of witnesses theoretically. In our protocol, a claim message would be forwarded to a deterministic destination node, which should always be a witness if there are clones, and every node, on average, holds one record in its cacheable for a claimed node ID as the destination. In addition, the g predecessor nodes of the destination can act as inspectors, and thus may hold a copy of the record too, and have potential to become the witnesses.

Those g predecessors, defined as Ei for i ∈ [1,g], together with the destination, occupy a part of periphery of the Chord ring and partition it into g consecutive segments. Considering the good randomness of Chord systems, the length proportions of g segments in the part of periphery are randomly distributed. We denote the probability of one specific message would go through predecessor Ei by Pi. According to the algorithm in Fig. 1, a message will pass one of the g predecessors before finally reaching the destination with probability N−1 N (The probability of the source being the destination is 1 N). Therefore, the g probabilities Pi are randomly distributed under the condition of g i=1 Pi = N−1 N .

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]. Z. Zheng, A. Liu, L. X. Cai, Z. Chen, and X. Shen, "ERCD: An energy-efficient clone detection protocol in wsns," in Proc. IEEE INFOCOM, Turin, IT, Apr. 14-19 2013, pp. 2436–2444.

[2]. R. Lu, X. Li, X. Liang, X. Shen, and X. Lin, "GRS: The green, reliability, and security of emerging machine to machine communications," IEEE Communications Magazine, vol. 49, no. 4, pp. 28–35, Apr. 2011.

[3]. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," Computer Networks, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[4]. A. Liu, J. Ren, X. Li, Z. Chen, and X. Shen, "Design principles and improvement of cost function based energy aware routing algorithms for wireless sensor networks," Computer Networks, vol. 56, no. 7, pp. 1951–1967, May. 2012

[5]. T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 941–954, Jul. 2010.

[6]. P. Papadimitratos, J. Luo, and J. P. Hubaux, "A randomized countermeasure against parasitic adversaries in wireless sensor networks," IEEE Journal on Selected Areas in Communications, vol. 28, no. 7, pp. 1036–1045, Sep. 2010.

[7]. R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in VANETs," IEEE Transactions on Vehicular Technology, vol. 61, no. 1, pp. 86–96, Jan. 2012.

[8]. Z. M. Fadlullah, M. Fouda, N. Kato, X. Shen, and Y. Nozaki, "An early warning system against malicious activities for smart grid communications," IEEE Network, vol. 25, no. 5, pp. 50–55, May. 2011.

[9]. R. Lu, X. Lin, X. Liang, and X. Shen, "A dynamic privacy-preserving key management scheme for location based services in VANETs," IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1, pp. 127–139, Jan. 2012.

[10]. M. Conti, R. D. Pietro, L. Mancini, and A. Mei, "Distributed detection of clone attacks in wireless sensor networks," IEEE Transactions on Dependable and Secure Computing, vol. 8, no. 5, pp. 685–698, Sep.Oct. 2011.

[11]. B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in Proc. IEEE Symposium on Security and Privacy, Oakland, CA, USA, May. 8-11 2005, pp. 49–63.

[12]. Y. Zeng, J. Cao, S. Zhang, S. Guo, and L. Xie, "Random-walk based approach to detect clone attacks in wireless sensor networks," IEEE Journal on Selected Areas in Communications, vol. 28, no. 28, pp. 677–691, Jun. 2010.

[13]. B. Zhu, S. Setia, S. Jajodia, S. Roy, and L. Wang, "Localized multicast: Efficient and distributed replica detection in large-scale sensor networks," IEEE Transactions on Mobile Computing, vol. 9, no. 7, pp. 913–926, Jul. 2010.

[14]. Y. Xuan, Y. Shen, N. P. Nguyen, and M. T. Thai, "A trigger identification service for defending reactive jammers in WSN," IEEE Transactions on Mobile Computing, vol. 11, no. 5, pp. 793–806, May. 2012.

[15]. R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen., "BECAN: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 1, pp. 32–43, Jan. 2012.

[16]. J. Li, J. Chen, and T. H. Lai, "Energy-efficient intrusion detection with a barrier of probabilistic sensors," in Proc. IEEE INFOCOM, Orlando, FL, USA, Mar. 25-30 2012, pp. 118–126.