COMPUSOFT, An international journal of advanced computer technology, 3 (6), June-2014 (Volume-III, Issue-VI)

ISSN:2320-0790

An International Journal of Advanced Computer Technology

Detection of Clone Nodes in Wireless Sensor Networks Using RED and Chord Algorithm

R.Sindoori[1], S.Mahalakshmi[2], T.Sowkarthika[3], S.Imavathy[4], K.Pravinkumar[5]
[1,2,3,4]Assistant Professor,[5] Research Scholar,
Department of Computer Science and Engineering, E.G.S.Pillay Engineering College, Nagapattinam, Tamilnadu, India

Abstract— The major issue in WSN is vital information accessed by unauthorized party by clone node. Once a node is captured, the attacker can re-program it and replicate the node in a huge number, thereby easily take over the network process. The detection of node clone attacks in a wireless sensor network is a fundamental problem. Here, the proposed protocols are used to detect clone nodes. The protocols used here are Randomized, Efficient and Distributed (RED) protocol, Chord algorithm and distributed hash table (DHT). The first one RED, a new protocol for the detection of clone attacks. And it is used to generate the random number by group leader. The next protocol is chord algorithm for maintaining the neighbor's details. The distributed hash table (DHT) is a class of a decentralized distributed system that provides a service similar to a hash table and any participating node can efficiently retrieve the neighbor's details. The DHT is used to store the member ID, MAC address, preceded ID and successor ID. A witness node is used to verify the random number, member ID, MAC address. The witness node is able to detect the message is send from the authorized party or not by using random key.

Keywords: Nodes, WSN, Distributed hash table, Witness node

I. INTRODUCTION

Wireless sensor networks (WSNs) have a lot of attention for their huge applications in military and civilian operations, environmental monitoring, factory instrumentation, and climate control. WSNs are implemented in many environments like battlefield and homeland security situation. Hence, security process provides data integrity, confidentiality, non-repudiation and authentication.

Providing security is challenging in sensor networks because of inadequate resource of sensor nodes [3]. An unauthorized party can able to capture the information within nodes. The data in the nodes are key materials and codes. Once it is attacked, that attackers have capacity to replicate the seized sensors and activate them in network to produce insider attacks. This type of attack is known as *clone attack*.

Cloned node has taken part in network as same as original node and they make a variety of attacks. For example, a cloned node may create a black hole, and inject false data or aggregate data in such a way to prejudice the result. Sensors have fewer resources such as small memory size, low processing capability, and power supplies, unlike traditional network. Self-configured manner is major advantage of many sensors networks [8].

A sensor network has hundreds or thousands of small, less cost nodes spread in a large area. Their functions are not change even new one is introduced or old one is neglected due to power loss or any damage. Some networks have a centralized process for data gathering and

sometimes they process in distributed manner. Data can be collected from any nodes in network.

A wireless sensor network is a collection of nodes organized into a cooperative network. Node contains microcontrollers, DSP or CPUs chips for processing capability. It also has many memory types like flash memories, RF transceiver. Solar cells and batteries are used for power source, and actuators. In an ad hoc fashion, nodes are communicating and self-organized.

II. RELATED WORK

Eschenauer and Gligor [10] proposed centralized node revocation in sensor networks. *In Distributed hash table* (DHT) [2], data items are inserted and have a unique *key* for that data. With help of DHT, the ability of node can be determined for storing data with their key. Here every node maintains data about their IP address, their neighbors, and routing messages. In [3], proposed *location-based keys* (LBKs) based on a new cryptographic concept and node-to-node neighborhood authentication protocol based on LBKs. Then, pair wise sdhared keys between any two nodes and location-based threshold-endorsement scheme (LTE) attack.

LEAP (Localized Encryption and Authentication Protocol) is a key management protocol designed for supporting network processing [4]. In [5], key distribution problem is solved with passive adversary. It means node able to communicate with other nodes by generating a symmetric key. Then, sends it to their neighbors.

Mauro Conti [6] proposed protocol for detection of node replication attack called a new randomized, efficient, and distributed (RED) protocol. This has efficient memory, and high computation ability. Heesook Choi[8], discovered effective scheme termed as SET, for detecting clone attacks. It used set operations like union and intersection for detection process.

Zhu et al. [4] determined a key management protocol to remove the master key in sensor network. Parno et al. [1] proposed random and line-selected multicast process, where neighbor nodes of sensor choose random multiple witness nodes and send location, identifier of the node to them. Bo Zhu. et al [7] proposed Localized Multicast for detecting node replication attacks. In this, the witness nodes are randomly selected from the nodes.
Brooks *et al.* [9] proposed a clone detection protocol in the context of random key pre-distribution. In [11], a cryptographic scheme has pair of users to communicate securely.

In sensor network, communication cost is crucial performance metric due to energy is valuable resource for nodes. Not only that, transmission of message requires power than any of the other operations [13].

Many types of DHT process like CAN [14], which has low efficiency in both terms of scalable and communication cost. In real systems, this is rarely used. Chord [15] is used to form virtual ring, where every node is located at one point.

III. RED PROTOCOL

We propose a new Randomized, Efficient, and Distributed (RED) protocol for the detection of clone node attacks and we show that it is completely suitable with respect to the requirements. General simulations also show that our protocol is highly efficient in transmission of data, storage capacity, and processing speed, it has an improved attack detection probability compared to last techniques. And it is protective to the new attacks.

RED executes at fixed intervals of time. In the first step a random value is to be generated; this random value can be broadcasted with centralized and distributed mechanism to all nodes.In the second step, each node digitally signs and locally broadcasts its claim—ID and geographic location. RED does not send the claim to a specific node ID because this kind of a solution does not scale well: A claim sent to anode ID that is no more present in the network would be lost; nodes deployed after the first network deployment could not be used as witnesses without updating every nodes in network. However, RED has ability to adapt to work when a specific node is used as the message destination.

IV. DHT

A distributed hash table (DHT) is a class of a decentralized distributed0 system that provides a lookup service similar to a hash table. There, they has *key and value* pairs are stored. And every node present in the network can able to retrieve the value associated with a given key efficiently. Responsibility for reviewing the mapping from keys to values is distributed to all the nodes in the network, some change in the set of nodes may leads to minimum amount of problem.

DHT has capacity to note joining of new node, failure of node and disconnection in node. It able to scale large number of node in network. DHTs build an infrastructure by which more complex services can be handled. Services like distributed file systems, Web caching, domain name services, multicast, instant messaging, content distribution systems peer-to-peer file sharing.

A. Properties of DHT

1) *Autonomy and Decentralization:* the nodes collectively form the system without any central coordination.
2) *Fault tolerance:* the system should be reliable (in some sense) even with nodes participating, disconnecting, and failing.
3) *Scalability:* the system should function efficiently even with thousands or millions of nodes.

The basic requirements are that data be identified using unique numeric keys, and that nodes be willing to store keys for each other [2]. DHT has one important operation called key, which represents the IP address of the node. For implementing DHT, following issues to be concentrated:

B. Keys to be mapped to nodes

By using standard hash function nodes and keys are mapped in the format of string of digits. The digits are in binary format in CHORD. The digits are in high order base in CAN. Then given digit string is assigned to node with nearest digit string. It represents the node is the nearest to successor.

C. Forwarding a lookup for a key to an appropriate node

The node that gets key identifier can send it to node whose id is near to it. Thus nearest concept is suited to this situation. For knowing these details, each node has table which contains information about choosing the nearest node.

If key ID is greater than the current node, then send it to node which is greater than current node. If it is smaller than key, then it is numerically closer. Then it holds key ID is smaller than current node ID. Otherwise, send to node where ID has common key ID. Node has ID 8115 and a key has ID 8815, then forwarding to node 8365.

D. Structure of a DHT

The structure of a DHT can be decomposed into several important components. The core part is an abstract keyspace, which is 160 bit string as set. This ownership is partition among the nodes present in the network. It then connects the nodes in the network, and then they find their owner.

E. Constructing routing tables

Every node must know the other nodes for sending messages to other nodes. For getting closer to the key ID, every node need to know its *successor and it's ID.* By knowing this only, the node able to send message to successor node.

Then every node should know the matching identifiers. Maintaining the routing tables which contains the node join and leave information.

V. CHORD ALGORITHM

Chord is an algorithm and protocol for a peer-to-peer distributed hash table. A distributed hash table having key and value assigning it to different nodes. Function of Chord is assigning keys to nodes and discover value for the key. Chord algorithm is construction of the chord ring and localization of nodes [2].

By using Chord protocol, keys are arranged in a circle format has at most 2^m nodes. Range may be varying from 0 to 2^{m-1}.

Chord [14] assigns ID's to both keys and nodes from the same one-dimensional ID space. The node responsible for key k is called its *successor*, defined as the node whose ID most closely follows k. The ID space wraps around to form a circle, so ID 0 follows the highest ID.

Chord requires each node to keep a "finger table" containing up to *m* Chord requires each node to keep a finger table containing up to i^{th} entry of node *n* will contain the address of successor $(n+2^{i-1} \bmod 2^m)$.

Chord performs lookups in *O(logN)* time, where *N* is the number of nodes, using a per-node *finger table* of *logN* entries. A node's finger table contains the IP address of a node halfway around the ID space from it, a quarter-of-the-way, and so forth in powers of two.

A node forwards a query for key k to the node in its finger table with the highest ID less than k. The power-of-two structure of the finger table ensures that the node can always forward the query at least half of the remaining ID-space distance to k. As a result Chord lookups use *O(log N)* messages.

Chord ensures correct lookups despite node failures using a *successor list*: each node keeps track of the IP addresses of the next *r* nodes immediately after it in ID space.

This allows a query to make incremental progress in ID space even if many finger table entries turn out to point to crashed nodes. The only situation in which Chord cannot guarantee to find the current live successor to a key is if all *r* of a node's immediate successors fail simultaneously, before the node has a chance to correct its successor list. Since node ID's are assigned randomly, the nodes in a successor list are likely to be unrelated, and thus suffer independent failures.

Small values of r (such as logN) make the probability of simultaneous failure vanishingly small. A new node n finds its place in the Chord ring by asking any present node to look-up n's ID. All that is required for the new node to participate correctly in lookups is for it and its predecessor to update their successor lists.

Chord has to check correctness even though nodes with same IDs join in the network. New nodes and old nodes are have to be updated their tables. In the background, it is happening due to it is not used for performance. New node know the data about successor and relationship between them.

Randomness to make sure that Chord key and node IDs are distributed unevenly in ID space, balanced load along the nodes. Chord takes consideration over the key space present on node with help of *virtual nodes*. Every node participate in network with many ID of virtual node.

The Chord design is strength and correctness, even there is problems and failures. Chord is in use as a part of the experimental CFS [3] wide-area file store, and as part of the Twine [1] resource discovery system.

VI. SYSTEM ARCHITECTURE

Fig. 1 represents the architecture diagram. Network Construction is developed in order to create a dynamic network. In a network, nodes are interconnected with the admin, which is monitoring all the other nodes. All nodes are sharing their information with each others.

Chord Algorithm can verify the Neighbor nodes information of the Requested Node. So that by verifying the Id's and location we can detect the Clone Node. For this purpose we have to create the List of the Neighbor Nodes information for each node so that the Server/ Witness Node can verify the nodes request.

Witness Node Distribution 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: ID-based prediction and Location-based prediction.

Verification of Random Number is 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.

Verification of User ID in 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 User Information. If both the data are matched then the Witness node will confirm that this node with that Location is Genuine.

Clone Detection and Data Transfer is 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.

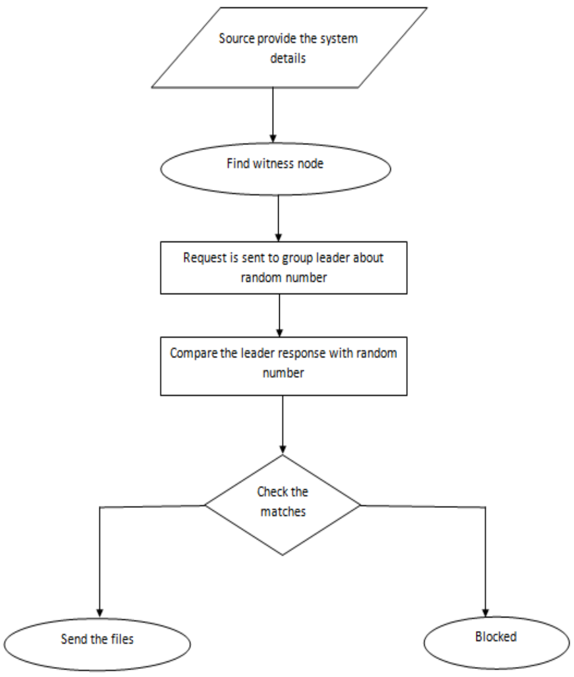
Source provide the system details
Find witness node
Request is sent to group leader about random number
Compare the leader response with random number
Check the matches
Send the files
Blocked

Fig. 1 System Architecture

VII. CONCLUSION

In this paper, three detection protocols are used: One is based on a RED, other is distributed hash table, and third one is chord. The DHT protocol gives high security for all types of sensor networks. We have implemented the CHORD algorithm and RED protocol to identify clone attack in wireless sensor networks. By implementing this technique we are able to identify the attacks more efficiently than the existing approaches.

Also we are encrypting the data packet during transmission will also increase the security level. Since we are implementing the level-wise security for retrieving the data from other node in the network will be more useful in military applications.

VIII. FUTURE ENHANCEMENT

In future Enhancement for DHT-based protocol we 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 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.

[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.

[15] I. Stoica, R.Morris, D. Liben-Nowell, D. R. Karger,M. F.Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[16] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Int. Conf. Distrib. Syst. Platforms Heidelberg*, 2001, pp. 329–350.

[17] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *Proc. 1st Int. Conf. Simulation Tools Tech. Commun., Netw. Syst. Workshops*, Marseille, France, 2008, pp. 1–10.

[18] A. Awad, C. Sommer, R. German, and F. Dressler, "Virtual cord protocol (VCP): A flexible DHT-like routing service for sensor networks," in *Proc. 5th IEEE MASS*, 2008, pp. 133–142. [19] R. Diestel, *Graph Theory*, 3rd ed. New York: Springer, 2006.