International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 05 | May 2019 www.irjet.net p-ISSN: 2395-0072

A Novel Mechanism for Clone Attack Detection in Hybrid IoT Devices

Meghana S[1], Rampur Srinath[2]

[1]Dept. of Information Science & Engineering, National Institute of Engineering, Mysuru, Karnataka, India
[2]Associate Professor, Dept. of Information Science & Engineering, National Institute of Engineering, Mysuru, Karnataka, India

---

Abstract - *Node replication attack is a very serious type of attack using which an attacker can affect the operations of the network by inserting a replica or clone in the network. Internet of Things has become a victim of this attack since it is very easy for an attacker to collect the information and authentication credentials from a week node in the network. In this paper, a new clone detection method has been proposed keeping Multidimensional scaling (MDS) as the base for clone detection. The proposed technique is apt for IOT network, because (i) Geographical locations of the nodes is not required to detect the replicas, (ii) this method can be used in hybrid IOT networks that includes both static and mobile nodes and (iii) the core part of the detection rule can be parallelized, which leads to speed-up the entire detection process. Taking all these factors into consideration, we propose this clone detection method as assuring method for a practical node replication detection design in IOT.*

Key Words: Node Replication Attack, Internet of Things, Hybrid IOT, Multidimensional Scaling (MDS), Localization via MDS

1.INTRODUCTION

Internet of Things (IoT) is an emerging networking paradigm in which a large number of interconnected devices communicate with each other to facilitate people-object communication. A smart city, for example, consists of several smart sectors, such as smart homes, smart hospitals, and smart cars, which are important IoT applications. Each IoT gadget is equipped with built-in sensors and wireless communication capabilities in a smart home scenario. The sensors can gather information about the environment and communicate with each other, as well as the owner of the house and a central monitoring system. Patients wear implantable sensors that collect body signals and send the data to a local or remote database for further analysis in a smart hospital scenario that could be implemented using body sensor networks (BSN). As another example, sensors embedded in cars can detect accident events or traffic information and exchange such information collaboratively in a smart traffic scenario.

IoT devices are vulnerable to several security threats due to their restricted features and capabilities. IoT devices could easily be captured, for example, resulting in a node replication attack (also known as a clone attack). The captured device is reprogrammed, cloned, and returned to the network in such a scenario. In addition, devices that are supposed to be trusted can cause clone attacks in special cases (e.g. misconfiguration or production by untrusted manufacturers with adversarial intentions). A clone attack is extremely harmful because it will be considered as legitimate devices for clones with legitimate credentials. Such clones can therefore easily perform various malicious activities in the network, such as launching an insider attack (e.g. blackhole attack) and injecting false data leading to IoT scenario hazards.

1.1 Problem Statement

While there is quite extensive literature on approaches to clone attack detection in WSNs, when it comes to IoT scenarios, this remains an open problem. Two unique characteristics of the IoT environment compared to conventional WSNs make the establishment of clone detection schemes in IoT a more challenging issue. First, the devices lack accurate geographical position information. For example, devices embedded in smart cars are likely to derive location information through the car navigation system, i.e. geographic positioning system (GPS), whereas devices in a smart home or BSN are unlikely to have GPS capability embedded due to their high energy consumption and additional hardware requirements.

Secondly, IoT networks are hybrid networks made up of static and mobile devices with no a priori mobility pattern (they can be static or moving at high or low speeds), e.g. a patient carrying wearable sensors and living in a smart home. Wearable devices could be considered as mobile nodes because the patient can move around while most devices are still in a smart home. Indeed, IoT nodes can be relocated without a priori mobility pattern (they can be static, high-speed moving, or slow moving). While some of the existing clone detection methods for mobile networks could be applied to hybrid networks (consisting of both stationary and mobile devices), they suffer from a degradation of the probability of detection. We explain how we address these challenges and advance the state-of - the-art solutions to detect clone attacks in section 3.

2. RELATED WORKS

In recent years, there has been a surge of interest in providing WSN-specific security solutions, among which clone attack detection has attracted considerable attention due to the growing interest in adopting WSNs in several applications. In this section, we review the methods of clone detection that are closely related to our work and clarify the

difference between our proposal and the related work that is currently underway.

Researchers [1] have proposed several classifications for clone detection approaches based on the information required (i.e., location-based or location-independent), methods of detection (i.e., centralized, distributed or partially distributed) and network type support (i.e., mobile or static networks). Our proposed approach to MDSClone falls within the category of centralized location-independent methods that support hybrid (static and mobile) networks.

Findings in [2] show that only when the network is partitioned into cells will SDC and P-MPC be effective. The protocol proposed in [2], i.e. the randomized, efficient and distributed (RED) protocol, provides an almost perfect guarantee of clone detection compared to the other approaches. RED uses a special centralized broadcasting device, such as a satellite and UAV, to broadcast the node IDs responsible for detecting conflicting location claims on a regular basis.

[3] proposed alternative clone detection approaches, such as social fingerprints. A key issue in the security of the sensor network is that sensors are susceptible to physical capture attacks. The adversary can easily launch clone attacks once a sensor is compromised by replicating the compromised node, distributing the clones across the network, and starting a variety of insider attacks. Previous work against clone attacks has either a high overhead of communication / storage or poor accuracy of detection.

[4] proposed an alternative approach, such as pre-distributed keys, to detect clones. Because of their low overhead, random key pre-distribution safety schemes are well suited for use in sensor networks. However, cloning attacks can compromise a network's security using pre-distributed keys. An opponent breaks into a sensor node in this attack, reprogram it, and inserts several node copies back into the sensor network. Researches in

[5] suggested an alternative approach, such as random clustering, to detect clones. Sensor nodes are vulnerable to capture and compromise when deployed in hostile environments. An opponent may obtain, clone, and intelligently deploy private information from these sensors in the network to launch a variety of insider attacks. This process of attack is widely referred to as a clone attack.

XED is presented in [6] for mobile sensor networks as a simple challenge-and-response strategy, presenting the first distributed clone detection method for mobile networks. It is vulnerable to cloned node collusions, however. This paper addresses the challenge of detecting node replication.

In summary, the existing methods of clone detection designed for static networks cannot be applied to scenarios where node mobility would destroy neighborhood and node distance relationships. On the other hand, as mentioned above, the adoption to hybrid networks of most mobile clone detection methods results in a degradation of the probability of clone detection. Therefore, to deal with clones in IoT environments, we need to provide a method that is "particularly" designed for hybrid networks and does not rely on any mobility pattern assumptions, if any. Furthermore, prior solutions are largely based on the assumption that each node is aware of their geographic position. This is not the case with IoT devices, though. Consequently, the existing methods of clone detection do not apply to IoT environments.

3. SYSTEM MODEL

3.1 Proposed System

We propose MDSClone in this paper, a new mechanism for clone detection for IoT environments. MDSClone specifically by adopting a multidimensional scaling (MDS) algorithm circumvents the two major issues mentioned above that emerge in IoT scenarios. The following are our main contributions:

- We are proposing a method of clone detection that does not rely on node geography. Instead, we generate the network map based on the relative neighbor-distance node information by adopting the MDS algorithm. While most state-of - the-art methods of clone detection assume that each node is always conscious of its geographical position, this assumption does not apply to all IoT devices. Therefore, we are significantly advancing the existing clone detection solutions for IoT by removing such an assumption in MDSClone.
- Without considering any specific mobility pattern, our proposed MDSClone method is capable of detecting clones in the network based on topology distortion. This is an important feature of MDSClone, as IoT nodes do not follow a specific mobility pattern as explained earlier, and existing clone detection methods for mobile networks do not have reasonable performance in hybrid networks (see Section II for more details). MDSClone method is applicable to all pure static, pure mobile, and hybrid networks compared to the related work, and MDSClone's detection probability remains the same for all these network topologies.
- We demonstrate that MDSClone is efficient in terms of overhead computing, because the main computing is performed by the base station(BS), and server-side computing can be easily paralleled to significantly improve performance. Compared to the state-of - the-art, this is an outstanding feature of MDSClone, as the parallelization capability of existing clone detection methods remains unclear.

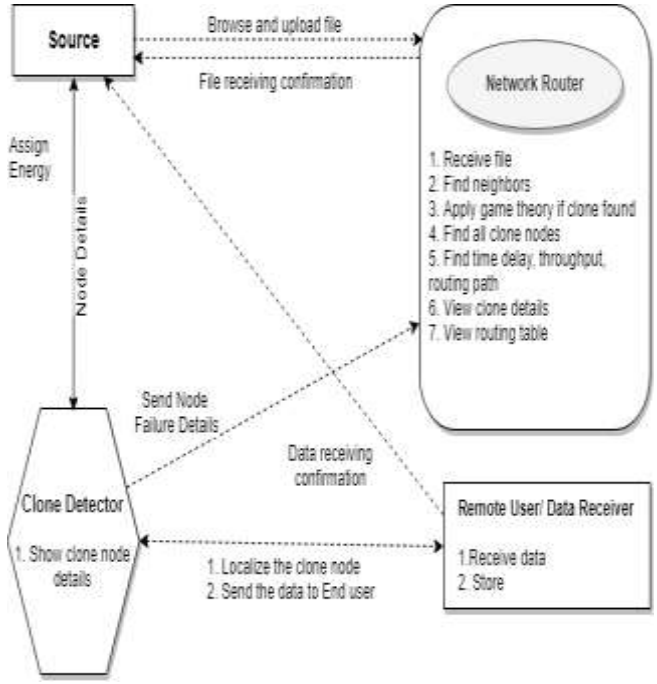

Fig -1: Proposed System Architecture

3.2 Network Model

We regard an IoT network as a hybrid network consisting of two main entities: 1) n static and mobile nodes with unique IDs: a base station (BS) ID {1,...,n}; and 2). Each IoT device measures its distance with its neighboring nodes periodically and sends the information to the BS. The BS is responsible for executing our proposed MDSClone algorithm in our system model and locating the "clones in the network. In particular, for each node in the network, the BS periodically receives neighboring information and builds a location map (based only on the information received from the nodes) to detect clones. The BS executes offline MDSClone, and at time t each location map generated is dedicated to a network snapshot. The main idea in our proposed method is that a node x cannot have two different neighborhood sets at the time t, which means that x cannot be at the time t at two different network locations. The following assumptions are made in our network model:

- We assume nodes are unaware of their exact geographical position "necessarily." This assumption is based on the following two factors i) using GPS is costly in terms of energy and extra hardware requirements, and ii) researchers believe that indoor scenarios GPS-based positioning is not efficient. We therefore assume that some nodes (e.g. smartphones) may be enabled for GPS, and others (e.g. home appliances) may not. Our proposed method therefore does not rely on node geographic positions. We make this assumption to consider the first challenge we mentioned in the "Problem Statement" section, i.e., lack of the device's accurate geographic position information.
- Mobile nodes are assumed to move without any specific mobility pattern. This assumption makes our network model more realistic because, node mobility patterns (e.g. wearable sensors) are unpredictable in IoT scenarios. We make this assumption to consider the second challenge we mentioned in the "Problem Statement" section, i.e. IoT networks are hybrid networks made up of static as well as mobile devices without a priori mobility pattern.
- We also assume that IoT devices can enact device-to-device short-range communication. Each node can therefore measure its distance from its neighboring nodes by means of radio signal strength (RSS) or arrival time (ToA). Although the estimated distances are not perfectly accurate, for our approach they are sufficient. We make this assumption, as each IoT device should regularly measure its distance with its neighboring nodes and send them to the BS in our proposed approach.
- We assume that the BS is familiar with the geographical position of IoT devices at the very beginning (only during network initialization). However, the BS is no longer aware of the device's positions after the network deployment. We make this assumption because the network designer generally performs the setup and deployment of IoT devices in the network, and it is therefore reasonable to take such an assumption. This assumption helps the BS detect and locate the clone nodes by comparing the built location map with the node-received information and the original network map.
- We also assume that loose time synchronization exists between nodes1, and that network operating time is divided into time intervals, each of which has the same length. We make this assumption because at time t each location map generated is dedicated to a network snapshot.
- We assume that, unless otherwise stated, the exchanged messages are digitally signed before being sent. We make this assumption to ensure that the exchanged neighboring information is confidential and accurate based on which the location map is generated.

3.3 Attack Model

IoT devices are generally not considered to be resistant to manipulation. In other words, in the event of a device being compromised, all stored security credentials can be extracted. In addition, the opponent may compromise a device immediately after the deployment of the node. There is no secure time to bootstrap. Thus, the opponent can access all the compromised device's legitimate credentials. In this paper, we consider an adversary capable of performing "clone attack," which means they can manufacture compromised devices and store the legitimate credentials of the compromised devices within several manufactured devices. A compromised node is called clones, as are the manufactured nodes with the same identity and credentials

as the compromised node. Clones can communicate with each other and collaborate, trying to stealthily subvert the detection functionality. It should be noted that we only consider cloning attacks and assume that there is no simultaneous "node compromise" attack, which means that no other nodes (beyond clones) act in a malicious way.

We're dealing with clone attacks in particular and not one-node compromise attacks. Essentially, a "clone attack" can be considered as a special type of node compromise attack in which at the same time there are two or more compromised nodes in the network with the same ID. Clone nodes are, in other words, exact copies of the original compromised node. Although compromising a single node is the first step of conducting a clone attack, we only consider the aftermath of compromise and cloning. Note that attacking a node compromise is different from attacking a clone. The former usually refers to a case where the attacker compromises a specific node and then places that compromised node back into the network, while the latter refers to a case where the attacker compromises a specific node and places multiple replicated copies (clones) of the compromised node back into the network. Detection solutions for clone attacks are also different and independent from detecting a single node compromise. This is because clone detection methods are usually based on clone node relationships with the same ID with their neighboring nodes, or their network placement, and these methods are not capable of detecting "one single" node compromise attack.

We consider a simplified clone model rather than a generic clone model consisting of s clone groups, each of which contains at most z clones. There is only one clone group in our model, with exactly two clones having the same identification. Unless otherwise stated, the clone ID refers to the identification of two clones in a specific clone group. Using such a simplified model is to ease the presentation of our main idea while, of course, our method can be applied to a generic clone model without compromising the safety.

4. IMPLEMENTATION

Multidimensional scaling (MDS) is a way to visualize the similarity level of a dataset's individual cases. It refers to a set of related ordering techniques used in the visualization of information, especially to display the information contained in a distance matrix. It is a form of reduction of non-linear dimensionality. An MDS algorithm is intended to place each object in N-dimensional space so as to preserve as well as possible the distances between the objects. In each of the N dimensions, each object is then assigned coordinates. An MDS plot N's number of dimensions can exceed 2 and is a priori specified. Choosing N=2 optimizes a two-dimensional scatter plot for object locations.

Suppose there are n randomly distributed nodes for which distance is known between each pair of sensor nodes, then estimate position of unknown node Multidimensional Scaling Law of Cosines and linear algebra which helps to reconstruct relative node positions on the basis of pair wise distances. The technique can be performed using an algorithm with the following four steps:

1. Collect data from the network and create a distance matrix X, where xij is the range between i and j nodes.
2. To develop a complete matrix of internode distances R, execute an algorithm to determine the shortest path for example Dijkstra, Floyd etc on X.
3. To find estimated node positions P, run a classical metric MDS on X,
4. Transforming the metric P solution into global coordinates.

Following are the modules present in the system.

- Source

In this module, the Sender will browse the file, Initialize the nodes, distribute Mac address for every node and then upload to the particular Receiver (receiver1, receiver2, receiver3 and receiver4). And router will connect to the particular receiver. After receiving successfully, it will give response to the sender. The Sender can have capable of manipulating the data file.

- Router

The Router manages a multiple node (node A, node B, node C, node D, node E, node F….) to provide data storage service. In a router we can view the node details, assign cost and view clones. The sender will upload data file to the router, the Router will select the smallest distance path and send to the particular receiver. If any clone is found in a particular node, the route replay will send to the Trusted Authority and then it will select another path. In a router service provider can view the node information details and view the routing table details.

- Trusted Authority

In this module, the Trusted Authority is responsible for identify the intrusion in the network. If the router found any type of clones, then it transfers the flow to Trusted Authority. Then the Trusted Authority is responsible for capturing the clones and identifies which type of clone (fake key clone, Destination IP clone and cost clone) and then response will send to the router. After getting a response from the TA, router will select another path and send to the particular receiver (receiver1, receiver2, receiver3 and receiver4). The Trusted Authority will make a list of failed node details and then all failed nodes are stored with tags such as node name, IP address, MAC address, node cost, time and date.

- Receiver

In this module, there are an n-numbers of receivers are present (receiver1, receiver2, receiver3 and receiver4). All the receivers can receive the data file from the sender via router. The sender will send data file to router and router will select the lesser distance path and send to the particular receiver (receiver1, receiver2, receiver3 and receiver4),

without changing any file contents. The receivers may try to receive data files within the router or network only.

- Clone

In this module, the clone can attack the node in three ways fake node clone, Destination IP clone and cost clone. Fake key clone means he will inject fake key to the particular node; IP clone means he will change the destination IP address to the particular node, cost clone means he will inject fake cost to the particular node.

5. EXPECTED RESULTS

While most state-of-the-art methods of clone detection assume that each node is always conscious of its geographical position, this assumption does not apply to all IoT devices. Therefore, we are significantly advancing the existing clone detection solutions for IoT by removing such an assumption in MDSClone. MDSClone method is applicable to all pure static, pure mobile, and hybrid networks compared to the related work, and MDS Clone's detection probability remains the same for all these network topologies. We show that MDSClone is efficient in terms of computational overhead, because the base station (BS) performs the main computation, and the computation on the server side can be easily parallel to significantly improve performance. Compared to the state-of- the-art, this is an outstanding feature of MDSClone, as the parallelization capability of existing clone detection methods remains unclear.

6. CONCLUSION

In this paper, we proposed a clone detection solution for a heterogeneous IoT environment, called MDSClone, based on the multidimensional scaling algorithm (MDS). In designing MDSClone, we took into account the specific features of IoT devices, i.e., lack of awareness of geographic positions, the possibility of being both static and mobile, and the lack of a specific mobility pattern. We have shown that MDSClone provides an outstanding approach compared to the existing clone detection methods, as it is the first method to support hybrid networks, while its memory cost is of order O(1), its communication cost is affordable, and it is a location-independent method. In addition, we showed that MDSClone's clone detection probability is nearly 100 %, and the MDS calculation algorithm could be parallelized, resulting in a shorter detection delay. Therefore, considering all its advantages, we believe that in real-world IoT scenarios, MDSClone could be considered as a top candidate for clone detection. However, our proposal may impose a communication overhead on the network in the case of dense network topologies. Hence, we aim to provide a distributed version of MDSClone for IoT scenarios in future work.

REFERENCES

[1] A. K. Mishra and A. K. Turuk, "A comparative analysis of node replica detection schemes in wireless sensor networks," Journal of Network and Computer Applications, vol. 61, pp. 21–32, 2016.

[2] M. Conti, R. D. Pietro, L. V. 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, 2011.

[3] K. Xing, F. Liu, X. Cheng, and D. H. Du, "Real-time detection of clone attacks in wireless sensor networks," in ICDCS'08. IEEE, 2008, pp. 3–10.

[4] R. Brooks, P. Govindaraju, M. Pirretti, N. Vijaykrishnan, and M. T. Kandemir, "On the detection of clones in sensor networks using random key pre-distribution," IEEE Transactions on SMC, Part C, vol. 37, no. 6, pp. 1246–1258, 2007.

[5] H. Choi, S. Zhu, and T. F. La Porta, "Set: Detecting node clones in sensor networks," in SecureComm'07. IEEE, 2007, pp. 341–350.

[6] C.-M.Yu, Y.-T.Tsou, C.-S.Lu, and S.-Y.Kuo, "Localized algorithms for detection of node replication attacks in mobile sensor networks," IEEE Transactions on Information Forensics and Security,, vol. 8, no. 5, pp. 754–768,2013.