A Honeypot-assisted Industrial Control System to Detect Replication Attacks on Wireless Sensor Networks

Abbasgholi Pashaei[1], Mohammad Esmaeil Akbari[2*], Mina Zolfy Lighvan[3], Asghar Charmin[4]
1,2,4- Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran.
Email: a-pashaei@iau-ahar.ac.ir, m-akbari@iau-ahar.ac.ir (Corresponding author), a_charmin@sut.ac.ir.
3- Department of Electrical and Computer Engineering Faculty, Tabriz University, Tabriz, Iran.
Email: mzolfy@tabrizu.ac.ir

Received: May 2022 Revised: June 2022 Accepted: July 2022

ABSTRACT:
Industrial Control Systems (ICSs), which work based on Wireless Sensor Networks (WSNs), are prone to hacking and attacks. In node simulation attacks against ICS networks, the enemy may capture a sensor node and then make multiple copies with the same identifier (ID), code, and encryption of the recorded node. Unfortunately, many Intrusion Detection Systems (IDSs) are not efficient to detect clone attacks in ICSs. An alternative solution to improve the performance of early detection is a honeypot. This paper proposes a centralized architecture for detecting copy or clone nodes using a local multicast intrusion detection system. We divide the WSN into sections and give each one an inspector node. Each inspector node monitors its region and uses the node ID to identify clone nodes. We offer solutions for situations where the cluster-head is endangered. We also provide solutions for other cases where the natural node is compromised. Our evaluations show that the proposed system maximizes the detection probability and, at the same time, has a low connection overhead.

KEYWORDS: Intrusion Detection System, Honeypot, Industrial Control Systems (ICS), WSN Replication Attacks.

1. INTRODUCTION

One of Wireless Sensor Networks (WSNs) is industrial facilities. With the expansion of the application of IEEE 802.1x networks, the transmitting of large volumes of big data in Industrial Control Systems (ICSs) is possible in just a few seconds. One of ICS's challenges is the infiltration of unauthorized parties for intrusion. These intrusions cause a lot of damage to the equipment annually. WSN nodes are constantly exposed to frequent node (or clone) attacks [1]. Clones are characterized by acting as legitimate nodes or authorized participants in the network [2]. Therefore, the need to design Intrusion Detection Systems (IDSs) is very high.

In an attack, the enemy physically captures one or more sensor nodes and creates similar instances of the nodes it takes. The enemy can then design many internal attacks and destructive activities using these replicas or clones. For example, when several clones are combined, the enemy can create a black hole and launch a wormhole attack. It can also establish a Distributed Denial of service (DDoS) attack, inject incorrect data into the network, and monitor most network traffic. Another potential attack is to slow down and eliminate other WSN nodes.

In this research, we aim to counter the intrusion of repetitive node attacks. We present a centralized scheme for identifying copies and clone nodes in WSNs [3]. First, the network is divided into sections [2]. Then, the inspector nodes are randomly selected for each cell in a large and unpredictably recorded deployment by the enemy [1]. The inspector node examines similar nodes from the relevant sections for the cryptographic identifier and key. We also offer the honeypot idea to improve replica recognition performance. Honeypot is a hardware environment with an active telemetry program to attract and deceive attackers.

Traditional IDSs are practically inefficient for early detection due to the high false alarms. Therefore, this article introduces WSN honeypot technology to detect early intrusion of attacks on ICSs with wireless radio equipment. Honeypot detects possible attacks on the WSN in the 800 MHz and 1.2 GHz frequency bands to complement the primary data transmission equipment. All attackers' interactions will be reported to the ICS monitoring center when communicating with the WSN

Paper type: Research paper
DOI: 10.30486/mjtd.2022.695925
How to cite this paper: A. Pashaei, M. E. Akbari, M. Zolfy Lighvan, and A. Charmin4, "**A Honeypot-assisted Industrial Control System to Detect Replication Attacks on Wireless Sensor Networks**", Majlesi Journal of Telecommunication Devices, Vol. 11, No. 3, pp. 155-160, 2022.

honeypot. At the same time, these interactions are stored in an EIDS database designed for this purpose. We also provide solutions for blacklist replica nodes of WSNs. This solution improves network performance in two ways: a) when a Cluster Head (CH) is detected as a replica and is deleted or blocked; b) when a normal node is detected as a replica and is deleted or blocked.

The remainder of this paper is organized as follows: Section 2 reviews the most critical studies; Section 3 explains the system model and the proposed method; Section 4 evaluates the results; Finally Section 5 concludes the study.

2. RESEARCH BACKGROUND

Regarding clone attacks, the LSM method is proposed in [2]. It uses the network routing topology to select the node location. The goal is to reduce communication costs and increase the likelihood of detection. It uses geometric probability distribution to identify duplicate nodes. In [4], a centralized system called the speed test scheme is proposed. Another study [3] suggested a method based on place and time vector information. Sometimes, a node has to announce its time position and its list of neighbors. This design requires the coordination of precise time and exact location dynamically. Generally, there are two ways to get a dynamic position at all times: a) using an unreliable ideal signal; b) providing a costly GPS [3]. Kou et al. [5] proposed the NBDS method. It stops at the new location and successfully reconnects to the network. The node distributes the counter-claim to its new neighbors with a non-neighbor signature to join the new site. A node is only accepted as a legal node by new neighbors if the previous neighbor's approval is successful.

A review of the literature shows that researchers have proposed other innovative methods. For example, in some schemes [6] [7], each node locally identifies its location and notifies adjacent nodes. Each neighboring node that receives a claim sends the claim to a set of randomly selected "control" pseudo-networks. Each control node then checks to see if it has already received a claim with the same ID from the node. If not, the control node stores the received position claim. Otherwise, it checks whether they were sent from there or not. If the answer is no, the control node starts the empty process. If the control node finds the duplicate identifier, it realizes that the node is a replica.

Some research has used a more heuristic design [8] [1]. Nodes must confirm the signature after receiving a response at the central station while the node is repeatedly attacked. If authorized, the node’s claim will be broadcast on the network. If the timer expires and the new node’s neighbors do not receive any cancellation messages, copies of the node recovery from the new neighbors are sent to the node's previous neighbors. The Base Station (BS) accepts the new node and adds its ID [1]. The drawback of this scheme is that an intelligent attacker may block the clone detection process. This method also has a high communication overhead. In [8], a note-based scheme is proposed. In the replica identification step, each node (claimant node) first sends the signed location claim to neighboring nodes. After receiving the claim, each neighboring node confirms the signature and acceptability of the location of the claimant node. Adjacent nodes are, by some probabilities, transformed into inspector nodes. The control nodes then perform a two-step operation. In the first step, called area selection, reporters from each specific area use the area selection method to claim the location of a claimant node. Each randomly selected node that receives the location claim in each region, after confirming the signature, becomes the control node of the claimant node. When the control node reaches a contradiction, it receives two different location claims with the same node ID. It cancels the replicas by spreading two contradictory claims as evidence [2]. This method has additional overhead for the inspector. The RED protocol is described in [9]. The main idea of the protocol is that each sensor node first signs and maintains a list of nearby nodes [6]. Each control node sends encrypted information and then transmits it to its nearest neighbor. When the BS receives all the information and senses it, it considers it a repeated node if it finds the sensor value higher than the specified threshold value. Otherwise, it records the node information [6]. An area-based method is proposed in [10]. The WSN nodes are centrally controlled [8]. The network is then divided into smaller sections, and an inspector node is assigned to each area [2]. The whole sub-area has an equivalent angle from the primary node. All nodes transmit the position of their neighbors to the BS by the control node. Duplicate nodes are then canceled based on a cross-comparison of the received data. The major drawback is that such a centralized design contradicts the emerging nature of WSNs and creates a point of failure [8].

A hierarchical clone node identification protocol is presented in [1]. It is based on a clustering method using a bloom filter to detect replicas. Each CH shares its encrypted material (node IDs, signatures, messages) with other CHs through the bloom filter mechanism [6]. A CH can detect the presence of an enemy node [3]. This design's major drawback is having additional overheads [11]. In addition, it requires a large number of bits for the bloom filter.

Interested readers can refer to [12] to study ways to improve IDS performance with early detection and honeypot.

3. PROPOSED METHOD AND SYSTEM MODEL

This section describes the system architecture and then the honeypot detection technique.

3.1. The architecture of the detection system

The architecture of the honeypot diagnostic system consists of the following modules:

3.1.1. Path module

The route module consists of packet analysis for path responses, a fake packet generator, and a fixed bit-rate Cooperative Balancing Routing (CBR) unit. In this suggested approach, the honeypot agent is positioned next to the test subject and directs the RREQ to a designated destination. When a test subject gets an RREQ, it generates an RREP packet and determines whether the RREP is genuine.

The module for analyzing false RREP packets examines the received response packet. This module examines the RREP packet and logs the packet's sequence number and number of steps. It then produces the bogus packets and delivers them to the test subject. Such traffic is routed via the "subject" to a specified destination. In this scenario, the false packet maker employs the CBR unit, which creates UDP packets at a predetermined bit rate.

3.1.2. Feedback module

The feedback module is critical in identifying malicious nodes. An inquiry packet is sent to a particular destination to ascertain if it has received traffic packets from the test subject. This information is maintained in the feedback module to offer information about the duplicate route. If the target node accepts the packet, it sends a tracking response to the honeypot. Based on this answer, the module deems the test subject valid if it is not a hostile attacker.

3.1.3. Alert module

If the module identifies harmful behavior, it will warn the user and use this information as input to the module. In these instances, a positive output indicates normal operating circumstances, while a negative output indicates the existence of an assault. When an attack is discovered, the alert module generates a warning to inform users to avoid engaging in hostile node behavior. The malware identity module broadcasts a malicious node to all other nodes on the network, instructing them to suspend communication with it.

3.1.4. Interactive report

It describes the tactics used by the honeypot to deceive the destructive node. In addition, it records information about the path responses that an attacker uses to deceive other nodes. This step records all the reports and alerts of the route discovery step.

3.2. Honeypot agent

This unit shows malware node detection using the honeypot, introduced as a software detection agent. The assault of the malicious node *M* is shown in Fig. 1. The figure shows that node M gets all data communication from neighboring nodes to reduce data traffic. By raising the number of sequence numbers and minimizing the number of steps, node *M* establishes itself as the optimal route for other nodes (AODV). Additionally, it may promote the route's lowest end-to-end latency.

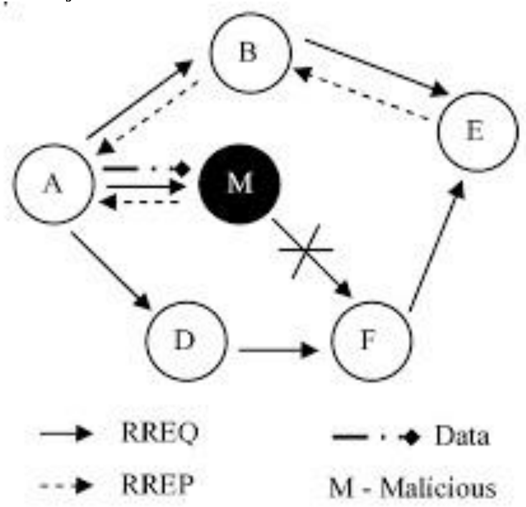


Fig. 1.M is a destructive node.

3.3. Honeypot is unaware of the network's topology.

When the honeypot is unaware of the network architecture, it is tested next to the node and estimates the delay by sending traffic to the test subject. Any difference between the actual delay and the estimated end-to-end delay is examined. When the test subject certifies that the end-to-end latency is small, the honeypot connects to the next test subject and starts sending traffic. We install honeypots on nodes to identify malicious attackers. The suggested approach is divided into the following steps:

- The honeypot agent communicates with the test subject through an RREQ packet. The source address is the node on which Honeypot is installed. Destination Address is a previously determined random destination address. Honeypot is presumed to be aware of the destination's path and sends a special RREQ to verify the nodes in its neighborhood.
- The "test subject" communicates with the honeypot through an RREP packet. This RREP may be genuine or fraudulent. A malicious "test" consists of a bogus RREP with a large sequence number and a few steps. On the other hand, a legitimate test subject creates an RREP only if it is aware of the path to the target.

- The next step is to generate a data packet for testing and transmit it to the "patient." The tester package is identical to any other data package. However, since its contents are obscured and filled with a random data stream, the victim may not deduce that it came from the honeypot.
- Honeypot sends the destination a "Query" packet inquiring about the packet of "test subject" in step 3. The packet's format is shown in Fig. 2. The feedback module retrieves information using a duplicate (periodic) path table and then routes the query packet via this path. The query package has many fields, including the sequence number, the source IP address, the destination IP address, and the test id.

Test ID	Destination address	Source address	Sequence number

Fig. 2. The packet's format.

- The source IP address is encrypted with the node address of the honeypot, and the destination IP address is chosen as the destination address. Additionally, it has a test identification field, which provides the IP address of the evaluation point's test source.
- When the destination gets the query package, it adds it to the cache. This memory contains information on the most recent traffic received from multiple sources, including source IDs, the time the traffic was received, and the number of packets received from the source.
- If the destination discovers the test subject's ID in its traffic cache, it creates a "query response packet" and changes the destination address to match the source address of the honeypot. The "Answer to Question" packet contains information such as the total number of packets received and the time of the most recent one. As a result, the query answer packet is delivered to the honeypot in the same manner as the tracking packet. Fig. 3 illustrates the fields of the "Search Answer" packet.

Closed counting	Destination address	Time label	Source address	Sequence number

Fig. 3. The fields of the "Search Answer" packet.

- After receiving the request packet, the honeypot agent forwards it to the feedback module. The content of the information field establishes the identification of the test node. If the packet is successfully sent to the target, the "test" is regarded as a "good node," but if the field is empty, the "test" is considered a "malicious node."
- Then, the honeypot's alert module designates the "test subject" to be a hostile attacker. As a result, other network nodes refrain from transmitting harmful node packets.

3.4. Network pattern

The geographical location of nodes is not required here. A secure communication channel is used so that the sensor nodes can communicate with BS or CH [3]. As mentioned, the monitoring scheme is centralized because all nodes transfer the location of their neighboring nodes to the central node (station). Duplicate nodes are then canceled or blocked based on the reciprocal comparison of the received data [8].

3.5. Enemy model

An enemy must first be able to capture one or more nodes from the network. It can then simulate so-called clone IDs, encryption keys, code, data, and other information through captured nodes. Then, as mentioned, it creates one or more clones of these nodes with the same value of ID and places these clones in different network positions [13]. An attacker can compromise both a natural node and a CH node.

3.6. Node-based inspection

We first divide the sensor networks into smaller sections [2]. The basic idea is that a segment is composed of one or more levels of network nodes in a hierarchical manner [7]. Each inspector node distributes a request to all nodes in the appropriate sections for their ID and encryption key. The inspector node listens to the patrol request for the first time and only stores the information. For other nodes and whenever the patrol receives the ID and key, it must compare the declaration with the previous records before saving it. In case of non-compliance, it will record the ID and key and continue working. Otherwise, it sends an alert to BS requesting the cancellation of the clones [7]. A centralized list is maintained in the BS that includes all the identifiers. If a new node is added, the corresponding CH sends its identifier to the BS to determine if it is legal [3]. This requires all nodes and control nodes to send their list of neighbors and their claimed location to the BS. The BS monitors the deployment and then reports the presence of duplicate nodes to all nodes [3].

The proposed design algorithm is as follows. The control node checks the ID with the location information in the list for that ID. If a similarity is found, the control node immediately sends an alert to BS, blocking the network ID information, thus, asking other nodes not to welcome them in the future. This means that communication through that ID is blocked on the network. BS then receives these reports and removes the blacklist or replica nodes from the WSN.

3.7. Honeypot identification plan

As is shown in Fig. 4, we use honeypot to identify replica nodes more efficiently. The attacker is easily trapped because the honeypot is indistinguishable from other nodes. A honeypot deceives an attacker every time in a new way without having a history of WSN nodes. Honeypot checks some information, such as the attacker ID. It considers a node a clone if the ID is unknown. Honeypot investigates suspicious activity on the network and notifies BS [14].

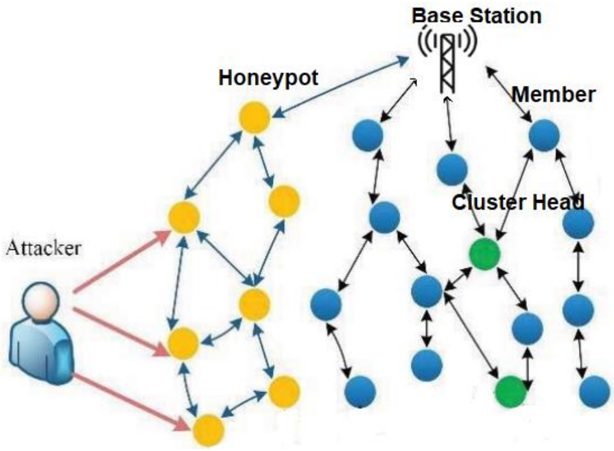


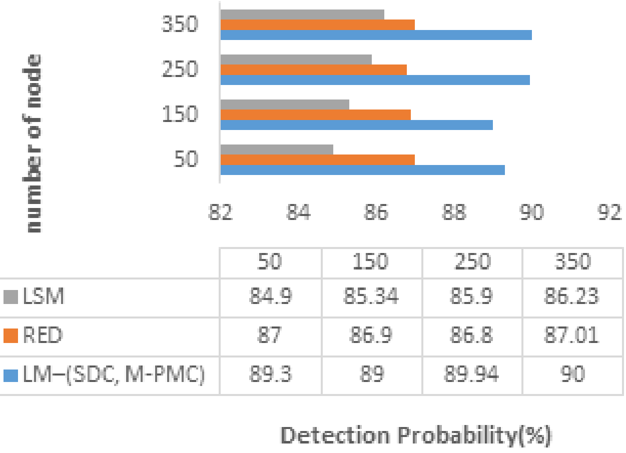
Fig. 4. The proposed architecture for protecting WSN with honeypot.

For blacklist replica nodes, we have two solutions: a) When CH is detected as a replica, it is removed or blocked; b) when a normal node is detected as a replica, it is deleted or blocked. For case a, when CH is blacklisted, BS distinguishes and selects CH from other nodes due to its higher energy than other nodes [2]. In case b, when a normal node is blacklisted or removed, the other node communicates with its shortest neighbor node according to distance. These solutions improve network performance.

4. PERFORMANCE EVALUATION

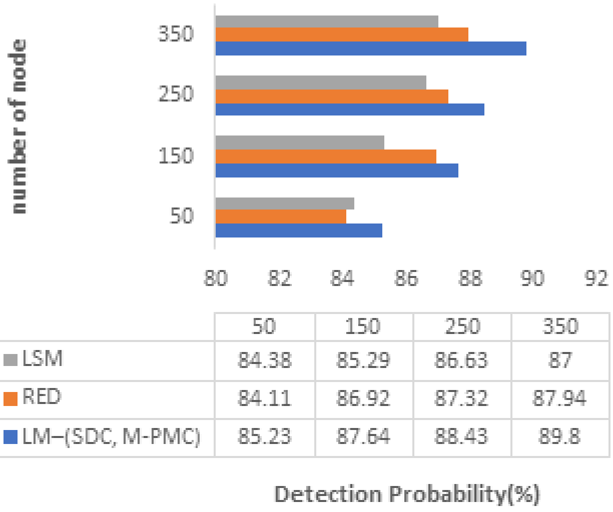
The NS2 simulation software is used for simulation. The network is a square of area (1500 × 1500) meters, and the nodes are placed arbitrarily. Here, two parameters, communication overhead and probability of identification, are considered to measure the performance of the proposed design. The proposed method Localized Multicast (SDC, M-PMC), is compared against the well-known LSM design [2] and RED Protocol [9]. Node deployment varies from 50, 150, 250, 350 nodes, and packet length is 512 bits. The initial energy of the node here is 100 joules. Fig. 5, 6, 7, and 8 compare the probability of identifying replica nodes between the proposed Localized Multicast – (SDC, M-PMC) methods, LSM, and RED protocol. Here the simulation is performed by considering 1, 3, and 7 replica nodes in the WSN.

Fig. 5 shows the probability of detection with one replica node. The proposed scheme creates the possibility of excellent diagnosis. Each inspection node works in parallel and does not need a control node. Duplicate detection is performed by checking the node ID and encryption key. All replica detection reports are sent to BS by inspector nodes. BS interrupts these reports and takes action. For this reason, the proposal is more effective based on the probability of diagnosis. As shown in Figs. 6 and 7, although the number of similar nodes increases, the detection probability is about 28% each time. Also, for the LSM method, the efficiency is close to 12% and for the RED protocol, close to 15%. Fig. 8 shows the packet delivery ratio for different numbers of nodes. The average packet transfer is negligible in the proposed method for communication between inspector nodes and BS.



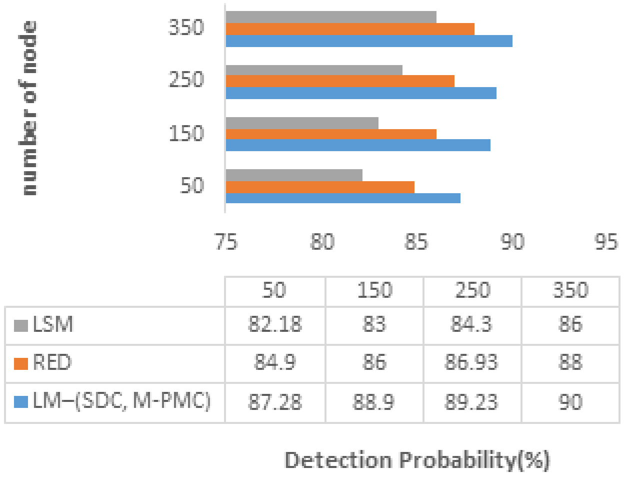
	50	150	250	350
■ LSM	84.9	85.34	85.9	86.23
■ RED	87	86.9	86.8	87.01
■ LM–(SDC, M-PMC)	89.3	89	89.94	90

Fig. 5. The detection probability with one replica node.



	50	150	250	350
■ LSM	84.38	85.29	86.63	87
■ RED	84.11	86.92	87.32	87.94
■ LM–(SDC, M-PMC)	85.23	87.64	88.43	89.8

Fig. 6. The detection probability with three replica nodes.



	50	150	250	350
■ LSM	82.18	83	84.3	86
■ RED	84.9	86	86.93	88
■ LM–(SDC, M-PMC)	87.28	88.9	89.23	90

Fig. 7. The detection probability with seven replica nodes.

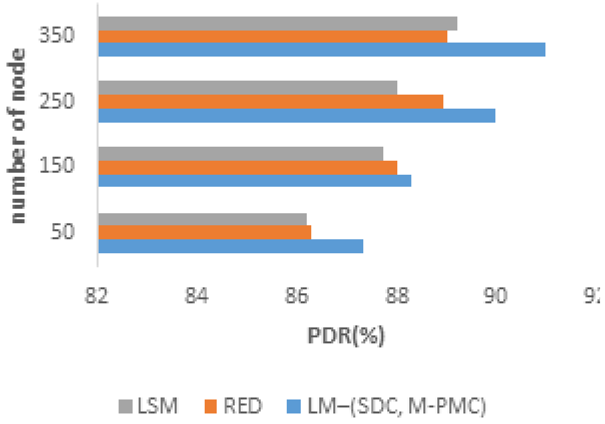


Fig. 8. The packet delivery ratio for different numbers of nodes.

5. Conclusion

This paper proposed a honeypot system for the early detection of node replication attacks in industrial control systems. The proposed method involves centralized inspector nodes, each of which operates in a distributed manner. An ID and an encryption key identify the replicated node. Here, no location information is required to determine the replica. We also introduced a honeypot scheme that increases the likelihood of detecting attacker nodes and clones. First, the base station learns the pattern of the honeypot attack. It then warns all network nodes about the enemy. Finally, we proposed two solutions for similar nodes in the blacklist in which CH and a natural node can be intentionally compromised. This increases network performance. The simulation result showed that the proposed scheme produces the maximum detection probability and, at the same time, has a low connection overhead.

REFERENCES

[1] Zhu, W. T. (2011). Node Replication Attacks in Wireless Sensor Networks: Bypassing the Neighbor-Based Detection Scheme. 2011 International Conference on Network Computing and Information Security.

[2] Wazir Zada Khan, M. Y. (2014). Detecting Replicated Nodes in Wireless Sensor Networks using Random Walks and Network Division. IEEE WCNC'14 Track 3 (Mobile and Wireless Networks .

[3] Sudipto, R., & Manisha, J. N. (2016). Prevention of Node Replication in Wireless Sensor Network using Received Signal Strength Indicator,Link Quality Indicator and Packet Sequence Number. 2016 Online International Conference on Green Engineering and Technologies (IC-GET).

[4] Umrao, S., Verma, D., & Kumar Tripathi, A. (2013). Detection and Mitigation of Node Replication with Pulse Delay Attacks in Wireless Sensor Network.

[5] P.Abinaya, & C.Geetha. (2014). Dynamic Detection of Node Replication Attacks using X-RED in Wireless Sensor Networks. ICICES2014 .

[6] Kaur, H., & Saxena, S. (2017, July 3-5). A Review on Node Replication Attack Identification Schemes in WSN. 8th ICCCNT 2017 .

[7] Farah, K., & Nabila, L. (2014). The MCD Protocol for Securing Wireless Sensor Networks against Nodes Replication Attacks. 2014 International Conference on Advanced Networking Distributed Systems and Applications.

[8] D.Sheela, Priyadarshini, & G.Mahadevan, D. (2011). EFFICIENT APPROACH TO DETECT CLONE ATTACKS IN WIRELESS SENSOR NETWORKS

[9] Markert, J., & Massoth, M. (2014). Honeypot Effectiveness in Different Categories of Attacks on Wireless Sensor Networks. 25th International Workshop on Database and Expert Systems Applications .

[10] Yu, C.-M., Lu, C.-S., & Kuo, S.-Y. (2008). Mobile Sensor Network Resilient Against Node Replication Attacks.

[11] Marjit Singh, M., Singh, A., & Mandal, J. K. (2014). PREVENTING NODE REPLICATION ATTACK IN STATIC WIRELESS SENSOR NETWROKS.

[12] Pashaei, A., Akbari, M. E., Lighvan, M. Z., & Teymorzade, H. A. (2020, June). Improving the IDS performance through early detection approach in local area networks using industrial control systems of honeypot. In 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe) (pp. 1-5). IEEE.

[13] N.S.Usha, & Anita, D. (2017). A novel zone based routing protocol for detection ofreplicas in static wireless sensor networks. International conference on information,communication & embedded systems (icices 2017).

[14] Muraleedharan, R., & Ann Osadciw, L. (2009). An Intrusion Detection Framework for Sensor Networks Using Honeypot and Swarm Intelligence.