ISSN 2278-3091
Volume 10, No.3, May - June 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse111032021.pdf
https://doi.org/10.30534/ijatcse/2021/111032021

Mobile Wireless Sensor Networks: A Framework Based Solution for Physical Security Attacks

Mumtaz Qabulio[1], Yasir Arafat Malkani[2], Muhammad Suleman Memon[3], Ayaz Keerio[4]
[1]Department of Software Engineering, Faculty of Engineering, University of Sindh, Jamshoro, Pakistan, mumtaz.qabulio@usindh.edu.pk
[2]Institute of Mathematics & Computer Science, University of Sindh, Jamshoro, Pakistan, yasir.malkani@usindh.edu.pk
[3]Department of Information Technology, Dadu Campus, University of Sindh, Jamshoro, Pakistan, msuleman@usindh.edu.pk
[4]Institute of Mathematics & Computer Science, University of Sindh, Jamshoro, Pakistan, ayaz@usindh.edu.pk

ABSTRACT

Mobile wireless sensor networks are made up of resource-constrained sensor nodes. The nodes in Mobile wireless sensor networks tend to move wirelessly from one location to another. In physically unattended and insecure areas, these networks are installed where they are vulnerable to numerous security threats, including physical security threats. Clone node attack is a physical attack in which the adversary physically compromises legitimate nodes and copies their data onto fake nodes to create clones of captured nodes. In static wireless sensor networks, comprehensive work is already done to mitigate and detect the underlying attack. Mitigating and detecting this assault on mobile wireless sensor networks, however, remains a critical issue. To our knowledge, there is no framework-based method for bolstering multiple attacks on mobile wireless sensor networks.Consequently, this article proposes a framework-based approach that mitigates multiple attacks on mobile WSNs. To provide proof-of-concept for the framework, along with clone node attacks solution to jamming attacks is also incorporated in the framework. The implementation of the framework is done with the COOJA simulator and Contiki OS.

Key words : Mobile Wireless Sensor Networks, Security Attacks, Clone node attack, Jamming Attack, Security Framework.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] consist of very small , low-power units. Such small devices are referred to as sensor nodes, which are responsible for communication between nodes and other devices. In WSNs, the nodes are densely deployed throughout the network. WSNs have a wide range of applications in everyday life, including the measurement of physical environmental phenomena such as temperature, fire, heat, dust, pressure, and soil, object movement, the presence and absence of specific objects, human activity detection, and so on. Static and mobile WSNs are the two types of WSNs. The nodes are fixed in static WSNs and do not have any locomotion properties, where the nodes will switch from one position to another position within the network in mobile WSNs. The versatility aspect of mobile WSNs, however, poses different critical issues in the deployment of mobile WSNs. Some of MWSNNN's active research problems are [2]–[4].are localization, synchronisation, heterogeneity, routing, connectivity, scalability, energy consumption, testing , deployment, fault-tolerant connectivity, topological problems, data aggregation problems and protection. The security of mobile WSNs is a critical issue because they are deployed in physically unprotected and unattended environments. In addition to this, the mobility feature fosters another security barrier for the Mobile WSNs. Due to these reasons, doors have been unlatched for security attackers to instigate several attacks in the network. Security threats are categorised as active and passive attacks, including eavesdropping, selective routing, Man-in-the-Middle, sniffing, Sybil, subversion of the network, HELLO flood, black hole, sinkhole, fake network, Denial of Services, wormholes, jamming, and targeting the clone node [5]. This article focuses on the problem of **Clone Node** and **Jamming** attacks in mobile WSNs. In the literature, the clone node attack is often quoted as a node replication attack [6], [7]. The attacker first captures one or more legitimate nodes from the network in this attack, extracts credentials and data stored on the local memory of the node, copies extracted data to one or more nodes in order to establish clone nodes, and redeploys those nodes to the network. One of the important attacks in the literature is the clone node attack [2], [5], [8]–[12] if an attacker successfully launches clone nodes in the network, as in this attack, the attacker may then launch additional possible attacks. And he/she can eventually take the overall network in control. Furthermore, for the legitimate nodes of the network, the cloned node appears the same as the genuine node of the

network as it has the same credentials, and uses the same protocols used by the genuine nodes of the network. As a result, based on data stored, it is almost impossible for the rest of the network nodes to differentiate between the legitimate node and the cloned node of the network. However, by closely observing the behaviors of the network nodes, it is possible to Ascertain and differentiate between both nodes. As, despite having the same credentials and data, the cloned node will behave differently, unwontedly, and maliciously within the network. As far as the jamming attack is considered, a different type of jamming attack is there (e.g. constant jamming, random jamming, deceptive jamming, reactive jamming) attacks. ***The constant jammer*** sends high power random signals continuously over the network channel without using any protocol format [13]. This never-ending signal transmission ultimately converts to a denial-of-services attack. The deceptive jammer sends out false but regular packets on a regular basis, or replays old packets. They are familiar with the network nodes' link-layer communication protocol [14], [15]. Thus, they are more dangerous than constant jammers. [16], [17]. ***The random jammer*** switches their states between active and sleep states. During the active state, it either transmits random bits or regular messages over the channel. ***The reactive jammer*** remains in receiving mode while listing the channel for activities, they send jamming signals whenever any preamble bits are detected. These jammers can corrupt a large amount of network packet [15]. After defining different types of jammer, it can be noticed that each of the jammers has specific behavior, which is also different from the legitimate nodes.

Consequently, the article provides a solution that detects clone nodes and the jammer nodes from the mobile WSNs based on their malicious behaviors.

2. RELATED WORK

2.1 Di-Sec Framework

Di-Sec is a modular framework proposed for static WSNs by Valero et.al in [18]. Di-sec provides security against jamming, Selective forwarding, and Sybil Attacks. the framework is composed of four major components. The first one is Sense (Sensing Module), it performs the basic job of the network that is sensing. The second component is *COMM* (Communication model), it provides interfaces for the communication between the framework and the application layer. The third one is the M-Core (Monitoring-Core), it gathers, aggregates, and distributes the sensed data. The final one is DDMs (Detection and Defense Module), it is responsible for detecting and providing security mechanisms against the aforementioned attacks. To deal with each attack, the DDMs have separate modules. To detect Jamming, Selective Forwarding, and Sybil attacks, each module receives data from the sensor portion and performs security checks. The framework is implemented and test on the TinyOS operating system Tmote Sky motes.

2.2 Bonaci's Framework

Bonaci et.al in [19] has proposed a framework based solution for static WSNs to handle node capture attacks. The framework is based on control theory. In the underlying framework, a linear dynamical model is derived in which all the parameters in the network are characterized according to the parameters of the model. For identifying node capture attacks, controllers are categorized by using Linear-Quadratic-Gaussian (LQG) and Linear-quadratic Regulator (LQR) optimal control theories.

2.3 Kalpana Sharma's Framework

The authors in [20] have presented a cross-layered framework based on key management protocol for static WSNs. In the underlying framework, first, a master key is distributed among all legitimate sensor nodes, the sensor nodes after receiving the master key, compute three session keys, and delete the master key. Finally, the sensor nodes use RC5 bock cipher along with keys and monotonically increase the counter to protect the network against replay attacks, authentication attacks, and node capture attacks. From the literature, it is found that all the frameworks are proposed for static WSNs and there is not any framework proposed which can secure mobile WSNs from multiple attacks concurrently. The proposed architecture, in comparison to these implementations, is usable for static as well as mobile wireless sensor networks. The implementation of the framework is done via Contiki OS and COOJA simulator. The testing and evaluation of the framework performed with different methods which are discussed in forthcoming sections.

I. PROPOSED FRAMEWORK

At the general level, the architecture comprises three layers, namely the front-end layer, the middle layer, and the hardware layer. Figure 1 depicts the proposed framework along with its layers. The first layer is the hardware layer it contains the sensor nodes, actuators, and optionally an additional power source. The third layer is the front-end layer or Application layer which provides interface. The middle layer is the security providing layer, it is responsible to detect the cloned node and jamming attacks and to discard such nodes from the network.

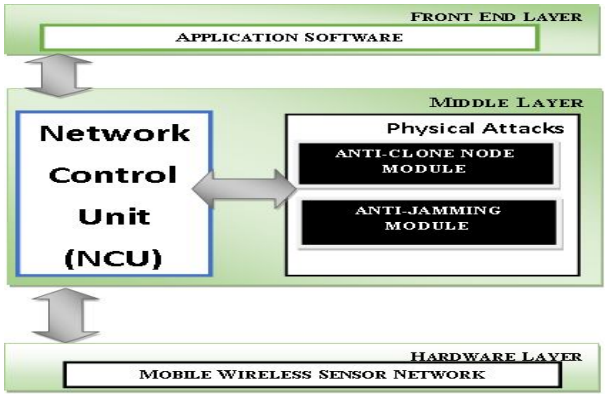

Figure 1: Proposed Framework

The middle layer is further comprised of three modules, Network-Control-Unit (NCU) module, anti-jamming modules, anti-clone node module respectively. The anti-clone node and anti-jamming modules contain attack specific logic to detect the attacks. While Network-Control-Unit is the core module of this layer, it stores and records behaviors and data related to the sensor nodes. The NCU also provides communication between the hardware and application layers. The data (packets) sent from mobile WSNs are received by NUC. The NCU examines received packet against clone node and jamming attacks and forwards the packet to the application layer if packets are verified. The logic of the middle layer needs to be executed on the base station (BS). And to simplify the working mechanism of the framework, the layer running middle-layer will be referred to as BS in upcoming explanations. As the base station will be running middle-layer logic, the following are some assumptions made for the Base station:

- The BS is an authentic and trusted entity
- BS is static and is placed at a fixed location.
- It is a physically protected entity

The NCU for storing data and behaviors of the sensor nodes maintains two tables in memory which are referred to as *Member-nodes* and *clone-nodes respectively*. Table 1 and Table 2 presents attributes stored in each of the tables along with attribute description and possible values for the attribute.

Table 1: Clone Node Table

Attribute	Description	Possible Value
Node Id	Two bytes long unique identifier of the node	Node id
Node Status	Status of the node as clone node	2

The *member-nodes* table is the foremost table, it stores all information related to each sensor node, that information is required by anti-clone node and anti-jamming modules to find the jammer nodes and clone nodes in the networks. The NCU uses clone node table to keep record of the clone nodes found from the MWSNs.

Furthermore, a light-weight protocol called JamP is designed for the network packets. JamP is an application layer protocol that defines the format for the network packets. Figure 2 illustrates the JamP protocol. The main motivation behind designing this protocol is to detect jamming attacks instead of clone node attacks. The JamP protocol helps NCU to verify packets sent by legitimate nodes of the network.

Table 2: Member Node Table

Attribute	Description	Possible values
Node Id	Two-byte long unique identifier of the node	Unique
Node-Status	Shows status of the network nodes	0 Busy in communication 1 Available for communication 2 Clone node 3 Out of range node
node-time stamp	Time on the system in milliseconds.	System time in milliseconds
jammer-node	status of node as jammer or non-jammer	0 Jammer Node 1 Non-jammer Node
channel-access-frequency (CAF)	The number of times that the medium is accessed by the required node.	Numeric value
threaten-node	status of the node as threaten or non-threaten node	0 Threaten Node 1 Non-threaten Node
packets-received	Number of packets received successfully at the destination.	Numeric value
packet-delivery-ratio	The ratio between the packets received successfully and the total sent.	$PDR = \frac{P_{suc}}{P_{tran}}$
session-id	ID used to maintain the respective node 's contact session with another network node.	System time

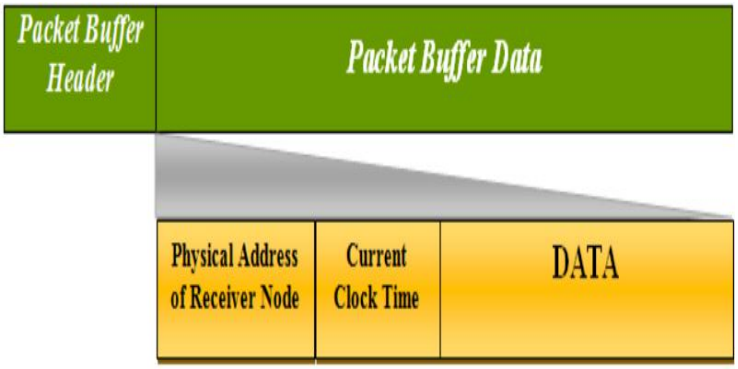

Figure 2: JamP Protocol Format

A. *Communication Process*

In this suggested structure, the network is initially implemented with only genuine nodes, a unique identifier is allocated to every node in the network, and the BS maintains and maintains full records for the network nodes. Both nodes send beacon messages to the BS periodically. Nodes use beacon messages to display their network availability; they include a node ID and a text message. The BS also periodically broadcasts a list of available nodes to all network nodes. The network nodes interested in communication can pick id to form the list of available nodes and can send a request for initiating the communication session. Furthermore, the framework is based on connection-oriented communication between base station and network nodes which is responsible for creating and terminating the connections. The network node requires three-way handshaking mechanisms for establishing the connection. For this reason, there are various types of packets used by the framework. Table 3 shows various types of packets used by the proposed framework along with their purpose.

Table 3: Packet Types used by Proposed Framework

Packet Type	Abbreviation	Purpose
RTS	Request to Send	Sent by nodes to request BS for establishing the connection with the desired node.
CHK	Check	The purpose of the CHK packet is twofold, first, it is used to detect clone nodes for both sender and receiver motes. Second, it checks the presence of the receiver node in the network
RSP	Response	Used to respond to CHK message
EoC	End of Communication	Used to terminate the connection
Data	Data packet	Contains data

B. *Clone Node Attack Detection*

The proposed system detects clone nodes in two ways, via the CHK test and periodic unicasting of beacon message. Both tests are described below.

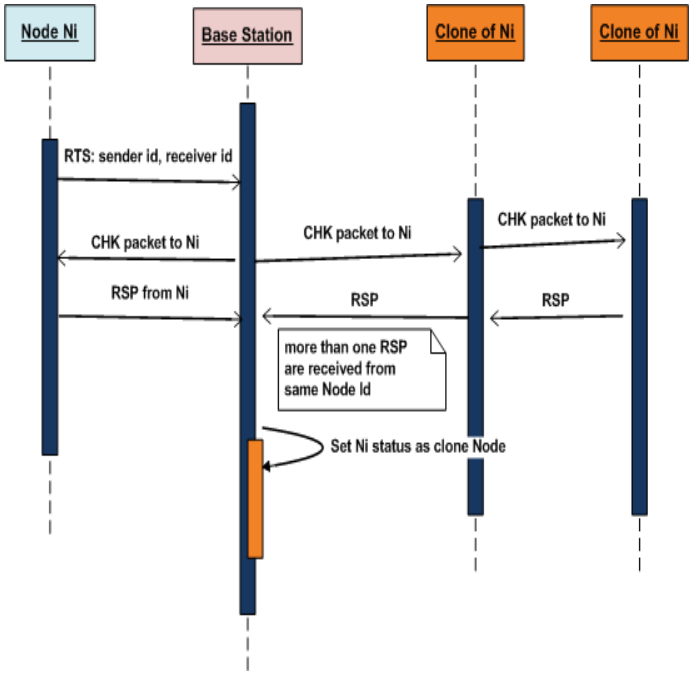

Figure 3: CHK Test Sequence Diagram

1) *Periodic test for clone node detection*

As mentioned earlier, each network node must periodically send its beacon message to the BS to exhibit its existence in the network. The BS after receiving beacons from corresponding nodes first verifies the packet format defined through *JamP protocol*. After format verification, the base station looks overs the node status, if it is not 2 (that means the node is a legitimate one, not a clone) the BS generates the current timestamp for the corresponding node. After this, the timestamp is subtracted from the generated timestamp. In this scenario 3 possible conditions are considered:

- The result of the case is equal to some pre-defined threshold value; a valid node is considered by the BS and its old time stamp is updated with a new one.
- In case results are higher than the predefined threshold value, node status is updated to 3 (out of range node).
- The status of the node is modified to 2 if the result is below the predefined threshold value.

Figure 4 depicts this examination graphically. Where node 2 clone exits, and the base station recognises node 2 as a clone node after receiving a beacon message and thus updates the status of that node as 2.

C. *Jamming Attack Detection*

Before discussing the working mechanism of jammer detection, we need to understand that what jammers are? Jammers are devices that continuously send packets over the network channel to jam the network or to launch jamming attacks in the network. There are four different types of jammer the random jammers, constant jammers, deceptive jammers, and reactive jammers respectively. Each jammer has different behavior though the same intention.

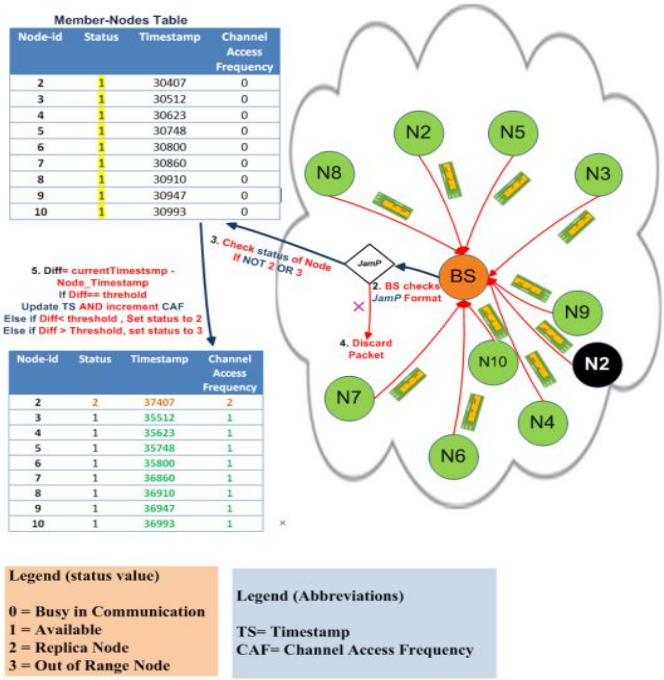

Figure 4: Clone Node Detection with Periodic Test

The constant jammers are devices that *continuously* send *random bits* over the network. As a result, will have a high number of channel access frequency. Random jammers are devices that *randomly* send *random bits* over the network. The deceptive jammers are devices that *continuously* send *regular packets* over the network. Hence, it will have a high channel-access-frequency value. In addition to this, these jammers are aware of network protocols and formats. Finally, the reactive jammers are devices that send *regular packets* over the network when they sense any activity over the channel as a result, they will have almost zero packet delivery ratio. The proposed framework detects jammer nodes at two levels

1) *At the reception of each packet*

The proposed framework detects the constant, random, and simple deceptive jammers as soon as the first packet is received from these jammers. Whenever packets from these jammers are received to the base station, the base station verifies packets against Jamp packet format. As constant and random jammers only send random bits over the network and deceptive jammer are aware of network defined packet formats such as MAC, they are not aware of JamP. As a result, checking packets against JamP format helps the base station to detect all three mentioned types of jammers.

2) *At the execution of periodic test*

In addition, the base station checks in on a regular basis to identify and remove smart deceptive and reactive jammers from the network. For this examination, the base station examines the network's channel-access-frequency and PDR values. Reactive jammers are nodes with a CAF greater than 1 and a PDR less than 70%, and they are automatically added to the jammer list and removed from the network. The situation is different in the case of dishonest jammers. In this case, the base station first tests each network node's channel access frequency, and if the CAF of any node exceeds the predefined threshold value, the BS adds that node to the list of threat nodes by setting the attribute value to 0. When the same test is run a second time and the base station discovers the same node with a channel-access-frequency greater than the threshold value and the node is already marked as a threat-node, the BS will set the node's jammer-node attribute to 0. The BS discards the packets obtained from such nodes. Furthermore, a periodic test is run at random intervals rather than at a set interval. The procedure calculates the interval's minimum and maximum lengths. The shortest time interval is one minute, and the longest time interval is ten minutes. The logic behind conducting this test after a random period of time is simple: a brute force attack with fixed threshold values will allow an attacker to guess the threshold values. As a consequence, the adversary will limit the jamming device's access to the medium to less than the specified maximum channel-access-frequency value after obtaining a fixed threshold value. With this, the jamming device will easily be bypassed by the periodic test. Consequently, the attacker will get succeed to launch an undetectable jammer in the network. Consequently, the proposed framework computes random threshold values for a random time. First, there is a need to examine the minimum requirement for a node to access the medium (channel-access-frequency) per unit of time to compute the threshold value for each random time period. Here, as a unit of measurement, 1 minute is set. The channel-access-frequency specifications are evaluated for a unit of time and, finally, the following formulas are derived for the computing threshold for each random time period, based on the appropriate channel-access-frequency.

Expected CAF / minute = (sent sensed data) + (sent beacon messages)

RandomTime $_x$ = Randomly selected time interval where $x >= 1$ and $x <= 10$

CAF for random time period = RandomTime $_x$ * (Expected CAF/minute)

Threshold CAF = 3 * CAF for a random time

II. IMPLEMENTATION

The proposed framework is implemented using Contiki operating system and COOJA simulator. The proposed framework is tested on various sensor nodes which includes Tmote Sky [21], Zolertia Z [22], MicaZ [23], and Embedded Sensor Board (ESB) [24]. Table 4 shows the specifications of all four emulated sensor nodes. The Random Waypoint Model is used to test and examine the proposed

solution under mobility conditions. The model allows mobile motes to randomly new destination, direction, and speed randomly from the already defined values. Pause times are used before switching at new directions or speeds during simulation. Table 5 summarizes simulation parameters.

Table 4: Features of emulated sensor nodes

Mote Type	***Wireless Transceiver***	***Microcontroller***	***Frequency band***	***Data rate***
Tmote Sky	IEEE 802.15.4	MSP430	2.4 GHz	250 Kbps
Zolertia Z1	IEEE 802.15.4	MSP430 and CC2420	2.4 GHz	250 Kbps
MicaZ	IEEE 802.15.4	ATmega128L	2.4 GHz	250 Kbps
ESB	IEEE 802.15.4	MSP430	2.5GHz	250 Kbps

Table 5: Simulation parameters

Simulation Parameter	**Value**
Simulator used	COOJA
Emulated sensor motes	Tmote Sky, MicaZ, Zolertia Z1, ESB
Network range	200m
Node Deployment type	Random
Mobility type	The random waypoint mobility model
Pause time	3 seconds
Simulation time	Minimum 30 seconds, maximum of 12 minutes
Radio medium type	Unit disk graph medium: distance loss

Figure 5: Simulation with 200 Nodes

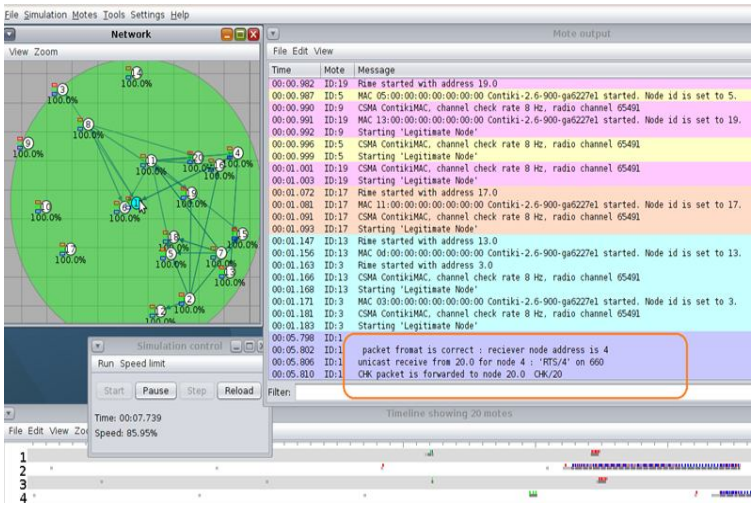

Figure 6: Execution of CHK test

Figure 7 Execution of beacon test

Figure 8: Clone Nodes Found via Periodic Test

III. Performance and Security Evaluation of Proposed Framework

The proposed system is evaluated with different evaluation methods. First, the mean of times taken by the system to each of the attacks is calculated. Second, the time, communication, and space complexities of the algorithms used by both modules are calculated. Finally, the system is executed under various real scenarios specific to the mobile wireless sensor networks to inspect the system behavior and its reliability for the detection of attacks under real-world scenarios.

A. *Performance Analysis*

The performance analysis of both modules is carried out to detect each attack in terms of the average time taken by the modules. For this, for each of the experiments, several simulations were conducted, and individual times were quantified along with the average time taken by modules to detect clone nodes (e.g. CHK test, beacon test) and jammer nodes (e.g. constant, random, reactive, deceptive). A time graph for all test executions is shown in Figure 9 through Figure 12.

B. *Algorithm Complexities*

5.2.1 Time Complexity of Proposed System

This system detects clone nodes via CHK and beacon tests.

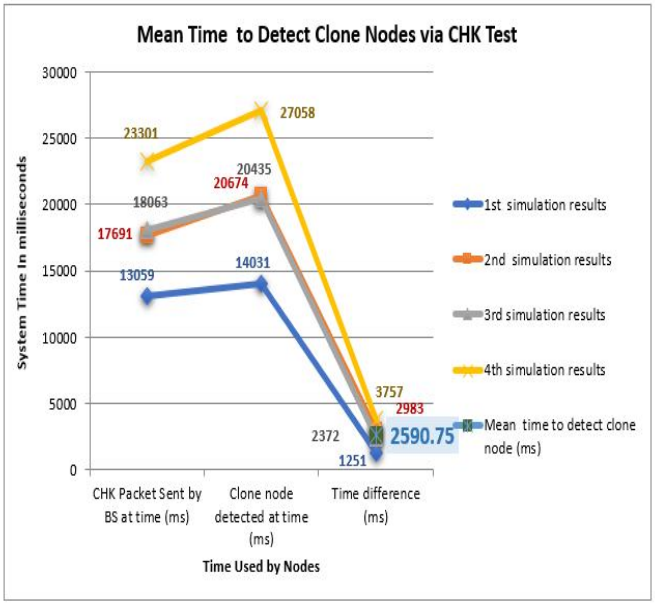

Figure 9: Performance Analysis of CHK Test

The full member-Node table is examined by the beacon test, which is why the time complexity for detecting clone nodes through the beacon test is O (1) for the best case, O (log N) for the average case, and O (N) for the worst case. For all three (best, average, worst) instances, the time complexity for detecting clone nodes via the CHK test is O (1) since only one comparison is involved with the JamP format.

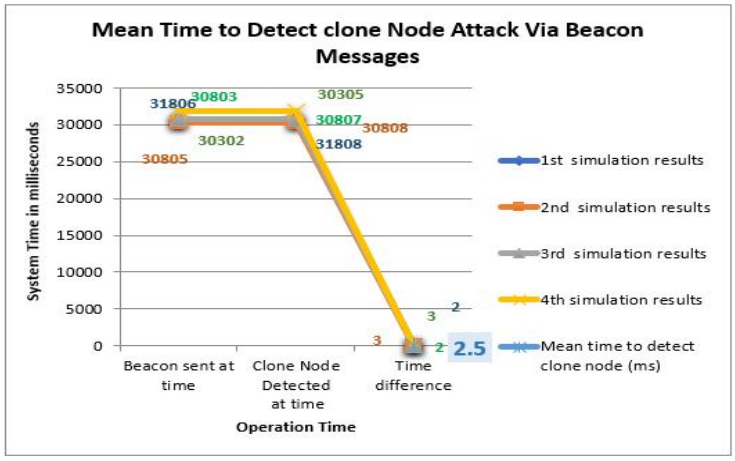

Figure 10: Performance Analysis of Periodic Test

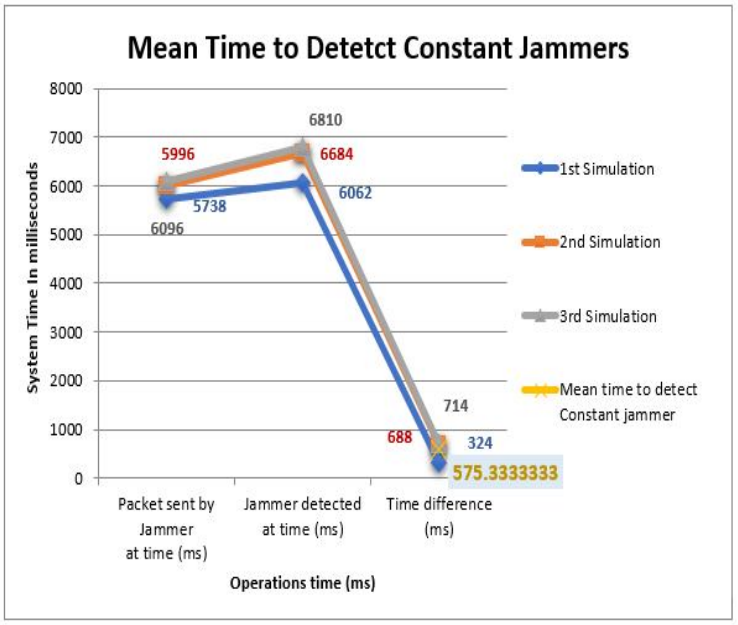

Figure 11: Performance Analysis of Constant Jammer Detection

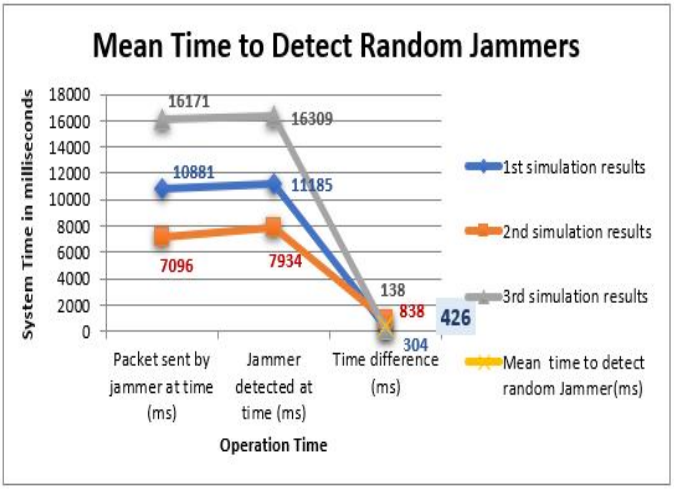

Figure 12: Performance Analysis of Random Jammer Detection

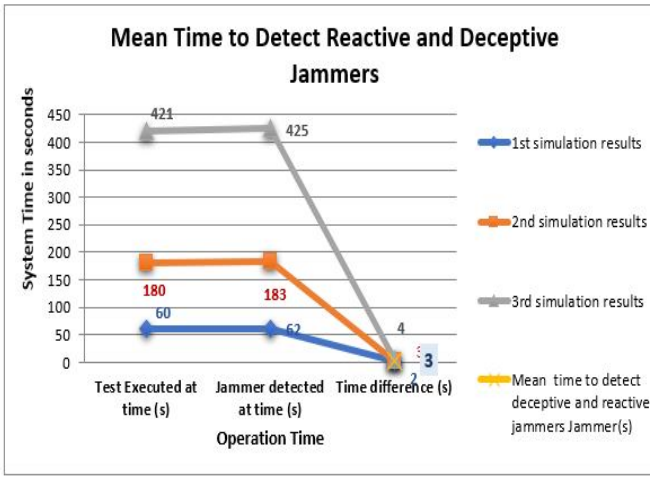

Figure 13: Performance Analysis of Reactive and Deceptive Jammer Detection

Constant and random jammers are found in the anti-jamming module. Since BS receives the very first packet from these jammers, in all three cases, time problems are present for both jammers O (1). The identification of deceptive and reactive jammers is carried out through a periodic test in which the base station tests the attributes of the channel-access-frequency and packet-delivery-ratio of each node available in the table of the member nodes, so the difficulty of detecting both jammers is O(N) for all cases. The table below presents all discussed complexities.

Table 6: Time Complexities of the Modules

Type of Test	*Best-case*	*Worst-case*	*Average case*
CHK test	O (1)	O (1)	O (1)
Beacon test	O (1)	O (N)	Log (N)
Constant Jammer	O (1)	O (1)	O (1)
Random Jammer	O (1)	O (1)	O (1)
Deceptive Jammer	O (N)	O (N)	O (N)
Reactive Jammer	O (N)	O (N)	O (N)

5.2.2 Communication Complexities/ Overheads

In the beacon test, during each beacon test, the total number of messages sent over the network is N. All nodes regularly unicast their beacon message to the BS. Consequently, the overhead / complexity of communication for the detection of clone nodes via beacon test is O (N) for all cases. Each packet is reviewed by BS upon reception in the CHK test, so the communication overheads for this test are O (1) since it requires a single packet to be reviewed. O(1) is also the overhead for communication to detect constant and random jammers, since it only requires sending a single packet to BS. The deceptive and random jammers are detected by the base station by locally executing a periodic test. The test does not require any data transfer between network nodes. As a result, there are zero communication overheads for detecting these two jammers. The table below provides a summary of the communication overheads.

Table 7: Communication overheads of the used algorithms

	Worst-case	**Average-case**	**Best-case**
CHK test	O (1)	O (1)	O (1)
Beacon test	O(N)	O(N)	O(N)
Constant Jammer	O (1)	O (1)	O (1)
Random Jammer	O (1)	O (1)	O (1)
Deceptive Jammer	Zero	Zero	Zero
Reactive Jammer	Zero	Zero	Zero

3) Storage Complexities/ Overheads

As the approach used by the proposed system is a centralized approach, wherein records of each detected malicious node (clone node or jammer node) is stored at the base station only. As a result, the storage complexity/overheads of detecting clone nides and jammer nodes is *O (1).*

5.3 Test-case Scenarios

To examine the efficiency and reliability of the proposed framework in attack detection, it is evaluated by employing 13 real-world test-case scenarios. Table 7 presents notations employed to depict those scenarios while table 8 describes the scenarios along with their details.

Table 8: Notations used to define scenarios

Notation	**Corresponding meaning**
Ni	Sender Node
Ni’	Clone of Ni
Nj	Genuine Receiver Node
Nz	Another Genuine Network Node
Ts	Timestamp
Jcont	Constant Jammer
Jdecp	Deceptive Jammer
Jrand	Random Jammer
Jreac	Reactive Jammer

Table 9: Scenario to test Framework

Scenario	***Scenario Description***
Ni does not exist or is not part of the network and attempts to connect with Nj.	*The Ni is an unregistered node, and it tries to send data to a registered network node.* *10 such type of the nodes were deployed and BS detected all those nodes*
By cloning one or both nodes, Ni and Nj are busy communicating and sending RTS to the base station.	*Ni and Nj are exchanging data with each other, their statuses are busy, during their session Ni' or Nj' sends RTS to BS.* *8 such nodes were deployed; the BS detected all those clone nodes.*
There is a Ni clone node on the network, and both send messages from the beacon to the base station.	*The Ni and Ni' both send Beacon message to the BS, the BS during a periodic test can detect Ni'.* *The network deployed 9 clone nodes, and all were found during the periodic examination.*
Ni is silent and Ni' tries to communicate with Nj	*The legitimate node Ni is free is not exchanging data with network nodes, the Ni' sends RTS message to the BS.* *5 clone nodes were deployed for different non-communicating nodes, all were detected successfully.*
Ni' is Silent, and Ni initiates communication with Nj	*Here, the cloned node of Ni is free, and Ni sends RTS to the BS.* *5 silent clone nodes were deployed for this scenario, and BS detected all 5 clone nodes successfully.*
Out of range nodes were deployed which tried to re-join the network.	*This scenario tested nodes who were registered but went out of range and now requires rejoining the network.* *6 nodes out of 15 were brought out of the range and then brought in a range of the BS, the BS successfully identified all those nodes*
All three types (unregistered, out of range, and clone node) of nodes were deployed.	*In this scenario, unregistered, clone, and out of range nodes where deployed.* *In this scenario, the base station also detected unregistered, clone, and out of range node.*
Constant Jammers are deployed	*In this scenario, 11 constant jammers were deployed in the network.* *The system successfully detected 11 constant jammers from the network.*
Random Jammers are deployed	*In this scenario, 11 random jammers were deployed in the network.* *The system successfully detected 11 random jammers from the network.*
Deceptive Jammers are deployed	*11 Smart deceptive jammers were deployed along with genuine nodes, the system successfully detected smart deceptive jammers from the network.*
Reactive Jammers are deployed	*In this scenario, 11 reactive jammers were deployed, and all were detected by the system.*
All four type of Jammers are deployed	*In this scenario, all four types of jammers were deployed in the network and they were detected by the base station.*

IV. LIMITATIONS OF THE WORK

This section of the papers presents some of the limitations of the proposed system.

First, the framework proposed, and its modules are implemented and tested on the COOJA simulator, not on the real-world hardware. Second, the absolute security of the system relays on base-station, which causes the system prone to a single point of failure (SPOF) threat. Third, the framework and algorithms are implemented on Contiki OS only, which causes platform dependency.

V. CONCLUSION

Mobile wireless sensor networks are networks with minimal or no human interaction and are deployed in an open, hash, and unprotected environment. Wherein they are open to many physical security attacks. In this regard, the article provides a framework-based solution for mitigating two physical attacks including clone node and jamming attacks. Separate algorithms were designed to deal with each attack, respectively. A prototype of the framework was implemented for proof-of-concept using Contiki OS and COOJA simulator. To test the compatibility of the algorithms it is executed on Tmote sky, MicaZ, Zolertia Z1, and ESB motes. the framework and algorithms executed successfully. Finally, the performance and security analysis of the system and algorithms were performed and presented using different methods.

VI. REFERENCES

[1] Kim and S. Han, "An Efficient Sensor Deployment Scheme for Large-Scale Wireless Sensor Networks," vol. 19, no. 1, pp. 98–101, 2015.

[2] M. Erdelj, "Mobile Wireless Sensor Network Architecture: Applications to mobile sesnor deployement," Université des Sciences et Technologie de Lille-Lille I, 2013.

[3] M. M. Javad Rezazadeh Abdul Samad Ismail, "Mobile Wireless Sensor Networks Overview," *IJCCN Int. J. Comput. Commun. Networks*, vol. 2, no. 1, 2012.

[4] M. B. Raja Waseem Anwar Anazida Zainal, Abdul Hanan Abdullah and Kashif Naseer Qureshi, "Security Issues and Attacks in Wireless Sensor Network," *World Appl. Sci. J.*, vol. 30, no. 10, 2014.

[5] D. S. Dr. G. Padmavathi, " A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks," *Int. J. Comput. Sci. Inf. Secur.* , vol. 4, no. 1, 2009.

[6] R. D. Pietro Mauro Conti Luigi V. Mancini, "A Randomized, Efficient, and Distributed Protocol for the Detection of Node Replication Attacks in Wireless Sensor Networks," *MobiHoc'07*. .

[7] M. Faisal, S. Abbas, and H. U. Rahman, "Identity attack detection system for 802.11-based ad hoc networks," *EURASIP J. Wirel. Commun. Netw.*, vol. 2018, no. 1, p. 128, 2018.

[8] R. D. Pietro Mauro Conti Luigi V. Mancini, and Alessandro Mei, "Distributed Detection of Clone Attacks in Wireless Sensor Networks," *IEEE Trans. DEPEDABLE Secur. Comput.*, 2005.

[9] C.-S. L. Chia-Mu Yu and Sy-Yen Kuo, "Efficient and Distributed Detection of Node Replication Attacks in Mobile Sensor Networks," *Vehicular Technology Conference Fall (VTC 2009-Fall)*. IEEE, 2009.

[10] V. G. K. A. Bo Zhu Sanjeev Setia,Sushil Jajodia, Sankardas Roy, "Efficient Distributed Detection of Node Replication Attacks in Sensor Networks," *Computer Security Applications Conference*. IEEE, 2007.

[11] M. M. Wassim Znaidi and Stéphane Ubéda, "Hierarchical Node Replication Attacks Detection in Wireless Sensor Networks," *Int. J. Distrib. Sens. Networks*, 2013.

[12] S. A. Sinthiya, "In Mobile Sensor Networks Localized Algorithms for Detection of Node Replication Attacks," *Int. J. Res. Stud. Comput. Sci. Eng.*, vol. 1, no. 1, 2014.

[13] M.-L. Messai, "Classification of Attacks in Wireless Sensor Networks," *International Congress on Telecommunication and Application'14*. 2014.

[14] K. M. Wenyuan Xu Wade Trappe, and Yanyong Zhang, Rutgers University, "Jamming Sensor Networks: Attack and Defense Strategies," *IEEE Networks*, 2006.

[15] D. G. Aristides Mpitziopoulos Charalampos Konstantopoulos, and Grammati Pantziou, "A Survey on Jamming Attacks and Countermeasures in WSNs," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS*, vol. 14. 2009.

[16] P. N. R. S.M.K .Chaitanya Y.N.V.L. Ayyappa and Vundavalli Ravindra, "Analysis and Study of Denial of Service Attacks in Wireless Mobile Jammers ," *Int. J. Comput. Sci. Telecommun.*, vol. 2, no. 5, 2011.

[17] R. D. C. Ju young Kim Kyung Jung Kim, "A Review of the Vulnerabilities and Attacks for Wireless Sensor Networks," *J. Secur. Eng.*, 2012.

[18] S. S. J. M. Valero A. Selcuk Uluagac and Y. Li and R. Beyah, "Di-Sec: A distributed security framework for heterogeneous Wireless Sensor Networks," in *INFOCOM*, 2012.

[19] T. Bonaci Linda Bushnell, and Radha Poovendran, "Node capture attacks in wireless sensor networks: a system theoretic approach," *49th IEEE conference on Decision and Control (CDC)*. 2010.

[20] M. K. G. and K. Kalpana Sharma, "Complete Security Framework for Wireless Sensor Networks ," *Int. J. Comput. Sci. Inf. Secur.*, vol. 3, no. 1, 2009.

[21] H. Harvard school of Engineering and Applied sciences, "Tmote Sky datasheet." 2006.

[22] W. S. N. Zolertia, "platform, Z1 Datasheet." ed.

[23] M. MICAz, "Wireless sensor networks," *Doc. Part*, no. 6020–0042, p. 4, 2013.

[24] F. U. of Berlin, "http://www.mi.fu-berlin.