MITIGATING DDOS ATTACKS IN IOT: A THRESHOLD-BASED COUNTERMEASURE AGAINST REPLAY ATTACKS

V.N.Vasanthi[1], S.Perumal[2]

[1]Research Scholar, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
[2]Professor, Department of Computer Science, Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai, Tamil Nadu, India-600117.
Email: vnvasanthi78@gmail.com[1], perumal.scs@velsuniv.edu.in[2]

Article History: Received: 19.04.2023 Revised: 02.05.2023 Accepted: 10.06.2023

Abstract: DDoS attacks present a significant danger to computer networks, as they can cause disruption and deny access to legitimate users. Although various techniques have been proposed to detect and prevent jamming attacks, there is a need for more efficient solutions to tackle DDoS attacks. This work presents a taxonomy of different DDoS attacks in the Internet of Things (IoT) network and proposes a solution for replay attacks. Using Unified Modeling Language (UML) based activity modeling, all variations of DDoS attacks are modeled to understand their behavior and performance in the environment. This knowledge is used to develop a Threshold Based Countermeasure for replay attacks, allowing the attack inside the network and blocking it after detecting its behavior. Simulation results under realistic conditions prove the effectiveness of the proposed work concerning energy, time, and computational cost. Furthermore, we compare the proposed work with the currently available models and the results show that our proposed work outperforms the other solutions in terms of the above-mentioned performance parameters, leading to enhanced network performance. Therefore, the proposed solution provides a more efficient and effective method for detecting and preventing DDoS attacks in the IoT environment.

Keywords: Jamming; detection; prevention; DDoS

DOI: 10.48047/ecb/2023.12.si12.050

1 INTRODUCTION

The devices in the IoT have experienced explosive growth over the last few years, and the total number of devices of IoT is projected to increase by 25 billion by the year 2025 [1]. With the proliferation of these devices, there has been a corresponding increase in the number of security threats, including Distributed Denial-of-Service (DDoS) attacks one of the dangerous threats to network security by disrupting service availability to authorized users. DDoS attacks are particularly challenging to detect and prevent in the IoT environment. Most of the current works focused on speeding up the basic security functions such as confidentiality, integrity, and availability. However, the detection and prevention of jamming attacks such as DDoS is not efficiently done. As the number of IoT devices continues to grow, the threat of DDoS attacks is expected to become more severe, making it essential to develop more efficient and effective solutions to detect and prevent such attacks.

Motivated by these challenges, this work proposes a Threshold Based Countermeasure for replay attacks in the IoT environment. The proposed solution uses Unified Modeling Language (UML) based activity modeling to classify different variations of DDoS attacks and understand their behavior and performance in the environment. This knowledge is used to develop a Threshold Based Countermeasure for replay attacks, which allows the attack inside the network and blocks it after detecting its behavior. The proposed solution is evaluated through simulation under realistic conditions, and the results demonstrate its effectiveness in terms of energy consumption, time, and computational cost. Furthermore, we compare the proposed work with two latest solutions, namely by Farruh et al. [2], and Feroz et al.[3]. The outcome of the work proves that the proposed work is better than the currently available solutions in terms of the above-mentioned performance parameters, leading to enhanced network performance.

Therefore, the proposed solution provides a more efficient and effective method for detecting and preventing DDoS attacks in the IoT environment, which is critical in the face of the speedy growth of the devices in IoT and the corresponding increase in security threats.

2. Modeling of various attacks on IoT

In this section, we modeled various types of DDoS attacks using UML diagrams to demonstrate the functional aspects of the system and its logical operations. The flow of each activity represents the completion of an action and helps to understand the behavior of the attacks. The modeled attacks included constant attacks, replay attacks, random attacks, and deceptive attacks, each with unique characteristics.

Constant attacks violate the MAC policy by sending signals continuously to the channel, preventing legitimate traffic in and out of the network [4-8]. An attacker initiates a constant attack and makes a node a jammer. The infected node will start blocking the network, and if the attack is unsuccessful, the node operates normally without any malicious activity. If a non-malicious node has something to another node, it checks for the availability of the channel. The node sends the data when the channel is available; otherwise, it checks repeatedly after regular intervals. In contrast, the malicious node sends random data without checking the channel's availability, potentially jamming the entire traffic of the network if the time interval is low.

Replay attacks are difficult to detect because they activate only when other nodes are busy. The jammer node checks if the channel is busy and generates random data repeatedly to cause a collision in the network. If the channel is idle, the jammer node becomes silent and does not perform any malicious activity.

Random attacks are considered intelligent attacks that save energy through the jammer node, which can be executed in two states: jamming state and silent state. In the jamming state, the node repeatedly sends random data, violating the MAC policy, and the normal node receives data continuously until the jammer node goes into a sleeping state. In the sleeping state, the jammer node is silent and does not perform any malicious activity.

Deceptive attacks completely jam the channel after execution. An attacker initiates the attack, and upon successful execution, the node gets infected. If the attack is unsuccessful, the node operates its regular work. The regular node checks the channel's availability and sends data to other nodes if the channel is available. If the channel is not available, the node checks the channel after regular intervals until it is busy. The unique characteristic of this attack is that the jammer node sends packets continuously without leaving any time gaps, making the channel busy for a long time. The remaining sections are structured as follows: Section 2 discusses the modeling of various attacks that are commonly encountered in wireless communication systems. Section 3 provides a literature review that presents the current state of the art and identifies gaps in the existing research. Section 4 describes the proposed work, which is a novel algorithm that aims to improve network throughput and mitigate the impact of malicious attacks. Section 5 offers a detailed discussion of the results and their implications, including a comparison of the proposed algorithm with existing techniques. Finally, the work is concluded in section 6.

3 RELATED WORK

The authors in [9] proposed a distributed jammer detection mechanism for WSNs to counteract jamming attacks. The scheme consists of two parts: a node clustering algorithm and a jammer detection algorithm. The clustering algorithm groups nodes into clusters to reduce the effect of the jamming attacks. The jammer detection algorithm uses a neighbor agreement mechanism to detect the presence of jammers. The scheme increases the accuracy of the detection rate and decreases the false positive rates. However, the scheme suffers from increased communication overhead due to the node clustering algorithm. In [10], the authors proposed a game-theoretic approach for defending against jamming attacks in wireless networks. They formulated the problem as a non-cooperative game and developed a distributed algorithm to calculate the Nash equilibrium. The proposed approach can adaptively adjust the transmission power and channel selection to mitigate jamming attacks. The approach is effective in reducing the impact of jamming attacks and is scalable for large-scale networks. However, the approach requires a high computational complexity and assumes the availability of the channel state information.

The authors in [11] proposed a cooperative packet forwarding protocol to defend against node capture attacks and jamming attacks in wireless sensor networks. To verify the integrity of the packets and forward them to the destination, the work utilized only trusted nodes. The protocol can mitigate the impact of jamming attacks by re-routing the packets through alternative paths. The protocol achieves a high packet delivery ratio and low delay. However, the protocol suffers from increased communication overhead and reduced network lifetime due to the need for trusted nodes. In [12], the authors proposed a defense mechanism against jamming attacks in wireless sensor networks by combining frequency hopping and power control. The mechanism uses frequency hopping to avoid interference and power control to adjust the transmission power dynamically. The mechanism can mitigate the impact of jamming attacks and improve network performance. However, the mechanism requires synchronization among nodes and may suffer from increased communication overhead due to frequency hopping. Authors of [13] propose a joint anti-jamming mechanism called JAM-Free for IoT networks. JAM-Free combines adaptive frequency hopping (AFH) and interference-aware transmission power control (IATPC) techniques to overcome the limitations of existing anti-jamming mechanisms. The proposed JAM-Free mechanism dynamically adjusts the transmission power and hopping frequency based on the interference level in the network to maintain network connectivity and minimize the effects of jamming attacks. The proposed mechanism is evaluated using a Network simulator and compared with other anti-jamming mechanisms. The results show that the JAM-Free mechanism outperforms other mechanisms in energy, delay, and packet delivery ratio. Authors of [14] present a scheme for secure communication in wireless networks even under strong jamming attacks. But the drawback of this scheme is that it assumes perfect knowledge of the jamming signal, which may not be feasible in practice. In [15], the authors utilized a game theory-based solution to identify the interactions between jammers and anti-jammers and proposes a Stackelberg game for optimal anti-jamming strategies. Authors of [16] present an adaptive OFDM scheme that can adjust to varying jamming conditions and evaluate its performance under different types of jamming attacks. But the model is limited to OFDM-based systems, and may not be directly applicable to other wireless networks. In [17], the author uses cognitive radio technology to detect and mitigate jamming attacks and proposes an intelligent anti-jamming scheme that can adapt to changing jamming conditions. However, it requires specialized hardware for cognitive radio functionality which leads to performance degradation.

After analyzing the literature we discussed here, it can be concluded that various techniques have been proposed to mitigate the impact of jamming attacks in wireless networks. These techniques include frequency hopping, power control, channel surfing, and spread spectrum modulation. While these methods have shown to be effective to some extent, they have their limitations and can be circumvented by advanced attackers.

Among the discussed methods, the proposed threshold-based measure in our study appears to be a robust scheme as it uses a combination of physical layer characteristics and statistical methods to detect jamming attacks. By setting appropriate thresholds, our proposed scheme can detect even low-power jamming attacks and reduce false positives. Additionally, the proposed scheme is independent of the jamming technique used and can be applied to a wide range of wireless networks.

4. The proposed method for jamming attacks

This paper proposes the Threshold Based Countermeasure (TBC) to enhance the performance of the Internet of Things (IoT) against replay attacks. After studying various classifications of jamming attacks like constant, random, replay, and deceptive attacks, it was found that the replay attack was the most destructive form of attack and had the greatest impact on network performance. The proposed protocol work in two phases. In the first phase, the base station (BS) decides on the threshold value for all the nodes in the environment. The threshold values are computed based on the average amount of data each node sends to the BS. The second phase involves maintaining three states for each node: regular, doubtful, and attack. In the regular state, nodes perform their regular activities without any malicious behavior. In the doubtful state, a node may be suspected of being a malicious attacker, while in the attack state, the node is confirmed to be an attacker and actively causing harm to the network. If a node tries to push more data than its assigned threshold value, the BS will mark it as a doubtful node and initiate a path analysis. If the malicious node is within one hop distance, it can be easily detected. If it is located at a multi-hop distance, the algorithm will analyze the sending threshold values of all nodes in the path. If any of the threshold values exceed the average, the node will be marked as an attacker

and removed from the network. This information is then shared with all neighboring nodes. The TBC algorithm effectively detects replay attacks by using the sending threshold values maintained at each node in the network. The outcome of the work shows that the proposed algorithm outperforms previous works in terms of energy efficiency, traffic rate, delay, and computational cost.

4.1 The game theory-based detection mechanism

This section discusses the game theory-based approach to overcome the security issues related to jamming attacks. There are two kinds of attacks possible in the network environment active attack and passive attack.

Active attacks can cause physical damage to the system to disrupt the network services and passive attacks can be implemented to block the communication from the outside world. The important challenges in IoT security are to preserve energy conservation, scalability, key management, and minimizing time delay. Jamming attacks block the network for a long time by making the resources unavailable continuously which can reduce the overall performance of the network. This jamming attack mainly occurs in two layers: The physical layer and the MAC layer.

Bargaining games, repeated games, and coalition game approaches have belonged to cooperative games. Zero-sum, non-zero-sum, Stackelberg, jamming approach, stochastic game approach, Bayesian approach, and Evolutionary game approach are non-cooperative game-based approaches.

If the request for resources is greater than the available then the request is rejected. In the repeated game approach the players are repeatedly playing the game either finite or infinite number of times. The former has a fixed number of stages and the latter will have an infinite number of stages for playing the game. In the coalition approach, there is cooperation among a set of players to maximize the mutual outcome based on coalition value. Zero-sum is a non-cooperative game where one player will maximize the gain and the other play will minimize the losses. In a non-zero game approach, two or more players are involved and all the players are striving to maximize the gains and minimize the losses. There are no constraints here as in zero-sum. Stackelberg involves two players, one player will play the defender role and the others will play the attacker role. Jamming games are played between defenders to protect the security of wireless sensor networks. Stochastic games are played based on the probabilistic transitions of the players in each stage. The Bayesian game approach is used in IDS to analyze the interactions between the players. The various jamming attacks that can be classified in the network are constant, deceptive, random, and reactive attacks. There are different game theory-based approaches for the security of the sensor nodes that we discussed above. All these approaches are introduced to combat both internal attacks and external attacks on the network. External attackers can use the vulnerable node in the network to execute their attacks to degrade the performance of the network. Attackers will send the error packet to the network to harm the environment. They will prevent the regular nodes to send data to other nodes in the destination. In case of internal attacks, the infected node will not forward the regular packet to other nodes it will just drop the packets. The role of all types of jamming game approaches is given in Tables 1, 2, & 3 which helps to learn the behavior of the jamming game approach.

Table 1: Constant jamming game approach

Players	**Sensor nodes**	**Constant jamming**
Role	The role of the player is to send data if the channel is idle	Constantly sends packets after a regular time gap by violating the MAC policy.
Effect	Prevent jamming inside the network. All the data are sent successfully without collision	Network's throughput is reduced with increased energy and collision.
Impact	Regular energy consumption is required to sense the channel before sending the data and to transmit the data after finding the channel is free. Extra energy is needed if retransmission is done due to collision or jamming.	Energy consumption is increased for the generation of noise packets after regular intervals of time

Table 2: Deceptive jamming game approach

Players	**Sensor nodes**	**Deceptive jamming**
Role	The role of the player is to send data if the channel is idle	Sends the data repeatedly without sensing the channel for its availability.
Effect	Prevent jamming inside the network. All the data are sent successfully without collision	Keeps all the nodes in receiving state which produces jamming in the network. The nodes cannot send the data because the channel is always busy in receiving state.
Impact	Regular energy consumption is required to sense the channel before sending the data and to transmit the data after finding the channel is free. Extra energy is needed if retransmission is done due to collision or jamming.	Due to the continuous generation of random packets, the energy consumption of the network is increased.

Table 3: Random jamming game approach

Players	**Sensor nodes**	**Random jamming**
Role	The role of the player is to send data if the channel is idle	Sends the data repeatedly without sensing the channel for its availability or Constantly sends packets after a regular time gap by violating the MAC policy. Later it will go to sleep mode
Effect	Prevent jamming inside the network. All the data are sent successfully without collision	Keeps all the nodes in receiving state most of the time which produces jamming in the network. The nodes cannot send the data whenever the channel is busy and in receiving state.
Impact	Regular energy consumption is required to sense the channel before sending the data and to transmit the data after finding the channel is free. Extra energy is needed if retransmission is done due to collision or jamming.	The intelligent behavior of the DDoS attack, such as keeping the nodes in a sleeping state, can lead to increased energy consumption in the network

4.2. Threshold Based Countermeasures (TBC)

The evaluation of our proposed algorithm TBC involved the use of a sending threshold value in each node of the network to detect misbehavior and treat replay attacks. Our results demonstrate that TBC is effective in mitigating replay attacks, with its performance measured in terms of energy consumption, delay, computational cost, and the number of malicious nodes present.

While other related works are successful in detecting jamming attacks, they often require the combination of multiple detection schemes, which can result in additional overhead for resource-constrained IoT devices. The TBC algorithm, on the other hand, enhances network performance in the presence of replay attacks by maintaining a sending threshold value at each node.

The internal process of TBC is described in the algorithm. The key feature of TBC is its ability to detect jamming through the use of the sending threshold value. When a malicious node is detected, the algorithm performs a path analysis. If the malicious node is nearby, the attack can be easily detected. In the case of multi-hop distance, the algorithm analyzes the sending threshold values of all nodes connected in the path.

Our results indicate that TBC outperforms other related works in terms of energy efficiency, traffic rate, delay, and computational cost.

The threshold values are computed using the following formula for Node A:

$$N_A = \frac{Nr}{Ns} \quad (1)$$

To determine the trustworthiness of a node, the proposed method utilized trusted values, which are verified using trusted route metrics.

Trustworthiness score (T_i): It is calculated based on the number of successful and failed interactions of node i with other nodes. A successful interaction increases the trustworthiness score, while a failed interaction decreases it. The trustworthiness score can be computed as follows:

$T\_i = (S\_i - F\_i) / (S\_i + F\_i)$

where S_i is the number of successful interactions of node i, and F_i is the number of failed interactions.

Direct trust (DT_ij): It is the trustworthiness score of node j as perceived by node i based on their direct interactions. The following equation computes the direct trust value.

$DT\_ij = T\_j$

Indirect trust (IT_ij): It is the trustworthiness score of node j as perceived by node i based on the recommendations of other nodes in the network. The indirect trust can be calculated using the following equation:

$IT\_ij = (1 - w) * T\_j + w * (SUM\_k(T\_k * R\_kj) / SUM\_k(R\_kj))$

where w is a weighting factor that determines the importance of the recommendations, R_kj is the recommendation of node k for node j, and the sum is taken over all nodes k that have recommended node j.

These are just some examples of equations used for calculating trusted metrics in a WSN, and many other variations and approaches can be used. The above equation helps to find a malicious node by calculating its Trusted Value (TV). The TV is based on several metrics, such as packet forwarding ratio, residual energy, and hop count. If a node's TV falls below a certain threshold, it is considered untrusted, and its packets are not forwarded to the network. When a node's TV falls below the threshold, it is moved to the "suspicious" state. In this state, the node is monitored to see if it continues to behave maliciously. If the node's behavior improves and its TV rises above the threshold, it is moved back to the "normal" state. However, if the node's behavior remains suspicious or worsens, it is moved to the "attack" state and is removed from the network

// ALGORITHM: TBC ROUTE CHANGE ALGORITHM
// CLUSH: Cluster Head, BS: Base Station
// Input: data sending threshold value and fixed variables
// Output: route change notification to CLUSH or BS
1. Start
2. Set the threshold value to transfer data
   Level 1: CLUSH, Level 2: BS
      // CLUSH and BS check data and the threshold value for each node after particular intervals
3. If data sending value > threshold
   For each node on the route:
    If data sending value > threshold
      Declare the node as a blocking node inside and outside the clusters
4. Notify CLUSH or BS and update the route information
5. Stop

5 RESULTS DISCUSSION

The implementation of the work is performed by utilizing the network simulator NS-2. The parameters used in the NS-2 are given in Table 4.5. The performance of the algorithm is measured in terms of energy, and time delay by considering time intervals and increasing the adversary nodes in the network. First, evaluate the performance of all the jamming attacks are evaluated with no attack situation. This section discusses the performance of the network under all the attacks by considering traffic intervals.

Table 4: Simulation Parameters

Parameter	**Value**
Area	1000m x 1000m
Number of nodes	100
Network topology	Random
Proximity	100m
Data packet size	500 bytes
Jamming power	1Watt
Jamming technique	Replay
Modulation scheme	QPSK
Carrier frequency	2.4 GHz
Bit error rate (BER)	10^-6
Jammer type	Reactive and Proactive

There are two sets of simulations we have done for the evaluation of the work. The initial set of simulations assesses the network's performance using three key parameters: energy consumption, delay, and throughput. These parameters are measured while varying the traffic interval. The second set of simulations evaluates the network's performance for energy, delay, and throughput by increasing the attacker nodes.

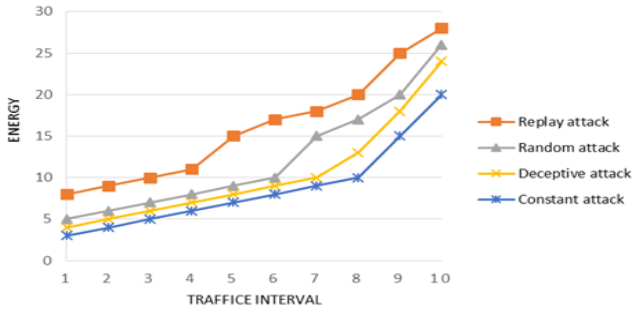

Figure 1: Energy consumption of the network in time intervals

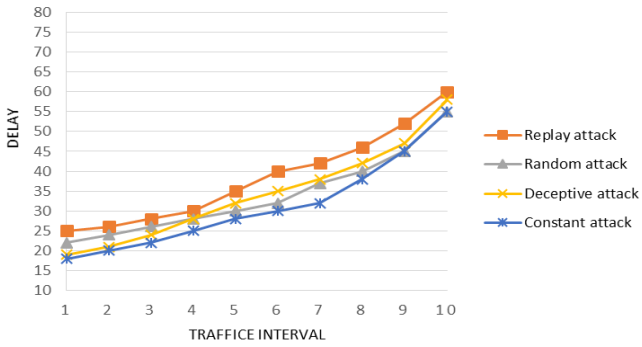

Figure 2: Analysis of delay in time intervals

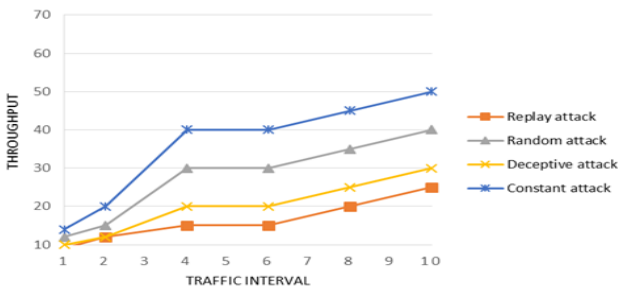

Figure 3: Throughput of the network in time intervals

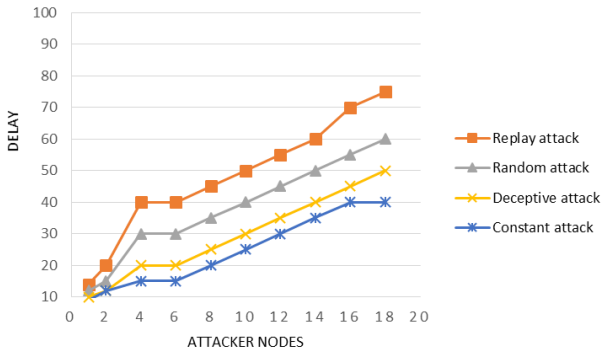

Figure 4: Analysis of delay with attacker nodes

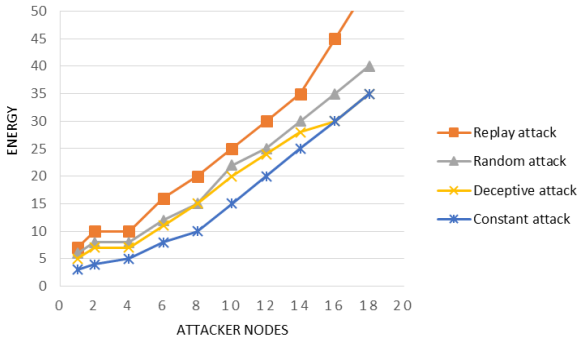

Figure 5: Analysis of energy with attacker nodes

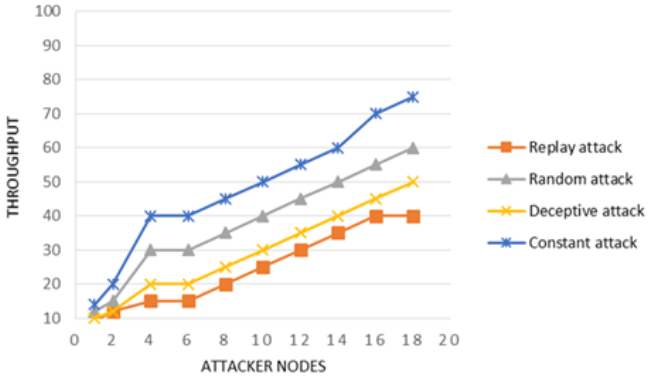

Figure 6: Analysis of throughput with attacker nodes

From the above performance analysis, we observed that the network consumes less energy under no attack situation. The energy of the network is increased during attack situations. Among all the attacks, the network consumes more energy under replay attack than all the other attacks during the evaluation of the work by considering traffic interval and attacker nodes. The replay attack also increases the delay of the network more than all the attacks and the throughput of the network is highly reduced under replay attack than other attacks. Hence, we consider replay attacks for the assessment of our work.

5.1 Performance measurement in terms of energy, delay, and throughput

Figure 1 shows the consumption of energy by changing the time interval of data. The consumption of energy is compared during the execution of the replay attack and after implementing the algorithm under the replay attack.

The result shows that the energy is less after the TBC is implemented in the environment. Figure 2 reveals the performance of TBC in terms of delay. It reduces the time delay under replay attacks. The reason for less delay than in an attack situation is that the algorithm detects the jamming node and removes it from the network. Since the channel is available without jamming issues, the algorithm helps

in reducing the time delay. Figure 3 shows the performance of TBC in network throughput. The algorithm enhances the throughput of the network by removing the jamming problem which makes the network unavailable for a long time and each node has to wait until the channel becomes available. Fig.4, 5, & 6 show the performance of TBC with increased malicious nodes in the network. The number of attacker nodes added in the evaluation is 1 to 16. The result shows that the algorithm efficiently detects the jammer nodes even after the attacker nodes are increased in the network. The performance of the algorithm under increased nodes is measured in terms of energy, delay, and throughput.

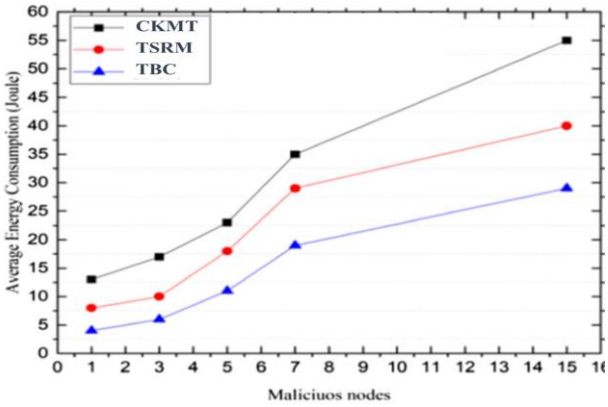

Figure 7: Performance in terms of energy under malicious nodes

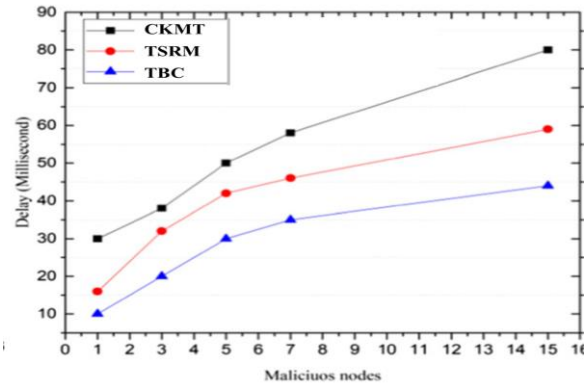

Figure 8: Performance under increased attacker nodes in terms of delay

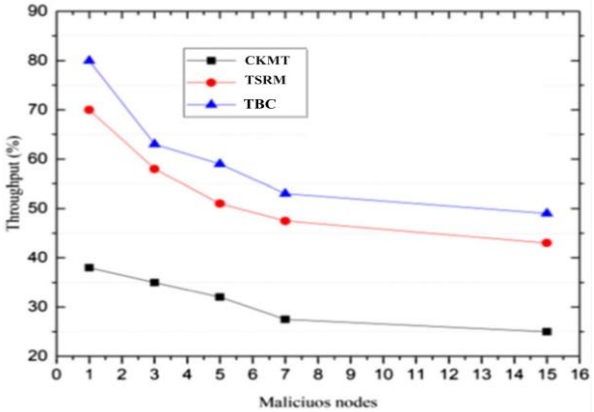

Figure 9: Performance under increased attacker nodes in terms of throughput.

As the time interval increases in all approaches, the energy consumption of the network also increases. Among all the approaches, CKMT consumes the highest amount of energy since it only transmits signals when it senses data sent by the regular node. Fig. 7 illustrates the consumption of energy by differing the numbers of malicious nodes, set to 1, 3, 5, 7, and 15, for the evaluation of the algorithm during the increases of jamming. Results indicate that energy consumption levels increase when

malicious nodes are present. The energy consumption of the network is significantly reduced by the proposed algorithm due to its efficient detection mechanism. It avoids unnecessary energy consumption of the jamming node by not sending unwanted data. The average energy consumption at nodes 1 and 15 decreases significantly, by approximately 4% and 29%, respectively, when the proposed algorithm is implemented under realistic conditions. Furthermore, the network performance also shows improvement. Under mobility and with an increase in the number of malicious nodes, the proposed algorithm reduces energy consumption to about 3% and 23% at nodes 1 and 15, respectively, thereby further improving the network performance.

The proposed algorithm is capable of detecting, stopping, and isolating or removing the jamming node from the network, which indirectly eliminates traffic congestion in the channel and reduces delays. In a typical reactive jamming scenario, the proposed algorithm results in a nearly 9% reduction in traffic delay, which is 3% better than the existing algorithm. However, the percentage continues to increase proportionally to an increase in the interval. Figure 8 illustrates the performance of TBC with traffic delay under a varying number of malicious nodes. The proposed algorithm reduces traffic delay to approximately 8% and 38% at malicious nodes 1 and 15, respectively, compared to the existing approach with approximately 12% and 50% traffic delay at the same number of malicious nodes. Overall, the results from Figure 9 demonstrate that the proposed technique performs well than the traditional technique in terms of network throughput under various conditions and with different numbers of malicious nodes.

6 CONCLUSION

This paper proposes a threshold-based countermeasure for replay attacks. The UML-based activity modeling approach was used to model all variations of DDoS attacks, and the proposed solution was evaluated through simulations under realistic conditions. The results prove the effectiveness of the introduced protocol concerning energy consumption, time, and computational cost compared to the three latest solutions. Our proposed solution provides a more efficient and effective method for detecting and preventing DDoS attacks in the IoT environment, leading to enhanced network performance. This work contributes towards the development of more secure and resilient IoT networks, which are becoming increasingly important in the era of digital transformation. Future improvements for this work could include exploring additional types of DDoS attacks and modeling them to further improve the proposed countermeasures. Additionally, enhancing the proposed protocol with machine learning algorithms and developing a more comprehensive intrusion detection system could improve its efficiency and accuracy. Moreover, exploring the potential of blockchain technology in DDoS mitigation could be an interesting area of research. Finally, testing the proposed solution in a real-world IoT environment and comparing its performance with existing commercial solutions would be a valuable contribution to the field

REFERENCES

1. H. Li, Y. Li, and W. Zhang, “An anti-jamming algorithm based on multi-objective optimization in wireless sensor network,” in Proceedings of the 2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications, New York, USA, Oct. 2015, pp. 417-422.
2. Farruh Ishmanov, Yousaf Bin Zikria, Trust mechanisms to secure routing in wireless sensor networks: current state of the research and open research issues”, Hindawi, J. Sensors Volume (2017), 472485216.
3. Feroz Khan, A.B., Anandharaj, G. A cognitive energy efficient and trusted routing model for the security of wireless sensor networks: CKMT. Wireless Pers. Commun. (2021).
4. Zhang, Y., Liu, C., Liu, A., & Xie, S. (2018). A Deep Learning-Based DDoS Detection System for Internet of Things. IEEE Access, 6, 36192-36201. doi: 10.1109/access.2018.2845798
5. Zhao, X., Li, L., & Wu, Z. (2019). IoTEdgeIDS: An intelligent edge-based intrusion detection system for IoT networks. Future Generation Computer Systems, 90, 123-134. doi: 10.1016/j.future.2018.08.032

6. Yu, Z., Yang, X., Li, F., & Wu, X. (2019). A Blockchain-Based Detection and Mitigation Mechanism for DDoS Attacks in IoT. IEEE Access, 7, 10783-10790. doi: 10.1109/access.2019.2894613
7. C. Zhang, J. Zhang, H. Xiong, S. Wang, and Z. Shi, "A Novel Anti-jamming Scheme for Multihop Wireless Sensor Networks," in Proceedings of the 2016 IEEE International Conference on Communications, Kuala Lumpur, Malaysia, May 2016, pp. 1-6.
8. K. Li, K. Ren, and W. Lou, "Secure and Efficient Data Transmission for Cluster-Based Wireless Sensor Networks," IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 8, pp. 1010-1021, Aug. 2007.
9. N. N. Qureshi and H. A. Khan, "Survey on jamming attacks and countermeasures in wireless sensor networks," Journal of Network and Computer Applications, vol. 36, no. 2, pp. 826-849, 2013.
10. S. S. Sharad, S. A. Mhetre, and S. M. Jagdale, "Jamming attacks and countermeasures in wireless sensor networks: a survey," Wireless Personal Communications, vol. 92, no. 1, pp. 135-158, 2017.
11. M. N. Karim, N. Islam, M. A. Rahman, and H. Song, "A survey on jamming attacks and countermeasures in wireless sensor networks," International Journal of Distributed Sensor Networks, vol. 14, no. 3, 2018.
12. R. Chen, C. Tseng, H. Chao, and K. Lo, "Defending Against Jamming Attacks with a Distributed Jammer Detection Scheme in WSNs," IEEE Transactions on Information Forensics and Security, vol. 13, no. 4, pp. 855-868, April 2018.
13. S. Liu, J. Wu, X. Yang, and Q. Xie, "A Game-Theoretic Approach for Defense Against Jamming Attacks in Wireless Networks," IEEE Transactions on Information Forensics and Security, vol. 11, no. 10, pp. 2229-2242, Oct. 2016
14. M. Younis and K. Akkaya, "Strategies and techniques for combating node capture attacks in wireless sensor networks," Computer Communications, vol. 30, no. 11-12, pp. 2430-2447, July 2007.
15. D. Chen, L. Qiao, and S. Zhu, "Defense against Jamming Attacks in Wireless Sensor Networks Based on the Combination of Frequency Hopping and Power Control," International Journal of Distributed Sensor Networks, vol. 8, no. 6, pp. 1-12, June 2012.
16. A. Zaidi, J. J. Rodrigues, and I. Aldmour, "JAM-Free: A Joint Anti-Jamming Mechanism for Internet of Things," IEEE Internet of Things Journal, vol. 6, no. 5, pp. 8287-8300, Oct. 2019
17. M. Mozaffari-Kermani, C. Lu, and R. D. Wesel, "Secure Communication in Wireless Networks Under Jamming," IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4336-4345, Oct. 2006.
18. C. Rong, X. Liu, and Y. Zhang, "Collaborative Jamming and Anti-Jamming in Wireless Networks: A Stackelberg Game Perspective," IEEE Transactions on Mobile Computing, vol. 14, no. 2, pp. 352-365, Feb. 2015.
19. M. A. Al-Qutayri and E. M. A. El-Medany, "Performance of Adaptive OFDM Systems Under Jamming," IEEE Transactions on Wireless Communications, vol. 5, no. 11, pp. 3246-3255, Nov. 2006.
20. J. Zhao, Z. Wu, Y. Zhang, and Y. Liu, "An Intelligent Anti-Jamming Scheme Based on Cognitive Radio Networks," IEEE Transactions on Vehicular Technology, vol. 60, no. 1, pp. 56-67, Jan. 2011
21. C. Rong, X. Liu, and Y. Zhang, "Collaborative Jamming and Anti-Jamming with Power Constraint in Wireless Networks," IEEE Transactions on Wireless Communications, vol. 13, no. 6, pp. 3050-3061, Jun. 2014.
22. J. Liu, L. Zhang, J. Zhang, Y. Zhou, and X. Wang, "A Dual-Antenna Scheme for Jamming Attack Detection in Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 10, pp. 2775-2784, Oct. 2017.
23. S. Ni and S. Xu, "Jamming attack detection and defense in wireless sensor networks based on compressed sensing," in Proceedings of the 2014 IEEE International Conference on Communication Problem-Solving, Harbin, China, Aug. 2014, pp. 141-145.
24. R. F. Alsharif, A. Alhasanat, and H. Shibli, "An intelligent jamming detection and mitigation system for wireless sensor networks," Sensors, vol. 20, no. 17, 4795, Aug. 2020.