Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4335871, 8 pages
https://doi.org/10.1155/2022/4335871

Research Article

Hierarchical and On-Demand Attack Defence Framework for IoT Devices

Pradeep Sudhakaran,[1] **Manikannan Kaliyaperumal**,[2] **T. Senthilkumar**,[1] **R. Jeya**,[1] **and B. Sowmiya**[1]

[1]*Department of Computing Technologies, School of Computing, SRM Institute of Science and Technology, Potheri, Kattankulathur, 603203 Chengalpattu District, Tamil Nadu, India*
[2]*Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology and Management, Dundigal, 500043 Telangana, India*

Correspondence should be addressed to Pradeep Sudhakaran; pradeeps1@srmist.edu.in

Received 3 March 2022; Revised 25 March 2022; Accepted 31 March 2022; Published 26 April 2022

Academic Editor: Kalidoss Rajakani

Internet of Things (IoT) devices are lightweight such that they generally possess low battery power. Hence, the chances of battery exhaustion and flooding attacks are more. In order to perform attack response actions against various attacks, this paper proposes Hierarchical and On-Demand Attack Defence Framework (HOAD) for IoT security. In this framework, primary (PC) and secondary controller (SC) nodes are deployed in the network along with the IoT devices. The SC scan be moved on-demand by the PC. The response agent at PC will first establish a new route via the SCs by excluding the intruders and the suspected nodes. Then, it will resend the stored packets to their destination via the newly established route. The proposed HOAD framework is implemented in NS2 and compared with the MECshield framework. Simulation results show that the HOAD has reduced end-to-end delay, increased packet delivery ratio, and increased residual energy.

1. Introduction

IoT is considered as the third industrial revolution. It is defined as "the interconnection, via the Internet, of computing devices embedded in everyday objects, enabling them to send and receive data." IoT devices are capable of gathering information from specific region at specific time intervals. IoT is useful in various applications such as smart homes, education, and healthcare [1].

IoT networks face many challenges with respect to connectivity, computing, and security. Since IoT devices possess low battery power, the chances of battery exhaustion and flooding attacks are more [2].

The complexity in the nature of IoT security rotates around the reality that, since, it is a great challenge to combine several technologies into one; the system tries to connect devices securely which have limited computation capability, storage, and power. Few of the devices utilized by IoT can hold only a little basic mechanism of security measures, some of which are not capable to maintain the confidentiality and integrity of the users' information data.

There are three primary entities which poses threats to the privacy and security in IoT: dishonest users, bad manufacturer, and outside attackers. Various-type attacks targeted towards IoT devices include tampering of device, information revealing, denial of service (DoS), and spoofing [3].

Owing to their resource limitations and heterogeneous nature, conventional security solutions may not be applicable for IoT systems. Hence, there is a need for developing alternate solutions for defending against attacks in IoT networks [4].

The defensive techniques can be useful to develop the efficient model for securing the Internet of Things (IoT). The effort for developing defensive techniques will be easier and efficient once we understand the behaviour of the attacks completely [5].

1.1. Problem Identification. In our previous work, an authorization, attack detection, and avoidance (AAA) framework

Table 1: IN_RES message.

PC ID	SC ID	Packet drop ratio	Delay	Intruders ID	Detection time

for IoT devices has been developed. The detection agent checks the collected traffic information against attack rule table. If any matching attack pattern is found, it informs the attack type to response agent. Once the response agent obtains the attack type from detection agent, then it estimates the severity of attack by computing the attack frequency over different time windows, and appropriate action will be performed.

In order to perform attack response actions against various attacks, we propose a hierarchical self-healing framework for IoT security, as an extension work.

In this framework, whenever the RA receives the intrusion confirmation message from the detection agent, it triggers the response action, by broadcasting the reroute information to the PC. Then, PC establishes a new route via the SCs by excluding the intruders and the suspected nodes. After that, it will resend the deposited packages to their terminus through the freshly reputable way.

2. Related Works

A localized DDoS prevention framework known as MEC-Sheild has been developed [2].

Nhu-Ngoc et al. [2] have proposed MECshield, a restricted DDoS avoidance outline leveraging MEC power to set up numerous shrewd sieves at the verge of related attack-source/terminus systems. The support amongst the shrewd sieves is overseen by a dominant regulator. The dominant regulator confines every shrewd sieve by serving suitable teaching factors into its self-organizing map (SOM) module, centred on the offensive conduct. The presentation of the MEC defence outline is tested using three typical IoT traffic scenarios.

Daz López et al. [4] have proposed a safety way out centred on the administration of safety activities inside IoT situations so as to precisely recognize doubtful actions. To this conclusion, diverse susceptibilities discovered in IoT strategies are defined, along with exclusive structures that make these strategies an alluring objective for outbreaks. Lastly, three IoT outbreak situations are offered, defining oppressed susceptibilities, safety activities produced by the outbreak, and precise reactions that could be propelled to support lessening the influence of the outbreak on IoT strategies.

Ketan et al. [6] have proposed a novel method that influences verge figuring to set up verge operations that collect info about inbound congestion and transfer that info through a fast-path with a close discovery facility. This quickens the discovery and the capture of such outbreaks, restraining their destructive effect. Initial examination displays assurance for up to 10x quicker discovery that decreases up to 82% of the Internet congestion because of IoT-DDoS.

Ali et al. [7] have proposed new discovery methods or refining prevailing ones, but there is a scarcity of awareness about the recent sorts of Sybil outbreaks and their counter measures. The determination of their article is to discover the diverse sorts of Sybil outbreaks and possible countermeasures.

```
    Then
            PC starts fault detection using the received
information
            (shown in Table 1)
            Repair the damage caused attacks.
    Else
    PC will perform the recovery
    Stored packets will be deleted
    End if
```

Algorithm 1.

Vincentius et al. [8] have proposed a wide-ranging home system protection, Pot2DPI, and utilise it to increase an assailant's improbability about strategies and allow the home system to observe congestion, sense irregularities, and sieve spiteful packages. The safety presented by Pot2DPI arises from a combination of applied methods: honeypot, deep packet inspection (DPI), and a understanding of moving target defense (MTD) in port forwarding. In specific, Pot2DPI has a series of honeypot and DPI that gathers doubtful package suggestions, obtains outbreak initials, and connects sifting instructions at a home router timely. In the meantime, Pot2DPI scuffles the plotting of ports amid the router and the strategies associated to it, creating a besieged outbreak hard and protection more real.

3. Proposed Solution

3.1. System Model. Here, S represents the sink node. PC and SC represent the primary and secondary controllers. Z1-Z8 represent the IoT devices.

In this framework, primary (PC) and secondary controller (SC) nodes are deployed in the network along with the IoT device. The SCs can be moved on-demand by the PC [9].

It is assumed that the response agent (RA) resides at the PC. The SCs are connected to a set of IoT devices as well as with each other. The PC will have the accurate location information of each node and SCs at the time of deployment. When a SC becomes mobile, it will update its network topology information.

3.2. Overview. In this paper, we propose a hierarchical self-healing framework for IoT security. Whenever the RA receives the intrusion confirmation message from the detection agent, it triggers the response action, by broadcasting the reroute information to the PC. Then, PC establishes a new route via the SCs by excluding the intruders and the suspected nodes. After that, it will resend the deposited packages to their terminus through the recently recognised way.

3.3. Attack Recovery Procedure. The steps involved in this process are as follows:

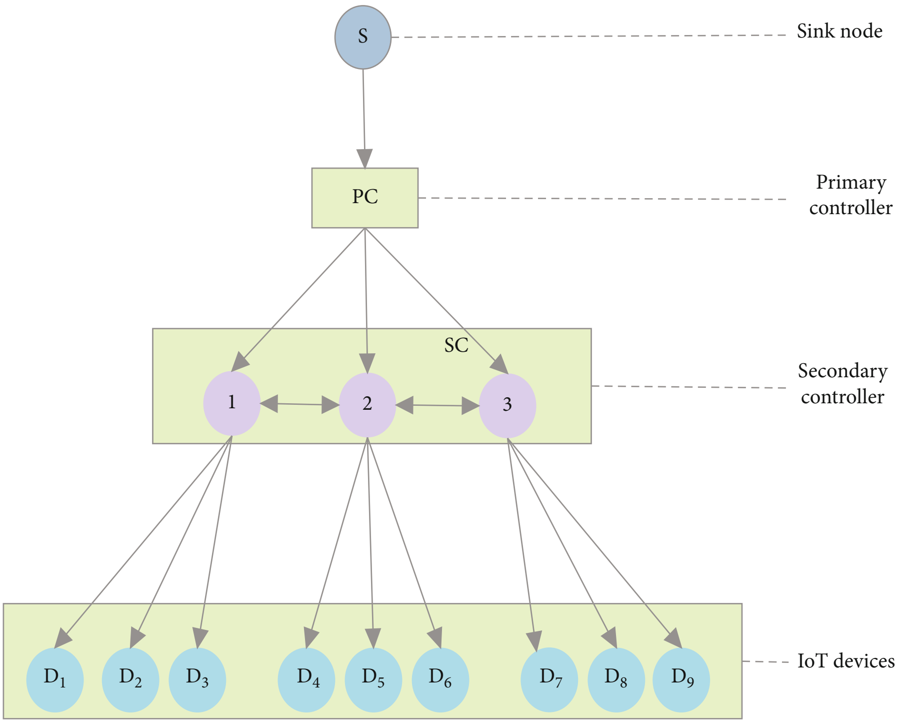

Figure 1: System model.

(1) RA broadcasts intrusion information message (IN_MES) to SCs

(2) Each SC upon receiving IN_MES responds with the response message IN_RES that includes the following details in Table 1.

(3) Based on the information received, PC will measure the damage caused by the intruders and performs the following processes:

(i) PC stores the copy of each sent packet during each detection interval

(ii) Checks whether any messages about anomalous events are received

(iii) If it is received

(4) A new route is determined bypassing the suspected nodes in the current route [10, 11]

(5) The buffered packets are retransmitted through this new route to the destination

3.4. Backup Route Establishment. During route discovery phase, a secondary route is determined. When a RREP is received in the main path, a secondary path is formed by broadcasting another RREQ packet with the backup flag set as TRUE.

When a standby RREQ is transmitted, the dependability track is utilised to decide the finest suitability standby path also. If a nodule on the major path that has only one standby path info expected a standby RREQ, the dependability track from base to terminus in the standby RREQ is initially taken into consideration.

If the dependability of the track along its principal standby track from itself to the terminus is better than terminus of the standby RREQ, it removes the acknowledged standby RREQ noiselessly to stop generating a very little dependability standby track for the demanding nodules. Else, a standby RREP package is engendered and unicasted again to the standby path demanding nodule. When a nodule which is on the principal track acknowledged a standby RREQ, it removes RREQ message and avert a dismissal message. If a standby RREQ with subsequent step along the principal track is acknowledged by the terminus, then it is noiselessly rejected so as to avert a creation of unusable standby track coinciding with the principal track; else, a standby RREP or appeal to standby path via the equivalent track is engendered. When the recovery process is initiated, the main route is changed to the secondary route, and the packets are delivered to destination. Figure 1 shows the primary and backup route setup process, whereas Figure 2 shows the backup route switching process.

As exposed in Figure 3, once the base obtains the principal RREP, the path of data transfer from base to terminus ($S\longrightarrow 4\longrightarrow 3\longrightarrow D$, $S\longrightarrow A\longrightarrow B\longrightarrow C\longrightarrow D$, $S\longrightarrow 1\longrightarrow 2\longrightarrow D$) is recognized and then receipts the dependability track as choosing principal path technique. If the path $S\longrightarrow A\longrightarrow B\longrightarrow C\longrightarrow D$ is the greatest (43 $R > R$ SABCD s D, 12 $R > R$ SABCD s D) dependability, and then, the path is the b principal path. In principal path, every nodule needs to take standby path, e.g.,

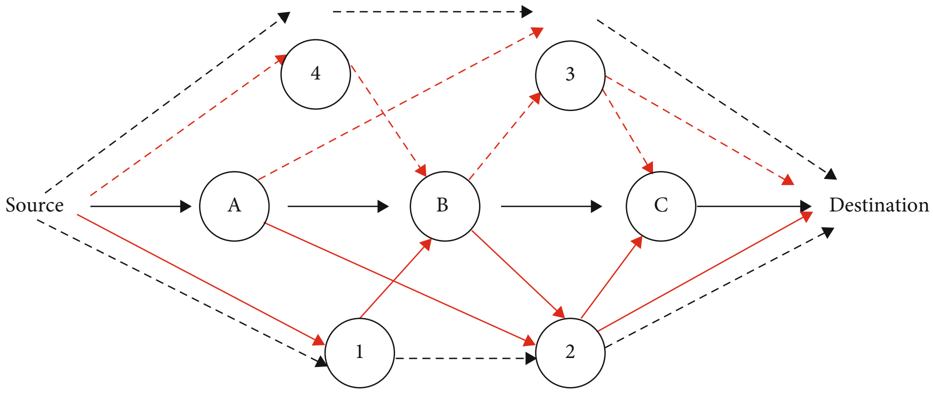

FIGURE 2: Primary and backup route establishment.

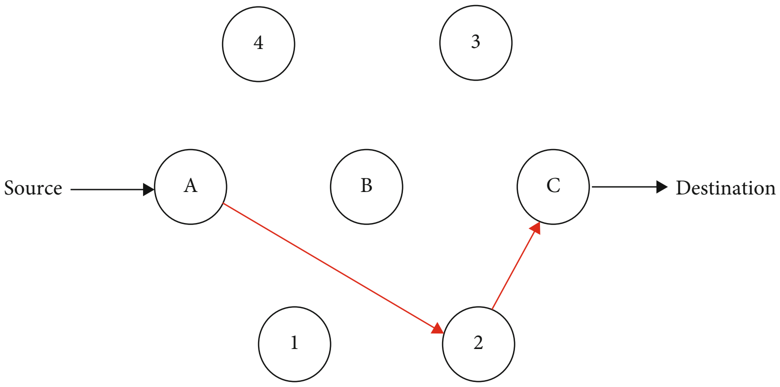

FIGURE 3: Backup route selected when node B is suspected.

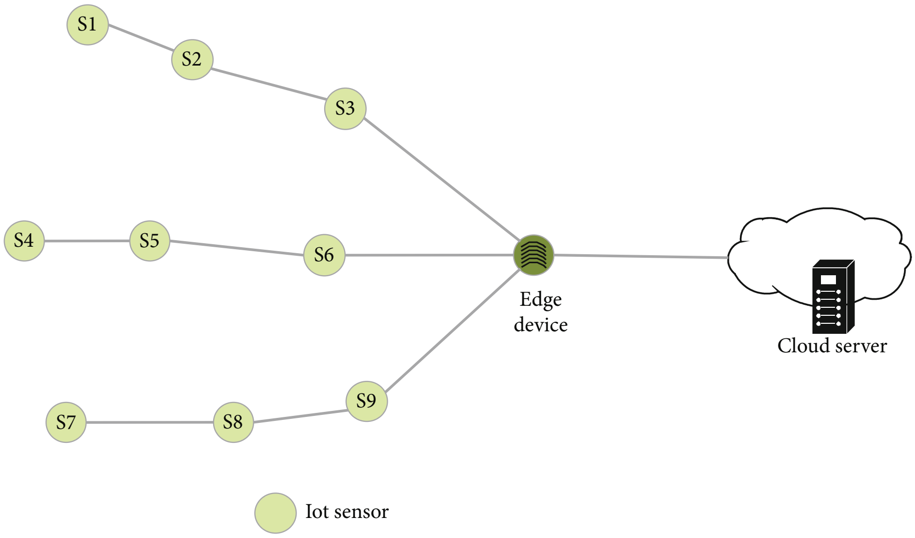

FIGURE 4: Processing data transmission with SC devices.

standby path for nodule A is $S \rightarrow 1 \rightarrow B$, for nodule B is $S \longrightarrow A \longrightarrow 2 \longrightarrow C$, and for nodule C is $S \rightarrow A \rightarrow B \rightarrow 2 \rightarrow D$ are also formed. If the nodule B is sensed as an assailant, the principal track immediately shifts to the standby path as exposed in Figure 1.

On establishing the route, devices that are deployed over the network plane are responsible for reporting to the edge device, which is further connected to the cloud server for processing. The process is given in Figure 4.

4. Experimental Results

4.1. Experimental Settings. The proposed Hierarchical and On-Demand Attack Defence Framework (HOAD) is

Table 2: Experimental settings.

Network size	12
Size of topology	150 × 150 m
MAC protocol	802.15.4
Monitoring interval	20, 40, 60, 80, and 100 sec
Attack frequency	50 to 150 Kb
Packet size	512 bytes
Propagation model	Two ray ground
Antenna model	Omni antenna
Initial energy	12.0 joules
Transmission power	0.8 watts
Receiving power	0.3 watts

Table 3: Result table of E2D for time scenario.

Monitoring interval (sec)	HOAD	MECSheild
20	0.006184	0.008052
40	0.006082	0.008052
60	0.006053	0.008054
80	0.006039	0.008054
100	0.006031	0.008054

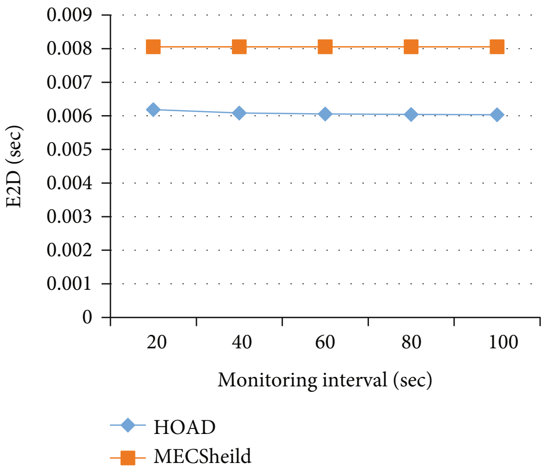

Figure 5: E2D vs. monitoring interval.

Table 4: Result table of PDR for time scenario.

Monitoring interval (sec)	HOAD	MECSheild
20	0.999583	0.99659
40	0.999815	0.99798
60	0.999881	0.99868
80	0.999912	0.99901
100	0.999931	0.99921

implemented in NS2 and compared with the MECshield [2] framework. MECshield is a localized DDoS prevention framework utilizing mobile edge computing (MEC) power

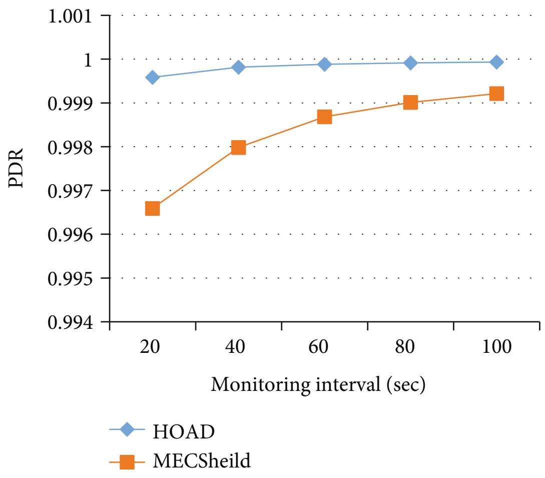

Figure 6: PDR vs. monitoring interval.

Table 5: Result table of throughput for time scenario.

Monitoring interval (sec)	HOAD	MECSheild
20	9599	4871
40	21599	10000
60	33599	15126
80	45599	20255
100	57599	25379

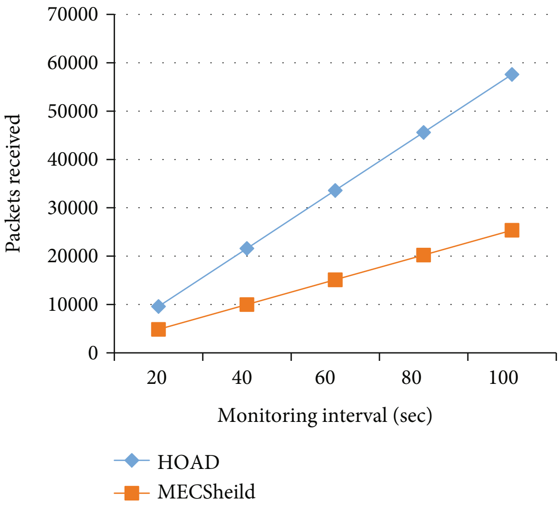

Figure 7: Throughput vs. monitoring interval.

Table 6: Result table of residual energy for time scenario.

Monitoring interval (sec)	HOAD	MECSheild
20	11.72123	11.45819
40	11.37577	11.15523
60	10.99889	10.87197
80	10.66189	10.40992
100	10.30918	10.10261

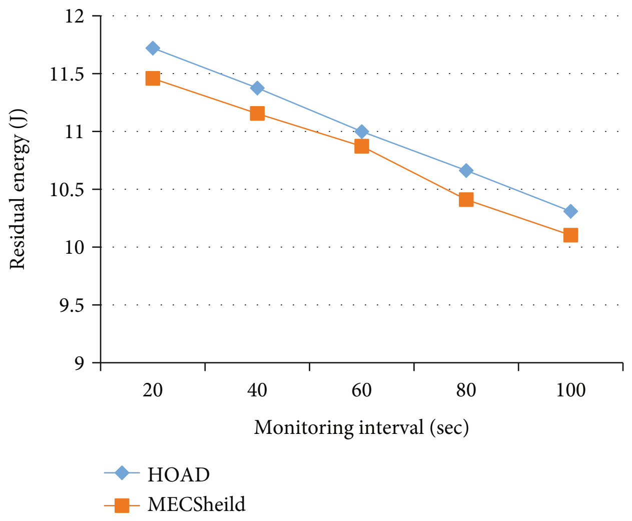

Figure 8: Residual energy vs. monitoring interval.

Table 7: Result table of E2D for frequency scenario.

Frequency (kb)	HOAD	MECSheild
50	0.006064	0.008049
75	0.006064	0.008051
100	0.006065	0.008052
125	0.006064	0.008053
150	0.006064	0.008053

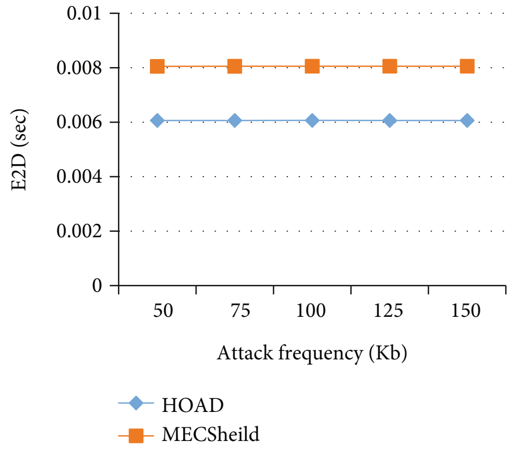

Figure 9: E2D vs. the attack frequency.

to deploy smart filters at the attack-source/destination pairs. But it mainly depends on a centralized controller, which may be subjected to single point of failures. The experimental settings are shown in Table 2.

4.1.1. Based on Time. In this section, results are plotted by varying the monitoring interval from 20 to 100 seconds.

Table 3 and Figure 5 show the E2D occurred in case of both frameworks. The figure shows that HOAD has 25% lesser E2D when compared to MECSheild.

Table 8: Result table of PDR for frequency scenario.

Frequency (kb)	HOAD	MECSheild
50	0.999783	0.999523
75	0.999855	0.999682
100	0.999891	0.999761
125	0.999826	0.999809
150	0.999855	0.999841

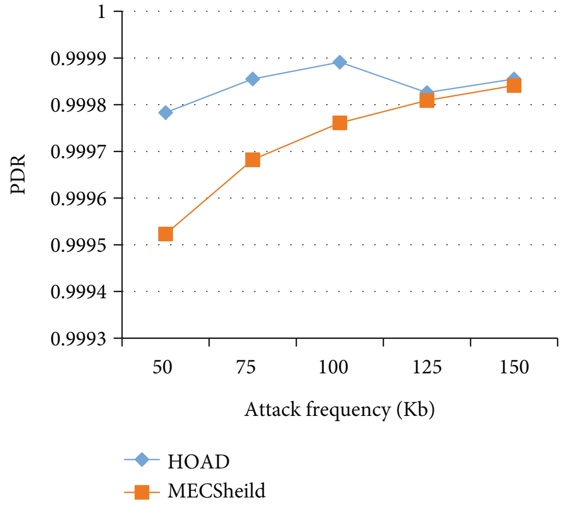

Figure 10: PDR vs. attack frequency.

Table 9: Result table of throughput for frequency scenario.

Frequency (kb)	HOAD	MECSheild
50	9203	4190
75	13801	6283
100	18406	8376
125	23001	10469
150	27599	12562

Table 4 and Figure 6 show the PDR measured in case of both frameworks. It was seen that HOAD has 0.84% higher PDR than MECSheild.

Table 5 and Figure 7 show the throughput measured in case of both frameworks. It has been seen that HOAD has 54% higher throughput than MECSheild.

Table 6 and Figure 8 show the average residual energy measured, in case of both frameworks. It was observed that HOAD has 1% higher residual energy than MECSheild.

4.1.2. Based on Attack Frequency. In this section, results are plotted by varying the attack frequency from 50 to 150 Kb.

Table 7 and Figure 9 show the E2D occurred in case of both frameworks. From the figure, it can be seen that HOAD has 25% lesser delay than MECSheild.

Table 8 and Figure 10 show the PDR measured in case of both frameworks. It was seen that HOAD has 0.1% higher PDR when compared to MECSheild.

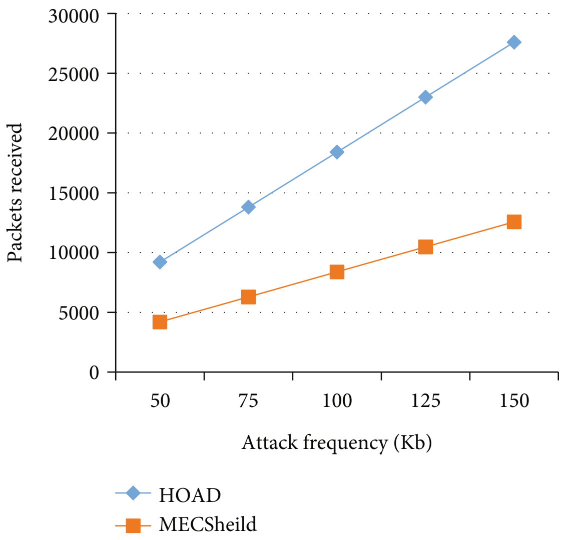

FIGURE 11: Throughput vs. attack frequency.

TABLE 10: Result table of residual energy for frequency scenario.

Frequency (kb)	HOAD	MECSheild
50	11.04721	10.55329
75	11.11845	10.81944
100	11.15534	10.9549
125	11.17708	11.03598
150	11.19194	11.09014

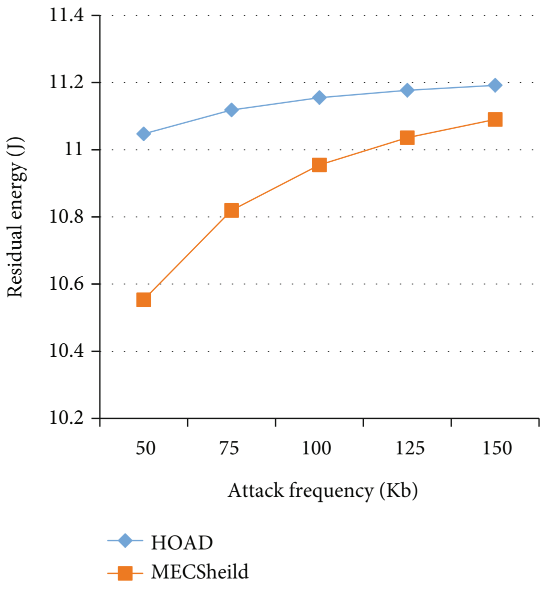

FIGURE 12: Residual energy vs. attack frequency.

Table 9 and Figure 11 show the throughput measured in case of both frameworks. It has been seen that HOAD obtains 55% higher throughput than MECSheild.

Table 10 and Figure 12 show the average residual energy measured for both the frameworks. It was observed that HOAD has 2% higher residual energy than MECSheild.

5. Conclusion

In this paper, we have proposed a hierarchical self-healing framework for IoT security. In this framework, primary (PC) and secondary controller (SC) nodes are deployed in the network along with the IoT devices. The SCs can be moved on-demand by the PC. The response agent at PC will first establish a new route via the SCs by excluding the intruders and the suspected nodes. Then, it will resend the stored packets to their destination via the newly established route. By simulation results, we have shown that the proposed technique increases the efficiency and reduces overhead and energy consumption. Advanced cryptographic standards can be used to tighten the security process in a more efficient manner by evaluating more types of threats. The security paradigm can be used in a 5G-based IoT network as well.

Data Availability

No data were used to support this research work.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] C. Nadia, M. Mohamed, Z. Akka, S. Cyrille, and F. Parvez, *Network Intrusion Detection for IoT Security Based on Learning Techniques*, IEEE Communications Surveys and Tutorials, 2019.

[2] D. Nhu-Ngoc, V. Trung, S. Umar, K. Joongheon, B. Thomas, and C. Sungrae, "Securing heterogeneous IoT with intelligent DDoS attack behavior learning," 2019, https://arxiv.org/abs/1711.06041v3.

[3] T. A. Ahanger, "Defense scheme to protect IoT from cyber attacks using AI principles," *International Journal of Computers Communications & Control*, vol. 13, no. 6, pp. 915–926, 2018.

[4] D. Daniel, B. U. Mar-a, S. Claudia et al., "Shielding IoT against cyber-attacks: an event-based approach using SIEM," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3029638, 18 pages, 2018.

[5] A. B. Feroz Khan and Anandharaj, "A new framework and defensive techniques for DDOS attack on IoT," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 8, p. 1, 2019.

[6] B. Ketan, M. Joaquin, and G. Ada, *Towards IoT-DDoS Prevention Using Edge Computing*, HotEdge, 2018.

[7] A. Ali, Z. Mohamed, D. Debatosh, O. Richard, and C. George, "Sybil attacks and defenses in internet of things and mobile social networks," *IJCSI International Journal of Computer Science Issues*, vol. 15, p. 6, 2018.

[8] M. Vincentius, C. Qiang, and B. Theophilus, *Fending Off IoT-Hunting Attacks at Home Networks*, CAN'17Incheon, Korea, 2017.

[9] M. Khalid, K. Muhammad, H. Mahmood, M. Ansar, A. Shahzad, and K. Muhammad, "Intelligent on-demand connectivity restoration for wireless sensor networks," *Wireless*

Communications and Mobile Computing, vol. 2018, Article ID 9702650, 10 pages, 2018.

[10] M. Leila and T. Fatiha, "A twofold self-healing approach for MANET survivability reinforcement," *International Journal of Intelligent Engineering Informatics*, vol. 5, no. 4, pp. 309–326, 2017.

[11] W. Jing, Q. Wu, Y. Liu, and Q. Zhang, "A reliable primary-backup routing algorithm in wireless sensor network," *Physics Procedia*, vol. 24, pp. 1462–1468, 2012.