

Lightweight Denial of Service (DOS) Detection System Algorithm (LIDSA)

Abdul Fuad Abdul Rahman¹, Azni Ab Halim², Shazwani Salleh³, Nur Farahin Jamaludin⁴, Nurul Syazwani Kamarulzaman⁵, Madihah Zulfa Mohamad⁶

^{1,3,4,5,6}MyVAC Department, CyberSecurity Malaysia, Cyberjaya, Malaysia

²Faculty of Science and Technology, Universiti Sains Islam Malaysia (USIM), Malaysia

abdufuadrahman@gmail.com, ahazni@usim.edu.my, shazwani.salleh@cybersecurity.my,
nurfarahin@cybersecurity.my, nurul.syazwani@cybersecurity.my, madihah@cybersecurity.my

Abstract—The Internet of Things (IoT) is becoming an increasingly growing topic. IoT encompasses everything connected to the internet. IoT requires multi-facet security solutions where the communication is secured with confidentiality, integrity, and service availability. However, IoT Sensor Node has the requirement to minimize power usage and computational power due to the requirement to be miniature. Therefore, the challenge of implementing security in the IoT Sensor Node must be addressed. Even if the IoT Sensor Node is protected with encryption and authentication, the protection is not comprehensive to sustain a Denial of Service (DoS) attack. The DoS attack is considered one of the security threats that may affect IoT network's quality service and reduce IoT Sensor Node's lifespan. Hence, mitigation shall be needed to secure IoT Sensor Node. This paper aims to propose a lightweight Denial of Service (DoS) Detection Systems algorithm to secure IoT Sensor Node. The approach uses data from previous experiments and translated it to develop mitigation to secure IoT Sensor Node, thus increasing IoT Sensor Node's lifespan.

Keywords— Denial of Service, Internet of Things, Detection System, Lightweight, Sensor Node

I. INTRODUCTION

Since the Internet of Things (IoT) ecosystem will consist of a complex infrastructure from Sensors to Cloud, the challenge to cybersecurity is to choose which area to be secure first and worth investing. Any approach taken should consists and cover all angles of IoT components, and the most important is that the approach is easy to implement (Rahman et al., 2016). If the developed security approach is too complicated, it will be challenging to implement it. Therefore, this paper proposed a lightweight algorithm targeted to ease the DoS Detection System implementation on IoT Sensor Node by using data obtained from experiments and then testing it on an actual environment without relying on simulated data.

An algorithm is a finite set of instructions which, if followed, accomplish a particular task. In designing and developing an algorithm, some criteria should adhere to. The criteria are Input, Output, Definiteness, Finiteness, and Effectiveness. The difference between an algorithm and a program is that a program does not necessarily satisfy the

fourth (Finiteness) condition. The finiteness condition states that the algorithm must terminate after a finite number of steps. One important example of such a program for a computer is its operating system, which never terminates (except for system crashes) but continues in a wait loop until more jobs are entered.

An algorithm may be expressed in a number of ways, including flow charts and pseudocode. In this paper, a flowchart and pseudocode are created in order to assist the process of developing the lightweight algorithm. The algorithm is designed specifically for detecting a DoS attack on IoT Sensor Node. It shall be lightweight to adhere to IoT Sensor Node low computational power and low energy consumption. Results from Rahman et al. (2018) and Rahman et al. (2019) were used and translated into the algorithm.

II. PREVIOUS WORKS

The experiment conducted in Rahman et. al. (2018) and Rahman et al. (2019) uses basic wireless data logging hardware with a very basic 'Sensor to Cloud' architecture. This research was adopted to develop a Smart Monitoring System that will function as the testbed.

The Smart Monitoring System testbed uses Wi-Fi IEEE 802.11 that is widely used in the 'Sensor to Cloud' architecture. The testbed reads the value of a voltage, time, humidity, and temperature every five (5) minutes and sends it to the Cloud server listening on TCP port 80. The experiment measured energy consumption by using the voltage data sent to the Cloud.

The Cloud server will listen for communications on port 80 and print out data received to the standard output (Dashboard). The Dashboard is the preferred data presentation for the testbed because it is suitable for achieving a real-time data presentation and interactive data visualization to humans, which help in data analysis and decision-making (Kamal et al., 2018). However, the Cloud was set up just to receive data without any optimization or security installed and configured since the testbed's approach is to measure the energy consumption of sensor to cloud without any optimization. Cloud and IoT are two complementary technologies. Cloud has virtually unlimited capabilities in terms of storage and

processing power (Rahman et al., 2018), which makes it suitable to be used in the testbed with active information interactions.

The IoT Sensor Node used in the Smart Monitoring System was designed in minimal weight, miniature form-factor, limited processing power using 2 AA batteries, storage, nRF24L01 operating on 2.4GHz ISM (Industry Scientific and Medical) bandwidth and standard-based interface protocols, as shown in Fig 1 and Fig 2. Humidity and temperature RHT03 sensors are part of the sensor node. The RHT03 (also known as DHT-22) is a low-cost humidity and temperature sensor with a single wire digital interface. The sensor is calibrated and does not require extra components, so it can directly measure relative humidity and temperature.

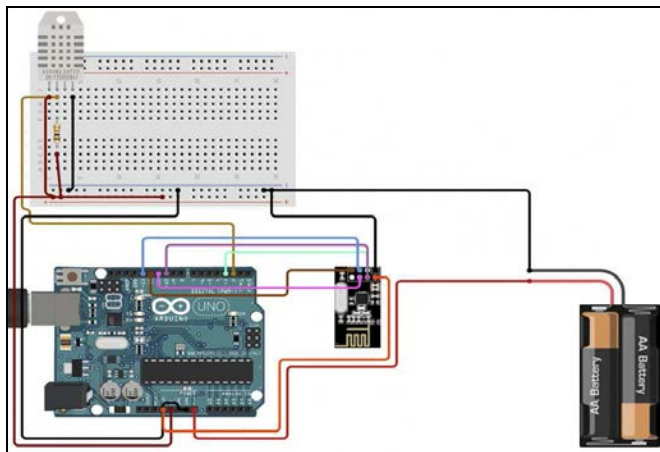


Figure 1. Prototype of 1 unit of IoT Sensor Node

A. Experiment IoT Sensor Node Lifespan without DoS Attack

Node in regular operation (without DoS attack). The experiment conducted was categorized into two, one with a wired connection and another with a 2.4GHz wireless connection. However, this paper focuses on the result of wireless connection since the proposed algorithm is embedded on IoT Sensor Node with wireless capability.

The initial voltage for both experiments is 3V, and the time measured will provide inputs to estimate the amount of the battery's lifetime for each experiment. Based on the data obtained through the Cloud for IoT Sensor Node using wireless 2.4GHz, a graph Voltage over Time was produced, as shown in Fig. 3. Based on the figure, the voltage significantly drops at 2.6V after operating more than 925 minutes (15 hours 25 minutes), and the IoT Sensor Node voltage drop to 0V after running more than 1515 minutes (25 hours 15 minutes). Further analysis on the graph of Voltage over Time, shows that when the hardware is consuming more energy to send data, the voltage drops slightly. The pattern shows how changing energy demands affect the battery voltage over time. The same pattern is also applicable for IoT Sensor Node with wired connection. The difference is the time of the voltage drop, which is after operating more than 925 minutes (15

hours 25 minutes) for IoT Sensor Node using wireless, compared to 1025 (17 hours 5 minutes) minutes for IoT Sensor Node using wired.

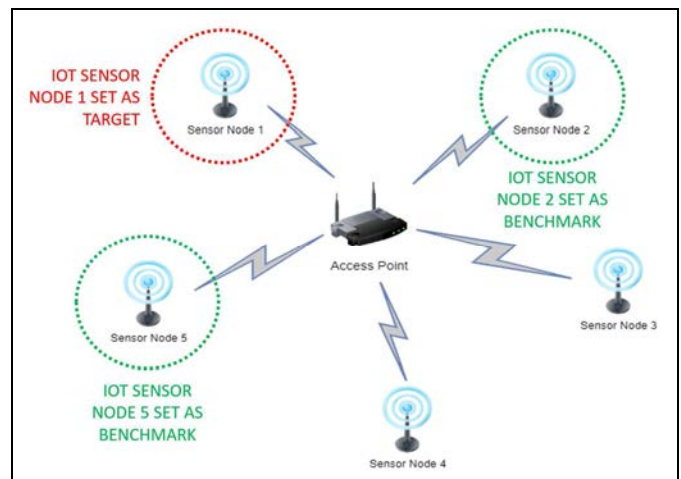


Figure 2. Topology of IoT Sensor Node

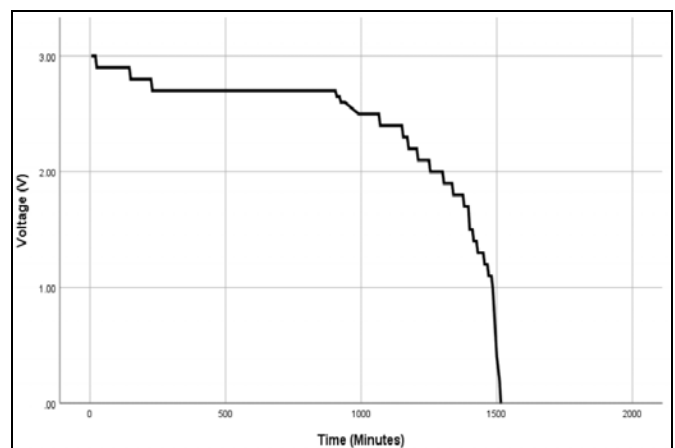


Figure 3. Voltage over Time for IoT Sensor Node using Wireless 2.4GHz

B. Experiment IoT Sensor Node Lifespan with DoS Attack

Rahman et al. (2019) study the impact of DoS attacks on a 'Sensor to Cloud' architecture. Rahman et al. (2018) state that the voltage drops slightly when the hardware is consuming more energy to send data. The same pattern also occurs in the experiment conducted by Rahman et al. (2019). DoS attack will cause the IoT Sensor Node to consume more power as it needs to process all the packets. Based on Daud et al. (2018), the energy drop calculation depends on the relationship between actual energy consumption (volt) per minute (min). Energy drop may be affected by the number of devices connected, high network utilization, and data overload in each cluster node. However, during a DoS attack, the incoming anomaly packets will lead to the higher energy consumption of infected nodes (Daud et al., 2018).

The explanation behind this result is due to high electromagnetic emissions that cause exhaustion of a sensor battery (Shakhov, 2013). When attacked, the nodes need to handle the anomaly packets and keep probing the network to

ensure connectivity. The situation will lead to higher electromagnetic emissions and, consecutively, higher energy consumption. Thus, affecting the network and at the same time, can drain the device's power faster. As IoT Sensor Node relies on the limited battery-powered resource, this situation will shorten the node's life span.

The results for IoT Sensor Node during DoS attack are presented in Fig. 4. The graph shows that the DoS attack towards IoT Sensor Node is feasible. For reference, Sensor Node 1 was configured as the DoS victim (can also be referred to as a malicious node). The impact of the attack causes Sensor Node 1 to be defective, and no data sent to the Cloud after the attack. There is 12.5 percent (12.5%) of data loss within five (5) minutes, and after the fifth minutes of testing, no data reading is received on the Cloud. The node is still alive, but due to the attack, it was unable to operate correctly. As seen from Fig.4, Sensor Node 1 was completely shut off after 145 minutes of testing.

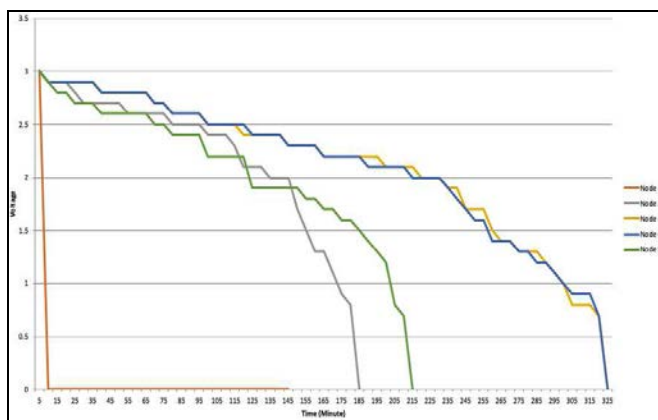


Figure 4. Graph Voltage over Time during DoS attack

Since there is no data received from Sensor Node 1 to the Cloud during the DoS attack, data from the nearest nodes (based on topology as shown in Fig. 2, Sensor Node 2 and Sensor Node 5 are the nearest neighbors) is used as a benchmark for the algorithm. The assumption is that impact of DoS attack on the victim significantly affects neighbor nodes as explained by (Lawyer, 2015) and (Hadi & Miften, 2018). Table 1 compares the time taken for the power to drop to a certain percentage during normal operation and during DoS attack. These values are then used to set the threshold in the algorithm.

TABLE I. COMPARISON OF ENERGY DROP

Energy Drop (%)	With DoS Attack (minutes) (Daud et al., 2018)	Without DoS Attack (minutes) (Rahman et al., 2018)
3	20	120
10	30	720-820
50	155-185	1200-1300

Therefore, Daud et al. (2018) recommend the adoption of security measures to counter the vulnerability. For example, implementation of the various Intrusion Detection System to

detect DoS attack patterns and signature, and clustering sensor nodes to increase network lifespan to ensure the availability and increase IoT system lifespan. Thus, the proposed algorithm in this paper will be focusing on developing a lightweight algorithm for IoT Sensor Node.

III. PROPOSED LIDSA

An algorithm can be described in several ways. One way is to give it a graphical form of notation, known as a flowchart, that places each processing step in a box and uses arrows to indicate the next step. Another way is with pseudocode. The pseudocode is an implementation of an algorithm in the form of annotations and informative text written in plain English without strict programming language syntax. As previously mentioned, both flowchart and pseudocode are developed to assist the process of developing the lightweight algorithm.

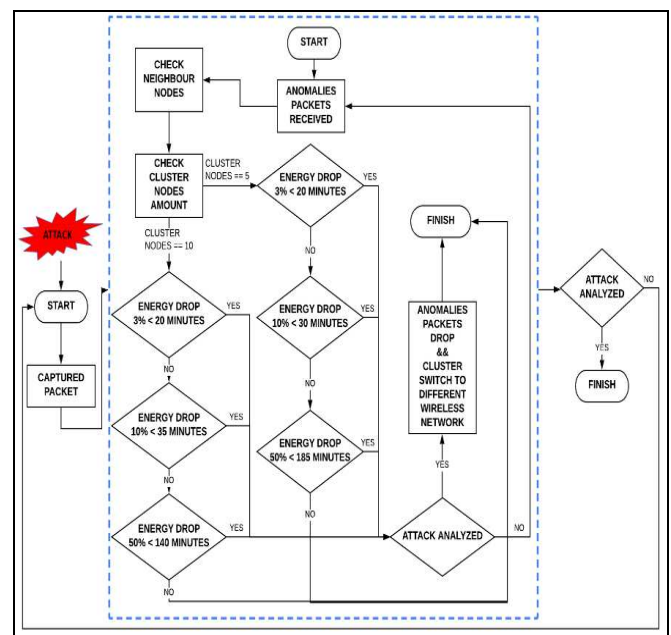


Figure 5. Flowchart of the proposed algorithm

The lightweight algorithm shall be lightweight to adhere to IoT Sensor Node low computational power and low energy consumption. The lightweight algorithm for IoT Sensor Node shall prevent abnormal energy consumption by monitoring, calculating, and evaluating energy drop from each cluster node based on few conditions. Fig. 5 shows the proposed algorithm's flowchart, and Fig. 6 shows the pseudocode of the proposed algorithm.

The lightweight algorithm was designed specifically for mitigating a DoS attack on IoT Sensor Node, using data from the experiment as shown in Table I. The proposed lightweight algorithm allows the IoT Sensor Node to switch to a different wireless network during the attack in order to increase the lifespan of the battery and thus improve the quality service of the IoT network. Based on the Table I, in abnormal cluster node behavior, if energy drop reaches 3% in less than 20 minutes anomalies packet will be analyzed, and abnormality is classified as an attack. Anomalies packets will be dropped,

and the affected cluster will switch to a different wireless network to terminate the outbreak. There are following conditions after that. If the voltage drop is less than 3% within 20 minutes, packets will be transmitted as usual. If there is energy drop reaches 10% in less than 30 minutes, the abnormal is classified as an attack, so the anomalies packet will be dropped, and the cluster will switch to a different wireless network. If not, the testbed operates as usual. Then, if the energy drop reached 50% in less than 185 minutes, the algorithm will detect the situation as abnormal behavior. The anomalies packets will be dropped, and the IoT Sensor Node will switch to a different wireless network.

```

1 Begin
2   if cluster_nodes == 5
3     loop
4
5       if node == nodes
6         if EnergyPercentage == 3% && RunningTime < 20 minutes
7           Switch different wireless network
8
9         else if EnergyPercentage == 10% && RunningTime < 30 minutes
10          Switch different wireless network
11
12        else if EnergyPercentage == 50% && RunningTime < 185 minutes
13          Switch different wireless network
14
15        else
16          while nodes < 5
17
18      else if cluster_nodes == 10
19        loop
20
21        if node == nodes
22          if EnergyPercentage == 3% && RunningTime < 20 minutes
23            Switch different wireless network
24
25          else if EnergyPercentage == 10% && RunningTime < 35 minutes
26            Switch different wireless network
27
28          else if EnergyPercentage == 50% && RunningTime < 140 minutes
29            Switch different wireless network
30
31          else
32            while nodes < 10
33
34      else
35        print "No Node found";
36        return;
37 END

```

Figure 6. Pseudocode of the proposed algorithm

IV. IMPLEMENTATION OF LIDSA

The experiment's analysis is used to measure the effectiveness of the proposed algorithm in preventing DoS attacks on the IoT Sensor Node. The result after the implementation of LIDSA is presented in Fig. 7.

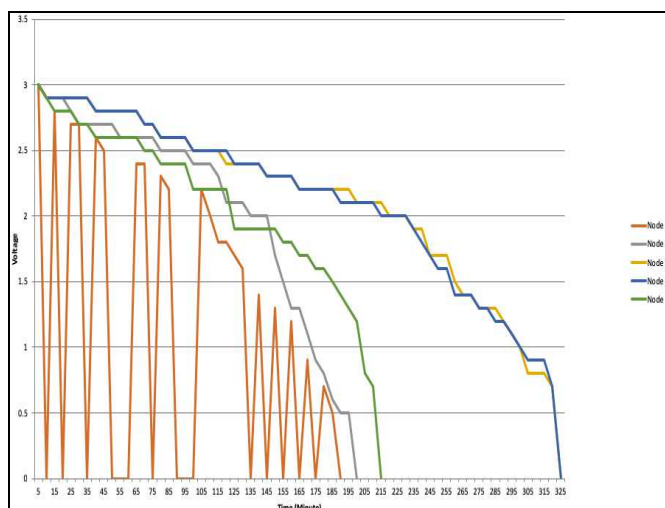


Figure 7. Result of LIDSA implementation

A. The Lifespan of IoT Sensor Node

A closer look at the graph shows an irregular pattern. Once the DoS attack launched, the Sensor Node 1 voltage reading has a sharp, sudden drop off to zero voltage at tenth minutes. Five minutes after that, the node operated and sent data to Cloud with a voltage a little bit lower than 3V before it was disabled again. The pattern was repeated until the node was completely shut off after 190 minutes. The longest time it was able to operate without being interrupted is during minute 105 until 135 (30 minutes).

However, due to the DoS attack, the voltage has decreased to 2.8V. The DoS attack occurred again in minute-20, causing the node to be temporarily defective. The situation met the algorithm's first condition, so the packet was dropped, and Sensor Node 1 switched to another wireless network. The pattern was repeated whenever the conditions of the algorithm matched. If the conditions do not match, the packets will be transmitted as usual, which means the IoT Sensor Node will be able to send data to the Cloud without interruption. If that is the case, the graph line will be one with slow and steady voltage drops. However, despite the irregular pattern, the figure proves that Sensor Node 1 can send data to Cloud during the DoS attack and with an improved lifespan with the implementation of the proposed algorithm.

The algorithm executes the same conditions on other nodes (Sensor Node 2 until Sensor Node 5). For example, voltage for Sensor Node 2 decreased to 2.9V within 20 minutes of operation. This value calculated about 3.33% of the voltage drop, which met the algorithm's first condition. Therefore, Sensor Node 2 dropped the package and switch to another wireless network. However, the algorithm's next condition does not match, as the voltage drop was less than 10%, so the node continues to operate as usual. The voltage reading was monitored continuously in order to measure the voltage drop.

At 155-minute, the voltage drop reached 50% (1.5V), which met the third condition. The node was triggered to switch to another wireless network. Even so, a better inspection of Fig. 7 shows that there is a sharp voltage drop from minute-155. This value is possibly due to the node consumes more energy when it keeps trying to connect to other wireless networks.

Analysis of the implementation of LIDSA on the testbed shows a significantly improved lifespan of IoT Sensor Node. From Fig. 4, The lifespan of Sensor Node 1, which was configured as the DoS victim, is less than 150 minutes. However, after the implementation of the proposed algorithm, the lifespan of Sensor Node 1 was increased up to 190 minutes.

This result shows that the algorithm was able to improve the lifespan of the IoT Sensor Node by up to 40 minutes. Therefore, the objective of using data from previous experiments and translated it to develop a DoS mitigation to secure IoT Sensor Node, thus increased the lifespan of IoT Sensor Node, was successfully achieved.

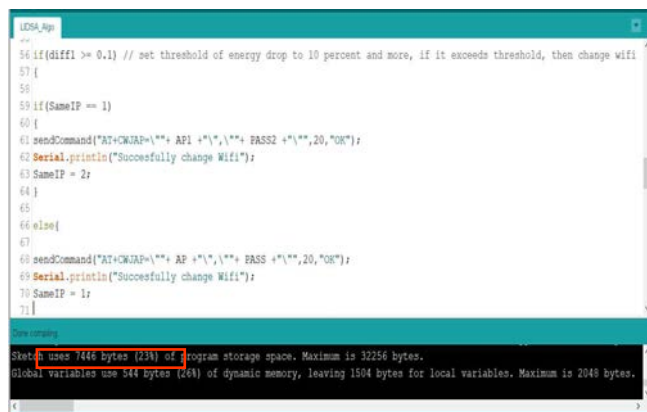
TABLE II. LIFESPAN DIFFERENCE OF IoT SENSOR NODE

Sensor Node	Without LIDSA (minutes)	With LIDSA (minutes)	Lifespan Difference (minutes)
1	145	190	40
2	185	200	15
3	325	330	5
4	325	330	5
5	215	220	5

B. Lightweight Algorithm

With the limitation of memory and power resources, LIDSA should be lightweight and ideal to be implemented on IoT Sensor Node. Program Memory Usage (PMU) is used for testing this criterion. When the size of a program is smaller, less processing power is needed to run the program.

The IoT Sensor Node that set as the testbed, uses the Arduino UNO board with the ATmega328P microcontroller. There are three (3) different types of memory defined in Arduino (Arduino - Memory, n.d.), which are Flash Memory, Static Random-Access Memory (SRAM), and Electrically Erasable Programmable Read-Only Memory (EEPROM). The Program Memory Usage (PMU) depends on the size of the assembly code for the algorithm. The percentages are calculated based on the size of Flash Memory and SRAM of this microcontroller. The program's size is displayed after compiling and uploading the program in the notification window at the bottom of the Arduino IDE. A warning of 'Low memory available' will be displayed if the PMU percentage is 70 percent (70%) or more. Fig. 8 shows the size of the proposed algorithm after successfully being compiled. Based on the red box as shown in the figure below, the algorithm consumes 7446 bytes of program storage space out of a maximum 32KB. This data means the algorithm takes about only 23 percent (23%) of the flash memory.

**Figure 8.** Size of proposed algorithm of 7446 bytes

For an optimized and efficient operation, the PMU of IoT Sensor Node should not exceed 50 percent (50%) of total available memory space, even at peak usage (Recommended Alert Thresholds, n.d.). The PMU for the proposed algorithm

is less than 50% in each testing. Deployment of the proposed algorithm on a different program may result in different PMU percentage. However, the proposed algorithm's size is fixed and consumed a small portion of memory space only. The experimental results proved that the proposed algorithm is lightweight, minimal-consumed memory capacity, and suitable to be implemented on the IoT Sensor Node

V. CONCLUSION

Further research could be directed towards enhancing the hardware and software for practical testing. Therefore, it is proposed that for future research, a new hardware and software development specifically designed to assess the IoT security that not limited to measuring security impact only, but also discovering a new vulnerability and threats. Further enhancement on the lightweight algorithm is a must to keep pace with the rapid technology development. For suggestions, the improvement may also include cryptography or encryption mechanism in the algorithm. For future research, the impact and mitigation of DoS attacks on Cloud are recommended to be tested and evaluated in order to secure the IoT ecosystem as a whole.

In conclusion, this research suggested that the proposed lightweight algorithm is essential to secure the IoT Sensor Node from DoS attacks. The algorithm can equip any sensor with the capability of preventive and detective mechanisms. This research can be a future reference for other researchers who want to improve what is achieved in this research.

ACKNOWLEDGMENT

The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM) for the support and facilities provided. This research paper is supported by Fundamental Research Grants No: USIM/FRGS/FST/32/51414. We also would like to thank the CyberSecurity Malaysia (CSM), an Agency under the Ministry of Communication and Multimedia (KKMM) for their contribution.

REFERENCES

- [1] Rahman, A. F. A., Halim, A. A., Alwi, N. H. M., Alwi, K. S., Mohamad, F. A., Taufiq, M. N., Mohamad, M. Z., & Abidin, K. A. Z. (2018). Measuring Sensor to Cloud Energy Consumption. *Proceedings of the 2018 2nd High Performance Computing and Cluster Technologies Conference on ZZZ - HPCCT 2018*, 43–47. <https://doi.org/10.1145/3234664.3234672>
- [2] Rahman, A. F., Halim, A. A., Syazwani, N., Daud, M., Mohamad, M. Z., & Yahood, M. I. (2019). Development of Denial of Service (DoS) Mitigation for Internet of Things (IoT) Sensor Node. *Global Journal of Engineering and Technology Review Vol.4 (2) April-June. 2019 Global Journal of Engineering and Technology Review*, 4(2), 26-34. doi:10.35609/gjetr.2019.4.2(2)
- [3] Daud, M., Rasiah, R., George, M., Asirvatham, D., Rahman, A. F. A., & Halim, A. A. (2018). Denial of service: (DoS) Impact on sensors. 2018 4th International Conference on Information Management (ICIM), 270–274. <https://doi.org/10.1109/infoman.2018.8392848>
- [4] *Arduino - Memory*. (n.d.). Arduino. Retrieved July 10, 2020, from <https://www.arduino.cc/en/tutorial/memory>
- [5] Shakhov, V. V. (2013). Protecting Wireless Sensor Networks from Energy Exhausting Attacks. *Lecture Notes in Computer Science*, 184–193. https://doi.org/10.1007/978-3-642-39637-3_15

- [6] Syamsul Kamal, S. S., Abdul Rahman, A. F., Salleh, M. N. T., Mohamad, F. A., Daud, M., & Mohd, M. (2018). Comparison on Scorecard and Dashboard in Smart Water Monitoring Application. *In Proceedings of the 23rd Conference of Open Innovations Association FRUCT (FRUCT'23)*, 549–555. <https://dl.acm.org/doi/10.5555/3299905.3299982>
- [7] Lawyer, G. (2015). Understanding the influence of all nodes in a network. *Scientific Reports*, 5(1). doi:10.1038/srep08665
- [8] Hadi, I., & Miften, F. S. (2018). A Graph Clustering Algorithm Based on Adaptive Neighbors Connectivity. *Journal of Telecommunication, Electrical and Computer Engineering*, 9, 2-11. doi: https://www.researchgate.net/publication/326235250_A_Graph_Clustering_Algorithm_Based_on_Adaptive_Neighbors_Connectivity
- [9] R. J. Vidmar, On the use of atmospheric plasmas as electromagnetic reflectors. *IEEE Trans. Plasma Sci.* [Online]. 21 (3). pp. 876–880. Available: <http://www.halcyon.com/pub/journals/21ps03-vidmar>, Aug. 1992.



Abdul Fuad Abdul Rahman, is a the Head of Malaysia Vulnerability Assessment Centre (MyVAC) of CyberSecurity Malaysia. Prior to that, he has a System Security Certified Practitioner (SSCP) from The International Information Systems Security Certification Consortium (ISC)2, GIAC Assessing Wireless Network (GAWN) from SANS Institute of America and Certified Network Engineer IPv6 (CNE6).



Azni Ab Halim is the Deputy Dean of Postgraduate Centre of Studies of Universiti Sains Islam Malaysia (USIM). She has a Doctorate Degree in the field of ICT from the Universiti Teknikal Malaysia Melaka (UTeM) in 2012. She is currently focusing her research works in the field of ICT security.



Shazwani Salleh is an Analyst at CyberSecurity Malaysia in Cyberjaya where she is involved in Research and Development (R&D) for Internet of Things (IoT) projects as well as Vulnerability Assessment and Penetration Testing (VAPT) services.. She received her Bachelor of Science (B.Sc), in field of Business Computing, from Universiti Teknologi Mara (UiTM), Malaysia, in 2015. She completed her Master of Science (Msc), in field of Information Management, from Universiti Teknologi Mara (UiTM) in 2016. In 2017, she started her career as Associate Analyst at DHL Asia Pacific IT Services.



Farahin Jamaludin is an Analyst at CyberSecurity Malaysia in Cyberjaya where she is involved in Research and Development (R&D) for Internet of Things (IoT) projects as well as Vulnerability Assessment and Penetration Testing (VAPT) services. She received her Bachelor's Degree in the field of Engineering Technology (Manufacturing), from Universiti Malaysia Pahang, Malaysia, in 2019.



Nurul Syazwani Kamarulzaman is an Analyst at Malaysia Vulnerability Assessment Center (MyVAC) department where she is involved in Vulnerability Assessment Penetration Testing (VAPT) and Research and Development (R&D) for Internet of Things (IoT). She received her Bachelor's Degree in the field of Science Computer (Network Security), from University Sultan Zainal Abidin (UniSZA), Terengganu Malaysia in 2016.



Madihah Zulfa Mohamad, a Manager with the CyberSecurity Malaysia. Prior to that, she has a Certified Ethical Hacker (CEH) from EC-Council and Microsoft Certified System Engineer (MCSE) from Microsoft. She is currently focus in the research on the field of IoT and ICT Security

