Computers, Materials & Continua
DOI:10.32604/cmc.2021.016938
Article

Tech Science Press

Detecting Man-in-the-Middle Attack in Fog Computing for Social Media

Farouq Aliyu[1,*], Tarek Sheltami[1], Ashraf Mahmoud[1], Louai Al-Awami[1] and Ansar Yasar[2]

[1]King Fahd University of Petroleum and Minerals, Dhahran, 31261, Saudi Arabia
[2]Transportation Research Institute (IMOB), Hasselt University, Hasselt, 3500, Belgium
[*]Corresponding Author: Farouq Aliyu. Email: g201303650@kfupm.edu.sa
Received: 15 January 2021; Accepted: 02 April 2021

Abstract: Fog computing (FC) is a networking paradigm where wireless devices known as fog nodes are placed at the edge of the network (close to the Internet of Things (IoT) devices). Fog nodes provide services in lieu of the cloud. Thus, improving the performance of the network and making it attractive to social media-based systems. Security issues are one of the most challenges encountered in FC. In this paper, we propose an anomaly-based Intrusion Detection and Prevention System (IDPS) against Man-in-the-Middle (MITM) attack in the fog layer. The system uses special nodes known as Intrusion Detection System (IDS) nodes to detect intrusion in the network. They periodically monitor the behavior of the fog nodes in the network. Any deviation from normal network activity is categorized as malicious, and the suspected node is isolated. Exponentially Weighted Moving Average (EWMA) is added to the system to smooth out the noise that is typically found in social media communications. Our results (with 95% confidence) show that the accuracy of the proposed system increases from 80% to 95% after EWMA is added. Also, with EWMA, the proposed system can detect the intrusion from 0.25–0.5 s seconds faster than that without EWMA. However, it affects the latency of services provided by the fog nodes by at least 0.75–1.3 s. Finally, EWMA has not increased the energy overhead of the system, due to its lightweight.

Keywords: Fog computing; man-in-the-middle attack; intrusion detection system and prevention system; network security; social media

1 Introduction

Lately, there is an explosion in the number of Things connected to the Internet [1]. It is estimated that by the year 2025, the number of Internet of Things (IoT) devices may reach 21 billion [2]. As a result, an overflow of data will be experienced at the cloud layer. Hence, the cloud will not provide service to the IoT effectively. Moreover, due to this increase in data, cloud servers consume more energy and time processing data of whom 40% could be processed physically close to the user [3].

Due to the aforementioned problems, scientists proposed the deployment of a heterogeneous network close to the IoT devices. The network provides services to the IoT on behalf of the

cloud. Thus, improving the system's quality of service at an affordable rate. For example, Wu et al. [4] proposed a Fog Computing security as a Service (FCSS) system for information-centric social networks. The proposed system provides low-latency content filtering security service. In [5], the authors proposed a Fog Computing (FC) device called "Droplet" that is used as a distributed server for social network applications like Friendica [6]. Abdurrahman et al. [7] used FC on the Hajj social network that manages pilgrims' activities in the Hajj season. The system provides services such as locating lost pilgrims, informing pilgrims on the bus timing, Hajj activities schedule, and other services at low latency. Other applications of FC in social media can be found in [8–10].

Unfortunately, security in FC is becoming a great concern [11]. One of the most notorious attacks in FC is man-in-the-middle (MITM) attack [12,13]. A MITM attack is an insider attack where messages from the source node pass through a third party (attacker) before reaching the destination node while both the source and the destination are convinced that one is communicating directly with the other [14]. The two main types of MITM attacks are passive and active attacks [15]. A passive MITM attacker is only interested in the information transmitted. As such, he/she eavesdrops on the packets passing through without tempering with them, while an active attacker manipulates/modifies the packets received before forwarding them to the destination node.

Some researchers [16,17], argue that the MITM could be the most prevalent attack on FC systems because it allows the attacker access to the information from the user as well as the cloud during the communication session. A recent research shows that MITM is very difficult to detect [18]. Furthermore, an attacker is highly motivated to eavesdrop or temper with packets in an FC system because, in most cases, fog nodes process deeply personal information such as health information, and other sensitive information like the speed and destination of a vehicle, etc. It is easier for the attacker to attack the fog nodes than the server, since the fog nodes are resource-constrained devices. Traditional detection and prevention techniques for MITM attacks are impractical to implement in FC systems because fog nodes are often resource-constrained [19]. Therefore, lightweight security techniques for detecting and preventing MITM attacks must be developed.

In this paper, we extend our previous work on the development of IDPS for a MITM attack on a distributed FC system [1]. The work focuses on MITM attacks at the fog layer, where one or more fog nodes are compromised and they serve as middlemen between the IoT nodes and the cloud. One of the most effective ways of preventing MITM is by packet encryption [20]. It makes the packet useless and deters attackers. One recommended lightweight encryption technique is Advanced Encryption Standard (AES) [21], which we use in this paper. The contributions of this research are as follows:

1) A novel IDPS for MITM in the fog layer of an FC system.
2) Specialized nodes known as Intrusion Detection System (IDS) nodes for monitoring and probing the fog layer of an FC network.
3) The use of Exponentially Weighted Moving Average (EWMA) [22,23] to overcome the noisy nature of the network.

The remaining part of this paper is as follows: Section 2 is in two folds; Sections 2.1 and 2.2 provide a comprehensive literature review of the current Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) available in the literature. Section 3 describes the proposed system, the network model, and the attacker model simulated in this paper. Section 4 discusses, in detail, the performance of the proposed system. Finally, Section 5 concludes the paper.

2 Literature Review

According to the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) [24]; an Intrusion is the deliberate or accidental unauthorized access to information systems, networks, or a network-connected system. Tab. 1 shows some network intrusions using Man-in-the-Middle (MITM) attacks and their mitigations. An Intrusion Detection System (IDS) is a system that detects attempted intrusion, or whether an intrusion is taking place, or has already occurred. An Intrusion Prevention System is a variant of IDS with active response upon intrusion, while an Intrusion Detection and Prevention System (IDPS) is a combination of IDS and IPS where IPS fends off attacks once they are detected by the IDS part of the system.

Table 1: MITM vectors with their prevention and mitigations on traditional network

Vectors	Description	Preventions and/or Mitigations
Address Resolution Protocol (ARP) Spoofing Attack	The attacker intercepts a legitimate ARP request, then forge a reply with fake IP-MAC address pair base on the request. Thus, fooling the sender that it is the receiver.	(1) Tracking IP-MAC address pairs in the network. (2) Injecting ARP request and TCP SYN packets to probe inconsistencies [25]. (3) Using static routing tables [26]. (4) Using Digital Signature Algorithm (DSA) [27]. (5) Using Anticap [28] and Antidote [29] patches. (6) By checking ARP packets against a table of trusted Hosts [30].
ARP Port Stealing	The attacker sends fake ARP many packets with a nonexistent address to the switch, which causes an overflow of the switch table [31], forcing the switch to act as a hub.	(1) Use port security to limit MAC addresses per port [32], or (2) Dynamic ARP Inspection (DAI), Dynamic Host Configuration Protocol (DHCP) snooping, and MAC address monitoring [33].

(Continued)

Table 1: Continued

Vectors	Description	Preventions and/or Mitigations
DHCP Spoofing	The attacker takes down the server using Denial of Service (DoS) or Address starvation [34], then the attacker issue addresses in its place.	(1) Enable DHCP snooping. (2) Enable the port security feature of the router.
Domain Name Service (DNS) Spoofing	The attacker intercepts a legitimate DNS request, then uses the information therein to forge a reply. Thus, placing itself between the DNS and the victim [35].	(1) DNS Security Extension (DNSSEC) [36]. (2) E-DNSSEC [37].
Internet Control Message Protocol (ICMP) Redirect	The attacker forges ICMP packets that updates the hosts' routing table to make the attacker gateway [38].	Disable ICMP redirect messages on hosts or nodes.
ICMP Router Disc Protocol (IRDP) Spoofing	The attacker forges IRDP packets, thereby overriding the DHCP Configuration.	(1) Disable IRDP. (2) Use Static routing [39].
Route Mangling	The attacker injects fake network reconfiguration commands that allow the attacker to assume the role of the compromised router.	(1) Encrypting packets [40]. (2) Marking packets with unique keys or tags [41,42]. (3) Routers can use network prefixes to construct filters [43].
Spanning Tree Protocol (STP) Mangling	The attacker masquerades as a Root Bridge by forging fake packets, which forces other nodes to use the attacker as a new route [33].	(1) Bridge Protocol Data Unit (BPDU) guard [44]. (2) Root guard [45]. (3) Loop guard [33].

The literature barely provides any solutions to insider threats like MITM for fog computing (FC) [46]. This claim is supported by bibliometric analysis of the Web of Science database.

We use the search term, "(intrusion detection AND fog) OR (edge AND intrusion detection)". VOSviewer [47] is used to analyze the aforementioned database and as it shown in Fig. 1, the MITM attack is not on the map. As such, this literature review touches on solutions in fog computing-related fields such as the Internet of Things (IoT), wireless sensor networks (WSN), cloud computing, etc.

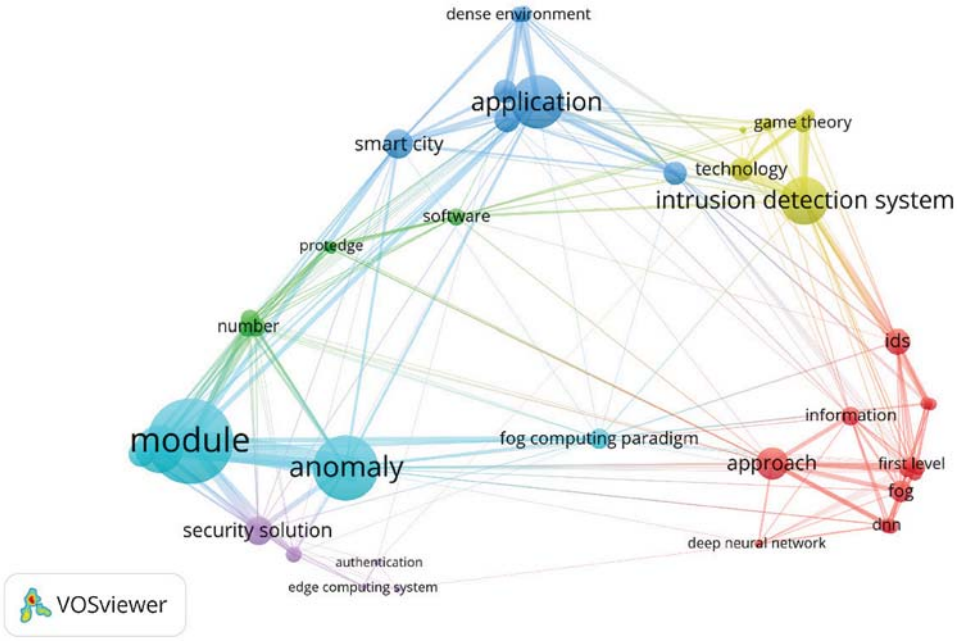


Figure 1: Bibliometric map of papers containing intrusion detection in fog computing

The literature review section is divided into two subsections: Subsection 2.1 discusses the different Intrusion Detection Systems (IDSs), and Subsection 2.2 presents the different Intrusion Prevention Systems (IPSs) available in the literature.

2.1 Man-in-the-Middle (MITM) Intrusion Detection Systems (IDS)

An IDS is a passive system that detects, classifies, and alerts the network administrator of intrusions, attacks, or violations of the security policies [48]. An IDS does not participate in mitigating or stopping the attack. To effectively detect MITM attacks, the behavior of nodes in the network must be observed. In a grey hole attack, the attacker forwards some of the packets while destroying others. Authors in [49] developed a technique for detecting grey hole attacks in Dynamic Source Routing (DSR) protocol. The system is a non-cryptographic technique. It compares the number of messages sent by a source and the number of messages received by a destination. When the receiver notices a difference in the two, it notifies an IDS node that the intermediate node is malicious. The IDS node, in turn, notifies other nodes in the network, thereby isolating the malicious node.

On the contrary, unlike grey hole attacks, MITM attacks do not destroy packets. Therefore, the aforementioned technique cannot work for MITM attacks since all packets sent are eventually received at the destination. Authors in [50] proposed the use of packet arrival time to detect the possibility of an attack. The technique detects an attack along a path when the difference between

the actual- and expected arrival time is greater than a threshold value T_{diff}. However, using a fixed threshold to detect MITM attacks may be difficult because of the noisy and heterogeneous nature of an IoT network, which causes a wide variation in the arrival time of packets.

Alternatively, Faria and Cheriton [51] proposed the use of signalprint to detect Masquerading and Resource Depletion attacks. A signalprint is analogous to a fingerprint in wireless devices. Each device has a unique signalprint. The authors argued that signalprint is; 1) hard to spoof, 2) strongly correlated with the physical location of clients, and 3) highly likely to have similar signalprints for packet bursts transmitted by a stationary node. In this technique, the server stores all Received Signal Strength Indicator (RSSI) of nodes in the network, which is appended to the packet automatically. The server then compares the RSSI in its database to that of the received packet; if the two differ by a threshold value, then the node is considered malicious. The authors used differential signal strength to ensure the development of a robust system. But it may still be difficult to account for the channel's characteristics, especially in a noisy environment like offices where there are constant human activities.

A packet experiencing a wormhole attack finds itself in a distant part of the network away from its destination [52]. Glass and Portmann [53] developed a MAC-layer-based intrusion detection system. The system is designed to detect both MITM and Wormhole attacks. In this technique, the source and the destination secretly agree on the number of frames to be transmitted without acknowledgment. Therefore, an intruder is detected when it sends acknowledgment before the agreed number of packets are transmitted. The authors were able to show that with a small trade-off of bandwidth, their system provides accurate detection of attack events.

Wang and Poster [54] proposed a network-based IDS to detect a wormhole attack using signed acknowledgment. The proposed system uses challenge-response acknowledgment, where the sender challenges the receiver with a message **r** and receiver adds a universally agreed secret value **s**, encrypt it with key **k** to form an acknowledgment packet $\{\mathbf{r},\mathbf{s}_k\}$. The receiver will carry out the same computation with the same values and compares it with the acknowledgment packet to determine whether the node is malicious. Since every packet has a unique acknowledgment, it is difficult for the adversary to carry out attacks undetected. However, the secret values and the encryption keys must be known by all legitimate nodes.

Recently, artificial intelligence is being used in MITM IDS [55]. Thamilarasu and Chawla [55] proposed an anomaly- and a network-based IDS for IoT using deep learning. The proposed system uses an independent integrated intrusion detection system that connects to the network and analyzes data at the transport layer. The system uses a virtual network client (VNC), a controller, and a connection prober: The VNC module is a client-based network emulator that is responsible for connecting with the IoT devices. The connection prober connects the VNC module with the connection prober. Then, the connection prober collects packets from the network and feeds them into a cache. The cache sends them to the data collection & transformation module, where they are reformatted. Finally, the reformatted data is fed to a feed-forward Deep Neural Network (DNN) developed using a Deep Belief Network (DBN), where intrusion detection takes place. The system was tested for opportunistic service attacks, black hole attacks, distributed denial-of-service (DDoS) attacks, sinkhole attacks, and wormhole attacks using real-network traces. It shows an accuracy of approximately 98%. However, the system may incur overhead due to the IDS system replicating packets. In addition, machine learning has been used in developing IDS for fog computing [56,57]. An et al. [56] proposed an IDS using an Extreme Learning Machine (ELM) called Sample Selected Extreme Learning Machine (SS-ELM). The technique uses the cloud to gather training samples from the fog nodes. The cloud then filters the samples and

sends the samples worthy of training the fog nodes. The authors used KDD Cup 99 dataset. The authors show that SS-ELM outperforms the classical backpropagation algorithm and support vector machine (SVM) in terms of detection accuracy. In [57], the authors use a single-layer perceptron to develop an IDS that monitors attacks on the fog nodes rather than the network. The proposed system was tested and trained using Australian Defense Force Academy Windows Dataset (ADFA-WD) and Australian Defense Force Academy Linux Dataset (ADFA-LD). The proposed system shows 94% accuracy. However, the dataset trains the nodes on past events. Therefore, there is no guarantee that fog nodes can protect themselves from future attacks.

2.2 Man-in-the-Middle (MITM) Intrusion Prevention Systems (IPS)

An IPS is also known as Intrusion Detection and Prevention System (IDPS). It is an active system that detects and mitigates malicious activities in a network [58,59]. A typical IDPS system uses anomaly detection, stateful protocol analysis, signature analysis, or a combination to detect cyberattacks. IDPS may be a single system or an amalgamation of many systems working together.

Several works' techniques for securing IoT can be found in the literature [60–62]. The most common way of preventing MITM attacks is by encrypting communication and isolating malicious and compromised nodes [46]. However, the aforementioned solutions were designed for IoT systems (without fog computing). To the best of our knowledge, there are no standard security systems tailored for Fog computing in the market [46].

Authors in [63] proposed an IoT-based authentication system. It is designed to guard against; man-in-the-middle, eavesdropping, replay attack, and key control attacks. The proposed system moved all computation to a Registration Authority (RA). An RA is a computer with more computational resources than IoT devices. Hence the IoT devices are alleviated from computation overhead due to authentication. The RA is tasked with the responsibility of authenticating and cataloging Things in the network. In a fog computing system, the fog nodes can be used as RAs. However, this will lead to a single point of failure as well as an increase in the complexity of the fog nodes.

In [64], the authors adopted the public key cryptography employed for traditional Internet and IoT authentication scheme to cloud computing. However, Public Key Infrastructure (PKI) based authentication is not suited to Fog computing due to the cloud's distance from the network's edge. Also, the technique lacks scalability and efficiency.

Authors in [65] view the fog computing architecture as a publish-subscribe system. The author developed a lightweight security solution for publish-subscribe protocol-based IoTs in Fog networks using Elliptic curve cryptography (ECC). The proposed technique reduces the number of handshakes and the size of messages transmitted in each handshake. Furthermore, ECC has a smaller size public key, which is convenient for the end devices. However, there is a computational overhead on the fog node. Thus, leading to energy consumption and an increase in latency.

Truelink [66] is a true IDPS developed to guard IoT systems from wormhole attacks. The system consists of two phases: Rendezvous phase and Authentication phases. The earlier is part of the IDS subsystem, while the latter is part of the IPS subsystem. During the rendezvous phase, the sender and the receiver exchange nonce (α_i and β_j). The arrival time of the nonce helps the two nodes to deduce their adjacency. When there is a node in the middle of the duo, the transmission time will, in principle, be higher than a given threshold value. Hence, making stealth difficult for the attacker. In the Authentication phase, the sender **(i)** and the receiver **(j)** exchange sign messages

(α_i and β_j). Thus, allowing both parties to mutually authenticate each other; as the source of the rendezvous packets. The proposed system assumes that arrival time is fixed, which is not the case in wireless networks. Often, the margin between the arrival times of malicious packets and the normal packets is narrow.

Shafi, Saad, and Abdul proposed a software-defined network (SDN) based IDPS that works in both cloud and fog computing layer [67]. The proposed system uses machine learning to detect the possibility of distributed denial of service (DDoS) in the network. The authors designed the SND controller for anomaly detection with three machine learning algorithms; Recurrent Multi-Layer Perceptron (MLP), Neural Network (RNN), and Alternate Decision Tree (ADT). Initially, the SDN controller uses RNN and MLP to vote "attack" or "normal". In the event of a tie, the controller uses the ADT to break it. To mitigate the attack, the SDN controller issues an access list to the switches, who then serve as a sink for the incoming traffic from the blacklisted prefixes. The system is tested in both the cloud and the fog layer. The authors found that the proposed system performs better in terms of latency, throughput, and packet loss in the fog computing-based network. However, using three machine learning techniques will incur high overhead in memory and latency in both networks.

Doshi, Mozaffari, and Yilmaz proposed a real-time host-based IDPS for DDoS in IoT [68]. In the proposed system, IDS is deployed locally in each node. The nodes periodically analyze their neighbors using statistical techniques such as the mean and standard deviation, or upper and lower bounds of the packet transmission rate. When a node detects a deviation from the statistical data, the node carries out further analysis to determine the alarm's accuracy. Then it localizes the threat. Finally, the node blocks the localized node. This technique has some limitations: The proposed system assumes that the network is static, which is not the case in most IoT-based networks—nodes come and go, and the topology is constantly changing. Also, it assumes that the behavior of the nodes in the network does not change over time.

In Summary, a MITM attack can be investigated based on the following behaviors exhibited by a node:

1) **Change in content of a packet:** This usually happens when malicious nodes deliberately alter transit packets.
2) **Delay in arrivals time of a packet:** As a consequence of the malicious nodes copying passing by packets or modifying them, the arrival time of the packet to its destination increases.
3) **Change in the direction of a packet:** The direction of a packet may be altered, especially in Wormhole attacks (which is a variant of MITM). Alternatively, the attacker may not know what to do with the packet if it has no complete knowledge of the protocol, this may also lead to the change in packet destination.

In this paper, an IDPS system for MITM attack is proposed. The system has two types of nodes: Fog nodes and IDS nodes. The fog nodes are responsible for providing services to the IoT devices while the IDS nodes are special nodes known as IDS nodes that interrogate the fog nodes in the network and also observe their behaviors based on the three (3) outlined behaviors mentioned earlier to conclude whether a fog node is malicious.

3 Proposed System

The proposed system consists of two types of nodes in the fog layer: Fog nodes (FN) and Intrusion Detection System (IDS) nodes. Fig. 2 shows a typical distributed fog network. Here, the Fog nodes receive requests from the IoT nodes, which are providing service on behalf of the

cloud. Since the fog nodes are physically closer to the IoT nodes relative to the cloud, the network latency is reduced. Furthermore, they have more computing resources than the IoT nodes and can perform more complex tasks.

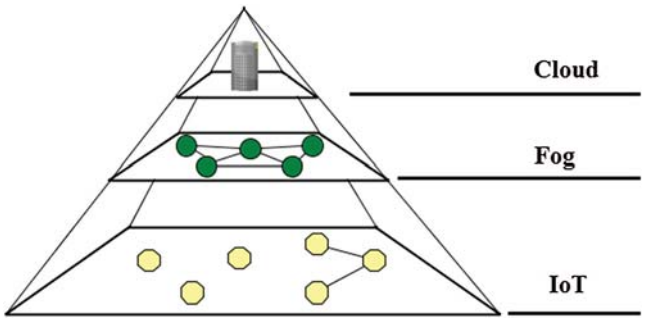


Figure 2: A fog network

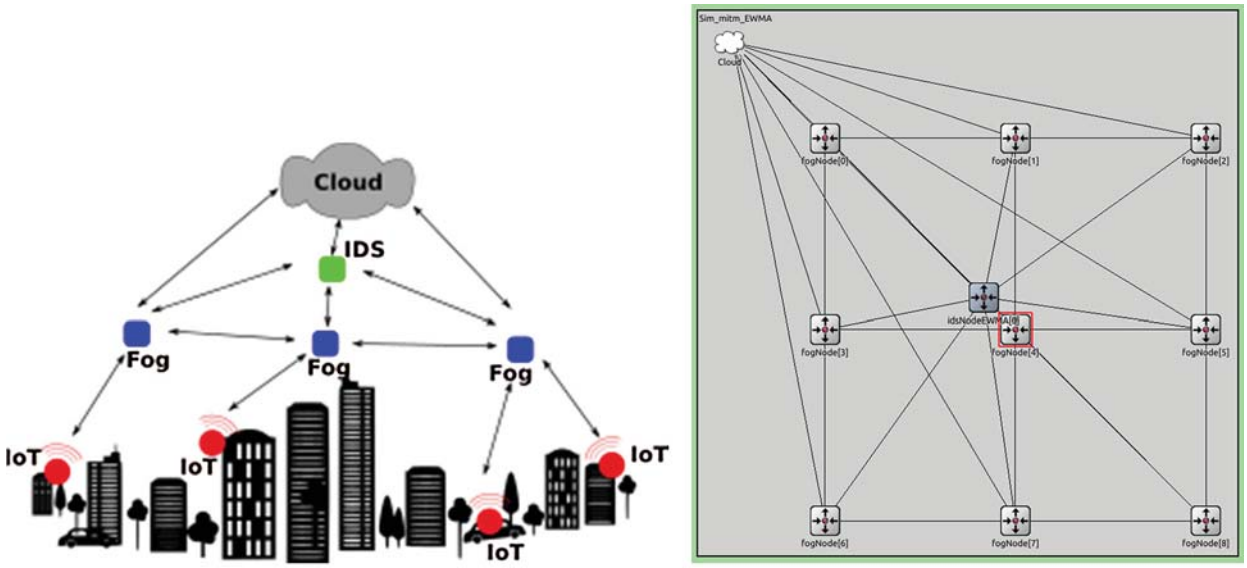
The fog layer is the link between the IoT and the cloud layer. Depending on the design, the fog layer can connect to the cloud with one communication medium while using another to communicate with the IoT layer. Similarly, the same protocols or different ones can be used by the fog layer to connect with the cloud layer and the IoT layer. However, it is advantageous for the fog layer to use the same medium and protocol because it reduces the complexity of the design of the fog nodes. Also, it reduces the energy and latency of communication, since there is no need to convert packets format.

The connection between the fog node and the cloud in most cases requires a change in medium and protocol stack. This is due to the long distance between them, may lead to significantly high latency. In this case, conversion from one protocol to another is worthwhile. In this paper, the performance of the fog layer is investigated when an IDPS is introduced.

3.1 Network Model

Fig. 3 shows an application scenario for the proposed system. In Fig. 3a, the fog nodes collect data from the IoT layer for service provision. Several services could be provided by the fog layer, for example; it can improve the connectivity of IoT nodes, provide social media analytics, and/or provide social media users with low-cost bandwidth. The nodes communicate over a wireless network. To ensure secure communication, the Advanced Encryption System (AES) is used by the nodes, which is an efficient encryption algorithm for Fog Computing (FC) and IoT [69]. The latest ARM microcontrollers that are designed for IoT applications have an on-chip AES module [70]. Also, Diffie-Hellman key exchange [71] is used by joining fog nodes to obtain a cryptographic key from the IDS nodes, which the IDS nodes obtained from the cloud. Like the fog nodes, the IDS nodes can communicate with the cloud but not with the IoT nodes. One or more IDS nodes can be deployed to observe a set/region of fog nodes.

Fig. 3a shows the framework of the proposed system and in Fig. 3b shows the proposed system using OMNET++ [72]. The FNs are responsible for providing services to the IoT layer and the IDS nodes are responsible for; observing the network, detecting intrusion, and notifying other nodes in the network of impending threats.



(a) (b)

Figure 3: The proposed fog network (a) Application Scenario of proposed system (b) Simulation of proposed system

3.2 Attacker Model

In this model, we assume that the attacker carries out MITM in the fog layer. The attacker intercepts packets from the IoT, the fog layer, and the cloud. The attacker may have more resources than the fog nodes but fewer resources than the cloud. Also, the attacker may know about the existence of IDS nodes and the protocol they are using. However, he does not know the nature of the interrogation since it was chosen and pre-programmed before the deployment of the nodes.

3.3 Proposed IDPS

As shown in Fig. 3, the IDS nodes observe the network by regularly interrogating the FNs. This is done through the use of interrogation packets. They are encrypted packets that consist of a prime number, which the FNs are expected to process and reply to the IDS node. The IDS node expects the receiver to decrypt the packet and multiply the payload by 2, then encrypt it and send the result back to the interrogator IDS node. Multiplication by 2 is chosen because it is easy. It is done by simply shifting the binary representation of the number to the left by one bit. When the IDS node receives the reply and compares the result with the original payload, it concludes whether the targeted FN is malicious.

Moreover, the IDS also records the round-trip time (RTTpkt) of the interrogation packet. The target FN is considered a malicious node when its RTTpkt exceeds a certain threshold value. Finally, if the IDS node does not receive a reply (i.e., RTTpkt $= \infty$), then it is assumed that the packet has been sent elsewhere in the network. There may be two reasons for such behavior, either the attacker is applying a wormhole attack, or the target node is ignorant of the network protocol.

The algorithm in Fig. 4 is the pseudocode describing how the IDS algorithm works. The IDS nodes send interrogation packets to FN one hope away from them, as shown by Line 8. The IDS

then measures the roundtrip time (RTT). To smooth out the noise, EWMA is used, as shown in Line 15. This technique was not used in our previous paper [1]. Whenever a given time (t_{out}) is exceeded without the IDS node getting a reply, then the target node is deemed malicious. This is described by lines 19–24. However, when the packet is received on time, that is to say, RTT $<$ t_0, but the content is altered, then the target node is considered a malicious node carrying out an eavesdropping attack (see: Line: 31–32). But when the RTT $>$ t_0, this implies that the target node may have done some extra processing on the packet. To find out what the target node did to the packet the content is checked. Line 27 explains that the attack is probably a packet altering attack, or the attacker could not reply with the proper answer because they don't know what to do with it (i.e., lack of context). If the node replies with the correct answer, but the reply came after RTT exceeds t_0, then the target node may have tried to alter other parts of the packet, or it has aborted the altering when it realized the packet is an interrogation packet.

Line 37–42 shows how the IDS node alerts the fog nodes in its region about malicious nodes to ensure intrusion prevention. The IDS node broadcasts the ID of the malicious fog node to the fog nodes it manages. They add the culprit to their blacklists. Any node in the blacklist will be cut-off from the network. However, the IDS nodes still include it in their investigation in case the previous verdict was erroneous. Whenever a blacklisted node is found to be benign, is the IDS node removes from the blacklist. Then it notifies all nodes in its region to do the same.

Tab. 2 shows the Truth table that describes the rules followed by the IDS nodes. The rules help the IDS nodes to determine whether a node is malicious. The rules are as follows: (1) $\Delta >$ t_{out}, where Δ is the instantaneous (EWMA computed) latency of the fog node under investigation, and t_{out} is time beyond which a packet is considered lost. (2) $\Delta > t_0$, where t_0 is the allowable network latency. (3) "$Relay_{pkt}$ received", checks whether the fog node's reply to the investigation is received by the IDS, and (4) $y \neq x \times 2$, checks whether the answer replied by the fog node is correct. The fifth column of the table is the output. It is used to detect an attack. The last column of the table is the comment column that elaborates on why a combination of events is an attack or not an attack. The first five columns are Boolean with "1" representing true, which is affirming the heading of the column as the event that occurred and "0" otherwise.

The algorithm in Fig. 5 shows how the proposed system routes packets from the source to the destination. Nodes that are blacklisted by the IDS send their service requests to the cloud. They send their service to the cloud, because it is assumed that the cloud has abundant resources to protect itself from the attacker. Furthermore, this eliminates packet dropping on the account of the IDS wrongly classifying a benign FN as malicious. However, sending the packet to the cloud increases latency and energy consumption. But it is necessary because no FN in the fog layer will communicate with a supposed malicious node, as shown in Line 3. However, if the next hop is benign and the node is not found in the blacklist, then the packet is forwarded to it. The packet is forwarded until it reaches its destination. Since the nodes are deployed in a grid manner, packets are routed along the Y-axis towards the destination. Then the packets move along the X-axis until they reach their destination. This provides us with the shortest path since there are no diagonal connections in the network. Regarding the deployment of the proposed system, the IDS nodes acquire the key from the cloud and distribute them to the FNs. From then, on all packets are encrypted (excluding header) to prevent intrusion.

```
1: t_out ← timeout          \\t_out and t_0 To be set by network administrator
2: t_0 ← allowed Delay
3: λ ← degree of weighing \\an adjustable smoothing parameter 0 ≤ λ ≤ 1
4: for each fog node i do
5:   t_1 ← time()                                   \\get current time
6:   t_2 ← 0
7:   pkt ← x                          \\x is randomly generated prime number
8:   IDS sends pkt to fog node i      \\"i" is an FN one hop from the IDS
9:   repeat
10:    t_2 ← time()          \\While waiting for packet keep checking time
11:    dT ← t_2 − t_1
12:    if (t_1 = 0) then
13:      T_t ← t_0
14:    else
15:      T_t ← λdT + (1 − λ)T_(t−1)
16:    end if
17:    Δ ← T_t
18:  until (Reply_pkt not received OR Δ < t_out )
19:  if ( Δ > t_out) then
20:    if (Reply_pkt not received) then
21:      \\Once Δ > t_out IDS stops listening for Reply_pkt
22:      attack is possibly wormhole
23:    end if
24:  else
25:    \\Fog Node should change x to 2x and reply to IDS
26:    y ← Extract fog Node's answer from Reply_pkt
27:    if (Δ > t_0 AND Reply_pkt is received AND y ≠ x × 2) then
28:      attack possibly packet altering or lack of context
29:    else if (Δ > t_0 AND Reply_pkt is received AND y = x × 2) then
30:      attack possibly packet altering
31:    else if (Δ < t_0 AND Reply_pkt is received AND y ≠ x × 2) then
32:      attack possibly eavesdropping
33:    else
34:      fog node is not compromised
35:    end if
36:  end if
37:  if fog node i is malicious AND not on blacklist then
38:    insert i to blacklist
39:    Notify all neighbours to blacklist i
40:  else if fog node i is not malicious AND is on blacklist then
41:    delete i from blacklist
42:    Notify all neighbours to remove blacklist i
43:  end if
44: end for
```

Figure 4: Algorithm for the proposed IDS node

4 Results

We simulate the proposed system using OMNET++ [72]. We list the simulation parameters in Tab. 3. We run the simulation for one thousand (1000) seconds, with a 95% confidence interval. Experiments are carried out to investigate the accuracy of the system developed in [1] and the accuracy of the proposed system with Exponentially Weighted Moving Average (EWMA). In

addition, we compare the energy and latency of the two systems. The experiments are carried out using the deployment shown in Fig. 3b. To fairly compare the two systems, we use the parameters in [1]. For clarity, the technique used in [1] shall henceforth be referred to as No EWMA or NEWMA, while the technique in this paper shall be referred to as EWMA.

In Eqs. (1)–(4) we calculate the following:

T_{crypto}: the latency for encryption or decryption,

P_{run}: the power consumption of the MCU,

E_{crypto}: the energy consumed due to encryption or decryption and

P_{sleep}: the power consumption of the MCU when the Transceiver is sleeping.

$$\begin{aligned} T_{crypto} &= \frac{N_{crypto}}{f} \\ &= 7,429/(4 \times 10^6) \\ &\simeq 1.86 \simeq \end{aligned} \tag{1}$$

$$\begin{aligned} P_{run} &= I_{run} \times f \times V \\ &= (140 \times 10^{-6}) \times 4 \times 3 \\ &\simeq 1.68 \simeq \end{aligned} \tag{2}$$

Table 2: Truth table for the IDS

$\Delta > t_{out}$	$\Delta > t_0$	Reply_pkt received	$y \neq x \times 2$	Attack	Comment
0	0	0	0	0	Waiting for Reply_pkt
0	0	0	1	x	Impossible, Reply_pkt is not yet received
0	0	1	0	0	Node is safe
0	0	1	1	1	Attack, eavesdropping, or lack of context
0	1	0	0	x	Waiting for Reply_pkt
0	1	0	1	x	Impossible, Reply_pkt is not yet received
0	1	1	0	1	Attack, possibly content altering
0	1	1	1	1	Attack, altering the content or lack of context
1	0	0	0	x	Not possible, $t_{out} > t_0$
1	0	0	1	x	Not possible, $t_{out} > t_0$
1	0	1	0	x	Not possible, $t_{out} > t_0$
1	0	1	1	x	Not possible, $t_{out} > t_0$
1	1	0	0	1	Attack, possibly wormhole attack
1	1	0	1	x	Impossible, Reply_pkt is not received
1	1	1	0	x	Impossible, if $\Delta > t_{out}$, Reply_pkt is ignored
1	1	1	1	x	Impossible, if $\Delta > t_{out}$, Reply_pkt is ignored

```
1: if (Current node is Blacklisted by IDS) then
2:    Send request to Cloud
3: else if Next hop is Blacklisted by IDS then
4:    Drop Packet
5: else
6:    Send to next hop
7: end if
```

Figure 5: Routing algorithm

$$\begin{aligned} E_{crypto} &= P_{run} \times T_{crypto} \\ &= (1.68 \times 10^{-3}) \times (1.86 \times 10^{-3}) \\ &\simeq 3.12\ \mu\text{J} \end{aligned} \tag{3}$$

$$\begin{aligned} P_{sleep} &= I_{sleep} \times f \times V \\ &= (37 \times 10^{-6}) \times 4 \times 3 \\ &\simeq 0.44 \simeq \end{aligned} \tag{4}$$

4.1 Validation

To validate the simulation, we use Eq. (5) to calculate the total time and it should be equal to the simulation time, where:

P: packet size
N_{f_tx}, N_{f_rx}: the number of transmissions and receptions in the fog layer respectively,
N_{c_tx}, N_{c_rx}: the number of transmissions and receptions in the cloud layer respectively,
R_{f_tx}, R_{c_tx}: transmission rates (Bps) for the fog and the cloud respectively,
ρ_p: The Cumulative Time For The Processing And
T_s: the cumulative time spent by the system sleeping.

Table 3: Simulation parameters

SN	Parameter	Value	Comment
1	Packet size (P)	1500 byte	Packet size in bytes
2	Process delay (ρ_p)	0.5 s	Time needed to process the data (seconds)
3	Investigate time ($t_{investigate}$)	2 s	Time it takes the IDS node to investigate its fog nodes' behavior
4	Packet time-out (t_{out})	0.5 s	Number of seconds to consider packet as lost
5	Voltage (V)	3.0 v	Voltage powering the system
6	Idle (I_i)	320 μA	Transceiver idle listening current requirement
6	Data Rate at Fog (R_f)	100 Mbps	Bandwidth of network at the fog and the IoT layer and between the two
7	Data Rate at Cloud (R_c)	10 Mbps	Bandwidth of network between the fog and the cloud layer
8	Transmission power (I_{tx})	19.3 mA	Transceiver transmission power
9	Reception power (I_r)	21.5 mA	Transceiver reception power
10	MCU Clock (f)	4.0 MHz	MCU processor is in MHz
11	Encryption decryption cycle (N_{crypto})	7,429 cycles	Using AES, the clock cycles needed for encryption and decryption are 6,637 ~cycles and 7,429 ~cycles respectively. Max was chosen for the worst-case scenario
12	MCU Running current requirement (I_{run})	140 μA	Current (A) needed per MHz while MCU processes data
13	MCU sleeping current requirement (I_{sleep})	37 μA	Current (A) needed per MHz while MCU sleeps

$$Total \quad Time = \left(\frac{P}{R_{f\_tx}}\right)(N_{f\_tx} + N_{f\_rx}) + \left(\frac{P}{R_{c-tx}}\right)(N_{f\_tx} + N_{f\_rx}) + \rho_p + T_s \tag{5}$$

4.2 Accuracy

This paper improves the proposed system in [1] by using EWMA on the IDPS's input. Where the input is the latency of the interrogation packets that are sent to the fog nodes. Eq. (6) describes the EWMA for the IDPS in the proposed systems, where I is the instantaneous input, X(t-1) is the value from previous iterations, X(t) is the value of the present iteration, and λ is the smoothing parameter. The smoothing parameter determines how long the effect of an input should last in terms of iteration, albeit its effect decays exponentially. Fig. 6a shows the impact of EWMA on a unity input with each iteration. The input decays approximately to zero after 50, 24, and 15 iterations, where λ is 0.1, 0.2, and 0.3 respectively. This ability of EWMA to include past interrogation results helps it improve the performance of the proposed system.

$$X(t) = \lambda I + (1 - \lambda)X(t - 1) \tag{6}$$

We compare EWMA with NEWMA, which is similar but does not account for noisy network environments. We use Eq. (7) to calculate the accuracy of the two systems at a PER of 0.1, which is a reasonable estimation for a noisy IoT environment [73,74]. Tab. 4 defines the terms in the equation.

$$Accuracy = (tn + tp)/(tp + tn + fp + fn) \tag{7}$$

In Fig. 6b, it can be seen that the EWMA outperforms NEWMA by 15%. This increase in accuracy is because the EWMA-based system decides includes the previous and the current input when classifying the fog nodes. However, it should be noted that the system accuracy degrades with an increase in λ. As λ increases, only more recent interrogation results are considered.

Another factor that affects the accuracy of the two systems is the noisiness of the network. Fig. 6c investigates the accuracy of the proposed system in terms of PER. EWMA performs better than NEWMA, however, both systems degrade with an increase in PER. Fig. 6d shows that the number of malicious MITM nodes in the networks has little or no effect on both systems.

4.3 Energy

Energy consumption is an important performance metric, especially for resource-constraint fog computing applications. NEWMA was simulated with a PER of 10% and then with a PER of 40%. Then, EWMA is simulated with all combinations of λ and PER for the values of λ = 0.1, λ = 0.4, PER = 10%, and PER = 40%. These experiments are designed to investigate whether the proposed system can withstand high PER.

Fig. 7a shows the total energy consumption of all the fog nodes in the network. It can be seen that the energy consumption rate (i.e., the slope) of the total energy consumed by fog nodes is the same in all cases where the PER is 0.1. However, the energy consumption rate increases as the PER increases to 0.4. Thus, we can conclude that the energy overhead of both networks is solely due to PER. The good news here is that adding EWMA to the system has improved the system accuracy without incurring energy overhead on the network.

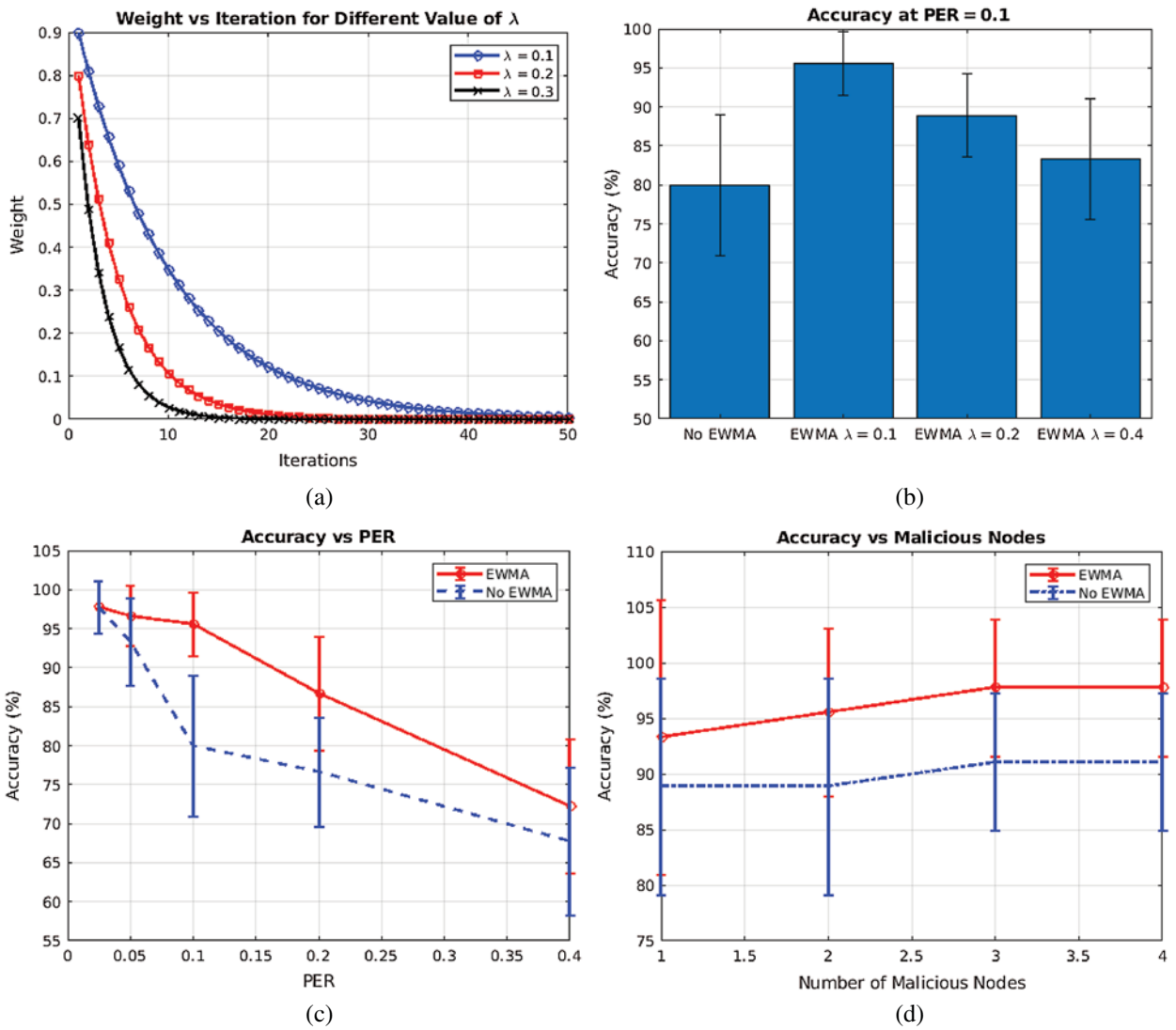


Figure 6: Accuracy of the proposed system compared to the accuracy of NEWMA (a) Decay of a value with iteration in EWMA (b) Accuracy of the system with change in λ (c) Accuracy at different packet error rate (PER), (d) Accuracy with number of malicious nodes

Table 4: Definition of terms

Variables	Definition according to ISO/IEC [75]
True positive (tp)	IDPS alert when there is an attack
True negative (tn)	No IDPS alert when there is no attack
False negative (fn)	No IDPS alert when there is an attack
False positive (fp)	IDPS alert when there is no attack

In Fig. 7b, it can be seen that NEWMA has a higher energy consumption rate at PER = 0.1 than at PER = 0.4. This anomaly is explained by Eq. (8): Since packet time-out (t_{out}) »

transmission time (t_{tx}), then the smaller the value of PER, the larger the number of IDS investigation transmissions (N) to the fog nodes. In other words, when the channel is noisy, the IDS takes longer to finish an investigation round, because it has to wait for t_{out} seconds whenever a packet is lost, making the investigation session long. Hence, reducing the total number of investigation rounds and by extension, reducing the number of IDS's transmissions. Next, to support this hypothesis, we carry out two identical simulations with PER = 0.1 and PER = 0.4, each for 1000 sec. The number of IDS investigation sessions are 467.8±1.62 and 395.8±1.84 at PER = 0.1 and PER = 0.4, respectively.

$$N = T_{simulation}/[(1 - PER).t_{tx} + PER.t_{out}] \tag{8}$$

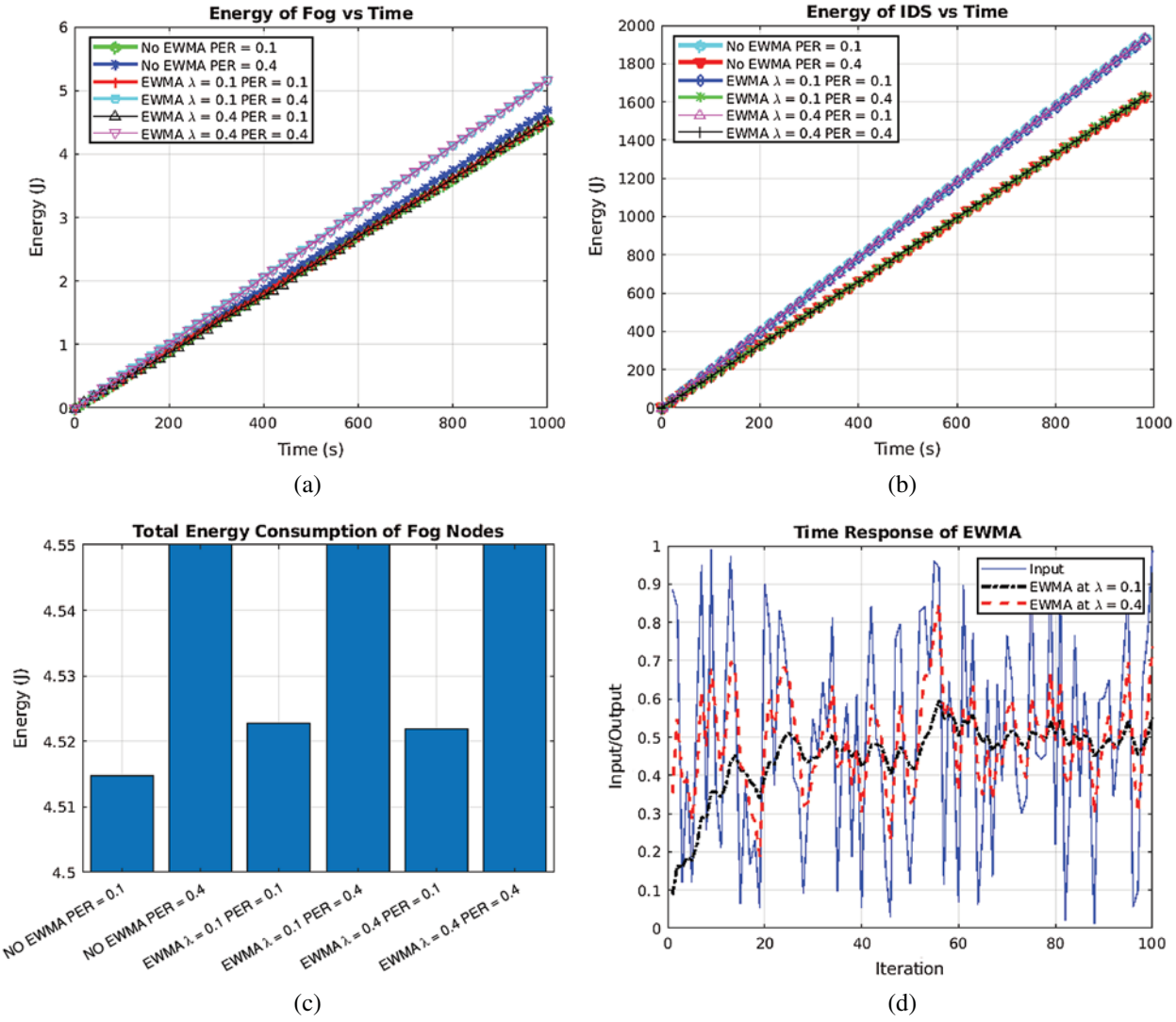


Figure 7: Energy consumption of the proposed system compared to NEWMA (a) Energy consumption of Fog nodes (b) Energy consumption of IDS nodes (c) Total energy consumed by Fog nodes (d) Time response illustration of EWMA

For the same reason, the EWMA-based system with $\lambda = 0.1$ and PER $= 0.1$ ($EWMA_{(\lambda=0.1 PER=0.1)}$), and EWMA where $\lambda = 0.4$ and PER $= 0.1$ ($EWMA_{(\lambda=0.4,PER=0.1)}$) have the same energy consumption rate as NEWMA at PER $= 0.1$ as shown in Fig. 7c. In addition, the delay in response of $EWMA_{(\lambda=0.1,PER=0.1)}$ and the high PER of $EWMA_{(\lambda=0.4,PER=0.1)}$ exacerbates the energy consumption overhead. This can be seen in Fig. 7d where EWMA is tested with a uniform random variate which represents noise around the value of 0.5. The test shows $\lambda = 0.1$ gets closer to the mean value of 0.5 than $\lambda = 0.4$, which oscillates. However, the accuracy of $\lambda = 0.1$ is at the expense of increased rise time. As shown in the figure, it can be seen that when $\lambda = 0.1$, it takes at least 20 iterations for the system to arrive at the mean.

4.4 Latency

One of the applications of fog computing is to reduce latency. The latency of service (LS) and the latency of detecting attacks (LDA) are investigated. The LS is the time it takes a request from the IoT layer to be serviced by fog nodes in the fog layer, while LDA is the time it takes the IDS node to detect an attack. The two latencies are investigated for NEWMA and EWMA with PER $= 0.1$ and PER $= 0.4$, and smoothing parameters $\lambda = 0.1$ and $\lambda = 0.4$.

Fig. 8a shows that both EWMA and the PER affect the latency of the proposed system. At the same PER, it can be seen that the system without EWMA has less latency than that with EWMA. Also, in the proposed system, we observe that the lower the PER the better the performance. In fact, when one observes the average latency of each experiment, it is clear that the smoothing parameter (λ) has little to no effect on the latency of service of the system.

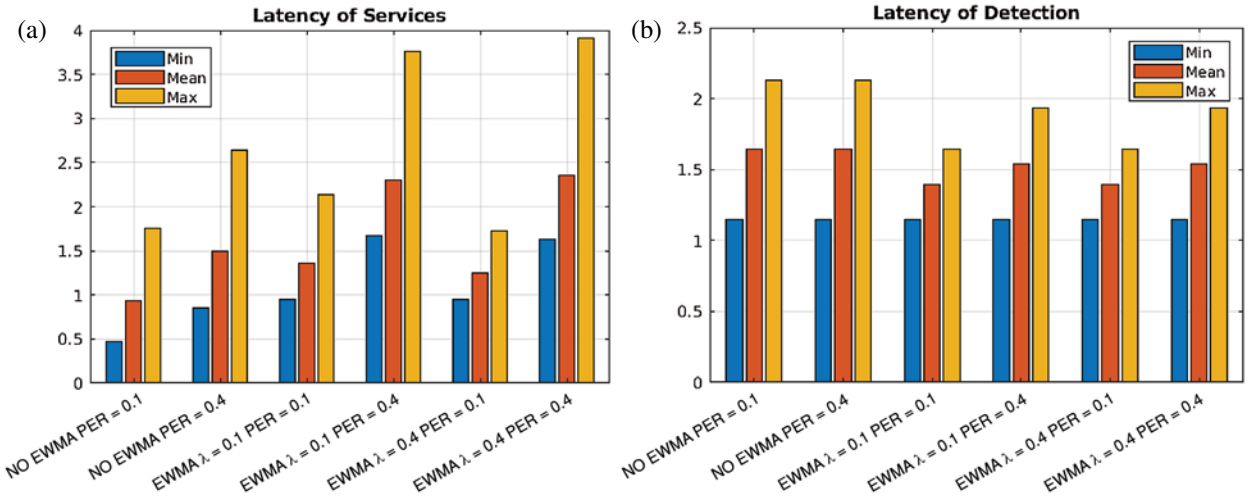


Figure 8: Latency of the proposed system compared to NEWMA (a) Latency of Services (b) Latency of Detection

In Fig. 8b shows that the average latency of the proposed system is slightly better with 0.5 s less in $EWMA_{(\lambda=0.1,PER=0.1)}$ and $EWMA_{(\lambda=0.4,PER=0.1)}$ than in $NEWMA_{(\lambda=0.1)}$ and $NEWMA_{(\lambda=0.4)}$. Moreover, we learn that the PER is inversely proportional to the latency of detection. This notion makes sense, since PER is the noise that prevents the system from accurately detecting an attack. Thus, the system needs more samples (of investigation) to make a detection. Hence, making the attack detection slower.

5 Conclusion

In this paper, an intrusion detection system for Man-in-the-Middle (MITM) attack is proposed. The system detects attacks through special intrusion detection nodes called IDS nodes. The IDS node periodically probes the fog nodes in its region by sending interrogation packets. The fog nodes must answer the question in the interrogation packet and reply immediately. The IDS node checks the answer given by the fog nodes and the roundtrip time (RTT) of the communication to determine whether the fog node under investigation is malicious. Since one of the factors considered by the IDS node is the reply time of the fog nodes, a noisy network may affect IDS nodes decisions. As such, an Exponentially Weighted Moving Average (EWMA) technique is used to overcome the noisy nature of the network. The simulation results showed that EWMA improves the accuracy of the system by 15% and can detect the intrusion 0.25–0.5 s faster than that without EWMA. However, the use of EWMA affects the latency of services provided by the fog nodes by at least 0.75–1.3 s. Finally, the energy consumption of the system shows identical behavior with and without EWMA. This lack of energy overhead is because EWMA is a lightweight technique consisting of only two multiplications and an addition.

Funding Statement: The Authors would like to acknowledge the support of King Fahd University of Petroleum and Minerals for this research.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] F. Aliyu, T. Sheltami and E. M. Shakshuki, "A detection and prevention technique for man in the middle attack in Fog Computing," in *Proc. Procedia Computer Science*, vol. 141, pp. 24–31, 2018.

[2] J. Gervais, "The future of IoT: 10 predictions about the Internet of Things," Mountain View, CA, USA: NortonLifeLock, 2020. [Online]. Available: https://nr.tn/3lXTWHt.

[3] J. Covitz, "Edge Computing: Making the server room mission critical," *IDG Communications*, 2019, [Online]. Available: https://www.networkworld.com/article/3340829/edge-computing-making-the-server-room-mission-critical.html.

[4] J. Wu, M. Dong, K. Ota, J. Li and Z. Guan, "FCSS: Fog-computing-based content-aware filtering for security services in information-centric social networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 4, pp. 553–564, 2019.

[5] O. A. Nasr, Y. Amer and M. Abobakr, "The 'Droplet': A new personal device to enable fog computing," in *Proc. FMEC*, Barcelona, Spain, pp. 93–99, 2018.

[6] M. Heisel, W. Joosen, J. López and F. Martinelli, *Engineering Secure Future Internet Services and Systems: Current Research*. London, UK: Springer International Publishing, 2014. [Online]. Available: https://www.springer.com/gp/book/9783319074511.

[7] A. Rahman, E. Hassanain and M. S. Hossain, "Towards a secure mobile edge computing framework for Hajj," *IEEE Access*, vol. 5, pp. 11768–11781, 2017.

[8] A. Hosseinian-Far, M. Ramachandran and C. L. Slack, "Emerging trends in cloud computing, big data, fog computing, IoT and smart living," in *Technology for Smart Futures*, London, UK: Springer International Publishing, pp. 29–40, 2017.

[9] G. Raja and A. Thomas, "SAFER: Crowdsourcing based disaster monitoring system using software defined fog computing," *Mobile Networks and Applications*, vol. 24, no. 5, pp. 1414–1424, 2019.

[10] Y. Jiang, Z. Huang and D. H. K. Tsang, "Challenges and solutions in fog computing orchestration," *IEEE Network*, vol. 32, no. 3, pp. 122–129, 2018.

[11] Z. Ashi, M. Al-Fawa'reh and M. Al-Fayoumi, "Fog computing: Security challenges and countermeasures," *International Journal of Computer Applications*, vol. 175, no. 15, pp. 30–36, 2020.

[12] A. Ceccarelli, M. Cinque, C. Esposito, L. Foschini, C. Giannelli *et al.*, "FUSION—Fog computing and blockchain for trusted Industrial Internet of Things," *IEEE Transactions on Engineering Management*, vol. 14, no. 8, pp. 1–15, 2020.
[13] K. Gu, N. Wu, B. Yin and W. Jia, "Secure data query framework for cloud and fog computing," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, 2020.
[14] A. Mallik, "Man-in-the-middle-attack: Understanding in simple words," *Cyberspace: Jurnal Pendidikan Teknologi Informasi*, vol. 2, no. 2, pp. 109–134, 2019.
[15] J. Shen, T. T. Yuen, K. R. Choo and Q. Zeng, "AMOGAP: Defending against man-in-the-middle and offline guessing attacks on passwords," in *Information Security and Privacy*, London, UK: Springer International Publishing, pp. 514–532, 2019.
[16] B. N. B. Ekanayake, M. N. Halgamuge and A. Syed, "Review: Security and privacy issues of fog computing for the Internet of Things (IoT)," in *Cognitive Computing for Big Data Systems Over IoT: Frameworks, Tools and Applications*, London, UK: Springer International Publishing, pp. 139–174, 2018.
[17] S. Goundar, S. B. Bhushan and P. K. Rayani, *Architecture and Security Issues in Fog Computing Applications*, PN, USA: IGI Global, 2019.
[18] R. A. Sowah, K. B. Ofori-Amanfo, G. A. Mills and K. M. Koumadi, "Detection and prevention of man-in-the-middle spoofing attacks in MANETs using predictive techniques in Artificial Neural Networks (ANN)," *Journal of Computer Networks and Communications*, vol. 2019, pp. 1–14, 2019.
[19] K. Lee, D. Kim, D. Ha, U. Rajput and H. Oh, "On security and privacy issues of fog computing supported Internet of Things environment," in *Proc. Network of the Future (NOF)*, Montreal, Quebec, Canada, pp. 1–3, 2015.
[20] A. Bhattacharyya, A. Banerjee, D. Bose, H. N. Saha and D. Bhattacharya, "Different types of attacks in Mobile ADHOC network," arXiv preprint arXiv:1111.4090, 2011.
[21] T. Eisenbarth and S. Kumar, "A survey of lightweight-cryptography implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 522–533, 2007.
[22] F. Harrou, B. Bouyeddou, Y. Sun and B. Kadri, "Detecting cyber-attacks using a CRPS-based monitoring approach," in *Proc. SSCI*, Bangalore, India, pp. 618–622, 2019.
[23] N. Ye, S. Vilbert and Q. Chen, "Computer intrusion detection through EWMA for autocorrelated and uncorrelated data," *IEEE Transactions on Reliability*, vol. 52, no. 1, pp. 75–82, 2003.
[24] ISO and IEC, "Information technology — security techniques — selection, deployment and operations of intrusion detection and prevention systems (IDPS)," in *ISO/IEC Standard*, Geneva, Switzerland: ISO/IEC, pp. 1–48, 2016. [Online]. Available: https://www.iso.org/standard/56889.html.
[25] V. Ramachandran and S. Nandi, "Detecting ARP spoofing: An active technique," in *Information Systems Security*, Heidelberg, Berlin, Germany: Springer International Publishing, pp. 239–250, 2005.
[26] I. Alsmadi, R. Burdwell, A. Aleroud, A. Wahbeh, M. A. Al-Qudah *et al.*, "Network security," in *Practical Information Security*, London, UK: Springer International Publishing, pp. 121–138, 2018.
[27] D. Bruschi, A. Ornaghi and E. Rosti, "S-ARP: A secure address resolution protocol," in *Proc. ACSAC*, Las Vegas, NV, USA, pp. 66–74, 2003.
[28] M. Barnaba, "Anticap," in *Antifork Research*, Inc., Italy: Hacker Researcher Virtual Lab, 2003, [Online]. Available: http://cvs.antifork.org/cvsweb.cgi/anticap.
[29] I. Teterin, "Antidote," in *Security Focus, Mountain View*, CA, USA: Symantec Connect, vol. 1, 2010. [Online]. Available: https://www.securityfocus.com/archive/1/299929.
[30] Cisco Headquarters, *Catalyst 4500 Series Switch Cisco IOS Software Configuration Guide*. San Jose, CA, USA: Cisco Systems, Inc., 1999. [Online]. Available: https://bit.ly/3cMJQXH.
[31] C. Calvert, T. M. Khoshgoftaar, M. M. Najafabadi and C. Kemp, "A procedure for collecting and labeling man-in-the-middle attack traffic," *International Journal of Reliability, Quality and Safety Engineering*, vol. 24, no. 1, pp. 1750002, 2017.
[32] Y. Bhaiji, Understanding, preventing, and defending against layer 2 attacks, San Jose, CA, USA: Cisco Systems, Inc., 2007. [Online]. Available: http://www.nanog.org/meetings/nanog42/presentations/Bhaiji_Layer_2_Attacks.pdf.

[33] F. Gontharet, "Man-in-the-middle attacks and countermeasures analysis," *MS Thesis*, University of Abertay Dundee, Scotland, 2015.
[34] M. Plch, "Practical man-in-the-middle attacks in computer networks," *BSc Thesis*, Masaryk University, Czech Republic, 2015.
[35] I. Green, "DNS Spoofing by the man in the middle," *Technical Report, SANS Institute*, Bethesda, Maryland, USA, 2005. [Online]. Available: https://bit.ly/3oReguh.
[36] C. Grothoff, M. Wachs, M. Ermert and J. Appelbaum, "Towards secure name resolution on the Internet," *Computers & Security*, vol. 77, no. 1, pp. 694–708, 2018.
[37] K. Chetioui, G. Orhanou and S. El-hajji, "New protocol E-DNSSEC to Enhance DNSSEC Security," *International Journal of Network Security*, vol. 20, no. 1, pp. 18–23, 2018.
[38] M. Conti, N. Dragoni and V. Lesyk, "A survey of man in the middle attacks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2027–2051, 2016.
[39] C. Russell, "Example of IGP exploits," *Information Security Alliance*, pp. 1–10, 2001. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi= 10.1.1.552.3337&rep=rep1&type=pdf.
[40] X. Liu, A. Li, X. Yang and D. Wetherall, "Passport: Secure and adoptable source authentication," in *Proc. NSDI*, pp. 365–378, 2008.
[41] H. Lee, M. Kwon, G. Hasker and A. Perrig, "BASE: An incrementally deployable mechanism for viable IP spoofing prevention," in *Proc. ACM symp. on Information, Computer and Communications Security*, Singapore, pp. 20–31, 2007.
[42] A. Bremler-Barr and H. Levy, "Spoofing prevention method," in *Proc. Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, vol. 1, pp. 536–547, 2005.
[43] J. Li, J. Mirkovic, M. Wang, P. Reiher and L. Zhang, "SAVE: Source address validity enforcement protocol," in *Proc. Annual Joint Conf. of the IEEE Computer and Communications Societies*, New York, NY, USA, vol. 3, pp. 1557–1566, 2002.
[44] M. G. M. Santos and P. A. A. Marcillo, "Security in the data link layer of the OSI model on LANs wired Cisco," *Journal of Science and Research: Revista Ciencia e Investigación*, vol. 3, no. CITT2017, pp. 106–112, 2018.
[45] A. Mustafa, N. Siddique and M. Zubair, "Data link layer security problems and solutions," *BSc Thesis*, Halmstad University, UK, 2015. [Online]. Available: http://www.diva-portal.org/smash/get/diva2:783188/FULLTEXT01.pdf.
[46] S. Khan, S. Parkinson and Y. Qin, "Fog computing security: A review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 19, 2017.
[47] D. Wong, "VOSviewer," *Technical Services Quarterly*, vol. 35, no. 2, pp. 219–220, 2018.
[48] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat *et al.*, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.
[49] M. Mohanapriya and I. Krishnamurthi, "Modified DSR protocol for detection and removal of selective black hole attack in MANET," *Computers & Electrical Engineering*, vol. 40, no. 2, pp. 530–538, 2014.
[50] B. Aziz and G. Hamilton, "Detecting man-in-the-middle attacks by precise timing," in *Proc. International Conference on Emerging Security Information, Systems and Technologies*, Athens, Greece, pp. 81–86, 2009.
[51] D. B. Faria and D. R. Cheriton, "Detecting identity-based attacks in wireless networks using signalprints," in *Proc. ACM Workshop on Wireless Security*, Los Angeles, CA, USA, pp. 43–52, 2006.
[52] A. S. K. Pathan, *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*. Boston, MA, USA: CRC Press, 2016.
[53] S. M. Glass, V. Muthukkumarasamy and M. Portmann, "Detecting man-in-the-middle and wormhole attacks in wireless mesh networks," in *Proc. Int. Conf. on Advanced Information Networking and Applications*, Bradford, England, UK, pp. 530–538, 2009.
[54] W. Wang and T. Minohara, "Poster: Wormhole attacks on asynchronous duty-cycling sensor networks and their countermeasures," in *Proc. Int. Conf. on Embedded Wireless Systems and Networks (EWSN)*, Madrid, Spain, pp. 183–184, 2018.

[55] G. Thamilarasu and S. Chawla, "Towards deep-learning-driven intrusion detection for the Internet of Things," *Sensors*, vol. 19, no. 9, pp. 1977, 2019.

[56] A. Xingshuo, Z. Xianwei, L. Xing, L. Fuhong and Y. Lei, "Sample selected extreme learning machine-based intrusion detection in fog computing and MEC," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–10, 2018.

[57] B. S. Khater, A. A. Wahab, M. Idris, M. A. Hussain and A. A. Ibrahim, "A lightweight perceptron-based intrusion detection system for fog computing," *Applied Sciences*, vol. 9, no. 1, pp. 178, 2019.

[58] G. Blokdyk, *Intrusion Prevention System: A Complete Guide - 2019 Edition*. Aspley, Queensland, Australia: Emereo Pty Limited, 2019.

[59] A. S. K. Pathan, *The State of the Art in Intrusion Prevention and Detection*. Boston, MA, USA: CRC Press, 2014.

[60] K. Zhao and L. Ge, "A survey on the internet of things security," in *Proc. Computational Intelligence and Security (CIS)*, Emeishan, China, pp. 663–667, 2013.

[61] P. I. Radoglou-Grammatikis and P. G. Sarigiannidis, "Securing the smart grid: A comprehensive compilation of intrusion detection and prevention systems," *IEEE Access*, vol. 7, pp. 46595–46620, 2019.

[62] R. Chapaneri and S. Shah, "A comprehensive survey of machine learning-based network intrusion detection," in *Smart Intelligent Computing and Applications*, Singapore: Springer, pp. 345–356, 2019.

[63] J. Liu, Y. Xiao and C. L. P. Chen, "Authentication and access control in the internet of things," in *Proc. Distributed Computing Systems Workshops (ICDCSW)*, Macau, China, pp. 588–592, 2012.

[64] I. Stojmenovic and S. Wen, "The Fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. on Computer Science and Information Systems*, Warsaw, Poland, pp. 1–8, 2014.

[65] A. A. Diro, N. Chilamkurti and N. Kumar, "Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog Computing," *Mobile Networks and Applications*, vol. 22, no. 5, pp. 1–11, 2017.

[66] J. Eriksson, S. V. Krishnamurthy and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack in wireless networks," in *Proc. IEEE ICNP'06*, Santa Barbara, CA, USA, pp. 75–84, 2006.

[67] S. Qaisar, S. Qaisar and A. Basit, "Software defined machine learning based anomaly detection in fog based IoT network," in *Proc. ICCSA*, Saint Petersburg, Russia, pp. 611–621, 2019.

[68] K. Doshi, M. Mozaffari and Y. Yilmaz, "RAPID: Real-time anomaly-based preventive intrusion detection," in *Proc. ACM Workshop on Wireless Security and Machine Learning*, Miami, FL, USA, pp. 49–54, 2019.

[69] U. Verma and D. Bhardwaj, "Security challenges for fog computing enabled Internet of Things from authentication perspective," *International Journal of Computational Intelligence & IoT*, vol. 2, no. 1, pp. 382–387, 2019.

[70] NXP, LPC55S6x MCU Family. Austin, TX, USA: NXP, 2019. [Online]. Available: https://www.nxp.com/docs/en/fact-sheet/LPC55S6XFS.PDF.

[71] D. P. Jablon, "Strong password-only authenticated key exchange," *ACM SIGCOMM Computer Communication Review*, vol. 26, no. 5, pp. 5–26, 1996.

[72] A. Varga, A Quick Overview of the OMNeT ++ IDE. Omnet++, 2017. [Online]. Available: https://doc.omnetpp.org/omnetpp/UserGuide.pdf.

[73] S. Ahmadi, "Performance of IEEE 802.16m and 3GPP LTE-advanced," in *Mobile WiMAX: A Systems Approach to Understanding IEEE 802.16m Radio Access Technology*, 1st ed., Burington, MA, USA: Elsevier, pp 657–722, 2011.

[74] Y. Alper, T. Kramp, P. Dufour, R. Gupta, R. Soss *et al.*, "3-LoRaWAN protocol: specifications, security, and capabilities," in *LPWAN Technologies for IoT and M2M Applications*, Cambridge, MA, USA: Academic Press, pp. 37–63, 2020.

[75] Committee ISO/IEC JTC 1/SC 27, ISO/IEC 27039:2015 (en) information technology—security techniques—selection, deployment and operations of intrusion detection and prevention systems (IDPS), Vernier, Geneva, Switzerland: ISO/IEC, 2015. [Online]. Available: https://www.iso.org/standard/56889.html.