Asian Journal of Research in Computer Science

Volume 17, Issue 4, Page 1-18, 2024; Article no.AJRCOS.112490
ISSN: 2581-8260

An Overview of Iot Architecture Security Issues and Countermeasures

Walla Khalaifat [a*]

[a] *College of Information Technology, University of Bahrain, Bahrain.*

Author's contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/AJRCOS/2024/v17i4427

Open Peer Review History:
This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here: https://www.sdiarticle5.com/review-history/112490

Review Article

Received: 27/11/2023
Accepted: 01/02/2024
Published: 07/02/2024

ABSTRACT

The Internet of Things (IoT) has revolutionized the way we interact with our surroundings. As the number of IoT devices continues to increase, along with limited resources and diverse technologies, the risk of security attacks increases. Therefore, it is important to integrate security measures throughout the development process and system architecture. However, it is crucial to continually assess and update security measures to avoid emerging threats and ensure the confidentiality, integrity, and availability of IoT systems. This study aims to explore security issues in the IoT, highlighting the associated challenges. It examines various threats, attacks, and vulnerabilities that arise within a three-layer architecture and discusses potential solutions to enhance security in each layer. By addressing these concerns, it is possible to establish a secure and reliable foundation for expanding IoT systems.

Keywords: Internet of things; security issues; attacks; countermeasures; layer architecture; privacy.

1. INTRODUCTION

In the 21st century, the Internet of Things (IoT) has become one of the most important technologies. It connects billions of things around the world, enabling communication between everyone and everything. According to [1], the director of MIT, the phrase "Internet of Things"

**Corresponding author: E-mail: walaakhalaifat@hotmail.com, 20072182@stu.uob.edu.bh;*

originally coined by him in 1999. This phrase is used to reflect his vision of connecting different electronic devices through a network, where every electronic device is tagged with the data corresponding to it.

IoT is described as an interaction between the physical and digital worlds that employs several types of sensors and actuators [2]. IoT is described in [3] as a paradigm in which networking and computing capabilities are integrated into various objects. The fundamental concept of this advanced technology is to connect devices through the Internet for automation purposes and to collaborate to perform complex tasks that require a high level of intelligence and connectivity. The authors of [4] believe that the IoT is a combination of different technologies that work together, including specialized actuators, sensors, processors, and transceivers that collect, analyze, and process information to provide accurate results to users. In IoT, with every "object" connected, there is a risk of significant security threats targeting the service and data.

Security has recently become the most important concern in the development of the IoT. In general, the IoT is a complex system that incorporates various heterogeneous devices, networks, and applications, making it challenging to establish a reliable system. Moreover, the use of technologies such as RFID, sensors, embedded systems, and nanotechnology further complicates the task of ensuring data security in the IoT. Therefore, security vulnerabilities can have disastrous consequences with the widespread implementation of IoT, which is used in many fields such as environmental monitoring, home automation, transportation, and healthcare.

Therefore, IoT appliances affect our daily lives in several ways. These IoT appliances are exposed to security threats, which are the biggest concerns of the Internet of Things (IoT). Cybersecurity and privacy risks are the main concerns in today's world, especially with the rise of heterogeneous technologies and large amounts of heterogeneous data that are difficult to manage [5]. These risks may cause massive damage, such as the loss of important data. Due to the limited resources of IoT devices, lightweight algorithms are usually the preferred way to balance greater security with the lower capabilities of IoT systems. According to [6], among the multiple security challenges that must be overcome, it is necessary to address the following:

- **Data Security and Privacy**

It is important to secure and hide data to prevent theft and unauthorized access by hackers, while also ensuring that data can be transmitted seamlessly.

- **Technical Concern and Common Standards**

IoT devices can generate a large amount of data. It is challenging to store, secure, and analyze data. As the number of devices increased, the amount of traffic generated also increased. Consequently, the network should be able to manage a high density of devices and a large volume of traffic. The system should also be able to distinguish between permitted and rogue devices [7]. Moreover, there are several standards for IoT devices and many IoT companies. However, there is no industry-wide acceptance of a unified standard, which is a significant challenge [6]. Therefore, the most challenging factor is connecting authorized and unauthorized devices while there is a lack of unified standards.

- **Security attacks and System Vulnerability**

Security in IoT systems focuses on different security challenges, such as how to design guidelines for the security of a network and different security frameworks. IoT applications require application security and network security to secure IoT communication networks to connect different IoT devices [8].

- **Social and Legal Concerns**

It is impossible to address these social and legal concerns using a single mechanism. However, it is likely that users will choose various applications, and each application will have a large number of users. Therefore, it is crucial that a proper authentication mechanism is implemented to prevent illegal users from entering the system and taking control of the devices.

As the number of users utilizing IoT devices continues to increase, the probability of a cyberattack also increases. This is compounded by the fact that IoT devices often have limited resources and employ diverse technologies that can introduce security vulnerabilities to the entire IoT system. To address these concerns effectively, security must be incorporated throughout the development process and the system architecture. It should also be maintained on an ongoing basis to ensure the confidentiality, integrity, and availability of IoT systems. Therefore, it is crucial to view an IoT system as an entity, where security should be considered as a chain, with the weakest link potentially compromising the entire system. Consequently, several studies have been conducted to examine various attacks and vulnerabilities, along with mitigation strategies specific to each layer of the architecture. However, limited research has been conducted to develop a comprehensive understanding of security issues and countermeasures across all layers of the IoT architecture in a cohesive manner. To address this gap, it is necessary to gain a more comprehensive understanding of the security challenges present in each layer of the IoT architecture as a complete system. Filling this research gap is crucial for establishing a stronger and more robust security framework for IoT systems by focusing on the complementary nature of these layers. To fill this gap, the present study focuses on addressing the primary question, "What are the security challenges encountered within each layer of the IoT architecture and what are the corresponding countermeasures to address them effectively?"

This study aimed to explore the security challenges associated with the three-layer architecture of IoT systems. It seeks to identify vulnerabilities and potential threats that can compromise the security of the entire IoT system. In addition, it explores possible mitigation strategies to overcome these vulnerabilities and potential threats. By addressing this research question, this study can develop a comprehensive understanding of the security issues across all layers of the IoT architecture and subsequently propose effective measures to establish a stronger and more robust security framework for IoT systems. The contribution of this research is to explore the security challenges and countermeasures specifically related to the three-layer architecture of IoT systems. Therefore, this research attempts to fill this gap by providing a comprehensive understanding of security issues and countermeasures across all layers of the IoT architecture. These contributions can significantly enhance the overall security and privacy of IoT systems, enabling the deployment of IoT systems with confidence in different scenarios in which security and privacy are highly significant and prioritized.

The main sections of this paper are as follows: Chapter (2) discusses different security principles that need to be implemented to ensure that people, software, processes, and things communicate safely. Chapter (3) discusses the basic architecture of the IoT. Chapter (4) describes some common security attacks that affect the perception layer and their countermeasures. Chapter (5) discusses different security attacks and countermeasures facing the network layer. Finally, Chapter (6) explains the various security attacks that affect the application layer and their countermeasures.

2. SECURITY ISSUES OF THE IOT

In IoT, several smart devices are connected to each other through the Internet to provide different services for everyone, which has a significant impact on our daily lives. However, there are many limitations and restrictions associated with the IoT, including components and devices, computation, and power resources. In addition, IoT systems are subject to privacy and security concerns, including integrity, confidentiality, availability, and authenticity. The following security principles should be implemented to ensure safe communication between people, software, processes, and things.

2.1 Integrity

Integrity must be ensured to ensure the validity of the data. Integrity refers to the protection of information from cybercriminals and external interference during data transmission and storage. The IoT is based on data exchanged between different devices. Data integrity algorithms are important for preventing data alterations [5]. According to [9], data integrity is achieved through error detection methods such as checksum and cyclic redundancy checks, as well as the continuous syncing of data for backup purposes and version control. Another study by [10] considered that Secure Hash Algorithms (SHA) are important mechanisms for ensuring the integrity of data. However, due to the characteristic nature of IoT nodes, authors believe that the use of firewalls and protocols does not ensure the security of data traffic at the endpoints. In IoT, integrity can be enforced by maintaining end-to-end security to ensure accuracy and prevent tampering [11].

2.2 Confidentiality

Confidentiality is a critical security feature in the IoT. Confidentiality refers to ensuring that sensitive information is kept private and only accessible and controlled by authorized and authenticated individuals throughout the process. Sensitive information can include company information, security accreditations, patient data, or military information. Data confidentiality can be achieved through mechanisms such as data encryption and access control [5]. In [10], it was stated that there are different cryptographic algorithms, including symmetric key algorithms such as the advanced encryption standard (AES), which ensures the confidentiality of data, in addition to using Rivest Shamir Adelman (RSA) as an asymmetric algorithm for digital signatures and key exchanges. Therefore, data collected by a computer or sensor should never be sent to other devices unless properly encrypted to prevent malicious actors from accessing it, followed by a verification process [12]. However, these algorithms consume more battery and CPU power [10,13]. Different mechanisms for achieving confidentiality were suggested by [9], such as two-step and biometric verification and user awareness of data management mechanisms.

2.3 Availability

It is essential to have immediate access to authorized parties' information resources during normal conditions as well as in the event of a disaster [5]. Therefore, the system should automatically recover in the event of a crash. Nevertheless, data is not the only component of the IoT; devices and services must also be available when needed in a timely manner to meet IoT expectations. Data availability can be compromised by attacks, such as DoS attacks. Various mechanisms are used to maintain availability, including firewalls, intrusion detection systems, and redundancy techniques [9].

2.4 Authentication and Authorization

Cybersecurity relies heavily on authentication and authorization in the IoT. Authentication involves identifying the device, and authorization involves granting permission. While the IoT connects different smart devices, the ability to recognize these devices is crucial because malicious devices may misuse IoT networks through spoofing [9]. Each device must be able to identify and authenticate other devices; however, this can be a challenge because of the involvement of many entities and the need to interact with unknown devices. Consequently, mutual authentication is required for every interaction in the IoT [11].

3. KEY ROLE OF IOT LAYER

For a system to be secure, security must be incorporated into its entire development process and architecture. The strong security of the IoT architecture is increasingly important for supporting and managing IoT systems. Therefore, automated and smart systems can be realized using a well-developed IoT device architecture. IoT devices are integrated into complicated systems to gather and analyze data and produce useful outputs. However, no single architecture exists for all IoT systems. In general, the complexity of IoT systems depends on the tasks that must be addressed. Therefore, it is vital to know the IoT architecture layers to create a system that meets all the requirements and the maximum security requirements. The primary basic architecture introduced in the early stages of research in the area of IoT is a three-layer architecture [14]. This architecture comprises three layers: Perception, Network, and Application. Each proposed layer is defined to perform specific functions. The perception layer includes different edge devices and sensors that interact with the environment. The network layer attempts to connect these devices over the Internet to the application layer. The received data is processed using specialized services in the application layer.

3.1 Perception Layer

Sometimes, it is known as the sensor layer or the physical layer. It implies all types of sensors and a wide range of endpoint devices that can send and receive information about the environment, such as temperature, sound, light intensity, etc. This data can be preprocessed before sending to the network layer [9]. Perception devices can range from simple sensors to complex systems such as industrial control systems and medical devices. The perception layer is the lowest layer of the IoT architecture; however, it is considered one of the most sensitive layers.

3.2 Network Layer

A network layer connects all things, network devices, and servers. It manages all data transmissions between nodes in the network using different protocols. The network layer plays an essential role in intelligent event management and processing in the IoT by allowing the sharing of sensor data with connected objects. By acting as a bridge between the perception and application layers, it facilitates communication between them. There are several network technologies commonly used today with IoT, such as Wi-Fi, Bluetooth, 3G/LTE, Zigbee, Lora, and others [9].

3.3 Application Layer

In an IoT architecture, the application layer represents the final layer and provides community service. This layer generally ensures the integrity, confidentiality, and authenticity of the data [12]. The application layer provides users with an application-specific service. It is responsible for providing the customer with software resources. It is what the user interacts with, so application layers connect applications and end clients, allowing them to communicate. It defines different applications in which IoT can be deployed. These applications can be, for example, a smart home implementation or smart health. In this layer, end users can interact with all the connected devices.

4. SECURITY ATTACKS OF EACH IOT LAYER

One of the most important challenges for convincing users to adopt IoT technology is the protection of data and privacy. There are no specific solutions for IoT security that can be implemented at each layer [5]. Thus, the IoT system must be viewed as an entire system, and security is viewed as a chain, with the weakest link making the system insecure. When a system is designed and architected, security solutions across different layers must have some cooperation, which will help to overcome heterogeneous integration issues. Therefore, it is important to understand the possible threats and attacks on the system in order to add appropriate defenses.

4.1 Perception Layer

The perception layer is regarded as one of the most sensitive layers. It is the main target of attackers since several hardware components operate to collect information from an object, such as RFID, GPS, 2-D bar codes, sensors, and wireless sensor networks, etc. [15]. Depending on the requirements of the system, these components are selected to identify physical objects, collect and exchange information, and receive directions from the users. A variety of attacks can be applied to these parts, such as jamming, tampering, capturing nodes, etc. Attackers attempt to damage IoT devices. This type of attack is called a physical attack. Therefore, preventing unauthorized access and taking privacy measures are important.

In a study by [16], the authors emphasized the significance of preventing attackers from accessing objects of IoT perception to prevent physical damage or unintended changes in their operations. This ensures the integrity of the data as well as the confidentiality of the data. However, perception devices may still be vulnerable to various threats because technological heterogeneity makes it difficult to use only one type of security technology because perceptual environments are often open. The following are some common security attacks that affect the perception layer:

4.1.1 Jamming attacks

This denial-of-service attack is common among wireless IoT devices. As part of the attack, an attacker uses a jammer device that uses radio frequency (RF) to disrupt the signal between nodes, particularly when using wireless sensor networks. RF signals are transmitted at the same frequency as the targeted device, so the attacker can effectively block or overwhelm the communication between the two devices [17]. Therefore, the jamming attack interferes with the operation of the network in a way that makes users unable to use it [18]. To launch this type of attack, the attacker can bypass the protocols of the physical layers or emit a radio signal to scramble a particular channel until it runs out of energy. According to the ontology created by [19], jamming aims to disrupt a node's signal and will have varying impacts depending on the type, location, noise power, and type of jammer.

- **Countermeasures**

Different solutions have been proposed to protect against jamming attacks. Spread-spectrum communication techniques such as FHSS can be used to make it harder for attackers to locate and jam targeted devices [19]. This technique is considered complex and expensive since it uses complicated processing to switch between sensors due to the limited number of sensors that can switch efficiently between different frequencies. While channel surfing, priority messages, and spatial retreat are possible mitigations suggested by [8]. A cross-layer security mechanism called ‘Swarm Intelligence’ discussed by [20]. This mechanism predicts traffic patterns and detects malicious nodes to route information to alternative routes while maintaining network performance during jamming attacks. However, this mechanism may cause redundancy in the routing path and then denial of service resilience.

Two complementary methods are proposed by the author [21] to deal with jamming attacks. The first method is to use channel surfing or spatial retreat to avoid interference. A second method involves competing with jammers to achieve communication when jammers are present through frequency-hopping modulation or mapping of blocked regions of the sensor network. The attacker faces a greater challenge since he must know the frequency to jam it. However, collision risk may increase.

A mechanism proposed by [22] reduces the impact of jamming by detecting the signal and adjusting the authentication control threshold. However, physical layer authentication can be improved by ML-based learning methods since traditional authentication methods used for physical security are not sufficient due to the exact control threshold value used to detect unwanted signals [23]. A channel coordination protocol, SimpleMAC, has been developed by [24] that mitigates the effects of jamming with channel coordination. The SimpleMAC protocol utilizes a combination of the Simple Transmitter Strategy and Simple Signaling Scheme, which include a random backoff, frequency hopping, and carrier-sensing mechanisms. Combining these mechanisms increases the probability of successful transmission and reduces jamming attacks and collisions.

4.1.2 Tampering attacks

The tampering attack is considered one of the most famous physical attacks, focusing on the hardware components of the IoT system that usually operate in external or internal environments [5]. The attacker gains direct access to the physical components of a system through hub alteration or malicious code injection [8]. Alteration of the hub involves denial of access, altering sensitive information, or physical replacement of hardware, while injection of malicious code allows access to a node of the IoT system [5]. The hardware components can be damaged by altering the model; this will prevent them from communicating electronically with other sensors. Therefore, an attacker can replace or inject components and nodes as a form of tampering. By obtaining complete control of these components, the attacker aims to extract sensitive data and make the components unresponsive [23]. Usually, this sensitive information can be cryptographic keys, a routing table, or any sensitive data [10]. Tampering attacks may be classified into two categories: invasive attacks that require access to hardware components like chips, which require expensive equipment, and non-invasive attacks that take little time or effort and are easier to perform [19].

- **Countermeasures**

Tamper-resistant packaging is recommended by [13]. It is a complementary combination of physical security and logical security to prevent all attempts at tampering. The purpose of physical security is to ensure that the physical computing system is protected by placing a barrier to prevent unauthorized physical access to the system, while the purpose of logical security is to identify, authenticate, or control the access of users [25].

The author of [26] suggests that blockchain technology provides the key to data security and prevents tampering attacks by using a distributed and decentralized ledger to secure data and transactions between IoT devices, making it difficult for attackers to manipulate data. IoT access control mechanism with tamper-evident and inner product encryption based on blockchain proposed by [27] to prevent unauthorized access to IoT devices and the data collected. In this mechanism, access control can be fine-grained, policies can be completely hidden, and data is securely stored.

Multiple solutions suggested by [19] to prevent all tampering attacks, such as disabling the JTAG interface and using secure passwords for bootstrap loaders, can prevent unauthorized access to device hardware and firmware. A tamper-proofing and hiding method against a tampering attack was suggested by [8]. To mitigate tampering, [28] recommends enabling the usual firmware updates for devices.

4.1.3 Fake nodes

Also known as "malicious nodes,". It is a harmful attack on IoT's perception layer since its ability to disrupt networks, lose data, and violate privacy [29]. These fake nodes can be used to carry out various types of attacks, such as eavesdropping, data manipulation, or disrupting communication within the network and other nodes [15]. During the attack, attackers add a node; then the attacker can inject malicious data into the IoT system through the fake node in the network to prevent it from transmitting real information and causing the device to consume more energy [10]. Fake nodes could have an effect on the network layer by altering the route path and ultimately causing the system to be infected [18].

- **Countermeasures**

Securing the routing table is important to prevent malicious nodes from manipulating the routing table, and using whitelist and blacklist methods can be effective in eliminating fake nodes. All valid nodes are included in a whitelist, and all malicious nodes are in a blacklist [30]. To minimize the impact of fake nodes, it is important to remove them when detected. Embedding isolation and blacklisting of malicious nodes in the RPL protocol is essential to prevent malicious nodes from participating in the network and enhance IoT security [31]. Usually, maintaining a whitelist is easier, but large networks are better managed with blacklists. [29] proposes a method to detect malicious nodes in IoT networks using an online learning algorithm. The method involves calculating the credibility of each path on the network, modeling the reputation of the path, and detecting malicious nodes using a clustering algorithm. The authors show that the proposed method can detect malicious nodes with good stability.

The authors [10] emphasize the importance of secure device authentication and access control for the security of the IoT. Authors suggest various methods, including passwords, PKIs, and biometrics, to achieve this. The nodes should authenticate each other before communicating to prevent false node attacks and unauthorized access to sensitive information. While authentication is crucial in small networks, more efficient mechanisms were recommended for handling large numbers of IoT devices. As explained by [5], distributed environments are difficult to authenticate in, making it easy for malicious nodes to use fake identities for malicious or collusive purposes.

An artificial neural network (ANN) is suggested by [32] to detect malicious nodes in IoT networks by using the network to analyze node behavior, communication patterns, and network status to identify potential malicious activity, aiming to protect the IoT network from potential cyberattacks.

4.1.4 Timing attack

The timing attack is another confidential and threatening attack [11]. Based on a study by [15], this type of attack depends on the machine's processing power. Analyzing IoT device response times and identifying patterns of device behavior is critical for attackers. Also, attackers use information such as time consumption and the power consumed by sensor nodes to attack encryption mechanisms [10].

The majority of IoT devices can perform on-device processing, including data format conversion and data validation. Therefore, it is essential to enable IoT devices to perform secure and powerful processing with low power consumption [18]. Another type of timing attack that exploits information leaked through timing measurements of a system's activities is called a timing-based side-channel attack [33]. It enables an attacker to explore a device's vulnerabilities and extract secrets to use in his attack when it has weak computing capabilities and takes a long time to respond.

- **Countermeasures**

The fact that some devices provide on-device processing makes it imperative to evaluate source code against timing attacks. A study by [34] suggested the implementation of countermeasures in the form of software code since non-constant time functions, conditional operations, and cache access can cause timing leaks when implemented on a processor, which enables attackers to understand system timestamps. Moreover, cryptographic algorithms, when implemented on real systems, are vulnerable to timing side-channel attacks based on their execution behavior and on-device processing. This can cause timing leaks that can be exploited by attackers to understand system timestamps [35].

FISHER is suggested as a defense mechanism against timing-based side-channel attacks on IoT devices [36]. The objective is to minimize timing-based side-channel leaks by masking the device's reactive behavior and the system's timestamps. FISHER works by analyzing time stamps to identify leakages and implementing specific rules to disguise the behavior of IoT devices by inserting delays and generating fake packets with the aim of hiding the original traffic patterns from attacks.

4.1.5 Collision

Collisions can occur in various scenarios within IoT deployments. The attacker sends his own signal while the legitimate node transmits data to interfere with it, which can cause packet collisions between the two nodes transmitting at the same frequency. Moreover, conflicts can result if attackers tamper with important information, which leads to devices accidentally using the same address or identifier.

- **Countermeasures**

Based on [19], all defenses used against jamming attacks are also applicable to collision attacks. As noted earlier, [22] proposes a mechanism to reduce jamming's impact. According to their proposed mechanisms, management queue size and network size can accurately predict the frame collision probability caused by jamming attacks. Authors believe that when the network size is below the maximum management queue size, the frame collision probability stays low. However, as soon as the size of the network exceeds it, the collision probability increases significantly. According to [8], error correction codes are an efficient method to deal with collisions.

Time Division Multiple Access and Frequency Division Multiple Access approaches are proposed to prevent collisions. Time Division Multiple Access allows all groups to transmit sequentially, while a second solution uses Frequency Division Multiple Access, which transmits all groups in parallel and at a different frequency for each group [37]. To achieve both inter-channel parallelism and intra-channel parallelism while minimizing data gathering time, a combination of these two approaches is recommended.

4.1.6 Battery drain attacks

The attackers aim to exhaust the batteries of the nodes. There are numerous attacks that can increase the energy consumption of smart devices, which will exhaust the nodes. It occurs when attackers assign priority to a specific node that exhausts its battery. Sending data to a specific device with a higher priority will make it unfair and exhaust its battery [18,23]. Moreover, devices' batteries will become weak because of the constant pressure from this type of attack. Therefore, to reduce power, devices follow a sleep routine to conserve power. By keeping the device awake, attackers reduce battery life and force the node to shut down; this is called a sleep deprivation attack [38].

- **Countermeasures**

Battery-free technology using ambient radio signals could empower IoT devices instead of batteries, which can be used for the purpose of resolving the battery problem [39]. However, these signals are considered weak and have limitations. There are various forms of cyberattack that target depleting the energy of nodes, leading to a quick battery drain in battery-powered devices. It is possible to conserve the energy of the nodes and extend battery life by limiting the rate of incoming and outgoing requests [8].

To prevent wireless battery-draining attacks by combining power-switching methods with Wi-Fi power-saving mechanisms in smart devices, two security methods were proposed by [40]. To achieve power savings, one method suggested extending the waking state and regularly switching between wake mode and sleep mode, while the second method proposed extending the wake mode only when the frame received matches a shared secret key. The Wi-Fi power switch is used in the event of a battery drain attack to switch off the Wi-Fi functions while keeping the rest of the system operational. When Wi-Fi is off, the main system stores data in memory. When data needs to be transferred to the access point, it turns on and transfers it.

A study by [19] suggested that limiting the MAC admission control rate will prevent the sensor network from responding to excess requests, thus preventing energy loss. In addition, Authors believe that giving each sensor node a short period of time to access the channel and transmit data will reduce the long-term usage of the MAC channel.

4.1.7 RFID attacks

RFID plays an important role in IoT technology and has been considered one of the significant devices used to collect information. Disabling RFID tag attacks can disable and block tags permanently or temporarily, preventing radio signals from traveling between nodes. Permanently deactivating RFID tags will result in the destruction of these tags by tag removal, tag destruction, or a KILL command, while temporarily deactivating RFID tags may disrupt accurate and effective communication between nodes [41].A relay attack is another attack that may affect RFID. It uses a man-in-the-middle adversary to compromise the system. There is an adversarial device that is placed between a valid RFID tag and the reader to steal the data. Using this device, the legitimate tag and reader can intercept and modify radio signals [41].Moreover, RFID systems are considered vulnerable to cloning attacks. An attacker can clone an RFID electronic tag by copying its information. The clone tags will have the same characteristics as the original ones, which means that readers cannot differentiate between the two [42].

- **Countermeasures**

In general, encryption secures data; however, it does not provide enough security when reading information from tags, so there should be a mechanism to verify a reader's authenticity before giving them access to data [15]. [41] demonstrated that the RFID communication can be encrypted and provide a second form of authentication, such as a password, PIN, or biometric data, to protect against relay attacks. The authors also emphasized the importance of the distance between the RFID tag and the reader; the shorter the distance, the more difficult it would be for the adversary to launch a relay attack without detection. Furthermore, by identifying the geographic location of each node, it would be possible to detect cloned identities since no identity should be in two places at once [30].

In [43], authors suggest the use of Advanced Encryption Standard (AES) and the Rivest-Shamir-Adleman (RSA) algorithm to encrypt the data on the RFID tag, making it difficult for attackers to clone the tag. Also, implementing the Message Authentication Code (MAC) using a shared secret key between the tag and the reader or the server to prevent tag disabling was suggested to secure RFID. Additionally, the authors in [43] recommend using tamper-proof enclosures for RFID tags to physically protect them from tampering or destruction.Tag cloning threat can be alleviated using tag authentication. Blocking threat affects air interface and can be minimized if blocking devices are detected early so that suitable action can be performed. Several advanced solutions have been proposed to prevent cloning attacks however, these solutions require additional hardware resources, or they cannot detect clone tags in time [42]. Therefore, a method called adaptable clone detection (ACD) proposed by [42], which implement Floyd-Warshall shortest path algorithm and COTS RFID equipment in order display the position of abnormal tags in real time.

4.2 Network Layer

The network layer connects the perception layer with the application layer over the Internet. Data is collected and transmitted from sensor devices through different communication protocols such as IPv4/IPv6, 6LoWPAN, and RPL. However, because of the heterogeneity of the components of the network, current protocols cannot be used as is [11]. Therefore, IoT security also depends on the secure communication protocols that are used to ensure that data in transit is confidential, reliable, and available to prevent cyberattacks. Communication within the IoT is limited to machine-to-machine and has a security issue of compatibility, which makes it different from the internet. Consequently, a transmission system should be capable of managing a large number of devices without causing data loss. Many attackers target this layer by attempting to get unauthorized access to IoT systems and manipulate them without the user's permission. The following are some common security attacks that affect the network layer:

4.2.1 Blackhole attack

A blackhole attack is a type of cyberattack that can occur on computer networks. The blackhole attack uses the routing protocol of the hacked node to advertise itself as having the shortest route to the target. The attacker creates a situation where the hacked node drops all packets that should be forwarded, which can result in a complete halt of all data traffic [9]. When a network is compromised by a blackhole attack, energy is lost, congestion occurs, and there is an increase in network overhead, which affects the network performance [44,45].

- **Countermeasures**

Network Simulator 2.35 and TCL (Tool Command Language) can be used to detect and prevent malicious nodes in a simple network by injecting a malicious node, monitoring node behavior using IDS (Intrusion Detection Systems), and alerting the base station for removal and preventing Black Hole attacks. However, this method increases power consumption [46].

In [47], a fuzzy logic for trust management was suggested, which makes each node responsible for maintaining the trust value of its neighbor nodes to detect Black Hole attacks. Nodes maintain trust values using direct and indirect trust mechanisms, and digital signatures with RSA for packet integrity was used. In [48], a system that uses a first-route reply caching mechanism to prevent black hole attacks in the network was recommended. The first route reply packet that reaches the source node is ignored to mitigate the attack, and the protocol shows improvements in packet delivery ratio, delay, and throughput compared to existing protocols.

A novel system uses a deep learning model proposed by [44]. The system includes assigning nodes, data collection, detecting attacks, and preventing them with optimal path communication. Attacks are detected using Bait and round-trip time validation, and the data attributes are used to train an LSTM model. Optimal path selection is carried out using the fitness rate-based whale optimization algorithm based on energy, distance, delay, and packet delivery ratio.

A system that uses the artificial bee colony algorithm along with the reverse tracing technique was suggested by [49]. Nodes send data through their neighbors, and an RREQ packet is sent to the neighbor's node. If the node replies, data transmission begins; otherwise, the node is marked as a black hole, and the sender checks the next node for transmission. Therefore, the DoS attack can be prevented.

4.2.2 Sinkhole attacks

Attackers make the hacked node appear attractive to nearby nodes. A sinkhole attack is described as a destructive attack that compromises data integrity and reliability by routing packets to the wrong path or dropping packets [38]. A hacked node tries to direct traffic and packets from other nodes towards itself by promoting itself as the shortest path. Then the data can be changed. Hacked nodes may modify the right routing path during data collection and transmission, leading to cause a routing attacks [50]. Sinkhole attacks can lead to selective forwarding attacks, and, in combination with other attacks, sinkhole attacks can cause much more serious attacks. The sinkhole attack can cause congestion and speed up the energy consumption of the node [51].

- **Countermeasures**

SVELTE was proposed by [52], which is a real-time intrusion detection system for IoT networks that uses a hybrid approach of signature detection and anomaly detection to detect routing attacks. It is specifically designed for the new routing protocol (6LoWPAN) implementations and comes with an integrated mini firewall. The SVELTE system is designed to be small enough to be deployed on constrained nodes with limited energy and memory capacity, which primarily rely on signature detection to detect certain types of attacks.

The PRDSA (Probe Route-Based Defense Sinkhole Attack) approach was proposed by [53] to resist sinkhole attacks. The PRDSA approach implements the routing mechanism in addition to far-sink reverse routing, equal-hop routing, and minimum-hop routing with little impact on the network lifetime. The PRDSA approach can detect and bypass sinkholes, along with identifying the attacker's node location at the same time. [50] proposed a specification-based intrusion detection approach combined with the rules of the expert system knowledge base to detect sinkhole attacks. These rules are defined by users or experts based on thresholds and the expected behavior of network components. When the behavior inside a network diverges from a set of user-defined thresholds and rules, attacks are detected. However, this approach needs an improvement to the rule so that it can be implemented in different environments. Another way to avoid sinkholes is to use routing protocols that verify the bidirectional reliability of a route using end-to-end acknowledgments containing latency and quality data [54]. A study by [8,30] stated that geo-routing protocols could reduce the sinkhole attack.

4.2.3 Selective forwarding

The selective forwarding attack is a type of blackhole attack called a greyhole attack. Attackers compromise single or multiple nodes in order to interrupt network data flow and change the IP address of the traffic by dropping some messages and not forwarding them. So, an attacker selects a portion of the information and forwards it to the destination; the remaining are dropped [30]. Selective attacks can cause massive damage to networks in general, and especially to IoT networks with low-power IPL (Routing Protocol for Low Power and Lossy) networks [55]. This type of attack can remain undetected for a long time, which can damage a network.

- **Countermeasures**

Many secure routing solutions are too computationally heavy for direct application on resource-constrained IoT networks. Therefore, a lightweight, trust-based defense scheme is needed to prevent selective forwarding attacks [55]. It consists of three modules: detection, notification, and isolation. Based on the received data packets, the detection module analyzes the trust value of each node, and the notification module informs all nodes of the presence of malicious nodes. Isolation modules allow children of malicious nodes to isolate them and choose new parent nodes. The result is a changed propagation path for data packets.

AIPDR (AI-based packet drop ratio) is an artificial intelligence-based detection technique proposed by [56] to mitigate the selective forwarding attack that occurs in RPL protocols. Based on the packet delivery ratio value of each node and the border router node of the nodes, AIPDR will detect and eliminate malicious modes from the RPL network. However, the proposed approach is considered more efficient with a small number of network nodes.

Creating disjoint paths between the source and destination nodes is a solution proposed by [30] to prevent selective forwarding attacks. However, Authors believe that creating completely disjoint paths network-wide is difficult. Therefore, it is possible to dynamically select the paths. In addition, selective forwarding attacks can be prevented by making sure that the attacker cannot distinguish between different types of traffic, so that the attacker cannot forward any traffic or a certain amount of traffic [30]. However, the dynamic selection of the next-hop nodes and the localized information further reduce the adversary's control over the data flow [54].

A mechanism consisting of neighbor monitoring, attack detection, control packet collection, analysis, and new path identification to prevent selective forwarding was proposed to prevent selective forwarding [57]. The mechanism involves monitoring the behavior of neighboring nodes in the network to detect and identify the node(s) responsible for a selective forwarding attack. Then, control packets are used to manage and analyze network traffic. Finally, a new route is generated.

4.2.4 Denial of service attack (DoS)

The purpose of DoS is to overload the targeted machine with redundant requests to slow it down or prevent authentic users from using it, in addition to making the network unavailable to use, which may cause a collision, unfairness, exhaustion, and battery drain [38]. DoS attacks can deny the availability of data and can compromise the confidentiality and privacy of the network [5]. An ICMP flood is a DoS attack that uses spoofed source addresses to flood the target with ICMP echo requests. As a result, there will be a high rate of ICMP traffic.

A distributed denial-of-service (DDoS) attack occurs when multiple systems overload a target system. DDoS attacks use multiple IP addresses or machines, often infected with malware, to cause devastation. Hello-flood and SYN-flood are types of DoS attacks. The Hello flood attack involves overloading the channel with useless messages, creating high traffic and congestion on the channel [51]. While SYN flood attacks are designed to consume all resources by continuously requesting the connection and never completing the connection until all resources are exploited. DDoS attacks violate the availability, which is one of the essential components of IoT security issues, by preventing the accessibility of IoT components [58].

- **Countermeasures**

Traffic control, link authentication, active firewalls, and passive monitoring are all mechanisms that will mitigate denial-of-service attacks [8]. Furthermore, [28] emphasizes the limitations of accessing unused services and open ports, as well as how encrypting communication can prevent DoS attacks. [59] introduced a graph-

based outlier detection approach on the Internet of Things (GODIT), which detects DoS attacks in real time by analyzing each node in the IoT network as a graph stream and performing efficient data graph processing. As mentioned before, [20] suggests using a cross-layer security approach not only to detect physical-layer jamming attacks but also to detect DoS attacks. While [49] proposed a system that uses the artificial bee colony algorithm along with reverse tracing techniques to detect DoS attacks.
Furthermore, detection can occur at the border router node at an early stage in the application layer, ensuring the safety of the network device [60]. This method involves two algorithms. An algorithm determines whether the source of the threat is a confirmed threat, which is called a primary-check or suspicious threat, during the primary stage, and the validity of the suspicious input is validated during the second stage using datagram transport layer security (DTLS) as a security protocol for securing communication. However, their approach works wherever an access gateway or firewall acts as a proxy for IoT devices.

A Random Forest (RF) was proposed by [23], which is a special machine learning method based on a couple of Decision Trees (DT) to be used to detect DDoS attacks. An artificial neural network (ANN) algorithm can be implemented to detect DDoS attacks [61]. With this model, TCP, UDP, and ICMP DDoS attacks are detected using an Artificial Neural Network algorithm trained on characteristic patterns that separates genuine traffic from DDoS attacks and allows only real information packets to flow through the network. However, this model is not capable of defending against DDoS attacks using encrypted packet headers. One method of preventing hello-flood attacks involves verifying the bidirectionality of local links in addition to authentication, which verifies the identity of neighborhood nodes[54].

4.2.5 Sybil attack

The Sybil attack is one of the most dangerous routing attacks. The attacker seeks to establish fake connections in IoT networks by duplicating the identities of fake IoT nodes or fake sensors to impede network performance and undermine fault tolerance schemes. Honest IoT nodes are unable to distinguish valid connections from invalid ones. It may be possible for attackers to generate false reports and spam users with messages that may compromise their privacy [62]. The main purpose of a Sybil attack is to fill the memory of a neighboring node with useless information from non-existent neighbors [19]. Most Sybil attacks occur in a peer-to-peer network that affects performance, resource utilization, and data integrity [7]. This attack may reduce the effectiveness of fault tolerance schemes [51].

- **Countermeasures**

An authentication method, such as the SPIN algorithm, can be used to prevent Sybil attacks because identity fraud is at the core of this attack [19]. It is possible to detect suspicious Sybil users in the early stages using cryptographic schemes such as event signatures and authentication of identities.

Another solution was to use a Needham-Schroder protocol to verify the keys between nodes and a base station [54]. Neighboring nodes with keys establish an encrypted link. Therefore, limits on neighboring connections prevent insider attacks. Compromised nodes can only communicate with verified neighbors, restricting unauthorized access. Adversaries cannot eavesdrop on or modify communications despite creating artificial links, which will prevent Sybil attacks.

There are different types of Sybil defense schemes suggested, like social graph-based Sybil detection (SGSD), which enables a legitimate node to detect Sybil nodes using social graphs by traversing the graph in random walks or using community detection algorithms. While behavior classification-based Sybil defense (BCSD) enables Sybil users to be determined by analyzing their activities on the network and subsequently identifying users with a specific pattern of behavior on the network [62].

4.2.6 Wormhole attack

In this attack, a hacker relocates a piece of data on the network from where it was originally located. In this case, the data packets are relocated via a link of low latency [51]. The attacker, who is located at a distance from the target, uses out-of-bound channels to understate the distance between the two malicious nodes [54]. Wormholes can be implemented to exploit routing race conditions, which cause a malicious node to influence the topology by causing a node to receive routing information before it would normally reach them through multi-hop routing.

- **Countermeasures**

A wormhole is difficult to detect because it uses an out-of-band channel that remains undetectable to the underlying sensor network. However, many defenses are given to prevent wormhole attacks. A packet leash that uses a specific protocol called TIK that implements leashes was introduced as a general defense mechanism against wormhole attacks [63]. Authorizations and monitoring redundancy may reduce wormhole attacks [8]. While in [30], it is recommended to use separate link-layer keys for each segment of the network. As a result, there will be no communication between nodes in different segments, which can counteract the wormhole attack. In their paper another solution was implementing geographic routing protocols, which are resistant to wormhole attacks because of to pologies on demand by interacting locally and relying on local information without requiring a base station to initiate them [54].

4.3 Application Layer

The application layer defines all applications that utilize IoT technology. Through the application layer, users can interact with all connected devices in everyday life. So, it is responsible for providing application-specific services to users by processing the received data collected from the sensors. However, the application layer suffers from different threats and vulnerabilities from the inside and outside due to the lack of specific security software, which leads attackers to steal data. Attacks on the application layer can be performed by exploiting vulnerabilities in the operating system or system software [13]. It is an attack against software resources that takes the devices to an exhaustion state. The following are some common security attacks that affect the application layer:

4.3.1 Cross-site scripting (XSS)

It is a form of injection. It gives the attacker the ability to send malicious client-side scripts from a trusted web application. Therefore, an attacker can manipulate the application's content, the system will be controlled by attackers, and the data will be used in an illegal manner [15].

- **Countermeasures**

Detecting XSS attacks is one of the most important aspects of preventing them. The two most common types of XSS attack detection techniques are static and dynamic [64]. In static detection, the program's source code is examined to detect potential XSS vulnerabilities. A dynamic detection method based on simulating browser behavior, and also develop a headless browser-based web crawler to find hidden XSS injection points in pages by interpreting JavaScript code and retrieving Ajax content while considering complex scripts on web pages. Another dynamic method called Concolic Test can be used to detect XSS attacks [65]. This method applies machine learning algorithms to improve efficiency in detecting XSS vulnerabilities by determining dependencies and vectors; attacks could be executed automatically, dynamically detecting XSS vulnerabilities in applications.

4.3.2 Malicious code attack

The attacker embeds malicious code within designed software that damages and causes undesired effects. The main goal of this attack is to breach the confidentiality of the system and get the system infected, which enables the attacker to exploit the layer that vulnerable to start the attack [51]. Moreover, IoT applications will be affected by viruses and worms with malicious self-propagation attacks that can obtain or modify private data [38].

- **Countermeasures**

Safe programming and anti-virus software are the most important methods that can be used against malicious code attacks [8,51]. However, the nature of IoT devices is small and mobile, and have many limitations. As a result, installing a dynamic security patch might be very difficult, as the operating system or protocol stack may not support updated code and libraries. Secure boot mechanism, where only trusted programs are allowed to run on the device [28].

4.3.3 Mass data

Mass data is generated when a system lacks the capability to process data according to requirements due to the large number and volume of devices being used. Consequently, networks are disrupted and data is lost, and this will have a big impact on the availability of services [15]. In the network, there are no large number of network nodes that process a lot of data. As a consequence, some data can be lost during communication, which affects network efficiency [12]. However, due to the limited ability of the target system to handle large packets, attackers sometimes try to send a large ping packet to destroy the target system [32].

- **Countermeasures**

Usually, large amounts of data are collected from different environments. This will lead to disruption of the network, which will affect the availability of data and services. A novel data compression algorithm [lossy data compression algorithm (LCA) and lossless data compression algorithm (NLCA)] was proposed to handle mass data problems [66]. The author of [32] believes in the requirement of a multi-layered approach that focuses on both device security and network security to prevent mass data attacks.

4.3.4 Sniffer

A hacker can install sniffer programs on the system in order to collect data from network traffic. The primary purpose of sniffer programs is to steal passwords, emails, and files to manipulate them to gain illegal access and violate the privacy of users. Several protocols are vulnerable to sniffer programs, which enable the attacker to control the applications [38,51]. A lack of appropriate protection and different applications that have different authentication mechanisms will result in difficulties for the privacy of the user since sensitive data can be accessed by many unauthorized users.

- **Countermeasures**

Data encryption mechanisms and resource access control to prevent privacy leakage are suggested as countermeasures to sniffer attacks [12,51]. However, to ensure that only authorized persons can access data, it is recommended to implement an encryption mechanism in addition to a two-step verification process. Furthermore, the system must be able to detect any attempts to tamper with data through the use of the checksum and cyclic redundancy check [12]. Another thought by [10] to reduce the possibility of facing sniffer attacks is that users must be taught how to use complex passwords and implement access control mechanisms.

4.3.5 Buffer overflow attacks

Software security is greatly compromised by buffer overflows. When developers are writing non-standard code in software, buffer overflow vulnerabilities may occur, which could be exploited by hackers to their benefit [12]. Additionally, when a buffer overflow occurs, the system will crash, incorrect results will be generated, and memory access errors will occur. In addition, buffer overflows give attackers the ability to control the execution flow of the vulnerable program or overwrite its memory. Consequently, the path can be diverted, private information can be exposed, and damaged files can be compromised [67].

- **Countermeasures**

A buffer overflow detection hardware design that is architecturally enhanced was proposed by [67]. It includes instruction monitoring and verification for tracing program execution behavior. Secure tag validation is another feature that monitors the attributes of every memory segment. In their proposed technique, authors claim that it can detect a wide variety of buffer overflow attacks and that it can be implemented with low performance penalties and minimal overhead. DisARM, a new anti-buffer overflow defense method proposed by [68], prevents both code-injection and reuse-based buffer overflow attacks by preventing attackers from manipulating a function's return address.

4.3.6 Phishing

It is one of the main threats that causes data violations. Through fraudulent attempts, an attacker attempts to steal a user's credentials. So, attackers can bypass the IoT devices’ traffic to gather and use sensitive information about their intended targets [38]. Various communication channels are used in phishing attacks, including email, websites, instant messages, and mobile applications. The increasing sophistication of phishing techniques is causing the phishing phenomenon to increase and intensify [28].

- **Countermeasures**

To detect phishing attacks, several tools have been developed. Netcraft, AntiPhishing, and LinkExtend have all been developed as tools to detect phishing attacks [28]. Tools like this are installed as extensions in web browsers. These tools, however, cannot be used appropriately on IoT devices to detect phishing attacks since many IoT devices are controlled by smartphone apps through Bluetooth rather than a web interface. Therefore, a STRIDE threat modeling approach was proposed to identify and mitigate the cyber threats that can cause phishing attacks. Moreover, a proper authentication and access control mechanism must be implemented to prevent the illegal user from entering the system and taking control of the devices that gather and collect sensitive information [38].

5. CONCLUSIONS

In conclusion, this study has provided a comprehensive overview of the security issues and countermeasures associated with IoT architecture. As the Internet of Things continues to transform many industries, it is crucial to address the security challenges that appear in IoT systems. By understanding these challenges and implementing effective countermeasures integrity, confidentiality, and availability of IoT systems can be achieved, which promoting a secure and trustworthy IoT ecosystem. Moreover, the study has highlighted the importance of viewing IoT architecture as a holistic entity, where security should be considered as a chain. The weakest link in the architecture can compromise the overall security of the system. By identifying and analyzing the security issues at various layers of the IoT architecture, this study emphasized on the vulnerabilities and potential threats that can breach the security of IoT systems and explored a range of countermeasures that can be employed to mitigate these security risks effectively. The findings of this research contribute to enhancing the knowledge and understanding of IoT architecture security among researchers, practitioners, and stackholder which lead to protect IoT devices, networks, and data from potential attacks and breaches. However, it is important to note that the IoT landscape is continuously growing, and new security challenges may emerge over time. Therefore, this study serves as a foundation for further exploration and innovation in the field of IoT security. Future studies can be build to explore deeper into specific security issues, refine existing countermeasures, and explore emerging technologies and strategies for ensuring the long-term security of IoT systems.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Ashton K. That ‘internet of things’ thing. RFID journal. 2009 Jun 22;22(7):97-114.
2. Vermesan O, Friess P, Guillemin P, Gusmeroli S, Sundmaeker H, Bassi A, Jubert IS, Mazura M, Harrison M, Eisenhauer M, Doody P. Internet of things strategic research roadmap. InInternet of things-global technological and societal trends from smart environments and spaces to green ICT. River Publishers. 2022 Sep 1;9-52.
3. Union IT. ITU Internet Reports 2005: The Internet of Things. In: Proceedings of the Proc. Workshop Rep. Int. Telecommun. Union; 2005.
4. Sethi P, Sarangi SR. Internet of things: Architectures, protocols, and applications. Journal of Electrical and Computer Engineering. 2017 Jan 26;2017.
5. Frustaci M, Pace P, Aloi G, Fortino G. Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of Things journal. 2017 Oct 27;5(4):2483-95.
6. Singh S, Singh N. Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce. In 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). Ieee.2015 Oct 8;1577-1581.
7. Akhtar MS, Feng T. A systemic security and privacy review: Attacks and prevention mechanisms over IOT layers. EAI Endorsed Transactions on Security and Safety. 2022 Aug 5;8(30).
8. D. Singh, Pushparaj, M. K. Mishra et al. Security issues in different layers of iot and their possible mitigation. International Journal of Scientific and Technology Research. 2020;9(4): 2762–2771.
9. Krishna RR, Priyadarshini A, Jha AV, Appasani B, Srinivasulu A, Bizon N. State-of-the-art review on IoT threats and attacks: Taxonomy, challenges and solutions. Sustainability. 2021 Aug 23;13(16):9463.
10. Ali I, Sabir S, Ullah Z. Internet of things security, device authentication and access control: A review. arXiv preprint arXiv:1901.07309; 2019 Jan 9.
11. Mahmoud R, Yousuf T, Aloul F, Zualkernan I. Internet of things (IoT) security: Current status, challenges and prospective measures. In 2015 10th International Conference for Internet

Technology and Secured Transactions (ICITST). IEEE. 2015 Dec 14;336-341.

12. Swamy SN, Jadhav D, Kulkarni N. Security threats in the application layer in IOT applications. In 2017 International Conference on i-SMAC (Iot In Social, Mobile, Analytics And Cloud)(i-SMAC). IEEE. 2017 Feb 10;477-480.
13. Sayana LS, Joshi BK. Security issues in internet of things. Uttarakhand: ICFAI; 2016 Apr.
14. Wu M, Lu TJ, Ling FY, Sun J, Du HY. Research on the architecture of Internet of Things. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). IEEE. 2010 Aug 20;5:V5-484.
15. Burhan M, Rehman RA, Khan B, Kim BS. IoT elements, layered architectures and security issues: A comprehensive survey. sensors. 2018 Aug 24;18(9):2796.
16. Khan R, Khan SU, Zaheer R, Khan S. Future internet: The internet of things architecture, possible applications and key challenges. In 2012 10th International Conference on Frontiers of Information Technology. IEEE. 2012 Dec 17;257-260.
17. Borgohain T, Kumar U, Sanyal S. Survey of security and privacy issues of internet of things. arXiv preprint arXiv:1501.02211; 2015 Jan 9.
18. Smith R, Palin D, Ioulianou PP, Vassilakis VG, Shahandashti SF. Battery draining attacks against edge computing nodes in IoT networks. Cyber-Physical Systems. 2020 Apr 2;6(2):96-116.
19. Znaidi W, Minier M, Babau JP. An ontology for attacks in wireless sensor networks (Doctoral dissertation, INRIA).
20. Muraleedharan R, Osadciw LA. Cross layer denial of service attacks in wireless sensor network using swarm intelligence. In 2006 40th Annual Conference on Information Sciences and Systems. IEEE. 2006 Mar 22;1653-1658.
21. Xu W, Ma K, Trappe W, Zhang Y. Jamming sensor networks: Attack and defense strategies. IEEE network. 2006 Jun 5;20(3):41-7.
22. Yin W, Hu P, Zhou H, Xing G, Wen J. Jamming attacks and defenses for fast association in IEEE 802.11 ah networks. Computer Networks. 2022 May 8;208:108890.
23. Tahsien SM, Karimipour H, Spachos P. Machine learning based solutions for security of Internet of Things (IoT): A survey. Journal of Network and Computer Applications. 2020 Jul 1;161: 102630.
24. Chang SY, Hu YC, Laurenti N. SimpleMAC: A jamming-resilient MAC-layer protocol for wireless channel coordination. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking 2012 Aug 22;77-88.
25. Weingart SH. Physical security devices for computer subsystems: A survey of attacks and defenses. In International Workshop on Cryptographic Hardware and Embedded Systems. Berlin, Heidelberg: Springer Berlin Heidelberg. 2000 Aug 17;302-317.
26. Miller D. Blockchain and the internet of things in the industrial sector. IT professional. 2018 Jun 11;20(3):15-8.
27. Han P, Zhang Z, Ji S, Wang X, Liu L, Ren Y. Access control mechanism for the Internet of things based on blockchain and inner product encryption. Journal of Information Security and Applications. 2023 May 1;74:103446.
28. Abbas SG, Vaccari I, Hussain F, Zahid S, Fayyaz UU, Shah GA, Bakhshi T, Cambiaso E. Identifying and mitigating phishing attack threats in IoT use cases using a threat modelling approach. Sensors. 2021 Jul 14;21(14):4816.
29. Li B, Ye R, Gu G, Liang R, Liu W, Cai K. A detection mechanism on malicious nodes in IoT. Computer Communications. 2020 Feb 1;151:51-9.
30. Wallgren L, Raza S, Voigt T. Routing attacks and countermeasures in the RPL-based internet of things. International Journal of Distributed Sensor Networks. 2013 Aug 22;9(8):794326.
31. Sahay R, Geethakumari G, Mitra B. IB-RPL: Embedding isolation and blacklisting of malicious nodes in RPL for securing IoT-LLNs. In 2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). IEEE. 2021 Dec 13;1-6.
32. Khatun MA, Chowdhury N, Uddin MN. Malicious nodes detection based on artificial neural network in IoT environments. In 2019 22nd International Conference on Computer and Information Technology (ICCIT). IEEE. 2019 Dec 18;1-6.
33. Vuppala S, Mady AE, Kuenzi A. Moving target defense mechanism for side-

channel attacks. IEEE Systems Journal. 2019 Jun 27;14(2):1810-9.

34. Takarabt S, Schaub A, Facon A, Guilley S, Sauvage L, Souissi Y, Mathieu Y. Cache-timing attacks still threaten iot devices. InCodes, Cryptology and Information Security: Third International Conference, C2SI 2019, Rabat, Morocco, April 22–24, 2019, Proceedings-In Honor of Said El Hajji. Springer International Publishing. 2019;3:13-30.
35. Lyu Y, Mishra P. A survey of side-channel attacks on caches and countermeasures. Journal of Hardware and Systems Security. 2018 Mar;2:33-50.
36. Prates N, Vergütz A, Macedo RT, Santos A, Nogueira M. A defense mechanism for timing-based side-channel attacks on IoT traffic. In GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE. 2020 Dec 7;1-6.
37. Haiahem R, Minet P, Boumerdassi S, Azouz Saidane L. Collision-free transmissions in an IoT monitoring application based on LoRaWAN. Sensors. 2020 Jul 21;20(14):4053.
38. Jamali J, Bahrami B, Heidari A, Allahverdizadeh P, Norouzi F. Towards the internet of things. Springer International Publishing; 2020.
39. Calhoun BH, Wentzloff DD. Ultra-low power wireless SoCs enabling a batteryless IoT. In Hot Chips Symposium. 2015 Aug 1;1-45.
40. Lee IG, Go K, Lee JH. Battery draining attack and defense against power saving wireless LAN devices. Sensors. 2020 Apr 5;20(7):2043.
41. Mitrokotsa A, Rieback MR, Tanenbaum AS. Classifying RFID attacks and defenses. Information Systems Frontiers. 2010 Nov;12:491-505.
42. Huang W, Zhang Y, Feng Y. ACD: An adaptable approach for RFID cloning attack detection. Sensors. 2020 Apr 22;20(8):2378.
43. Singh AK, Patro BD. Security attacks on RFID and their countermeasures. In Computer Communication, Networking and IoT: Proceedings of ICICC 2020. Springer Singapore. 2021; 509-518.
44. Pawar MV. Detection and prevention of black-hole and wormhole attacks in wireless sensor network using optimized LSTM. International Journal of Pervasive Computing and Communications. 2023 Jan 6;19(1):124-53.
45. Ali S, Khan MA, Ahmad J, Malik AW, ur Rehman A. Detection and prevention of Black Hole Attacks in IOT & WSN. In 2018 third international conference on fog and mobile edge computing (FMEC). IEEE. 2018 Apr 23;217-226.
46. Kaurav A, Kumar KA. Detection and prevention of blackhole attack in wireless sensor network using Ns-2.35 simulator. International Journal of Scientific Research in Computer Science, Engineering and Information Technology. 2017;2(3):717. ISSN : 2456-3307
47. Arulkumaran G, Gnanamurthy RK. Fuzzy trust approach for detecting black hole attack in mobile adhoc network. Mobile Networks and Applications. 2019 Apr 15;24:386-93.
48. Jain AK, Tokekar V. Mitigating the effects of Black hole attacks on AODV routing protocol in mobile ad hoc networks. In 2015 International Conference on Pervasive Computing (ICPC). IEEE. 2015 Jan 8;1-6.
49. Hemalatha P, Vijithaananthi J. An effective performance for Denial of Service Attack (DoS) detection. In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC) IEEE. 2017 Feb 10; 229-233.
50. An GH, Cho TH. Improving sinkhole attack detection rate through knowledge-based specification rule for a sinkhole attack intrusion detection technique of IoT. Int. J. Comput. Netw. Appl. 2022 Mar;9:169.
51. Leloglu E. A review of security concerns in Internet of Things. Journal of Computer and Communications. 2016 Dec 30;5(1):121-36.
52. Raza S, Wallgren L, Voigt T. SVELTE: Real-time intrusion detection in the internet of things. Ad Hoc Networks. 2013 Nov 1;11(8):2661-74.
53. Liu Y, Ma M, Liu X, Xiong NN, Liu A, Zhu Y. Design and analysis of probing route to defense sink-hole attacks for internet of things security. IEEE Transactions on Network Science and Engineering. 2018 Nov 13;7(1):356-72.
54. Karlof C, Wagner D. Secure routing in wireless sensor networks: Attacks and countermeasures. Ad Hoc Networks. 2003 Sep 1;1(2-3):293-315.
55. Jiang J, Liu Y. Secure IoT routing: Selective forwarding attacks and trust-

based defenses in RPL network. arXiv preprint arXiv:2201.06937; 2022 Jan 18.

56. Neerugatti V, Rama Mohan Reddy A. Artificial intelligence-based technique for detection of selective forwarding attack in rpl-based internet of things networks. In Emerging Research in Data Engineering Systems and Computer Communications: Proceedings of CCODE 2019. Springer Singapore. 2020;67-77.
57. Mathur A, Newe T, Rao M. Defence against black hole and selective forwarding attacks for medical WSNs in the IoT. Sensors. 2016 Jan 19;16(1):118.
58. Abughazaleh N, Bin R, Btish M. DoS attacks in IoT systems and proposed solutions. Int. J. Comput. Appl. 2020 Jun;176(33):16-9.
59. Paudel R, Muncy T, Eberle W. Detecting dos attack in smart home iot devices using a graph-based approach. In 2019 IEEE International Conference on Big Data (Big Data). IEEE. 2019 Dec 9;5249-5258.
60. Kajwadkar S, Jain VK. A novel algorithm for DoS and DDoS attack detection in Internet of things. In 2018 Conference on Information and Communication Technology (CICT). IEEE. 2018 Oct 26;1-4.
61. Saied A, Overill RE, Radzik T. Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing. 2016 Jan 8;172:385-93.
62. Zhang K, Liang X, Lu R, Shen X. Sybil attacks and their defenses in the internet of things. IEEE Internet of Things Journal. 2014 Jul 30;1(5):372-83.
63. Hu YC, Perrig A, Johnson DB. Wormhole attacks in wireless networks. IEEE Journal on Selected Areas in Communications. 2006 Feb 6;24(2): 370-80.
64. Liu Y, Zhao W, Wang D, Fu L. A XSS vulnerability detection approach based on simulating browser behavior. In 2015 2nd International Conference on Information Science and Security (ICISS) IEEE. 2015 Dec 14;1-4.
65. Guo X, Jin S, Zhang Y. XSS vulnerability detection using optimized attack vector repertory. In 2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. IEEE. 2015 Sep 17;29-36.
66. Hu C, Pu Y, Yang F, Zhao R, Alrawais A, Xiang T. Secure and efficient data collection and storage of IoT in smart ocean. IEEE Internet of Things Journal. 2020 Apr 20;7(10): 9980-94.
67. Xu B, Wang W, Hao Q, Zhang Z, Du P, Xia T, Li H, Wang X. A security design for the detecting of buffer overflow attacks in IoT device. IEEE Access. 2018 Nov 15;6:72862-9.
68. Habibi J, Panicker A, Gupta A, Bertino E. DisARM: Mitigating buffer overflow attacks on embedded devices. In Network and System Security: 9th International Conference, NSS 2015, New York, NY, USA, November 3-5, Proceedings. Springer International Publishing. 2015; 9:112-129.

Peer-review history:
The peer review history for this paper can be accessed here:
https://www.sdiarticle5.com/review-history/112490