Migration Letters

Volume: 21, No: S6 (2024), pp. 1326-1342

ISSN: 1741-8984 (Print) ISSN: 1741-8992 (Online)

www.migrationletters.com

Exploring The Security Landscape: A Comprehensive Analysis Of Vulnerabilities, Challenges, And Findings In Internet Of Things (Iot) Application Layer Protocols

Sushil Kumar Bhoi[1,*], Umashankar Ghugar[2], Satyabrata Dash[3], Rakesh Nayak[2] and Dilip Kumar Bagal[4]

Abstract

Network security is significantly impacted by the difficult task of securing Internet of Things (IoT) ecosystems. Strong security protocols must be developed in order to protect the communication networks that link Internet-connected IoT devices. Compared to earlier networks, modern networks have a greater variety of communication protocols, standards, and device capabilities, which makes providing network security more challenging. The three tiers that comprise the traditional Internet of Things (IoT) architecture are Perception, Network, and Application. Allowing a broad variety of security limitations at every level is crucial to ensuring these apps function properly and efficiently. This study primarily focuses on application layer protocols including MQTT, AMQP, XMPP, CoAP, and DDS that are often used in the Internet of Things (IoT). The report also provides an understanding of the application layer protocol security issues. IoT security remains a big issue, which is why several drafts, standards, and research initiatives have been suggested. The root of many security challenges lies in the operational conditions of protocols, notably the absence of universally accepted standards for authentication, authorization, secrecy, and integrity Therefore, a comprehensive examination is necessary to develop effective countermeasures. Most security flaws that arise are caused by the protocol's operating circumstances, which are mostly caused by the lack of widely recognized standards for things like integrity, secrecy, authorization, and authentication. The usage of suitable protocol settings is required to address these common security issues.

Keywords: *Application Layer Protocol, Challenge, Internet of Things (IoT), Security Vulnerabilities.*

1 Introduction

The way we see and engage with the digital and physical worlds is being completely changed by the Internet of Things (IoT). This global paradigm shift revolves around the interconnectedness of a myriad of devices, facilitating the exchange of information and enabling the delivery of advanced services. But this unprecedented degree of connectedness also presents a new set of difficulties, chief among them being security-related. Our goal in conducting this comprehensive study was to identify vulnerabilities in Internet of Things (IoT) application layer protocols and assess potential security risks in this dynamic environment. In essence, the Internet of Things, or IoT, is a vast network of virtual and actual objects that are connected to one another and have the ability to process, act, [1]and sense data. This intricate web of connectivity allows for the seamless transmission of data, enabling real-time insights and fostering automation across diverse domains. Yet, as the IoT ecosystem continues to expand, the need for robust security measures becomes increasingly critical. The application layer of the IoT architecture plays a pivotal role in facilitating communication between devices, serving as the gateway for data exchange and interaction. It is within this layer that a

[1]Department of Electrical Engineering, Government College of Engineering Kalahandi, Bhawanipatna, Kalahandi, Odisha, India, PIN-766003
[2]Department of CSE, School of Engineering, OP Jindal University, Raigarh, Chhattisgarh, India, PIN-496107
[3]Department of CSE, GITAM School of Technology, GITAM (Deemed to be University), Vishakhapatnam, Andhra Pradesh, India, PIN-530045
[4]Department of Mechanical Engineering, Government College of Engineering Kalahandi, Bhawanipatna, Kalahandi, Odisha, India, PIN-766003
[*]Corresponding Author's Sushil Kumar Bhoi[1]

multitude of protocols operate, governing how devices communicate, share information, and execute commands. But these protocols' sheer design creates a complicated web of exploitable vulnerabilities, putting the availability, integrity, and secrecy of data in the IoT ecosystem at serious danger.

The urgency to address these security concerns arises from the fundamental reliance of utility systems on effective communication between devices. For these systems to function accurately and efficiently, the establishment of protocols that enhance security and prevent data loss becomes imperative. Considering the number of devices with differing resources and capabilities inside the Internet of Things (IoT), the issue comes in integrating distinct protocols across various platforms. In order to highlight the inherent security issues with commonly used IoT application layer protocols, this paper attempts to give a comprehensive analysis of such protocols. Particularly under examination are protocols like Data Distribution Service (DDS), AMQP (Advanced Message Queuing Protocol), CoAP (Constrained Application Protocol), and MQTT (Message Queuing Telemetry Transport). These protocols, derived and adapted from existing Internet protocols, have become the backbone of communication in IoT applications, especially those involving devices with limited resources. The multifaceted exploration will not only focus on the technical intricacies of these protocols but also on the broader security landscape within the Internet of Things. The interconnectedness of devices introduces complexities that extend beyond traditional security paradigms, requiring a holistic understanding of potential threats and vulnerabilities.

This research intends to add to the body of information already in existence by shedding light on the security issues that IoT application layer protocols are facing as we set out on this path. By unveiling vulnerabilities and examining potential risks, we aspire to pave the way for the development of effective countermeasures, industry standards, and best practices to fortify the security posture of IoT ecosystems. The ultimate goal is to foster a secure and resilient foundation for the continued growth and integration of the Internet of Things into our interconnected world.

1.1 The IoT reference model

The International Telecommunication Union-Telecommunication defines the Internet of Things (IoT) reference model as having four primary levels: application, network, device, service support and application support. Embedded within each layer are integral management and security functionalities, forming a comprehensive framework that addresses the multifaceted aspects of IoT connectivity and operation. The capabilities depicted in Figure 1 encompass both generic and particular attributes that are applicable to various levels.

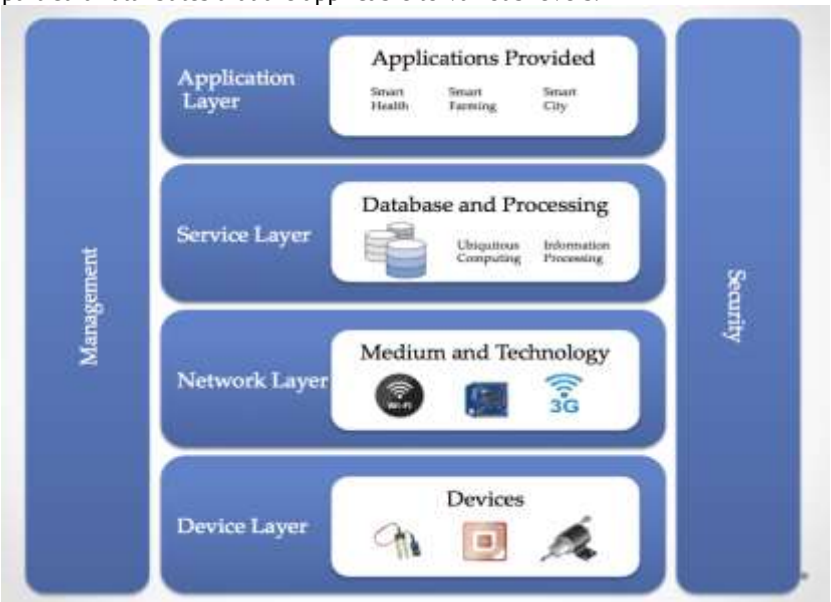
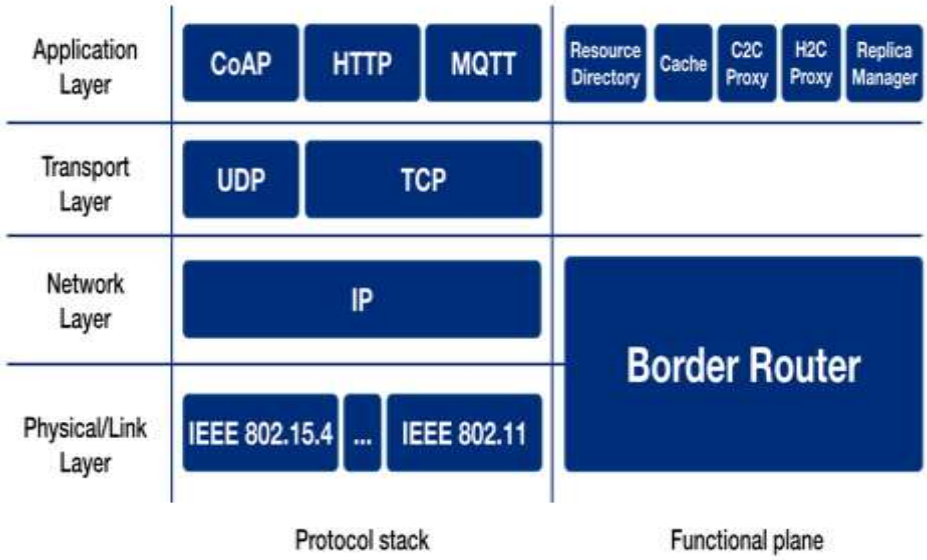


Figure 1 Reference model for IoT

Applications for the Internet of Things (IoT) may make use of a range of general support features that are included in the service and application support layer. These abilities might involve, among



other things, data processing and storage. Unique support capabilities are those that distinguish them from generic capabilities and are necessary to provide support for a wide range of applications [4]. Networking and transport functions are separated inside the network layer. While the transport capabilities concentrate on making it possible to convey data unique to IoT services and applications, the networking talents provide control activities linked to network connections. The capabilities of the devices are located in the device layer, which is the lowest level of the model. Here, they might have direct or indirect interactions with the communication network. Indirect contact necessitates the use of a gateway in contrast to direct engagement, which does not. Direct engagement needs the use of a gateway to facilitate information flow across the network. Two other elements that enable devices to form ad hoc connections and save energy are ad hoc networking and sleep/wake capabilities. Among other layers, authorization and authentication at the application, network, and device levels are among the security components that are not particular to a certain application. Additionally, each stratum has distinct characteristics. The aforementioned variables address various aspects of security at varying degrees. At the application layer, protections are specifically put in place against viruses, privacy invasion, security audits, and confidentiality and integrity of application data. At the network layer, protecting the integrity and secrecy of signaling data is crucial. The administration and security skills are strongly tied to the special demands of applications, including mobile payment [9].

1.2 IOT Protocol Stack

Choosing the right technology stack and tools for IoT projects is a difficult task for companies, architects, and developers. The lack of standardization in IoT protocols is the root cause of this problem, which leads to a complicated environment where selecting the best strategy is essential. The inherent limitations of the Internet of Things (IoT) further compound this challenge, marked by constrained memory capacity, low power availability, minimal bandwidth, and significant packet loss. These limitations make it difficult to use web technologies and the conventional TCP/IP stack in an IoT context. Within the realms of Internet of Things (IoT), Machine to Machine (M2M), and Home Automation, proprietary protocols have evolved to solve these challenges. Prominent instances are ZigBee and Z-Wave, which are backed by coalitions of suppliers of goods. However, these protocols lack the standardization observed in widely used internet protocols like TCP, IP, HTTP, or SMTP. This lack of standardization adds complexity to IoT implementations.

Wireless communication relies on the IEEE 802.15.4 standard, which contains the Media Access Control (MAC) and Physical layer (PHY) standards. While IEEE 802.11 is dedicated to Wi-Fi for wireless LANs, and IEEE 802.3 is standardized for Ethernet, the 802.15 series focuses on wireless personal area networks (WPANs) with diverse purposes. An example is the 802.15.1 standard, specifically tailored for Bluetooth technology. The primary objective of IEEE 802.15.4 is to facilitate communication among devices within limited environments characterized by restricted resources, including low memory capacity, limited power availability, and constrained bandwidth.

Figure 2: Protocol stack

Within the Internet realm, UDP (User Datagram Protocol) is favored for certain applications such as gaming and video streaming, whereas TCP (Transmission Control Protocol) is the standard option for the Transport Layer Protocol. However, UDP often turns out to be more appropriate inside the Internet of Things (IoT) for a variety of circumstances. In contrast to TCP, UDP is renowned for being lightweight and does not have the strong resilience characteristics of TCP, including guaranteed packet delivery. The trade-off is that UDP is faster than TCP because to its smaller header size, which makes it ideal for environments with few resources, such sensors and IoT devices. While akin to TLS/SSL in providing security features, DTLS caters to the unique demands of UDP, making it a vital component in securing communications within the IoT ecosystem. This strategic use of UDP and specialized security protocols like DTLS addresses the challenges posed by resource constraints in IoT environments while ensuring efficient and secure data transmission. The significance of these standardized protocols lies in their ability to provide a common framework, fostering interoperability and easing the implementation of IoT solutions. As the IoT landscape evolves, the adoption of these standards is anticipated to streamline development, enhance security, and contribute to the seamless integration of diverse devices and technologies.

2 Application-Layer Protocols

2.1 COAP

A customized protocol called restricted application protocol (CoAP) was created to make web transfers easier in the setting of constrained Internet of Things (IoT) nodes and networks. Figure 3 illustrates the different pathways for CoAP and HTTP communications, highlighting the interaction between various elements such as the CoAP Server and CoAP Clients.

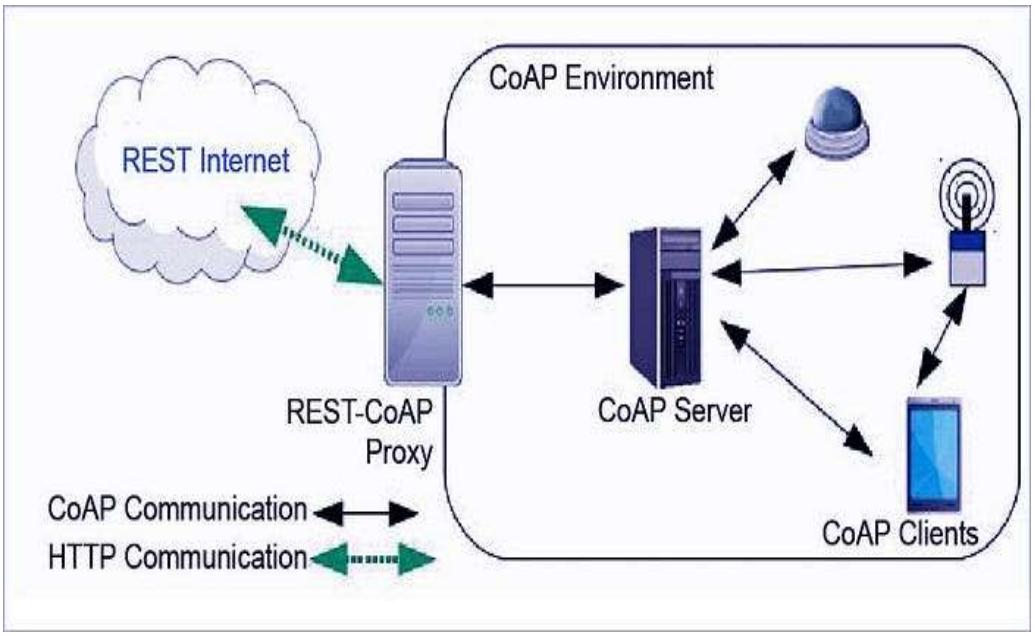


Figure 3: A schematic representation of the communication between REST Internet and CoAP Environment via a REST-CoAP Proxy

The Internet of Things (IoT) is made up of sensors, actuators, and devices that operate in constrained environments with low power availability, constrained bandwidth, limited memory, and a high probability of packet failure. The functioning of HTTP in resource-constrained situations, such the Internet of Things (IoT), was not taken into consideration in the protocol's initial architecture. Therefore, when using HTTP in IoT-constrained situations, its fairly bulky nature—which is typified by its big header size and text encoding—presents issues. This is the application of the Constrained Application Protocol (CoAP).

2.2 MQTT

The acronym for Message Queue Telemetry Transport is MQTT. It is characterized by its lightweight nature, making it suitable for resource-constrained environments. Figure 4 shows how

messages are published by various publishers, passed through a broker, and then subscribed to by the subscribers based on topics.

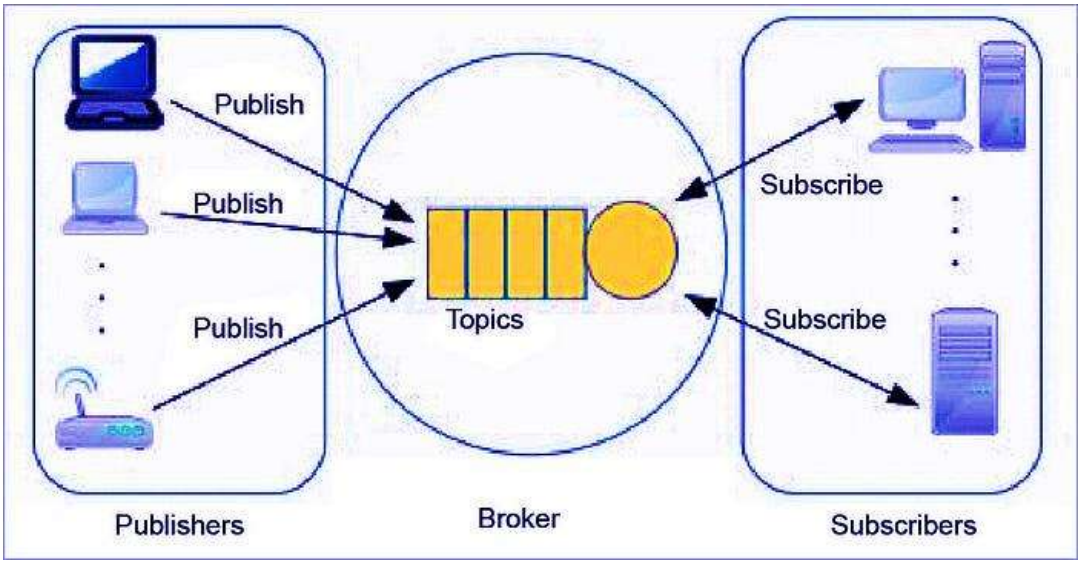


Figure 4: A diagram illustrating the Publish/Subscribe messaging pattern in IoT communication

In a nutshell, MQTT is the AMQP or JMS of the Internet of Things (IoT) for the context of restricted resources. The MQTT protocol was developed in 1999 by Andy Stanford Clark and Arlen Nipper as a communication system to lower battery usage and bandwidth utilization for connecting oil pipelines via satellite connections. Similar to the messaging systems used in the Web and Client Server domains, MQTT combines a pub-sub architecture with a broker-based technique inside the context of the constrained Internet of Things (IoT). Nothing in the user's work needs to be revised in an academic manner.

2.3 AMQP

The Advanced Message Queuing Protocol (AMQP) is a widely accepted and openly available standard that facilitates the exchange of business messages among various applications. The aforementioned technology establishes connections across various systems, facilitating the transfer of essential information to support corporate activities. Additionally, it ensures the accurate transmission of instructions necessary for accomplishing desired objectives. By providing encryption and compatibility, the Advanced Message Queuing Protocol (AMQP) enables safe and uniform communication across different entities, including companies and applications. Figure 5 shows how messages are published by various publishers, passed through an exchange, queued, and then subscribed to by the subscribers.

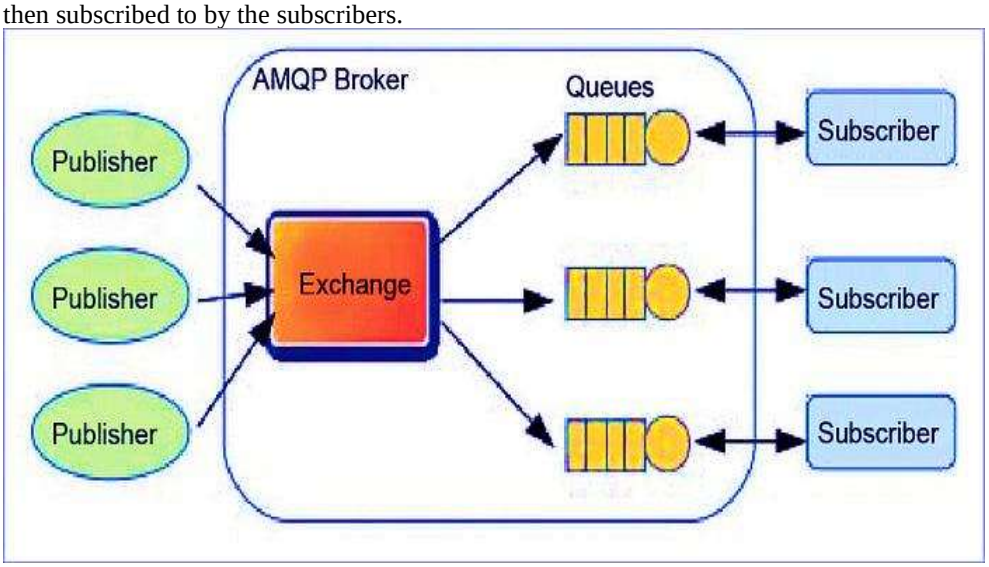


Figure 5: A flowchart depicting the structure and message flow in an AMQP (Advanced Message Queuing Protocol) Broker

The concept of AMPQ was initially formulated by John O'Hara, an individual affiliated with J.P. Morgan Chase, in the year 2003. Its implementation commenced as a collaborative endeavour, with the involvement of the iMatix Corporation. The origin of AMQP can be traced back to its development in the financial industry. Although the system may use many transport protocols, it depends on a stable transport protocol like TCP [12]. This technology's store-and-forward functionality, which ensures resilience in the case of network disruptions, is its main advantage [13]. A further investigation has shown that AMQP 6 can send more messages per second than the previously mentioned REST [14]. Furthermore, according to a report, a messaging system utilizing the Advanced Message Queuing Protocol (AMQP) and accommodating a user base of 2,000 individuals distributed across five continents has demonstrated the capability to handle a voluminous amount of 300 million messages on a daily basis [14].

2.4 DDS

Data Distribution Service (DDS) standard related to real-time systems. It specifically focuses on facilitating data transmission across the various nodes within a messaging architecture that operates on a publish/subscribe model. The Distributed Data Service (DDS) was introduced in the year 2004 and functions as a middleware architecture designed to facilitate the implementation of a publish/subscribe communications pattern. Figure 6 shows how various applications interact with the DDS and DLRL layers, and how data objects are exchanged between publishers and subscribers through a DDS Domain.

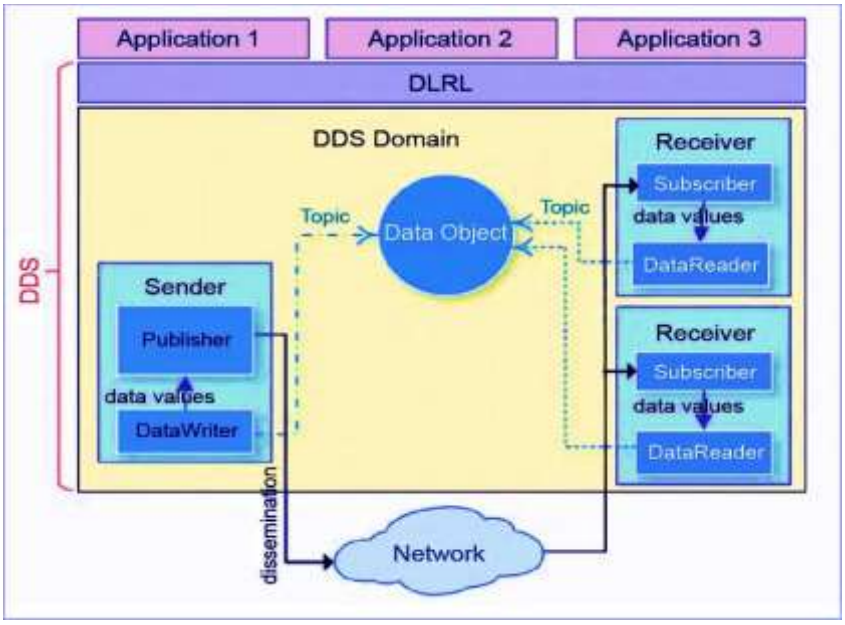


Figure 6: A diagram depicting the Data Distribution Service (DDS) for real-time systems

The Data Distribution Service (DDS) works by giving subscribers and publishers immediate, scalable, and effective communication. The goal of Data Distribution Service (DDS), a kind of networking middleware, is to make network programming less complicated. The system uses a publish-subscribe structure to make it easier for orders, events, and data to be sent and received amongst the different nodes. The organizations that are in charge of creating and sharing data—publishers—create different kinds of information, or topics, such pressure, temperature, and location. These publishers then disseminate certain examples, or samples, of the data. DDS sends the samples to subscribers who show interest in the relevant subject. In distributed systems, the publish-subscribe approach of the DDS effectively minimizes the need for intricate network programming. The protocol configurations are explained in depth in the following.

3 Security Vulnerabilities in IOT Protocols

This includes explaining the security issues that arise with application layer protocols. Security is still a major and urgent problem for IoT systems, which has prompted the creation of a number of

standards, drafts, and research projects. There are security elements incorporated into IoT protocols; nevertheless, these measures alone are insufficient to ensure comprehensive security for IoT systems. Therefore, doing a thorough study is necessary to devise effective countermeasures. The threats are categorized into four layers: Perception Layer, Network Layer, Support Layer, and Application Layer. Each layer lists specific types of threats, providing a detailed overview of potential security issues in IoT systems which is shown in Figure 7.

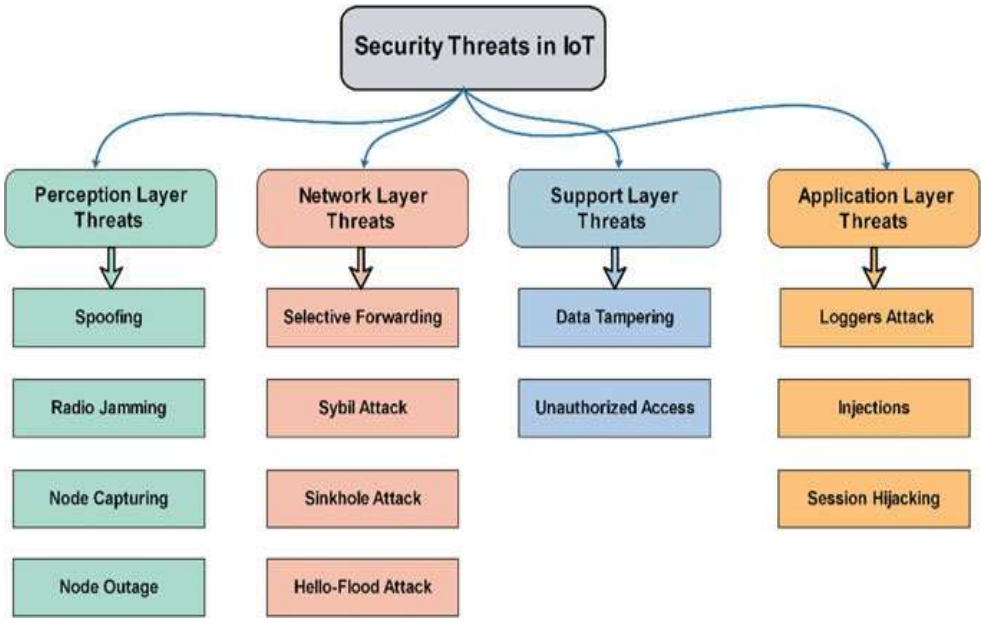


Figure 7: A comprehensive diagram depicting the various security threats in the Internet of Things (IoT)

3.1 Security vulnerabilities

Ever since its debut in the 1990s as a means of connecting several electronic devices via diverse technological means, the concept of the Internet of Things (IoT) has garnered significant attention. To do this, it is essential that you comprehend this. Because of this, the IoT system automatically assumes every security risk and vulnerability associated with the network technology it employs. In addition to using open standards and protocols designed expressly for the Internet of Things (IoT), it is crucial to consider the potential for additional security vulnerabilities resulting from the coexistence and integration of several technologies. The primary security objective of the Internet of Things (IoT) is to protect the acquired data, which may contain sensitive user information in addition to data gathered from physical devices. Thus, it is imperative that the security of any Internet of Things (IoT) system exhibit resilience to threats associated with data while upholding privacy, trust, and data security. Nothing in the user's work needs to be revised in an academic manner. The individuals in question range in age from eighteen to twenty years old. On the basis of the previously indicated challenge classification, this will provide potential directions for security improvements. The Internet of Things (IoT) system comprises three distinct layers, each exhibiting vulnerabilities and susceptibilities to security assaults. To counter these attacks and defend the IoT system, this section presents a multi-layered security approach that ought to be set up to provide the best possible security at every tier of an IoT system.

3.2 IoT Physical Layer Security

a) Secure Booting: The verification of program integrity and device authentication can be achieved through the utilization of cryptographic hash methods, hence enabling the generation of digital signatures. However, the bulk of devices have limited computing power and demand ultra-low power consumption, which makes most cryptographic hash algorithms difficult to implement. However, due to the limited processing capabilities and the requirement for minimal power consumption on the majority of devices, the implementation of most cryptographic hash algorithms

is not feasible. The exception to this is the NH and WH cryptographic hash functions, which have been specifically designed to be optimal for devices with ultra-low power consumption [27, 28].

b) Device authentication: It is a crucial process wherein a newly introduced device within a network is required to authenticate itself before to engaging in data transmission or reception. This authentication process serves the purpose of accurately identifying the device before granting authorization, hence safeguarding the system against potential threats posed by malicious devices.

c) Data integrity: Error detection systems must be included into every device to prevent sensitive data from ever being tampered with. Low-power mechanisms like checksums, parity bits, and cyclic redundancy checks (CRCs) are often preferred. On the other hand, using the WH cryptographic hash function is advised for improved error detection security [19].

d) Data Confidentiality: To preserve the secrecy of data, it is imperative that all RFID tags, identification numbers, and associated information be encrypted on each device prior to transmission. Nevertheless, the implementation of strong cryptographic encryption methods such as AES is hindered due to the ultralow power consumption. Both Blowfish and RSA algorithms exhibit reduced power consumption and need less computational power, making them suitable for implementation on physical layer devices.

e) Anonymity: The concealment of sensitive information, such as the position and identity of nodes, is of utmost importance in certain instances. While the Zero-Knowledge approach is considered the most effective method for ensuring anonymity [20], its implementation on low power devices is not feasible due to its computationally intensive nature. Consequently, the K-anonymity approach [21] is better suited for low-power devices, such as those utilized in an Internet of Things (IoT) system.

3.3 IoT Network Layer Security

a) Data privacy: By using point-to-point encryption and authentication procedures, it is possible to stop unauthorized access to the sensor nodes [22].

b) Routing security: The implementation of safe routing protocols is essential for the broad acceptance and efficient use of sensor networks in several applications. It is important to note, nevertheless, that many of the routing protocols now in use are deficient in the essential security features [23]. However, the security of routing can be guaranteed by using a network architecture that offers numerous paths for data routing. This enhances the system's capability to identify errors and continue functioning even in the presence of known failures [23]. Moreover, the implementation of encryption and authentication systems serves to enhance the overall security of routing data.

c) Data integrity: Cryptographic hash techniques verify the legitimacy of the information received from the other party. Error correction procedures may be used to mitigate the problem in the event that data tampering is shown to have taken place.

3.4 IoT Application Layer Security

a) Data security: At this point, data theft prevention and system privacy are ensured by the use of authentication, encryption, and integrity controls. By limiting unwanted access, they help protect the confidentiality of the system's data.

b) Access Control Lists (ACLs): Protecting the security of the Internet of Things system and the privacy of its data requires establishing policies and permits for who may access and operate it. ACLs have the ability to prohibit or permit incoming or outgoing traffic in addition to granting or refusing access to requests made by different users within or outside of the network.

c) Firewalls: These are very potent security mechanisms that may prevent assaults beyond the scope of authentication, encryption, and ACLs. Weak passwords may be exploited to undermine authentication and encryption. As packets get in, a firewall may filter them to prevent DoS attacks, unwanted packets, and annoying login attempts before the authentication process even begins.

d) Software for security, including anti-virus, anti-spyware, and anti-adware applications: These applications are critical to the IoT system's dependability, security, integrity, and confidentiality. The application layer security concerns related to IoT.

At every IoT layer, risk assessment, intrusion detection, physical security, and trust management must be required to guarantee a system's ongoing safety and credibility.

4 Application layer protocols have security vulnerabilities.

4.1 Security Flaws in the MQTT Protocol

Telemetry Message Queueing While there are a number of security measures in the transport protocol, most of them—like entity authentication and user data encryption—are not activated or set up by default. By registering the device's information when it tries to connect, the user may regulate authentication methods including using the device's physical address (MAC). The broker may perform access authorization using a tool known as an Access Control List (ACL). As its name suggests, the ACL keeps track of data, including the names and passwords of the many clients who are permitted access to certain things. It could also outline what the customer is permitted to do with these things. A secure system must have confidentiality, as stated in References [26, 27]. This may be accomplished at the application layer by encrypting the message that has to be released. You may use this encryption technique end-to-end or client-to-broker. When a broker employs client-to-broker encryption, the broker encrypts the values it must provide to other customers while also decrypting the information being published to a subject. The broker transmits the encrypted text to other devices when everything is end-to-end since it is unable to decrypt the data transmitted to topics. Because it just serves as a messenger in the second scenario and doesn't need extra modules for message encryption or decryption, the broker or user uses less energy and computer power. (7) On lower tiers, the extra security concerns could also be used and handled. As per the findings of [26], the Transport Layer Security protocol (for TCP) or the Datagram Transport Layer Security (for UDP) may be used to dependably ensure the security of a communication channel at the transport layer. Furthermore, any of the many available algorithms, such as Advanced Encryption Standard (AES) in Counter Block Mode or AES in Counter with CBC-MAC mode, often known as CCM mode, may be used to encrypt data at the link layer. Compared to previous approaches, this kind of security mechanism has a few more benefits, such increased efficiency because of the hardware acceleration capabilities of radio chips. Inside [27] Specifically designed for constrained devices, the publisher/subscriber Message Queuing Telemetry Transport protocol was created. For message queuing telemetry transfer (MQTT), TLS/SSL offers transport encryption. It offers protection from listening in on conversations. Device authentication may be enabled using the client ID and username/password credentials that are supplied by the MQTT application at the application layer. The usage of TLS/SSL, which is not meant for tiny devices, is one restriction on MQTT security. It does take time to use TLS/SSL with certificates and session key management for a variety of heterogeneous devices [24, 25]. This calls for the need for a more robust, transportable, and scalable security solution. In order to improve the security of the current MQTT protocol and its variations based on lightweight attribute-based encryption (ABE) over elliptic curves, a secure MQTT (SMQTT) is proposed in [22].

The advantage of using ABE comes from its built-in capability for broadcast encryption, which enables the delivery of a single encrypted message to many intended recipients. It is thus suitable for applications using the Internet of Things. Furthermore, simulations and performance review have shown that the SMQTT technique is viable. The effectiveness of two different ABE types—key-policy ABE and cipher text-policy ABE—has been thoroughly examined in [24] via testing on a range of mobile device classes, including laptops and smartphones. The major advantages of ABE over RSA (an asymmetric cryptographic technique) are scalable key management and flexible, fine-grained access control since senders and receivers are totally separated. ABE is slower, more energy-intensive, and has greater data overhead. It may be difficult to maintain privacy in the context of the Internet of Things since linked devices may produce massive amounts of data, some of which are really personal data. Furthermore, data flow management is challenging in the absence of a user interface or sufficient user tools. [14–16] describes an effective approach to enforcing security policy requirements in the Internet of Things. The SecKit security toolkit, which is included within the MQTT protocol, provides the foundation of this enforcement method. A bespoke C policy enforcement point (PEP) component forms the basis of MQTT's policy enforcement capability. The following tasks are carried out via the PEP connection: uses a publish-subscribe mechanism to 1) intercept the messages being exchanged inside the broker; 2) reports these messages to the broker as events in the SecKit policy decision point implemented in Java; and 3) optionally receives an enforcement action (allow, deny, modify, and delay) to be carried out. Furthermore, a security plugin

has been used to include this PEP into the Mosquitto Broker [23]. The following list lists the advantages of this approach over the features that are currently absent from MQTT implementations.

Message modification and identity obfuscation.

a. Postponing communications to avoid tracking users and devices in real-time.

b. In addition to enforcement upon client subscription to a subject, enforcement occurs upon delivery of a message to a client.

c. Assistance with reactive rules that alert, record, or ask for user permission.

d. Rules for misbehaviour checking to detect denial-of-service attacks.

This approach's primary disadvantage is its high overhead when a single publisher has a large number of interested subscribers, as each subscriber's policy needs to be verified.

4.2 Constrained Application Protocol Security Vulnerabilities (CoAP)

Constrained Application Protocol: Designed to meet IoT requirements for low overhead, this protocol is a fantastic modification of HTTP. The CoAP protocol uses UDP, and DTLS is often used for encryption, however IPSec is also sometimes used. AES/CCM enables non-repudiation, confidentiality, integrity, and authentication. DTLS is utilized at the transport layer. The Java-implemented Californium framework offers a set of security mechanisms for CoAP. For CoAP to use TLS, four security choices have been described [26]. Sensing devices that are pre-programmed with symmetric cryptographic keys do not allow any security PSK. Devices incapable of handling public key cryptography should utilize this mode. Use raw public key (RPK) on any device that requires public key authentication. This setting enables a TLS session without a certificate.

certificates that provide public key authentication, in which the keys are constantly verified by a reputable organization called the certificate authority. The primary disadvantages of utilizing certificates are their fixed costs and large data format. But the ability to revoke certificates in the event that the device is compromised is unquestionably advantageous. One of the CoAP security's shortcomings, key management is a problem that practically all protocols have. Message fragmentation is another issue brought on by the high handshake and high processing costs in the message. Numerous studies offered various approaches for DTLS compression. Indeed, in [25], a unique DTLS header compression approach named Lithe was suggested with the goal of utilizing the 6LoWPAN standard to greatly reduce energy consumption without sacrificing the end-to-end security features. Furthermore, when compressed DTLS is enabled, the evaluation results demonstrate considerable reductions in packet size, energy usage, processing time, and network-wide response times. One obvious drawback of this approach is that it is limited to 6LoWPAN networks and requires DTLS header compression. The use of the transport level protocol (UDP for CoAP and TCP for MQTT) is examined in a security research between CoAP and MQTT published in [26]. This essentially requires the use of DTLS for CoAP and TLS for MQTT. Furthermore, although CoAP has a number of security modes and ciphers that must be employed, the MQTT standard just raises security-related issues without recommending any particular implementations. The four previously described security solutions have been considered in the comparison investigation. This research suggests that, even in cases where MQTT does not support it, RPK is a mixed security option that may be used instead of lower PSKs and heavier certificates. The maximum level of security is provided by the conventional certificate-based encryption and authentication system. Furthermore, as HTTP has previously shown, the ability to revoke certificates strengthens HTTP's resistance against a variety of threats, including those resulting from illegal usage. Furthermore, the interoperability problem has a non-trivial solution due to the many standard security techniques, which mostly rely on security-level negotiation between IoT devices.

4.3 Advanced Message Queuing Protocol (AMQP) Security Vulnerabilities

The CIA trinity of cyber security—confidentiality, integrity, and availability—becomes increasingly crucial as the Internet of Things grows to include billions of devices worldwide. There is an exponential increase in the number of communication lines and data transmission channels, whether via wired or wireless connections, in tandem with the exponential expansion of IoT devices. In reality, the number of communications channels in a scenario where each device is able to communicate with every other device is n(n-1)/2, where n is the total number of devices. Every IoT

communication route is just as susceptible to a potential cyberattack by a man-in-the-middle as a straightforward email exchange between two end users.

These four kinds of aggressive assaults are as follows:

• Replay: To gather information by impersonating a user, an attack entity replays data in between communication sessions.

• Masquerade: By impersonating an authorized entity, an attack entity gets access to a system or does a harmful act.

• Modification: The content of the network communication is added to or removed by an attack entity.

Denial of service attacks stop authorized users from accessing computer services. Given enough time, skilled threat actors will be able to intercept and decode IoT interactions even if AMQP uses TLS/SSL-based encryption on an underlying TCP-based transmission protocol. Well-designed botnets are being introduced into IoT networks, presenting a danger to their security, in addition to being introduced into the market with inadequate security measures (Arias et al., 2015). An incident of this kind occurred at an American institution that has not been named yet in the beginning of 2017 [26, 27]. Cybercriminals used an IoT device's inadequate security features and a brute force assault to break the default or incorrectly set passwords. Once they gained control of this device, they were able to install malware that had been specially built (Palmer, 2017). The malware subsequently spread to other Internet of Things (IoT) devices using a botnet that again used brute-force techniques to break weak or default passwords. With every virus update, the botnet expanded, making it impossible for administrators to access impacted devices and requiring them to repeatedly change their passwords (Moss, 2017). All 5,000+ devices were infiltrated in a short amount of time and were each sending hundreds of DNS requests to seafood restaurants (Mezzofiore, 2017) [26].

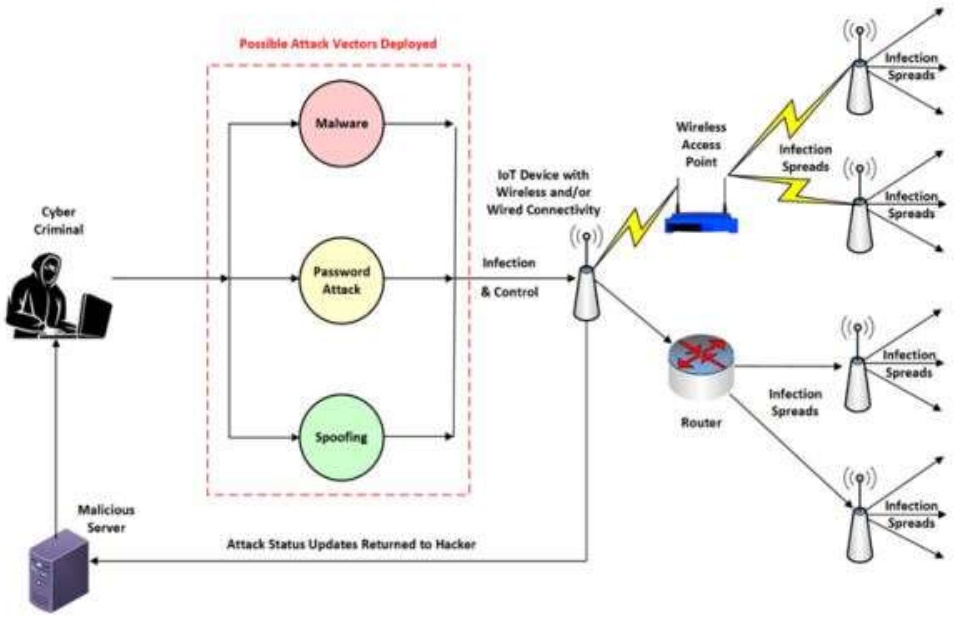


Figure 8: Hypothetical Interpretation of Attack

A theoretical graphical analysis of the potential origin and dissemination of the botnet attack against the University is presented in Figure 8. This DDoS attack caused the University's Internet connectivity to be severely slowed down, which prevented staff and students from accessing the resources that they needed (Palmer, 2017). Being one of the few instances of a botnet DDoS attack expanding and affecting the same network as the compromised devices, this occurrence is very noteworthy. A number of IoT networks may be seeded with such internally-spreading malware in

the event that such an assault entails the penetration of server-to-server interactions hosting AMQP, which might result in a widespread compromise of AMQP-enabled IoT devices [26].

4.4 Security Vulnerabilities in Data Distribution Service (DDS)

According to the Object Management Group (2015), the main purpose of the open standard known as Data Distribution Services (DDS) is to enable peer-to-peer communication between devices. In terms of data, the existing protocol establishes a publish/subscribe architecture where communication between a device and a server or between two servers is not as important as quick device-to-device connections. The standard defines DDS as a Data-Centric Publish-Subscribe (DCPS) paradigm that facilitates communication and integration across dispersed applications. This standard describes the Application Interfaces (APIs) and Communication Semantics (behavior and quality of service) that facilitate the effective transmission of information from information producers to relevant consumers. According to the Object Management Group (2015, p. 1), industrial control systems (ICS) have been seen as a serious concern in the recent past (Harp & Gregory-Brown, 2016). A great deal of legacy systems still depends on pre-established standards, protocols, and applications that were mostly put into place when device contact was low and physical assaults predominated. Nevertheless, in a globally interconnected society, Industrial Control Systems (ICS) are attracting the interest of malicious actors in the cyber realm. In a notable incident, the electrical grid of Ukraine saw a targeted attack in 2015 (Lee, Assante & Conway, 2016). The assailants successfully infiltrated the SCADA systems, thereby causing significant disruptions and impairing the overall functionality of the grid. This incident represents an early instance of a successful cyber-attack on electrical infrastructure, underscoring the increasing risk posed by advanced offensive operations targeting cyber-physical systems. A useful technique in many high-stakes businesses, such as the aerospace and energy sectors, as well as the military, is distributed denial of service, or DDS. Wang et al. (2008) performed a research that investigated the use of DDS in combat systems and network-centric operations. The study highlighted the increasing acceptance of DDS in security-sensitive scenarios. The widespread utilization of the DDS protocol in military contexts is to be expected, given its origins in Thales' TACTICOS Combat Management System (Thales, 2015). The utilization described has been a significant factor contributing to the implementation of demanding specifications for high performance and robust design in the context of DDS. The DDS protocol relies on TLS to primarily handle security measures, rather than implementing security measures at the application layer. Nevertheless, it is evident that relying solely on Transport Layer Security (TLS) is inadequate, as demonstrated by the development of a standardized security specification for post-protocol ratification known as DDS Security. By adding components like authentication, authorization, non-repudiation, secrecy, and integrity, this supplemental definition improves the security aspects of DDS implementations (Object Management Group, 2016). He and Liang (2015) looked at security issues in their examination of the DDS standard. They described an imaginary scenario in which unapproved publishers or subscribers may perhaps add data to the DDS network or get access to data that wasn't meant for them. The writers provide a thorough rundown of potential attacks against DDS, and the DDS Security architecture is designed to counter and mitigate these types of attacks. Unfortunately, there is not enough research available now on how well the DDS Security standard works to counter the potential attacks that are identified. To reduce the possible threats brought about by the range of vulnerable network protocols used in the Internet of Things (IoT), further research is needed. This is especially important since data breaches may have serious financial repercussions. Furthermore, it is crucial to investigate strategies that might reduce possible vulnerabilities in installations of vital infrastructure. A set of lab tests carried out to evaluate a subset of vulnerabilities specific to the DDS protocol are presented in this section.

This paper provides an overview of the MQTT application layer protocol, which is widely utilized in several domains. It also discusses the security challenges associated with MQTT and presents some solutions to mitigate these vulnerabilities. While the existing research on MQTT security is limited, there have been first efforts to address its security concerns. The majority of security issues that develop are attributed to the operational conditions of the protocol, primarily stemming from the absence of universally accepted standards. These deficiencies include a lack of authentication, authorization, secrecy, and integrity.

4.5 Problems and Hazards in the MQTT Protocol

Many different security techniques are included in MQTT, however most of them are not easily accessible by default or pre-configured. Entity authentication and data encryption are two examples of such techniques. There are authentication techniques that are managed by the broker, such as the

use of the Media Access Control (MAC) address as a form of identity. This is the procedure that registers the relevant data that a device tries to connect with on its first try. Through the use of an Access Control List (ACL) mechanism, the broker is able to carry out access authorization. The Access Control List (ACL) is a database that holds relevant information about which clients are authorized to access which objects, such as client IDs and passwords. The ACL also offers the ability to specify which particular activities a client is allowed to do on these objects. As mentioned in References [26,27], maintaining confidentiality is essential to guaranteeing system security. By encrypting the message that is meant to be distributed, encryption methods may be used at the application layer to efficiently accomplish this goal. There are two possible implementations of the encryption technique that is presented: client-to-broker and end-to-end. In client-to-broker encryption, the relevant values that are intended to be provided to other clients are encrypted, and the broker decrypts the data being exchanged on a specific topic. In a complete case, the broker transmits the encrypted data to other devices after failing to decode the information being sent to subjects. There is nothing in the user's content that has to be rewritten in an academic way. The broker uses less energy and processing resources in the latter method since it just acts as a messenger and doesn't need any additional modules for message encryption or decryption. Nevertheless, additional security elements may also be applied at lower layers. According to the cited references [26, 27], a reliable method of ensuring the security of a communication channel at the transport layer is to use the Datagram Transport Layer Security (DTLS) for User Datagram Protocol (UDP) or the Transport Layer Security (TLS) protocol for Transmission Control Protocol (TCP). Furthermore, as noted in reference [27], encryption at the link layer may be achieved using a number of algorithms, such as Advanced Encryption Standard (AES) in Counter Block Mode or AES in Counter with CBC-MAC mode, often referred to as CCM mode. This particular security mechanism has many other benefits over other approaches, chief among them being higher efficiency as a result of using radio chip hardware acceleration capabilities. The user did not offer any text for modification. The Message Queuing Telemetry Transport (MQTT) protocol is a communications protocol that is specifically designed for low-resource devices, using a publisher/subscriber architecture. The Message Queuing Telemetry Transport (MQTT) protocol cannot be secure unless data is encrypted during transmission using Secure Sockets Layer (SSL) or Transport Layer Security (TLS). It offers protection against unauthorized communication interception. At the application layer, the MQTT application offers client identification and username/password credentials that may be used for device authentication. One drawback of MQTT security is the usage of TLS/SSL, which may not be suitable for devices with limited resources. In reality, using TLS/SSL with certificates and session key management for a broad range of disparate devices may be considered challenging [28]. Thus, implementing a more dependable, lightweight, and scalable security solution is essential. The authors of [28] proposed the secure MQTT (SMQTT) protocol as a means of enhancing the security of the current MQTT protocol and its variants. Using lightweight attribute-based encryption (ABE) techniques over elliptic curves allows for this. One of the main benefits of using Attribute-Based Encryption (ABE) is its inherent broadcast encryption feature, which allows one encrypted communication to be safely sent to several intended recipients. This characteristic makes ABE particularly suitable for Internet of Things (IoT) applications. Furthermore, performance evaluations and simulations have shown the SMQTT technique's feasibility. In a previous work [27], the performance of two distinct kinds of attribute-based encryption (ABE) schemes—key-policy ABE and cipher text-policy ABE—was assessed on a variety of mobile devices, including laptops and smartphones. This research set out to provide a thorough analysis of ABE approaches and the accompanying efficacies. The RSA algorithm outperforms ABE in terms of performance, data overhead, and energy consumption. But the fundamental benefit of employing ABE is that it makes flexible and granular access control feasible, and it also makes scalable key management possible by isolating senders and recipients. The Internet of Things (IoT) generates a vast quantity of data, some of which may be deemed personally identifiable information. As a result, safeguarding privacy in this environment is a difficult undertaking. The user provided a reference number [27]. Furthermore, it is challenging to effectively control data flow in the absence of a user interface and insufficient tools. Academic research [18, 21, 22] offers a productive method for implementing security policy specifications in the Internet of Things (IoT). The model-based security toolkit SecKit, which is integrated within the MQTT protocol, is the recommended enforcement technique. The core of the MQTT policy enforcement capabilities is a specially designed policy enforcement point (PEP) component written in C. In the role of mediator, the PEP performs the following tasks: It does this by 1) intercepting the messages being delivered within the broker via a publish-subscribe technique; 2) notifying these messages as events in the Java-implemented SecKit policy decision

point; and 3) optionally requesting the execution of an enforcement action (such as approve, reject, amend, or postpone). Additionally, the PEP has been integrated into the Mosquitto Broker via a security plugin [23]. The benefits of this method over the features missing from current MQTT implementations are enumerated in the following list.

- Alteration of communication content and concealment of personal identification.
- Delays in message delivery are caused by the need to confirm the real-time monitoring of users and Internet of Things (IoT) enabled devices.

Not only should you let customers know when they get a message successfully, but you should also recognize that they have subscribed to a certain subject. A clause allowing for the addition of reactive rules makes it easier to notify, log, or ask for user permission. the guidelines for keeping an eye out for inappropriate behavior and, in particular, identifying Denial of Service (DoS) assaults.

One of the primary challenges associated with this technique pertains to the significant overhead that arises when a single publication attracts numerous interested subscribers, necessitating the evaluation of a policy for each individual subscriber. This overhead incurs a little lag of a few milliseconds. In this discourse, our focus is limited to the examination of security authentication inside the realm of protocols, specifically emphasizing the Application layer, with a particular emphasis on the MQTT protocol. The attainment of comprehensive security measures presents network obstacles as a result of the disparity between the substantial need for security protocols and the limited hardware capabilities currently envisioned. Unsecured protocols, lacking security-based implementations, are frequently susceptible to a range of network assaults, including eavesdropping and spoofing. It's crucial to remember that even with security measures like DTLS, IPSec, SSL/TLS, and other comparable protocols in place, complete security cannot be guaranteed. IPSec has to overcome challenges related to multicast communication, port address translation (PAT), and network address translation (NAT). Because group key management is absent from DTLS, multicast communications are not handled by it. Both DTLS and IPSec have limitations concerning network trust, access control, and Quality of Service (QoS). To overcome these limitations, they rely on extra protocols such as Extensible Authentication Protocol (EAP) and Internet Key Exchange (IKE). There are substantial expenses associated with using SSL/TLS on devices with minimal resources. There is nothing in the user's content that has to be rewritten in an academic way. Vulnerabilities are inherent weaknesses in a system that arise from less-than-ideal design decisions, making unapproved network incursions possible. An attacker may take advantage of faulty error handling, data manipulation, buffer overflow vulnerabilities, cross-site scripting vulnerabilities, improperly controlled network access and permissions, and lax data authentication procedures. Table 1 provide an exhaustive list of security issues and their related fixes for several protocols in the AL framework, including MQTT. The user has not provided any text to rewrite. The identified vulnerabilities include attacks on online interfaces, concerns with data privacy, vulnerabilities in mobile and cloud interfaces, inadequate authentication and authorization measures, privacy breaches, denial-of-service attacks, the presence of malicious code, and instances of social engineering.

Table 1: Solutions and Counter measure

Sl. No.	Vulnerability/ Challenges	Solutions Proposed
1	DoS Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
2	Sphear Phishing Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
3	Sniffing Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters 3 Risk transfer mechanism based security systems 4 Support for Software Defined Networks (SDNs) architectures
4	Overwhelm Attack	1 Dynamic threat anticipation ASTM 2 Adaptive learning technique with changing internal parameters

		3	Risk transfer mechanism based security systems
		4	Support for Software Defined Networks (SDNs) architectures
5	Insecure web interface & Data Privacy	1	Preference Based Privacy
		2	Protection Method - Third party evaluation, report to service provider and appropriate security level based sensed preferences
6	Insecure mobile interface & Cloud Interface	1	Stronger passwords
		2	Testing the interface against the vulnerabilities of software tools (SQLi and XSS)
		3	Using https along with firewalls
7	Insecure Remote Security Configuration	1	Remote safe configuration
		2	Scalable security enhancement system of the SMC model for distributed resources – SMSC
		3	Simplified security management of network security teams
8	Insecure Software/Firmware	1	Encryption with validation
		2	Anti-virus, anti-adware, firewalls, Real Time Intrusion Detection Systems (IDS)
		3	Security patches
		4	Code with languages such as JSON, XML, SQL and XSS needs to be tested carefully
9	Insufficient Authentication/Authorization	4	Cross-layer authentication and authorization Sensitive information isolation/Data leakage protection
		5	Administrator/Identity Manager Authentication Effective Key coordinate sharing, frequent key coordinate updates
		6	Identity Authentication and Capability based Access Control (IACAC)
		7	Strong Encryption schemes
		8	Cryptographic Hash functions & Feature Extraction
		9	Decentralized control of authentication using user-dependent security context
10	Risk Assessment/Trust Management	1	Security quantified in terms of incident and asset loss – CCM
		2	Mutual trust for inter-system security
		3	Agent-based and weight-based trust models
11	Lack of Protocol Standardization	1	Smart Object Lifecycle Architecture for Constrained Environments (SOLACE)
12	Existing protocols coping with newer & stronger threats	1	TLS/DTLS and HTTP/CoAP mapping
		2	Mirror Proxy (MP) and Resource Directory
		3	TLS-DTLS tunnel and message filtration using 6LBR .

The creation of a successful Internet of Things (IoT) application is still a difficult task fraught with many challenges, even with the large amount of effort and established standards in this industry. The following concerns are included in this context: cost, energy harvesting, availability, mobility, interoperability, scalability, and management. When it comes to traditional network security, it is easier to overlook and failsafe the network that links Internet of Things (IoT) devices to back-end systems. This is mostly due to the broader array of communication protocols, standards, and device capabilities involved in IoT networks. To ensure the proper and efficient functioning of apps, it is important to enable a multitude of security principles at every layer. MQTT is considered to be the most extensively utilized application layer protocol. This study focuses on the analysis and selection of security vulnerabilities and dangers pertaining to the Application Layer Protocol MQTT. The majority of security issues that develop are attributed to the operational conditions of the protocol, primarily stemming from the absence of universally accepted standards such as authentication, authorization, secrecy, and integrity. It is clear that resolving these common security issues calls for the deployment of suitable protocol configurations to improve Internet of Things (IoT) trust.

5 Conclusion

This paper offers a comprehensive analysis of the protocols used in the Internet of Things (IoT) and addresses security concerns associated with it. The primary objective of this research is to provide

developers and service providers a comprehensive understanding of the various Internet of Things (IoT) protocol tiers, with a focus on the Application Layer IoT protocols, such as MQTT, AMQP, XMPP, CoAP, and DDS. The study will go into the complexity of the MQTT protocol and examine its strengths and limitations. Countermeasures will also be investigated in order to identify and predict more effective setup processes for MQTT protocols. Security is one of the biggest problems in the realm of Internet of Things (IoT) technologies. On the other hand, in order to minimize vulnerabilities and thwart attacks, firms using Internet of Things (IoT) systems need to give first emphasis to offering consumers appropriate security standards. Attackers who have been successful have shown intelligence since their accomplishments are based on a solid understanding of the world. It is also clear that designers need to be more intelligent in order to keep ahead of possible attackers and guarantee the success of IoT initiatives. The ongoing rivalry between the two factions persists indefinitely, as neither possesses absolute wisdom, which denotes comprehensive knowledge and understanding. In order to ensure progress and advancement, it is imperative that research in the field of Internet of Things (IoT) remains ongoing indefinitely. It is foreseeable that any existing technological limitations will eventually be overcome, as is the case in several other disciplines. Future research will focus on the implementation of these protocols and conducting an experimental study to quantitatively compare them.

Ethics declarations
Not applicable.

Consent for publication
Not applicable.

Competing interests
No competing interests.

Funding
Not applicable.

Author information
Sushil Kumar Bhoi, Umashankar Ghugar, Satyabrata Dash, Rakesh Nayak and Dilip Kumar Bagal

Corresponding author
Correspondence to Sushil Kumar Bhoi (Contact: sushilkumarbhoi@gmail.com).

References

[1] P. N. Mahalle, N. R. Prasad, R. Prasad – "Object classification based context management for identity management in internet of things", International Journal of Computer Applications, vol. 63, no. 12, 2013

[2] Fei Hu – "Security and Privacy in Internet of Things (IoT). Models, Algorithms and Implementations", CRC Press, 2016

[3] Thamer A. Alghamdi – "Security Analysis of the Constrained Application Protocol in the Internet of Things", IEEE (2013)

[4] C. Hongsong, F. Zhongchuan, Dongyan – "Security and Trust Research in M2M System", Vehicular Electronics and Safety (ICVES), 2011 IEEE International Conference

[5] Vasileios Karagiannis, Periklis Chatzimisios, Francisco Vazquez-Gallego, Jesus Alonso-Zarate – "A Survey on Application Layer Protocols for the Internet of Things", Transaction on IoT and Cloud Computing 2015

[6] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," IEEE Communications Magazine, vol. 48, no. 9, pp. 140–150, 2010.

[7] Dan Dinculeană and Xiaochun Cheng, "Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices", Appl. Sci. 2019, ISSN 2076-3417 9, 848; doi:10.3390/app9050848

[8] A. Schmidt and K. Van Laerhoven, "How to build smart appliances?" IEEE Personal Communications, vol. 8, no. 4, pp. 66–71, 2001.

[9] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: a survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 1, pp. 402–427, 2013.

[10] ITU Telecommunication Standardization Sector, "ITU-T Recommendation database," 2012. [Online]. Available: http://handle.itu.int/11.1002/1000/11559-en?locatt=format:pdf&auth. [Accessed 13 April 2015].

[11] H. Ning and Z. Wang, "Future internet of things architecture: like mankind neural system or social organization framework?" IEEE Communications Letters, vol. 15, no. 4, pp. 461–463, 2011.

[12] Kun Wang Jianming Bao and Meng Wu Weifeng Lu "Research on Security Management for Internet of Things" 201O International Conference on Computer Application and System Modeling Vol-I5 pg- 133 to 137.
[13] Sarfraz Alam, Mohammad M. R. Chowdhury and Josef Noll "Interoperability of Security-Enabled IoT" Wireless Commun (2011) vol-61:567–586.
[14] Hui Suo, Jiafu Wan, Caifeng Zou and Jianqi Liu "Security in the Internet of Things: A Review" 2012 International Conference on Computer Science and Electronics Engineering pg- 648 to 651.
[15] Wang Chen "AN IBE-BASED SECURITY SCHEME ON INTERNET OF THING 2012 Proceedings of IEEE CCIS pg- 1046 to 1049.
[16] Stefan Poslad, Mohamed Hamdi and Habtamu Abie" International Workshop on Adaptive Security & Privacy management for the Internet of Things" 2013 pg-373 to 378
[17] Mirza Abdur Razzaq, Muhammad Ali Qureshi, Sajid Habib Gill, Saleem Ullah "Security Issues in the Internet of Things (IoT): A Comprehensive Study" International Journal of Advanced Computer Science and Applications, Vol. 8, No. 6, 2017.
[18] A. Arun Raja, R. Naveedha, G. Niranjanadevi and V. Roobini "An Internet Of Things (IoT) Based Security Alert System Using Raspberry Pi" Asia Pacific International Journal Of Engineering Science Vol. 02 (01) (2016) 37–41
[19] Hafsa Tahir, Ayesha Kanwer and M. Junaid "Internet of Things (IoT): An Overview of Applications and Security Issues Regarding Implementation" INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND ENGINEERING, VOL. 7, NO. 1, JANUARY 2016.
[20] Mario Ballano Barcena and Candid Wueest "Insecurity in the Internet of Things" http:// www.symantec.com Version 1.0 – March 12, 2015.
[21] J. Sathish Kumar and Dhiren R. Patel "A Survey on Internet of Things: Security and Privacy Issues" International Journal of Computer Applications (0975 – 8887) Volume 90 – No 11, March 2014 pg- 20 to 26.
[22] Suchitra.C and Vandana C.P "Internet of Things and Security Issues" IJCSMC, Vol. 5, Issue. 1, January 2016, pg.133 – 139.
[23] KANG Kai1, PANG Zhi-bo and WANG Cong "Security and privacy mechanism for health internet of thing" December 2013, 20(Suppl. 2): pg-64–68 www.sciencedirect.com/science/journal/10058885.
[24] Kai Zhao and LinaGe "Survey on the Internet of Things Security" 2013 Ninth International Conference on Computational Intelligence and Security. pg-663 to 667
[25] Pallavi Sethi and Smruti R. Sarangi "Internet of Things: Architectures, Protocols, and Applications," Hindawi Journal of Electrical and Computer Engineering Volume 2017, Article ID 9324035, 2017.
[26] http://en.wikipedia.org/wiki/Advanced Message Queuing Protocol, cited 28 Jul 2014.
[27] Frank T. Johnsen, Trude H. Bloebaum, Morten Avlesen, Skage Spjelkavik, Bjørn Vik, Evaluation of Transport Protocols for Web Services, Military Communications and Information Systems Conference (MCC), 7-9 Oct. 2013, pp. 1-6.
[28] Joel L. Fernandes, Ivo C. Lopes, Joel J. P. C. Rodrigues, Sana Ullah, Performance Evaluation of RESTful Web Services and AMQP Protocol, Fifth International Conference on Ubiquitous and Future Networks (ICUFN), 2-5 July 2013, pp. 810-815.
[29] Angelo P. Castellani, Mattia Gheda, Nicola Bui, Michele Rossi, Michele Zorzi, Web Services for the Internet of Things through CoAP and EXI, IEEE International Conference on Communications Workshops (ICC), 5-9 June 2011, pp. 1-6.
[30] Sye Loong Keoh, Sandeep S. Kumar, Hannes Tschofenig, Securing the Internet of Things: A Standardization Perspective, Internet of Things Journal IEEE (Volume: 1, Issue: 3), June 2014, pp. 265-275.
[31] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, Mischa Dohler, Standardized Protocol Stack for the Internet of (Important) Things, Communications Surveys & Tutorials IEEE 15(3), 2013, pp. 1389-1406.