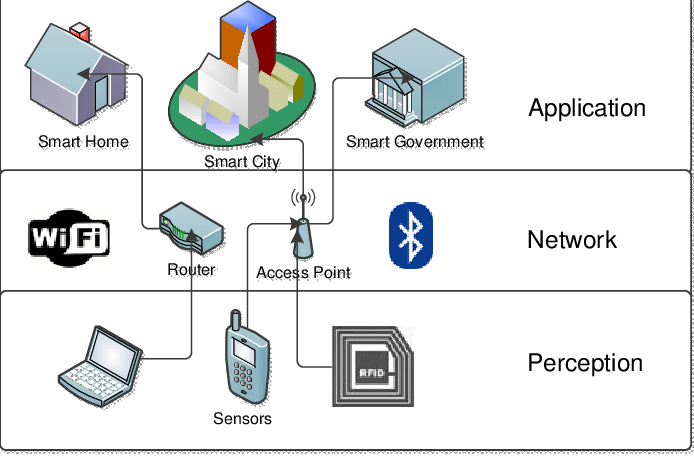
# **2.0 BACKGROUND STUDY**

This section will emphasise on all the research works that have been accomplished before starting this project. This will help us to build an understanding on the concept of the different layers in an IoT system, the communication protocols used by IoT devices, the concern of security threats prevailing in an IoT network and the existing security features that an IoT device possess.

## **2.1 IoT Architecture**

In IoT, each layer has different functionalities and different devices are used in each layer. Due to the lack of standardization in the IoT domain, there are different architectures for IoT that have been proposed namely the three-layer architecture and the five-layer architecture. However, many researchers assert that the three-hierarchical layer architecture is the generally known structure to describe how an IoT systems and its components operate. It is formed by the Perception, Network and Application layer.



*Figure 1: Three –layer IoT architecture*

1. Perception Layer

The perception layer is the first layer in the IoT three-layer architecture. The main purpose of this layer is that it takes in information from the real-world environment with the help of IoT devices such as IoT sensors, actuators and sensor gateway. It senses, retrieve and process the data from IoT devices and collected information gets transmitted to the above layer (i.e. the network layer).

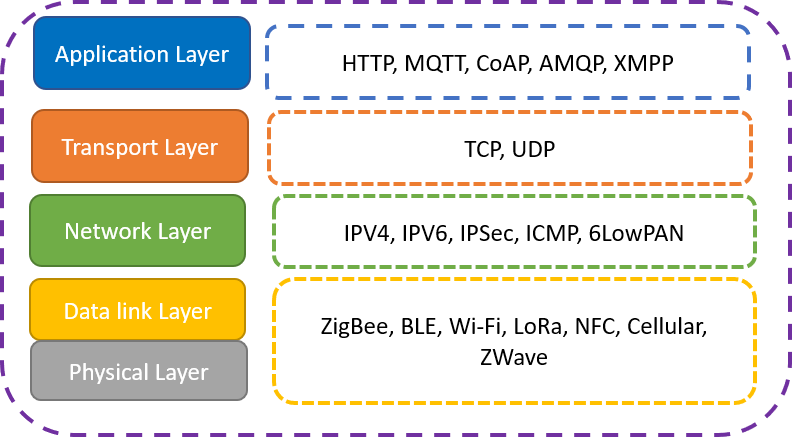
1. Network Layer

The network layer is the middle layer of the IoT three-layer architecture and it is used to determine the route of information provided by the perception layer. The collected data are transmitted to the uppermost layer (Application layer) securely over the Internet. This layer consists of different network components such as cloud-computing platforms, internet gateways, switching and routing devices and communicates using wireless protocol such as WiFi, LTE, Bluetooth, ZigBee, 5G, etc… Since more and more applications are converging towards cloud technology, Internet gateways are an important part of the network layer.

1. Application layer

The functionality of the application layer is to supply various application service to users. It does so by specifying different deployment scenarios for the IoT, such as in smart cities, smart transportation, smart agriculture and so forth. For example, in a smart home, users can turn on a coffee maker by pressing a button in the app.

**2.2 IoT Protocol Stack**



*Fig 2: IoT Protocol Stack*

IoT protocol stack refers to a suite of different communication protocols at each layer of the TCP/IP model that allows resource-constraint IoT devices to communicate. There are 4 layers in the protocol stack each of which have its own set of protocols.

**2.2.1 Application layer**

The application layer is responsible for facilitating interaction between the user and sensor application through the use of an interface. Some of the common protocols that sits in the application layer are:

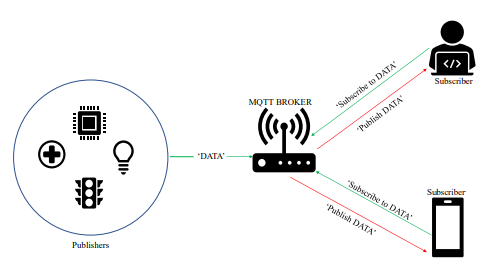
* MQTT (Messaging Queuing Telemetry Transport)

MQTT is a lightweight publish-subscribe protocol used mainly for resource-constraint devices with low bandwidth. It connects to port 1883 where data are transmitted insecurely. The architecture of MQTT is based on a client-server model where the client is the publisher or subscriber and the server is the broker.

Publisher are the sensor devices that send messages to the topic through a broker. These messages typically contain information about sensor readings that other clients can subscribe to and receive. Topic categorizes these messages and only interested clients that have subscribed to that topic will receive messages.

Subscribers are clients that receives messages from the broker. The subscriber subscribes to certain topics that they are interested in order to receive any messages that the publisher has published to those topics.

A broker acts as the central hub and is responsible for creating communication between the publisher and subscriber. The broker receives the messages from the publisher and forwards them to the subscriber. The broker will use the list of topics submitted by the publisher and will send specific topic to specific clients that have subscribed to it.



*Fig 3: MQTT architecture*

* Secure MQTT (SMQTT)

SMQTT is an extension to simple MQTT protocol which uses lightweight based encryption. Publishers encrypts the data and subscribers decrypts it with a master key.

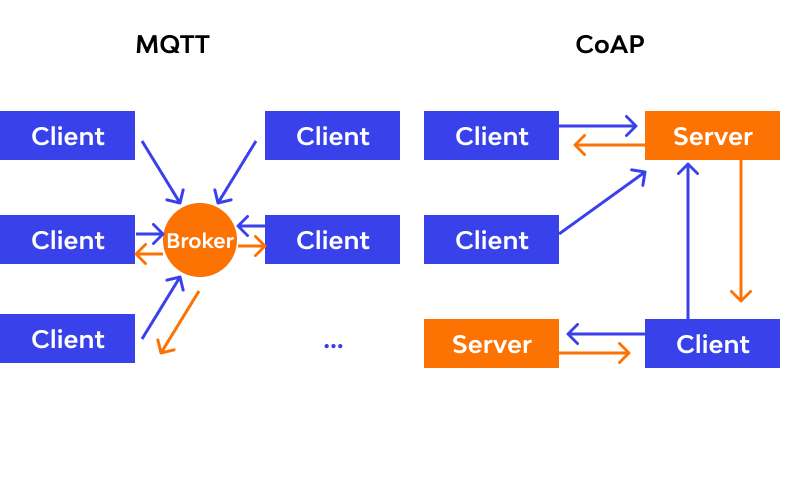
* CoAP (Constrained Application Protocol)

CoAP is a lightweight protocol designed for resource constraint devices with low power such as IoT devices. Just like HTTP which employs client-server model, this protocol has a simple request/response model where the clients initiate the request and server responds to those requests. Compared to HTTP, it uses less resources which makes it suitable to be used in IoT applications.

It uses UDP as an underlying transport protocol to allow for better communication and reliable delivery between endpoints. One of the key features of CoAP is its support for REST (Representational State Transfer) services with low overhead, making it easy to use a web interface between CoAP clients and server. It defined a set of methods such as GET, POST, PUT or DELETE to retrieve, create, update or delete resources.

It supports four different messaging types mainly:

* **Confirmable**- requires a response, either a positive or negative acknowledgement
* **Non-confirmable**- request used for unreliable transmission, that is messages that do not require an acknowledgement by the server
* **Acknowledgement-** sent by receiver to acknowledge a confirmable message
* **Reset-** indicates some kind of failure when message has been received



*Fig 4: MQTT vs CoAP communication*

**2.2.2 Transport layer**

**2.2.3 Internet layer**

**2.2.4 Data link layer**

## **2.2 IoT devices**

IoT devices are physical devices that are embedded with sensors, software, or other technologies, that can connect to a network wirelessly and transmit their data with other devices. They are mainly design to collect data from the real-world, analyze it and perform some operations based on the analysis. These devices can be smart bulbs, smart thermostats, security cameras, smart watches, etc… In order to access their data wirelessly, IoT devices have mobile or web-based interface that helps users to interact with them.

**2.2.1 Challenges in IoT devices**

There are some key challenges that need to be considered when enhancing the security of these devices. Some of the challenges are as follows:

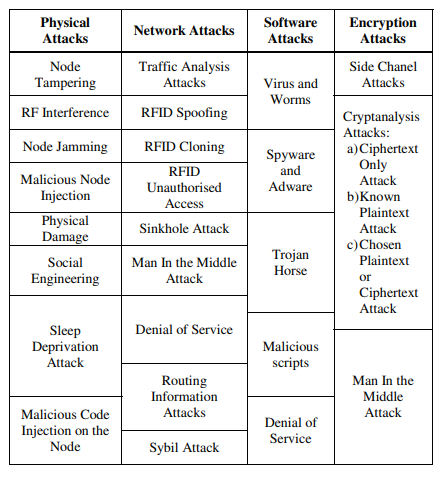
* Memory- They are integrated with small memory size, usually about 10KB of RAM
* Computing power- They do not have the capability to perform complex operations due to their reduced computing power
* Area- They are mainly deployed in small physical area due to their small frequency range
* Battery power- They have limited battery power or can have no battery
* Real-time response- They should provide a quick and accurate response with essential security using available resources

## **2.3 Types of Attacks in IoT System**

One of the most common approaches to IoT networking is that all traditional computers, smartphones and IoT devices are connected to a central gateway in a LAN (Local Area Network) in order to gain internet access. This is common in many application scenarios such as in a smart home or small enterprises.

It is worth noting that multiple users on the same network can control and change the settings of IoT devices remotely through their PC or smartphone. Since these constraint devices use lightweight communication protocols that are not secure, anyone on the local network can intercept the communication, thus gaining unauthorized access to sensitive data.

Moreover, multiple studies have shown a wide range of cyber-attacks occurring in IoT devices. The diagram below illustrates a range of different types of attacks that can occur in IoT networks. These include physical, network-based, firmware, and encryption attacks.



*Figure 2: Several types of attacks that IoT devices are vulnerable to*

### **2.3.1 Physical and Close Proximity attacks**

Normally, the IoT device are often deployed in uncontrollable and outdoor environments, making them vulnerable to physical attacks where attackers can easily interfere with hardware components of the devices by manipulating their power source, memory, etc. These types of attacks affect the functionality of the device and the latter may perform undesirable actions. Some physical attacks are discussed below:

1. Tampering Node

Tampering node is a type of physical attack in which an attacker gains physical access to an IoT device or node and modifies its hardware components or even examining the nodes in order to compromise its security or disrupt its operation. Once an attacker has access to the node, confidential information such as cryptographic keys or routing tables may be known.

1. Malicious code injection

This is also a type of physical attack where an attacker exploits a vulnerability in the firmware of a sensor to inject malicious code that would allow them to manipulate data being transmitted or take control of the sensor. For instance, this can occur if the attacker inserts a USB stick which has a harmful software onto the node. Alternatively, an attacker could exploit a vulnerability in the wireless communication protocols used to transmit data from the sensor to gateway/cloud, allowing them to inject malicious code into the data stream.

1. Node Replication

In this attack, an attacker typically creates a fake node on the network by replicating existing node’s ID. In this way, the fake node is regarded as the legitimate one. All the communication data that occurs between the legitimate node and its neighbors are cloned onto the fake node. Packets that are transmitted over the network can be corrupted or directed to a wrong destination.

1. Node Jamming in WSN

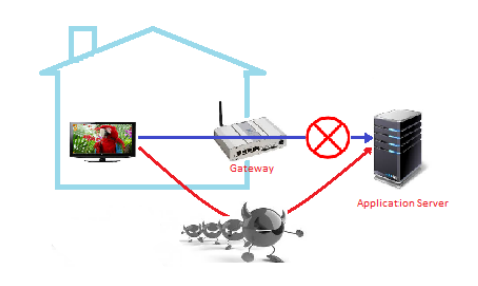
This is a type of attack that occur in the Wireless Sensor Networks where an attacker sends high-power radio signals to cause interference between the communication of the sensor nodes. This cause complete failure in the network since the nodes are unable to communicate effectively.

* + 1. **Network-Based attack**

The collected data from the sensors are transmitted wirelessly to a gateway or between sensor nodes using lightweight protocol that are not secure. Thus, this makes them more vulnerable to network-based attacks. Some network attacks are discussed below:

1. Falsification

In a smart home, when the collected data from sensors are transmitted to the application server, data packets are collected by attackers by altering the routes in the gateway. Although SSL certification is applied, attackers can still avoid the forged certificate. As a result, the original data transmitted by a sensor may be replaced with false one, which in turn can be used to manipulate the behavior of the IoT system. For example, a false temperature reading may cause the heating or cooling system to turn on or off incorrectly.



*Figure 3: An example of falsification*

1. Passive Traffic Analysis

This attack involves monitoring the network traffic from the IoT devices since they are wirelessly connected and as a result the attacker gain sensitive information such as login credentials. This is done by using packet sniffer software that capture and analyse the traffic. Also, by analyzing patterns in the traffic, an attacker can infer information about the network topology and the devices that are connected to the local network.

1. Spoofing

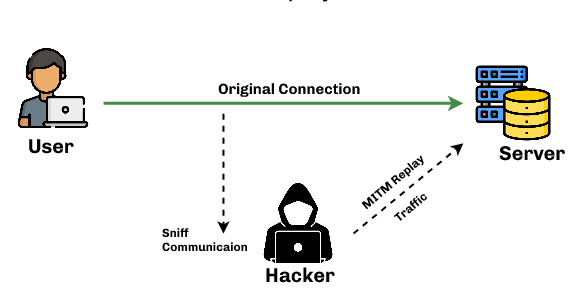
During such attacks, the attack tries to copy the identity of another device in the network and makes it as if the packet is being received from a trusted source. In this way, when the IoT device sends data to the source, that is the attacker’s PC, the latter can manipulate the data or gain sensitive information.

1. Sleep Deprivation

In an IoT system, the sensors normally go to power saving mode in order to preserve their battery life and reduce their power consumption. During this attack, an attacker may send unnecessary traffic to the devices and this may result into a complete exhaustion of the battery resources.

1. Replay attack

A replay attack is a type of network attack that occurs when an attacker intercept network traffic between two devices. The attacker then captures the data transmission between two IoT devices, and then replays it at a later time to one of the devices, which makes it to appear as if he was the sender. For example, an attacker could capture a valid command from an authorized user to unlock a smart lock, and then replay the command to the lock at a later time to gain unauthorized access. The goal of this attack is to trick a device to carry out a malicious action.



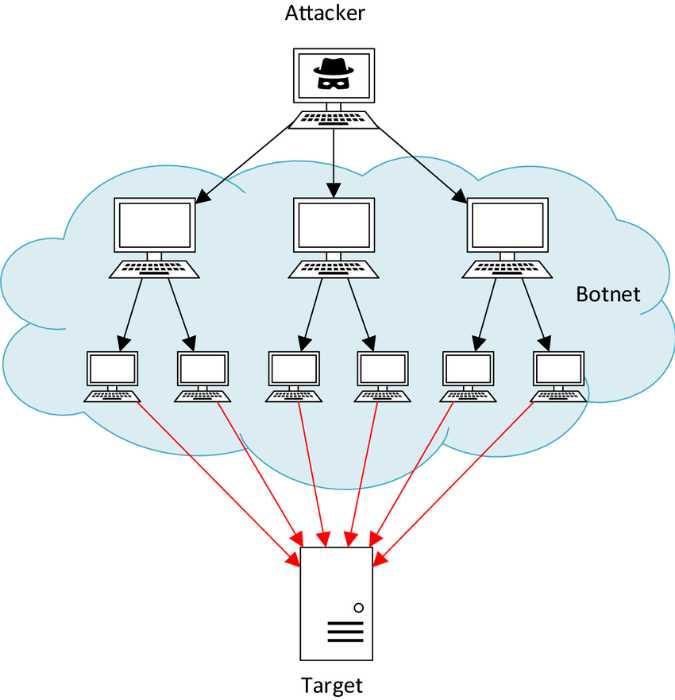
*Figure 4: An example of a replay attack*

1. DoS (Denial-of-Service)/ DDoS (Distributed Denial-of-Service) attack

A DoS attack is a type of network attack where an attacker sends a large volume of traffic or requests to the target server/system, thus making the network to be congested. This in turn exhaust the system’s resources such as memory, CPU, and network bandwidth. This results in the system becoming unresponsive or slow to respond.

A DDoS attack, on the other hand, work in the same way but attackers typically use botnet, a network of infected computers or devices which are controlled remotely by the attacker, to flood a targeted system with massive amount of traffic.

The aim of these attack is to create an inability for clean traffic to flow, thus preventing legitimate users to have authorized access to the targeted resources. There are several types of DDoS attack such as ping, UDP (User Datagram Protocol) and HTTP flood attack.



*Figure 5: An example of a DDoS attack*

1. Man-in-the-middle

MITM is a type of network attack where an attacker intercepts and modifies the communication, by eavesdropping the keying material, between two sensor nodes who believe that they are communicating directly with each other. As a result, the attacker gains unauthorized access to the network and can eavesdrop on the communication. The attacker can eventually exploit this false sense of security to steal sensitive information being transmitted.

An example of a MITM attack in the context of a smart environment network is provided below:

1. Bob’s smartphone communicates with his smart thermostat to adjust the temperature.
2. The attacker sets up its own wifi hotspot, with the aim to trick Bob into using it. The attacker also sets up a network sniffer to inspect any traffic as it passes through.
3. Once Bob connects to the unsecure wifi hotspot, the attacker is able to intercept the communication between the two devices with them knowing.
4. The attacker can then manipulate the communication by changing the temperature settings, preventing Bob from controlling the thermostat or even obtaining Bob’s login credentials.

MITM attack can be carried out in various forms like ARP spoofing, IP spoofing, SSL stripping, and more.

1. Routing attack

Routing attacks exploit vulnerabilities in the routing protocols used in a network to manipulate the flow of data, leading to unauthorized access, modification, or theft of data. In IoT networks, an attacker may undertake routing attacks in IoT networks in an effort to intercept, reroute, or prevent data packets as they travel across the network. This can be achieved through various methods, such as:

* Routing Table Poisoning

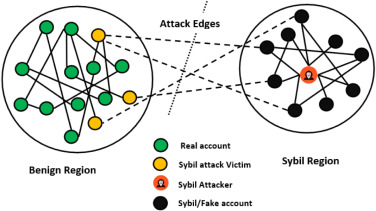
An attacker can corrupt the routing tables of routers in the network to redirect traffic to a malicious destination.

* Wormhole Attack

In this attack, an attacker creates a tunnel between two remote parts of the network to capture or modify data as it passes through.

* Sybil Attack

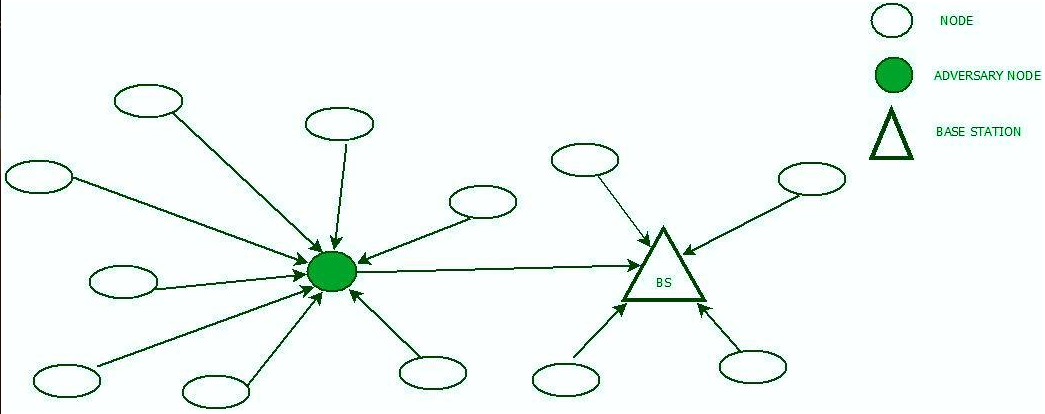
In this type of attack, an attacker creates multiple identities to other nodes in the network so that he can appear in more than one place at a time. As a result, he manipulates the flow of data. For example, in a WSN (Wireless Sensor Network), voting system single node can vote multiple times.



*Figure 6: Sybil attack*

* Sinkhole Attack

An attacker can compromise a node in the network to redirect all traffic towards it, causing a denial of service for the legitimate nodes. The compromised node does this by sending fake routing information to other neighboring nodes that it has the shortest distance path to the base station and guides the traffic from other nodes towards itself. In this way, the malicious node can manipulate the data or drop the packets, thus weakening the security of the network.



*Figure 7: Sinkhole attack*

* + 1. **Firmware Attacks**

Asdasdadasdasdasdasdadas

Asdasdasdasdassdassdasd

Assdasd

1. Backdoors
2. Unencrypted Information
3. Malicious Firmware
   * 1. **Encryption attacks**

Sdasdasdasdasd

Assdasdasd

Aasdasdasd

1. Side channel attack
2. Cryptanalysis-Based Attack
3. Ciphertext Only Attack
4. Known Plaintext Attack
   1. **Existing security features in IoT devices**

Some of the attacks that has been discussed can be mitigated if appropriate security solutions are implemented in those devices. Due to resource-constraint challenges in IoT devices such as low computation power, there is a need to secure those devices with lightweight security features. Some existing and common solutions are discussed below:

**2.4.1 Authentication**

1. **Password**

One of the main problem when authenticating login is that the user makes use of default username and password set by the devices. It is encouraged for users to make use of strong passwords in order to create a robust authentication system.

Advantage:

* Strong passwords are difficult to guess
* Making use of strong passwords makes it difficult to brute-force attack it

Disadvantage:

* Security is only assured at authentication level. An attacker may intercept the communication since it does not encrypt data
* Passwords can be leaked during a data breach

1. **One Time Password**

OTP is a password-based authentication that is valid for only one login session. Once a user has authenticated an OTP, it cannot be reused making the login process secure.

Advantage:

* Resistant against password-cracking attack and data breach
* Password used only once making it difficult for attackers to intercept it

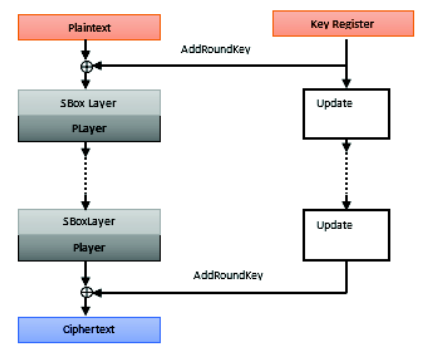
Disadvantage:

* Security is only assured at authentication level. An attacker may intercept the communication since it does not encrypt data
  + 1. **Encryption**

Encryption is important to preserve confidentiality of data. It makes it more difficult for a hacker to decipher a message. However, traditional encryption such as AES which usually makes use of large key size and involve high number of round iterations, making it not suitable to be implemented in resource-constraint device. Some lightweight encryptions are discussed below:

1. **Block ciphers**

Block ciphers are symmetric cryptography algorithm that encrypt a group of plaintext symbols as one block. In resource-constraint devices, lightweight block ciphers which are generally implemented make use of small-key size. Example are PRESENT block cipher- the approved and mostly suitable LWC algorithm for resource-constraint devices- where it makes use of the SPN technique with 31 rounds of key addition, 64-bit cipher text and plaintext.



*Fig 7: Block diagram of PRESENT*

Advantages:

* Information from one plaintext symbol is diffused into several cipher text symbols.

Disadvantages:

* It encrypts data slowly, that is an entire block must be accumulated before encryption/decryption which means higher energy consumption.
* It is not resistant against linear cryptanalysis which is a known plaintext attack

1. **Elliptic Curve Cryptography**

ECC is a public-key cryptography that uses private and public keys to encrypt and decrypt data during transmission. It creates security between key pairs of public cryptography based on mathematical formula of elliptic curves.

Advantages:

* Uses small key size as compared to RSA which makes it suitable for resource-constraint IoT devices
* Generates key faster

Disadvantages:

* Vulnerable to twist-security attack which involve invalid-curve attack, thus exposing the victim’s private key
* Very difficult to implement due to complex mathematical algorithm which is involved when finding appropriate elliptic curves and generating key parameters, thus resulting into a higher consumption power of the device

**2.4.3 Secure Communication Protocol**

There is some secure lightweight protocol such as MQTTS that have been created to be supported in some of the high-end IoT devices. These protocols were designed in order to add an additional layer of security by using TLS (Transport Layer Security) to encrypt data and ensure a secure communication.

Although TLS provides a high security, it requires more computational resources than standard MQTT due to the added overhead of encryption and decryption. As a result, this makes low computing IoT devices unable to support it.

**2.4.4 Firmware Update**

Since IoT devices are deployed in many areas, the manufacturers are encouraged to implement firmware or software updates in order to save the devices from attacks. By updating the firmware, devices can receive the latest security patches, add support for encryption algorithms or removing unnecessary features that can be exploited by hackers.

However, many manufacturers neglect this aspect of security and the devices remain susceptible to known attacks.

## **2.5 Study on Existing Security Threats over Wi-Fi with critical reviews**

The research articles that have been done will be summarized in this part, together with some critical analysis.

**2.3.4 Research Study 1- Security Issues in IoT: A Survey**

A survey of various security methods that address the issues in IoT security are discussed in this paper, while considering the problems facing with IoT devices such as low-performance CPU, low memory, and limited power battery. These issues are mainly focused on authentication, types of attacks, encryption issues, IoT security framework and IoT hardware-based support. Due to the limitations in the software and hardware components, the process of implementing existing security methods in IoT devices are not easy. Hameed et al. brings up the necessity of applying lightweight encryption algorithms to support the IoT devices and prevent the different types of attacks. The algorithm hides the IoT networking paths, making it difficult to attack. (SPE) which creates additional packet to the network is integrated in this algorithm. These packets that are produced are fake and the aim is to hide the real data packets in the network. However, in order to provide a trusted security, the algorithm still need to be improved before it is implement in an IoT system.

**Critical Analysis**

The algorithm under review proposed an idea of being protected against network attacks. However, the network traffic can be learnt by attackers and they can easily differentiate between the fake and real data. To maintain the confidentiality during network transmitting data, encryption mechanism such as WPA3 should be implemented so attacker will not be able to access sensitive data transmitting packets.

**2.3.5 Research Study 2- IoT Security in Wireless Devices**

Due to the rise in the number of IoT wireless devices connected to the internet, the IoT security still remains a concern. Gauliya et al. have proposed the idea of Elliptic Curve Cryptography (ECC) which is a technique of ensuring efficient encryption of data based on curvy algorithm. Moreover, an attempt has been made to prove the efficacity of this cryptographic algorithm to enhance the IoT security. It encrypts data at a 164-bit platform and in order to crack this algorithm, 1024-bit is required. Despite the fact that large key size is to be used to breach the network, attackers can still find a way to enter into the network. As a solution to this problem, this article introduces the idea of implementing ECC at each layer and every node on wireless devices. This will make the standard protocols to generate new encryption decryption code each time ensuring high security of the wireless devices.

**Critical Analysis**

This study discussed how effectively ECC security mechanism encrypts keys using small key size. However, implementing ECC at every layer of a network is challenging due to the complex mathematical algorithm which is involved when finding suitable elliptic curves and generating suitable parameters for the keys. An alternative is to use a security protocol such as SRPP to encrypt keys and it has a better response time in terms of its latency, scalability and packet overhead.

**2.3.6 Research Study 3- Enhanced IoT Wi-Fi protocol standard’s security using Secure Remote Password Protocol**

In IoT based application, Wi-Fi network plays a vital role in creating that network of connected devices. Sometimes, IoT devices are designed with low-security protocol which makes them vulnerable immediate attacks. This research proposes a password protection protocol, SRPP (Secure Remote Password Protocol), which can be implemented Wi-Fi network to prevent brute-force and dictionary attacks. The simulation is done using the GNS3 simulator and performance of the enhanced Wi-Fi security protocol is evaluated with the IoT security model's evaluation metrics such as packet overhead, integrity, network latency, and scalability (Hiba et al., 2022). ECC and TA were the two security mechanism that were selected to compared SRPP mechanism in terms of their response time. The overall simulation results prove that SRPP has better response time when generating encrypted keys and authenticating secret key. Therefore, it is considered as a suitable option of extending Wi-Fi’s module against immediate attacks discussed since it performs very well in terms of integrity, scalable, latency and packet overhead rate than ECC and TA mechanisms.



*Figure 9: SRP protocol against brute force and dictionary attacks*

**Critical Analysis**

This article proposes a security protocol that uses a large number of encrypted prime number group series as cryptographic techniques to make it difficult for attackers to get that exact combination to decrypt the secret key. However, multiple strategies for authenticating keys can be put together for future directions to strengthen Wi-Fi’s security.

References:

<https://www.diva-portal.org/smash/get/diva2:982570/FULLTEXT01.pdf>

<https://sci-hub.se/http://dx.doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00183>

<https://www.particle.io/iot-guides-and-resources/iot-protocols-and-standards/#:~:text=Layers%20of%20the%20IoT%20Protocol,being%20received%20by%20your%20systems.%22>

<https://blogs.oracle.com/javamagazine/post/java-coap-constrained-application-protocol-introduction>

<https://sci-hub.se/10.1109/WiMOB.2019.8923144>

<https://deliverypdf.ssrn.com/delivery.php?ID=172086089029005068116115098008071000059080034086059056072087065074118106074095107027060049049026019123051109007097070114004074111025007016051080118008085083014097126069014049071000117018083112025097096024086080068068095119100122113064095095002119067094&EXT=pdf&INDEX=TRUE>

<https://www.actapress.com/Abstract.aspx?paperId=28660>

<https://sci-hub.se/10.1109/ISCC.2015.7405513>

<https://aegisresearch.eu/wp-content/uploads/2020/02/Secure-Data-Exchange-for-Computationally-Constrained-Devices.pdf>

<https://sci-hub.se/10.1109/ICECA.2019.8822124>

<https://downloads.hindawi.com/journals/scn/2021/5533843.pdf>

<https://sci-hub.ru/10.5120/19547-1280>

<https://thesai.org/Downloads/Volume8No6/Paper_50-Security_Issues_in_the_Internet_of_Things.pdf>

<https://sci-hub.se/10.1109/MWSCAS.2019.8884913>

<https://sci-hub.se/10.1109/ICITCS.2013.6717816>

<https://sci-hub.se/10.1109/JIOT.2020.3026493>

<https://www.murraystate.edu/academics/CollegesDepartments/CollegeOfScienceEngineeringandTechnology/CollegeOfSciencePrograms/instituteOfEngineering/Programs/cnm/media/IoTSecuritypaper-editedv2.pdf>

<https://sci-hub.se/10.1109/ACCESS.2021.3052867>

<https://sci-hub.se/10.1109/JIOT.2019.2935189>

<https://www.sciencedirect.com/science/article/pii/S2542660522000592>

<https://www.kellton.com/kellton-tech-blog/iot-device-security-7-ways-secure-your-iot-devices>

<https://www.actapress.com/Abstract.aspx?paperId=28660>

<https://sci-hub.ru/10.1109/JIOT.2017.2767291>

<https://sci-hub.se/10.1109/ISCC.2015.7405513>