Development of A Lightweight IoT Security System

Faisal Ahmed Abdullah Assaig
ECE Department
International Islamic Univ. Malaysia
Kuala Lumpur, Malaysia
faisalahmed634@gmail.com

Othman Omran Khalifa
ECE Department
International Islamic Univ. Malaysia
Kuala Lumpur, Malaysia
khalifa@iium.edu.my

Teddy Surya Gunawan
ECE Department
International Islamic Univ. Malaysia
Kuala Lumpur, Malaysia
tsgunawan@iium.edu.my

Asmaa H. Halbouni
ECE Department
International Islamic Univ. Malaysia
Kuala Lumpur, Malaysia y
asmaa.halbouni@hotmail.com

Eki Ahmad Zaki Hamidi
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
akiahmadzaki@uinsgd.ac.id

Nanang Ismail
Department of Electrical Engineering
UIN Sunan Gunung Djati
Bandung, Indonesia
nanang.is@uinsgd.ac.id

***Abstract*—Especially for constrained node devices, the risk of security and privacy increases as the number of Internet of Things (IoT) devices increases. From smart homes to smart cities, IoT is ubiquitous, indicating that most devices will be connected to the internet soon. This exacerbates the problem of securing IoT devices. Where our privacy is at risk are IoT devices with inadequate security. These devices transmit sensitive and private data. To construct well-secured IoT devices, we must first overcome IoT device issues such as low computation power, small data storage, and low power consumption. This demonstrates the need for IoT device security systems to be lightweight. However, there is currently no adequate security system for IoT devices with limited functionality. Consequently, the objectives of this paper are to design a secure IoT system and to analyze the overall system's power consumption and latency. The lightweight security system was able to secure MQTT messages with a latency of 0.3s and power consumption of 1.683mJ, according to the obtained results. Therefore, the success of the paper will enable IoT devices with limited bandwidth to transmit data securely.**

Keywords—IoT, security, privacy, power consumption, latency.

I. INTRODUCTION

Currently, online IoT applications are expanding exponentially. Consequently, all devices will be linked to one another and the internet. Managing IoT issues, which will inevitably arise, would be the disadvantage of this development. The potential dangers and consequences to the security or privacy of things or individuals have increased dramatically.

As a result of rapid advancements in mobile communication, Wireless Sensor Networks (WSN), Radio Frequency Identification (RFID), and cloud computing, IoT devices are now able to communicate to accomplish common tasks. IoT connects the private, commercial, manufacturing, and public sectors. It has multiple applications and uses in transportation, healthcare, and development. IoT is a contemporary term for the Internet revolution, and it will alter our daily lives. It enables intelligent machines to perform our activities around us [1]. IoT consists of a network interface, such as Wi-Fi, Bluetooth or ZigBee, sensors, and actuators [2]. Future online IoT applications will grow exponentially, and all devices will be connected to the internet and each other. This estimate reached 30 billion in 2016, and it is projected to reach 60 billion by 2022 [3].

As the number of IoT devices rapidly increases, IoT-related problems will inevitably arise. There is an exponential increase in the number of IoT devices. As a result, the potential dangers and repercussions to the security or privacy of things or individuals have increased dramatically. In designing IoT devices, security-related research fields present engineers with numerous challenges. Unfortunately, these security requirements are not yet widely recognized. Therefore, it is necessary to study security [4]. Most IoT devices collect personal or sensitive information, making IoT security crucial. These sensitive data could be an open invitation for attackers to steal and exploit them in a variety of ways [5].

The current security system is very effective for devices with high processing power. When the current security system is applied to devices with low processing power or low energy consumption, however, it results in high latency and enormous energy consumption. There is a lack of standardization and support for security systems that operate efficiently on these devices, i.e. IoT devices with low processing power. Therefore, the security system must support these constrained devices. In this paper, a lightweight security system is developed for these types of devices, where it secures transmitted data with low latency and low power consumption. These are the two most important considerations that must be made when designing an IoT platform.

II. IOT SECURITY CHALLENGES

The advantage of the Internet of Things is that it can communicate with and collect data and information from virtually everything and anything without human intervention. However, these benefits are accompanied by threats and difficulties that may affect a variety of aspects of our daily lives. Privacy and security are one of the most significant challenges of the Internet of Things [6]. This section enumerated significant obstacles to constructing a secure IoT system.

A. Scalability

Every day, new IoT devices are connected to IoT networks, bringing with them addressing, naming conventions, data management, authentication, and service management.

B. Heterogeneity

The vast number of interconnected IoT devices varies in complexity, capabilities, and technical connections. IoT devices transmit data in a variety of formats and sizes, including text, audio, and video. Therefore, protocols must consider these parameters, necessitating a cryptographic system. The vast number of interconnected IoT devices varies in complexity, capabilities, and technical connections. IoT

978-1-6654-5122-2/22/$31.00 ©2022 IEEE

devices transmit data in a variety of formats and sizes, including text, audio, and video. Therefore, protocols must consider these parameters, necessitating a cryptographic system.

C. Firmware Updates

The majority of IoT devices lack support for live updates. Since the IoT devices are connected to the internet, this unsecured device creates security risks.

D. Power Consumption

Security concerns are a major concern in the design and development of IoT. However, numerous IoT devices operate with low power consumption as a result of the low memory capacity, low processing capability, and high power consumption. The IoT devices will fall victim to cryptographic processes if this measure fails.

E. Connectivity Challenges

Recent research indicates that by 2022, there will be sixty billion IoT devices, which necessitates the identification, localization, authentication, and authorization of each IoT device to control and manage such a vast number of devices.

The first challenge we face is identifying the IoT device in the network. The foundation of IoT is to find a proper and scalable identification method. This Identification methodology can define the IoT device's location and uniqueness. The reflection of a host's fully qualified domain name (FDNS) in naming policy, as well as the provision of address mapping via DNS resolution. The difficulty is ensuring the integrity of object records used in naming architecture. Although DNS provides name translation, it is not a secure naming system, as demonstrated by DNS cache position and man-in-the-middle attacks. Therefore, a new name service that is compatible with IoT architecture is required.

Every IoT system requires authentication depending on the type of IoT network, the computational power of the IoT device, and the sensitivity of the data in the IoT network. There are numerous authentication methods available, such as ID/password, public-key cryptosystem, and pre-shared secrets. For authorization, database-based or crypto-based access control may be used. Due to the complexity and computational power of IoT devices and networks, however, traditional authentication and authorization methods may not be applicable.

F. Software Vulnerability

The developer may produce programming errors during the software development phase; these errors may lead to security software vulnerabilities. An attacker can install backdoors in vulnerable IoT devices, subsequently increasing the number of backdoor security breaches.

G. Computational Power and Memory

A large number of IoT devices have low power consumption, which translates to low computational power, or they run on microcontrollers with limited processing power, such as the ESP 12E with 4Mb RAM. Additional security features include public-key cryptography, Symmetric-key cryptography, Transport Layer Security (TLS), and others. To secure the integrity and authentication of IoT systems, however, it is necessary to have cryptography algorithms and security protocols that consume less computational power, which remains a challenge for IoT security [7].

III. Proposed Lightweight IoT Security System

A. IoT Platform

This IoT network's security system consists of all techniques and technologies used to encrypt and authenticate data between IoT devices and servers. To ensure complete control over the data, we must build our own IoT platform, as depicted in Fig. 1. The platform will be deployed on a Linux Ubuntu 18.04 server utilizing the cloud infrastructure services of DigitalOcean to manage server requests, packet analysis, and more.

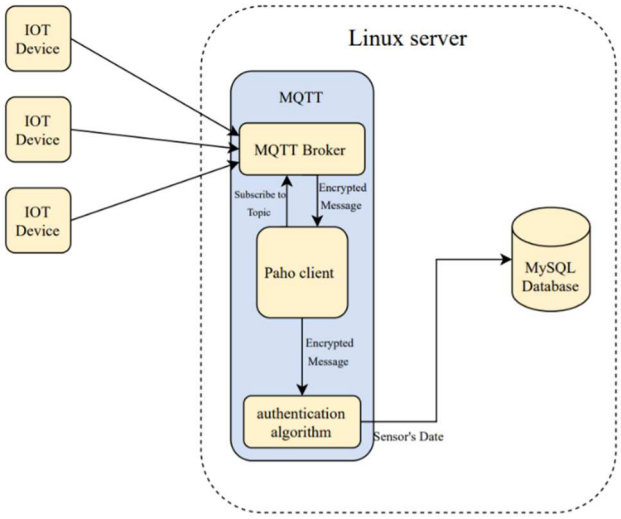

Fig. 1. IoT Platform

The IoT system initializes itself initially. IoT device authenticates to the MQTT Broker (IoT device logs in credentials to MQTT Broker with specific port), after the IoT device authenticates to the MQTT broker (First layer of authentication), the IoT device sends an MQTT message to a specific port and topic where the payload of the message contains the encrypted message which is sensor's data and data of message authentication or information of message authentication (second layer of authentication), the received message is decrypted (third layer of authentication), and The message authentication data that has been verified by the authentication algorithm. After confirming the message's authentication, integrity, and confidentiality, the IoT device's data is stored in a database.

An IoT device, such as ESP8266, began collecting sensor data for transmission. The IoT logs in to the MQTT broker using a username and password as the initial layer of security (Mosquito broker). The IoT device is then authorized to connect to the broker and subscribe to the topic if the password and username are valid. The IoT device then computes the message authentication code, concatenates the message with the MAC value, encrypts the message using the ChaCha20 algorithm [8], and publishes the ciphertext to the broker.

On the server-side, the MQTT broker receives the encrypted text over port 1883, and the server-client subscribes to a topic to receive the encrypted text. Enciphered text is decrypted using the ChaCha20 algorithm. Comparing MAC values to validate the message is the next step. If the values of the MACs are identical, the message is saved to the database.

ChaCha20 uses a 20-round 256-bit key. ChaCha converts sixteen 32-bit input words into sixteen 32-bit output words.

Conventionally, eight of the input words are 256-bit keys. ChaCha20 was chosen for this paper because it requires less computational power than AES. ChaCha20 uses substitution and confusion to generate the ciphered text, whereas AES relies more on arithmetic, which requires more computational power on the microcontroller.

B. Lightweight Security System

The lightweight security system is founded on the ChaCha20 Algorithm and MAC, with a minor modification to the MAC algorithm. To reduce the risk of keys changing between IoT device and server and also to reduce computation power,

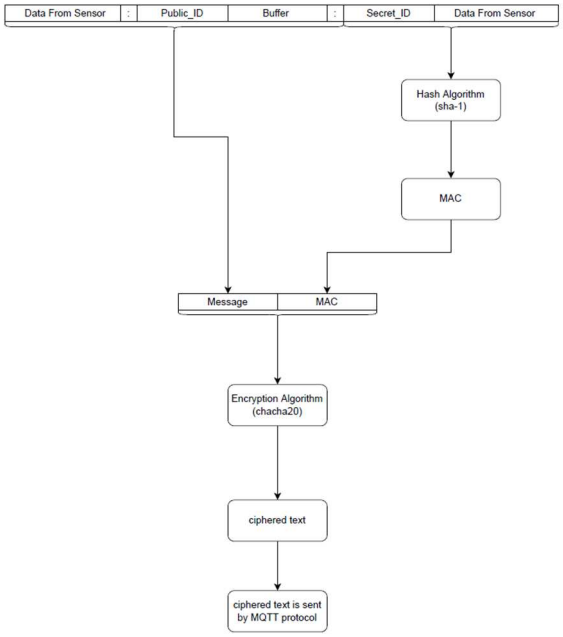

Fig. 2. Lightweight Security System on the IoT Device

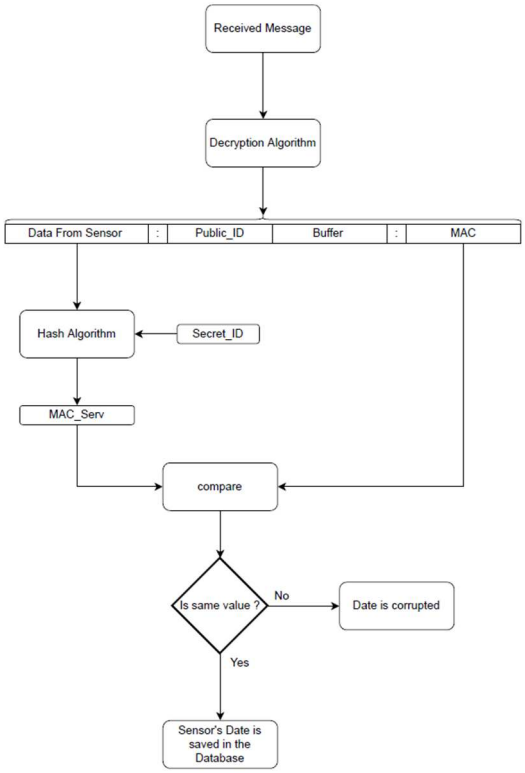

Fig. 3. Lightweight Security System on the Server

Secret keys are stored by default in IoT devices and servers, eliminating the need for keys to be changed between them. Therefore, the secret key is replaced with `secret_ID` in the MAC algorithm. This modification does not affect the security quality of the MAC because the `secret_ID` serves the same function as the secret key in the MAC algorithm, which is to have a secret value that is only known by the IoT device and the Server. The minimum size of a sent message is 64 bytes to add confusion for greater security. Fig. 2 depicts the security system on the IoT device side, while Fig. 3 depicts the server-side. This python script is a client-server and authentication algorithm. The python script subscribes to MQTT Broker and listens to port 1883 and a particular topic.

IV. Results and Discussion

A. IoT Platform Evaluation

The objective of the project is to ensure that the functionality of the system is operating as intended. The system must transmit encrypted messages with MAC values that are validated by the server. In this experiment, the sensor data is transmitted to simulate a real-world scenario in which an IoT device collects and transmits data with the value “45”. The client connects to the broker using the /test topic and port 1883.

In this experiment, important data is printed using Arduino's serial monitor. ESP8266 is first connected to the internet via WiFi, as shown in Fig. 4. The ESP8266 begins by generating the MAC value secret ID = "a733d*#/&!FEazx" and the sensor data value is 45, which is concatenated.

Fig. 4. IoT Device Client Published Data

On the server-side, the Linux server ran the sub.py file to subscribe to the MQTT broker and receive the ESP8266 message. The sub.py file is a Python script used for MQTT client-server and authentication. The client is connected to the broker and is listening for the topic "/test." When the ESP8266 sends a message, the server will receive it as depicted in Figure 5. The data is validated, the MAC serv and the MAC of the IoT device are identical, and the MAC values are both hexadecimal and decimal.

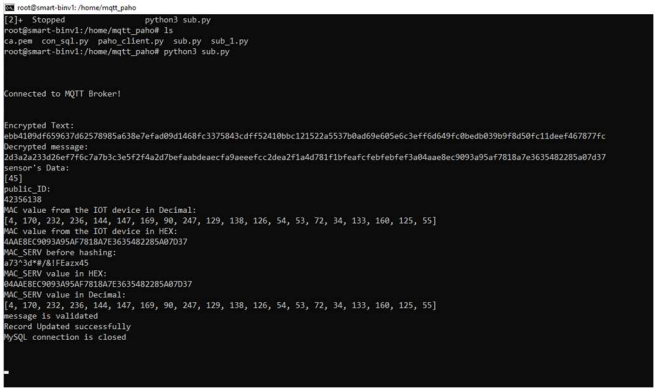

Fig. 5. Server Received the Message

B. *Security Audit of The Proposed Lightweight Security*

There are two types of attacks, including passive and active attacks.

1) *Passive Attack*

In passive attacks, the adversary attempts to read and analyze transmitted data. To simulate this type of attack, the tcpdump packet sniffer or package analysis tool is utilized. This instrument has also been used to capture TCP/IP packets transmitted or received over a network. After installing the tool on the Linux server, this tool captures and stores all TCP/IP packets in a file.

To illustrate the contrast between unsecured and secured systems and the effect of the passive attack on the IoT system, MQTT messages are first sent to the MQTT broker for an unsecured IoT system. ESP8266 transmits an MQTT message with the value 45 over port 1883. This data represents a sensor value. The server-side packet sniffer (tcpdump) captures all port 1883 traffic to simulate passive attacks. The captured packets are saved to a file before being transferred from the server to the local machine (computer) for analysis using the Wireshark tool. For packet analysis, Wireshark provides a graphical user interface and additional networking tools.

First, the captured packets are contained in the MQTT1.pcap file. The subsequent step is to copy this file from the remote computer (Linux) to the computer (local machine). After analyzing the captured packets, it has been determined that the data is exposed and readily accessible by the packet analysis tool. It is evident from Fig. 6 that the value 45 is exposed. This demonstrates that the system is susceptible to passive attacks, in which an adversary can easily read data by capturing network traffic.

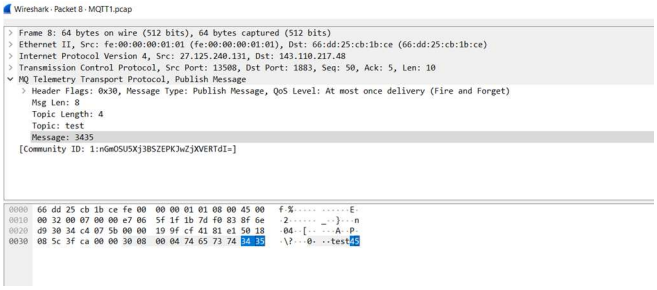

Fig. 6. Exposed Sensor Data Value of 45

2) *Active Attack*

In this type of attack, the adversary not only attempts to read the network's transmitted data but also attempts to alter it. This demonstrates that the value of the transmitted data may be altered by the adversary. Consequently, the recipient receives invalid data. An attacker intercepts the data being transferred between an IoT device and a server in this scenario. In this attack, the adversary attempts to impersonate the sender by convincing the client that it is still in communication with the server and vice versa. The name for this type of attack is Man in the Middle Attack. Consequently, the lightweight security system employs a message authentication code (MAC). As the hash value cannot be reversed or decrypted, this prevents the Man in the Middle Attack. A lightweight security system verifies the message on the server by validating the MAC value, where MAC is a secret identifier concatenated with the message and then hashed.

In this experiment, the validity of messages sent using lightweight security will be examined. Because the secret ID is only known by the IoT device and server, the Man in the Middle Attack is unable to determine the secret ID. However, the adversary attempted to guess the MAC value. In the experiment, the secret modifications simulating a Man in the Middle Attack could be implemented.

As shown in Fig. 7, the MAC of the incoming message is different from the MAC generated in the server. Therefore, the message is not validated, and the authentication algorithm succussed in detecting the invalid message.

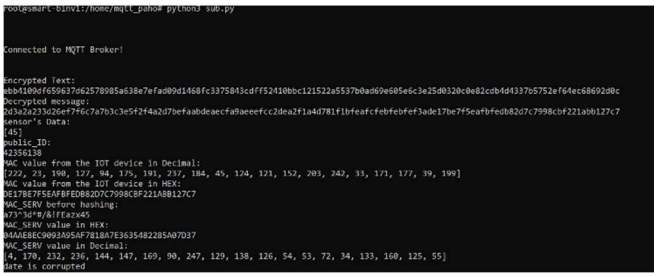

Fig. 7. Server-Side Validation of the Received Data

C. *Latency and Energy Consumption Evaluation*

Low latency is a crucial component of IoT systems. Moreover, additional circumstances require real-time data. To design a security system, latency is a crucial factor that must be taken into account. Low consumption is also an important aspect of IoT systems, as many IoT devices operate on batteries. In this experiment, the time required to transmit a message containing simulated sensor values was measured. In addition, the energy consumption of ESP8266 during message transmission was measured.

TABLE I. COMPARISON BETWEEN LIGHTWEIGHT SECURITY SYSTEM AND CONVENTIONAL SSL/TLS

Attempt No.	Lightweight security system		SSL/TLS	
	Latency (second)	Energy (J)	Latency (second)	Energy (J)
1	0.3	1.683m	4.6	0.151
2	0.28	1.571m	4.7	0.155
3	0.31	1.739m	4.5	0.149
4	0.27	1.547m	4.8	0.158
5	0.32	1.795m	4.9	0.162
6	0.29	1.627m	4.6	0.152
7	0.26	1.459m	4.4	0.145
8	0.33	1.851m	4.3	0.142
9	0.34	1.917m	4.8	0.158
10	0.35	1.964m	4.5	0.149
Average	***0.305***	***1.715m***	***4.61***	***0.152***

The comparison between the proposed lightweight security system and SSL/TLS is presented in Table I. Compared to conventional IoT systems with HTTP as the protocol and SSL/TLS as the security system over the protocol, the lightweight security system is observed to have lower latency and energy consumption.

V. CONCLUSIONS AND FUTURE WORKS

This paper focuses on securing the application-layer connection between IoT devices and servers. In addition, the security system has low latency and energy usage. At the beginning of this project, the web server will be configured and developed to run MQTT for the lightweight security

system and HTTP for testing and comparison. Developing the MQTT client or IoT device and the server to communicate with the MQTT broker after designing the lightweight security scheme. The testing results indicate that the system protects data from passive and active application-layer attacks. Comparing the system's lightweight security to HTTP-based and SSL/TLS-based conventional IoT systems. The results demonstrate a low latency of 0.3s compared to the conventional SSL/TLS-based security system's latency of approximately 4.6s. Additionally, the lightweight security system consumes less energy than the conventional security system employing SSL/TLS. Future work will include increasing the message size, which is currently limited to 38 bytes, and developing a mechanism for exchanging cryptography algorithm keys between IoT and the server.

REFERENCES

[1] A. Assiri and H. Almagwashi, "IoT security and privacy issues," in 2018 1st International Conference on Computer Applications & Information Security (ICCAIS), 2018: IEEE, pp. 1-5.

[2] P. Urien, "Blockchain IoT (BIoT): A new direction for solving Internet of Things security and trust issues," in 2018 3rd Cloudification of the Internet of Things (CIoT), 2018: IEEE, pp. 1-4.

[3] R. Gurunath, M. Agarwal, A. Nandi, and D. Samanta, "An overview: security issue in IoT network," in 2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on, 2018: IEEE, pp. 104-107.

[4] B. S. Krishna and T. Gnanasekaran, "A systematic study of security issues in Internet-of-Things (IoT)," in 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2017: IEEE, pp. 107-111.

[5] A. Hameed and A. Alomary, "Security issues in IoT: a survey," in 2019 International conference on innovation and intelligence for informatics, computing, and technologies (3ICT), 2019: IEEE, pp. 1-5.

[6] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," IEEE Access, vol. 7, pp. 82721-82743, 2019.

[7] N. M. Karie, N. M. Sahri, and P. Haskell-Dowland, "IoT threat detection advances, challenges and future directions," in 2020 workshop on emerging technologies for security in IoT (ETSecIoT), 2020: IEEE, pp. 22-29.

[8] A. Langley, W. Chang, N. Mavrogiannopoulos, J. Strombergson, and S. Josefsson, "ChaCha20-Poly1305 cipher suites for transport layer security (TLS)," 2070-1721, 2016.