UNIVERSITI TEKNOLOGI MARA

JIIICaS 2024
JOHOR INTERNATIONAL INNOVATION, INVENTION COMPETITION AND SYMPOSIUM

PROCEEDINGS OF JOHOR INTERNATIONAL INNOVATION INVENTION COMPETITION AND SYMPOSIUM 2024 (JIIICaS 2024)

"Flourish and Nurturing Sustainable Innovation for a Prosperous Nation"

JIICOS 2024
JOHOR INTERNATIONAL INNOVATION, INVENTION COMPETITION AND SYMPOSIUM

Editorial Board

e ISBN: 978-967-0033-25-9

Published in Malaysia by
Universiti Teknologi MARA Cawangan Johor
Kampus Pasir Gudang
81750 Masai

Preface

In the name of Allah, the Almighty who gives us the enlightenment, the truth, the knowledge and with regards to Prophet Muhammad (peace be upon him) for guiding us to the straight path. We thank to Allah for giving us guidance and strength to write this e-book.

This e-book compiles the extended abstracts that submitted to Johor International Innovation Invention Competition and Symposium 2024 (JIIICaS2024), where JIIICaS2024 is a virtual platform for all creative minds to share and present their invention and innovation. Each abstract gives a brief background on the innovation or project.

We hope that this e-book will help the readers to get to know the innovation done by the students and get some ideas to develop future innovation products.

Foreword Rector

Assalamualaikum warahmatullahi Wabarakatuh, Salam Sejahtera, Salam Malaysia MADANI and Salam UiTM Dihatiku.

In the name of Allah, the Most Gracious, the Most Merciful.

It is a great honor to welcome you to the Johor International Innovation, Invention, Competition, and Symposium 2024 (JIIICaS 2024). This event connects various disciplines, focusing on education and engaging educators, students, researchers, and innovators from all walks of life.

Innovation is not just about ideas; it demands perseverance, creativity, and determination to turn those ideas into reality. The remarkable projects showcased today highlight the dedication and spirit of all participants. Initiatives like this not only explore new technologies but also cultivate skills and leadership among our youth. At Universiti Teknologi MARA (UiTM) Johor Branch, we are fully committed to fostering a dynamic culture of innovation, promoting the commercialization of new products, and encouraging meaningful collaborations with industry and society.

As we celebrate this event, I would like to extend my heartfelt gratitude to all sponsors, judges, the College of Computing, Informatics and Mathematics, UiTM Pasir Gudang Campus as the event organizer, as well as to the researchers and participants for their hard work in making this event a success. Let us continue striving for innovation and excellence. May the ideas presented today inspire us and lay the groundwork for future achievements.

Thank you.

Associate Professor Dr. Saunah Zainon
Rector
Universiti Teknologi MARA (UiTM)
Johor Branch

(A-ST140) IMPLEMENTATION OF TRANSPORT LAYER SECURITY WITH AES ALGORITHM IN IOT NETWORKS USING MQTT PROTOCOL

Felix Gary[1], Hamid Azwar[1], Muhammad Diono[1], Dini Aulia Putri[1], Alfitrah Putra Buana[1], Muhammad Fathurrahman Alfadli [1]

[1]Jl. Umban Sari No. 1, Pekanbaru, Riau, Indonesia

Corresponding author: felixgary766@gmail.com

ABSTRACT

In the modern world of IoT (Internet of Things), data security is crucial as all technologies are now interconnected. While computing software and hardware have improved security, IoT devices still lack adequate protection. To address this issue, the researchers propose using Transport Layer Security (TLS) with the AES algorithm and the MQTT (Message Queuing Telemetry Transport) protocol to secure IoT devices. The goal is to prevent data theft by encrypting transmitted data, making it unreadable to attackers.

To test the impact of TLS implementation, the researchers evaluated RAM (Random Access Memory) usage, CPU (Central Processing Unit) utilization, and communication delay on the ESP32 microcontroller. With TLS enabled, the average remaining RAM was 180,256 bytes out of 520 KB, CPU usage was 0.53% out of 240 MHz, and the delay was 0.475 seconds. Without TLS, the average remaining RAM was 224,825 bytes out of 520 KB, CPU usage was 0.2% out of 240 MHz, and the delay was 0.248 seconds. These results indicate that the performance impact of TLS on the ESP32 is relatively small, suggesting that enhanced security measures can be implemented without significantly affecting system resources or performance.

Keywords: CPU, delay, Message Queuing Telemetry Transport, Transport Layer Security, RAM

1.0 INTRODUCTION

Internet of Things (IoT) is a concept where objects are embedded with technologies like sensors and software to facilitate communication, control, connectivity, and data interaction with other devices via the internet. Communication between devices requires protocols to transmit information and data securely. One of the protocols used in IoT devices is MQTT (Message Queuing Telemetry Transport). Generally, the MQTT protocol offers only basic authentication, which is makes the device vulnerable to attack. With this basic security, a network can be easily attacked, resulting in data theft or other attacks such as denial of service, man in the middle attacks, and sniffing.

The 2022 Digital Defense Report by Microsoft highlights that while security in software and hardware computing has improved recently, IoT devices' security has lagged, resulting in numerous attack incidents. To prevent these threats, encryption is essential. Encryption aims to maintain the security of data in the form of information or

messages to prevent attackers from reading the contents of the information or messages.

Previous research has identified the security shortcomings of the MQTT protocol in IoT devices, which by default lacks robust security features, making them susceptible to attacks. Researchers have suggested solutions like implementing TLS/SSL to enhance the security of the MQTT protocol. One study implemented TLS/SSL with the RSA algorithm to prevent sniffing attacks on IoT devices. Another study applied the AES algorithm, suitable for devices with limited resources, successfully protecting databases from external attacks with minimal resource usage. Additionally, a comparative study found that the AES algorithm is more efficient and faster than RSA for encryption security.

This study proposes enhancing the security of the MQTT protocol using the AES algorithm, which is more efficient and can improve the integrity of IoT devices.

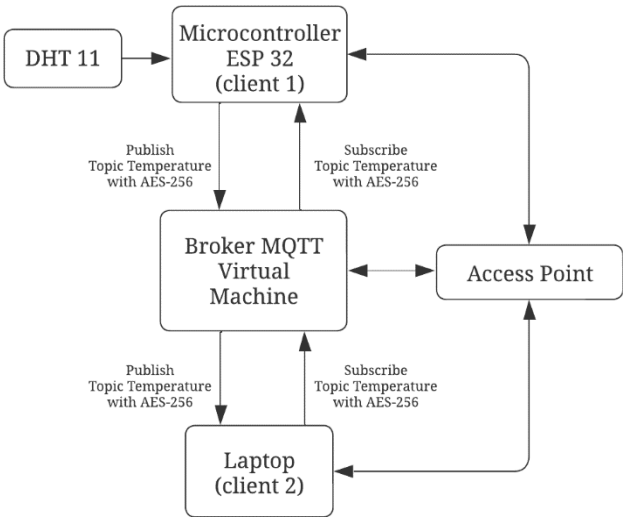
2.0 OBJECTIVE

The aim of this research is to secure the communication of IoT devices based on the MQTT protocol in terms of data integrity and to enhance the security of IoT devices during the data transmission process. The benefits of this research include providing information about security vulnerabilities in the MQTT protocol and knowledge about the weaknesses of IoT devices that use the MQTT protocol.

3.0 METHODOLOGY

3.1 Device System

This diagram is a design of the device that will be developed in this study.

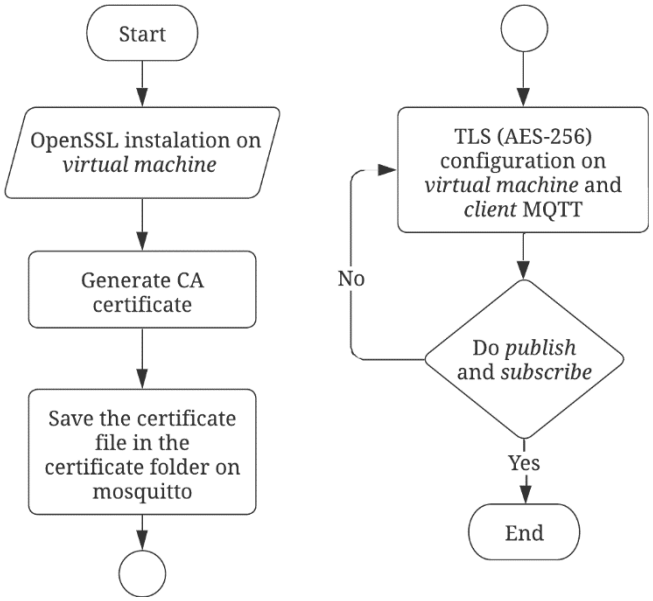


In the diagram above, client 1 acts as a publisher, which is an ESP32, and client 2 acts as a subscriber, which is a laptop. The broker is a virtual machine where the MQTT protocol is installed using Mosquitto. The broker's role is to receive and forward messages between the subscriber and publisher based on the designated topic. When the device operates, client 2 sends a subscribe message with a specific topic to the broker. The broker then forwards this subscription to its destination, client 1, which is the ESP32 with a DHT11 temperature sensor. Client 2 receives the message and

sends a response message containing a publish with the same topic to the broker, and the broker forwards the message to client 1.

3.2 Security System Design

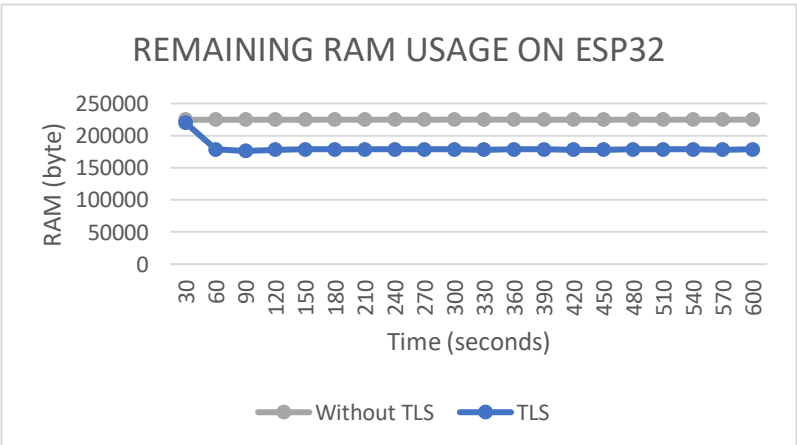
Communication between devices in the designed IoT system uses the MQTT protocol, which by default does not have security measures for its communication. Therefore, it is necessary to implement security to protect the communication of IoT devices. To achieve this, a certificate is required for the TLS handshake process. This certificate will be generated using OpenSSL and will contain an AES-256 key algorithm, which will be used for encryption and decryption processes. The certificate will be stored in a folder on the virtual machine that acts as the broker. Security will be applied to the communication between the broker and the client.



4.0 RESULTS

4.1 ESP32 RAM Usage

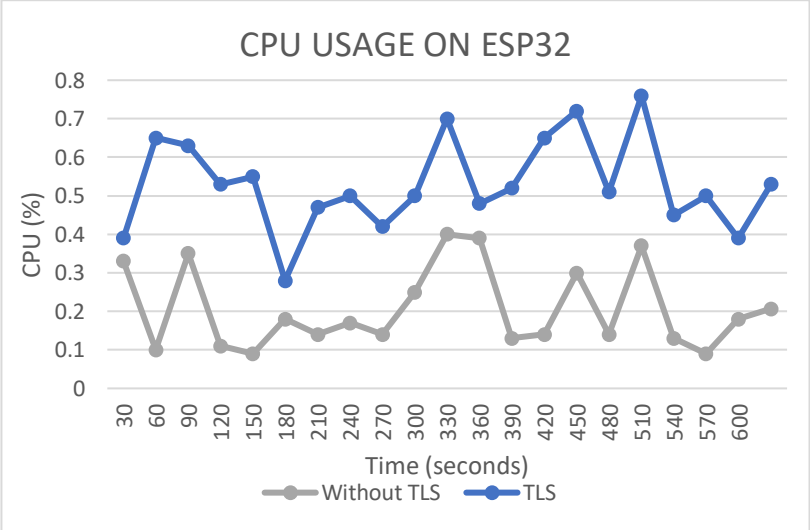
Performance testing on IoT devices is carried out by comparing the differences in RAM usage and delay before and after implementing TLS (Transport Layer Security) on the ESP32 microcontroller.



The performance test results of the IoT device based on RAM usage were tested for 10 minutes with data collection every 30 seconds. In the graph of Figure 4.33, the gray line indicates the remaining RAM usage on the ESP32 without implementing TLS with the AES algorithm, while the blue line shows the remaining RAM usage on the ESP32 with TLS implementation using the AES algorithm. In the first 30 seconds, RAM usage with the AES algorithm (219,884 bytes) and without the AES algorithm (224,800 bytes) is almost the same due to the initial communication process still performing validation between the client and broker, which does not use many resources from the ESP32 before proceeding with data transmission communication. Overall, the average remaining RAM usage on the ESP32 is around 180,256 bytes with TLS implementation and 224,825 bytes without TLS implementation.

4.2 ESP32 CPU Perfomance

Performance testing on IoT devices is conducted by comparing the differences in CPU usage before and after implementing TLS on the ESP32 microcontroller.



The performance test results of the IoT device based on CPU usage were tested for 10 minutes with data collection every 30 seconds. In the graph of Figure 4.34, the gray line indicates CPU usage on the ESP32 without TLS implementation, while the blue line shows CPU usage on the ESP32 with TLS implementation. CPU performance using TLS is higher compared to without TLS, as TLS requires multiple processes to secure data before sending it to the broker. In contrast, without TLS, data is sent directly to the broker without security measures. Overall, the average CPU usage with TLS is 0.53%, while without TLS it is 0.2%.

4.3 Client and Broker Delay Communication

The test measures the delay values produced before and after implementing TLS.

Table 7 : Delay Communication Comparison

	Pakcets	Time Span (s)	Delay (s)
TLS	285	600.381	0.475
Without TLS	145	585.761	0.248

Table 1 shows the number of packets and time span obtained from data collection over 10 minutes before and after implementing AES. The delay measured during AES implementation was 0.475 seconds, whereas without AES it was 0.248 seconds. The data indicates that implementing AES approximately doubles the delay compared to using the system without AES.

5.0 CONCLUSION

Based on the test results, the following conclusions can be made:

1. Communication using the MQTT (Message Queuing Telemetry Transport) protocol has security vulnerabilities that allow attackers to view the data being transmitted during the publish-subscribe communication process.
2. Implementing TLS (Transport Layer Security) with the AES algorithm has been shown to enhance the security of MQTT protocol by providing data integrity through encryption, for both the client and the broker.
3. The performance differences in IoT devices based on MQTT (Message Queuing Telemetry Transport) before and after implementing TLS (Transport Layer Security) are not very significant. This is reflected in the average remaining RAM usage of 180,256 bytes out of 520 KB, CPU usage of 0.53% out of 240 MHz, and a delay of 0.475 seconds with TLS implementation, compared to an average remaining RAM usage of 224,825 bytes out of 520 KB, CPU usage of 0.2% out of 240 MHz, and a delay of 0.248 seconds without TLS implementation.