1. Introduction

Background

IoT (Internet of Things) is a broad range of technology in the form of sensor devices connected to networks and cloud storage, as well as an ecosystem in which communication is connected to many types of devices such as environmental sensors, vehicles, remote controller actuators, smart cities, IIoT (Industrial devices Internet of Things), and health care sensors [2]. Currently, many communication protocols are used on IoT devices such as HTTP (Hypertext Transfer Protocol), CoAP (Constrained Application Protocol), XMPP (Extensible Messaging and Presence Protocol), AMQP (Advanced Message Queuing Protocol), and MQTT (Message Queuing Telemetry Transport) [3] [4].

MQTT (Message Queuing Telemetry Transport) is a messaging protocol that is commonly used by publishers / subscribe on [1] IoT (Internet of Things) devices. Which protocol is already marked OASIS (Organization for the Advancement of Structured information Standards) [2]. The application of MQTT is often used on the Internet of Things because there are publishers, subscribers and brokers running, where the publisher (e.g., sensor) is connected and sends data to the broker (e.g., server) to subscribers (e.g., an application that retrieves data from the broker). MQTT is a lightweight transport protocol in the design of devices and low bandwidth, high latency or unreliable networks, and MQTT acts as minimizing bandwidth networks and providing guarantees in data transmission [3].

Security in MQTT is IoT security as well, which we must pay attention to in terms of data, privacy, integrity, confidentiality, authentication and authorization. For example, where an attack is contained in; Large DDoS (Distributed Denial of Service) attacks, and Mirai malware, which attacks and hijacks IoT devices into bots [4], and in the real world there are many applications, web mail, internet banking that require channel security between several users. For this reason, the use of AKE (authentication and key exchange) protocol can be used by some users to obtain temporary session keys, which is useful for maintaining the security of each user's communication. Until now password only AKE PAKE (Password-Authenticated Key Exchange) is very widely used therefore it is very considered in terms of security, because PAKE security design itself is not easy, therefore the existence of offline dictionary attack on passwords is very vulnerable to attack [5].

All the PAKEs we know of use public key cryptography, the difference within each PAKE lies in how the public key communicates in the exchange. There are various kinds of PAKE that are used such as EKE (Encrypted Key Exchange) where this PAKE class uses Symmetric key cryptography, using a key in the form of a password encrypted in the form of an ephemeral public key, then this process performs a successful match which public key has been decrypted to obtain proof of recognition for sharing passwords. Another PAKE class that sends unencrypted public keys is J-PAKE (Password Authenticated Key Exchange by Juggling). In using this PAKE, a public key is required and the value to be sent is used to exchange shared passwords. Furthermore, SPEKE (Strong Password-Only Authenticated Key Exchange), also scheme is a public key exchange or known as Diffie-Hellman value exchange, in this scheme there is a generator that is used for the public key to be sent from the shared secret, after that only the Diffie-Hellman public value is doing the transfer, which generator is in a secret state, in which in both cases the value transmitted over the unsecured media is an element in a delimited field and not a stuck blob [6]. Therefore, of the various existing PAKE classes, the authors chose AugPAKE (Augmentation password only authentication and key exchange) as a suitable method for MQTT security in various existing attacks such as active attacks, passive attacks, and offline dictionary attacks.

In the application of MQTT with security, OASIS standards MQTT v5 [2] make strong recommendations on the use of SLL/TLS [7] security solution on MQTT. Which this solution provides significant additions in communication and complementarity in validation certificate examination such as CRLs [8] (Certificate Revocation Lists) and OCSP [9] (Online Certificate Status Protocol) to avoid using used certifications. From research [3] there is an MQTT security implementation that implements the AugPAKE protocol as an MQTT security and a security replacement recommended by OASIS [2] [7]. This research [3] it aims to prevent passive attacks, active attacks, and off-line dictionary attacks and MITM (Man in the Middle) attacks, as well as avoid significant communication additions in certification validation. AugMQTT does not require certification validation in the publisher/subscriber process at the broker. In this research [3] The implementation of AugMQTT uses open source MQTT Mosquitto 1.4.9 and aims to measure security efficiency. The area implementing the AugPAKE protocol is found on the Ubuntu 64-bit virtual machine (server) and on the Raspberry Pi2 Model B (client).

In this research, the authors are conduct research where the use of AugPAKE in windows OS and Kali Linux OS as research areas, as well as perform an attack scenario in the form of MITM (Man in The Middle) on MQTT in eclipse mosquitto v.2.1

Topics and Limitations

The formulation of the problem according to the background is how to prevent MiTM attacks on MQTT using the AugPAKE protocol, and analyze the performance of each OS when running the AugPAKE protocol, and in this study there are problem limitations including, the author implements the AugPAKE MQTT protocol on Windows OS and Kali Linux OS, as well as the broker used in the form of Eclipse Mosquitto v.2.1, and implements the MQTT security protocol using AugPAKE and runs Man in the Middle attacks based on sniffing attack.

Purpose

The purpose of this final project is to prove that the use of the AugPAKE protocol can prevent MiTM attacks on MQTT and analyze performance processing time on Windows OS and Kali Linux OS.