Final Project: Securing MQTT Communication in Smart Home IoT Systems

CEN455: CEN455: Fund. of Sec. for Computer & Embedded Systems

Instructor: Dr. Mohammed Fadhul Idris

Student Name & ID: Aliya Haider- 1082079

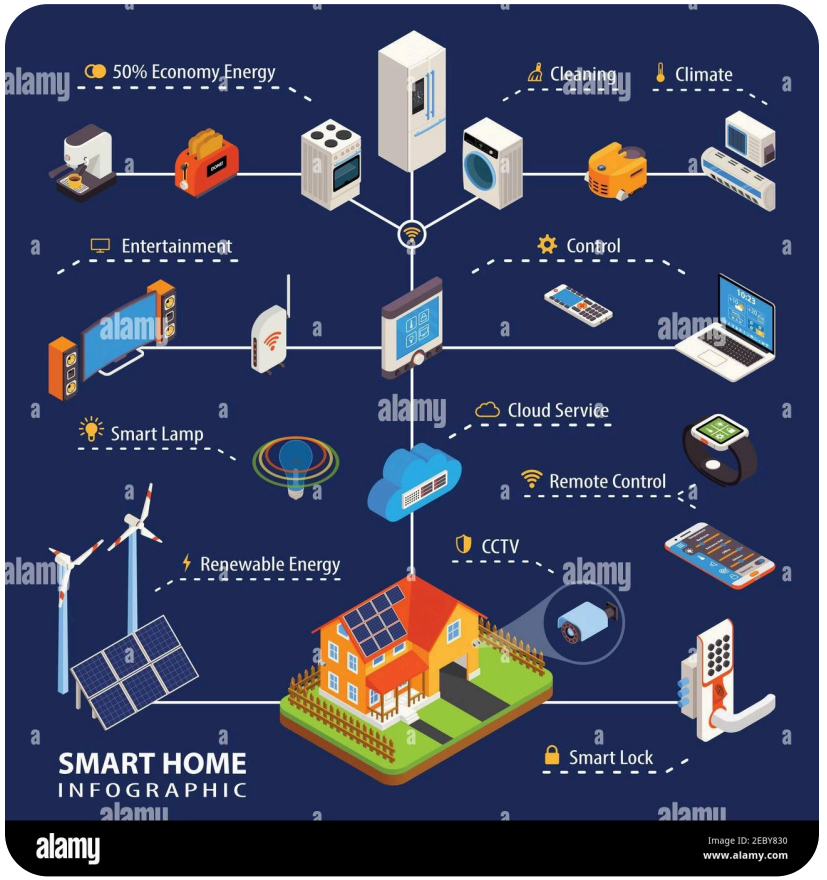
The Peril of Default MQTT: A Smart Home Vulnerability

Unsecured MQTT, a common protocol in IoT, exposes smart home devices to significant risks.

The default MQTT implementation is inherently vulnerable:

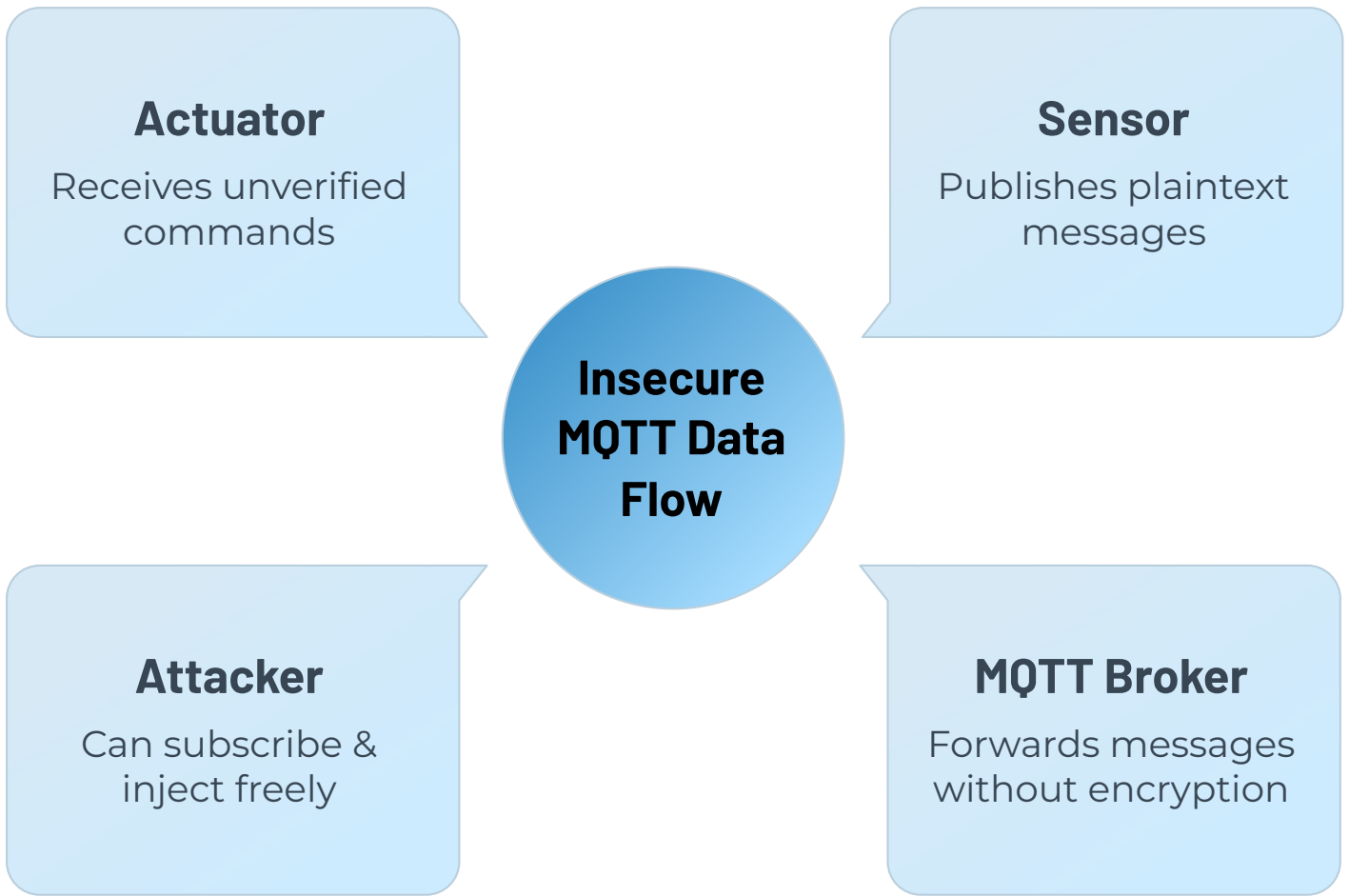
- Data transmitted in plaintext, easily intercepted.
- Lack of authentication allows unauthorised access.
- Susceptible to eavesdropping, spoofing, tampering, and replay attacks.

Confidentiality and integrity are paramount for smart home security.



Insecure System Architecture: An Open Door for Attackers

Without security measures, data flows unprotected, creating numerous attack vectors.



Key characteristics of an insecure MQTT system:

- Messages are transmitted in easily readable plaintext.
- Absence of digital signatures for message authenticity.
- No mechanisms to verify data integrity or prevent replay attacks.
- Attackers can effortlessly subscribe to topics and inject malicious data.

The Critical Flaws: Attacks on Insecure Systems

The absence of cryptographic protection leaves systems wide open to exploitation.

Eavesdropping

Sensitive data, like home automation commands, can be intercepted and read.

Message Injection

Malicious commands can be injected, leading to unauthorised device control.

Replay Attacks

Previously captured valid messages can be retransmitted to trigger unintended actions.

Tampering

Messages can be altered mid-transmission, corrupting data or changing commands.

All these attacks succeed due to the complete lack of cryptography.

A Robust Defence: Secure Architecture Overview

Implementing robust encryption and authentication protocols transforms MQTT into a secure communication channel.

Key protections in the secure system:

- AES-256-GCM encryption ensures data confidentiality and integrity.
- RSA-2048 facilitates secure key exchange between devices.
- DSA signatures verify message authenticity and origin.
- Counter-based replay protection prevents malicious retransmission of messages.
- Broker Access Control Lists (ACLs) enforce strict authorisation policies.

Understanding Secure Message Flow

Each component plays a vital role in encrypting, transmitting, and verifying data integrity.

01

Sensor Encrypts & Signs

The smart home sensor encrypts the payload using AES-256-GCM and adds a DSA signature, nonce, and authentication tag. It also includes an encrypted session key.

02

Broker Forwards Encrypted Payload

The MQTT broker receives and transparently passes the encrypted payload without decryption, maintaining end-to-end security.

03

Actuator Verifies Signature

Upon receiving the payload, the smart home actuator first verifies the DSA signature to confirm message authenticity.

04

Actuator Decrypts & Checks Counter

The actuator then decrypts the payload and checks the counter for replay protection, ensuring the message is fresh and valid.

Attacks on the Secure System: Complete Blockage

With robust cryptographic controls, all previously successful attacks are neutralised.

Eavesdropping

Blocked: Data is encrypted (AES-256-GCM), rendering intercepted information unreadable.

Message Injection

Blocked: DSA signatures and ACLs prevent unauthorised command injection.

Replay Attacks

Blocked: Counter-based protection detects and rejects replayed messages.

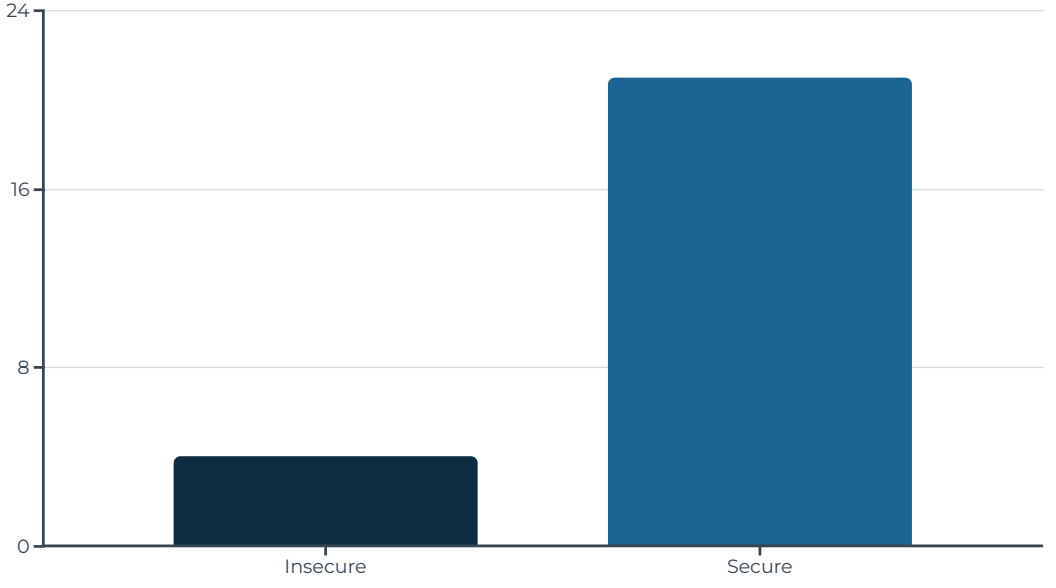
Tampering

Blocked: Message integrity is ensured by AES-256-GCM authentication tags and DSA signatures.

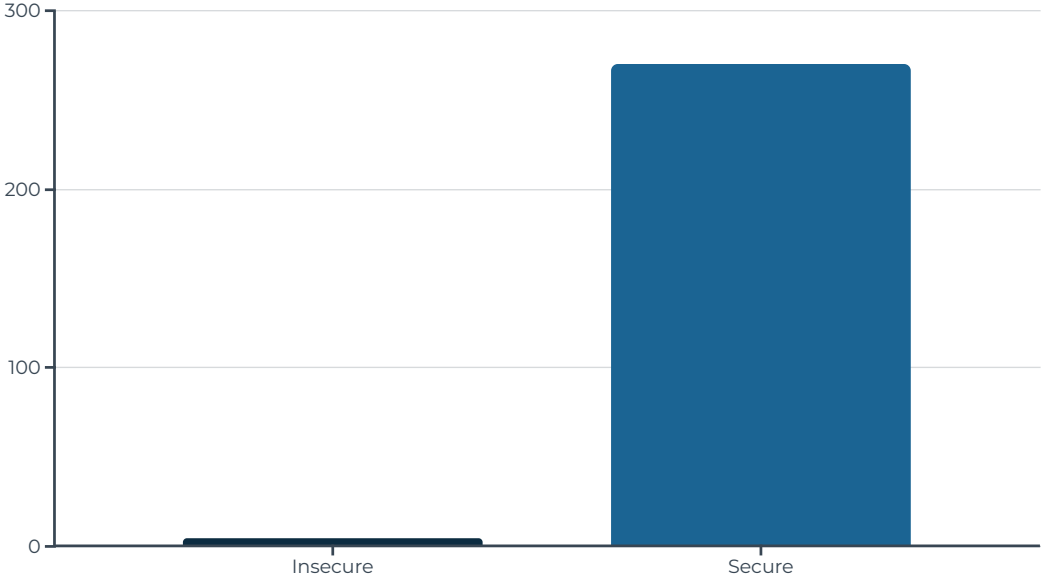
Our secure system successfully rejected over 270 malicious packets, resulting in zero successful attacks.

Quantitative Results: Performance vs. Security

While security adds a minimal overhead, the enhanced protection is invaluable.



Latency Comparison: Secure communication introduces a slight increase in latency (~21 ms) compared to the insecure system (~4 ms). This overhead is well within acceptable limits for most IoT applications, especially in smart homes.



Attack Attempts vs. Blocks: The insecure system blocked zero attacks, while the secure system successfully rejected over 270 malicious attempts. This demonstrates a substantial improvement in overall security posture.

Mitigating Timing Attacks: Constant-Time Operations

Eliminating timing leaks is crucial for cryptographic integrity, preventing side-channel attacks.

Vulnerable Function

The vulnerable function exhibited variable timing (0.767–1.514 µs), potentially leaking cryptographic information based on input data.

Secure Constant-Time Function

Our secure, constant-time function flattened timing (2.143–3.995 µs), effectively eliminating timing leaks and enhancing security.

Implementing constant-time operations is a critical step in building resilient cryptographic systems, ensuring that execution time does not reveal sensitive data.

Conclusion: Security is Paramount for Smart Homes

The imperative for secure IoT communication cannot be overstated.

Insecure MQTT is Unsafe

Default MQTT is fundamentally unsuitable for any real-world IoT deployment, especially in smart homes.

Comprehensive Protection

Our secure system provides robust confidentiality, integrity, authenticity, and replay protection.

Minimal Overhead, Major Gain

The minor performance overhead is a small price for significant security enhancements.

Future-Proof Smart Homes

This secure approach is essential for modern, reliable, and trustworthy smart home ecosystems.