차세대 IoT 보안: 하드웨어 보안모듈 내 ARIA 암호화 - MQTT 와 LwM2M 통합의 비교 분석

이크발 무함마드 [1], 락스모노 아구스 마하르디카 아리 [2], 프라타마 데리 [3], 김호원 [4]
[1] 부산대학교 정보융합공학과 박사과정
[2] 부산대학교 정보융합공학과 석사과정
[3] 부산대학교 정보융합공학과 교수

iqbal@pusan.ac.kr, agusmahardika@pusan.ac.kr, derryprata@gmail.com, howonkim@pusan.ac.kr

Next-Gen IoT Security: ARIA Cryptography within Hardware Secure Modules - A Comparative Analysis of MQTT and LwM2M Integration

Iqbal Muhammad, Laksmono Agus Mahardika Ari, Derry Pratama, Howon kim
Dept. of Computer Science Engineering, Pusan National University

Abstract

This paper investigates the integration of ARIA cryptography within hardware secure modules to bolster IoT security. We present a comparative analysis of two prominent IoT communication protocols, MQTT and LwM2M, augmented with ARIA cryptography. The study evaluates their performance, security, and scalability in practical IoT applications. Our experimental setup comprises FPGA-enabled hardware secure modules interfaced with Raspberry Pi acting as an MQTT and LwM2M client. We utilize the Mosquitto MQTT server and an LwM2M server deployed on AWS IoT. Through rigorous experimentation, we measure various performance metrics, including latency, throughput, and resource utilization. Additionally, security aspects are scrutinized, assessing the resilience of each protocol against common IoT security threats. Our findings highlight the efficacy of ARIA cryptography in bolstering IoT security and reveal insights into the comparative strengths and weaknesses of MQTT and LwM2M protocols. These results contribute to the development of robust and secure IoT systems, paving the way for future research in this domain.

1. Introduction

In the landscape of modern computing, ensuring the integrity and confidentiality of sensitive data has become imperative, particularly in the context of interconnected devices within the Internet of Things (IoT). Hardware Secure Modules (HSMs) emerge as fundamental components in bolstering the security posture of such systems.

At its core, an HSM is a dedicated hardware device or integrated circuit that provides a secure environment for key management and cryptographic operations. These modules are designed to safeguard cryptographic keys and sensitive data and perform cryptographic functions, thereby mitigating the risks associated with software-based implementations vulnerable to attacks such as side-channel attacks and malware exploitation.

2. Theoretical Background

2.1 Hardware Secure Module

Hardware Secure Modules (HSMs) represent a cornerstone in modern cryptographic systems, providing a dedicated hardware platform for secure key management and cryptographic operations. Central to their design is the assurance of confidentiality and integrity for cryptographic keys and sensitive data. HSMs are equipped with robust physical security measures, such as tamper-resistant casings and encryption techniques, to safeguard against unauthorized access or extraction of cryptographic assets. Additionally, HSMs incorporate dedicated hardware components for random number generation, ensuring the unpredictability and entropy required for cryptographic processes.

These modules offer a range of essential functionalities, including secure key storage, cryptographic operations, and secure communication interfaces. By centralizing key management functions within an HSM, organizations can enforce access controls, audit trails, and policy enforcement mechanisms to protect cryptographic assets effectively.

Moreover, HSMs accelerate cryptographic processing tasks, offloading resource-intensive operations from general-purpose computing devices and enhancing the efficiency and scalability of cryptographic operations in diverse computing environments.

In practical applications, HSMs are widely used in key management, digital signatures, authentication, and secure cryptographic processing. Their integration into IoT devices and infrastructure plays a crucial role in fortifying the security of interconnected systems, ensuring compliance with regulatory requirements, and safeguarding the confidentiality, integrity, and availability of sensitive data transmitted within IoT ecosystems.

2.2 MQTT

MQTT (Message Queuing Telemetry Transport) is a lightweight and efficient messaging protocol specifically designed for low-bandwidth, high-latency, or unreliable networks, typical characteristics of IoT deployments. MQTT operates on a publish-subscribe model, facilitating communication between devices and applications in a decoupled manner. Key to its design is the concept of topics, which serve as communication channels to which clients can subscribe or publish messages. MQTT brokers act as intermediaries, responsible for receiving messages published to topics and delivering them to subscribed clients. This architecture enables seamless communication between numerous IoT devices and applications without requiring direct connections between them. Furthermore, MQTT's lightweight nature minimizes network overhead, making it suitable for resource-constrained IoT devices. It supports Quality of Service (QoS) levels to ensure message delivery reliability, with options ranging from at most once to exactly once delivery semantics. Additionally, MQTT supports features such as last-will-and-testament messages, allowing clients to specify a message to be published in case of unexpected disconnection. With its simplicity, scalability, and reliability, MQTT has become a prevalent choice for IoT communication, facilitating the development of robust and scalable IoT ecosystems.

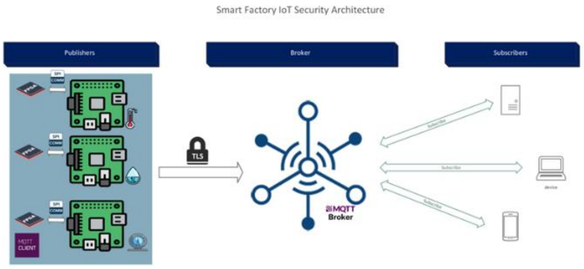


Figure 1. MQTT Stack Diagram

2.3 LwM2M

The Lightweight Machine-to-Machine (LwM2M) protocol emerges as a standardized communication protocol designed specifically for remote device management and monitoring in IoT deployments. Developed by the Open Mobile Alliance (OMA), LwM2M offers a lightweight and efficient solution tailored to the constraints of IoT devices, including limited processing power, memory, and bandwidth. LwM2M operates on a client-server architecture, wherein IoT devices (clients) communicate with remote management platforms (servers) to exchange information and perform management tasks. Central to LwM2M's design is its resource-oriented approach, where device capabilities and data are represented as a hierarchical tree structure known as the Object Model. This model allows for standardized access and manipulation of device resources, simplifying device management tasks such as firmware updates, configuration changes, and sensor data retrieval. LwM2M employs CoAP (Constrained Application Protocol) as its underlying communication protocol, leveraging its low overhead and efficient message exchange mechanisms. Furthermore, LwM2M defines various standard objects and resources, covering common IoT functionalities such as device configuration, connectivity status, and sensor data reporting. With its standardized approach, lightweight design, and support for device management tasks, LwM2M serves as a versatile and scalable protocol for IoT deployments, facilitating seamless communication and management of diverse IoT devices within interconnected ecosystems.

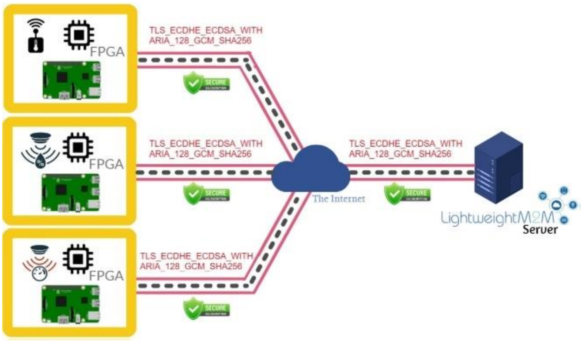


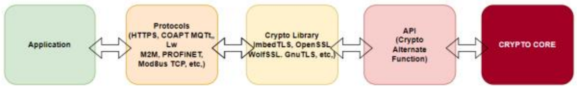
Figure 2. LwM2M Stack Diagram

2.4 Cryptographic Libraries

Cryptographic libraries serve as essential toolkits for implementing cryptographic algorithms and protocols in software applications. These libraries provide a comprehensive set of functions and routines for performing cryptographic operations such as encryption, decryption, digital signatures, and hash functions. They are designed to encapsulate complex cryptographic algorithms into easy-to-use interfaces, enabling developers to integrate strong security measures into their applications without needing in-depth knowledge of cryptographic principles. Cryptographic libraries typically support a wide range of algorithms, including symmetric and asymmetric ciphers, hash functions, and key exchange protocols, allowing developers to select algorithms based on their security requirements and performance considerations. Moreover, cryptographic libraries often undergo rigorous testing and validation to ensure compliance with industry standards and best practices, contributing to their reliability and trustworthiness in securing sensitive data and communications.

3. Proposed Design

3.1 Design Overview


Application
Protocols (HTTPS, COAPT MQTt,, Lw M2M, PROFINET, ModBus TCP, etc,)
Crypto Library (mbedTLS, OpenSSL, WolfSSL, GnuTLS, etc,)
API (Crypto Alternate Function)
CRYPTO CORE

\

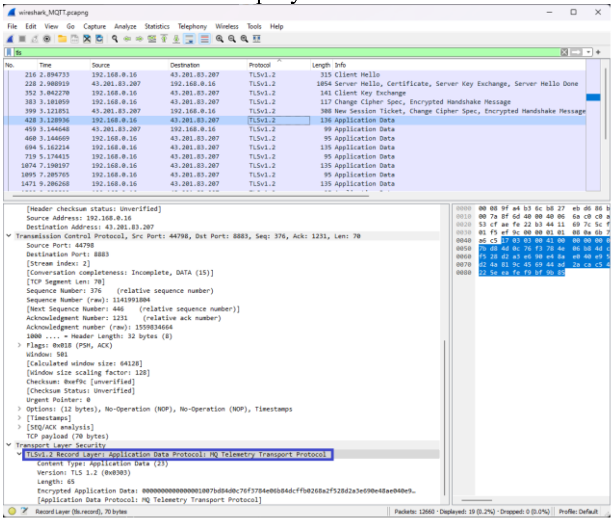
Figure 3. Crypto Hardware API Stack

The proposed system design encompasses a comprehensive architecture that seamlessly integrates hardware components, cryptographic protocols, and cloud-based services to create a robust and secure IoT ecosystem. At its foundation lies the Hardware Secure Module (HSM), leveraging field-programmable gate arrays (FPGAs) to provide a secure hardware platform for cryptographic operations. This HSM serves as the cornerstone for ensuring the confidentiality and integrity of sensitive data transmitted within the system. Moving up the stack, the system interfaces with IoT devices, such as Raspberry Pi, which act as MQTT and LwM2M clients. These devices utilize SPI (Serial Peripheral Interface) to communicate with the HSM, enabling secure cryptographic operations and key management. The application stack diagram showcases the flow of data from the hardware layer, where cryptographic operations are performed, to the application layer, where MQTT and LwM2M protocols are utilized for communication and device management tasks. This layered architecture ensures a clear separation of concerns, with each component fulfilling its specific role in enhancing the security and functionality of the IoT system.

Within this architecture, MQTT and LwM2M protocols play pivotal roles in facilitating communication and management between IoT devices and cloud-based services. MQTT serves as a lightweight and efficient messaging protocol for data exchange between devices and the cloud, with AWS IoT acting as the MQTT broker. This allows devices to publish data to topics and subscribe to receive messages, enabling real-time communication and data processing within the cloud environment. Additionally, LwM2M protocol is integrated into the system to provide standardized mechanisms for remote device management. AWS IoT serves as a management server for LwM2M devices, enabling device registration, authentication, and management tasks such as firmware updates and configuration changes. By leveraging the capabilities of AWS IoT as both an MQTT broker and a management server for LwM2M devices, the system achieves seamless integration with cloud-based services, enhanced device management capabilities, and robust communication mechanisms for IoT deployments.

4. Result Analysis

The experimental results demonstrate the effectiveness and efficiency of the proposed system architecture in securing IoT communications and managing device interactions. Through performance evaluation, it was observed that the integration of ARIA cryptography within the Hardware Secure Module (HSM) significantly enhanced the security of cryptographic operations while minimizing resource overhead. This ensured the confidentiality and integrity of data transmitted between IoT devices and cloud-based services. Furthermore, the comparative analysis of MQTT and LwM2M protocols highlighted their respective strengths and suitability for different IoT use cases. MQTT excelled in providing lightweight and real-time messaging capabilities, while LwM2M proved invaluable for standardized device management tasks such as firmware updates and configuration changes. By leveraging AWS IoT as both an MQTT broker and a management server for LwM2M devices, the system achieved seamless integration with cloud services, enabling scalable and secure IoT deployments.


wireshark_MQTT.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tls

```
No.   Time      Source         Destination    Protocol  Length Info
  216 2.894733  192.168.0.16   43.201.83.207  TLSv1.2    315 Client Hello
  228 2.908919  43.201.83.207  192.168.0.16   TLSv1.2   1054 Server Hello, Certificate, Server Key Exchange, Server Hello Done
  352 3.042270  192.168.0.16   43.201.83.207  TLSv1.2    141 Client Key Exchange
  383 3.101059  192.168.0.16   43.201.83.207  TLSv1.2    117 Change Cipher Spec, Encrypted Handshake Message
  399 3.121851  43.201.83.207  192.168.0.16   TLSv1.2    308 New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
  428 3.128936  192.168.0.16   43.201.83.207  TLSv1.2    136 Application Data
  459 3.144648  43.201.83.207  192.168.0.16   TLSv1.2     99 Application Data
  460 3.144669  192.168.0.16   43.201.83.207  TLSv1.2     95 Application Data
  694 5.162214  192.168.0.16   43.201.83.207  TLSv1.2    135 Application Data
  719 5.174415  192.168.0.16   43.201.83.207  TLSv1.2     95 Application Data
 1074 7.190197  192.168.0.16   43.201.83.207  TLSv1.2    135 Application Data
 1095 7.205765  192.168.0.16   43.201.83.207  TLSv1.2     95 Application Data
 1471 9.206268  192.168.0.16   43.201.83.207  TLSv1.2    135 Application Data

    [Header checksum status: Unverified]
    Source Address: 192.168.0.16
    Destination Address: 43.201.83.207
v Transmission Control Protocol, Src Port: 44798, Dst Port: 8883, Seq: 376, Ack: 1231, Len: 70
    Source Port: 44798
    Destination Port: 8883
    [Stream index: 2]
    [Conversation completeness: Incomplete, DATA (15)]
    [TCP Segment Len: 70]
    Sequence Number: 376    (relative sequence number)
    Sequence Number (raw): 1141991804
    [Next Sequence Number: 446    (relative sequence number)]
    Acknowledgment Number: 1231    (relative ack number)
    Acknowledgment number (raw): 1559834664
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 501
    [Calculated window size: 64128]
    [Window size scaling factor: 128]
    Checksum: 0xef9c [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (70 bytes)
v Transport Layer Security
  v TLSv1.2 Record Layer: Application Data Protocol: MQ Telemetry Transport Protocol
      Content Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 65
      Encrypted Application Data: 0000000000000001007bd84d0c76f3784e06b84dcffb0268a2f528d2a3e690e48ae040e9…
      [Application Data Protocol: MQ Telemetry Transport Protocol]

0000  00 08 9f a4 b3 6c b8 27  eb d6 86 b
0010  00 7a 8f 6d 40 00 40 06  6a c0 c0 a
0020  53 cf ae fe 22 b3 44 11  69 7c 5c f
0030  01 f5 ef 9c 00 00 01 01  08 0a 6b 7
0040  a6 c5 17 03 03 00 41 00  00 00 00 0
0050  7b d8 4d 0c 76 f3 78 4e  06 b8 4d c
0060  f5 28 d2 a3 e6 90 e4 8a  e0 40 e9 5
0070  d2 4a 81 9c 45 69 44 ad  2a ca c5 4
0080  22 5e ea fe f9 bf 9b 85

Record Layer (tls.record), 70 bytes    Packets: 12660 · Displayed: 19 (0.2%) · Dropped: 0 (0.0%)    Profile: Default
```

Figure 4. MQTT wireshark result

However, challenges were also encountered during the implementation and evaluation phases, particularly in ensuring interoperability and compatibility between different components of the system. Additionally, further optimization may be required to address performance bottlenecks and scalability concerns, particularly in large-scale IoT deployments. Nevertheless, the results obtained underscore the potential of the proposed architecture in addressing the security and management challenges inherent in IoT ecosystems, laying the groundwork for future advancements in secure and efficient IoT deployments.

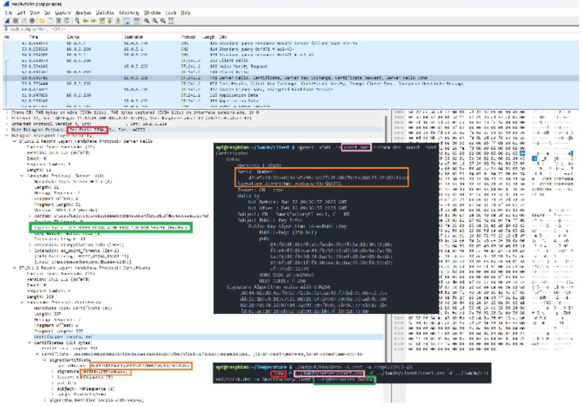


Figure 5. LwM2M wireshark result

5. Conclusion

In comparing LwM2M and MQTT, distinct advantages emerge, catering to diverse needs within IoT deployments. LwM2M's multifaceted approach, spanning transport and application layers, harnesses CoAP for streamlined communication across varied transport protocols like I-JDR SMS, TCP, and NIDD. With support for defined data models and multiple payload formats such as TLV, JSON, Opaque, and CBOR, LwM2M fosters interoperability and standardized device management. Notably, its low bandwidth consumption and robust security protocols like TLS and DTLS 1.2+ position it as an optimal choice for IoT environments prioritizing efficient resource utilization and secure device interactions.

Conversely, MQTT offers simplicity and scalability, albeit predominantly at the transport layer, affording flexibility in payload encoding sans standardized data models. While excelling in real-time messaging and event-driven architectures, MQTT's bandwidth usage, contingent on payload encoding, may incur higher data overhead relative to LwM2M. Nonetheless, MQTT's wide adoption and extensive community support render it a preferred option for IoT deployments necessitating versatile communication patterns. Thus, the choice between LwM2M and MQTT hinges on nuanced project requirements, encompassing device management needs, bandwidth constraints, and security imperatives, mandating a discerning evaluation to ensure alignment with the objectives of IoT deployments.

6. Acknowledgement

“This work is financially supported by Korea Ministry of Land, Infrastructure and Transport(MOLIT) as 「Innovative Talent Education Program for Smart City」".

Reference

[1] M. Iqbal, A. M. Ari Laksmono, A. T. Prihatno, D. Pratama, B. Jeong and H. Kim, "Enhancing IoT Security: Integrating MQTT with ARIA Cipher 256 Algorithm Cryptography and mbedTLS," 2023 International Conference on Platform Technology and Service (PlatCon), Busan, Korea, Republic of, 2023, pp. 91-96, doi

[2] A. M. A. Laksmono, M. Iqbal, D. Pratama, A. T. Prihatno, D. Yun and H. Kim, "Secure Sensor Data Transmission in IoT: Robust Implementation of LwM2M on the Lightweight Device Communication," 2023 International Conference on Platform Technology and Service (PlatCon), Busan, Korea, Republic of, 2023, pp. 85-90, doi: 10.1109/PlatCon60102.2023.10255201.

[3] Köppel, B. (2013). Analysis of a Hardware Security Module’s High-Availability Setting. In 2013 IEEE Security & Privacy.

[4] Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C (2nd ed.). Wiley