Transactions of KSAE, Vol. 32, No. 8, pp.625-631 (August, 2024)

Check for updates

Copyright © 2024 KSAE / 225-01
pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2024.32.8.625

차량용 Secure OTA 네트워크 프로토콜 연구

신 윤 제[1] · 전 상 훈[*2]

국민대학교 소프트웨어학부[1] · 국민대학교 자동차IT융합학과[2]

A Research on Secure OTA Network Protocol for Vehicles

Yunje Shin[1] · Sanghoon Jeon[*2]

[1]*Department of Software, Kookmin University, Seoul 02707, Korea*
[2]*Department of Automobile and IT Convergence, Kookmin University, Seoul 02707, Korea*
(Received 12 January 2024 / Revised 19 March 2024 / Accepted 12 April 2024)

Abstract : In the evolving landscape of software-defined vehicles(SDVs), the demand for a robust and secure firmware update mechanism has grown substantially. Traditional verification methods that rely on vendor signatures, metadata, and timestamps are limited in guaranteeing absolute integrity. Despite the rising traction of decentralization in blockchain technology, its validation within automotive over-the-air(OTA) update scenarios remains inadequate. Moreover, within the automotive OTA domain, reliability takes precedence, though the pivotal role of performance must also be emphasized. Therefore, this research attempted to address these challenges by introducing an innovative strategy to enhance the integrity and reliability of over-the-air firmware updates(FOTA). The strategy can harmoniously integrate the widely acknowledged Message Queuing Telemetry Transfer(MQTT) protocol, a cornerstone in IoT applications, with blockchain-based verification methods, particularly focusing on Merkle trees. Its primary objective is to elevate both the dependability and efficiency of SDV firmware updates.

Key words : Software defined vehicle(소프트웨어 정의 차량), Over the air updates(무선 업데이트), Vehicle-to-everything (차량 사물 통신), Merkle tree(머클 트리), Message queuing telemetry transport(발행-구독 기반의 메시징 프로토콜)

1. 서 론

차량이 소프트웨어 정의 차량(SDV, Software Defined Vehicle)으로 변화함에 따라, 차량의 아키텍처도 중앙집중형으로 변모하고 있다. 이러한 변화의 일환으로, 차량과 다양한 요소들 간의 통신(V2X, Vehicle to Everything)을 가능하게 하는 OTA(Over the Air) 기술을 통한 펌웨어 업데이트의 도입이 증가하고 있다. 특히, 차량의 전자제어장치(ECU, Electronic Control Unit)에 대한 FOTA(Firm ware Over the Air)는 서비스 센터 방문 없이 리콜 수준의 문제를 해결할 수 있어 미래의 커넥티드 카에 필수적인 기술이다.

최근 ECU를 위한 펌웨어 OTA 업데이트 방식에 대한 관심이 증가하고 있다.[1] 그러나 단순히 각 ECU의 펌웨어에 업체의 서명, 메타데이터, 타임스탬프를 사용한 검증만으로는 위변조 가능성이 있어, 펌웨어의 무결성을 충분히 보장하기 어렵다.[2,3] 이에 따라, 블록체인 기술을 활용한 FOTA 보안 연구가 주목받고 있다.[4] 차량용 OTA는 다른 형태의 OTA와 달리 안전성이 최우선이며, 펌웨어 업데이트를 위한 블록체인의 탈중앙화와 무결성 검증 방식은 합의 알고리즘에 의존한다. 그러나 현재의 기술로는 분산 노드에서 발생할 수 있는 보안 문제나 잘못된 업로드의 영향을 완전히 제거하기에는 부족하여, 실제 차량에 적용하기에 충분히 검증되지 않았다.

차량용 OTA는 안정성뿐만 아니라 성능 측면에서도 중요하다. 현재 http만을 사용하여 OTA를 구현하는 것은 MQTT(without TLS)에 비해 많은 Overhead가 발생한다.[5]

*A part of this paper was presented at the KSAE 2023 Fall Conference and Exhibition
*Corresponding author, E-mail: sh.jeon@kookmin.ac.kr

마찬가지로 TLS를 적용한 https와 MQTT(with TLS)를 비교했을 때, https만을 사용하는 방식은 처리 시간과 페이로드 측면에서 더 큰 오버헤드를 유발한다.[6] 이에 기존에 수십 년 동안 사용된 중앙집중형 아키텍처를 바탕으로, 안전성 강화를 위해 TLS(Transport Layer Security)가 적용된 HTTPS와 성능 개선을 위해 사실상 IoT에서 표준 프로토콜로 자리 잡은 MQTT(with TLS)를 결합하여 사용한다. 그리고 블록체인 검증 기술인 머클트리(Merkle tree)를 이용해 펌웨어 무결성과 신뢰성을 보장하는 새로운 방식을 제안하고 구현한다. 본 연구는 다음과 같은 두 가지 주요한 기여를 제시한다.

- 중앙집중형 아키텍처를 기반으로, HTTPS와 MQTT 프로토콜의 장점을 효율적으로 결합하여 차량용 OTA 업데이트의 안전성과 성능을 동시에 향상시키는 새로운 접근 방법을 제안한다. 이 방식은 기존의 단일 프로토콜 접근법에 비해 더욱 강력한 보안과 더 높은 데이터 전송 효율성을 제공한다.
- 본 연구에서 개발한 TLS 기반의 HTTPS와 MQTT(with TLS)의 통합 사용 방안은 기존의 HTTPS만을 사용하는 OTA 방식과 비교하여 처리 시간 및 페이로드 관련 오버헤드를 현저히 감소시켰다.

2. 연구 배경

연구 배경 섹션에서는 본 연구가 제안한 프로토콜에서 사용하는 MQTT와 Merkle tree에 대한 설명과 사용 이유에 대한 배경에 대해서 논한다. 마지막으로 Scope 서브 섹션을 통해 본 연구의 범위를 설명한다.

2.1 MQTT

MQTT(Message Queuing Telemetry Transport)는 발행/구독(Publish and subscribe) 기반의 메시지 큐잉 프로토콜이다. 이 프로토콜은 특정 '주제(Topic)'를 중심으로 메시지가 발행되며, 해당 주제를 구독하고 있는 수신자들이 이 메시지를 받는다. MQTT의 가장 큰 장점 중 하나는 그것의 가벼운 구조로, 이는 낮은 대역폭과 제한된 리소스 환경에서도 효율적인 통신을 가능하게 한다. 이러한 특성은 IoT 환경에서 특히 유용하다.

MQTT는 0, 1, 2의 세 가지 수준의 QoS(Quality of Service)를 제공하여 메시지 전송의 신뢰성을 보장한다. QoS 0(At most once)는 메시지가 한 번만 전송되며, 전달 여부를 보장하지 않는다. QoS 1(At least once)는 메시지가 적어도 한 번 전달되도록 보장하며, QoS 2(Exactly once)는 메시지가 정확히 한 번만 전달되도록 보장한다. 이러한 다양한 QoS 수준을 통해 MQTT는 다양한 통신 환경 및 요구 사항에 맞춰 메시지 전달의 신뢰성을 조절할 수 있다.

MQTT의 성능이 빠른 이유는 그것의 경량화된 프로토콜과 메시지 구조 때문이다. MQTT 메시지는 헤더가 매우 간단하고, 메시지 전송에 필요한 추가적인 오버헤드가 적다. 이는 HTTP와 같은 더 복잡한 프로토콜에 비해 상대적으로 낮은 네트워크 대역폭을 사용하며, 더 빠른 메시지 전송을 가능하게 한다.

보안이 중요한 환경에서는 TLS(Transport Layer Security)를 적용한 MQTT를 사용하여 안전성을 확보할 수 있다. TLS를 적용함으로써, 메시지가 암호화되어 전송되며, 이는 데이터 무결성과 기밀성을 보장한다. 비록 TLS 적용 시 일정 수준의 오버헤드가 발생하나, MQTT의 경량화된 구조 덕분에 HTTPS보다 여전히 빠른 성능을 제공한다.

2.2 Merkle Tree

머클 트리는 블록체인 기술에서 중요한 역할을 하는 데이터 구조이며, 이진 트리(Binary tree) 형태를 취한다. 이 트리의 각 노드는 자신의 자식 노드들의 값들을 해싱(Hashing)하여 얻은 결과값을 저장한다. 이러한 방식으로, 머클 트리는 각 노드의 무결성을 보장하며, 전체 데이터 구조의 일관성을 유지한다.

펌웨어 업데이트 과정에서 머클 트리를 활용할 경우, 먼저 모든 펌웨어 파일을 작은 청크(Chunk)로 읽어와, 해시값을 계산한다. 이 해시값들은 머클 트리의 리프 노드(Leaf node)를 형성하며, 상위 노드는 이들 리프 노드의 해시값들을 다시 해싱하여 그 결과를 저장한다. 이 과정은 트리의 최상위 노드인 루트(Root)에 도달할 때까지 반복된다.

머클 트리의 강력한 특징 중 하나는, 단 하나의 데이터 청크가 변경되어도 이는 전체 트리에 영향을 미친다는 점이다. 즉, 하나의 청크의 데이터나 해시값이 바뀌면, 이는 해당 청크를 포함하는 모든 상위 노드의 해시값 변경으로 이어진다. 결국 루트 노드에 저장된 루트해시(Root hash)도 변경되어, 데이터의 변경이나 조작을 쉽게 감지할 수 있다. 이러한 특성 덕분에, 머클 트리는 펌웨어 업데이트 과정에서 펌웨어의 무결성과 신뢰성을 효과적으로 보장하는 데에 큰 역할을 한다.

2.3 Scope

본 연구는 SecureOTA의 세 가지 핵심 영역인 엔드 디바이스의 보안, 펌웨어 업데이트 방식, 그리고 통신 프로토콜 중에서 특히 통신 프로토콜에 대한 접근 방식에 중점을 두고 있다. 이 연구는 네트워크 보안성 확보에 중점

을 두고 진행되었으며 엔드 디바이스에서의 보안은 HSM을 통해 확보 된다고 가정한다.

따라서 MQTT와 HTTP 통신 프로토콜에 TLS(Transport Layer Security)를 적용하여 두 프로토콜의 장점을 활용하며 네트워크 보안을 강화하는 첫 단계로 삼는다 TLS의 사용은 데이터전송 과정에서의 암호화를 통해 데이터의 기밀성과 무결성을 보장한다. 추가적으로, 본 연구에서는 머클 트리(Merkle tree)를 활용하는 방법도 고려한다. 머클 트리는 네트워크를 통한 데이터 전송 과정에서 데이터가 손상되거나 변경되었을 때 이를 효과적으로 감지할 수 있는 구조이다. 각 데이터 블록의 해시가 트리의 노드로 구성되며, 이 해시들은 상위 노드로 집계되어 전체 트리의 무결성을 나타낸다.

따라서, 이 연구에서는 MQTT와 HTTPS 프로토콜을 결합하여 사용함으로써, 두 프로토콜의 장점을 활용한다. 이러한 결합은 네트워크 보안을 강화하는 첫 단계이다. 이어서, TLS를 적용하여 데이터 전송 과정에서의 암호화를 통해 보안을 더욱 강화한다. TLS는 데이터의 기밀성과 무결성을 보장하는 중요한 역할을 한다.

그러나 보안 시스템은 언제나 잠재적인 위협에 노출되어 있으며, TLS가 손상될 가능성도 배제할 수 없다. 이러한 상황에 대비하여, 본 연구에서는 머클 트리를 추가적인 보안 수단으로 활용한다. 머클 트리는 네트워크를 통한 데이터 전송 과정에서 데이터가 손상되거나 변경되었을 때 이를 효과적으로 감지할 수 있는 구조이다. 만약 TLS 보안이 어떤 이유로든 뚫리게 되면, 머클 트리를 통해 데이터의 무결성을 검증하고, 손상되거나 변경된 데이터를 신속하게 감지할 수 있다. 특히 통신 프로토콜의 보안 강화 방법에 대한 심도 있는 분석과 개선 방안을 모색하는 데 그 목적이 있으며 이와 같은 방식으로, 본 연구는 SecureOTA의 구현에 있어서 통신 프로토콜의 보안 강화에 초점을 맞추며, 네트워크를 통한 데이터 전송 과정에서 발생할 수 있는 다양한 보안 위협에 대응할 수 있는 강력하고 포괄적인 보안 체계를 제안한다.

3. 관련 연구

차량용 OTA(Over the Air) 업데이트 시스템에 관한 연구를 '통신의 신뢰도 및 데이터 전송 효율성 연구'와 '보안성 강화 연구'로 나누어 논의한다. 이러한 분류는 OTA 업데이트 시스템의 중요한 두 측면인 효율성과 보안성을 체계적으로 검토하기 위한 것이다.

'통신의 신뢰도 및 데이터 전송 효율성 연구' 부문은 OTA 시스템에서의 데이터 전송의 효율성과 신뢰성을 개선하는 방법에 중점을 둔다. 이는 네트워크 환경 내에서 데이터를 신속하고 안정적으로 전달하는 데 필요한 프로토콜과 기술의 최적화에 초점을 맞춘다. 여기에는 MQTT와 같은 경량 통신 프로토콜의 사용과 이를 보다 안전하게 만드는 방법들이 포함된다.

반면, '보안성 강화 연구' 부문은 데이터의 무결성과 안전성을 강화하는 방법에 중점을 둔다. 이는 펌웨어 업데이트 과정에서 데이터 보호를 강화하고, 중앙 집중식 시스템의 취약점을 극복하기 위한 방법들을 탐색한다. 이러한 접근은 블록체인 기술과 같은 분산 데이터베이스 솔루션의 사용을 포함한다.

본 연구는 이 두 분야를 통합적으로 고려하며, 특히 프로토콜 기반의 접근 방식에 중점을 둔다. 이러한 복합적인 접근 방식은 차량용 OTA 업데이트 시스템의 신뢰성과 효율성을 동시에 향상시키는 데 기여하며, 다양한 요구 사항에 대응하는 효과적인 솔루션을 개발하는 데 중요한 역할을 한다.

3.1 통신의 신뢰도 및 데이터 전송 효율성 연구

기존의 연구들 중 일부는 P2PMesh 아키텍처를 이용한 업데이트 시스템을 제안하고 있다.[7] 이와 더불어 펌웨어 파일을 슬라이싱하여 MQTT를 통해 전송하는 OTA 업데이트를 하거나,[8] 비보안 MQTT 통신에 대하여 256 비트 대칭 암호화 알고리즘을 사용하여 MQTT 프로토콜의 보안을 강화하는 연구가 진행되어 왔다.[9] 그러나 이러한 P2P 기반 연구들은 기존 프로토콜이 제공하는 인증 수준을 충분히 고려하지 않았기 때문에, 정보 유출과 같은 보안 문제를 야기할 수 있다.

이러한 문제점을 해결하기 위해 TLS를 적용한 MQTT를 사용한다.[12] 특히, 특정 Topic을 구독할 때는 최소한의 인증 절차를 구현하여 보안을 강화하였다. MQTT를 사용하여 펌웨어 파일을 슬라이싱 하는 방식은 데이터 손실 또는 펌웨어 파일의 무결성 손상의 위험이 있을 수 있다. 따라서, 본 연구에서는 이러한 위험을 방지하기 위해 업로드된 펌웨어 파일의 위치를 JSON 형태의 URL 주소로 제공함으로써, 펌웨어 파일의 무결성을 유지하고 데이터 손실을 방지한다.

3.2 보안성 강화 연구

블록체인을 이용한 OTA(Over the Air) 업데이트 관련 연구는 제조업체 중심의 중앙집중형 방식을 벗어나 탈중앙화 접근 방식을 모색하고 있다.[10] 이러한 연구들 중 일부는 제조업체 대신 분산 데이터베이스 시스템을 통해 IoT 장치의 펌웨어를 직접 다운로드하는 방법을 탐구한다. 또한 해시 테이블과 공개 블록체인을 이용하여 OEM의 펌웨어 무결성과 신뢰성을 보장하는 연구도 진

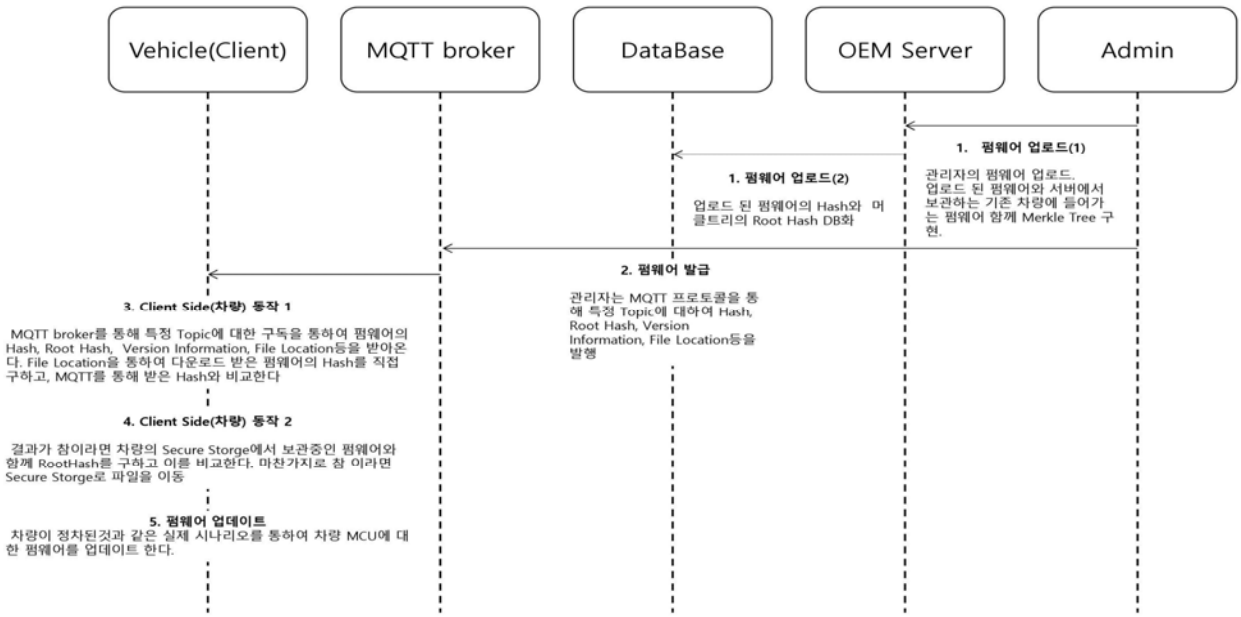

Fig. 1 OTA implementation sequence

행되고 있다.[11] 특히, 지속적인 업데이트가 필요한 펌웨어의 경우, 블록체인 기술을 활용함으로써 무결성과 신뢰성을 효과적으로 보장할 수 있다. 그러나 이러한 접근법은 벤더사의 분산 구조화를 필요로 하며, 이로 인해 분산된 노드의 안전성을 보장하기 위해 상당한 노력과 시간이 소요된다는 문제가 있다.

이러한 문제에 대응하기 위해 중앙집중형 방식을 유지하면서도 머클 트리(Merkle tree)를 활용한다. 머클 트리를 사용함으로써, 펌웨어의 신뢰성과 무결성을 보장할 수 있으며, 동시에 중앙집중형 구조의 효율성과 안정성을 유지할 수 있다. 이러한 접근 방식은 블록체인 기반의 탈 중앙화 방식이 요구하는 복잡성과 비용을 줄이면서도, 펌웨어 업데이트 과정에서 요구되는 높은 보안 수준을 달성할 수 있는 장점을 가진다.

4. 본 문

본문에서는 OTA(Over the Air) 펌웨어 업데이트 프로세스의 구체적인 구현 과정을 설명한다. 이 과정에서 펌웨어의 업로드와 발급 작업은 항상 순차적으로 진행되며, 보안을 위해 해시 알고리즘으로는 SHA-2 이상을 사용하며 해시 알고리즘의 선택은 보안 요구사항과 호환성, 계산 효율성 등 다양한 요소를 고려하여 결정될 수 있다. 또한, 추가적인 보안 조치로 화이트리스트 방식을 적용하여 펌웨어의 업로드 및 다운로드 과정을 관리한다. 이는 무단 접근을 방지하고 펌웨어 업데이트의 신뢰성을 높이는 데 중요한 역할을 한다.

Fig. 1은 OTA 구현의 전체적인 순서를 도식화하여 보여준다. 이 도식은 펌웨어 업로드 과정, MQTT를 활용한 펌웨어 발급, 그리고 최종적으로 차량(Client)에서의 동작까지의 과정을 단계별로 설명한다. 이러한 순서는 OTA 업데이트 프로세스의 효율성과 안전성을 보장하는 데 중요한 기준을 제공한다. 구현 목적으로, Gateway 역할은 Jetson Orin이 맡으며, MCU 역할은 아두이노가 수행한다.

OTA 업데이트 시스템의 각 단계의 통합적인 작동 방식에 대해 설명하며, 각 단계에서 적용되는 보안 조치의 중요성을 강조한다. 이는 전체 시스템의 신뢰성과 효율성을 향상시키는 데 중요한 역할을 한다.

4.1 펌웨어 업로드

무단 업로드나 변조를 방지하기 위해, 인증 및 인가 권한을 가진 특정 MCU를 담당하는 부서가 펌웨어 파일을 업로드하는 과정을 채택하고 있다. 이는 펌웨어의 안전한 관리와 보안을 확보하는 데 중요한 역할을 한다. 업로드된 펌웨어 파일은 서버에 저장되며, 서버는 저장된 모든 MCU의 펌웨어와 함께 머클 트리를 생성한다. 이 머클 트리는 서버에서 구현되며, 각 펌웨어 파일의 해시값과 루트 해시를 데이터베이스에 저장한다. 이러한 구조는 펌웨어의 무결성과 신뢰성을 확인하는 데 매우 중요하다.

4.2 펌웨어 발급

블록체인 기반 SOTA 접근 방식과는 달리, 본 연구에서는 머클 트리를 활용한 중앙집중식 네트워크 아키텍처를 도입한다. 이를 통해 발생할 수 있는 보안 위협을 최소화하기 위해, 본 시스템은 업로드 관리자와는 별개로 업데이트를 담당하는 관리자를 두어 접근 제어를 강화함으로써 보안성을 향상시킨다. 이 관리자는 MQTT 프로토콜을 이용하여, 차량이 구독하고 있는 Topic에 대한 신규 펌웨어의 해쉬값, 루트 해쉬값, 버전 정보, 파일 위치 등을 발행한다.

Topic에 대한 접근 인증을 위해 사용되는 키는 일반적으로 사용자 아이디와 비밀번호 형태를 취하며, 이러한 중요한 인증 정보는 보안성이 높은 HSM과 같은 안전한 저장 매체에 보관된다. HSM은 높은 수준의 보안을 제공하여, 키 정보가 외부로 유출되거나 타인에 의해 접근되는 것을 방지한다. 이러한 방식으로, 차량은 안전하게 MQTT 서버에 연결되며, 구독 중인 Topic에 대한 접근 권한을 확보하게 된다.

펌웨어 정보의 전송 및 업데이트 과정은 이와 같이 강화된 보안 메커니즘을 통해 높은 수준의 보안과 신뢰성을 유지한다. 이는 차량이 신뢰할 수 있는 출처로부터만 펌웨어 업데이트를 받도록 보장하며, 시스템을 무단 접근 및 악의적인 공격으로부터 보호한다.

4.3 차량 내 펌웨어 다운로드 및 검증 절차

차량 내의 Gateway 역할을 수행하는 Jetson Orin은 MQTT 프로토콜을 통해 펌웨어의 URL 주소를 수신한다. 이 주소를 사용하여, HTTP의 확장성과 신뢰성에 기반해 TLS를 통해 펌웨어를 안전하게 다운로드 받는다. 다운로드는 차량의 임시 디렉토리에 이루어지며, 이때 데이터 손실의 위험이 최소화된다.

다운로드된 펌웨어 파일은 4096바이트 크기의 청크로 나누어 읽혀지고, 각 청크에 대해 SHA-256 해시 알고리즘을 사용하여 해시 값을 계산한다. 이 과정은 펌웨어 파일의 무결성을 확인하는 데 중요하다. 계산된 해시 값은 이후 MQTT를 통해 전송받은 원본 펌웨어 파일의 해시값과 비교되어, 펌웨어의 정확성을 검증한다.

GateWay는 평상시에 차량의 네트워크 환경을 모니터링하며, 새로운 펌웨어 업데이트가 가능할 때 해당 정보를 수신하고 처리한다. 이는 주기적으로 서버와의 통신을 통해 최신 펌웨어 업데이트의 존재를 확인하는 과정을 포함한다. 이러한 방식으로, 차량의 펌웨어가 항상 최신 상태를 유지하도록 보장하며, 차량의 보안과 성능을 지속적으로 강화한다.

4.4 차량의 펌웨어 동기화 및 저장 절차

4.3절에서의 검증 과정을 성공적으로 마친 후, 차량의 Secure storage에서는 각 펌웨어 파일을 머클 트리의 리프 노드로 취급하여 처리한다. 즉, 차량에 보관된 모든 펌웨어 파일은 각각 머클 트리의 하나의 리프 노드에 해당하며, 이들 각각의 해시값을 계산하여 트리에 저장한다. 이러한 방식으로 머클 트리를 구성하고, 모든 리프 노드의 해시값을 결합하여 상위 노드의 해시값을 순차적으로 계산해 나간다. 이 과정은 루트 노드에 도달할 때까지 계속되며, 최종적으로 생성된 루트 해시는 차량에 보관된 모든 펌웨어 파일의 무결성을 대표한다. 차량에서 계산된 루트 해시는 서버에서 받은 루트 해시와 비교되며, 일치함을 확인하면 신규 펌웨어는 Secure storage 내 적절한 위치에 저장된다. 이러한 절차는 차량의 펌웨어 시스템이 안전하고 무결한 상태를 유지하는 데 중요한 역할을 한다.

4.5 펌웨어 업데이트

최종적으로 차량의 펌웨어 업데이트는 차량이 안전하게 정차된 상태를 확인한 후에 진행된다. 이는 실제 운행 상황에서 펌웨어 업데이트가 차량의 운행에 영향을 주지 않도록 보장하기 위함이다. 본 논문의 시나리오에서는, 차량 내에 설치된 Jetson Orin이 중앙 제어 장치로서의 역할을 하며, 아두이노 CLI(Command Line Interface)를 사용하여 아두이노 기반의 MCU에 펌웨어를 업로드한다.

단지 구현의 용이성과 접근성을 고려한 선택으로 Jetson Orin을 사용하여 아두이노 기반의 MCU에 펌웨어를 업로드하는 방식을 채택했다.이 과정에서 GateWay는 먼저 안전한 네트워크 연결을 확인하고, Secure storage에서 검증된 신규 펌웨어를 불러온다. 그 후, MCU로 펌웨어를 전송하며, 이 과정은 특별히 설계된 스크립트나 명령어 세트를 통해 자동화될 수 있다. 펌웨어 업로드가 완료되면, GateWay는 업데이트의 성공 여부를 확인하고, 필요한 경우 추가적인 후처리 작업을 수행한다.

이러한 접근 방식은 차량의 펌웨어 업데이트 과정을 안전하고 신뢰할 수 있게 하며, 차량의 성능과 신뢰성을 유지하는 데 기여할 수 있을 것이다.

5. 검 증

본 연구에서 제안한 방법을 검증하기 위해 아래와 같은 두 가지 측면에서 검증을 한다.

1) 본 논문이 제안한 방법은 어떤 공격을 방어할 수 있는가?: OTA 프로토콜을 통해 발생할 수 있는 공격을 4

가지(Spoofing, Man In-the-Middle aTtack, Distributed Denial of Service, Duplicate update)로 구분하고, 각 공격에 대한 방어 가능 여부를 분석하였다.

2) 본 논문이 제안한 방법의 성능(Latency)은 어느정도인가?: 제안한 방법(MQTT with tls)은 HTTPS와 MQTT without tls 대비 Latency 측면에서 성능 향상 및 감소 여부를 측정한다.

5.1 보안성 검증

Table 1 Mqtt(wtih TLS) + Merkle trees

Dos/DDos	MITM	Spoofing	Duplicate update
X	O	O	O

먼저, Table 1은 OTA 프로토콜을 통해 발생할 수 있는 대표적인 공격은 크게 4가지로 정의하였다. Dos/DDoS 공격은 시스템의 자원을 고갈시켜 서비스 접근을 차단하는 방식으로 작동한다. MITM 공격에서는 공격자가 통신 과정에서 메시지를 변조하거나 가로챈다. Spoofing은 공격자가 다른 사용자나 시스템으로 위장하여 불법적으로 통신을 수행한다. Duplicated update 공격은 공격자가 이미 전송된 업데이트 패킷을 캡처하여 다시 전송함으로써 시스템을 혼란시키려는 공격 유형이다. MQTT 프로토콜의 특성상, 이러한 공격에 대한 취약성이 있을 수 있다. MQTT는 경량 메시징 프로토콜로, 효율적인 데이터 전송을 위해 설계되었지만, 기본적으로 메시지의 중복 전송을 자동으로 감지하고 처리하는 메커니즘을 포함하지 않는다.

이러한 특성 때문에, MQTT를 사용하는 시스템에서는 중복된 업데이트 메시지가 원본 메시지로 오인되어 처리될 위험이 있다. 예를 들어, 공격자가 펌웨어 업데이트 메시지를 캡처하고 이를 다시 시스템에 전송한다면, 시스템은 이미 적용된 업데이트를 새로운 것으로 인식하고 재적용할 수 있다. 이는 불필요한 리소스 소모, 시스템의 불안정성 증가, 심지어는 오작동을 초래할 수 있다.

TLS를 사용하게 되면 Spoofing과 MITM 공격에 대해서는 암호화와 인증서를 통해 신원을 검증할 수 있어 이러한 공격에 대응할 수 있다. 만약 업데이트 패킷을 캡처하여 재전송하는 중복 업데이트 공격을 시도한다면, 머클 트리를 이용한 무결성 검증으로 이를 방어할 수 있다. 그러나 서비스 거부 공격(Denial of Service, DoS)의 경에는 본 연구에서 제안한 방법으로는 방어할 수 없으며, 추가로 방화벽이나 DDoS 대응 기술이 필요하다.

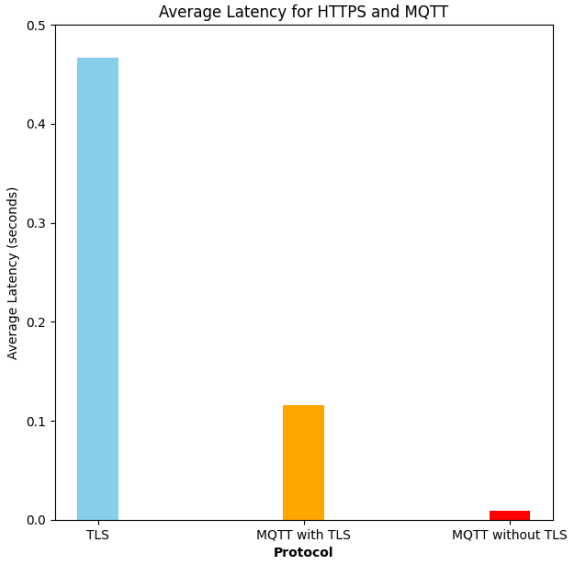

Fig. 2 Average Latency for HTTPS and MQTT

5.2 성능 검증

Fig. 2에서 보여주는 성능 측정 결과는 총 10번의 독립적인 트랜잭션을 수행한 후, 이들의 평균 레이턴시를 계산하여 얻어졌다. 실험 환경은 서버 측에서 스프링 부트를 활용하고 클라이언트 측에서는 파이썬을 사용하여 구성되었다. 이러한 시스템은 100 Mbps의 대역폭을 갖는 공유기를 통해 네트워크 환경을 설정하였다. 실험에 사용된 각 트랜잭션은 4.2절에서 설명한 바와 같이, OTA에 필요한 정보를 JSON 형식으로 변환한 데이터를 기반으로 진행되었다.

HTTPS를 사용했을 경우, 평균 레이턴시는 0.467초로 측정되었다. 반면, TLS가 적용된 MQTT를 사용했을 때는 평균 레이턴시가 0.116초로, HTTPS 대비 약 4배 빠른 성능을 보였다. 이는 보안 요구사항을 충족하면서도 성능을 향상시킬 수 있음을 시사한다. 그러나 TLS가 적용되지 않은 MQTT의 경우 평균 레이턴시가 0.0092초로 가장 빠르나, 보안이 중요한 환경에서는 적합하지 않을 수 있다는 점을 감안해야 한다.

이러한 결과들을 통해 OTA 프로토콜을 구현하는 경우, HTTPS만을 사용하는 것보다는 TLS를 적용한 MQTT를 함께 사용하는 것이 성능 측면에서 상대적으로 우수함을 알 수 있다. 보안과 성능이라는 두 가지 중요한 요소를 모두 고려할 때, MQTT with TLS의 사용이 HTTPS를 사용하는 것보다 효과적인 대안으로 제시될 수 있다.

6. 결 론

본 연구에서는 중앙 집중형 아키텍처를 기반으로 하

면서 머클 트리를 활용하여 차량용 OTA(Over the Air) 업데이트의 펌웨어 무결성을 보장하였다. 기존의 HTTP만을 사용하는 방식과 달리, MQTT 프로토콜을 병행 사용함으로써 더욱 효율적인 차량용 OTA 업데이트 시스템을 구현하고 제시했다. 이러한 접근 방식은 펌웨어 업데이트 과정의 신뢰성과 효율성을 높이는데 중요한 기여를 한다.

향후 연구 계획에서는 본 연구의 결과를 바탕으로 추가적인 개선 사항을 도입할 예정이다. 첫 번째로, 서버의 이중화와 DDoS 공격에 대응하는 방안을 마련함으로써 단일 장애점의 문제를 해결할 계획이다. 이는 시스템의 안정성과 지속 가능성을 강화하는데 중점을 둔다. 특히, 차량이 정지 상태에 있을 때 업데이트를 진행하는 것은 차량 및 주변 환경의 안전을 보장하기 위해 필수적이다. 이는 업데이트 과정 중 차량의 운행으로 인해 발생할 수 있는 잠재적 위험을 방지하고, 업데이트가 차량 시스템에 올바르게 적용되도록 하기 위함이다. 따라서, 차량이 정지된 상태인지 여부를 정확히 판단하는 추가적인 시나리오를 적용함으로써, 업데이트 과정의 안전성을 더욱 강화할 예정이다.

마지막으로, 머클 트리의 구현에 있어서 성능 최적화를 위한 알고리즘을 개발하는 것도 중요한 과제로 설정되어 있다. 이는 전체 시스템의 효율성을 향상시키는데 기여할 것이다. 이러한 계획을 통해, 차량용 OTA 업데이트 시스템은 보안성과 효율성 면에서 지속적으로 개선되며, 더욱 신뢰할 수 있는 차량 통신 환경을 제공할 수 있을 것으로 기대된다.

후 기

본 연구는 대한민국 정부(교육부)의 재원으로 한국연구재단의 4단계 BK21사업(5199990814084)의 지원을 받았음.

References

1) G. Kornaros, O. Tomoutzoglou, D. Mbakoyiannis, N. Karadimitriou, M. Coppola, E. Montanari, I. Deligiannis and G. Gherardi, “Towards Holistic Secure Networking in Connected Vehicles Through Securing CAN-Bus Communication and Firmware-over-the-Air Updating,” Journal of Systems Architecture, Vol.109, Paper No.101761, 2020.
2) D. Mbakoyiannis, O. Tomoutzoglou and G. Kornaros, “Secure Over-the-Air Firmware Updating for Automotive Electronic Control Units,” Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, pp.174-181, 2019.
3) K. Zandberg, K. Schleiser, F. Acosta, H. Tschofenig and E. Baccelli, “Secure Firmware Updates for Constrained IoT Devices Using Open Standards: A Reality Check,” IEEE Access, Vol.7, pp.71907-71920, 2019.
4) S. Dhakal, F. Jaafar and P. Zavarsky, “Private Blockchain Network for IoT Device Firmware Integrity Verification and Update,” IEEE 19th International Symposium on High Assurance Systems Engineering (HASE), pp.164-170, 2019.
5) B. Wukkadada, K. Wankhede, R. Nambiar and A. Nair, “Comparison with HTTP and MQTT in Internet of Things (IoT),” International Conference on Inventive Research in Computing Applications (ICIRCA), pp.249-253, 2018.
6) Y. Tang, F. Wu, Z. Liu and W. Mai, “Research on NAT Traversal Communication Based on MQTT,” 9th International Conference on Communications and Broadband Networking, pp.186-191, 2021.
7) H. Chandra, E. Anggadjaja, P. S. Wijaya and E. Gunawan, “Internet of Things: Over-the-Air (OTA) Firmware Update in Lightweight Mesh Network Protocol for Smart Urban Development,” 22nd Asia-Pacific Conference on Communications (APCC), pp.115-118, 2016.
8) K. Sahlmann, V. Clemens, M. Nowak and B. Schnor, “MUP: Simplifying Secure Over-the-Air Update with MQTT for Constrained IoT Devices,” Sensors, Vol.21, No.1, Paper No.10, 2020.
9) I. Stoev, S. Zaharieva, A. Borodzhieva and G. Staevska, “An Approach for Securing MQTT Protocol in ESP8266 WiFi Module,” XI National Conference with International Participation (ELECTRONICA), pp.1-4, 2020.
10) W. J. Tsaur, J. C. Chang and C. L. Chen, “A Highly Secure IoT Firmware Update Mechanism Using Blockchain,” Sensors, Vol.22, No.2, Paper No.530, 2022.
11) G. Falco and J. E. Siegel, “Assuring Automotive Data and Software Integrity Employing Distributed Hash Tables and Blockchain,” arXiv preprint arXiv:2002.02780, 2020.
12) Y. Shin, B. Kang, S. Cho and S. Jeon, “Research on a Secure OTA Protocol for Vehicles Using Merkle Trees and MQTT,” KSAE Fall Conference Proceedings, pp.788-793, 2023.