SecOTA Universal Secure Flash Over The Air Framework for Embedded and Internet of Things Systems

Omar Elshopky
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201700552@cis.asu.edu.eg

Ahmed Eltaher
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201701048@cis.asu.edu.eg

Mahmoud Nabil
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201701177@cis.asu.edu.eg

Eslam Gomaa
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201701056@cis.asu.edu.eg

Enas Rizk
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201700169@cis.asu.edu.eg

Rana Abd Elkahlik
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
20201700267@cis.asu.edu.eg

T.A. Fatma Elwasify
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
fatma.gamal@cis.asu.edu.eg

Dr. Karim Emara
Computer Systems Department
Faculty of Computer and Information Sciences, Ain Shams University
Cairo, Egypt
Karim.emara@cis.asu.edu.eg

***Abstract*— This thesis addresses the lack of secure, easy-to-deploy on-premise flash over the air (FOTA) frameworks that consider physical security, providing companies with an efficient way to stop the financial losses caused by recalls, due to software failures, emerging security threats, or rapid development bugs, and save time developing in-house FOTA solutions. Our project develops a universal and secure FOTA comprehensive framework for embedded and IoT systems, implementing the Uptane standard, which is based on The Update Framework (TUF). TUF is primarily used for software updates by companies like Microsoft, Google, and Amazon, while Uptane adapts it to the environment and capabilities of embedded and IoT devices. The development was conducted based on the ISO 21434 standard, ensuring adherence to automotive cybersecurity guidelines. Our work introduces a specialized Threat Analysis and Risk Assessment (TARA) focusing on client-side attacks, often overlooked in favor of backend attacks. Using both the STRIDE methodology and the MITRE ATT&CK framework for identifying threats and tactics, techniques, and procedures (TTPs), we identified and mitigated risks related to physical and client app attacks on the Electronic Control Unit (ECU). Mitigations include updating deprecated cryptographic algorithms to the latest standards, network Intrusion Prevention Systems (IPS), and process manipulation countermeasures. Our framework, SecOTA, is designed with security by design principles and employs a microservice architecture for compromise resilience. It is ready to deploy universally across a wide range of boards, ensuring a robust and adaptable solution for secure FOTA in embedded and IoT systems.**

Keywords—SecOTA, FOTA, Security, Embedded Systems, IoT Security, Uptane, TUF, TARA, Cybersecurity, Cryptographic Security, HSM, Secure Boot, ECU Security, Microservice.

I. INTRODUCTION

In today's rapidly evolving technological landscape, embedded systems and IoT devices are becoming increasingly integral to various industries, particularly the automotive sector. The necessity for efficient, secure, and reliable over-the-air (OTA) updates has grown in parallel with the expansion of these systems. Traditional methods of firmware updates are often cumbersome, insecure, and inefficient, posing significant risks and challenges.

This paper introduces the SecOTA framework, a secure and comprehensive solution for OTA updates in embedded and IoT systems, specifically tailored to meet the demands of automotive cybersecurity standards. SecOTA addresses a critical gap in existing OTA update solutions by focusing on both the physical and cyber threats that target embedded systems. The framework is designed to comply with the Uptane standard, which extends The Update Framework (TUF) for automotive environments. Uptane adapts TUF's robust security measures to the specific needs of embedded devices, providing a multi-layered approach to ensure the authenticity and integrity of updates.

The SecOTA framework is developed in accordance with the ISO 21434 standard, which outlines requirements for automotive cybersecurity. A key component of this work is a specialized Threat Analysis and Risk Assessment (TARA), emphasizing client-side attacks that are often overlooked in favor of server-side threats. By employing methodologies such as STRIDE and the MITRE ATT&CK framework, we identified and mitigated risks associated with physical and client-side attacks on Electronic Control Units (ECUs).

Our research and development process encompassed several phases, including extensive literature review, system design and architecture, backend and frontend development, security-enhanced client development, integration, testing, and deployment. The resulting framework not only ensures

979.83503-2210-1/23/$31.00 @2023 IEEE

secure OTA updates but also enhances the overall resilience and adaptability of the system.

This paper delves into the specifics of the SecOTA framework, detailing its architecture, security features, implementation process, and the results of our extensive testing. The goal is to provide a robust, adaptable solution for secure OTA updates that can be deployed across a wide range of embedded and IoT devices, significantly improving their security posture and operational efficiency.

II. Literature Review

A. The Update Framework (TUF)

The Update Framework (TUF) is pivotal in securing over-the-air (OTA) updates for embedded and IoT systems. It employs cryptographic methods to guarantee the authenticity and integrity of software updates, utilizing decentralized trust roles such as Root, Targets, Snapshot, and Timestamp. This modular architecture supports both centralized and decentralized deployment models, enhancing security and simplifying management across IoT deployments. By incorporating metadata like version numbers and cryptographic hashes, TUF ensures verifiability under compromised conditions, bolstering overall software deployment security.

B. Uptane

Uptane builds upon TUF's principles, focusing specifically on automotive and high-stakes embedded systems. It introduces additional security layers, including hierarchical signing roles like Director, Targets, and Image Repositories. This framework ensures the authenticity and integrity of OTA updates throughout the supply chain, with mechanisms for rollback protection to maintain system integrity in the face of failures or malicious attacks. While Uptane's hierarchical trust model enhances security for automotive environments, it poses challenges in interoperability and governance across multiple vendors.

C. Comparative Analysis and Critique

Both TUF and Uptane offer robust solutions tailored to specific deployment needs and security requirements. TUF's versatility supports diverse IoT environments and integration with various update infrastructures, whereas Uptane's specialized focus on automotive cybersecurity provides crucial protections against physical and cyber threats to Electronic Control Units (ECUs). Future research should concentrate on improving interoperability between these frameworks, enhancing usability for end-users, and addressing emerging threats in dynamic IoT ecosystems. Leveraging the strengths of TUF and Uptane can enable stakeholders to develop comprehensive strategies for secure and reliable OTA updates in embedded and IoT systems.

III. System Architecture

The SecOTA framework is tailored to meet the demanding requirements of Over-The-Air (OTA) updates in embedded and IoT systems, specifically designed to enhance automotive cybersecurity standards. It features a comprehensive architecture centered around four main components: the Director Server, Image Server, Time Server, and Targets (ECUs). This architecture is characterized by its robustness, adaptability, and security, ensuring seamless deployment and operation across diverse automotive environments.

A. Director Server

The Director Server manages device and ECU provisioning, cryptographic key management, and update campaign orchestration. It comprises a Node.js server for core logic, a Fastify HTTP API for external interactions, a PostgreSQL database for structured data storage, a Keys Vault for secure key management, and Valkey Cache to optimize data retrieval performance.

B. Image Server

Responsible for image file management, the Image Server handles upload, signing, and retrieval of update images. It includes a Node.js server for operational logic, a Fastify HTTP API for user interaction, a PostgreSQL database for image metadata storage, Blob Storage for binary data handling, and Valkey Cache for efficient data caching.

C. Time Server

Utilizing the Network Time Security (NTS) standard, the Time Server securely synchronizes time for primary ECUs, ensuring reliable timestamping for update operations without relying on external, potentially compromised sources.

D. Target (ECUs)

Targets consist of Primary ECUs with internet connectivity and Secondary ECUs connected via wired protocols to Primary ECUs. Primary ECUs manage full update verification and employ firewalls for network security, while Secondary ECUs perform minimal verification and rely on Primary ECUs for updates.

E. Inter-server Communication

Communication between the Image Server and the Director Server is facilitated by a Message Queue, ensuring asynchronous message delivery for enhanced system resilience and scalability.

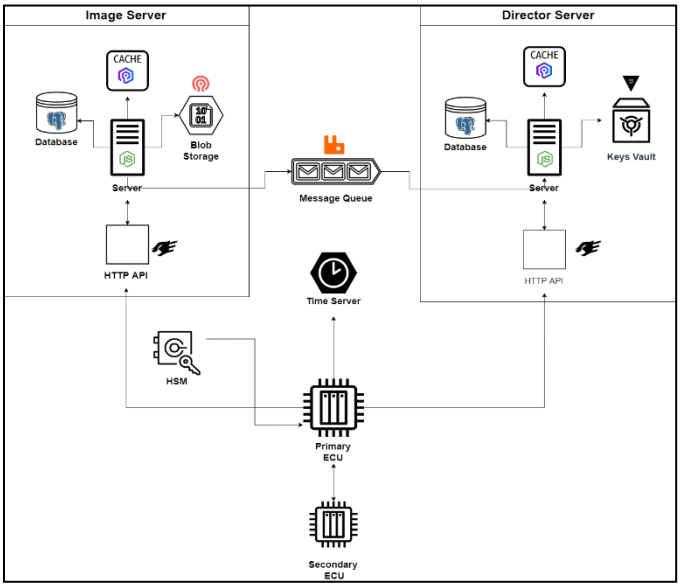

Fig. 1. SecOTA system architecture

The SecOTA architecture integrates robust security measures and scalable infrastructure to support secure OTA updates in diverse embedded and IoT environments. This structure adheres to automotive cybersecurity standards, emphasizing operational efficiency and resilience against evolving threats.

IV. Servers Infrastructure Setup Configuration

The server infrastructure for the SecOTA framework is designed to accommodate various deployment scenarios, addressing scalability, separation requirements, and cost considerations:

A. Scenario 1: Single Machine

This scenario consolidates all SecOTA services (Director, Image, and Time) onto a single robust server. It offers a cost-effective solution with simplified management but may present scalability limitations and a single point of failure risk. The machine contains the following instances:

- PostgreSQL DBs (2 instances)
- Redis Instances (2 instances)
- RabbitMQ
- Node.js Applications (2 instances)
- React Applications (2 instances)
- CEPH Instance
- HashiCorp Vault
- Nginx

B. Scenario 2: Logical Separated

In this configuration, the SecOTA framework is distributed across six distinct servers, each serving specific roles crucial for secure OTA updates. This setup supports medium to large-scale deployments, optimizing infrastructure for robust and secure operations in embedded and IoT systems.

V. Threat Analysis and Risk Assessment

This section details the comprehensive Threat Analysis and Risk Assessment (TARA) conducted for the SecOTA framework, adhering to ISO 21434 standards. The assessment focused primarily on identifying security threats and vulnerabilities associated with Over-The-Air (OTA) updates in embedded and IoT systems, with a particular emphasis on the client-side (primary and secondary ECUs).

The methodology integrated several industry-standard frameworks to systematically evaluate potential risks and their impacts. The STRIDE framework was employed to categorize threats into Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service (DoS), and Elevation of Privilege. This analysis highlighted vulnerabilities such as unauthorized access attempts and data tampering during firmware transmission.

Furthermore, leveraging the MITRE ATT&CK framework provided insights into adversarial tactics, techniques, and procedures (TTPs) that could exploit identified vulnerabilities. This included assessing threats such as command and control, lateral movement, and data exfiltration within OTA update processes.

The assessment also utilized the Common Vulnerability Scoring System (CVSS) to quantify the severity of identified vulnerabilities. Each finding was categorized based on its impact on the primary and secondary ECUs, ranging from Critical-Risk to Low-Risk. Critical-Risk findings, such as deprecated OpenSSL vulnerabilities and process manipulation with firmware update privileges, posed immediate threats to system integrity and security.

Fig. 2 illustrates the Item Definition Model used to visualize the components and interactions within the SecOTA framework, emphasizing the critical points of vulnerability and the flow of secure update processes between servers and ECUs.

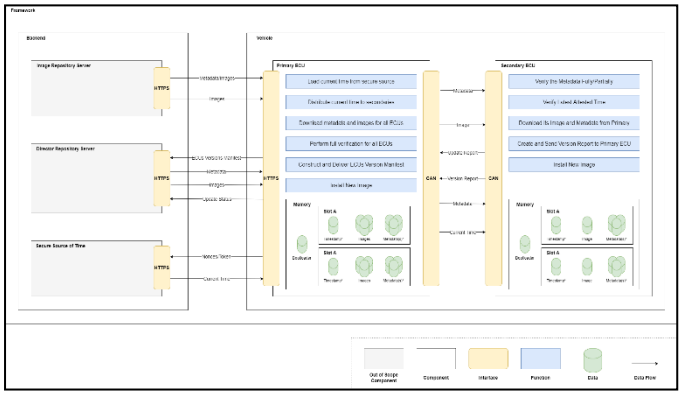

Fig. 2. SecOTA TARA Item Definition

The TARA process identified critical vulnerabilities in the uptane client application, categorized by severity and impact on OTA update security.

A. Critical-Risk Findings

1) Process Manipulation Vulnerability

A critical risk finding involved vulnerabilities in the firmware update process, where malicious processes could manipulate privileged access. This manipulation could lead to unauthorized modifications and compromise the integrity of ECUs, posing serious risks such as unauthorized code execution and persistent malware. Mitigation efforts included sandboxing privileged processes and implementing strict access controls to prevent unauthorized firmware modifications.

2) Open Ports and Insecure Connections Vulnerability

Other vulnerabilities were identified in the system's use of open ports and unsecured network connections. These vulnerabilities exposed the system to unauthorized access and potential data breaches, making it susceptible to denial of service attacks and compromising system integrity through known service vulnerabilities. Mitigation measures included deploying Intrusion Prevention Systems (IPS) to monitor network traffic, detect suspicious activities, and block unauthorized access attempts, thereby enhancing network security and protecting OTA update processes.

B. High-Risk Findings

1) Deprecated OpenSSL Vulnerability

The TARA process identified a high-risk vulnerability related to deprecated versions of OpenSSL. These outdated versions exposed the system to significant security risks due to their reliance on insecure cryptographic algorithms and lack of security updates. This vulnerability posed a severe threat to OTA update integrity and confidentiality. Mitigation strategies focused on updating cryptographic operations to leverage the latest OpenSSL versions, ensuring robust encryption and protection against known vulnerabilities.

Known OTA vulnerabilities were identified during the threat analysis and risk assessment. The **drop-request attack**, a high-risk OTA vulnerability, involves blocking network traffic both inside and outside the vehicle, effectively preventing ECUs from receiving updates. Similarly, the **slow retrieval attack** delays OTA update delivery to ECUs, exploiting security vulnerabilities before patches can be applied. The **mix-and-match attack**, rated critical, allows attackers with compromised repository keys to release arbitrary combinations of new OTA image versions. The **freeze attack**, also critical, perpetually sends the last known OTA update to an ECU, even in the presence of newer updates. High-risk OTA vulnerabilities include the **partial bundle installation attack**, which allows incomplete updates by selectively dropping OTA traffic to ECUs, and the **rollback attack,** tricking ECUs into installing outdated OTA software with known vulnerabilities. Another high-risk OTA threat, the **mixed-bundles attack**, disrupts ECUs by causing them to install incompatible OTA software versions simultaneously. Finally, the **endless data attack** induces ECU crashes by inundating them with excessive OTA data until storage exhaustion occurs.

VI. Conclusion

In conclusion, this paper has presented the SecOTA framework, a robust solution designed to address the intricate challenges of OTA updates in embedded and IoT systems, with a specific focus on automotive cybersecurity standards. Through a comprehensive literature review, meticulous system architecture design, rigorous threat analysis, and risk assessment following ISO 21434 guidelines, this framework integrates advanced security measures to ensure the authenticity, integrity, and confidentiality of OTA updates. By leveraging standards like Uptane and methodologies such as STRIDE and MITRE ATT&CK, vulnerabilities and potential threats targeting ECUs were identified and mitigated effectively. The deployment scenarios underscored the framework's scalability, resilience, and adaptability across varying infrastructures. Future enhancements could further fortify the framework's capabilities, particularly in addressing emerging threats and accommodating evolving industry standards. Ultimately, SecOTA stands poised as a versatile and dependable solution, enhancing the security posture and operational efficiency of embedded and IoT systems.

Acknowledgment

All praise and thanks to ALLAH, who bestowed upon us the ability, experience, and support to complete this work. We hope this work will be accepted from us and prove beneficial for future research and products.

We are deeply grateful to those who helped us through the long hard nights, our parents, families, and friends, whose unwavering support sustained us throughout our years of study. We hope to reciprocate their kindness.

Our sincerest gratitude goes to our supervisors, Dr. Karim Emara and T.A. Fatma Elwasify, whose patience, knowledge, and experience were invaluable throughout our thesis. Thank you for the countless hours spent in meetings, reviewing work, and brainstorming.

We also extend our heartfelt thanks to our mentor, Eng. Mark Attia, who generously gave more than his work hours to share his professional experience and insights from top-tier FOTA-related projects, which made this work as professional as it is.

Finally, we would like to thank everyone who offered us support and encouragement.

References

[1] Uptane Framework, "Secure Software Update Framework for Automobiles."

[2] E. Cebel, N. Donum, and H. Karacali, "Platform Independent Embedded Linux OTA Method," The European Journal of Research and Development, vol. 2, no. 4, pp. 243–252, 2022.

[3] S. E. Jaouhari and E. Bouvet, "Toward a generic and secure bootloader for IoT device firmware OTA update," International Conference on Information Networking (ICOIN), Jeju-si, Korea, 2022, pp. 90-95, doi: 10.1109/ICOIN53446.2022.9687242.

[4] L.-C. Duca, A. Duca, and C. Popescu, "OTA Secure Update System for IoT Fleets," International Journal of Advanced Networking and Applications, vol. 13, Dec. 2021.

[5] M. Opdenacker, "Implementing A/B System Updates with U-boot," Embedded Linux Conference Europe, 2022.

[6] M. Nottingham, Ed., "The Network Time Security (NTS) Protocol," RFC 8915, Internet Engineering Task Force, Sep. 2020.