International Journal of Applied Mathematics
Volume 38 No. 6s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

SECURE BOOT AND FIRMWARE PROTECTION IN AUTOMOTIVE SEMICONDUCTORS: A REVIEW OF CURRENT APPROACHES

Dr. Quazi Taif Sadat

Director

Bangladesh University

taif@bu.edu.bd

Abstract

The increasing reliance on software-driven vehicles has elevated the importance of firmware security in automotive semiconductors. Secure Boot is a foundational mechanism that ensures only authenticated firmware executes, forming a robust chain of trust from hardware to application code. This review paper provides a comprehensive analysis of Secure Boot architectures, hardware roots of trust, cryptographic enforcement, and secure firmware update mechanisms, with a focus on automotive electronic control units (ECUs). Key challenges—including post-quantum security, formal verification, and supply chain vulnerabilities—are discussed. Emerging research directions, such as control-flow integrity enforcement and multiprocessor ECU security frameworks, are highlighted to guide future development and standardization efforts.

Keywords: Secure Boot, firmware protection, automotive cybersecurity, ECU security, cryptography, hardware root of trust, OTA updates

1. Introduction

Modern vehicles integrate an extensive network of 50–100 electronic control units (ECUs), each responsible for managing safety-critical systems, including braking, steering, powertrain control, and advanced driver-assistance systems (ADAS). The firmware running on these ECUs dictates their operational integrity; any compromise can result in catastrophic safety failures or data breaches.

Secure Boot serves as the first line of defense by ensuring that only authenticated firmware executes during the ECU startup sequence. It relies on a hardware root of trust, implemented through components such as Hardware Security Modules (HSMs), Trusted Platform Modules (TPMs), or One-Time Programmable (OTP) memory. During the boot process, cryptographic signatures stored in these secure elements verify the authenticity of each software layer, thereby establishing a verified chain of trust from the hardware to the application code.

This paper examines the state-of-the-art approaches in Secure Boot implementation for automotive systems, highlighting both the technical mechanisms and research gaps that need addressing to improve overall vehicle cybersecurity.

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 6s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

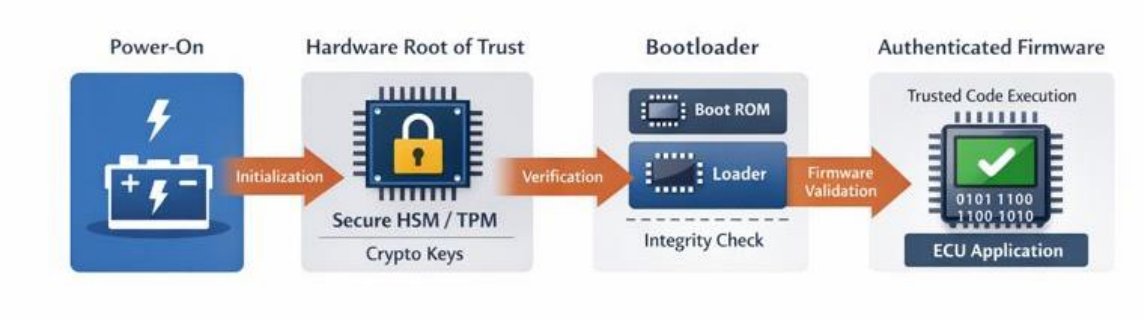

Figure 1. *Illustration of Secure Boot in Automotive ECU*

2. Background and Threat Landscape

Automotive firmware is increasingly targeted by sophisticated attacks due to the rising connectivity of vehicles. Threats include:

- **Malicious backdoors:** Unauthorized firmware insertion to manipulate vehicle behavior.
- **Firmware tampering:** Modification of operational code to bypass safety checks.
- **Rollback attacks:** Replacing current firmware with older, vulnerable versions.
- **Runtime tampering:** Interference during firmware execution, potentially bypassing security checks.

Secure Boot mitigates these risks by establishing a cryptographically verified chain of trust, beginning with immutable hardware elements and extending to bootloaders and main firmware. Regulatory frameworks, including ISO/SAE 21434 and UN R155, now mandate such mechanisms for vehicle cybersecurity compliance.

Table 1. *Common Automotive Firmware Threats and Mitigation via Secure Boot*

Threat Type	Description	Mitigation via Secure Boot
Malicious firmware injection	Unauthorized code execution	Signature verification
Rollback attack	Replacing firmware with older vulnerable version	Version checks, rollback protection
Supply-chain compromise	Tampering before deployment	Hardware root-of-trust, key management
Runtime tampering	Code alteration during execution	Control-flow integrity mechanisms

3. Literature Review

3.1 Secure Boot in Automotive ECUs

Sanwald et al. (2020) demonstrated that mismanaged keys in ECUs can undermine Secure Boot's effectiveness, emphasizing the need for stringent hardware root-of-trust and boot

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
sequence validation. Proper implementation ensures that unauthorized code cannot bypass the initial verification, even if attackers gain access to the ECU firmware storage.

Selvan (2025) explored cryptographic signature validation using RSA and ECC in embedded automotive systems, highlighting compliance with MISRA C and ISO 21434 standards as critical for secure deployment.

3.2 Cryptography and Future-Proofing

Emerging studies are evaluating the integration of post-quantum cryptography to future-proof Secure Boot against quantum-computing threats. TitanCFI, for example, introduces control-flow integrity enforcement at the root-of-trust, ensuring that runtime attacks are mitigated even after successful firmware verification.

3.3 Secure OTA Firmware Updates

Ahmed et al. (2025) presented an AUTOSAR-based FOTA framework enabling authenticated over-the-air updates. This framework ensures that firmware integrity is maintained throughout its lifecycle without requiring physical access, aligning with modern vehicle connectivity requirements.

Table 2. *Comparison of Prior Secure Boot Research in Automotive Systems*

Reference	Focus Area	Key Contribution	Limitations Identified
Sanwald et al. (2020)	Secure Boot implementation	Highlighted risks of improper key management and weak boot sequencing	Limited discussion on runtime protection
Zhang et al. (2022)	Multiprocessor Secure Boot	Formally verified authentication for multi-core ECUs	Increased design complexity
Selvan (2025)	Bootloader cryptography	Compared RSA and ECC for automotive ECUs; emphasized standards compliance	Did not address OTA integration
Parisi et al. (2024)	Runtime integrity	Introduced CFI enforcement at root-of-trust	Requires hardware support
Ahmed et al. (2025)	Secure OTA updates	AUTOSAR-based FOTA with boot-time verification	Focused mainly on update phase
WolfSSL (2025)	Industry compliance	Practical guidance on secure boot cryptography and deployment	Vendor-oriented perspective

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 6s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Secure Boot Architecture

Secure Boot typically implements a multi-stage verification process, ensuring authenticity and integrity at every step:

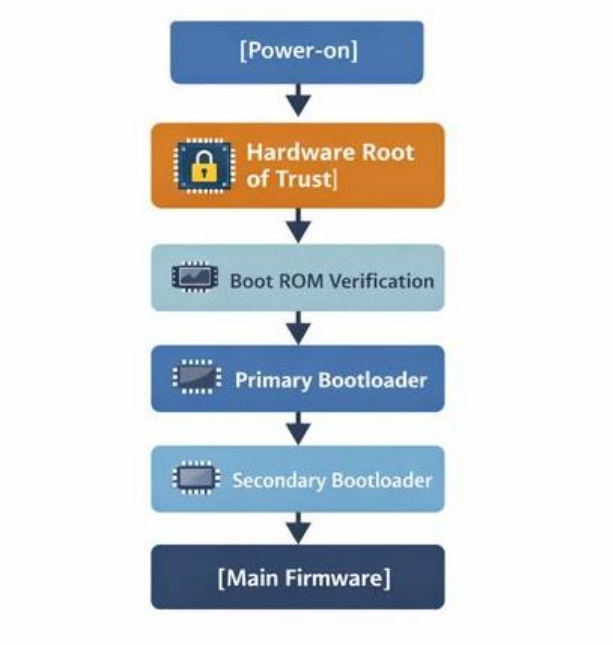

Figure 2. *Multi-stage Secure Boot Process in Automotive ECU*

Secure Boot typically implements a multi-stage verification process in which each boot phase authenticates the next before transferring execution control. The process begins with immutable Boot ROM code acting as the hardware root of trust, which uses cryptographic keys or key hashes securely stored in HSM, TPM, or OTP memory to verify the digital signature of the primary bootloader. Upon successful validation, the bootloader proceeds to authenticate subsequent stages, including secondary bootloaders and the main firmware image, thereby maintaining an unbroken chain of trust. Public-key cryptographic mechanisms such as RSA or ECC are commonly employed, often supported by hardware accelerators to reduce boot-time overhead. Any deviation, including invalid signatures, version mismatches, or evidence of code modification, causes the boot sequence to halt or enter a safe state, preventing unauthorized or unsafe firmware execution in automotive ECUs.

Table 3. *Comparison of Secure Boot Mechanisms*

Mechanism	**Description**	**Advantages**	**Challenges**
Hardware RoT	HSM/TPM keys	High trust	Cost, integration complexity
Bootloader signatures	Verify code authenticity	Prevents unauthorized execution	Performance overhead
Version control	Rollback protection	Blocks older vulnerable firmware	Key/version management

Received: August 03, 2025

Volume 38 No. 6s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Control-flow integrity	Runtime execution enforcement	Protects execution flow	Requires hardware support

5. Hardware Roots of Trust and Cryptography

HSMs provide secure key storage, cryptographic accelerators, and tamper-resistant memory. These elements serve as the foundation for the chain of trust and are embedded in automotive-grade microcontrollers to meet functional safety and cybersecurity requirements. Hardware RoTs are integral in protecting sensitive firmware operations and cryptographic key management from both physical and remote attacks.

6. Firmware Lifecycle Protection and OTA Updates

Secure Boot ensures authenticity at startup, but continuous protection is required throughout the ECU's lifecycle. Secure OTA frameworks verify signed firmware packages before installation, enforcing boot-time checks post-update.

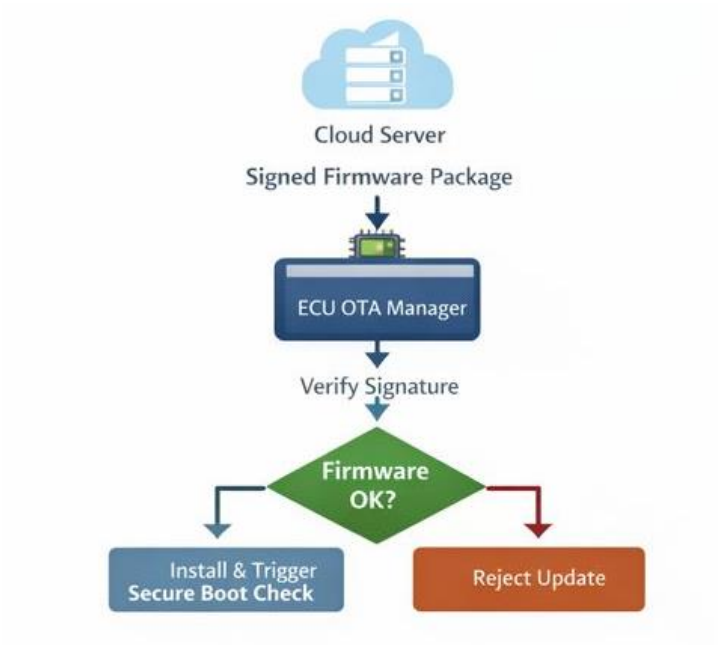

Figure 3. *Secure OTA Firmware Update Workflow*

Table 4. *Firmware Lifecycle Protection Mechanisms*

Stage	Mechanism	Benefit
Deployment	Secure Boot	Prevent initial tampering
Update	Signed OTA	Prevent unauthorized updates

Received: August 03, 2025

International Journal of Applied Mathematics
Volume 38 No. 6s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Operation	Runtime checks	Detect memory/code modifications
End-of-life	Secure decommission	Ensure data/firmware security

7. Discussion and Research Gaps

Despite robust implementations, several research gaps persist:

- Integration of **post-quantum cryptography** to protect long-lived vehicles.
- **Formal verification** of boot sequences and cryptographic chains.
- **Supply chain key management** to prevent pre-deployment compromise.
- **Runtime integrity enforcement** beyond boot, such as continuous monitoring of firmware execution.

Advanced authentication protocols and formally verified multiprocessor Secure Boot frameworks are actively being explored to address these gaps, ensuring consistency across OEMs and Tier-One suppliers.

8. Conclusion

Secure Boot and firmware protection are foundational to automotive cybersecurity. Through a combination of hardware roots of trust, cryptography, and secure OTA update frameworks, automotive ECUs maintain integrity and resist unauthorized modifications throughout the vehicle lifecycle. Future research must focus on post-quantum resilience, formal verification, and standardized practices to safeguard the next generation of software-defined vehicles.

References

[1] Ahmed, M. A. M., Elsayed, M. K. M., & Abdelmohsen, R. W. E. (2025). Enhancing AUTOSAR-based firmware over-the-air updates in the automotive industry. *arXiv*. https://arxiv.org/abs/2503.05839

[2] Sanwald, S., Kaneti, L., Stöttinger, M., & Böhner, M. (2020). Secure boot revisited: Challenges for secure implementations in the automotive domain. *SAE International*. https://www.autosar.org/fileadmin/standards/R24-11/FO/AUTOSAR_FO_EXP_SecurityOverview.pdf

[3] Selvan, A. M. (2025). Utilization of secure bootloaders in embedded systems. *Preprints.org*. https://www.preprints.org/manuscript/202511.1351/v1

[4] Parisi, E., et al. (2024). TitanCFI: Toward enforcing control-flow integrity in the root-of-trust. *arXiv*. https://arxiv.org/abs/2401.02567

[5] WolfSSL. (2025). Meeting secure boot compliance requirements. https://www.wolfssl.com/meeting-secure-boot-compliance-requirements/

[6] Synopsys. (2025). Automotive cybersecurity starts with chips. https://www.synopsys.com/articles/automotive-cybersecurity-starts-with-chips.html

Received: August 03, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
[7] Microchip Technology. (2025). What is secure boot. https://ww1.microchip.com/downloads/aemDocuments/documents/MCU32/ProductDocuments/SupportingCollateral/What-is-Secure-Boot-DS90003373.pdf
[8] Winbond. (2025). Meeting automotive security demands with TrustME W77T Secure Flash. https://www.winbond.com/hq/support/online-learning/articles-item/Meeting-Automotive-Security-Demands-with-TrustME-W77T-Secure-Flash
[9] ESET Research. (2025). Under the cloak of UEFI Secure Boot: CVE-2024-7344. https://www.welivesecurity.com/en/eset-research/under-cloak-uefi-secure-boot-introducing-cve-2024-7344/
[10] Zhang, Z., Yu, C., Chang, R., et al. (2022). PA-Boot: A formally verified authentication protocol for multiprocessor secure boot. *arXiv*. https://arxiv.org/abs/2209.07936
[11] ResearchGate. (2025). Secure boot implementation in automotive electronic control unit. https://www.researchgate.net/publication/387655779_Secure_Boot_Implementation_in_Automotive_Electronic_Control_Unit
[12] AUTOSAR. (2025). Security overview. https://www.autosar.org/fileadmin/standards/R24-11/FO/AUTOSAR_FO_EXP_SecurityOverview.pdf
[13] Kaushik, A. (2025). The role of secure boot and secure updates in automotive ECUs. *LinkedIn*. https://www.linkedin.com/pulse/role-secure-boot-updates-automotive-ecus-arpit-kaushik-om9pc
[14] DataIntelo. (2025). Secure boot and firmware signing for ITS market report. https://dataintelo.com/report/secure-boot-and-firmware-signing-for-its-market
[15] Wikipedia. (2025). CAN bus. https://en.wikipedia.org/wiki/CAN_bus

Received: August 03, 2025