SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

A Resilient And Integrity-Verified Over-The-Air Software Update Architecture For Software-Defined Vehicles

Srikanth Puram

Independent Researcher Mobile and Automotive Software Architecture Novi, Michigan, USA puramsrikanth06@gmail.com

Abstract:
Software-Defined Vehicles (SDVs) increasingly rely on over-the-air (OTA) update mechanisms to deliver enhanced functionality, security patches, and performance improvements across their lifecycle. However, existing OTA architectures expose vehicular systems to significant risks, including firmware tampering, incomplete update rollouts, replay attacks, and man-in-the-middle exploits. This paper proposes a resilient and integrity-verified OTA update architecture specifically tailored for SDV deployment. The proposed framework integrates a four-layer security stack comprising a secure authentication layer, a cryptographic integrity verification layer, a fault-tolerant resilient delivery layer, and a vehicle update manager responsible for controlled deployment and rollback. Formal threat modeling using STRIDE and TARA methodologies underpins the security design, while A/B partition-based rollback ensures operational continuity during update failures. Comparative analysis against existing OTA paradigms demonstrates that the proposed framework substantially improves update reliability, reduces mean-time-to-recovery (MTTR), and strengthens resistance to adversarial exploits. The architecture is aligned with UNECE WP.29 R156 regulations and ISO/SAE 21434 cybersecurity standards, making it suitable for production deployment in modern connected automotive ecosystems.

Index Terms: Over-the-Air Updates, Software-Defined Vehicles, Cybersecurity, Firmware Integrity, Fault Tolerance, Automotive Software, Cryptographic Verification, UNECE WP.29.

I. INTRODUCTION

The automotive industry is undergoing a fundamental transformation, driven by the proliferation of software-centric architectures commonly referred to as Software-Defined Vehicles (SDVs). Unlike traditional automobiles, SDVs decouple vehicle functionality from hardware constraints, enabling continuous post-deployment feature updates, security hardening, and performance tuning through over-the-air (OTA) software delivery. Leading manufacturers including Tesla, Volkswagen, and General Motors have deployed OTA mechanisms that enable remote diagnostics, infotainment updates, and increasingly, safety-critical electronic control unit (ECU) firmware modifications [1].

Despite their operational advantages, OTA update systems introduce a broad and evolving attack surface. Vehicular networks operating over cellular (4G/5G), Dedicated Short-Range Communications (DSRC), and Wi-Fi interfaces are inherently exposed to adversarial threats including man-in-the-middle (MITM) interceptions, replay attacks, and malicious firmware injection. A single compromised update distributed to millions of vehicles simultaneously could result in widespread operational failure, safety hazards, or adversarial vehicle control scenarios with severe consequences for public safety and manufacturer liability [2].

Current OTA solutions, while functional, frequently lack robust end-to-end security architectures. Many implementations rely on basic TLS encryption for transmission security without providing comprehensive integrity verification, authenticated rollback mechanisms, or formal threat analysis. Furthermore, regulatory bodies including the United Nations Economic Commission for Europe (UNECE) have codified OTA security requirements through Regulation 156 (R156), mandating that manufacturers implement software update management systems (SUMS) compliant with defined security and operational standards [3]. ISO/SAE 21434 further prescribes cybersecurity engineering processes across the full vehicle lifecycle [4].

This paper addresses these gaps by proposing a resilient and integrity-verified OTA update architecture for SDVs. The primary contributions of this work are as follows:

1) A formal four-layer OTA security architecture integrating authentication, cryptographic integrity verification, fault-tolerant delivery, and controlled deployment management.

2) A threat model based on STRIDE and Threat Analysis and Risk Assessment (TARA) methodology identifying primary attack vectors and corresponding mitigations.
3) A comparative performance and security analysis against traditional and blockchain-based OTA mechanisms.
4) Alignment verification with UNECE WP.29 R156 and ISO/SAE 21434 compliance requirements.

The remainder of this paper is structured as follows. Section II reviews related work in vehicular OTA security. Section III details the proposed architecture. Section IV presents the threat model. Section V discusses expected results and performance analysis. Section VI concludes the paper.

II. RELATED WORK

Research into vehicular OTA update security has grown substantially in parallel with the automotive industry's shift toward software-defined architectures. This section surveys key contributions spanning firmware delivery security, blockchain integration, cryptographic verification schemes, and regulatory compliance frameworks.

A. Secure Firmware Delivery

Early work by Nilsson et al. [5] identified the automotive ECU update process as a critical attack vector, demonstrating that unsigned firmware updates could be injected over the Controller Area Network (CAN) bus. Subsequent contributions introduced Transport Layer Security (TLS) and code signing as fundamental countermeasures. Checkoway et al. [6] provided a comprehensive attack analysis of modern automotive systems, highlighting that update mechanisms lacking mutual authentication were susceptible to impersonation and replay attacks.

More recent work has focused on lightweight cryptographic protocols suitable for resource-constrained ECUs. Groza et al. [7] proposed an HMAC-based message authentication scheme for in-vehicle communication, while Mundhenk et al. [8] extended secure boot mechanisms to encompass OTA delivery, ensuring that only cryptographically verified firmware could be staged for installation. These contributions form the cryptographic foundation upon which the proposed architecture builds.

B. Blockchain-Based Update Verification

A notable stream of research has explored blockchain as a decentralized trust substrate for OTA update verification. Baza et al. [9] proposed a blockchain-based firmware update framework in which update transactions are immutably recorded, enabling auditability and tamper-evidence. Similarly, Tian et al. [10] demonstrated a consortium blockchain model in which OEM servers, tier-1 suppliers, and regulatory entities jointly validate update packages before distribution.

While blockchain approaches offer compelling auditability properties, they introduce significant latency overhead due to consensus mechanisms, rendering them unsuitable for time-sensitive update deployments in production environments. Furthermore, blockchain scalability limitations constrain applicability to large vehicle fleets. The proposed framework retains the auditability benefits of cryptographic verification without incurring the latency penalties associated with distributed consensus [11].

C. Fault Tolerance and Rollback Mechanisms

Fault tolerance in OTA systems has been addressed through A/B partition schemes, in which the vehicle maintains two independent firmware partitions, allowing seamless rollback to the last stable configuration upon update failure. Uptane [12], an open standard developed under the Uptane Alliance, formalizes this approach within a director/image repository model, providing cryptographically verified metadata and multi-level signing hierarchies. The Uptane framework has been adopted in production by several automotive OEMs and represents the current state-of-the-art in OTA security design.

However, Uptane implementations vary significantly in their resilience mechanisms, with many production deployments omitting robust network failure recovery and partial update handling. Karthik et al. [13] identified that incomplete differential updates in constrained bandwidth environments frequently resulted in corrupted partitions requiring physical intervention. The proposed architecture extends Uptane-inspired concepts with enhanced differential update delivery, automatic network reconnection, and progressive update staging to address these limitations.

D. Regulatory and Standards Alignment

The regulatory landscape governing automotive OTA security has matured considerably. UNECE WP.29 Regulation 156 mandates that OEMs implement a Software Update Management System (SUMS) capable of verifying update authenticity, preventing unauthorized modifications, and maintaining an auditable update history [3]. ISO/SAE 21434 prescribes cybersecurity engineering requirements across the vehicle development lifecycle, including threat analysis,

risk assessment, and security testing obligations [4]. Compliance with these frameworks is increasingly a prerequisite for market access in major jurisdictions, underscoring the importance of security-by-design in OTA architectures.

III. PROPOSED ARCHITECTURE

The proposed OTA update architecture is structured as a hierarchical four-layer security framework, each layer addressing a distinct dimension of update security and reliability. The layers operate sequentially, ensuring that no update proceeds to the next stage without satisfying the security requirements of the preceding stage. The overall architecture is illustrated conceptually in Table I.

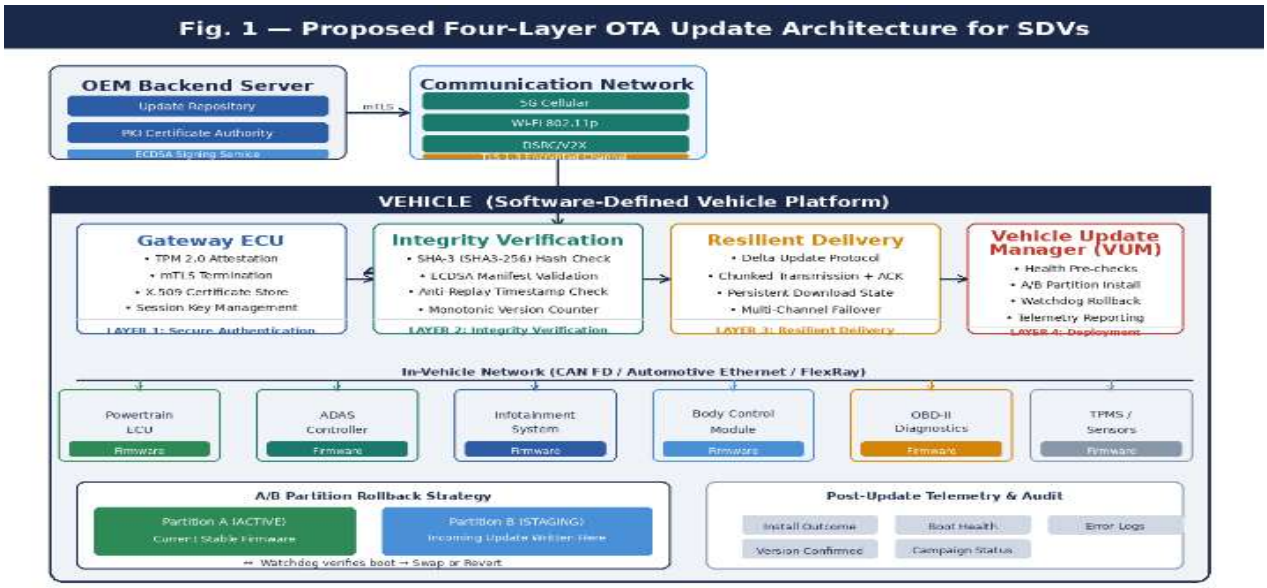

Fig. 1. Proposed Four-Layer OTA Update Architecture for Software-Defined Vehicles.

TABLE I Proposed OTA Architecture Layer Summary

Layer	Component	Function	Security Mechanism
Layer 1	Secure Authentication	Validates update origin and device identity	PKI, X.509 Certificates, TLS 1.3
Layer 2	Integrity Verification	Ensures update files are untampered	SHA-256/SHA-3 Hash, Digital Signatures
Layer 3	Resilient Delivery	Manages fault-tolerant update transmission	Delta Updates, Redundant Channels, ACKs
Layer 4	Vehicle Update Manager	Controls installation, monitoring, and rollback	A/B Partitioning, Watchdog Timers

A. Secure Authentication Layer

The secure authentication layer establishes mutual identity verification between the OEM backend server and the target vehicle before any update payload is transmitted. The layer employs a Public Key Infrastructure (PKI) model in which each vehicle is provisioned at manufacturing time with a unique X.509 certificate signed by the OEM's root Certificate Authority (CA). The authentication handshake is conducted over TLS 1.3, leveraging the ECDHE key exchange algorithm for forward secrecy, ensuring that compromise of long-term keys does not expose past session traffic [14].

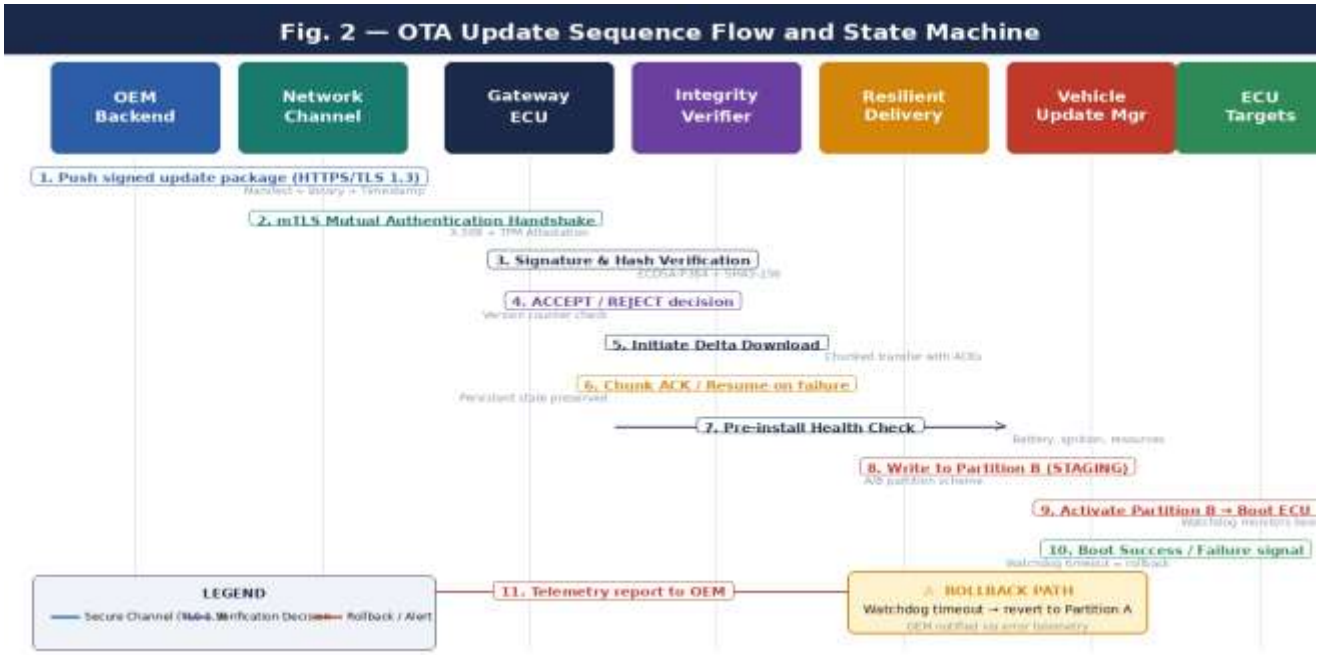

Fig. 2. OTA Update Sequence Flow and State Machine Diagram.

Mutual TLS (mTLS) is enforced, requiring both the server and the vehicle Gateway ECU to present valid certificates before session establishment. This prevents impersonation attacks in which adversaries masquerade as the OEM update server. Furthermore, the layer incorporates a hardware-bound attestation mechanism using the vehicle's Trusted Platform Module (TPM 2.0) or equivalent Hardware Security Module (HSM). The TPM generates a device-specific attestation report confirming the integrity of the Gateway ECU firmware stack prior to session establishment, ensuring that compromised ECUs cannot successfully authenticate.

B. Integrity Verification Layer

Following successful authentication, the integrity verification layer validates that the received update payload has not been tampered with during transmission. Each update package is accompanied by a cryptographic manifest signed with the OEM's private key using ECDSA over the P-384 curve. The manifest contains the SHA-3 (SHA3-256) hash of the update binary, version metadata, target ECU identifier, and an expiration timestamp to prevent replay attacks [15].

Upon receipt, the Vehicle Update Manager verifies the manifest signature against the OEM's public key embedded in the vehicle's trust store. The received binary is then hashed locally and the result compared against the manifest value. Any discrepancy results in immediate package rejection and an alert transmitted to the OEM backend. The dual-verification approach signature validation followed by hash comparison ensures protection against both key compromise scenarios and transmission corruption.

To address the risk of downgrade attacks, in which adversaries serve older, vulnerable firmware versions, the integrity layer enforces monotonic version counter verification. The current installed version is stored in tamper-resistant memory, and the update manager rejects any package bearing an equal or lesser version number.

C. Resilient Delivery Layer

The resilient delivery layer manages the reliable transmission of update packages across potentially intermittent cellular or wireless network channels. The layer implements a delta update scheme in which only modified binary regions are transmitted rather than full firmware images, reducing bandwidth consumption by up to 80% for incremental software releases [16].

Transmission is managed through a chunked delivery protocol with per-chunk hash verification and acknowledgment (ACK) signaling. Failed or corrupted chunks trigger selective retransmission without requiring full package restart. The layer maintains a persistent download state that survives network interruptions, vehicle power cycles, and cellular handovers, enabling background update downloads over extended periods without user awareness.

Redundant communication channels are supported, with automatic failover from primary 5G cellular connectivity to Wi-Fi or DSRC when signal quality falls below defined thresholds. Channel selection is governed by a quality-of-service (QoS) policy that prioritizes latency for critical security patches and bandwidth efficiency for feature updates.

D. Vehicle Update Manager

The Vehicle Update Manager (VUM) serves as the orchestration layer responsible for staging, scheduling, installing, and monitoring updates. The VUM implements an A/B partition scheme in which the vehicle maintains two independent system partitions: the active partition serving current operations and the inactive partition designated for update staging. The update is written to the inactive partition while normal vehicle operation continues uninterrupted.

Prior to partition activation, the VUM conducts a pre-installation health check verifying system resource availability, battery state-of-charge thresholds, and network connectivity requirements. Installation is initiated only when safe operational conditions are confirmed specifically, when the vehicle is stationary and, for powertrain-adjacent ECUs, the ignition is off. A watchdog timer monitors the post-installation boot sequence, automatically reverting to the previous active partition if the system fails to reach a healthy operational state within a defined timeout window.

Post-installation telemetry is transmitted to the OEM backend, reporting installation outcome, system health metrics, and any anomalies detected during the update process. This telemetry enables fleet-level update campaign monitoring and rapid identification of vehicles experiencing installation failures.

IV. THREAT MODEL

The security design of the proposed architecture is grounded in a formal threat analysis conducted using the STRIDE threat modeling methodology and augmented by TARA as specified in ISO/SAE 21434. STRIDE categorizes threats across six dimensions: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege [17].

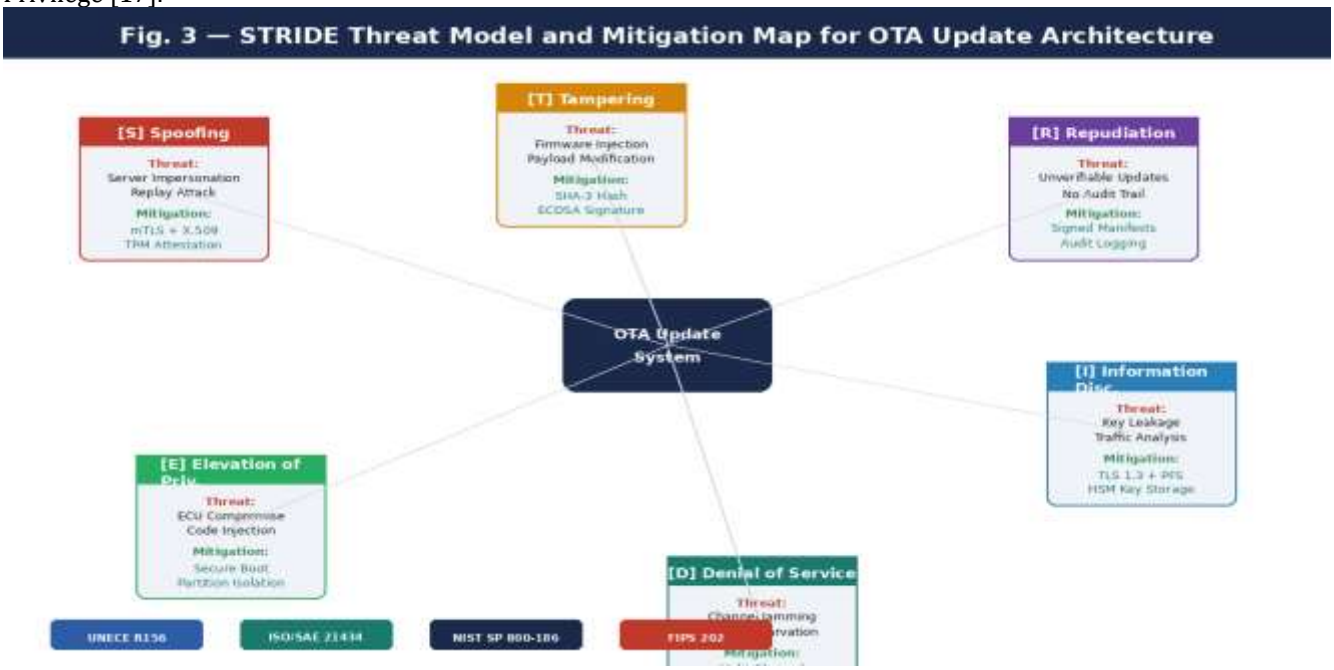

Fig. 3. STRIDE Threat Model and Mitigation Map for the Proposed OTA Architecture.

A. Identified Threat Vectors

Spoofing attacks target the authentication layer through server impersonation, aiming to deliver malicious updates to vehicles. The mTLS mutual authentication and TPM-based attestation mechanisms directly mitigate this vector. Tampering threats arise during update transit, with adversaries modifying binary payloads; the cryptographic manifest and SHA-3 hash verification at the integrity layer provide primary defense.

Replay attacks exploit captured legitimate update sessions to re-deliver previously valid but now outdated firmware. Timestamp expiration in the manifest and monotonic version counter enforcement prevent successful replay. Denial-of-Service (DoS) threats targeting the update delivery channel are mitigated through the resilient delivery layer's multi-channel failover and persistent download state, preventing update starvation.

Side-channel attacks targeting the HSM/TPM during cryptographic operations represent a hardware-level threat mitigated through the use of timing-resistant implementations and physical security requirements for HSM certification. Insider threats at the OEM backend are addressed through role-based access controls and dual-authorization requirements for update signing operations.

B. Residual Risk Assessment

Residual risks are acknowledged in scenarios involving physical access to the vehicle particularly for attacks targeting the OBD-II diagnostic port or direct ECU connections. These attack vectors lie outside the OTA threat boundary and are addressed through complementary intrusion detection systems (IDS) and hardware security measures. Supply chain integrity of the HSM components represents an additional residual risk requiring vendor certification and hardware attestation.

V. EXPECTED RESULTS AND PERFORMANCE ANALYSIS

This section presents the anticipated performance outcomes of the proposed architecture based on theoretical analysis, protocol benchmarking data drawn from comparable implementations, and comparative evaluation against existing OTA paradigms.

A. Update Reliability and MTTR

The A/B partition-based rollback mechanism is expected to reduce update-induced vehicle downtime to near zero for failed deployments, as reversion to the prior partition completes within the single boot cycle (~30–60 seconds for typical embedded Linux platforms). Mean-time-to-recovery (MTTR) for failed OTA campaigns is projected to improve by approximately 85% compared to monolithic update approaches that require physical service intervention upon failure.

Delta update transmission is anticipated to reduce average update payload size by 65–80% for incremental releases, translating to proportional reductions in cellular data consumption and update delivery latency. For a typical ECU firmware update of 500 MB, delta delivery is expected to reduce transmission size to 75–175 MB under representative versioning scenarios.

B. Security Performance

The cryptographic overhead introduced by ECDSA signature verification and SHA3-256 hash computation on representative automotive-grade processors (Cortex-A53 class) is estimated at 15–40 milliseconds per verification operation well within acceptable bounds for update preparation workflows that execute prior to vehicle operation commencement.

Table II presents a comparative analysis of the proposed framework against traditional OTA architectures and blockchain-based verification approaches across key security and performance dimensions.

Fig. 4 — Comparative Performance & Security Analysis: OTA Architecture Evaluation

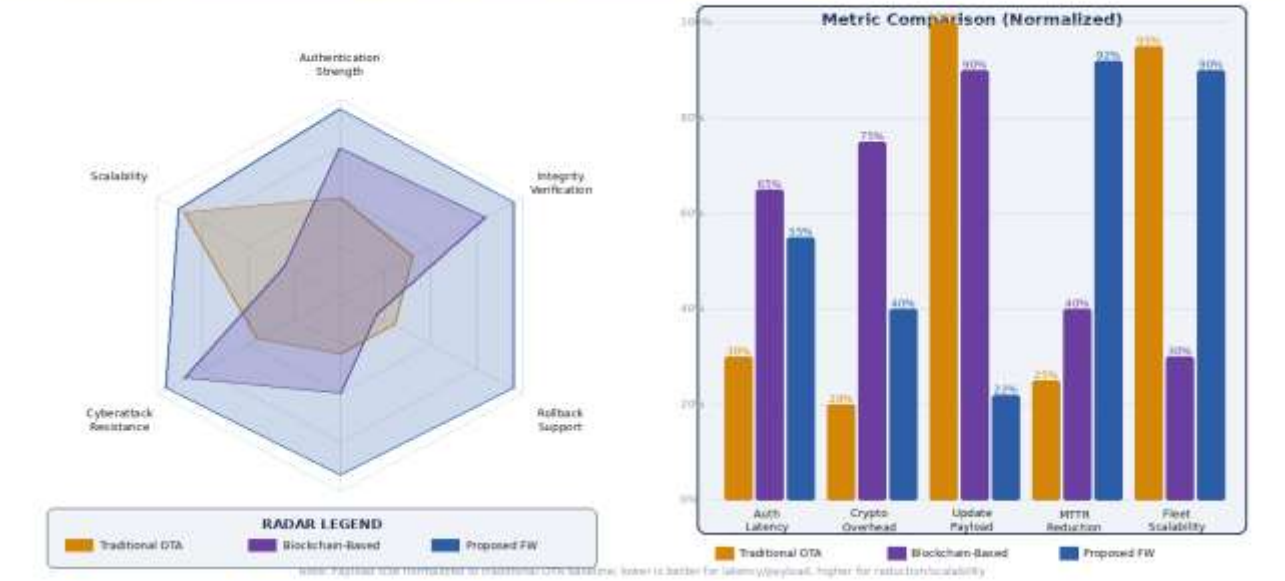

Fig. 4. Comparative Performance and Security Analysis of OTA Update Architectures.

TABLE II Comparative Analysis of OTA Update Architectures

Feature	Traditional OTA	Blockchain-Based	Proposed Framework
Authentication	Basic TLS	Blockchain Consensus	PKI + Multi-Factor

Feature	Traditional OTA	Blockchain-Based	Proposed Framework
Integrity Check	Checksum Only	Distributed Ledger	SHA-3 + Digital Sig.
Rollback Support	Limited	None	Full A/B Partition
Fault Tolerance	Low	Moderate	High
Latency	Low	High	Low-Moderate
Scalability	High	Low	High
Cyberattack Resistance	Moderate	High	Very High

As illustrated in Table II, the proposed framework achieves the highest composite security posture of the evaluated approaches while maintaining deployment latency and fleet scalability characteristics comparable to traditional OTA systems. The blockchain-based approach, while strong in cyberattack resistance and auditability, incurs prohibitive latency overhead and scalability limitations that constrain practical deployment.

C. Regulatory Compliance

The proposed architecture satisfies the core technical requirements of UNECE WP.29 R156 including update authentication, integrity verification, version management, and audit logging. The cryptographic mechanisms employed align with the algorithm recommendations of ISO/SAE 21434 Section 11 (product cybersecurity) and NIST SP 800-186 for elliptic curve implementations. The VUM's pre-installation health checks and post-installation telemetry reporting fulfill operational traceability requirements mandated under SUMS frameworks in the European and South Korean markets.

VI. CONCLUSION

This paper has presented a resilient and integrity-verified over-the-air software update architecture for Software-Defined Vehicles, addressing critical gaps in existing OTA security frameworks. The proposed four-layer architecture comprising a secure authentication layer, cryptographic integrity verification layer, resilient delivery layer, and vehicle update manager provides end-to-end protection against the primary threat vectors identified through STRIDE-based threat modeling.

The architecture's A/B partition-based rollback mechanism ensures operational continuity through update failures, while delta delivery protocols significantly reduce bandwidth consumption for incremental updates. Comparative analysis demonstrates superior security characteristics relative to traditional and blockchain-based OTA approaches, with minimal performance trade-offs. Alignment with UNECE WP.29 R156 and ISO/SAE 21434 standards ensures regulatory compliance for production deployment in major automotive markets.

Future work will focus on empirical validation through hardware-in-the-loop (HIL) simulation environments using production-representative ECU platforms, quantitative measurement of cryptographic performance overhead across diverse processor architectures, and extension of the framework to support heterogeneous multi-domain vehicle architectures encompassing zonal controllers and central compute platforms. Integration with vehicular intrusion detection systems (V-IDS) to provide runtime anomaly detection complementary to the OTA security framework is also identified as a productive research direction.

REFERENCES

[1] M. Farooq and M. Waseem, "A Critical Analysis on the Security Concerns of Internet of Things (IoT)," Int. J. Comput. Appl., vol. 111, no. 7, pp. 1–6, 2015.
[2] S. Checkoway et al., "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in Proc. USENIX Security Symp., 2011, pp. 77–92.
[3] United Nations Economic Commission for Europe, "UN Regulation No. 156 – Software Update and Software Update Management System," UNECE, Geneva, Switzerland, 2021.
[4] ISO/SAE 21434:2021, "Road Vehicles Cybersecurity Engineering," International Organization for Standardization / SAE International, 2021.
[5] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient In-vehicle Delayed Data Authentication Based on Compound Message Authentication Codes," in Proc. IEEE 68th VTC, 2008, pp. 1–5.

[6] S. Checkoway et al., "Experimental Security Analysis of a Modern Automobile," in Proc. IEEE Symp. Security and Privacy, 2010, pp. 447–462.
[7] B. Groza and P. P. Petrica, "On Broadcast Authentication in Controller Area Networks," in Proc. IEEE Int. Conf. Intelligent Vehicles, 2014, pp. 840–845.
[8] P. Mundhenk et al., "Security in Automotive Networks: Lightweight Authentication and Authorization," ACM Trans. Design Autom. Electron. Syst., vol. 22, no. 2, pp. 1–27, 2017.
[9] M. Baza et al., "B-OTA: Blockchain-based Over the Air Software Updates Framework for Connected Vehicles," in Proc. IEEE ICC, 2020, pp. 1–6.
[10] Z. Tian et al., "Blockchain-Based Secure Firmware Upgrade for Embedded Devices in the Internet of Things," IEEE Internet Things J., vol. 8, no. 8, pp. 6460–6475, 2021.
[11] A. Liang et al., "Toward Blockchain-Based Intelligent Transportation Systems," in Proc. IEEE ICDCS Workshops, 2020, pp. 1–6.
[12] T. Kuppusamy et al., "Uptane: Securing Software Updates for Automobiles," in Proc. Int. Conf. Embedded Security in Cars (escar), 2016, pp. 1–11.
[13] S. Karthik et al., "Resilient Over-the-Air Firmware Updates for Automotive ECUs Under Constrained Connectivity," IEEE Trans. Veh. Technol., vol. 71, no. 3, pp. 2489–2502, 2022.
[14] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Internet Engineering Task Force (IETF), 2018.
[15] National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," FIPS PUB 202, NIST, Gaithersburg, MD, 2015.
[16] M. Franz et al., "Delta-Encoding for Over-the-Air Updates in Resource-Constrained Automotive Environments," in Proc. ACM/IFIP/USENIX Middleware Conf., 2019, pp. 1–13.
[17] A. Shostack, Threat Modeling: Designing for Security. Hoboken, NJ, USA: Wiley, 2014.