# Securing OTA Updates: Addressing Integrity Challenges in Autonomous Vehicle Software

Tausif Zaman
Masters of Applied Computing Student, University of Windsor

## I. Introduction

The advent of smart vehicles has revolutionized the automotive industry by integrating cutting-edge technologies such as sensors, cameras, radar, and internet-connected data models. These innovations enable vehicles to perform complex operations autonomously, significantly reducing the need for human intervention [1]. Central to this functionality is software, which processes sensor data to ensure precision and coordination.

Over-the-Air (OTA) updates provide a seamless, wireless method to deliver critical software enhancements to these vehicles. These updates include performance improvements, feature refinements, and vulnerability patches, playing a vital role in maintaining the security and reliability of smart vehicle systems. By eliminating the need for physical intervention, OTA updates ensure that vehicles remain secure against emerging cyber threats and compliant with evolving industry standards, underscoring their importance in modern automotive technology.

The shift away from human intervention in delivering software updates has streamlined processes but also introduced significant cybersecurity challenges. Additionally, there is a prevalent lack of physical monitoring when it comes to OTA updates, whereas more traditional maintenance of vehicles strictly implemented physical hands-on labor. This highlights the importance of cybersecurity standards like ISO 26262 and SAE J3061. These standards specifically aim to maintain assurance cases to address vulnerabilities present in the OTA mechanism, thereby ensuring both functional safety and data integrity [2].

In the autonomous vehicle (AV) industry, cybersecurity is paramount, as compromises can lead to malicious attacks, data breaches, or system failures, posing serious risks to user safety and lives. Beyond these immediate dangers, such vulnerabilities can damage a company's reputation and compromise public trust in AV technologies.

This report focuses on the integrity of OTA updates, a critical component of cybersecurity aimed at ensuring the accuracy, consistency, and trustworthiness of data throughout an AV's software lifecycle. Maintaining integrity is essential to prevent tampering or corruption during transmission. We will explore the challenges, tools, and techniques—such as hashing and digital signatures—that verify and safeguard data integrity. Furthermore, this report will examine how to protect vehicles from compromised updates and highlight the risks associated with inadequate integrity checks.

## II. Importance of OTA Integrity Assurance

The integrity of Over-the-Air (OTA) updates in autonomous vehicles (AVs) is a crucial challenge within the automotive industry. OTA assurance is vital for preserving data integrity, service integrity, and cybersecurity across wireless networks. Without robust integrity checks, the risks include:

- **Malicious Updates:** Attackers can exploit vulnerabilities to introduce tampered updates, potentially enabling vehicle control and endangering lives.
- **Data Breaches:** Unauthorized access to critical update files can expose sensitive information or lead to ransomware attacks.
- **System Failures:** Faulty or malicious updates may disrupt operations, posing immediate threats to user safety.

Examples like the Jeep Cherokee hack of 2015 [3] illustrate how compromised OTA updates can allow attackers to manipulate vehicle systems remotely, highlighting the dire consequences of inadequate assurance mechanisms. The CIA triad framework, where integrity is assured during the process of data transmission [1] (data remains unaltered throughout) during the OTA process aligns with this. Hashing and digital signatures work to validate the integrity of these updates, and features such as anomaly detection help discern abnormal update behaviors. This underscores the importance of implementing robust measures to safeguard the integrity and reliability of AV software against emerging threats. The Uptane framework highlights how unsigned plaintext communication can sometimes result in Man-in-the-Middle (MIMT) attacks [2]. Packets of numbers used only once, when signed, can mitigate this vulnerability, showcasing the importance of such measures when attempting to enhance OTA integrity assurance.

## III. Challenges of Integrity Checks

### A. Data Integrity

This refers to preserving the original state of update files as developed by the software provider. Attackers often exploit vulnerabilities in wireless networks, such as LTE or Wi-Fi, to manipulate the update process. For instance, data integrity can be compromised if cloud data is modified by an attacker, potentially enabling vehicle control and endangering lives [1]. According to the STRIDE threat model [2], tampering is one of the more abundant and critical threats to OTA updates. Techniques such as hashing, cryptographic signatures, and cyclic redundancy checks (CRC) are commonly used to detect and prevent tampering, ensuring the update file remains

unchanged from source to destination. A specific case study has demonstrated how unsigned communications in Uptane allowed tampering of transmitted OTA updates through MITM attacks, thus highlighting the importance of signing packets for data integrity assurance.

### B. Service Integrity

Service integrity ensures that processes like access, download, and installation of updates are free from malicious interference. Threats include:

- **Man-in-the-Middle (MITM) Attacks:** Attackers can eavesdrop, manipulate, or impersonate entities during the update process.
- **Spoofing and Sinkhole Attacks:** Legitimate traffic may be redirected to malicious servers, disrupting communications and potentially denying critical services.

One more key mechanism that helps maintain service integrity is Version Control. It prevents the installation of outdated and unauthorized software versions that would compromise system functionality. Unintended bugs and issues end up never being deployed since version control (VC) allows for a robust way of tracking changes made to the software. Additionally, it allows easy rollbacks to previous versions if a new update ever causes issues, thus minimizing downtime.

### C. Security Requirements

Private key protection is an important part of cryptographic practices, ensuring attackers cannot extract keys to tamper with the update files. Preventing unauthorized modifications during OTA updates is achieved by securing these keys. To address these challenges, manufacturers implement requirements such as:

- **Data Integrity Requirements (DIR):** Techniques like hashing, digital signatures, and cryptographic encryption verify the authenticity and integrity of update files.
- **Service Integrity Requirements:** Secure transmission channels, proper update storage, and frameworks like blockchain ensure end-to-end transparency and resilience.

## IV. Tools and Mechanisms for Integrity Checks

### A. Hashing and Digital Signatures

Hashing generates a unique, fixed-length hash value for the update file. Recipients verify data integrity by recalculating the hash and comparing it to the original. A mismatch signals tampering. Anomaly detection tools (such as CrunchMetrics, utilizing statistical methods and AI-ML techniques to identify critical incidents in real-time, making it suitable for monitoring OTA updates) can be implemented alongside hashing and digital signatures [1]. Such tools will enhance integrity by flagging unusual behaviors spotted during the update process. Digital signatures extend this by using cryptographic keys to sign data, ensuring authenticity and integrity.

### B. Encryption Techniques

- **Symmetric Encryption:** A single key encrypts and decrypts data, ensuring only authorized parties have access.
- **Asymmetric Encryption:** A public-private key pair enhances security by encrypting data with one key and decrypting it with the other.

### C. Blockchain Technology

Blockchain's decentralized architecture validates transactions using consensus algorithms like Proof of Work (PoW) or Proof of Stake (PoS). This makes tampering costly and impractical, ensuring that critical records remain immutable. However, scalability and latency pose challenges for real-time AV systems.

### D. Emerging Frameworks

- **SecUp:** Uses secure hashes and NFC-based authentication to verify updates.
- **ARA::UCM:** Filters malicious data and validates updates for safety.
- **Uptane with TUF Core:** Prevents rollback attacks by synchronizing metadata and maintaining update order.

An ACT-based framework was proposed which mixes functional safety with cybersecurity measures to address OTA integrity challenges [2]. Integrating multiple levels of security with frameworks like Uptane, manufacturers are able to guarantee better safeguards against rollback/tampering attacks.

### E. Hardware Security Platforms

- **Hardware Security Modules (HSMs):** Secure cryptographic key management.
- **Trusted Platform Modules (TPMs):** Validate system integrity at startup, ensuring malicious software cannot load during boot.

These security platforms are able to provide a secure environment for private key management, enabling cryptographic operations to remain tamper-resistant.

## V. Case Studies on OTA Updates

### A. Case Study 1: Tesla Autopilot Update Failure

In January 2024, Tesla pushed out an OTA update that was planned to improve and tighten the driver monitoring system for their Autopilot feature. This was in the works after an investigation by the National Highway Traffic Safety Administration (NHTSA) that identified more than a dozen crashes of Tesla vehicles due to defective systems resulting in serious injuries and fatalities [4]. The core of the issue stemmed from Tesla's $8,000 add-on Full-Self Driving (FSD) feature [5]. This update for the FSD was meant to increase restrictions and therefore improve safety, but the update did not serve its intended purpose, highlighting an OTA integrity issue. Tesla owners started reporting widespread software failures, some of the worst being disabled driver-assist functions, while others could not even finish downloading the update.

*1) Issues and Consequences:*

- Owners were not able to download the update completely, with the vehicle stuck in a download loop where it would be stuck at 100%, requiring the user to start again after reaching that point. The repeated auto-download attempts resulted in excessive battery drainage.
- Cases of Autopilot and FSD being disabled after the update were reported, a critical safety issue, rendering the vehicles unsafe to use while on that patch.
- The update left several users perplexed, as they were unsure how faulty the software could have been pushed out in the first place, leading some to mistakenly assume the issue was hardware-related.

This case study exemplifies the critical need for robust integrity checks to prevent broken updates and highlights the importance of transparent communication with users afterward to facilitate prompt troubleshooting.

### *B. Case Study 2: GM's Software Issue Misdiagnosis*

Ontario resident Charles Jakl, owner of a Chevrolet Volt Hybrid, suddenly faced a severe issue with his autonomous vehicle. Right after his warranty period expired, the vehicle stopped working [6]. GM's initial diagnosis indicated that a battery replacement was necessary, which would cost over $33,000—far more than the vehicle's value. The owner, contemplating the high cost, deemed the repair not worth it, and the car sat idle for months. Afterward, GM conducted a deeper investigation and discovered the issue could be fixed with a simple software update.

This case study demonstrates the other side of the coin. Due to the thorough investigation and collaboration, they were able to address a problem that was fixable via a software update without the need for an expensive hardware replacement. This highlights how modern AVs are increasingly managing issues via software updates, signaling a shift from traditional hardware maintenance to more rapid software fixes. With the increasing adoption of electric vehicles, the demand for OTA updates is seeing a massive upward trend. The Automotive Over-The-Air Updates Market Size was valued at USD 4.35 billion in 2023 and is expected to reach USD 19.29 billion by 2032 [7].

| Approach | Strengths | Weaknesses | Use Case |
|---|---|---|---|
| Hashing (e.g., SHA-256) | Fast and efficient for integrity verification. Detects any tampering with data. | Vulnerable to brute force attacks if used without additional security measures | Small-scale integrity verification, ensuring OTA updates remain unchanged. |
| Digital Signatures | Provides high authenticity and non-repudiation. Ensures data origin verification. | Computationally expensive, especially for large-scale deployments. | Securing critical OTA updates and verifying sender authenticity. |
| Microsoft STRIDE Framework | Comprehensive threat analysis. Identifies and categorizes threats like tampering. | Requires deep technical expertise for implementation. Can be resource intensive. | Developing threat models for OTA integrity. |
| Blockchain Technology | Immutable transaction records. Decentralized and tamper resistant. | High latency and scalability challenges for real-time updates. | Verifying OTA update authenticity in supply chain management. |
| Uptane Framework | Designed specifically for automotive OTA security. Mitigates rollback attacks. | Vulnerable to MITM attacks if not fully implemented. Complex to integrate with legacy systems. | Ensuring safe and ordered OTA updates in AV systems. |
| Private Key Protection | Secures cryptographic keys from being extracted. Reduces tampering risks. | Requires additional hardware (e.g., HSM) or secure storage solutions. | Protecting OTA update mechanisms from unauthorized access. |
| Version Control | Prevents outdated or unauthorized updates. Streamlines software maintenance workflows. | Relies on proper configuration and continuous monitoring. | Ensuring only approved versions are deployed via OTA. |

TABLE I
COMPARATIVE ANALYSIS OF APPROACHES

## 7. INSIGHTS AND CONCLUSION

This report emphasizes the critical role of integrity in OTA updates for autonomous vehicles, highlighting key aspects such as data consistency, accuracy, and security at each stage of the OTA update lifecycle. Some of the key findings from the research bring forward mechanisms like hashing, digital signatures, and anomaly detection, as well as frameworks like Uptane, which address concerns surrounding tampering, MITM attacks, and rollback vulnerabilities.

Over the last decade, the industry has witnessed significant advancements in the cybersecurity field of autonomous vehicles, but these do not come without new challenges. Hashing is an effective method for detecting tampering; however, it does not prevent unauthorized access to secure files. Similarly, while digital signatures provide high-level assurance of integrity, they are computationally expensive for large-scale software releases. The case studies examined underscore the risks of inadequate integrity assurance and demonstrate the importance of thorough analysis both before and after the deployment of an update.

The shift from hardware-dependent maintenance to software-centric updates has proven to be a double-edged sword. While smart vehicles have made maintenance more convenient for users, they have also opened the door for malicious actors to exploit vulnerabilities. This bolsters the need for robust systems that address critical components affecting users, even if those components remain unseen. In this era, the industry has developed numerous security protocols and regulations to protect both users and manufacturers. However, with rapidly advancing technology, the focus must remain

equally on enhancing security as it is on introducing new features.

Looking forward, the industry must explore lightweight technologies that offer the same level of robustness as current multi-layered security systems. Achieving this goal will require collaboration between industry stakeholders and regulators to establish universal standards that foster a more secure and efficient OTA ecosystem. By combining advancements in security protocols with industry-wide cooperation, we can ensure that security evolves in step with the increasing complexity of autonomous vehicles.

## REFERENCES


[1] G. Kim and I. Y. Jung, "Integrity assurance of ota software update in smart vehicles," *International Journal on Smart Sensing and Intelligent Systems*, vol. 12, no. 1, pp. 1–8, 2019.

[2] T. Chowdhury *et al.*, "Safe and secure automotive over-the-air updates," in *Computer Safety, Reliability, and Security: 37th International Conference, SAFECOMP 2018, Västerås, Sweden, September 19-21, 2018, Proceedings 37*. Springer International Publishing, 2018.

[3] A. Greenberg. (2015, Jul) Hackers remotely kill a jeep on the highway—with me in it. [Accessed: 21-Nov-2024]. [Online]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

[4] E. Goldberg. (2024) Tesla ota update is breaking autopilot computer, owners report. [Accessed: 21-Nov-2024]. [Online]. Available: https://www.thedrive.com/news/tesla-owners-say-autopilot-adas-failing-as-recall-update-hits-2-million-cars#:~:text=Tesla%20News

[5] D. Shepardson. (2024, Apr) Tesla cuts price of full self-driving software by a third to $8,000. [Accessed: 21-Nov-2024]. [Online]. Available: https://www.reuters.com/business/autos-transportation/tesla-cuts-price-full-self-driving-software-by-third-8000-2024-04-21/

[6] C. News. (2024, Sep) Ontario man told his ev needs $33k battery; software update fixes the problem. [Accessed: 21-Nov-2024]. [Online]. Available: https://toronto.ctvnews.ca/ontario-man-told-his-ev-needs-33k-battery-software-update-fixes-the-problem-1.7023994

[7] E. Newswire. (2024, Oct) Automotive over-the-air updates market to reach usd 19.29 billion by 2032 owing to rising demand for electric vehicles. [Accessed: 21-Nov-2024]. [Online]. Available: https://www.einnews.com/pr_news/761579572/automotive-over-the-air-updates-market-to-reach-usd-19.29-billion-by-2032-owing-to-rising-demand-for-electric-vehicles