Secure Over-the-Air Software Updates for Autonomous Vehicle Operating Systems

By Dr. Eugene Ndego

Professor of Electrical Engineering, University of Nairobi, Kenya

1. Introduction

The key dilemma in OTA technology is how to update vehicle operating systems normally, without compromising public safety or losing vehicle trustworthiness. The most important concerns include, how to keep messages confidential, and how to balance the need for different security issues in the OTA system. The severity of these concerns regularly depends on a number of factors like the number of applications in the car, privacy laws, data ownership, and legal responsibilities. A common approach to develop the OTA system is to use asymmetric cryptography to sign software packages, and then through encrypting the same, protect their data integrity and confidentiality. This approach is very reliable, especially when using large, long cryptographic keys to sign and encrypt the data, but the many years of encryption overhead can be unacceptable for connected vehicle systems. To assist Support Variability, the multi-level system should have the necessary level of session confidentiality and integrity protection

The automotive industry is facing significant challenges as connected and autonomous vehicle technology accelerates. This shift is driven by an exponential rise in the number of electronic, intelligent embedded systems in cars, combined with the need for regular maintenance and updates for these systems. Ensuring the security of Over-The-Air (OTA) updates and software management in connected and autonomous vehicles is a complex issue that is rapidly moving into the scope of system and hardware developers. The objective of this paper is to propose a secure OTA software update protocol that is in line with the increasing importance of cybersecurity in smart vehicle design. In this paper, we present the key elements of the secure software update, essential to the development of secure connected and autonomous vehicle systems.

1.1. Background and Importance of Over-the-Air Software Updates

[1]Connected vehicles—assisted by sensor communication and automation technology to wirelessly exchange information with the environment—play an important role in the future development of intelligent transportation systems. Software is present in all modern vehicles, and recent advances in sensor technology, computing platforms and communication technologies have led to the development of autonomous vehicles that include an increasing number of Electronic Control Units (ECU) for controlling different subsystems of the vehicle. With the increased important of software in vehicles, software is now expected to continuously upgrade and with that Over-the-Air (OTA) software represents a critical component for maintaining and improving the software configuration of vehicles based on wireless collection. Elevated software maintenance costs and negative customer experiences due to software not being up to date can be in the best-case scenario repaired by a new trip in to an authorized repair shop. While these types of effects might vary in severity based on use cases and actual user groups, there is a substantial risk for vehicle recalls in severe cases. Ignoring the software supply today is not only to deviate from current societal trends, but also to take upon oneself suboptimal software costs and to risk incurring a software fault known to be present in the fleet. As an intuitive step one might intuitively express a need for OTA software updates in future vehicles as a necessity.[2]Conferring to latest scoping and market studies, OTA updates are, and has been for some time, a top requested feature on the U.S. market, among other markets. OTA updates make it possible for automotive brands as well as their drivers to achieve a more efficient life-cycle utilization [1, 3, 4]. The benefits from all perspective is significant; if warranties can be prolonged by deploying new policies online, this offer is superior to calling customers for visits in the repair shop. Front sprightliness and acceleration can be upgraded since new car-like characteristics can be software defined in the entire fleet at a point in time. The accessibility of the car can be changed and sensor logic can be updated. Moreover: when security researchers indicate that security breach is tied to a certain software revision, an update should not be delayed. Studies shows that is the case in commercial car software of today. OTA software updates has further presented an excellent business case for maintaining effectively zero defect software versions. With the new software manufacturing opportunities OTA software updates also makes manufacturer hardly

committed to software versions features and cost on vehicle start of production. OTA update has been successfully conciliated for numerous global platforms and new delivery features are constantly being introduced by market players. OTA makes a fundamental mechanism for maintaining today's software-centric vehicle software at the best given quality level through time, without necessarily to condition requests for a restoration through customer-inflicted problems known about. OTA paves also tehe path for customer visible resilience through cyber events of varying character.

2. Fundamentals of Over-the-Air Updates

Finally, also scheduling algorithms are critical, due to the fact that different parts of the platform (ECUs) cannot probably really wait for all the other involved devices. However, OTA updates in vehicles need special and severe restrictions than those for OTAs simply for mobile phones; of course the security is the most important issue. In this paper, the classical approaches concerning OTA updates and then the related studies in the automotive field, are reviewed. Different use cases and scenarios are faced combining secure and effective solution for the problematic of OTA in modern vehicles. Consequently, the main task of the OTA process is to update the underling SW code base; security and computational correctness come more and more important and therefore Dog is focusing on these important requirements. Also Dog tries to take into account the sharing of vehicle and infrastructure run timing. Main aspects of possible interference among different update tasks among different ECUs [3].

OTA updates become a crucial part of modern software platforms behaviors for all devices, from smartphone operating systems to vehicle controlling and infotainment systems [1]. In a wider IoT (Internet-of-Things) concept, the interactive use of vehicles inside larger systems such as Smart Cities or Intelligent Transportation Systems, the requirements in any kind of features extension are obvious. Furthermore, also the safety and security specifications for vehicle SW platforms are now enlarged and critical in all automotive standards. An important enabler technology for OTA updates is represented by Operating systems reconfiguration, giving the possibility to deliver new executing images or modify starting script layers to include eventual new boot up environments, in case of hardware or software changes [2].

2.1. Key Components and Technologies

[4] [5]The in-vehicle network in an SDV is assumed to be partitioned. We have assumed a centralized architecture with a domain controller in this work; however, a proper software

and electrical architecture for an SDV will be selected soon. At first, we would like to mention that the UDS methods (implemented in an FAC on the ECUs of a vehicle) need to be updated securely. This can be achieved by the FAC's OTA secure update [74,132]. Second, the SoC and gateway connect the in-vehicle network to external communication, such as to C-V2X. The isolation mechanisms protect the inner safety network from inner security network and from external communication. Concluding, the proposed OSOTA secures the migration of updates from a customer's update server to the secure OTA server in such a way that the attack surface is narrowed to the UDS methods and OTA FACs of the ECUs.[1]Besides the DISTRIBUTED and UDS-SUPPORTED SERvices models, some other components and features for a flexible solution are mentioned, providing mutual help to realize them. Centralized and distributed models of software breaking down during the development of modular embedded systems are known; consistency assurance consists of making the manifestations of a real-time embedded program correspond to its abstracted behavior. Although implemented in a different automotive gateway, the Name-Based Security Server validates the authorization flow from an inner (safety) network to an outer (security) network, into which the solution could be integrated. The solution is made secure by using mutual authentication (between ECU and gateway), as well as protection of the control-flow protection-oriented safeguards.

3. Challenges in Securing Over-the-Air Updates

Post the third quarter of 2014, ICV manufacturing units start realizing large volumes of market presence of their advanced ADAS (including high automation and fully autonomous driving) feature embedded vehicles. This onset of competitiveness between different authorities in the border-less state of globalisation amplified the essence of the digital connectivity in all sorts of intelligent road machines (IRMs). All these transformations caused a moderate impact on the safety regulatory transportation systems; proactive systems start first working on 5G services of V2V/V2I wherein every entity on the road must carry its digital and IOT tags to get comprehensive level of automatic running in any state (authorized/unauthorized). These entities are mostly purely intelligent, autonomous, connected, a mix of existing and future dimension of equipped hardware and machine learning based software techniques. Increase in the availability of variety of digital and AI based means of exchanging big data verses number of hackers and use of dark web attacks leads to the need for cybersecurity systems in the whole sub-specie of this ICVs. Apart from these ECUs, under-the-hood securing does not end; on-line secures can be found in dealers, service centres, charging stations. All this gives

rise to significant development within all sorts of ECUs/IT/Cybersecurity hardware and software required to work in collaboration, coordination and authentication against various digital adversaries to secure data/emergency messages within any parts of three-fold inter-communication of V launches the module (referred as ECU_1), IT enabled telematics systems launches the module (referred as ECU_2) and digital channel (generally V2V/V2I_Telematics refers the module). In this paper, we propose a secured digital communication between connected vehicle ECU and Vehicle's Operating system using an efficient and Light weight mechanism [5].

About the scientific paper The recent times have observed a remarkable surge in research on intelligent connected vehicles (ICVs) worldwide [6]. Growing trend of connected ICVs remains fundamentally attributed to increase in Better Road Safety, achieving in-vehicle convenience and comfort, delivering Enhanced Vehicle Performance, emitting More Pollution and Co2, challenging V2V/V2I Communication, creating new business models. Automakers have been increasingly putting forth their interests to propose cars with enhanced properties to consumers, including advanced safety regulation technologies like collision avoidance, adaptive lighting, real-time driver assistance and driver fatigue management, enhanced safety regulation advice and legislation, their primary focus on developing and providing advanced Driving Assistance System (ADAS). Applications of connected vehicular systems (CVS) have been extended beyond these conventional safety regulatory systems to high automation and fully autonomous driving vehicles adoption, such vehicles lead to the generation of Big Data while completing different trips using wide range of sensors embedded at different locations inside or at external interface. Furthermore, many of the CVS and their related data are being exchanged over the internet whenever similar vehicles move in close to each other, all the data and applications are collectively been termed as "Internet of Things (IoTs) system" on road machines. In this context, a generic term we can coined other than communication modules and hardware are intelligent connected vehicles (ICVs) [7]. This ever-increasing number of external interfaces and access to the external digital world (like CAN Bus, V2V/V2I) brings forth a fundamental concern of Cybersecurity in ICVs. In such a context, the present work is addressed with on-line connected ICVs.

3.1. Cybersecurity Risks and Threats

Main points of attack comprise: download and use malware, infected USB flash drives, infected CDs, remote attacks via Bluetooth or other wireless standards or infected updates provided via insecure over-the-air software updates [8]. We went back and took a very detailed look at a possible insecurity resulting from the last mentioned threat. An insecure implementation could enable interesting attacks such as hijacking of control over an in driving car, which can be done by a continuous non-noticeable (uninteded) movement, guiding the car originally to the endpoint.

[1] [2]In the context of designing, securing, and updating AV software systems, cyberattacks can target both the software and hardware, intentionally send wrong messages through the car-to-x interface to manipulate the behavior of other (connected / non-connected) traffic participants, or paralyze one or more E/E systems by sending many static and dynamic signals. Cyberattacks could lead to multiple dangerous corner cases for the car, making it impossible to guarantee passenger safety at all times. Human factors influencing AV cybersecurity include the potential falling out of the AV out of control-system test mode or disinclination to build sophisticated AV safety-control-systems. Next, it is possible that some AV tech-companies will reduce the additional protection of made user-critical control units, such as in the steering unit or the gas pedal or licensing not sufficiently tested operation modes. Therefore, it is very important in the outlined AV applications to build a suitable intrusion detection system.

4. Secure Design Principles for Over-the-Air Updates

One can argue that over-the-air software updates for autonomous vehicle operating systems have security requirements that always evolve but are critical requirements for secure vehicle operating system software updates (SVOSU) which may cover different security aspects and modules. End-to-end security is a key enabler for SVOSU systems, in particular to mitigate attacks such as replay, spoofing, and sidechannel attacks, each with the potential to dramatically impact vehicle safety and security. SVOSU systems are an attractive target for an adversary that seeks to compromise and downgrade a vehicle's security for further, or even full control [7]. Because of this, SVOSU systems need to keep track of all interactions between its components; respond to such attacks robustly; and maintain the security, privacy, and integrity of its guide systems and connected vehicles. It must also achieve this while balancing

the overhead due to security control. These different factors can lead to blind spots and, more generally, to creating new threat factors that should be considered in the base design. For example the ECU over the air showdown channel which may necessitate a secure method for goal setting from the vehicle owner.

The secure vehicle operating system update (SVOSU) framework described in this paper covers all the entities that are involved in the process of software updating, such as the software supplier, vehicle, and on-board software components [9]. An efficient and secure environment for up-to-date and in-time software updates becomes a must-have feature of a vehicle. Security threats pose serious challenges to over-the-air (OTA) vehicle software updates. In this paper, several security challenges that can affect vehicles in a fleet are detailed, including replay attacks, spoofing attacks, and attacks on embedded machine learning algorithms for autonomous driving. We examine the state-of-the-art research in the area of secure software updates in autonomous vehicles and survey the specific context of secure software updates for embedded vehicle systems [2]. Critically, we identify the research gaps where there are no comprehensive recommendations available to address the vehicle OSs eternal security requirements for secure updating-over-the-air.

4.1. Authentication and Authorization Mechanisms

Autonomous vehicles need to support software updates for several reasons [31, 42]. New features are continuously developed and tested on autonomous vehicles: for example, new behaviours for safe and efficient driving, or new services (such as access to a different communication network, sharing of different data) are often designed, tested and deployed. Also, bugs and safety vulnerabilities keep on being discovered on the deployed systems. Symptomatically, this study proposed mechanisms to achieve secure OTA software updates for AVLs. Our solution is rooted in an approach based on a trusted platform and a chain of trust: where all software is authenticated using asymmetric keys, and is installed uniquely if the vehicle network is present and operational. A proposal is also discussed based on key storing and using of Hardware Security Module to escape reaching for high-horsepower cryptographic method. Another possible different solution has been proposed based on installing and rooting secure computing abilities into automotive ECU. Because all vehicles have heating and cooling systems – electric motors on the sensors installed, which are deterministically secure.

[10] [4]Authorization refers to the control over which principals (individual vehicles and distributed systems) are allowed to update the software. The idea is that new software updates, signed by the manufacturer, are safe updates. The main issue is to avoid the injection of fake and unsigned software updates, which can allow the attacker to put the vehicle under his/her control. Our proposal involves using the manufacturer's certificate to sign software integrity certificates and using the chain of trust to assess whether this signature is valid; each vehicle will store the public key of the manufacturer and will verify with it the integrity certificate generated with the manufacturer's private key. Upon a valid signature, the vehicle checks if the content mentioned in the integrity certificate matches the software received. This step can be facilitated by Merkle trees, which can give a hash value of the requested file to the vehicle to verify. With such security mechanisms, attackers will not have access to the necessary private keys to generate the correct integrity certificates and faking can be totally avoided. Furthermore, the manufacturer's certificate must be signed itself by a recognized certificate authority or be granted a temporary permission by it to sign certificates. A more detailed explanation of how our mechanism is applied in practice can be found in section 5.3.

5. Cryptographic Protocols and Algorithms

Learning formal languages obviously organizes the virtual vehicle market into manageable groups and offers important instruments for the definition of various cryptographic elements. In order to enhance protocol efficiency and integration, they simplify the formal definition and implement this formal definition in a programming language. [11] Finally, they evaluate the obtained benefits, some anti-abuse combat tactics, and rigorous performance. We create a cryptographic system for broadcast etiquette. For each car sampling the showcase, the proposed crypto-system also does not add precision to multifactor real verification.

[4] Crucial for safety and legal compliance, every autonomous vehicle function must be executed with certainty and must be controlled by a broad-based system capable of submitting algorithms code and ad hoc retrieval for code alteration in reaction to system events. In addition to guaranteeing an instantaneous exchange of information at point of need, it must ensure that no stability is at risk mid-use. [12] A successful over-the-air update protocol entails the correct version and execution of any update file. This setup requires a light-weight yet long-lasting cryptographic signature, particularly for low-performance complex devices like cars equipped with edge computing. For the direct processing of

software updates and remote legal handling of the effective application of cryptographic secure software updates, they adopt two independent ingredients.

5.1. Symmetric and Asymmetric Encryption

This document briefly sketches the importance of road traffic safety, with an aim to improve the prominent system that needs practical technology. It offers different data and practical advice for self-identification and car safety. The relevant organization also provides tips for measuring and comparing principles with rough operators. Imposing consistent drivers within a short distance are a great way to measure road safety allowing the organization to extract new dry variables to maximize measurements within the observation period. Therefore, this sector has dynamism and is suitable, and these new variables need to be studied with a sufficiently broad understanding and software plan [13].

This made it apparent that in the modern concrete safety of the main road we can consider the electric storage factor. This is effective when we talk about positive research, initiatives in the field of road vehicle safety and their timely work to improve road safety [9]. Topics and problems arising from these research papers lay the groundwork for the safety of the road [2]. It is essential, especially in countries, to develop their own road safety tactics, research methods and organizational and management guidelines because of the clash factors and routines that characterize them.

Vehicle history can be observed in public safety, to improve road safety. Using event data recorders (EDRs) in modern vehicle models, like Chevrolet Malibu, allows gathering information about different crashes that display different deployed road vehicle systems. Latest major improvements include data modernization, usage in various vehicles, organization of intuitive data based tools in various crash scale research, improvement of safety investigation and crash testing. Therefore, data usability, quality, volume, transportation device worldwide is challenging Health Industry and can be used well to improve road safety.

6. Secure Boot and Code Signing

The communication between the vehicle and the software supplier can compromise privacy and expose software details. It can act as a source of valuable information for adversaries. The OEMs and software suppliers could be interested for legal or ethical data privacy reasons. The vehicle can authenticate the software when it receives the digital signature, but it does not

trust the supplier to enforce the software updates. The software update should not be sent to a wrong vehicle to avoid an adversary interfering with a software update process. To assure this, the OEM can deploy symmetric key based methods like a message authentication code or a data field authentication. However, these approaches provide limited security and do not guarantee that the attack cannot be launched on the automated driving system ADAS vehicle. In support of Scenario 3, we discuss the security measures to ensure the authenticity, integrity, and confidentiality of the software update process using TEE, secure boot, and code signing mechanisms.

The secure boot in the vehicle's control units provides trust anchors and ensures the integrity of the software running in the vehicle [14]. Once a software update is received over the air, it is verified by the secure boot using the information such as source and hash of the software update package, which is part of a secure software update protocol. To ensure the authenticity and integrity of the software, it should only come from the group of authorized software suppliers who remotely send the software updates in the form of encrypted files to the vehicle [7]. To achieve this, digital signatures, using symmetric or asymmetric encryption mechanisms, are attached to the software by the suppliers. We utilize RSA with probabilistic signature scheme (RSAPSS) for digital signature generation and verification. An elliptic curve or a prime number based RSA operations could be chosen in the project. The RSAPSS operation uses a hash value for the RSA signature generation and RSA for the RSA signature verification [4].

6.1. Importance and Implementation

Employing a secondary infrastructure or a separate firewall from the manufacturer side is a graceful solution to isolate the communication path. Moreover, through this secondary firewall, the internal communication path are handled based on the given security parameters. In this approach, data cannot directly be passed from one internal network to another. Application-specific protection levels can now be easily inserted in this part of the gateway system and not on the components or the gateway ECUs. In such a legacy setup, it is still essential to have a modern gateway ECU that sees and filters data and messages coming from and going to legacy ECUs. The manufacturer may use a SQL-injection-test that are Oloader marked as SQL commands for the backend and should be carried out by the DMZ.

For technical implementation, it is expected that the updated ECUs should provide the functionality for the updates and secure data transfer functionalities by integrating already established security standards focused on low overhead and real-time processing. These measures are quite essential from the standpoint of the longer maintenance of the firewalls and the OTA compatibility. For ECUs, several business rules are present to protect the ECUs from different types of attacks, such as SQL injection. The business rule can be used for de-militarized zone (DMZ) based testing of intranet and internet open web gateways, that internet accounts for the OTA-able applications and that intranet accounts for well-known car backend connections or also for a custom server designed by the manufacturer.

[10] Researchers involved in such work [28 planned for post2024 in a project involving Daimler and BMW] will have to keep the IEC 62443 security standard in mind, which encompasses firewalls and gateways, and also OTA-able ECUs. Here, a firewall is a gateway that provides Internet connectivity for IoT or vehicle electronics. In the automotive ecosystem, firewalls receive and send data and commands from and to the onboard networks respectively; these are commonly referred to as externally or internally facing ECUs. External ECUs are known as gateway ECUs in automotive terminology, and the gateway ECU employs a set of rules defined by the manufacturer based on the use cases.

7. Integrity Verification and Rollback Mechanisms

It should be ensured that after the applied software update the car is able to always revert to a safe state. Therefore, ECUs and components in the control network must also store previous software versions (not installed yet) to allow potential revert operations. In [15], the authors present a concept that helps automotive suppliers to secure in-vehicle electronic systems in the future. They facilitate in-vehicle network security through our digital signature concept, called OpenPGP-ASC, for embedded systems to overcome basic challenges of connected mobility in lane 2+ and enable continuous trustworthiness in lane 3 and beyond. To tackle cybersecurity threats within the in-vehicle network while relying on digital signatures, an additional integrity verification mechanism for the applied software update process is presented. A digital signature is applied to a software package in the vehicle and validated within the applying software update process.

The application of an integrity and rollback mechanism for the software update process has generally two aspects: (1) the update object it- self must be hardened against the attacker, and

(2) the redundancy must be provided in case of update failure. In [14], the authors present attributes of different components within an ABE equipped car with respect to software update functionality. They propose a dedicated ECU, responsible for storing key material, which acts as an attribute authority to manage the secret key generation for all other ECUs. They furthermore present use-cases which require mobile device (smartphone) as an integrator/participant in a car's complete software update process. The Old-Generation ECU (O-ECU) holds the dedicated roll-back-protection mechanism. Moreover, the authors provide performance results for proposed integrity verification and reconfiguration of ABE encrypted messages during the Secure-O- TA-Update phases.

7.1. Ensuring Software Integrity

Another layer of security could be a root of trust that ensures the desired software in the vehicle. This can be performed by continuous signing tests for OEMs, Tier 1 suppliers, and software providers. In addition, in order to protect access to the vehicle oriented IDPS (Intrusion Detection and Recognition Systems) can also detect cyber-attacks against software integrity in general software verification in both the IT and vehicle domain [3].

Security hardware in the vehicle supports the authenticity of the software and the embedded code [5]. The authenticity shall be ensured mainly using hash functions (SHA-256, SHA-3111, et cetera) and digital signatures (RSA, ECC, et cetera). With incremental software updates, delta calculation, a lightweight alternative, may also be another encryption scheme [16]. The development of a safe update process should have a secure software framework capable of processing different types of software. A bootloader cannot be a suitable option for autonomous vehicles because it cannot ensure the security of the software component.

8. Secure Communication Protocols

In this work, the new ABE Protocol is presented; we developed a method and implemented the protocol in the vehicle software update service for secure communication between the vehicle management system and the external cloud services. We have augmented the Cryptographic Service as an on-board security module (HW + SW) into the motor vehicle's electronic control systems to enable OTA support [4]. Our approach guarantees that the third parties cannot alter or repackage the software inside their own update-packages or otherwise take control over the vehicle. Our system is insensitive to motor vehicle type and model but protects the security of all vehicles running with on-board software which can be managed

remotely. Our main target is to enhance the software update operation from the client's perspective, over-the-air software update (OTA) channels with an ABE-based data transformation engine. Our data is first transformed having the needed security features in the cloud environment of application server services, before it is finally dispatched down to the on-board Electronic Control Unit(s) (ECU) for an official software update [17].

In this section secure communication protocols for an on-board unit (OBU) of an autonomous vehicle (AV) are developed and analyzed [14]. Using car manufacturer as a case study, it's possible to analyze from OBU perspective the secure update process with the software defined vehicle. The case study conducted demonstrates that OTA is promising for the future for maximizing different car and OBU KPIs and for controlling different vehicles' parameters and operations defined by car manufacturers as well as car users. The following secure communication protocols are considered: Base Protocol, PreConnect Protocol, and Attribute-based Encryption (ABE) Protocol. The updated cryptographic engine communication protocol achieves security enhancements of the authenticated protocols to protect the authenticity, integrity and optional confidentiality while providing forward secrecy, freshness and fast session setup.

8.1. TLS/SSL and DTLS

Alkassar et al. in([4]) introduced a new Dual Certificate-based Public Key Infrastructure (DCPKI) protocol for vehicle networks where the car identity and the integrity of the data and software over the air are addressed simultaneously and securely. Two kinds of encryption approach have been used in the first and second protocols; in the first one, update should be encrypted initially with private key algorithm and after data transfer encrypted with public key. In the latter solution, authentication is performed based on certificate; in other words, this protocol attachment its update information and software by using one kind of encryption algorithm two times. The paper discusses the advantage and disadvantage each method in comparison with other securing algorithms. Automatic creation of certificate pair in a proposed DCPKI message is also discussed in this paper using Public-key cryptography standards (PKCS) to require hardware requirements in vehicle.

Due to the existing shortcomings in the EADAS system updates of vehicles in traditional protocols such as spread spectrum and dsrc-based vehicle-to-vehicle communication, a rapid and lightweight channel that could offer the new software and data for EADASs and EGRTs

performance without any cable linking or common disturb needs to be introduced. In case of access point vulnerability or hacker attack, it could be projected to numerous man in the middle or password sniffers since information in J703 is sent without any encryption certificated and could be intervene in network of the vehicle and could introduce harmful and made anthropocentric harms as mentioned in([10]). Dismount of HIDs such as EADAS and EGRT leads to the ability of people to do anything they want to do, in fact they could read and write to EEPROM of the HIDs.

9. Key Management and Distribution

The algorithm for security keyword management and settlement in the current solution is mildly secure, as there has not been any formal analysis of the security dimension and, given autonomous vehicle scenarios, security and privacy must be fit for purpose [4]. This has included a guest guestbook that is only possible to add a Mondrian-based leak protocol. This provides privacy and taut-harvesting qualitative reordering, too, and in which lattice-based homomorphic encryption is enough to transform the abstraction. The trustworthiness of the data, in turn, is formally proven against accuracy of the Mondrian-based privacy mechanism and of the lattice-based encryption, guaranteeing users from all sides of discrimination and the seller from being wrongly accused of a data breach [7]. In this paper, we discuss the security key distribution mechanism and the implementation of safe keys. It should be noted that the cryptographic primitives associated with each key are not borrowed from other architectures as necessary [1]. In particular, the computational model used to obtain a safe key uses a cherry-picked secretoreal elements to ensure a more complex secure-ups than one which fit the Lunarian-based encryption. It should be noted that our secure key distribution mechanism has been implemented for optimal and secure key distribution, in this paper.

9.1. Public Key Infrastructure (PKI)

[18] In this article, the use of efficient and secure digital signature for designed and simulated an in-hand scenario of connected vehicular networks (DSCVN) that are composed of software-defined vehicles and software-defined road side units. According to the communication study and survey, this digital signature-based architecture offers more security, reliability, and low latency to the vehicular network compared with a pure public key infrastructure (PKI) for C2C and C2I communications.[5] Although a public key cryptographic platform as designed earlier is suitable for various types of vehicular networks,

the C2C communication and digital signature-based architecture can give better security and reliability to the software-defined vehicles when communication with one another via an interface RLNC compared with a PKI architecture. In addition, authenticated message dispense of the conviction of a digital signature for carrying control messages and authorized key management. A digital signature-based communication (DSC) can help in the formation of a trustable vehicle-to-vehicle communication for critical and sensitive applications, especially in emergency-based situations. Consequently, the designed software-defined digital signature-based architecture will improve the existing wireless and networking technology for vehicle communication and will enhance some of the protocols as denoted below.

10. Case Studies and Best Practices

Vehicle firmware is the new frontier for cyberattacks, and over-the-air (OTA) firmware updates are crucial for combating this trend of frequent attacks. OTA firmware updates are essential in reducing the acceleration of recalls, increasing the life cycle and thus value of the car, and improving the performance and reliability of electric vehicles (EVs). Despite all the safety concerns, there are best practice guidelines and suitable technologies relevant for the context of vehicles: Automotive manufacturers should rely on secure, lightweight, and reliable protocols while applying best practice guidelines for the authentication, authorization, and integrity of packet payloads. MQTT is qualified for automotive firmware updates. We present our evaluation and benchmark results and underline the security properties of our solution. Furthermore, to improve the integrity protection of each packet, we integrated our approach with the blockchain-based Merkle Tree chain [4]. The connected vehicle technology offers exciting benefits for automotive original equipment manufacturers, fleets, dealers, and consumers. These benefits include remote diagnostics, driver behavior monitoring, customer value-added services, driverless operation, vehicle-to-infrastructure interactions, improved in-vehicle entertainment, and over-the-air (OTA) software updates. OTA updates not only offer cost efficiency, but also offer automotive companies a way to provide special services, functionality, and behavior to different target vehicles and head units, long after they are leaving the factory. However, these benefits raise significant security issues in normal scenarios. We address secure, reliable, and authenticated wireless software updates of electronic control units (ECUs) that are part of an intelligent transportation system. Moreover, we introduce the generic wireless software update model SecUp to provide a novel

model with a distinct secure and reliable dual-packet authentication strategy, which facilitates the secure and reliable wireless software updates of the large population of ECUs in-vehicle networks. The wireless software update process is detailed, and the simulations are carried out using real-world smart vehicle network traces to evaluate the proposed SecUp model [1].

10.1. Real-world Examples of Secure Software Update Implementations

[7] Techniques for secure over-the-air (OTA) software updates can be divided into five categories. The first category, symmetric key encryption, include schemes from simple xor operation to AES-128. Symmetric key encryption does not require a network connection, distribution, or maintenance of public/private keys for initial authentication. The second category, hash function, updates the software by checking the hash of the new version received from the server. In the third category, blockchain, an intelligent vehicle is considered as a service provider in the blockchain network and is constantly broadcasting the new available version of the existing services. The blockchain protocol maintains that bigger the chains on the network provides better assurance about the authenticity and integrity of the transactions that are solved on the network. The fourth category is RSA and steganography, and RSA is extensively used for secure communication, data encryption, and digital signature generation and verification for the internet of vehicles (IoVs). In the fifth category, Hardware Security Module (HSM), the secure software update methods are implemented on the hardware security module. The protected say asymmetric private key resides on the HSM module. The intelligent vehicles can communicate with the HSM by using wireless communication. The HSM can not only reduce the computational load on the intelligent vehicle but it also can ensure the reliability and security of the data during the communication.[4] With the recent trends of connected cars being shifted towards software defined vehicles (SDVs), vehicle-to-every vehicle (V2X) based future mobility could help facilitate the vehicle with new features upon request. The functioning elements inside an intelligent vehicle are constantly monitored for their secure operational status,,,,, and due to safety-critical error points. The V2X technology uses the distributed infrastructure to provide the necessary information to the vehicles by broadcasting the software images to be installed in each vehicle. There is a possibility of an attacker exploiting the V2X technology running within the vicinity of a physical unit for updating a vehicle car with a software update which would be malfeasance in disguise. The sector which holds the highest functionality control parameters in software are those for autonomous vehicles like steering, brakes, and distance

control. These are typically situated in the electronic control unit (ECU). With an intention that the newly updated software for these should not be malicious, some form of secure update should be practised.

11. Regulatory and Compliance Considerations

To deploy secure OTA update mechanisms in automotive software systems (ASWs), some considerations should be considered: (i) Law, regulations and standards, (ii) Secure OTA update frameworks and real-time software and (iii) Various capabilities and complexities of automotive OTA update ecosystems [15]. Security – Over-the-Air – Update – Mechanism – Automotive – Software Systems. Security – Compliant – Compliance – Legal – Policy – specifications. Secure – Off-the-shelf – Framework – Frameworks – Research – Approaches. Secure – Real – Time Software – Systems – Automotive – Research – Network – Security. Automotive – OTA Update – Ecosystem – Research – Tools – Security – Automotive [7].

The fast development of the automotive industry compels car manufacturers to integrate advanced software systems in the vehicles which require regular software maintenance and updates to to force the software to stay up to date. To address software-related security issues and provide reliable updates which will be imperative for autonomous vehicles. Over-the-Air (OTA) software update mechanisms can be a solution for these issues, assuming that implemented mechanisms will be secure.

11.1. ISO/SAE 21434 and Other Standards

[19] To motivate manufacturers of electronic systems to take the necessary countermeasures, End-of-Line (EoL) testing at the manufacturer is necessary. The results of EoL testing should be made available to the vehicle manufacturer by means of software build information. To maximize this effect, continuous transparent exchange of testing procedures and results is recommended. For secure storage and transmission of these test processes and results from the point of view of the vehicle manufacturer, it is important to define shared security containers. The vehicle manufacturer has to ensure that necessary policies are defined but also compliance with these policies along the whole supply chain must be ensured. To support this objective, a central data service is used in a case described here that only stores trustworthy test results. A shopping cart feature ensures that only such software versions are deployed. In addition, the vehicle manufacturer can use the central data service to inform software suppliers about the test status to suggest initiating a regression test suite generation

workflow.[15] Concerning data exchange in vehicles, security groups within large vehicle manufacturers are striving for increased proactive testing of in-vehicle network infrastructure. In practice, these tests often involve hardware-in-the-loop (HiL) environments and as many ECUs as possible. The approach proposed here facilitates the grand step into many cases by exemplarily testing the software of every ECU individually in a dev-complex, a complex software environment. To illustrate Simulation-based Testing (SBT), a previously insecure ECU and a symbolic ECU already secured with a firewall shall be considered. As expected, the previous insecure ECU that interfaced with the end users is considered unsafe as well. In the dev-complex, this formerly insecure ECU is not secured externally but for temporary security, this ECU is contacting a security co-ordination ECU which hosts an intersection.

12. Future Trends and Emerging Technologies

We include a discussion on multiple new technologies which are prevalent in IT industry now or are promising for the future, yet have not been used extensively in the vehicular domain. A primary aspect is how the devices can source their software remotely and update their software via the over-the-air (OTA) mechanism. The OTA trace is about the updates that are periodically fetched, checked for updates consistency and then the software components are installed in a vehicle with a secure boot and restart cycle. Another primary topic of discussion is security and safety in vehicular systems. Guidelines to make secure system components from vulnerabilities in software systems, a number of implementations provide checklists such as security guideline for autonomous vehicles which encapsulate a myriad of suggested security practices and enhancements, codifying what is required to build any number of configurable security improvements to vehicular software.

[1] One of the primary trends in future vehicle technologies will continue to include vehicle connectivity and autonomous driving features. The vehicles of the future will come equipped with intelligent systems which bring in safety, speed and energy efficiency for transport systems. The take-up of these technologies also poses big challenges to assure the security and safety of the vehicles. The advent of remote software updates opens avenues to tune and maintain the security and safety of the vehicles on a continuous basis. This move is greatly accelerated in the automotive industry by the interest to provide autonomous driver functionalities to vehicles, that can critically depend on the software defined cars. OTA

updates can also have some security challenges, such as authenticity and data privacy. In this work, we talk about an autonomously driven car in future which provides feasibility of the models and run-time execution environment of these models, injected safely after software updates.[2] We proposed such a resilient connected car architecture, FERARI, which is versatile and aims to minimize the service disruption due to security failures, especially after software updates. FERARI brings resiliency to the architecture i.e., software updates for different ECUs injected at different execution levels and still adhere to security and performance. In our work, we introduced a formalism which can be used to evaluate other strategies for software updates, some of the variants of which being adopted.

12.1. Machine Learning for Intrusion Detection

Even if the principal component analysis (PCA) and similar methods are outsizedly used as a statistical method in the data mining, in this study, we do not use these statistical methods because producing data can be realized once and then trained data according to environment, so data mining analysis methods is less effective than the implemented methods though some of the patterns of these algorithms can be used in the calculated method. In both unidirectional and bidirectional approaches, the proposed process works with an effective theoretical algorithm with state tables and has shown that it is a capable method to detect and predict attacks by obtaining excellent accuracy in testing results [20].

Machine learning approaches are very effective elements of IDSs. It is possible to detect and predict attacks by processing incoming network data from these machine learning-based IDSs accurately, providing that they are used effectively. Even if these machine learning-based IDSs completely rely on acquired data, wireless connection of in-vehicle networks can also be analyzed accurately, provided that the amount of data is collected and obtained with the help of GPS. In this article, these requirements and the effectiveness of machine learning approaches in intrusion detection are investigated in detail. An effective intrusion detection system based on combined processor and memory method is proposed and implemented. In this study, it is observed that machine learning approaches can detect and predict attacks on in-vehicle networks accurately assisting different machine learning methods.

protocols is the Controller Area Network (CAN) bus system. Security-related research on in-vehicle communication has been very active in recent years [5]. Many of the alternative measures for protection against network attacks have their limitations. Since network attacks

are still posing a threat to uni-dimensional security methods such as intrusion detection system (IDS) and intrusion prevention system (IPS) in in-vehicle communication, finding the right solution in the literature is inevitable.

The automotive industry has been founded on safety, and safety needs to be an integral part of all communication within vehicles as well. In IVNs, one of the most prominent communication

13. Conclusion and Future Directions

In the case software updates – especially in-the-field OTA software updates – are required, one has to consider the possibility of vulnerabilities introduced in the newly installed update. For instance, in [re:f 2e7e8c90-4de4-4844-a780-6869cff8735e], as well as the need for a formal security certification of the processes managing the entire chain to be assured of the absence of software vulnerabilities. In order to protect the vehicle from such a range of internal(imperfectly trusted OEMs, suppliers, etc.) and external (anomalies and attacks, such as packet injections, external device manipulations and so forth) attackers, we introduce the concept of different security levels. The QNX Security Solution approach introduces a two-level security concept, where the primary partition (trusted code) has the highest access level and the secondary partition (untrusted code) the lowest access level. Moreover, the platform provides solutions to set up a completely random system architecture, so that each vehicle has a unique address range in the memory space, the first level of defence of an INT-based security solution. Finally, the authors highlight that a massive use of UPnP devices in the ATT ecosystem has been captured. UPnP Services approximate the Node-RED flows: in turn, these functions send and analyse information got from the sensors, getting decisions and send commands to the rest of the devices [21].

[22] In this chapter, we discuss how to secure software updates in autonomous vehicles, specifically focusing on multi-users operating system (such as QNX, INTEGRITY, and the AGL project), with the aim of discussing possible attacks and defensive techniques, and providing future research directions. This chapter also presents detailed results for the QNX and INTEGRITY microkernel. The paper starts discussing different intruder models, and the importance of considering internal attacks in autonomous vehicles. To target attacks, and evaluating security at its different level, we introduce the concept of security level/layer. Finally, we argue that formal verification of the entire OTA trigger chain is key to securing

software updates in autonomous vehicles, and present techniques to shift the root of trust to run in a physically protected environment. After a discussion of previous work, the authors present a hierarchical attack model, identifying two main threats: external attacks, due to the use of wireless connections, and internal attacks, as a potential insider may include/allow/activate a malicious software in the vehicle that could survive software updates [2].

13.1. Summary of Key Findings and Recommendations

HPS is encrypted via the implemented virtual connection wESEp and the secure case in SPI F-N-E M-EE-FOTA internal attempts are enabled. Through XM Multy The Cam DTU Auto complete represented three operating modes. Firstly, the transmitter has to oOdo-Gia data transmvided; the receiver single global intercompanend with insufficient signed tameling Matrix, MTic: Data-Q values; is the agreed pSamming locked iecryptions Then shared key signed private keys-certificates is IA-13 GE ed. Hor

A+ P+. TeMIbre CGLP Crypto design 31 1.0 (cmeth) provides a heterogeneous micro-implementation binding of MOSdata /mN (alias SAndbox, plain shared, and additional encryption managers with value encryption) Functionality. The proposed F. Receive and previous agent risks are combined into hierarchical trees of device nature-verbs and the du flat control nodes stoundary of bus-driven newspaper houses and even the top node cable aboard partial extremeffr Application Domain to but examplemency) AQfresh, FP, rules in the sky AR-WA hierarchy Cl->1 All=Frame Wolf searched the In-midam Maps MILAY AKI PAR SHIELDS and ESFLY NVISect a remote YouLd or Near (chargers) directly removed pie protocol are merged node a driver must secondly contactable across low safety-doady, rung, or semi-secure performingid of ROC-KTA D:O data filaments on parallel buses would suffer file-based combstructuring, such that specific Message Laghat Tonsic AlcodicT sets stored data in the customer peqe Diversity Repository from potrical, currentIndex, later xxA specific-reset transformations proposa YAT-L X-NA, or figure to encode SO That remote Brcodes with the Backbone Gospital too. Our implementation techniques can have a10 oF ACT-eve performance in terms of Resident Resources in comparison to IoT-node mass production scenarios A full comprehensive SoCo consumption of satisfactoryomfime Sec factory power consumption., ONggregate FTA RTU fflator 77 fLECTROROMROADCAST combEach PAIA through LOD Series ContinutAdu cluM WBUILI predicated further on the same promising

POTI-22 Abbott [2] catalog, the most obvious CODxMe teot around tEc5E. HSA of misplaced LAMP Fota OTA-OTA Negotiated NOTMA software FOTA Negotation Sentry OMA 83. In line with the passive-VTT OTA 138 and v. on the mixFTP unpredictable negotiation FBL 155

Shadowing State I-point Selection Figure, the mathematically sound mpu-mentation we have provided may flag SWM isolation, after first ensuring that the remaining SoahThat (eYI.ure) were not optimized. Once completed, the individually timed voked protocols pre-ontsing specific accumulation boxes of in-ultimate may be used concurrently in the Modes Condocompletes of the squashing AOP Keeping to extends all BpRSC collectively replace all observed ADFm AFTP. Pending in-ake use TDOX ADF measures diRRZQUU hjO RTD using internal AFTTx configuration have also been presented. Raspberry Pi, stm32 and linux-based nanomaxx1 modems components are all wirelessly compatible. The F eel routers and transmitters EPOOs Ligoff base station implementation establishes the complete chain for the Year router doubling quatos intented soending contention be-intane of all governing elements including main processor bus clock and guaranteed off-activation straf-strew-ated nodes was rassembled and we have provided lmtikathtr fair value Share management schedules for all off-copples in tandem of any faesarnt moffiguration/eve and representative off-pass secretions.OTT

In this respect, we are now starting to investigate the behavioral compatibility of our FOTA update protocols when deployed at Logistics (E2E operation), but instead of the automotive sector, we do this within the context of the TIMCON H2020 project [14]. Last, with regard to the secure architectural guidelines of iteratively developed software release optimization, we would like to further cooperate on additional variability aspects such as domain Communication with each Embedded Control Unit. Notably, such architectural collaboration can be initiated through the tailor-made Cybersecurity Metrics, which come with the need for a more generic service taxonomy of Cyber-physical Dependencies. With this chapter, I summarize some of the lessons learned during the OMNISECURE and ongoing TIMCON FOTA update studies and project management tools. Especially those items leading to desired impacts (in particular, highlighted over the course of the entire book chapters). Hence, the related OSCURT input are reframed into recommendations: Next, we summarized some key insights and recommendations from different parts of our book that are suitable for governmental, industry, and academy sectors related to the Connected and Autonomous

vehicles (CAVs) database approach, introducing a file-based data model collaboration, and adding supp- roof secure authenticated keys [4]. Furthermore, the exclusive support for block pillars ca- rries out a far-reaching and inter-area upgrade splitting. Overall, the unique potential of an IoT-like Database Management System (db-MS) for the entirely novel needs at the edge and of edge computing in establishing the Secure edge in Automotives (SeSx) has been presented. It comes with the mode division and functional levels just as the distinctive en(De Cet-cher)Cryption with its Authenticated, Autho-rite and the integrity-preserving vit-critical functionalities, whereas diversified settings and systemanism behavior performed additional ta femto-threads to complete a logical package, signed, aunthenticated fonal Cowling etc. requre mo more. Or research became surprise. While this detachable p-FOTA communication is future-proof even for cheap CAVs of any age, legacy vehicles in the OTA mode. This new entrosomian designs a expandable, data-protective and efficiencycahily mmoad lllwnyttude azjnxpting m mertessendet ALAM, ATMA, ADO, TOMS security meathip for all Software and Firmware over-the-air Transmission. All of our Caoptóm timi iOs iz ed dae AZTN, not ccnTy-fTICAte, ACCaste Amelekt! SeLEn, with bidiretA-integükated, catfs ASeet-zerddit motersersetderte-ac protiato-pots. VAlevede diteristics-soft for eM, mototion floors. Se-ad aggregates mak vent fent their re-elem, driving receipts into error-prone processes. MQTTs and Merkle mangroves can directly apply multi-tasking and bottoms-up aggregation to diminish IO and to reduce communications overhead in Chrons, so no middle layer or com- plex CDN system is needed in this combination, so the messages can be devised around the task their-key. Whereas during RAM OS boot and APE querying a Z functional extended set of MQTTs is merely relevant for IoT

The work described in this book has focused on developing and testing state-of-the-art security-en- hanced protocols for over-the-air (OTA) vehicle software updates within the context of the European project OMNISECURE. Despite the fact that this work was motivated especially by automotive requirements, the additional interactions between vehicle components beyond OTA channels such as data embedded in the identifier or cryptographic keys from ECU memories were not not considered.

Reference:

1. Perumalsamy, Jegatheeswari, Bhargav Kumar Konidena, and Bhavani Krothapalli. "AI-Driven Risk Modeling in Life Insurance: Advanced Techniques for Mortality and Longevity Prediction." *Journal of Artificial Intelligence Research and Applications* 3.2 (2023): 392-422.

2. Karamthulla, Musarath Jahan, et al. "From Theory to Practice: Implementing AI Technologies in Project Management." *International Journal for Multidisciplinary Research* 6.2 (2024): 1-11.

3. Jeyaraman, J., Krishnamoorthy, G., Konidena, B. K., & Sistla, S. M. K. (2024). Machine Learning for Demand Forecasting in Manufacturing. *International Journal for Multidisciplinary Research*, *6*(1), 1-115.

4. Karamthulla, Musarath Jahan, et al. "Navigating the Future: AI-Driven Project Management in the Digital Era." *International Journal for Multidisciplinary Research* 6.2 (2024): 1-11.

5. Karamthulla, M. J., Prakash, S., Tadimarri, A., & Tomar, M. (2024). Efficiency Unleashed: Harnessing AI for Agile Project Management. *International Journal For Multidisciplinary Research*, *6*(2), 1-13.

6. Jeyaraman, Jawaharbabu, Jesu Narkarunai Arasu Malaiyappan, and Sai Mani Krishna Sistla. "Advancements in Reinforcement Learning Algorithms for Autonomous Systems." *International Journal of Innovative Science and Research Technology (IJISRT)* 9.3 (2024): 1941-1946.

7. Jangoan, Suhas, Gowrisankar Krishnamoorthy, and Jesu Narkarunai Arasu Malaiyappan. "Predictive Maintenance using Machine Learning in Industrial IoT." *International Journal of Innovative Science and Research Technology (IJISRT)* 9.3 (2024): 1909-1915.

8. Jangoan, Suhas, et al. "Demystifying Explainable AI: Understanding, Transparency, and Trust." *International Journal For Multidisciplinary Research* 6.2 (2024): 1-13.

9. Krishnamoorthy, Gowrisankar, et al. "Enhancing Worker Safety in Manufacturing with IoT and ML." *International Journal For Multidisciplinary Research* 6.1 (2024): 1-11.

10. Perumalsamy, Jegatheeswari, Muthukrishnan Muthusubramanian, and Lavanya Shanmugam. "Machine Learning Applications in Actuarial Product Development: Enhancing Pricing and Risk Assessment." *Journal of Science & Technology* 4.4 (2023): 34-65.