International Journal of Cyber Warfare and Terrorism
Volume 10 • Issue 2 • April-June 2020

The Security Aspects of Automotive Over-the-Air Updates

James Howden, De Montfort University, Leicester, UK

Leandros Maglaras, De Montfort University, Leicester, UK
https://orcid.org/0000-0001-5360-9782

Mohamed Amine Ferrag, Guelma University, Guelma, Algeria
https://orcid.org/0000-0002-0632-3172

ABSTRACT

Over-the-air (OTA) update is a method for vehicle manufacturers to remotely distribute maintenance updates, performance, and feature enhancements through the vehicle's lifespan. Recalls of vehicles cost the manufactures a lot of money. OTA solves the recall issue, while allowing consumers to pay for services and features via an update. The OTA ecosystem includes the coders who first developed the firmware, the 1st Tier suppliers, the vehicle manufacturers, and the vehicle itself. Currently, manufacturers designed the networks for speed and responsiveness, and not security. This article examines these elements and drills into the security available for each. The slowest and one of the most vulnerable parts of the system is the communications within the vehicle. The vehicle networks must ensure the integrity and authenticity of messages transmitted to guarantee software programmed onto ECUs are authorized and tamper-free. Specialist hardware within the vehicle makes this possible in an operation environment, such as hardware security modules.

KEYWORDS

Electronic Control Units, Automated Cars, Over-The-Air Updates, Connected Vehicles, Security, Cryptography, Hardware Security Module, Trust

1. INTRODUCTION

Modern connected vehicles contain up to 100 Electronic Control Units (ECUs) and up to 100 million of lines of code (Continental, 2019, Riggs et al., 2018, Petri et al., 2016, Holmes, 2018). The increase in functionality, i.e. advanced driving-assist system and autonomous driving, has increased the value of the vehicle's software.

With the introduction of new concepts, i.e. paying for vehicle functionality per feature aka Software-as-a-Service (SaaS), the automobile is evolving into a software-driven mobility service defined by the experience it delivers whilst transporting passengers (Windriver, 2018; Stevic, 2018). The consumer's desire to stream music, receive emails and social media posts has intensified this issue (Petri et al., 2016). The increase in software functionality means more potential for bugs, and so we have seen an increase in after sales software/firmware updates (Teraoka et al., 2016; Windriver, 2017).

Most vehicle software updates have occurred at dealerships either for a recall or an upgrade (Quain, 2018; Chowdhury, 2017; Continental, 2019). This can be inconvenient to the user, expensive for the dealership and affect the brand image (Windriver, 2017).

DOI: 10.4018/IJCWT.2020040104

The software industry has implemented over-the-air (OTA) updates for many years to fix bugs and make enhancements to software on computers, mobile devices, and even televisions (Chowdhury, 2017; Quain, 2018). OTA is wireless delivery of the software update to the vehicle (Rouse, 2018). In the automotive industry it is still rare. OTA on vehicles eliminates the need to take the vehicle to the dealership, protects brand-image and saves the waiting for updates (Riggs et al., 2018; Holmes, 2018).By 2022 the car-parc will contain hundreds of millions of connected vehicles, and consequently VMs would save $35 billion by using OTA rather than vehicle recalling (Neiger, 2016).

Some VMs have implemented software OTA systems but have concentrated on non-essential systems such as updates to the vehicle's infotainment systems, maps, and telematics software. This is known as SOTA. Some VMs, notably Tesla, have ventured into updating the firmware on individual ECUs including safety-critical systems, such as the brakes. This is known as FOTA (Quain, 2018; Kim & Park, 2018).

Even though OTA updates seem like a panacea, new vulnerabilities still occur. An attacker could take control if they introduced a malicious update to a vehicle, either at the back-end or in transit (Pedroza, 2011; Verma, 2018). History has shown software systems are most at risk to lapses in safety and security when undergoing change (Chowdhury et al., 2017). An OTA update is a substantial change; therefore, it is essential security is paramount. This paper explores OTA concept with an emphasis on security. Proceeding sections dive deeper into the OTA ecosystem, into the back-end systems, and the vehicle internals. Although many of the concepts discussed are in relation to software and firmware, this paper concentrates on the less trivial FOTA.

2. RELATED WORK

Recently several scholars have researched the area of security issues of smart cars (Maglaras, 2015) and especially those related to OTA. An OTA update system must be resilient to spoofing, tampering, repudiation, information-leakage, denial-of-service, and escalation-of-privileges attacks among others (Vasenev, 2019). Several security and privacy issues may arise from different parts of the ecosystem, like cloud, Service station, car or OEM backend and several security methods must be combined in order to counter those threats. The freshness of the update information also needs to be preserved in order to prevent replay attacks (Halder, 2019). Moreover, software distribution during OTA updates must be arranged in such a way that high security; low latency and continuous data protection are guaranteed.

Hash algorithms and digital signatures are some common methods that can be used in order to ensure authenticity and integrity of software updates, but more sophisticated methods were recently proposed (Zhou, 2019). Another promising idea is to use fog computing in smart vehicles (Fizza, 2019). Fog nodes are distributed in an ad-hoc fashion, and hence lead to a more reliable and robust system, without a single point of failure.

Having in mind to present a holistic model of a smart car that includes all different technologies that make the car 'smart' ENISA issued a report last year where relevant threats and cybersecurity risks pertaining to smart cars are analyzed and several security measures are proposed (Enisa, 2019). The report takes into account the particularities of this highly complex, heterogeneous and volatile environment and the fact that no modern system can be analyzed in isolation. In this report the interconnection of sensors, AI, machine learning algorithms, cloud computing and connectivity is presented and threats that come from all these components of smart car are analyzed. In one of the scenarios presented in the report a large-scale deployment of a rogue firmware after hacking OEM back-end servers affects OTA and consequently the proper behaviour of the vehicle. In this article we focus mainly on the OTA rather than the Wi-Fi and within a V2V/V2I ecosystem.

3. OTA

3.1 What is OTA?

OTA update is a method of remotely distributing maintenance updates, performance and feature enhancements throughout vehicle lifespan (Windriver, 2019). It allows VMs to manage upgrade process of each individual vehicle via cellular, Wi-Fi, Bluetooth or other connectivity. It allows the update of individual vehicle ECUs/systems (Khurram et al., 2016). Each ECU in the vehicle has firmware controlling its operation. Firmware is software that is semi-permanently placed in hardware (Flashrouters, 2017). Firmware is associated with embedded systems where hardware requires bespoke software.

ECU suppliers update VMs on firmware updates for their ECUs. Typically, VMs either combine a set of compatible updates together in service-packs, or as an individual update. Once a vehicle's suitability for update is established, update is downloaded to vehicle. The vehicle then installs updates when possible. Service-packs may contain other service-packs when a combination of updates is required (Petri et al., 2016; Balmus et al., 2017).

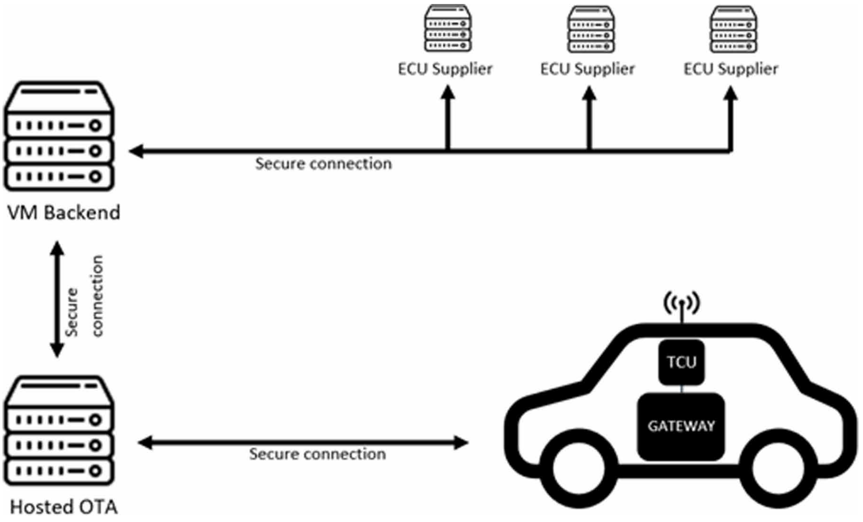
3.2 Significance of OTA

Rather than the customer needing to return vehicle to the dealership to receive update, VMs can push the update directly out to owner. One issue with the current method is some customers ignore the requests even after repeated requests (Quain, 2018). Recalls are expensive for VMs, with some VMs incurring costs of nearly a billion dollars (Riggs et al., 2018).

There are several advantages to OTA updates over traditional methods conducted in dealerships (Continental, 2019; Rouse, 2018, Windriver, 2017, 2019; Riggs et al., 2018):

- Improved customer experience and remove need car to visit dealership
- Customers receive latest features and applications
- Immediate update of security fixes
- VMs save money e.g. Tesla adapter-plug update
- Convenience functionality such as pause, resume, rollback and recovery
- New revenue streams through monetization of data analysis, enhanced features and performance.

Figure 1. Ecosystem



3.3 Current Solutions

Several VMs are currently offering an OTA service for their customers. Most of these VMs offer OTA for non-critical systems i.e. the infotainment system. Critical systems such as engine control and brakes are reserved for dealership updates. A few suppliers are starting to release full OTA solutions with full FOTA capacity such as Wind River and Harman (Stevic, 2018). A few OTA incidents have caused more issues than they have solved. Fiat Chrysler released an OTA update for its Uconnect infotainment system in 2017/2018 which caused the vehicle's head-unit into an infinite reboot cycle. A similar update by Lexus in 2016 caused the infotainment system to be unresponsive and 'bricked' (Golson, 2016). Tesla has made numerous OTA updates to their vehicles. Often with safety related items, such as brakes.

4. OTA ECOSYSTEM

There are two elements to the OTA ecosystem (Figure 1), back-end operations and vehicle operations. These two processes have numerous sub-processes. Securing the mechanism is tough as there are so many attack vectors to consider. The ecosystem general security perspective:

1) All connections between back-end systems and within the vehicle must be secure
2) Authenticate all entities within system
3) The transmission payload must be correct and tamper free
4) Once installed firmware must be cryptographically verified

The following sections describe how the elements of ecosystem operate and how security must be designed into system.

4.1 Back End

The ECU suppliers produce firmware updates. A long supply-chain of sub-tier suppliers may produce their hardware and software. The VMs collects relevant firmware updates from the ECU suppliers and collates them into a service-pack, or an individual update. The VMs may store the update on a hosted OTA server for distribution.

4.1.1 ECU Supplier

A vulnerable element of ecosystem due to large supply-chains involved in developing vehicle parts is ECU suppliers. Development of firmware occurs across numerous suppliers and geographical areas (Windriver, 2017). Suppliers must comply with all relevant information and cloud security standards and vehicle cyber security standards i.e. ISO/SAE 21434. There are several factors to consider:

- Protection of firmware Intellectual Property (IP)
- Insider actions i.e. malicious back-doors
- Recording UID of ECU to firmware version
- Secure coding methods:
    - Secure Code best practices (i.e. buffer-overflow attacks)
    - Authentication and authorisation of firmware contributors
    - Secure firmware and code storage accessed by VPN
    - Security of 3rd party software libraries (i.e. back-doors/bugs)

The firmware produced to accompany ECU must be cryptographically signed.

4.1.2 Vehicle Manufacturers

VMs must, in a similar fashion to their ECU suppliers, comply with all relevant information security, cloud security and vehicle cyber security standards, and independently audited. VMs must monitor for updates from supplier and deal with the event. An update provided by a supplier could ‘break’ another part of the vehicle and therefore VM must request further updates, potentially from a different supplier. The VM must monitor status of firmware on a vehicle ask the vehicle what firmware it is running and compare it with its list of known compatible firmware. The VM may consider hosting updates on a separate hosting site which may have built in security services.

4.1.3 Download to Vehicle

The downloaded update can take a few forms. It can be a full copy that just needs swapping into place. It can be a differential image where only the parts that have changed are transmitted. The latter’s benefit is a smaller update, although ECU must conduct work to re-assemble firmware (Balmus et al., 2017).

Mobile device management (MDM) describes software to manage mobile devices within a system, including updating. Open Handset Alliance Device Management (OMA-DM) is a protocol to communicate between back-end server and client and is vendor/device neutral. OMA-DM specifies a firmware-update-management-object (FUMO) and a high-level client/server interface for download, upload and status reporting (Doddapananeni et al., 2017; Oma, 2009; Windriver, 2011). PUSH method, used for resource-constrained devices, where server pushes data to client. PULL method involves server providing client with the data URL and managing process of download itself. PULL is suitable to more capable ECUs such as Gateway. OMA-DM supports PUSH and PULL, but PULL is typical download procedure (Doddapananeni et al., 2017).

The server provides a URL for client to PULL download down from it. Once complete the client replies with an acknowledgement. The server then requests for client to update the firmware. Once the client has completed the firmware update it sends an acknowledgement. The system must restart the process if failure occurs, this negatively impacts battery life. A known vulnerability around downloading using the PULL mechanism is to supply an incorrect URL to pull malicious firmware (Riggs et al., 2018).

OMA-DM communicates over HTTPS (OMA 2015), which utilises transport layer security (TLS). TLS utilises symmetric cryptography to encrypt transmitted data, using keys exchanged via asymmetric key-exchange (GDS, 2016). TLS supports Advanced Encryption Standard (AES) with 128/256bit keys. TLS uses certificates for server authentication. TLS supports Message Authentication Codes (MAC) within each transmission to prevent loss or interference. It utilises a shared secret-key and publicly-known algorithm to generate and check authenticity (Tutorialpoints, n.d.).

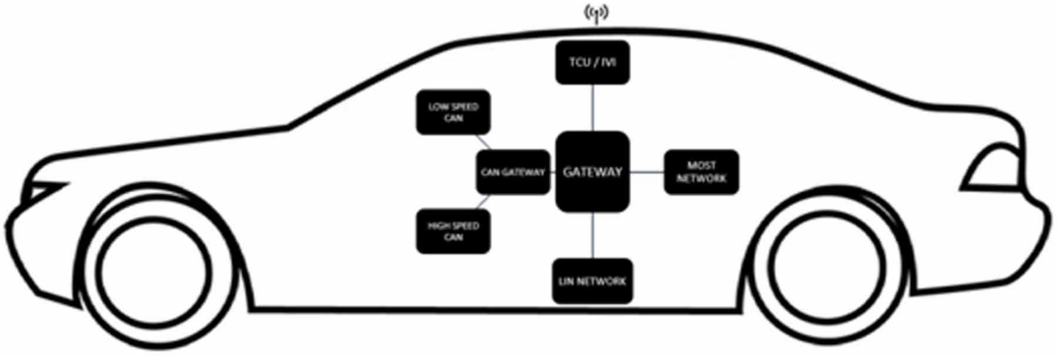
4.2 Vehicle Network

The multitude of ECUs within the vehicle are connected to multiple networks with different properties. Networks include (Verma, 2018; Kim et al., 2018; Ryu et al., 2008) (Figure 2):

- CAN (Controller Area Network) and CAN-FD (higher bandwidth)
- MOST (Media-Oriented-System-Transport)
- LIN (Local-Interconnect-Network)
- FlexRay
- Ethernet
- K-Line (ISO 2012)

Two ECUs involved in OTA process is telematics control unit (TCU) and gateway. TCU provides communications functions via mobile phone network. It also provides emergency call features. in-

Figure 2. Typical network



vehicle infotainment system also provides communications with other protocols such as Bluetooth (Balmus et al., 2017).

The gateway functions as a data router and a central computing unit between different networks. The gateway often contains the vehicle's central storage used within OTA process. For vehicles with OTA, Gateway contains the update server which uses TCU to download and store updates. The gateway will also contain the DM client as described in previous section. The update server resides in gateway because it has parallel access to other networks and simultaneously updates different ECUs. The gateway performs security tasks i.e. firewalling and bus separation (Continental, 2019; Balmus et al., 2017). The update server subsequently distributes individual updates to ECUs.

The Gateway must include fast network connections to TCU and other networks, and flash memory storage for updates/backup (Balmus et al., 2017). Due to speed restrictions networks, the bottleneck of OTA process is sending update from central storage to ECUs. If 10% of CAN bus is available for update, then approximately 98% of time of update is updating ECUs (Teraoka et al., 2016). Other factors can also affect performance of OTA such as ability to transfer the firmware in parallel, and method Gateway opens and distributes service-packs (Balmus et al., 2017).

4.2.1 ECU

The vehicle's electronic features are controlled by ECUs. Modern ECUs contain FLASH memory to store firmware and persistent data. The ECU has a FLASH Boot loader program, often resident of ECU ROM, responsible for activities such as booting-up system and loading FLASH contents into RAM. The boot loader has functions to erase/read/write to FLASH and coordinate the update process. The boot loader has functionality to communicate with the Gateway with protocols such as UDS (see below). Reprogramming of ECU goes through the stages of making the network conditions correct, programming, and then validating/resetting ECU (Shi et al., 2015; Embitel, n.d.).

Balmus et al. (2017) describes two approaches to ECU reprogramming. The traditional approach to updating an ECU is to use central storage to send updated firmware to ECU FLASH in small chunks in one step via internal network. The ECU needs to be ready for update to occur, vehicle contains enough power, and back-up stored in central storage. OTA also uses the method to update ECUs via the Gateway with central storage.

This method is slow, and ECU are uninterruptable and therefore inoperable. Balmus et al. suggests time comparisons of 4MB ECU update in Table 1. Using a CAN, updating 20 ECUs in a service-pack the update would make the vehicle uninterruptable for 1h40 minimum.

Another update method is A/B swap method. If within normal operation the ECU is booting up firmware in FLASH storage A, this process copies update from central storage to FLASH storage B and tells ECU boot loader to boot from firmware in B on next power up. Benefits include little

Table 1. Traditional update

Protocol (nominal data rate)	Realistic transfer rate	Time of transfer
CAN 500 kbits/s	16.8 kbytes/s 50% bus load	250s
CAN-FD 2Mbits/s	88.5 kbytes/s 50% bus load	47.4
FlexRay 10Mbits/s	300 kbytes/s 50% dynamic segment usage	13.6s
Ethernet 100baseT	5 to 10 Mbytes/s	0.8 – 1.6s

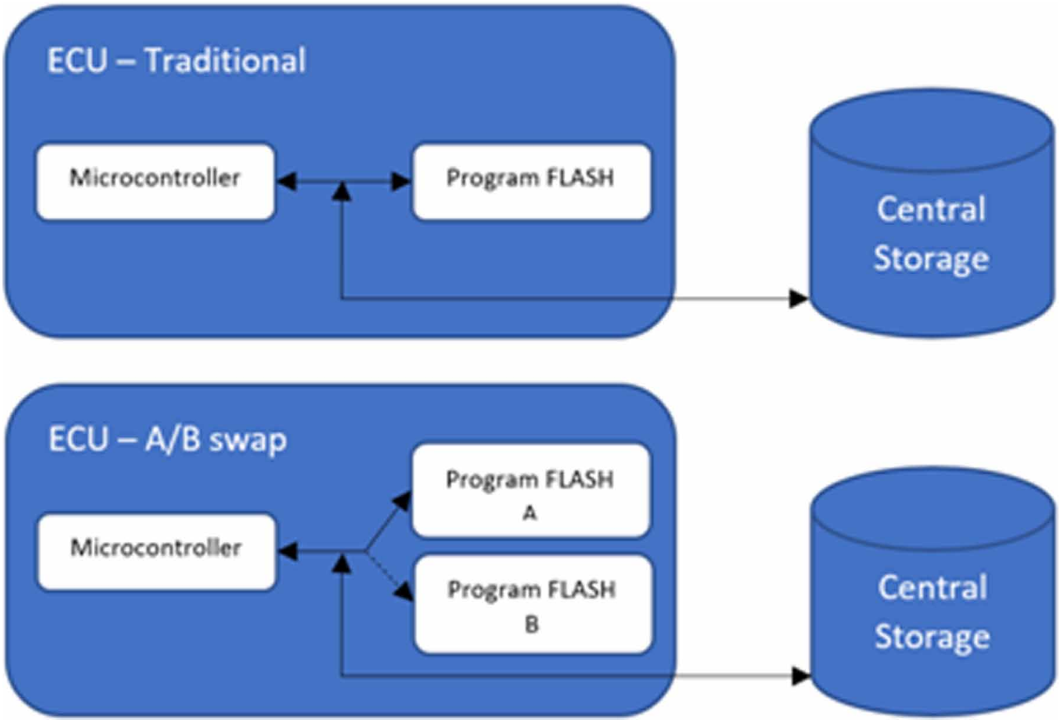
downtime for the power up/down cycle and back-up firmware stored (Balmus et al., 2017). The downside is internal on-board FLASH is expensive and this method would need double, and few ECUs currently have this capability. Figure 3 shows the approaches.

From a security perspective, transmission of small data chunks from central storage to ECU FLASH storage needs protection against eavesdropping. Once in place, update needs verification before activation. Multiple signatures can accompany firmware, where Gateway checks outer and target ECU checks inner. A failure causes Gateway to swap back to previous firmware from central storage (Balmus et al., 2017).

4.2.2 UDS

A typical CANframe contains a maximum of eight bytes of data. CANs networks are fast and busy with small packets (WCT, n.d.). Updates are always going to be much larger than eight bytes. Unified Diagnostics Services (UDS) is a protocol used in application layer (OSI network-stack) and has

Figure 3. Update methods



upload/download functionality (Shi et al., 2015). UDS is comprised of several vertically stacked protocol layers (Table 2).

In the CAN example, the bottom layer is low-level drivers responsible for CAN frame reception. The middle-layer is used to pack/unpack data up or down to other layers. At this level all transmitted data over 8 bytes is sent in a series of multi-frame CAN packets including timing and error handling. This level handles maximum payload of 4095 bytes. The highest level is the UDS application level where relevant services are carried out (Yu & Luo, 2016), i.e. *ECUReset*, *SecurityAccess*, *RequestDownload* and *TransferData* (Shi et al., 2015).

Depending upon network architecture the appropriate application layer is utilised (ISO 2012):

- CAN (UDSonCAN/ISO14229-3)
- FlexRay (UDSonFR/ISO14229-4)
- Ethernet (UDSonIP/ISO14229-5)
- K-Line (UDSonK-Line/ISO14229-6)

Each application layer calls their relevant stacks. The stacks are unified and present ECUs with the same interface regardless of network architecture underneath. UDS commands are used to distribute a chunked update file over a vehicle regardless of network architecture. Many dealership diagnostic/update tools utilise UDS to update ECU firmware. Many modern ECUs support UDS. It is also used to update from the Gateway via OTA (Yu & Luo, 2016).

UDS implements seed/key security for mutual authentication. To access security-critical functionality, i.e. updating firmware (request download & transfer data), Gateway requests security access from target ECU. The Gateway requests a seed from the ECU. Once received the Gateway uses the shared cryptographic algorithm to calculate a key to send back to the ECU. If ECU receives expected key it grants security access. The UDS standard doesn't specify algorithm, or seed/key length, or whether seed needs to be static or alternating (Ring et al., 2014). Some systems implement a challenge-key system built into the back end (Brirus, 2017).

4.2.3 Cryptography

Cryptography counteracts threats such as firmware IP, data theft, sensor manipulation, fraudulent feature activation, firmware downgrade and ECU counterfeiting (Petri et al., 2016).

Symmetric cryptography uses a shared secret key. For example, in production/assembly VM supply a key, such as AES 128/256 bit key, to the ECU. The VM stores the same key in secure storage (HSM). The back end/Gateway encrypts information with the key, later decrypted by the ECU with embedded key. Other systems implement supplying the symmetric secret-keys later. The main drawback is keeping the key secret. Asymmetric cryptography, known as public-key cryptography, uses a private-key and a public-key. For example, in production/assembly the VM supplies private-key and public-key to the ECU. The back end/Gateway encrypts information using the ECU's public-key, that the ECU decrypts with its own private-key. Asymmetric cryptography's benefit is no shared secret. Asymmetric is several orders of magnitude slower than symmetrical. Many asymmetrical systems use

Table 2. UDS CAN-Stack

Application Layer ISO 14229
Network Layer ISO15765
CAN Driver Layer ISO11898

symmetric system once the secret-key is shared using asymmetrical (SSL2BUY, n.d.). Asymmetric examples are Rivest-Shamir-Adleman (RSA) or Elliptic-Curve-Cryptosystems (ECC) (BALMUS et al., 2017). ECC has a significantly smaller key, therefore less RAM/CPU overhead during operation. ECC is more susceptible to an attack from a quantum computer (Squemish Ossirage, 2018).

Signing is achieved using Asymmetric Cryptography. For example, the back end/Gateway hashes information and encrypts the hash with their private-key. The information and encrypted hash are sent to the ECU. The ECU uses public key to decrypt hash, then hashes the information itself. The ECU checks the information integrity by comparing the decrypted hash with the new hash (INSTANTSSL, n.d.). If information stays encrypted between the back-end and ECU, there is a performance benefit as gateway does not need to decrypt. If update resides in gateway unencrypted, then gateway can sign the update for integrity and authentication. If unencrypted, the gateway sets up keys and encrypt the data blocks within transit to ECU. The gateway initiates contact with the ECU bootloader who decrypts each encrypted block sent to it. Encryption ensures data confidentiality.

Balmus et al. (2017) conducted speed comparison studies on encrypted and non-encrypted messages. The studies showed that using hardware to conduct cryptographical services was faster by two orders of magnitude. Table 3 shows the encryption and hashing speed in seconds on CAN/CAN-FD compared to the update frames per second sent of 1MB update. The table suggests that OTA isn’t practical under CAN/CAN-FD whilst the vehicle is operated (low frame rate). If the vehicle was moving and one frame every five seconds sent, then it would take 9.7days to complete.

Many ECUs control safety-critical systems, require real-time responses, so cannot be impaired by cryptographic obligations or the network being overwhelmed with update packets (Jordan, 2018).

4.2.4 HSM

A hardware security module (HSM) is a security device added to a system to generate, manage and securely store cryptographic material (Gibson, 2015). Having a separate HSM module onboard the vehicle allows the HSM to encrypt, decrypt or sign data without having to export the keys from the HSM (Gibson, 2015; Petri et al., 2016). ECUs have real-time application requirements that need balancing with the ECU security functions. HSMs address this issue by offloading the crypto functionality onto a separate processor/unit.

Although expensive to implement, through dedicated RAM, FLASH and peripherals like timer, hardware accelerators for algorithms and random-number generators, dedicated HSMs are far more secure than co-processors. ECUs can request cryptographical operations without the keys leaving the HMS (Jordan, 2018). HSMs support the secure-boot functionality, where ECUs bootloader requests the HSM to check the firmware integrity and trust level before code execution (Balmus et al., 2017).

Table 3. encryption and hashing speed in seconds on CAN/CAN-FD

Bus Utilisation Frame /s	Classical CAN (s) for 1MB		CAN-FD (s) for 1MB	
	AES-128	SHA256	AES-128	SHA256
0.2	839680	860160	20480	20480
0.5	335872	344064	8192	8192
1	167936	172032	4096	4096
2	83968	86016	2048	2048
5	33587	34406	819	819
10	16793	17203	409	409
20	8396	8601	204	204
50	3358	3440	-	-

4.2.5 TPM

A trusted platform module (TPM) is a dedicated hardware crypto processor built into the ECU. TPMs often have encryption keys burnt into them for asymmetric encryption. Some TPMs, like HSMs, can generate and securely store encryption keys (Gibson, 2015). The Trusted Computer Group (TCG 2015) consortium produced international standard (ISO-11889) for TPM and released TPM2.0 library specification in 2015.

TPM2.0 supports RSA 2048/2072 bits and SHA512 hashing, and symmetric encryptions such as 128bit AES and Hashed-key MAC (HMAC).TPM2.0 supports the upgrading of encryption algorithm, which is essential for vehicles with 10-15 year lifecycles when the original algorithms become obsolete (Petri et al., 2016).

Hardware TPMs are expensive. Under TPM2.0, It is also possible to implement the TPM within firmware in virtual environment, supported by a hardware isolated execution environment. Figure 4 shows boot sequence. The hardware calls the bootloader which boots a hypervisor that creates two environments. One being the TPM in secure OS, the second being ECU application in a richer OS. The TPM runs in secure OS in an isolated hardware area inaccessible to the application (Petri et al., 2016). This solution is cost-effective but requires increased RAM/ROM/CPU.

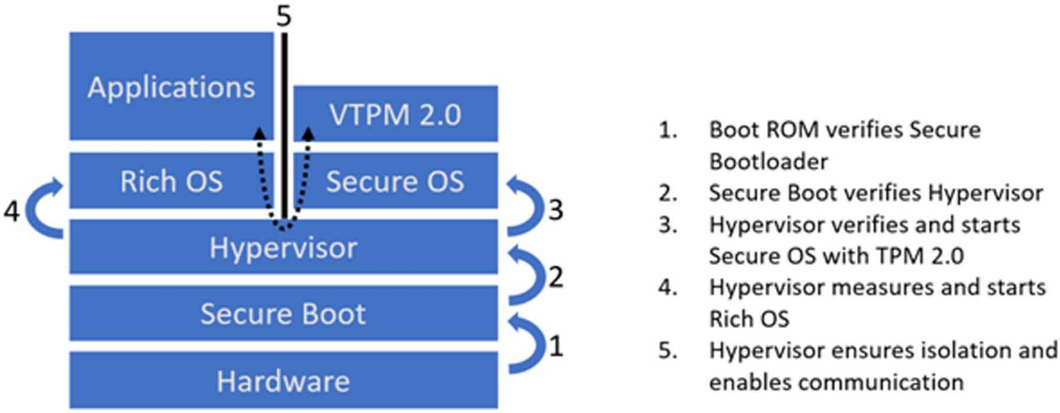
The EVITA project (2008) was co-funded by the EU, to standardise the on-board vehicle networks architecture to ensure security-relevant components were protected against tampering. EVITA defines three levels of TPM/HSM (Petri et al., 2016):

- Full - Rich TPM – i.e. Infotainment systems / Gateway with smart phone level computing power. Corresponds to full TPM2.0 specification.
- Medium - Engine control units with moderate computing power
- Light - Thin TPM – All ECUs with scarce ROM/RAM resources running non-conventional OS. i.e. sensors/actuators.

4.2.6 SHE

The security hardware extension (SHE) is an open standard used by Audi and BMW. SHE specification includes implementation of a dedicated securely firewalled secure zone built into the same silicon as the main microcontroller CPU and memory. Within the secure zone is an AES hardware accelerator. SHE defines software functions and an API that allows a secure zone to co-exist within any ECU. The area handles storage and key management. It handles authentication, encryption and decryption algorithms. Whereas TPM2.0 supports asymmetric cryptography, SHE defines a protocol for securely

Figure 4. Virtual TPM (Petri et al., 2016)



1. Boot ROM verifies Secure Bootloader
2. Secure Boot verifies Hypervisor
3. Hypervisor verifies and starts Secure OS with TPM 2.0
4. Hypervisor measures and starts Rich OS
5. Hypervisor ensures isolation and enables communication

changing symmetrical keys. SHE uses AES for signing, authentication and encryption (Soja, 2014), making a lighter weight and faster solution. SHE corresponds to light EVITA (Fujitsu, 2012).

4.2.7. AUTOSAR

VMs and suppliers developed Automotive Open System Architecture (AUTOSAR) to standardise the software architecture for automotive ECUs. AUTOSAR is based on a three-layered architecture model (Ryu et al., 2008; Embitel, n.d.), designed to separate software function from ECU hardware through a strictly defined middle layer:

1. Basic Software (BSW) - ECU specific modules
2. Runtime environment (RTE) – Middleware between AUTOSAR layers
3. Application Layer – Various applications specific software components.

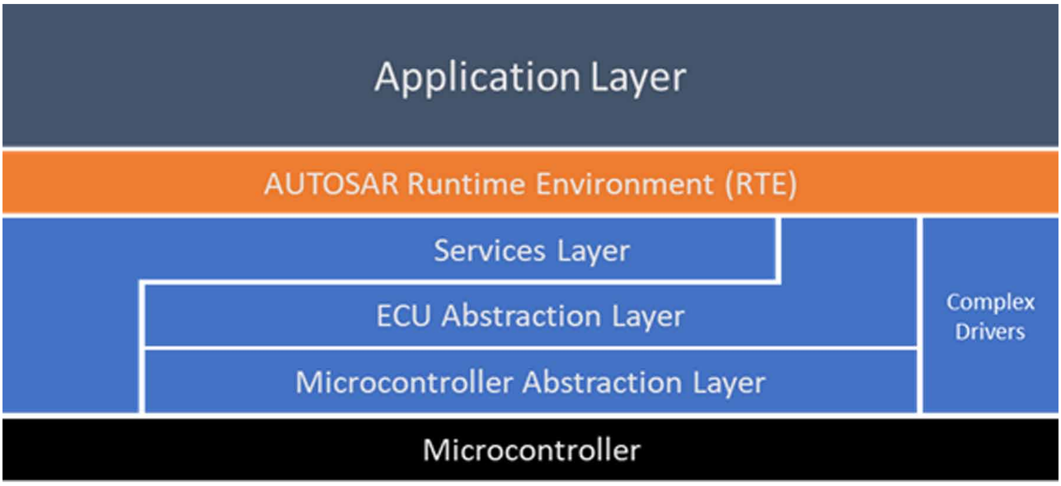
AUTOSAR implements functionality standardisation therefore software from various vendors can communicate. Many current firmware stacks are AUTOSAR compliant (Stevic, 2018). AUTOSAR software also allows for standardisation of software modules on ECUS connected to different networks architecture such as CAN, LIN, FlexRay (Ryu at al., 2008) (Figure 5).

AUTOSAR introduces Secure Onboard Communication (SecOC) concept to test the single Protocol Data Unit (PDU) authenticity, i.e. CAN packet, to detect replay, spoofing and tampering attacks. The SecOC generates a MAC in the PDU. AUTOSAR adds a freshness to PDU to protect against replay attacks. To fit into 8-byte CAN frame along with payload, AUTOSAR truncates MAC and freshness values (27bits for CAN), hence it is less secure. This security method is more suited to CAN-FD or Ethernet where full-size MACs are generated (Islinger et al., n.d.).

AUTOSAR Crypto Service Manager (CSM) provides cryptographic services to ECU application. CSM supports Symmetric and Asymmetric encryption, Random number generation, MAC verification, and key management. CSM utilises software and hardware cryptographic implementation, and lightweight TPM Secure Hardware Extension (SHE) (Vector, 2015).

The CSM functionality runs asynchronous to the ECU application, although must queue for a slot within Microcontroller (round-robin) (Jordan, 2018; Vector, 2015). Cryptographic services are processor heavy and low priority against main application processes. Therefore, ECUs that need to

Figure 5. AUTOSAR Architecture (Embitel, n.d.)



produce a real-time response, a cryptographic solution needs hardware support i.e. HSM (Jordan, 2018).

In 2017, AUTOSAR released Adaptive platform, designed to allow dynamic linking between services and clients during ECU runtime producing flexibility. Dynamic interfaces support implementation of autonomous driver functions, OTA and media streaming (Hellgren, 2018). Adaptive AUTOSAR provides a full OTA via update configuration management (UCM) module adaptive AUTOSAR also supports Ethernet with advanced cryptography i.e. TLS (Stevic, 2018).

5. LEGISLATION

Within Europe, type approval (TA) is confirmation that the vehicle production will meet specified performance standards and is based on UN regulations and EC Directives (VCA, 2019).OTA update can alter vehicle performance/capability after its initial certification for TA, which would constitute a new vehicle item of equipment subject to approval and so need reassessment (UN SECRETARIAT, 2016).

Under UNECE WP29 draft recommendations (UNECEWP29, 2018) OTA post TA:

1) VMs shall assess whether update will impact TA compliance and document result;
2) If update does not impact TA, e.g. bug fixes, VM conducts update in a safe secure manner without contacting TA authority;
3) If update impacts TA then VM contacts relevant authority for an extension or new certification.

6. DISCUSSION

A set of principles for an ideal OTA system as follows (Verma, 2018; Windriver, 2017; Quain, 2018; Petri et al., 2016; Teraoka et al., 2016).

VMs and suppliers must:

- Maintain chain-of-trust ensuring Software from trusted sources and validated across chain
- Cryptographic Key maintenance and confidentiality ensured
- Accredited to information and cloud security standards with independent audits
- Maintain back-end servers:
  - Protect against IP breach and interference
  - Protect against firmware downgrades
  - Protection against Denial-Of-Service attacks
  - Monitor against anomalies and attacks
  - Log/Audit events, errors and irregular behaviour
- Follow industry standards/protocols for in-vehicle and back-end
- Firmware:
  - Designed with security in mind
  - Cryptographical signed, potentially multiple times for different ECUs
  - Incremental/Differential updates to save bandwidth

Vehicles must:

- Mutually authenticate with back end, along with individual ECUs within vehicle
- Ability to recover from partial/failed updates i.e. Gateway returning ECU back to its last working state
- Implement Secure-boot procedure for integrity check of firmware before execution

- Constructed of vehicle networks utilising protocols that support encryption and cryptographic integrity and authentication of messages
- Implement ECUs with enough dedicated resources so security does not concede to application for CPU/Memory
- Future-proofing cryptographic services. NIST stated in 2015 the minimum for Symmetric key size must be 112bits (128bits typical usage). Asymmetrical must be minimum of RSA 2048bits and ECC 224bits. With threat of quantum computing looming, these may need to be sufficiently higher in the future. Security measures, such as UDS Seed and Key must be implemented with futureproofing in mind.
- Cryptographic Key maintenance and confidentiality ensured

There is little standardisation across industry. Although industry has produced standards, i.e. AUTOSAR and TPM2.0, adoption is far from standard. Due to system and vehicle capability, VMs seem to be implementing some and not all the principles above.

Integrity of the firmware update is important as it detects tampering. Equally authentication of OTA process is important for the same reason. Both must be secured with cryptographical services. Confidentiality is of less importance throughout the process. It is important to protect IP against snooping but not as important as integrity of update, which indicates a malicious change to operation of a safety critical ECU.

A full encryption system for OTA is also a demanding situation for VMs. They can use unique private/public keys or shared secret keys for every single ECU and have the issue of managing all information or have reduced strength of encryption by using one private/public or shared secret over all ECUs.

7. CONCLUSION

This paper examined security elements around Over-the-air updates. Overall OTA is very positive for the security of the whole CAV environment but comes at a cost. Manufacturers must design and build vehicles where entire OTA mechanism is secure throughout, which requires resources and effort.

A significant issue with CAN bus is it just not sufficient as a modern automotive network architecture, due to lack of available speed and bandwidth. CAN-FD is better as each packet has room for cryptography, but even better would be automotive ethernet with full-blown TLS 1.3 implemented.

Currently 4G download speed is approximately 21Mbps (2.6MB/s), average Wi-Fi is approximately 46Mbps (5.8MB/s) and 5G will provide a minimum of 50Mbps (6.3MB/s) (Jackson, 2018; 4G, 2019; Hill, 2019). With current vehicle networks, the bottleneck is internal vehicle networks. Once faster vehicle networks are standard, this will reverse.

The encryption described is already getting old and less secure as computing power to brute-force increases. Quantum Key Distribution (QKD) is an encryption key generation method that relies on light's quantum properties. ECUs will detect an interception of a transmission by reading quantum states of the key, so is secure from interference. QKS works in tandem with traditional methods only, and overhead is significant (Soja, 2014).

Automotive OTA is an important subject. This paper focused on the matter and has also touched upon other subjects as downloading from Wi-Fi and within a V2V/V2I ecosystem. Regardless of apparent issues with implementing OTA on vehicles, the benefits in the long-term far outweigh the disadvantages it may bring. It is imperative all parties involved collaborate on hardware, software and protocols standardisation to develop and implement the highest level of security and protection.

Increased connectivity of Smart cars along with advanced automation methods, expose them to several crucial cyber threats. Those threats may directly target smart cars or come from assets that are directly or indirectly connected to them such as RSUs, traffic signs/lights or even remote servers of the OEM or third-party service providers. The current article focuses on the security aspects of OTA and proposes a set of principles for an ideal OTA system. Those principles include cryptographic Key maintenance, cloud security, secure networking protocols, recovery mechanisms and many other elements that prove that the security of a smart car demands a holistic approach,

REFERENCES

4G. .2019). *Vodafone 4G Coverage and Network Review.* 4G.co.uk. Retrieved from https://www.4g.co.uk/vodafone-4g-network-summary/

Balmus, A., Freiwald, A., & Wunderlich, C. (2017). *Over the Air Software Update Realization within Generic Modules with Microcontrollers Using External Serial Flash*. SAE. doi:10.4271/2017-01-1613

Brirus. (2017). *Example Seed and Key Algorithm.* StackExchange. Retrieved from https://security.stackexchange.com/questions/152270/example-seed-and-key-algorithm

Chowdhury, T., Lesiuta, E., Rikley, K., Lin, C.W., Kang, E., Kim, B., Shiraishi, S., Lawford, M., & Wassyng, A. (2017). Safe and Secure Automotive Over-The-Air Updates.

Continental. (2019). *Over-the-air Updates.* Continental AG. Retrieved from https://www.continental-automotive.com/en-gl/Passenger-Cars/Interior/Software-Solutions-and-Services/Over-the-air-Updates-for-multiple-electronic-contr

Continental. (2019). *Gateways.* Continental AG. Retrieved from https://www.continental-automotive.com/en-gl/Passenger-Cars/Interior/Control-Units/Gateways

Doddapananeni, K., Lakkundi, R., Rao, S., Kulkarni, S., & Bhat, B. (2017). Secure FoTA object for IoT. Wireless Innovation Networking Group. In *Proceedings of the 2017 IEEE 42 Conference on Local Computer Networks Workshops*. IEEE Press.

Embitel. (n.d.). *Decoding the “Component Concept” of the Application Layer in AUTOSAR.* Embitel.com. Retrieved from https://www.embitel.com/blog/embedded-blog/decoding-the-component-concept-of-the-application-layer-in-autosar

Embitel. (n.d.). *Understanding What is a Flash Bootloader and the Nuances of an Automotive ECU Re-programming*. Embitel.com. Retrieved from https://www.embitel.com/blog/embedded-blog/what-is-flash-bootloader-and-nuances-of-an-automotive-ecu-re-programming

European Union Agency for Cybersecurity (ENISA). (n.d.). Good practices for security of smart cars. doi:10.2824/17802

Evita. (2008). *Objectives.* EVITA Fraunhofer SIT. Retrieved from https://www.evita-project.org/objectives.html

Fizza, K., Auluck, N., Azim, A., Maruf, M. A., & Singh, A. (2019, December). Faster OTA Updates in Smart Vehicles using Fog Computing. In *Proceedings of the 12th IEEE/ACM International Conference on Utility and Cloud Computing Companion* (pp. 59-64). Academic Press. doi:10.1145/3368235.3368842

Flashrouters. (2017). *What is the difference between firmware and software.* FlashRouters LLC. Retrieved from https://www.flashrouters.com/blog/2011/11/01/what-is-the-difference-between-firmware-and-software/

Fujitsu. (2012). SHE Secure Hardware Extension. Data Security for Automotive Embedded Systems. In *Proceedings of the Workshop on Cryptography and Embedded Security Embedded World*. Academic Press.

GDS. (2016). *Guidance: Using Transport Layer Security (TLS) in your organisation.* Government Digital Service GOV.UK. Retrieved from https://www.gov.uk/government/publications/email-security-standards/transport-layer-security-tls

Gibson, D. (2015). *TPM and HSM Hardware Encryption Devices.* Get Certified Get Ahead. Retrieved from https://blogs.getcertifiedgetahead.com/tpm-hsm-hardware-encryption-devices/

Golson, J. (2016). *Many Lexus navigation systems bricked by over-the-air software update.* The Verge. Retrieved from https://www.theverge.com/2016/6/7/11879860/lexus-navigation-broken-software-update-bug

Halder, S., Ghosal, A., & Conti, M. (2019). Secure OTA Software Updates in Connected Vehicles. *Survey (London, England)*.

Hellgren, H. (2018). *Adaptive AUTOSAR in a nutshell.* Hackernoon. Retrieved from https://hackernoon.com/adaptive-autosar-in-a-nutshell-1cc609c1c5f5

Hill, S. (2019). *5G vs Wi-Fi: How they're different and why you'll need both.* Digital Trends. Retrieved from https://www.digitaltrends.com/mobile/5g-vs-wi-fi/

Holmes, F. (2018). Over-the-air update moving from 'nice to have' to 'vital.' Automotive World. Retrieved from https://www.automotiveworld.com/articles/over-the-air-updates-moving-from-nice-to-have-to-vital/

Holmes, F. (2019). *Premium connected car services must be secured. The automotive industry's innovate or die attitude to in-vehicle technology has cyber security experts worried, writes Freddie Holmes*. Automotive World. Retrieved from https://www.automotiveworld.com/articles/premium-connected-car-services-must-be-secured-2/

Instantssl. (n.d.). *What is a Digital Signature?* instantSSL. Retrieved from https://www.instantssl.com/digital-signature

Islinger, T., Mori, Y., Neumuller, J., Prisching, M., & Schmdit, R. (n.d.) *AutosarSecOC for CAN-FD.* DENSO AUTO. Retrieved from https://can-newsletter.org/uploads/media/raw/d904c90ba599c668e9758ae558dcb845.pdf

ISO. (2012). *Road Vehicles – Unified Diagnostic Services (UDS) – Part 4: Unified diagnostic services on FlexRay implementation (UDSonFR).* Organisation Internationale de Normalisation. Retrieved from https://www.iso.org/obp/ui/#iso:std:iso:14229:-4:ed-1:v1:en

Jackson, B. (2018). *An overview of TLS 1.3 – Faster and more secure.* Kinsta.com. Retrieved from https://kinsta.com/blog/tls-1-3/

Jackson, M. (2018). *Ofcom 2018 Study – Average Home Broadband Speeds Hit 46.2Mbps.* ISPReview. Retrieved from https://www.ispreview.co.uk/index.php/2018/05/ofcom-2018-study-average-home-broadband-speeds-hit-46-2mbps.html

Jordan, T. (2018). How hardware security modules enable AUTOSAR. Embedded.com. Retrieved from https://www.embedded.com/design/safety-and-security/4460819/How-hardware-security-modules-enable-AUTOSAR

Khurram, M., Kumar, H., Chandak, A., Sarwade, V., Arora, N., & Quach, T. (2016). Enhancing Connected Car Adoption: Security Framework. In *Proceedings of the 2016 International Conference on Connected Vehicles and Expo (ICCVE).* Academic Press. doi:10.1109/ICCVE.2016.5

Kim, B., & Park, S. (2018). ECU Software Updating Scenario Using OTA Technology through Mobile Communication Network. In *Proceedings of the 2018 IEEE 3rd International Conference on Communication and Information Systems*. IEEE Press.

Maglaras, L. A. (2015). A novel distributed intrusion detection system for vehicular ad hoc networks. *International Journal of Advanced Computer Science and Applications*, *6*(4), 101–106.

Neiger, C. (2016). *Carmakers Could Save $35 Billion in 2022 by Doing This.* The Motley Fool. Retrieved from https://www.fool.com/investing/general/2016/04/30/carmakers-could-save-35-billion-in-2022-by-doing-t.aspx

NIST. (2015). *Recommendation for Key Management. Part 3: Application-Specific Key Management Guidance.*

OMA. (2009). *Firmware Update Management Object version 1.0.2 28 Aug 2009.* Open Mobile Alliance. Retrieved from http://www.openmobilealliance.org/release/FUMO/V1_0_4-20090828-A/OMA-TS-DM_FUMO-V1_0_2-20090828-A.pdf

OMA. (2015). *OMA Device Management Protocol.* Open Mobile Alliance. Retrieved from http://www.openmobilealliance.org/release/DM/V2_0-20150122-C/OMA-TS-DM_Protocol-V2_0-20150122-C.pdf

Pedroza, G., Idrees, M. S., Apvrille, L., & Roudier, Y. (2011, September). A formal methodology applied to secure over-the-air automotive applications. In *Proceedings of the 2011 IEEE Vehicular technology conference (VTC Fall)* (pp. 1-5). IEEE.

Petri, R., Springer, M., Zelle, D., McDonald, I., Fuchs, A., & Krauß, C. (2016). Evaluation of lightweight TPMs for automotive software updates over the air.

Quain, J. (2018). With benefits – and risk – software updates are coming to the car. Digital Trends. Retrieved from https://www.digitaltrends.com/cars/over-the-air-software-updates-cars-pros-cons/

Riggs, C., Rigaud, C. E., Beard, R., Douglas, T., & Elish, K. (2018). A Survey on Connected Vehicles Vulnerabilities and Countermeasures. *Journal of Traffic and Logistics Engineering*, *6*(1).

Riggs, C., Rigaud, C. E., Beard, R., Douglas, T., & Elish, K. (2018). A Survey on Connected Vehicles Vulnerabilities and Countermeasures. *Journal of Traffic and Logistics Engineering*, *6*(1).

Ring, M., Rensen, T., & Kriesten, R. (2014). Evaluation of Vehicle Diagnostic Security – Implementation of a Reproducible Security Access. In *SECUREWARE 2014: The Eight International Conference on Emerging Security Information, Systems and Technologies. Academic Press.*

Rouse, M. (2018). *OTA update (over-the-air update).* TechTarget. Retrieved from https://searchmobilecomputing.techtarget.com/definition/OTA-update-over-the-air-update

Ryu, H. K., Cho, S. R., & Piao, S. (2008). The design of remote vehicle management system based on OMA DM Protocol and AUTOSAR S/W Architecture. In *Proceedings of the International Conference on Advanced Language Processing and Web Information Technology 2008*. Academic Press.

Schmitt, B. (2019). Why Haven't Over-The-Air Updates Taken Over The Auto Industry? TheDrive.com. Retrieved from https://www.thedrive.com/tech/26679/why-havent-over-the-air-updates-taken-over-the-auto-industry

Shi, G., Ke, Z., Yan, F., Hu, J., Yin, W., & Jin, Y. (2015). A Vehicle Electric Control Unit Over-the-Air Reprogramming System. In *Proceedings of the 2015 International Conference on Connected Vehicles and Expo (ICCVE). Academic Press.* doi:10.1109/ICCVE.2015.21

Soja, R. (2014). *Automotive Security: From Standard to Implementation.* NXP Semiconductors. Retrieved from https://www.nxp.com/docs/en/white-paper/AUTOSECURITYWP.pdf

Squemish ossirage (SO). (2018). *How effective is quantum computing against elliptic curve cryptography?* CRYPTOGRAPHY StackExchange. Retrieved from https://crypto.stackexchange.com/questions/59770/how-effective-is-quantum-computing-against-elliptic-curve-cryptography

SSL2BUY. (n.d.). *Symmetric vs Asymmetric Encryption – What is the differences? SSL2BUY.* Retrieved from https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

Stević, S., Lazić, V., Bjelica, M. Z., & Lukić, N. (2018, September). IoT-based Software Update Proposal for Next Generation Automotive Middleware Stacks. In *Proceedings of the 2018 IEEE 8th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)* (pp. 1-4). IEEE.

Trusted Computer Group (TCG). (2015, June 29). *Trusted Computing Group TPM 2.0 Library Specification Approved as an ISO/IEC International Standard.* Retrieved from https://trustedcomputinggroup.org/trusted-computing-group-tpm-2-0-library-specification-approved-isoiec-international-standard-date-published-june-29-2015/

Teraoka, H., Nakahara, F., & Kurosawa, K. (2016). Incremental Update Method for Resource-Constrained In-vehicle ECUs. In *Proceedings of the 2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE Press.

Tutorialpoints. (n.d.). *Message Authentication Codes (MAC).* Tutorials Point. Retrieved from https://www.tutorialspoint.com/cryptography/message_authentication.htm

UN Secretariat. (2016). Relations between Type Approval and post-sale over-the-air software updates for automotive related systems. Document No. ITS/AD-10-13 10th ITS/AD 16 November 2016.

UNECEWP29. (2018). Draft Recommendation on Software Updates of the Task Force on Cyber Security and Over-the-air issues of UNECE WP.29 GRVA. Informal Document GRVA-01-18 1st GRVA session, 25-28 September 2018 Agenda item 6(b).

Vehicle Certification Agency (VCA). (2019). *Type Approval for Cars.* UK GOV. Retrieved from https://www.vehicle-certification-agency.gov.uk/vehicletype/type-approval-for-ca.asp

Vector. (2015). AUTOSAR Security Modules – current status VECTOR 2015-05-27 v1.00.

Verma, A. (2018). *Securing Automotive Software Over the air updates*. Excelfore.com. Retrieved from https://excelfore.com/blog/securing-automotive-software-air-updates/

Vesenev, A., Stahl, F., Hamazayan, H., Ma, Z., Shan, L., Kemmerich, J., & Loiseaux, C. (2019). Practical security and privacy threat analysis in the automotive domain: Long term support scenario for over-the-air updates.

Warwick Control Technologies (WCT). (n.d.). CAN & J1939 Technical Training for Automotive / Transport Engineers and Technicians.

Wind River. (2011) Wind River Platform for Android.

Wind River. (2017). Implementing Over-the-Air Software Updates for Automotive Applications.

Wind River. (2018). Keeping Pace with the Software-Driven Car.

Wind River. (2019). Wind River Automotive Solutions.

Wind River. (2019). Wind River Edge Sync. Delivering comprehensive software, Firmware, and Data Management technologies.

Yu, J., & Luo, F. (2016). Research on Automotive UDS Diagnostic Protocol Stack Test System. *Journal of Automation and Control Engineering*, *4*(5).

Zhou, Y., Wu, X., & Wang, P. (2019). Secure software updates for intelligent connected vehicles. *Electrical Engineering and Computer Science*, *3*, 109–112.

James Howden has a bachelor's degree in Cybernetics and Computer Science from the University of Reading. James is current completing a master's degree in Professional Practice in Digital Forensics and Security at De Montfort University. James currently works as the Vehicle Security Manager for Thatcham Research, the research arm of the UK motor insurance industry. James is responsible for traditional security and cybersecurity and evaluating the relative risk for insurers for the majority of vehicles entering the UK market. James previously worked in law enforcement in the UK where he specialised in digital forensics, cyber crime, counter terrorism and serious organised crime.

Leandros A. Maglaras (PhD) is a Senior Lecturer in the School of Computer Science and Informatics of De Montfort University conducting research in the Cyber Security Centre. He obtained the B.Sc. (M.Sc. equivalent) in Electrical and Computer Engineering from the Aristotle University of Thessaloniki, Greece in 1998, M.Sc. in Industrial Production and Management from the University of Thessaly in 2004 and M.Sc. and Ph.D. degrees in Electrical & Computer Engineering from University of Thessaly, in 2008 and 2014 respectively. In 2018 he was awarded a second Ph.D. in Intrusion Detection in SCADA systems from University of Huddersfield. He served on the Editorial Board of several international peer-reviewed journals such as IEEE Access and the Wiley "Journal on Security & Communication Networks." He is a Senior Member of the Institute of Electrical & Electronics Engineers (IEEE). He is an author of more than 120 papers in scientific magazines and conferences and is a senior member of IEEE.

Mohamed Amine Ferrag received the bachelor's, master's, and Ph.D. degrees from Badji Mokhtar–Annaba University, Algeria, in 2008, 2010, and 2014, respectively, all in computer science. Since 2014, he has been an Assistant Professor with the Department of Computer Science, Guelma University, Algeria. He has edited the book "Security Solutions and Applied Cryptography in Smart Grid Communications" (IGI Global). His research interests include wireless network security, network coding security, and applied cryptography. He is currently serving in various editorial positions such as editorial board member with computer security journals like the "International Journal of Information Security and Privacy" (IGI Global), the "International Journal of Internet Technology and Secured Transactions" (Inderscience Publishers), and the EAI "Endorsed Transactions on Security and Safety" (EAI). He has served as an Organizing Committee Member (the Track Chair, the Co-Chair, the Publicity Chair, the Proceedings Editor, and the Web Chair) in numerous international conferences.