exeley INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.21307/ijssis-2019-011 Issue 1 | Vol. 12 (2019)

Integrity Assurance of OTA Software Update in Smart Vehicles

Geunhyoung Kim and Im Y. Jung

School of Electronics Engineering, College of IT Engineering, Kyungpook National University, Daegu, South Korea.

E-mails: striker112@knu.ac.kr, iyjung@ee.knu.ac.kr

The paper was edited by Subhas Chandra Mukhopadhyay.

Received for publication November 2, 2019.

Abstract

Smart vehicles tend to choose an over-the-air (OTA) software (SW) update service. In an environment with smart vehicles, software malfunctions may have serious consequences such as accidents involving human lives and property loss. Therefore, we must ensure that the software updates in smart vehicles are completed correctly. In this study, we focused on the assurance of both data integrity and service integrity in smart vehicles to improve the OTA SW update service security. To this end, the security features of integrity in smart vehicles were identified and discussed with an emphasis on its potential impact on future vehicular applications.

Keywords

Integrity assurance, OTA update service, Security, Smart vehicle, SW malfunction, Hacking.

The modern vehicle has become a complex system that can communicate with its environment and act autonomously with the support of its sensors and embedded computers (Mahmud et al., 2005; Le et al., 2018). The modern vehicle is called a smart vehicle and is distinguished from the legacy mechanical vehicle. Owing to cutting-edge technology, a smart vehicle can be defined as one with the functionalities of both an autonomous vehicle and a connected vehicle (Halder et al., 2019). Khurram et al. (2016) predicted that three-fourths of the modern vehicles will be connected to a network and the Internet by 2020. This change will result in vehicles with advanced functions, high performance, high efficiency, and increased user convenience (Le et al., 2018). The advanced functions of smart vehicles and the corresponding services for users are deeply related to the software (SW) developed in association with the hardware (HW) of these vehicles. The SW portion in smart vehicles has increased. In environments with smart vehicles, software malfunctions may have serious consequences, such as accidents involving human lives and property loss. In the case of a legacy closed-system vehicle, a malfunction is due to HW faults or bugs in the SW installed (Nilsson et al., 2008). The traditional solution was to recall with a physical collection the vehicles with the problem and fix these vehicles. However, the fixing of all of these vehicles was very time consuming. During the time until fixation, the persons inside the vehicles and/or outside would be in danger and property loss might occur.

In contrast, with an increase in the SW portion of smart vehicles, the SW update cost has increased.

However, because a smart vehicle has the functionality of a connected vehicle, its SW can be easily updated via a network connection without a physical recall process (Riggs et al., 2019). In particular, an over-the-air (OTA) SW update service via a wireless network without the constraints of the vehicles' physical location is being expanded because of its convenience and economic feasibility. An OTA SW update service denotes an SW update service provided by the SW update provider directly via a network connection (Jo et al., 2017). In the future, smart vehicles whose SW portion would be larger than that at present will require active OTA SW updates. The advantages of an OTA SW update are as follows (Nilsson et al., 2008; Idrees et al., 2011; Khurram et al., 2016;

Jo et al., 2017; Halder et al., 2019; Riggs et al., 2019): First, a real-time SW update can mitigate the cost problems caused by the vehicle guarantee. Second, a SW update related to security vulnerability can be done as soon as possible against a zero-day attack. Third, the costs of physical recall can be eliminated, and the time of the guests or users can be saved. Fourth, the SW update can be done continuously and frequently. In contrast, the disadvantages of an OTA SW update are as follows: if the SW update is done by the update SW, the update SW can be another vulnerable SW to be cared for. When an OTA SW update is done via a data network, it will be exposed to network security threats such as tampering and spoofing. In addition, it will get affected by the security vulnerability of the application domain such as that of a smart vehicle. In the studies that addressed the OTA SW update vulnerability over a wireless network (Mahmud et al., 2005; Nilsson and Larson, 2008; Khodari et al., 2019), integrity, confidentiality, authentication, and key management were stressed. Modern vehicles have flash memory in their ECU and are vulnerable to malicious code (Ford). In order to maximize the advantages of an OTA SW update, the right SW to be updated needs to be selected and downloaded into the smart vehicle while ensuring that the vehicle is not attacked during the download process. Then, it should be installed and should replace the old version of the SW, as defined, without any interruption by attackers. That is, the integrity of both the SW and the service should be ensured. Therefore, in this study, we focused on the assurance of both the data integrity and the service integrity in smart vehicles to protect the OTA software update service. To this end, the security features of the integrity of smart vehicles were identified, and the related research was classified and discussed with an emphasis on its limitations and potential impact on future vehicular applications.

The rest of this paper is organized as follows: in the second section, the OTA SW update service is analyzed in terms of its applications, its service architecture, and its security vulnerability. In the third section, the security requirements for the data and service integrity, and the related methods are analyzed. In the fourth section, we conclude this paper and briefly discuss the possible future work directions.

OTA SW update service analysis

OTA SW update service application

Smart vehicles can be at a considerable disadvantage because of an OTA SW update. When they recognize a danger due to vehicle malfunction during driving, they can control their driving and take a corrective action promptly by connecting to the service center because they have the functionality of autonomous driving and a network connection. If the malfunction is determined to be caused by an SW vulnerability, an SW update can be requested and done as soon as possible without any user interruption. Therefore, many manufacturers are expected to adopt an OTA SW update in their smart vehicles.

Ford supports SYNC™ 3 (Tesla), which is applied to Ford's vehicles for the users' convenience. The system has the functionality of an automatic OTA SW update when the vehicles are connected to a data network. Tesla also supports periodic SW updates to its vehicles of Model S, Model 3, Model X, and so on, when the vehicles are connected via a cellular or Wi-Fi network (Ford).

OTA SW update service structure

Figure 1 shows the cloud server, client vehicle, and the network (Cebe et al., 2018). The cloud server OEM is composed of a third-party manufacturer that develops the update SW, and OEM, which distributes the update SW. The network is based on the interfaces of LTE, 3G cellular, and Wi-Fi. In the near future, 5G may be added. The client vehicle gets the OTA SW update service from the cloud server OEM (Doddapaneni et al., 2017).

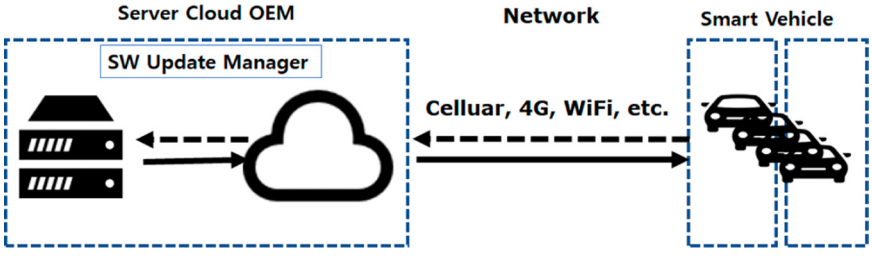

Figure 1: OTA SW update service structure.

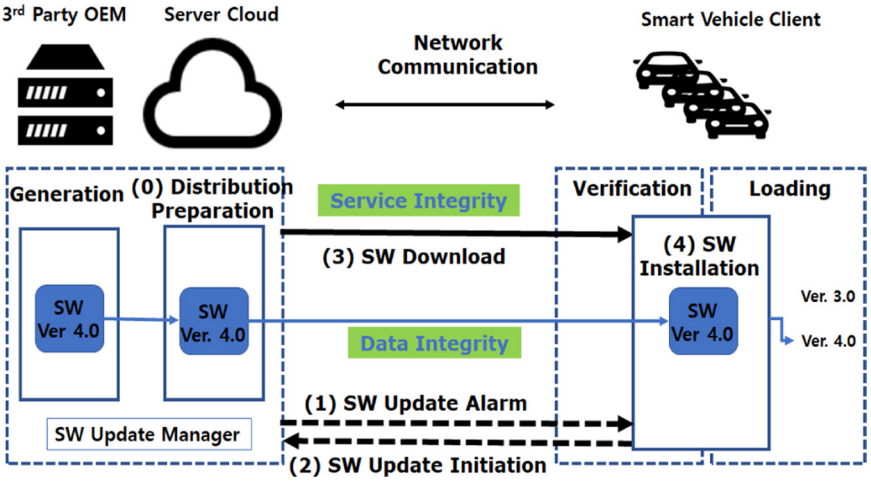

Figure 2: Data and service integrity.

The following steps are applied in detail as shown in Figure 2 (Odat and Ganesan, 2014):

1. The third-party manufacturer conveys an update SW to OEM. OEM uploads it to the cloud server that the client vehicle has access to.
2. The SW update manager notifies the vehicles or the users of the vehicles of the new update SW.
3. The vehicle requests the SW update and receives a path to reach the new update SW in the cloud server.
4. The client vehicle accesses the update SW image via the path in the server cloud. When the client vehicle finishes the SW image download, it verifies and stores the SW image for installation. The client transfers the download result of success or failure.
5. When the download fails, the SW download process is retried from Step 1. When the download succeeds, the vehicle starts to install the new SW and changes the SW version.

There may be no Steps 1 and 2. Instead, the SW update manager may push the update SW to the smart vehicles directly.

Threats for OTA SW update service integrity

The OTA SW update service is rapid and economical (Kong et al., 2016). However, because of the security vulnerability of the wireless network, the attacker can intervene and control the update process (Nilsson et al., 2008; Checkoway et al., 2011). Therefore, it is important to analyze the threats (Avizienis et al., 2004) to design more secure OTA SW update services for smart vehicles.

In order to ensure the integrity of the OTA SW update service, both data integrity and service integrity should be ensured. Data integrity denotes that no change or creation or deletion of data by an unauthorized party was done (Fongen, 2012; Mathur et al., 2015). Service integrity implies that the service process, i.e., SW access, download, and installation, is unadulterated by malware or other hostile modifications (Kuppusamy et al., 2016). Data integrity can be invaded if the data in the cloud are modified by an attacker (Knockel and Crandall, 2012). The contaminated SW may threaten vehicle safety and human lives. It may let the attacker control the vehicle remotely. Service integrity can be invaded by the attacks of man-in-the-middle (Ashibani and Mahmoud, 2017), spoofing, node capture, selective forwarding, and sinkhole (Mayilsamy et al., 2018). From the start of the update SW download to the SW installation, service integrity can be invaded.

Integrity assurance for OTA SW update service

Security requirement for OTA SW update integrity

In order to ensure data integrity, it should be difficult for an attacker to modify the update SW (Halder et al., 2019). Unauthorized access to the update SW, such as modification, creation, or deletion, should be controlled. Any modification to the original update SW should be detected.

For service integrity, the transmission channel of the update SW image, the protocol, and the service

architecture from the distribution point to the client vehicle via the transmission path should be structured securely. The service integrity should be checked at all transmission sections and relay points. All undefined changes should be detected.

Therefore, the security requirements of the OTA SW update integrity can be summarized as follows:

- At the start of the SW update, the SW update should be recognized by the user of the smart vehicle, the smart vehicle, and the SW distributor. The update service process and the SW image transmitted should be transparent and recognizable.
- The update SW image should be transferred from the distribution point to the destination vehicle through secure channels. Moreover, its transmission should be trackable. A secure channel will ensure that there is no data modification or can detect any modification made during the transmission.
- The update SW should be stored securely in the vehicle, and the download process should be secured before installation.
- Some parts of the OTA SW update structure installed in the participants such as smart vehicles, users, or distributors should be managed to check their security vulnerabilities and to be made secure against cyber-attacks.

In order to satisfy the security requirements, the following issues should be addressed:

- service integrity assurance for an OTA SW update between the vehicle, distribution center, and the update management center at the distribution time and at the SW update completion;
- service integrity assurance before and after the transmission of one transmission section from the distribution cloud to the vehicle;
- service integrity assurance at the relay points between the transmission sections;
- data integrity assurance between the original update SW and the distributed one; and
- service and data integrity for the OTA SW update process.

Method classification for OTA SW update integrity

Figure 3 shows the classification of the technologies to ensure data integrity and service integrity. Single methods or complex methods have been proposed for ensuring data integrity and service integrity.

Data integrity

Basically, a hash function is used to check the data integrity. Digital signatures and cryptography using a key to check the data authenticity are used additionally. Parity check, cyclical redundancy check, Hamming code, and block sum check are also used to detect some errors during a network transmission.

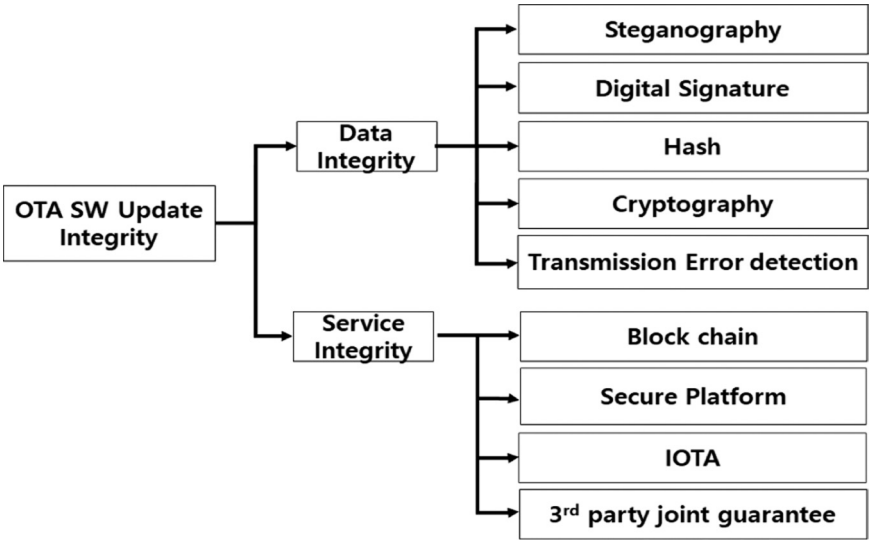

Figure 3: Method classification for OTA software update integrity.

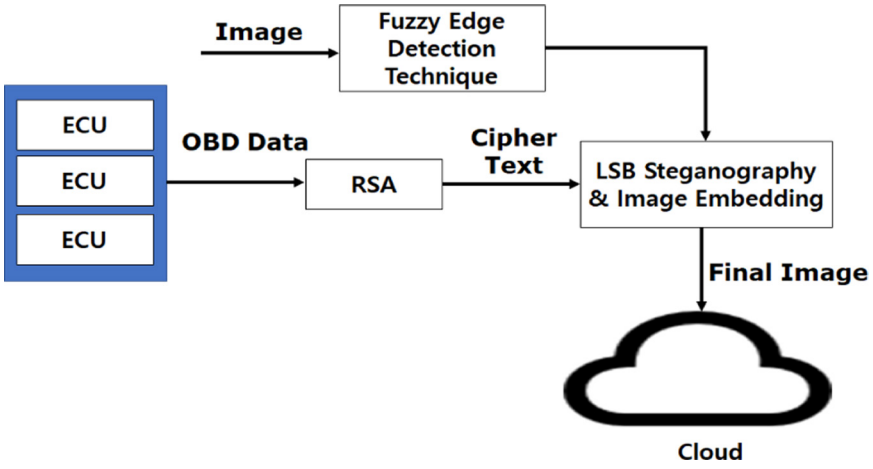

Figure 4: Integrated solution of RSA and steganography.

Nilsson (Mahmud et al., 2005) proposed an integrity check scheme using hash and cipher block-chain (CBC) in a vehicle and an SW update portal. The cypher text encrypted with a key shared by the portal and a vehicle cannot be decrypted without the key. A hash chain of a data fragment where the previous data fragment is used to make the next fragment hash is used to ensure data integrity. He also proposed a secure system for an ECU SW update. A hash function is used for ensuring data integrity (Checkoway et al., 2011). An authentication code from the hash chain and an update firmware binary are transmitted together. The verification of the authentication code ensures the firmware integrity. One drawback of his studies, memory overhead, was addressed in the study (Kuppusamy et al., 2016).

Mayilsamy (Abbas and Sung-Bong, 2019) proposed an integrated method of steganography and cryptography to verify the data integrity, as shown in Figure 4. He used cryptography to secure the data in the cloud. Figure 4 shows the process to store the update SW image in the cloud.

The methods using cryptography with keys have the traditional issue of key sharing and management. As the number of vehicles increases, the key management becomes complex. Because the security stability of cryptography depends on the key, the key should be handled carefully to ensure that it is not exposed. The secure key update and the secure sharing of key are other issues. Data integrity check refers to end-to-end integrity checking. That is, whether the correct data or SW are downloaded are checked without considering the update mid-process. The checking methods in detail include a hash comparison of the data or SW image, the distributor's signature check, and a joint signature together with the third parties. However, a genuine signature and a secure signature update are often assumed.

Service integrity

Blockchain ensures the integrity by the consensus of the distributed nodes that are connected to each other (Gao et al., 2018; Dhakal et al., 2019). As shown in Figure 5, the blockchain has the structure composed of a data layer and a network layer.

The data layer checks the data authenticity by a digital signature with a public key and checks the data integrity with a hash function and a hash chain of data blocks. Because the chain in a blockchain can be created and managed by the consensus of the blockchain nodes without no central control node, it is difficult to modify the blockchain without a majority vote (Gao et al., 2018).

Dhakal et al. (2019) proposed an IoT device firmware update model by using blockchain. The update model is composed of Firmware Manufacturer,

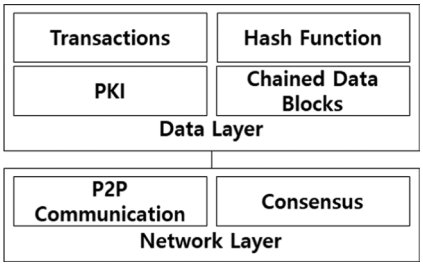

Figure 5: Blockchain structure.

Update Server, Update Manager, and IoT Device. When the update SW is prepared at manufacturer, the update server manages the metadata for the update SW by using the blockchain and starts the device update.

Fraga-Lamas (Abbas and Sung-Bong, 2019) stated that OEM, vehicles, service center, and their related nodes can be placed on a blockchain. They can exchange data with each other and communicate via the blockchain.

Steger (Fraga-Lamas and Fernández-Caramés, 2019) proposed a new SW update framework called SecUp based on IEEE 802.11s. SecUp utilizes the secure framework wpa_supplicant. The authentication of the wireless vehicle interface (WVI) uses both a symmetric key algorithm and a non-symmetric key algorithm. The NFC smart card and the PIN code are used as the first authentication of the system. Furthermore, with the WVI network key shared, SecUp is connected to the IEEE 802.11s network. Data integrity is ensured by the hash value verification.

Stević (Steger et al., 2018) proposed another update framework, ARA::UCM. It filters abnormal data by a package validation test and version tracking.

In contrast, the update framework (TUF) provides secure SW storage (Stević et al., 2018). Signed metadata are used to verify the hash of the SW downloaded. Uptane with TUF core was designed by Kuppusamy for vehicles (Knockel and Crandall, 2012; Kuppusamy et al., 2016). It protects the SW using data recovery by using additional storage area, metadata broadcasting, vehicle version manifest, and time server (Knockel and Crandall, 2012). The most recent metadata are shared by broadcasting from Primary ECU to Secondary ECU in the vehicle. Manifest signatures are collected using a symmetric key provided by OEM. The time server synchronizes time with all the ECUs in the vehicle. The synchronized time can indicate the outdated metadata.

HSM, which has a secure HW module, is used as a secure platform for the SW update. It uses a symmetric key for integrity.

Aust (Kuppusamy et al., 2018) used a wake-up receiver (WUR) to elevate the access level to the network device or module in the vehicle. WUR responds only when the vehicle stays within a predefined area, called the trusted zone. It blocks all unauthorized access. An SW update can be done only in the trusted zone.

Petri et al. (2016) (Aust, 2018) used TPM. At Boot ROM, a trust chain is created, and a secure boot is executed. During the secure boot process, only secure OS can access the security-related module. The other applications are executed separately from TMP.

The blockchain system uses the consensus algorithm between the distributed entities by a majority vote. That is, the larger the number of voters is, the more secure is the consensus process.

Moreover, to leverage the security of the consensus mechanism, public blockchains, such as Bitcoin, usually set a restriction on the block size and the time interval of the transactions, resulting in low transaction throughput. Scalability is thus a difficult issue and must be considered in designing blockchain applications (Dhakal et al., 2019). A vehicle is a fast-moving object. How to let the vehicle participate in the blockchain for integrity assurance and how to solve the delayed consensus should be addressed. A smart vehicle needs to make quick decisions for autonomous and safe driving. The communication environment of a smart vehicle in the real world should be considered. The connection status with fast-moving objects is not stable because of the frequent handover processes between different network cells. Moreover, the communication QoS of each network cell is different.

Thus far, secure storage and secure processing platforms have been proposed against partial bundle installation attacks, rollback attacks, endless data attacks, mixed-bundles attacks, mix-and-match attacks, and arbitrary software attacks (Kuppusamy et al., 2016). These studies focused on the inner vehicle security for a secure SW update. Secure methods using access control resolve the authentication issue, by the management issue of the key and the nonce remains unsolved. In addition, the SW structure of the SW update scheme should be handled from the viewpoint of its security vulnerability and its secure update issue.

Conclusion

In this study, we focused on the integrity of a secure OTA SW update service in smart vehicles. The security features of integrity in smart vehicles are identified from the viewpoints of data integrity and service integrity. Furthermore, the related studies were classified and discussed with an emphasis on their limitations. Because the OTA SW update service had security issues originating from smart vehicles as well as those of the services over a wireless network, more sophisticated and realistic solutions should be provided. In addition, the proposed schemes themselves should be verified in terms of their stability with respect to security because any trustee or entity cannot be assumed to be honest or to remain unattacked forever in a real application domain such as a smart vehicle.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Korea (2017R1D1A1B03034950).

Literature Cited

Abbas, Q. E. and Sung-Bong, J. 2019. A survey of blockchain and its applications. International Conference on Artificial Intelligence in Information and Communication, pp. 001–003.

Ashibani, Y. and Mahmoud, Q. H. 2017. Cyber physical systems security: analysis, challenges and solutions. *Computers and Security* 68: 81–97.

Aust, S. 2018. Software downloads in trusted zones with wake-up sensors for connected vehicles. IEEE Vehicular Technology Conference, pp. 1–5.

Avizienis, A., Laprie, J.-., Randell, B. and Landwehr, C. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing* 1(1): 11–33.

Cebe, M., Erdin, E., Akkaya, K., Aksu, H. and Uluagac, S. 2018. Block4Forensic: an integrated lightweight blockchain framework for forensics applications of connected vehicles. *IEEE Communications Magazine* 56(10): 50–57.

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F. and Kohno, T. 2011. Comprehensive experimental analyses of automotive attack surfaces. In Proceedings of USENIX Security Symposium, pp. 77–92.

Dhakal, S., Jaafar, F. and Zavarsky, P. 2019. Private blockchain network for IoT device firmware integrity verification and update. IEEE International Symposium on High Assurance Systems Engineering, pp. 164–170.

Doddapaneni, K., Lakkundi, R., Rao, S., Kulkarni, S. G. and Bhat, B. 2017. Secure FoTA object for IoT. IEEE Conference on Local Computer Networks Workshops, pp. 154–159.

Fongen, A. 2012. Identity management and integrity protection in the internet of things. International Conference on Emerging Security Technologies, Lisbon, Portugal, September 5-7.

Ford. Available at: www.ford.com.ph/engineering/sync/

Fraga-Lamas, P. and Fernández-Caramés, T. M. 2019. A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE Access* 7: 17578–17598.

Gao, W., Hatcher, W. G. and Yu, W. 2018. A survey of blockchain: techniques, applications, and challenges. International Conference on Computer Communication and Networks, pp. 1–11.

Halder, S., Ghosal, A. and Conti, M. 2019. Secure OTA software updates in connected vehicles: a survey. ArXiv, available at: https://arxiv.org/abs/1904.00685

Idrees, M. S., Schweppe, H., Roudier, Y., Wolf, M., Scheuermann, D. and Henniger, O. 2011. Secure automotive on-board protocols: a case of over-the-air firmware updates. *Lecture Notes in Computer Science* 6596: 224–238.

Jo, H. J., Choi, W., Na, S. Y., Woo, S. and Lee, D. H. 2017. Vulnerabilities of android OS-based telematics system. *Wireless Personal Communications* 92(4): 1511–1530.

Khodari, M., Rawat, A., Asplund, M. and Gurtov, A. 2019. Decentralized firmware attestation for in-vehicle networks. In Proceedings of the 5th on Cyber-Physical System Security Workshop–CPSS '19, Auckland, New Zealand, July 8.

Khurram, M., Kumar, H., Chandak, A., Sarwade, V., Arora, N. and Quach, T. 2016. Enhancing connected car adoption: security and over the air update framework. IEEE World Forum on Internet of Things, Reston, VA, December 12-14, pp. 194–198, available at: https://ieeexplore.ieee.org/document/7845430

Knockel, J. and Crandall, J. R. 2012. Protecting free and open communications on the internet against man-in-the-middle attacks on third-party software: we're FOCI'd. USENIX Workshop on Free and Open Communications on the Internet, Bellevue, WA, August 6.

Kong, H., Kim, T. and Hong, M. 2016. A security risk assessment framework for smart car. International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 102–108.

Kuppusamy, T. K., DeLong, L. A. and Cappos, J. 2018. Uptane: security and customizability of software updates for vehicles. *IEEE Vehicular Technology Magazine* 13(1): 66–73.

Kuppusamy, T. K., Torres-Arias, S., Diaz, V. and Cappos, J. 2016. Diplomat: using delegations to protect community repositories. USENIX Symposium on Networked Systems Design and Implementation, pp. 567–581.

Kuppusamy, T. K., Brown, A., Awwad, S., McCoy, D., Bielawski, R., Mott, C., Lauzon, A., Weimerskirch, S. and Cappos, J. 2016. Uptane: securing software updates for automobiles. In Proceedings of International Conference on Embedded Security in Car Europe, pp. 1–11.

Le, V. H., Hartog, J. D. and Zannone, N. 2018. Security and privacy for innovative automotive applications a survey. *Computer Communications* 132: 17–41.

Mahmud, S. M., Shanker, S. and Hossain, I. 2005. Secure software upload in an intelligent vehicle via wireless communication links. In Proceedings of IEEE Intelligent Vehicles Symposium, pp. 588–593.

Mathur, R., Agarwal, S. and Sharma, V. 2015. Solving security issues in mobile computing using cryptography techniques–a survey. International Conference on Computing, Communication and Automation, pp. 492–497.

Mayilsamy, K., Ramachandran, N. and Raj, V. S. 2018. An integrated approach for data security in vehicle diagnostics over internet protocol and software update over the air. *Computers & Electrical Engineering* 71: 578–593.

Nilsson, D. K. and Larson, U. E. 2008. Secure firmware updates over the air in intelligent vehicles. IEEE International Conference on Communications Workshops, pp. 380–384.

Nilsson, D. K., Phung, P. H. and Larson, U. E. 2008. Vehicle ECU classification based on safety-security characteristics. IET Road Transport Information and Control and ITS United Kingdom Members' Conference, pp. 1–7.

Nilsson, D. K., Sun, L. and Nakajima, T. 2008. A framework for self-verification of firmware updates over the air in vehicle ECUs. IEEE GLOBECOM Workshops, pp. 1–5.

Odat, H. A. and Ganesan, S. 2014. Firmware over the air for automotive, fotamotive. In Proceedings of IEEE International Conference on Electro/Information Technology, pp. 130–139.

Petri, R., Springer, M., Zelle, D., McDonald, I., Fuchs, A. and Kraub, C. 2016. Evaluation of lightweight TPMS for automotive software updates over the air. In Proceedings of International Conference on Embedded Security in Car, pp. 1–15.

Riggs, C., Rigaud, C., Beard, R., Douglas, T. and Elish, K. 2019. A survey on connected vehicles vulnerabilities and countermeasures. *Journal of Traffic and Logistics Engineering* 6(1): 11–16.

Steger, M., Boano, C. A., Niedermayr, T., Karner, M., Jillebrand, J., Roemer, K. and Rom, W. 2018. An efficient and secure automotive wireless software update framework. *IEEE Transactions on Industrial Informatics* 14(5): 2181–2193.

Stević, S., Lazić, V., Bjelica, M. Z. and Lukić, N. 2018. IoT-based software update proposal for next generation automotive middleware stacks. IEEE International Conference on Consumer Electronics, pp. 1–4.

Tesla. Software updates, available at: www.tesla.com/support/software-updates. (accessed December 6, 2019).