Assessing the Cost of Quantum Security for Automotive Over-The-Air Updates

Michele La Manna *Department of Information Engineering (DINFO)*
University of Florence
Florence, Italy
michele.lamanna@unifi.it
Pericle Perazzo *Department of Information Engineering (DII)*
University of Pisa
Pisa, Italy
Luigi Treccozzi *Department of Information Engineering (DII)*
University of Pisa
Pisa, Italy
Gianluca Dini *Department of Information Engineering (DII)*
University of Pisa
Pisa, Italy

***Abstract*—Over-The-Air (OTA) update is an innovative paradigm that is rapidly spreading through the automotive industry. Software updates can be capillary distributed thanks to the many Vehicle-to-Everything (V2X) communication infrastructures that are part of a Smart City. Unfortunately, the majority of the existing OTA frameworks and schemes are not quantum resistant, meaning that when quantum computing will become reality, they will not be secure anymore. The U.S. National Institute of Standards and Technology (NIST) has announced a contest to determine the post-quantum standards for digital signatures schemes. In this paper, we evaluate the performance of the digital signature verification algorithms of two out of the three finalists for the NIST contest, namely FALCON and CRYSTALS-DILITHIUM. These algorithms are tested on automotive-oriented evaluation board, namely the Xilinx Zynq Ultrascale+ ZCU102. The results show that FALCON is a more promising algorithm compared to DILITHIUM both regarding signature verification execution time and signature size.**

***Index Terms*—Post-Quantum Cryptography, Smart City, Automotive, Over-The-Air Update.**

I. Introduction

The automotive industry is offering vehicles with increasing features, such as autonomous driving, assisted driving, infotainment and more. Those features are meant to help the driver or to make the time spent in the vehicle more enjoyable. To enable such features, automotive companies equip vehicles with a growing number of *Electronic Control Units* (ECUs). The ECUs installed in a vehicle can be seen as a network of embedded systems dedicated each to a particular task. The resources (i.e., memory, connectivity, computing power, etc…) differ for each ECU, and they depend on the assigned task. Simple tasks require less resources, while complex tasks require more resources and also more lines of code to run. Clearly, each ECU needs maintenance, not only for the hardware that may become faulty with time and usage, but also for the software that may need some modifications. Modern vehicles have on board over 100 ECUs [1], so the software maintenance aspect has become an issue [2]. Many different frameworks and solutions have been proposed to overcome the challenges of safely distributing a software update "Over the Air" (OTA) [3, 4, 5]. Many of those solutions benefit from the Vehicle-to-Everything (V2X) communication infrastructures embedded in a Smart City [6], to rapidly disseminate the update to the deployed vehicles. However, such approaches adopt as a threat model an adversary that has not quantum computation capabilities. This is understandable, since there is not yet a *Post-Quantum* (PQ) standard algorithm for digital signature or encryption. At the time of writing, the U.S. *National Institute of Standards and Technology* (NIST) is conducting the third and final phase of the contest for the next PQ cryptography standards.

In this work, we evaluate two of the three finalists for the digital signature contest, namely FALCON [7] and CRYSTALS-DILITHIUM [8] (from now on DILITHIUM, for short). Those two digital signature algorithms have in common one very interesting aspect, which is that they both are lattice-based. Lattices are mathematical structures upon which hard problems can be formulated, which are considered to be intractable even for quantum computers [9]. Hence, safe quantum-resistant cryptographic schemes can be built using lattice-based mathematics. We did not evaluate the third final candidate, Rainbow [10], which is based on a different hard problem involving multivariate quadratic systems. According to many studies, the lattice-based algorithms are more efficient than the multivariate-based ones [11, 12, 13, 14], and therefore they are more likely to win the contest. We conduct some

This work was supported by the Italian Ministry of Education, University and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence), by the European Union's Horizon 2020 research and innovation programme "European Processor Initiative" under grant agreement No. 826647, and by the project PRA_2020_92 "Quantum computing, technologies and applications" funded by the University of Pisa.

experiments over a Xilinx UltraScale+ ZCU102 evaluation board, since its performance is representative of that of complex ECUs that are mounted on a modern vehicle. Our objective is to determine the impact of PQ digital signature algorithms that most likely will be selected the standard in about a year[1].

This paper is structured as follows: in Section II we describe related works; in Section III we describe the system and the adversary model; in Section IV we show the methodology used to perform the experiments and the corresponding results; finally in Section V we draw conclusive remarks, and illustrate future research directions.

II. Background and Related Work

It is important to consider the OTA update solution as the major future technique to maintain the ECUs inside a vehicle. Indeed, the adoption of the OTA update is an advantage for both the owner of the vehicle and the company that produced it. The owner is not required to bring the car to the nearest licensed workshop, while the company saves up to half the overall maintenance cost [15].

In 2016 Karthik et al. [16] released Uptane, a Framework for software update over the air, created for securing ground vehicles. Uptane *requires* one to sign the images of the update to transmit to the vehicles, however it refers to RSA and ECDSA. Being a framework, digital signature algorithm can be seamlessly changed, therefore it is possible to configure Uptane to be quantum resistant.

In 2018 Asokan et al. [17] proposed ASSURED, a framework for OTA software update, based on Uptane [16]. In their work, they claim that assured reaches 5 objectives:

- End-to-End authentication and integrity: the update must be signed by the manufacturer and verified by the device.
- Update Authorization from Controller: only authorized devices can install the update.
- Attestation of update installation: the device must provide proof of the update installation.
- Protection of Code and secret key *on device*: the update must be stored and then installed in secure storage and isolated execution of critical code.
- Minimal burden for the device.

However, ASSURED does not consider an adversary with quantum computing capabilities, and therefore the authors runs their experiments with an EdDSA variant, the ED25519 (which is not quantum resistant).

In 2020 Ravi et al. [18] proposed a novel authentication protocol called LASAN_M for secure automotive systems based on secure post-quantum cryptography. In their work, the authors compared the performance of pre-quantum schemes (i.e., ECDSA, ECDH) to the performance of the post-quantum ones (i.e., Kiber, DILITHIUM). The authors tested such algorithms on an automotive compliant board, as we do. In their work, the authors test the performances over small messages, since the use-case scenario is the V2V communication that has real-time constraint. Instead, in our work, we assess the impact of PQ digital signature verification algorithms over larger amount of data (up to almost 6 MiB). This is because in the case of a SW update there is not a real time constraint, in fact modern SW updates are several MB in size.[19]

In 2020 Wang et al. [11] proposed an implementation of an *Hardware Security Module* (HSM) to be mounted on automotive vehicles. In their work, the synthesized an hardware accelerator for many cryptographic primitives, among which there are the primitives for DILITHIUM. However, they did not test the performance by varying the size of the message that has to been verified, since they focus on the vehicle architecture and therefore short, ECU-to-ECU communications. Instead, in our work, we assess the impact of PQ digital signature verification algorithms over larger amount of data, as in the case of a SW update, which needs a larger amount of data to be verified. Moreover, in our work, we will also evaluate the performance of FALCON, another lattice-based post-quantum digital-signature algorithm that is among the three NIST finalists.

III. System Model

Fig. 1 depicts our system, which is composed of two main entities, the *Original Equipment Manufacturer* (OEM) and the vehicles that need the updates. Inside the vehicle, following the AUTOSAR specification, there is a dedicated ECU that performs the needed cryptographic operations concerning the updates, called the *Update and Configuration Manager* (UCM) [20]. In addition to the OEM and the vehicles, there are also two types of intermediary: a *Cloud Server*, and many *Edge Nodes*. The OEM produces a new software update for a specific ECU, and such an update must be distributed to the vehicles that will need it. The software distribution must be protected with *authentication* and *integrity*. To this end, the OEM possesses a *Signature Key* (SK), which is private, through which the OEM itself signs the updates before distributing them. Moreover, each vehicle stores a copy of the *Verification Key* (VK), which is public, inside the UCM. The UCM verifies the OEM's signature over the received update using the verification key.

A. Use Case Scenario

The OEM leverages third party cloud servers to improve the capillarity of the distribution [21]. Cloud servers are a critical resource since they manage all the connections to download the updates on behalf of the OEM. Moreover, the OEM uses also a distribution framework such as Uptane [16] to guarantee the integrity and the authenticity of the update. A framework is needed since it specifies all the protocols to be followed as well as the metadata that must be included in the OEM's digital signature along with the update itself. The metadata is additional data that describes the update, such as the update's version number and the update's size. The metadata, the update, and the OEM's signature over them are called *"update package"*. Edge nodes, which are typically distributed throughout a smart city, are also used to further improve the timeliness of the installation of the update after

[1]The third round of the NIST competition started at the end of 2020, and the NIST estimated its duration from 12 to 18 months. https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement

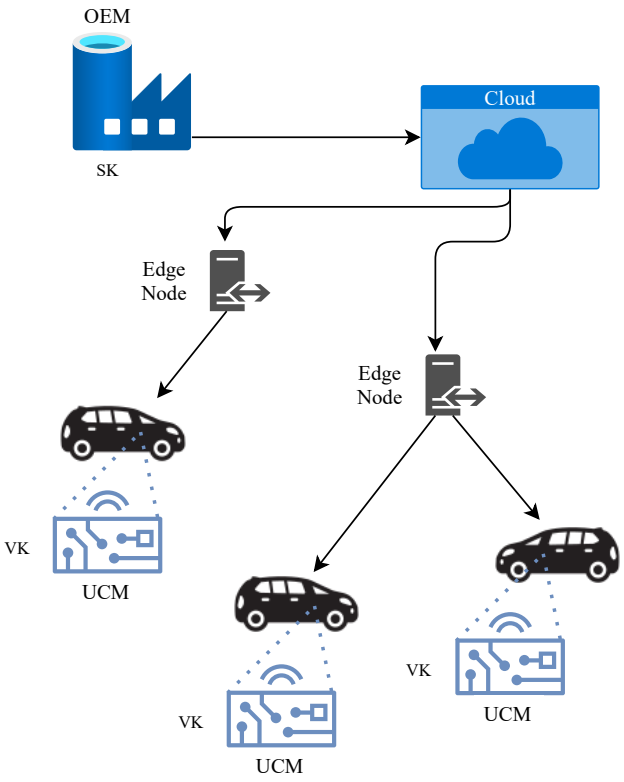

Fig. 1. Our system model. The OEM stores the updates inside a third-party cloud server. The edge nodes download the update and then disseminate it to the deployed vehicles in the smart city.

its release. The UCM is in charge of downloading the update package from the edge nodes, check the metadata, and then it must verify the signature of the OEM on it. If the control of the metadata does not raise any error and the signature verification is successful, the UCM shall forward the update to the ECU that needs it, which proceeds with the installation.

B. Threat Model

In this section, we describe the capabilities of an adversary, and how this application would resist his attacks. We consider an *active* attacker that can observe and modify the communication between the edge nodes and each vehicle. Moreover, the attacker has the capabilities to run *Polynomial Time Quantum* (PTQ) algorithms, such as the Shor's algorithm [22] or the Grover's algorithm [23]. The adversary can try two different attacks, namely the *rollback* attack and the *malicious update* attack. To perform the rollback attack, the adversary must capture a previous version of the vehicle software, which is affected by a known bug or vulnerability. This can be done by simply capturing from the network every update that the OEM releases. When this version of the software becomes outdated, or an attack has been found, the adversary can leverage a malicious edge node and send such signed message to the victim's vehicle. The adversary however, must modify the metadata field regarding the update version of the update package, to make it look like a new update.

In the malicious update attack, instead, the adversary crafts a malicious update that introduces an intended vulnerability in the vehicle. The objective of the adversary is to induce the installation of the malicious update on the victim's vehicle. To do this, the adversary again leverages a malicious edge node and notifies the victim's vehicle that a new update is released.

Both of this attacks are addressed by the digital signature computed upon the OEM's update package. Indeed, when the client receives a new update package, it checks the validity of the signature: in case of a rollback attack, the signature will not be valid since the metadata has been modified; in case of a malicious software attack, the adversary is not able to produce a valid signature on the maliciously crafted update.

IV. Performance Evaluation

The objective of this paper is to assess the impact of the PQ digital signature verification algorithm on a modern vehicle, therefore in our experiment we will simulate the operations performed by the UCM. Please note that the operations performed by the UCM are the only considerably affected by the transition from pre-quantum cryptography to post-quantum cryptography. In fact, also the OEM is affected since it digitally signs the update package. However we consider such a computational effort uninteresting given the full resources of the OEM.

This section is divided in two parts: first, we describe the used hardware and the methodology behind our experiments; then, we show the obtained results.

A. Experiment Details

We developed a client-server application that roughly emulates the behaviour of our system described in Sec. III. In particular, the developed server creates and signs the update packages, acting as the OEM, and it directly transmits the update package to the client, which verifies the signature. We used C as the programming language, and we leveraged the OpenSSL library (for pre-quantum algorithms), the FALCON code submitted to the NIST contest [24], and the DILITHIUM code submitted to the NIST contest [25]. The client runs on a Xilinx ZCU102 evaluation board equipped with a Zynq UltraScale+ MPSoC chip which features a quad Arm Cortex®-A53 cores with Arm Neon™technology. This board is marketed as automotive-compliant since its performance reliably represents an ECU with the UCM role inside a vehicle [5]. The server runs on a laptop featuring an Intel i7-9750H processor with Ubuntu 20.04 as the operating system.

We evaluate the performance of four digital signature verification algorithm: RSA and ECDSA (which are not quantum resistant), plus FALCON and DILITHIUM (which are quantum resistant). We measure the signature verification time for each of those four algorithm varying mainly two parameters: the size of the update to sign, and the required security level. We considered 3 security levels: 128-bit, 192-bit, and 256-bit. Table I shows the verification key size of each scheme with respect to the target security level.

Clearly, RSA and ECDSA are not quantum resistant algorithms, therefore their security levels are to be considered adequate in the classic setting, and not in the quantum one. Indeed, RSA algorithms are based upon the mathematical problem

Schemes	Security level	Verification Key Size
Pre-Quantum Digital Signature Algorithms		
ECDSA	128	32 bytes
	192	48 bytes
	256	64 bytes
RSA	128	384 bytes
	192	960 bytes
	256	1920 bytes
Post-Quantum Digital Signature Algorithms		
FALCON	128	897 bytes
	192	(not available)
	256	1793 bytes
DILITHIUM	128	1312 bytes
	192	1952 bytes
	256	2592 bytes

TABLE I
VERIFICATION KEY SIZE PER ALGORITHM IN RELATION TO THREE DIFFERENT SECURITY LEVELS.

of the factorization of large numbers, and the ECDSA ones are based on the discrete logarithm problem. These problems are considered to have a sub-exponential complexity for an adversary without quantum-computing capabilities, however these same problems are easily solvable by an adversary with quantum-computing capabilities. Therefore, the main difference between Pre-Quantum and Post-Quantum algorithms is that the security level of Pre-Quantum algorithms are reduced almost to zero in the Post-Quantum scenario.

Regarding the update we consider three different sizes, namely 1.1 MiB, 2.7 MiB, and 5.9 MiB. To the authors' knowledge, those sizes are a realistic size for a patch of a simple ECU (such as a sensor controller) or for a medium-complexity ECU (such as a domain-controller) [19].

Combining the selected parameters, we have a total of nine experiments. We briefly show how those nine experiments are denoted: i) a capital letter between "S", "M" or "L", which denotes the size of the update (small, medium, large, respectively); ii) the number of security bits considered, "128", "192", or "256". So, for example, the experiment considering the update size of 2.7 MiB with 192-bit security is denoted as "M-192".

For each experiment, we averaged over 500 independent repetitions, each with a different key pair of the four digital signature algorithms (Signature key and Verification key). The verification keys are loaded on the client (Xilinx ZCU 102). As for the metadata, we use two quantities for each update: the update version, and the update size. The server proceeds to sign, along with the metadata field, the appropriately sized update with each of the generated Signature Key, creating the update packages. Finally, the client receives the update, checks the version, and verifies the signature.

B. Results

Fig. 2 shows the execution time of the digital signature algorithms for the experiments S-128, S-192, and S-256. Fig. 3 shows the execution time of the digital signature algorithms for the experiments M-128, M-192, and M-256. Fig. 4 shows the execution time of the digital signature algorithms for the experiments L-128, L-192, and L-256. In each figure on the X-axis there are the three different security levels, while on the Y-axis there are the amount of milliseconds needed to perform the signature verification. Each point is calculated by computing all the iteration's average with 95% confidence interval, which are all lower than 1 ms and therefore unnoticeable in the figures. We did not run tests for FALCON in the experiment sets S-192, M-192, and L-192 since FALCON's authors did not provide an implementation for such a security level. Evaluated points are highlighted with a circle, a triangle, a square, and a diamond for RSA, ECDSA, DILITHIUM, and FALCON, respectively. The evaluated points are connected through solid lines to notice the time increase between security levels.

By analyzing the results, we notice two interesting trends. First, we see that post-quantum algorithms and pre-quantum algorithms are comparable when the update is small. In particular, in the S-256 set of experiments, the pre-quantum verification algorithms run in about 19 ms, while the post-quantum verification algorithms run in about 22 ms (FALCON) and in 28 ms (DILITHIUM). However, with the increasing size of the update, the verification time of the post-quantum algorithm increases drastically, while the pre-quantum signature verification time does not vary significantly. FALCON and DILITHIUM execution times, respectively, range from 22 ms and 28 ms (in the S-256 set) to 118 ms and 139 ms (in the L-256 set), while both RSA and ECDSA execution times range from 19 ms to 28 ms in the same sets. The motivation for this difference resides on the internal scheme of the digital signature itself. The OpenSSL implementations of the pre-quantum schemes RSA and ECDSA perform SHA-256 over the message to be signed. This clearly limits the differences between different sized updates, since the hash function (i.e., SHA-256) is hardware-accelerated and its computation time is dominated by the rest of the RSA and ECDSA digital signature algorithms. Instead, the post-quantum algorithms implementations submitted to the NIST contest do not have an embedded hardware-accelerated hash function. The message to be signed (and therefore verified) is fed through an *eXtendable-Output Function* (XOF), called *Secure Hash Algorithm and KECCAK* (SHAKE) [26]. Basically, SHAKE is an hash function with a customizable amount of output bits. In their implementation for the NIST contest, both FALCON and DILITHIUM chose to use SHAKE-256, which means that the output of the XOF function is exactly 32 bytes. FALCON's and DILITHIUM's execution time of their NIST implementation increases noticeably with the input size, rather than the security level.

Indeed, the second interesting trend is that the post-quantum algorithm verification seems to be unaffected by the security level, in terms of time needed. As for FALCON, the time taken is constant if the security level varies, while for DILITHIUM there is an almost unnoticeable increase (about 1 ms for each update size). These counter-intuitive results are due to the fact that SHAKE is not hardware-accelerated and it is the dominant cost of the signature verification algorithm for both DILITHIUM and FALCON.

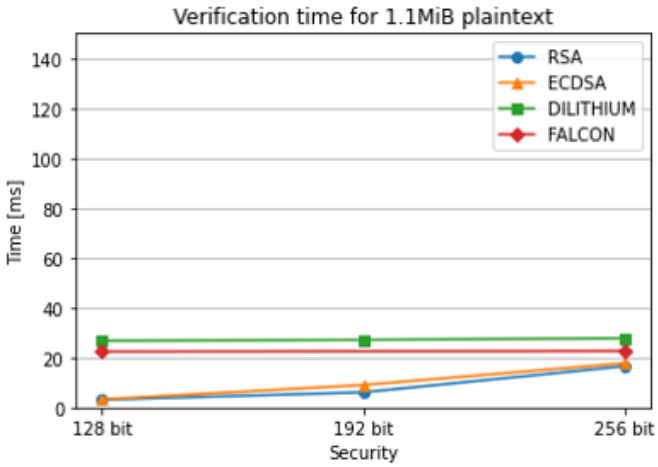

Fig. 2. S-128, S-192, and S-256 set of experiments.

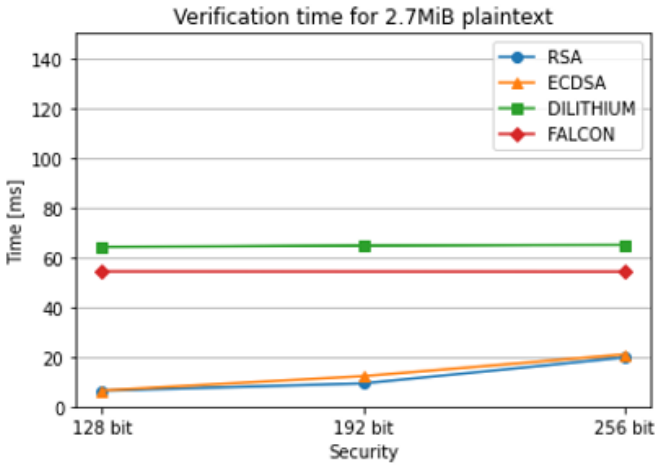

Fig. 3. M-128, M-192, and M-256 set of experiments.

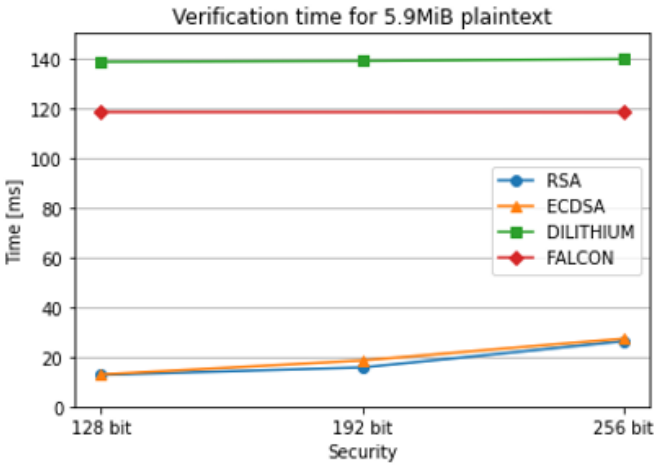

Fig. 4. L-128, L-192, and L-256 set of experiments.

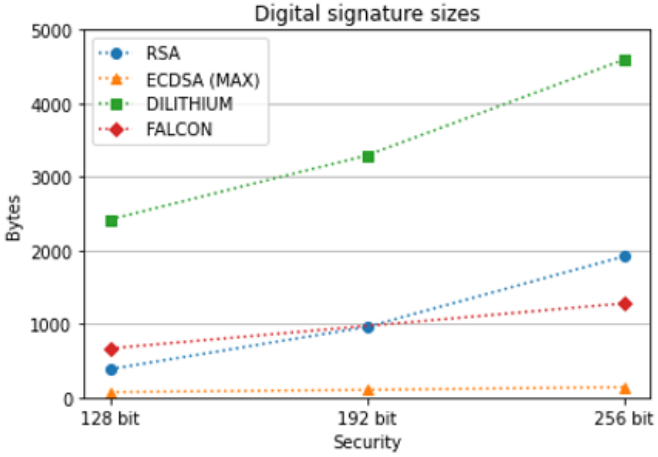

Fig. 5. Signature sizes for each digital signature scheme with varying security level.

Finally, we mapped the signature sizes for the different security levels analyzed, and the results are shown in Fig. 5. FALCON seems to be very efficient in this regard, since it is better than RSA in the 256-bit security scenario, and it is under 1 KB for the 128-bit security scenario. DILITHIUM instead has a greater signature size in every scenario, and it seems to scale worse than FALCON with the increase of the security level.

V. Conclusion and Future Works

In this work we ran some experiments over a Xilinx UltraScale+ ZCU102 evaluation board, which is automotive compliant, and reliably represents the performance of a UCM ECU. We assessed the impact of post-quantum digital signature algorithms that most likely will be the standard in about a year. The results of the experiments showed a slight advantage of FALCON over DILITHIUM regarding the execution time; moreover, the advantage increases noticeably on the digital signature size, as FALCON achieves sizes way smaller than DILITHIUM (from less than a half to almost a quarter, depending on the security level). In the near future we plan to investigate the performances of both FALCON and DILITHIUM with an hardware accelerator for SHAKE, since its software implementation seems to be the bottleneck of the verification algorithm for large input messages like the ones used in OTA update use-case.

References

[1] NXP, *Whitepaper NXP*, 2018, https://www.nxp.com/docs/en/white-paper/AUTOGWDEVWPUS.pdf.

[2] H. A. Odat and S. Ganesan, "Firmware over the air for automotive, fotamotive," in *IEEE International Conference on Electro/Information Technology*, 2014, pp. 130–139.

[3] M. Steger, C. A. Boano, T. Niedermayr, M. Karner, J. Hillebrand, K. Roemer, and W. Rom, "An efficient and secure automotive wireless software update framework," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 5, pp. 2181–2193, 2018.

[4] T. K. Kuppusamy, L. A. DeLong, and J. Cappos, "Uptane: Security and customizability of software updates for

vehicles," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 66–73, 2018.

[5] M. La Manna, L. Treccozzi, P. Perazzo, S. Saponara, and G. Dini, "Performance evaluation of attribute-based encryption in automotive embedded platform for secure software over-the-air update," *Sensors*, vol. 21, no. 2, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/2/515

[6] A. Ghosal and M. Conti, "Security issues and challenges in v2x: A survey," *Computer Networks*, vol. 169, p. 107093, 2020.

[7] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, 2018.

[8] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-dilithium: A lattice-based digital signature scheme," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 1, pp. 238–268, Feb. 2018. [Online]. Available: https://tches.iacr.org/index.php/TCHES/article/view/839

[9] D. J. Bernstein, J. Buchmann, and E. Dahmen, Eds., *Post-Quantum Cryptography*. Springer Berlin Heidelberg, 2009.

[10] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Applied Cryptography and Network Security*, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 164–175.

[11] W. Wang and M. Stöttinger, "Post-quantum secure architectures for automotive hardware secure modules." *IACR Cryptol. ePrint Arch.*, vol. 2020, p. 26, 2020.

[12] Z. Liu, K.-K. R. Choo, and J. Grossschadl, "Securing edge devices in the post-quantum internet of things using lattice-based cryptography," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 158–162, 2018.

[13] DarkMatter, "Introduction to post-quantum cryptography," 2019.

[14] W. B. et. al, "Post-quantum cryptography: Current state and quantum mitigation," 2021.

[15] Aptiv, *What Is Over-the-Air (OTA)?*, 2020, www.aptiv.com/newsroom/article/what-is-over-the-air-(ota).

[16] T. Karthik, A. Brown, S. Awwad, D. McCoy, R. Bielawski, C. Mott, S. Lauzon, A. Weimerskirch, and J. Cappos, "Uptane: Securing software updates for automobiles," in *International Conference on Embedded Security in Car*, 2016, pp. 1–11.

[17] N. Asokan, T. Nyman, N. Rattanavipanon, A.-R. Sadeghi, and G. Tsudik, "Assured: Architecture for secure software update of realistic embedded devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2290–2300, 2018.

[18] P. Ravi, V. K. Sundar, A. Chattopadhyay, S. Bhasin, and A. Easwaran, "Authentication protocol for secure automotive systems: Benchmarking post-quantum cryptography," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5.

[19] Tesla, *Tesla Update Timeline*, 2020, https://teslascope.com/teslapedia/software/timeline.

[20] AutosarAdaptive, *Specification of Update and Configuration Management*, 2019.

[21] M. La Manna, P. Perazzo, and G. Dini, "Sea-brew: A scalable attribute-based encryption revocable scheme for low-bitrate iot wireless networks," *Journal of Information Security and Applications*, vol. 58, p. 102692, 2021.

[22] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999. [Online]. Available: https://doi.org/10.1137/S0036144598347011

[23] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," *Phys. Rev. Lett.*, vol. 79, pp. 325–328, Jul 1997. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.325

[24] *FALCON submission for the third and final round of the NIST contest*, 2020. [Online]. Available: csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Falcon-Round3.zip

[25] *DILITHIUM submission for the third and final round of the NIST contest*, 2020. [Online]. Available: csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Dilithium-Round3.zip

[26] M. J. Dworkin, "Sha-3 standard: Permutation-based hash and extendable-output functions," 2015.