Post-Quantum Cryptography for ECU Security Use Cases

Fabio Campos[1], Michael Meyer[1,2], Steffen Sanwald[3], Marc Stöttinger[3], and Yi Wang[4]

[1] Department of Computer Science, University of Applied Sciences Wiesbaden, Unter den Eichen 5, 65195 Wiesbaden, Germany
{fabiofelipe.campos|michael.meyer}@hs-rm.de
[2] Department of Mathematics, University of Würzburg, Emil-Fischer-Straße 40, 97074 Würzburg, Germany
[3] Security & Privacy Competence Center, Continental AG, Wilhelm-Fay-Strasse 30, 60488 Frankfurt am Main, Germany
{steffen.sanwald|marc.stoettinger}@continental-corporation.com
[4] Security & Privacy Competence Center, Continental AG, Continental Building, 80 Boon Keng Road, Singapore 339780, Singapore
estelle.wang@continental-corporation.com

Abstract. Shor's algorithm imposes a threat to many classical and popular cryptographic algorithms because it can leverage the computing power of a quantum computer to efficiently factorize big numbers and solve the discrete logarithm problem. Due to the Mosca theorem the chance is 50% that quantum computers with enough computation power exist by 2031 and can break all standard asymmetric cryptographic signature and encryption schemes. As a consequence, attackers would be able to spoof the cryptographic signature of over-the-air updates for electronic control units and inject manipulated firmware/software into the vehicle to execute arbitrary code. Post-quantum cryptographic (PQC) schemes are resilient against attacks exploiting Shor's and other quantum algorithms. In this paper we analyze the constraints of promising PQC candidates (regarding the threat of quantum computers) for security use cases in the automotive domain. Practical results show that they can not only offer advantages regarding security against the future threat of quantum computers, but also outperform existing cryptographic algorithms.

Keywords: Automotive Domain · Electronic Control Unit · Post-Quantum Cryptography · Secure Software Update.

1 Introduction

Connected modern vehicles already contain up to 100 electronic control units (ECUs), forming in-vehicle networks and communicating to infrastructure. Although this enables instant access to information worldwide and increases the comfort in our daily life, each additional interface adds a new attack vector from

a cyber security perspective. Security countermeasures mainly leverage cryptography to reach its goals of integrity, authenticity and confidentiality. Those cryptographic algorithms rely on mathematical problems, which are not feasible to solve with the computational power of modern systems in reasonable time frame [11, 12]. However, the involvement and continuously improvement of quantum computer may break these security assumptions.

Classical computer systems use the binary states 0 and 1 as the smallest information. In contrast, quantum computers use so called 'qubits', which can describe the superposition of states by using the laws of quantum mechanics. This enables quantum computers to encode exponentially more information than traditional computer systems [25]. Shor's algorithm proves that the quantum architecture empowers computers to solve mathematical problems like integer factorization of the Rivest–Shamir–Adleman (RSA) algorithm or computing discrete logarithms of Diffie-Hellman in attainable polynomial time, cf. [12, 41]. Even the latest trend to use Elliptic-Curve Cryptography (ECC) in asymmetric cryptography is vulnerable to quantum computers [13]. According to [35], 2048-bit RSA will be broken with a chance of 50% in 2031. Also the study of the Federal Office for Information Security clearly indicates that today's cryptographic algorithms might imposed to a future threat due to the fast development in the research domain of quantum computing technologies [2]. Hence, building algorithms which on the one hand preserve their cryptographic assumption in the era of quantum computers, so called Post-Quantum Cryptography (PQC), and on the other hand run on traditional computers and embedded systems became the major challenge. Compared to asymmetric cryptography which is highly affected by quantum computers, symmetric cryptography, such as the Advanced Encryption Standard (AES), can preserve their security level in the age of quantum computers, by doubling the length of the key. Using Grover's algorithm quantum computers can only accelerate the exhaustive brute-force search for the secret symmetric key, cf. [21].

As a consequence, to prevent the threat of quantum computers, the American National Institute of Standards and Technology (NIST) initiated in 2016 an open competition to gather, evaluate and standardize quantum-resistant candidates. From the initial 69 candidates in the first round, only 26 got into the second round due to security weaknesses, merges, withdrawals and selection, cf. [14]. The competition collects PQC algorithms for signature schemes, encryption schemes, key exchange and key encapsulation mechanisms (KEMs) to replace asymmetric algorithms like RSA, Edwards-curve Digital Signature Algorithm (EdDSA), and Diffie-Hellman (DH). Since the PQC candidates follow various approaches, they are grouped by the underlying mathematical problems which shall ensure their security, cf. [13, 15]. Besides the security, the groups possess different characteristics according to the size of the keys, size of the ciphertexts/signatures, and their performance. While the candidates try to optimize those properties without decreasing security, some constraints already restrict the application of groups for specific use-cases.

Based on the latest outcome of the NIST competition, this paper investigates how PQC algorithms with suitable properties can be conveyed into the automotive domain. In its PQC-report, NIST names crypto agility—the capability of seamlessly replacing cryptographic algorithms (in the field after deployment)—as one of the most crucial challenges, cf. [15]. Achieving crypto agility becomes even more important and harder for resource efficient automotive embedded systems with limited computational power and product life cycles of about 20 years, cf. [22]. Therefore, getting an overview of the properties of PQC candidates and gaining experience about their runtime behaviors on such systems is of the utmost importance for this challenge. Further, this work shall show that suitable PQC candidates can run in the automotive environment, and replace existing cryptographic implementations with quantum-resistant algorithms.

The use case of secure software updates is taken as an example. This automotive use case leads to the highest risks in the age of quantum computers according to the results of our threat analysis and risk assessment (TARA). After a preselection process of PQC algorithms for the chosen use case, a demonstration setup compares the asymmetric signature scheme EdDSA to chosen PQC candidates as possible quantum-resistant replacements. The paper completes with a conclusion and an outlook of future PQC challenges in the automotive domain.

The rest of the paper is structured as follows. Section 2 provides an introduction to the various post quantum cryptographic algorithms and schemes. After an overview of related work in Section 3, the result of a threat analysis and risk assessment on ECU use cases is described in Section 4. Before we conclude the paper in Section 6 the results of our case study is provided in Section 5.

2 Background

This section gives an overview of PQC families and schemes that are resilient to the computation power of quantum computers and according to today's knowledge still provide security in the existence of quantum computers. This because up to now no efficient algorithm is found to solve the mathematical problems the schemes utilize in the cryptographic algorithms. The application contexts of these schemes are not always the same.While some schemes can only be used to handle signatures (i.e., generating and verifying signatures), other schemes only provide encryption and decryption operations [13, 14].

2.1 Hash-based Cryptography

Hash-based PQC schemes are among the oldest and most understood PQC schemes. This group of PQC schemes exploits one-way functions together with the concept of one-time signature schemes to generate a signature. An encryption service is not offered by this kind of PQC schemes. The security of hash based signatures relies on the unforgeability and second pre-image resistance of the used one-way hash function [12,13]. Quantum computers can only accelerate the exhaustive search for pre-image for a given hash value to the square root

using Grover's algorithm, and therefore these one-way functions are considered post-quantum secure [8, 20, 36].

Like in classic asymmetric schemes, hash-based signature schemes are generated by using a private key and can be verified with a public key. However, a property of this scheme is that each individual message needs to be signed with individual private keys and the amount of signatures that can be generated is limited. In other words, two different messages must be signed with two different private keys in order not to sacrifice the security of the scheme. Thus, a state management is required to avoid using the same key twice [12, 13]. Therefor, stateful hash-based signature schemes can only sign a limited a mount of messages pair private public key pair. The amount of required signatures needs to know during the key generation process when the private ephemeral keys and the root public key are generated. The most prominent representatives of hash-based signature schemes are SPHINCS, LMS, and XMSS— LMS and XMSS is currently standardized by IETF [14, 24].

2.2 Code-based Cryptography

Code-based PQC schemes utilize coding schemes and concepts known from the area of error-correcting, thus the principles of code-based schemes are well studied and the security behind the schemes is well understood. Their major drawback is the size of the parameters, such as huge public keys (most schemes require ≥ 1 Mbyte for the public key). Since code-based signing operations are inefficient in terms of resource consumption, their main application is public key encryption. Code-based encryption schemes are often leveraged as KEM to securely transport symmetric keys, encapsulated by the encryption with the receiver's public key, and only decipherable by the owner of the corresponding private key [13, 18, 38].

Error-correcting codes are traditionally used to detect and correct bit errors during the transmission of a message via an unreliable communication channel. This concept is adapted to establish a secure communication via an insecure channel by injecting errors in the original message in a way that a third party cannot reconstruct the original message anymore. The trusted receiving party can reconstruct the original message with the knowledge of the position of the injected errors and error correcting codes. The knowledge of the position of the errors are shared between the two parties by a public private key scheme. The security in this approach stems from the property that a random code cannot be distinguished from a code word, therefore the code generation element needs to be scrambled, but still efficient to use for the decryption operation [13, 18]. Most of the code-based schemes are based on McElice and Niederreiter cryptosystems [33, 37].

2.3 Lattice-based Cryptography

Lattice-based cryptography iscurrently the most popular candidate for research on PQC, due to its efficient implementations and simplicity. A lattice is gener-

ated by a set of points in n-dimensional space with a periodic structure and is defined as the set of all their integer combinations. The wide field of different lattice-based mathematical problems yielded various signature schemes, as well as encryption and KEM schemes as candidates for PQC. The most important lattice-based computational problem is Shortest Vector Problem (SVP), which is to approximate the minimal Euclidean length of non-zero vectors in a lattice. The non-polynomial (NP)-hard SVP problem is believed to be hard to solve even with abilities of quantum computers. Lattice-based PQC schemes attract public attention due to their balance between efficient computation and reasonable key sizes [13,14,19,36]. Prominent examples for lattice based signatures are Dilithium and qTesla, and NTRU and NewHope for lattice-based encryption/KEM [14].

2.4 Multivariate-based Cryptography

Multivariate-based cryptography builds asymmetric public key cryptosystems based on the NP-hardness of solving multivariate quadratic (MQ) systems of equations over a finite field $\mathcal{F}$. After [32] was published as a first proposal for multivariate public key cryptosystems (MPKC) in 1988, a lot of MPKC based signature and encryption schemes have been developed. On the one hand, they require large key sizes. On the other hand, they outmatch RSA and ECC on encryption and signature verification according to performance. Although most of the published MPKC based encryption schemes have been already broken, the signature schemes preserves the NP-hard MQ-problem. Four of these candidates are selected to the second round of the NIST post quantum submission. However, according to [23] only few security proofs for MPKC based algorithm have been published [14, 23, 32].

Most of the MPKCs use a composite function of multiple itself invertible maps as trapdoor function with the NP-hard problem of inverting after the composition. MPKC distinguishes between under-defined equations and over-defined equations. Multivariate signature schemes rely on under-defined equations, which have more variables than polynomials and, thus have more than one solution. In contrast, multivariate encryption schemes using over-defined equations have (at most) exactly one solution and less variables than polynomials [23, 34].

2.5 Isogeny-based Cryptography

Isogeny-based cryptography is a very young field compared to the post-quantum cryptosystems discussed above, that has first been proposed in 1997 and, thus, have relatively new security assumptions. The basic idea of isogeny-based schemes is to reuse operations related to ECC. However, the so called supersingular elliptic curve isogeny cryptography changes the mathematical problem from the hardness of point computations on a single curve to the hardness of computations between a large numbers of curves via isogenies. While the first problem—used in popular ECC algorithm like EdDSA, Elliptic Curve Diffie-Hellman (ECDH)—can be broken by quantum computers, the latter is assumed to resist the capabilities of quantum computers. Furthermore, the computation among multiple

curves can serve as the base to convey the exchange of symmetric key pendant of DH into PQC, as proposed by the Supersingular Isogeny Diffie-Hellmann (SIDH) scheme in [28]. SIDH has the advantage of small key sizes and the capability of supporting an ephemeral key exchange to achieve perfect forward secrecy [17,28].

2.6 Summary

All the various classes of schemes have their own advantages and disadvantages. Algorithms of each class are usually inheriting the class specific features, and the related advantages and disadvantages. Therefore, the selection of PQC algorithms fitting to a security use case should start with an investigation on the features of the classes of PQC schemes.

In addition, not all schemes can be used for encryption or signature operations. Table 1 summarizes the advantages and disadvantageous of each class of PQC schemes and the supported operations in terms of encryption or signature operations.

Table 1. Overview of the characteristics of the various classes of schemes from PQC [36]

Class of Scheme	Signature	Encryption or Key Exchange	Description
Hash-based	Y	N	Based on well-established security of symmetric hash function. Verification operation is reasonably fast. Statful hash-based signatures can only sign a limited number of messages per public private key pair.
Multivariate-based	Y	N*	Based on multivariate polynomial equations. Large public keys (27.9 kBytes to 75kBytes)
Code-based	N*	Y	Old coding schemes have a mature security level but large public keys (1 MBytes to several MBytes). The newer, more efficient coding-schemes with smaller public keys are less trusted at the moment. (e.g. QC-MDPC)
Lattice-based	Y	Y	The schemes have a good balance between performance and key sizes (1-4 kBytes). However the security is not as matures as it is for code-based or hash-based schemes.
Isogeny-based	Y	Y	Related to ECC algorithms, slow but very small ciphertexts/keys. Isogeny based schemes are a relatively new field of research, hence security proofs are not as well investigated as security proofs of hash-based schemes.

*: Proposals exist but they are currently not considered competitive.

3 Related Work

While in the last two years, the research and development of PQC candidates is mainly driven by the competition of the NIST standardization process [14] and the standardization of XMSS by the IETF [24], the origin of most PQC candidates (e.g. [32, 33, 37]) is older than the research field for PQC schemes itself. The constant evolvement and improvement of quantum computers, making the threat more and more realistic, led to an arising amount of research and development in this field already in the last few years.

In 2015, the EU funded a PQC research to encounter the threat of quantum computers on the security of classic cryptographic algorithms. One part of the published European Telecommunications Standards Institute (ETSI) whitepaper covers a case study about the vulnerability of modern communication protocols and standards—such as X.509, Secure Shell (SSH) and Transport Layer Security (TLS)—and further provides high level requirements for future PQC algorithms, which could replace the vulnerable algorithms in each protocol [13]. In 2016, Google conveyed the ongoing research activities on PQC candidates into a first draft for a real world implementation of the TLS handshake in Google Chrome Canary and combined RSA/(EC)Digital Signature Algorithm (DSA) with the lattice-based NewHope algorithm to assure the session key exchange is quantum secure [10]. The same algorithm has been ported on a security chip by the silicon vendor Infineon in 2017. In the corresponding press release, the silicon vendor emphasizes that the upcoming threat arising of quantum computers already needs to be taken in serious consideration today, especially for the automotive domain with the long production life cycles [4, 22]. For embedded systems, [29] ported optimized versions of most of the NIST's PQC candidates on a Cortex-M4 based microcontroller. Further, this project measured and documented properties of each ported algorithm, such as size of the keys and ciphertexts/signatures and the running time of the cryptographic operations on the targeted system [29]. In [26] the authors demonstrated that XMSS signature verification can achieve the same timings as EdDSA and RSA. Running on a smart-card with hardware accelerators for AES and EdDSA, they used AES-based hash functions to accelerate the hash generation.

4 Analysis of ECU Related Security Use Cases

In this section, we focus on a TARA of the automotive basic security use cases: secure software download, secure diagnosis, secure boot, and secure onboard communication. The architecture for the use cases was abstracted in order to get a very generic data flow graph fitting the use cases on an abstract level. The idea behind that is to get a general understanding of which use case out of these four has the highest risk impact, if the classic asymmetric cryptographic algorithms are not secure anymore.

4.1 Use Case description:

The basic automotive use cases from above are considered for the application of PQC-based algorithms, because these are applicable to most ECUs. Hence, this provides the opportunity for the coverage of most applications in the automotive domain.

Secure Software Update: Secure software update describes the update procedure of an ECU by flashing the software stored and running on an ECU. This update procedure needs to be secured in a way that only authentic software can be used for the update of the ECU. As of today, this typically means that only legitimated software of the manufacturer shall be flashed into the ECU [9].

Secure Diagnosis: The secured diagnosis service is used to read and update configuration and parameter settings or to initialize the secure software update process. Only authorized entities shall have access to the service of reading or updating the device settings and configuration or initiate the secure software update process, cf. [1].

Secure Boot: Secure boot shall ensure that only authentic and valid software is executed on the ECU. Usually the integrity and authenticity check of the software is done during the boot phase of an ECU [42].

Secure Onboard Communication: Secure Onboard communication allows for an authentic communication between ECU in the in-vehicle network. It safeguards message integrity and authenticity to secure safety-critical messages [3].

4.2 Threat Analysis and Risk Assessment

TARA has become a golden standard in the industry to identify appropriate countermeasures and actions to ensure the aimed security level of a product. For decades TARA is a common method in the software development industry for to identify threats and corresponding risk to lose assets that ensure the correct functionality of the developed software. However, performing a TARA is a quite young concept in the automotive industry, but is strongly recommended in automotive secure development guidelines like the *SAEJ3061* [39] and is also considered in the upcoming ISO-SAE standard *ISO-SAE21434 -Road vehicles – Cybersecurity engineering*. Up to now there is no standard on how to perform a TARA in the automotive industry, otherwise we oriented on the TARA methodology proposed in the research project *HAVENS* [31] and developed our own method based on different established standards from other industry branches - like the ISO27005 [27], and recommendations from governmental offices like the federal office for information security. The threat model used for our TARA is based on the STRIDE model. In addition, the threat identification step is

following the STRIDE approach by Microsoft [30]. The risk assessment part of our TARA is oriented on the proposal from the *HEAVENS* project and the risk assessment of the common criteria evaluation standard [16].

We performed a TARA for each of the automotive basic security use cases, focusing on the elements that utilize an asymmetric cryptographic operation in order to verify data, authenticate users. Due to the STRIDE approach we focus on the data transfer from and to the asymmetric cryptographic operation. The asymmetric cryptographic operation itself is modeled as an atomic process with all its STRIDE typical properties according to threats and possible security goals. Therefore, only the process element in the data flow diagram representing the authentication verification, or authorization process using asymmetric cryptography is investigated, as well as, all incoming and outgoing data communication to or from the process node. These identified threats are then quantified in terms of likelihood of exploitation and severity of the impact on the security concept of the use case in case the exploitation was successful. This combination of impact and likelihoods translates to a risk level starting with quality management (*QM*) over *Low*, *Medium*, *High* to *Critical*, where *QM* does only require very little to no actions, while *Critical* indicates a risk with a very high damage and a high probability so that immanent actions for placing sophisticated countermeasures are needed.

Tab. 2 summarizes the results of the TARA for each use case. This table depicts the summarized amount of identified risks per ECU-focused automotive security use case as well as the number of risks per risk level for each use case. The use case of *Secure Software Update* has the highest number of risks with the critical risk level and the most identified risk regarding quantum computer attacks, under the assumption security control for software verification is broken because the asymmetric cryptographic scheme is broken. The high and critical risk level ratings result from the ability of an attack that any arbitrary software can be uploaded and execute on the ECU. Also the upcoming security solution of security updates would not work in this case because the adversary can either replay an update of the malicious software. The second critical use case is *Secure Diagnosis*, followed by the use cases *Secure Boot* and *Secure Onboard Communication*. The high risk level for the use case *Secure Diagnosis* is based on the fact that an adversary might have the ability to change configuration parameter of an ECU, if he can overcome the secure access control. The changes in the configuration parameter can than result in functional changes of the ECU with

Table 2. Overview on TARA results on all the automotive basic security use cases

Use case	Number of Risk					
	QM	Low	Medium	High	Critical	**Total**
Secure Software Update	1	2	2	6	2	**13**
Secure Diagnosis			3	4	1	**8**
Secure Boot			1	4	1	**6**
Secure Onboard Communication				3	1	**4**

high safety related impacts, e.g., configuration settings of a brake controller. It is important to highlight that the TARA of *Secure Onboard Communication* is focused on the authentic communication of the in-vehicle network on typical automotive bus technologies, such as Controller Area Network (CAN) or CAN-FD bus, because the protected communication measure for Ethernet bus networks inside the vehicle can easily be adapted from solutions of the IT security world.

5 Case Study on Secure Software Updates

PQC schemes are important for secure software update or secure flashing since the basic signature and verification are mainly performed by public key cryptosystems, such as RSA, ECC. Our TARA retrieved the corresponding threat described as: 'Disclosure of public key', with a risk level of high. The threat impact is that the public key information might be used together with Shor's algorithm to retrieve the corresponding private key on a quantum computer. A follow up threat would be that the attacker would be able to create valid signatures for own spoofed firmware images, which could be deployed to the vehicle and would be considered as valid by the affected ECU. As a consequence, the attacker could compromise the ECU and execute arbitrary code. Furthermore, an attacker is able to send some spoofed messages to the in-vehicle network, e.g. to initiate brake commands. Therefore, it is suggested to use post quantum-based signature schemes.

5.1 Automotive Requirements

For the use case of secure software updates, the signature generation and verification have automotive specific requirements. The most important requirement is that the security of the signature scheme must be well understood and resistant against classical cryptographic attacks, as well as resistant against quantum computer attacks. The public key for the signature verification, which must be stored in ECU, shall have a small size. Either the public key itself or its hash value must be stored in a secure tamper-resistant memory location, e.g. one-time programmable memory. The selected scheme and its implementation for the signature verification shall focus on speed and code size to optimize the cost. In contrast, the performance of key-pair and signature generation is out of scope, since it is performed in a resource-rich backend. In addition, the verification performed at the telematic control unit (TCU) might be different to the one performed on ECU due to different platforms and costs, respectively.

5.2 Selecting Adequate PQC Schemes

From the requirements defined in Section 5.1, the proper PQC schemes for the use case of secure software updates shall be pre-selected. Today, EdDSA is a widely used signature scheme which is taken as a reference in this paper. Based on values measured by [40], the algorithm's properties are compared against

promising PQC schemes, which proceeded the second round of the NIST competition. EdDSA has small public key sizes, small signature sizes and a fast signature verification. However, from Shor's algorithm, we know that it can be broken by quantum computers [41].

The PQC classes of code-based and supersingular isogeny-based cryptography are neglected, since none of their signature candidates entered in the competition [14]. Multivariate-based signature schemes stand out by their fast signature verification and extreme small signature sizes. Nevertheless, their extreme large public key size and their rare security proofs conflict with two important automotive requirements. Thus, they are not selected as candidates for the use case of secure software updates in an automotive environment [23, 40].

Although hash-based PQC schemes have medium large signature sizes as well as medium verification duration, they outstand by their well understood security proofs and their small public key sizes. The hash-based algorithm XMSS is standardized by Internet Engineering Task Force (IETF), making it a reliable candidate for future (automotive) protocols. However the constraint of most hash-based schemes, especially of XMSS, to maintain a state and only provide a limited number of signatures per key, must be taken into account for the use case of secure software updates on automotive ECUs [12,24]. Compared to the update frequency of modern apps and smartphones, updates for automotive ECUs are quite rare. Therefore, the number of signatures for one XMSS key should be sufficient. Yet, considering the long life span of vehicles in the field and an increasing frequency for future updates, the amount of updates can increase. In that case, some public-key-infrastructure scheme should be established, to enable the roll-out of new keys by updates.

Lattice-based signature schemes have medium size public keys and medium sized signatures. Their signature verification is fast. Therefore, the PQC candidate Dilithium with an outstanding performance meets the automotive requirements for secure software updates, which is chosen for the experiment in this paper [40].

5.3 Experimental Results

In this section, discuss our reference implementation we used for a resource and performance comparison of the signature verification process. We implement a hash-based cryptographic algorithm (XMSS) with a security level of 4 and a tree

Table 3. Comparisons of XMSS, Dilithium and EdDSA on ARM Cortex M4

Algorithm (Security Level)	Public Key Size (Bytes)	Signature Length (Bytes)	Cycles	Cycles per Byte	Avg exec. time (msec @ 168 MHz)
XMSS C (4)	68	2,340	87.120.741	37,231	518
XMSS ASM (4)	68	2,340	32,635,088	13,946	194
Dilithium (3)	1,472	2,701	6,225,805	2,305	37
EdDSA	32	64	3,126,848	48,857	18

size of 64, a lattice-based cryptographic algorithm (Dilithium) with a security level of 3. The provided security levels of the PQC algorithms are referring to the security levels defined in the PQC NIST competitions [14]. Hence, level 3 equals the today's security level of AES-192, while level 4 is the same security level as finding a collision in SHA-384. For comparison with nowadays in use signature schemes we also implemented a current public key cryptographic signature standard (EdDSA). In our implementation the signature scheme uses the ed25519 curve as underling elliptic curve. One variant of XMSS implements the hash algorithm in optimized assembly instructions and the other variant implements the hash algorithm in non-optimized C code, to simulate automotive constraints. The other algorithms are solely implemented in C code.

In the use case of software updates, signatures will be generated at the backend followed by sending the signed package to the ECU to verify and install the package. While the backend is simulated by a Raspberry Pi 3 B [7] running Ubuntu 16.04, the ECU is simulated by a microcontroller running a 32-bit ARM Cortex M4 at 168 MHz, 196 kB SRAM memory, and 1 MByte of flash memory [5]. The code is cross-compiled using the GNU-ARM embedded toolchain version '6-2017-q2-update' at an optimization level of 2 [6]. The over-the-air update communication between the backend and the ECU is emulated by a serial wired UART connection. According to the risk analysis of Sect. 5, we focus on the performance of the verification on the ECU platform.

Based on the pre-selection in Sect. 5.2, the signature algorithms XMSS, Dilithium, EdDSA with ed25519 are ported to the ECU for comparison. From the performance of the signature verification process in C code, XMSS has the longest running time with 518 ms, which is slower by more than factor 10 compared to the competitors. Although the assembly optimized variations reduces the duration of XMSS to 194 ms and, thus, compared to the C variation reduced by the factor of 2.5, it still takes more 5 times as long as the competitors. While Dilithium takes approximately twice as long as EdDSA (37 ms vs. 18 ms), the ratio of clock cycles per signature bytes is significantly better than EdDSA (cf. Tab. 3) according to the clock cycles per signature byte. For the comparison of the signature verification process only the signature validation itself is measured and compared, the hashing of the message is excluded in this evaluation.

6 Conclusion and Future Work

This paper has shown that the threat arising from quantum computers affect the security of vehicles. In the use case of secure software updates, our TARA revealed the risk of arbitrary code execution by spoofing the signature of updates, once quantum computer have the ability to reveal the secret private key. In our demonstration setup we successfully proved that PQC algorithms can replace vulnerable signature schemes. We also pointed out that the quantum-resistant signature verification of Dilithium can outperform the well established EdDSA on the cycles per byte metric. Although our XMSS implementation did not achieve the same performance as EdDSA. In [26] the performance of the XMSS

implementation was similar to the performance of the EdDSA implementation. However, it is important to emphasize that the authors in [26] exploited a AES-based hash functions to accelerate the hash generation. This improvement might also be used to speed up the signature verification in our setup and make XMSS competitive to EdDSA and Dilithium on the signature verification.

The development of PQC algorithms is just at the beginning, especially for the automotive domain. Thus, this topic needs much more research. Future work shall analyze the security of other automotive specific use-cases against the threat of quantum computers. Especially the question shall be investigated, how PQC algorithms can serve as protection in applications with real-time constraints of today's automotive standards, e.g., the communication between electronic control units (ECU) within the in-vehicle network. Moreover, future researches can analyze whether the additional code size of promising PQC candidates is a selection criterion for automotive ECUs with the small memory. The analysis shall compare the PQC candidates against today's on-quantum resistant counterparts with the constraints for automotive software code, e.g., assembler optimizations are prohibited. From a hardware perspective, an open question would be how PQC algorithms perform on automotive ECUs when leveraging their hardware accelerators, e.g. for hash computations. Furthermore, future work can investigate the impact of PQC algorithms on the cryptographic architecture decisions of ECUs which have integrated hardware security modules following the EVITA specification [42].

References

1. Road vehicles – unified diagnostic services (uds). International Organization for Standardization (2013)
2. Entwicklungsstand quantencomputer. Federal Office for Information Security (2017), https://www.bsi.bund.de/DE/Publikationen/Studien/Quantencomputer/quantencomputer_node.html
3. Specification of secure onboard communication. AUTOSAR (2017)
4. World's first post-quantum cryptography on a contactless security chip. Infineon Technologies AG (2017), https://www.infineon.com/cms/en/product/promopages/post-quantum-cryptography/
5. Cortex-m4—the arm cortex-m4 processor is arm's high performance embedded processor. Arm Holdings plc (2019), https://developer.arm.com/ip-products/processors/cortex-m/cortex-m4
6. Gnu arm embedded toolchain. Arm Holdings plc (2019), https://developer.arm.com/tools-and-software/open-source-software/gnu-toolchain/gnu-rm
7. Raspberry pi 3 model b. RASPBERRY PI FOUNDATION (2019), https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
8. Amy, M., Matteo, O.D., Gheorghiu, V., Mosca, M., Parent, A., Schanck, J.M.: Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3. In: Avanzi, R., Heys, H.M. (eds.) Selected Areas in Cryptography - SAC 2016 - 23rd International Conference, St. John's, NL, Canada, August 10-12, 2016, Revised Selected Papers. Lecture Notes in Computer Science, vol. 10532, pp. 317–337. Springer (2016)

9. Bellissimo, A., Burgess, J., Fu, K.: Secure software updates: Disappointments and new challenges. In: HotSec (2006)
10. Braithwaite, M.: Experimenting with post-quantum cryptography. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html (2016), retrieved 2019-03-24
11. Buchmann, J.A.: Introduction to Cryptography. Springer (2002)
12. Buchmann, J.A., Dahmen, E., Hülsing, A.: XMSS - A practical forward secure signature scheme based on minimal security assumptions. In: Yang, B. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7071, pp. 117–129. Springer (2011)
13. Campagna, M., Chen, L., Dagdelen, Ö., Ding, J., Fernick, J.K., Gisin, N., Hayford, D., Jennewein, T., Lütkenhaus, N., Mosca, M., Neill, B., Pecen, M., Perlner, R., Ribordy, G., Schanck, J.M., Stebila, D., Walenta, N., Whyte, W., Zhang, Z.: Quantum safe cryptography and security: An introduction, benefits, enablers and challengers. Tech. rep., ETSI (European Telecommunications Standards Institute) (June 2015), http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf
14. Chen, D.L., Moody, D.D., Liu, D.Y.K.: NIST Post-Quantum Cryptography Standardization, [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization [2019, January 30]. National Institute of Standards and Technology - Computer Security Resource Center, Gaithersburg, MD. (2017)
15. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Nistir 8105: Report on post-quantum cryptography. Tech. rep., Tech. rep. NIST (2016)
16. Common Criteria: Introduction and general model v3.1 Release 4 (2012)
17. Costello, C., Longa, P., Naehrig, M.: Efficient algorithms for supersingular isogeny diffie-hellman. IACR Cryptology ePrint Archive **2016**, 413 (2016), http://eprint.iacr.org/2016/413
18. Döttling, N.M.: Cryptography based on the Hardness of Decoding. Ph.D. thesis, Karlsruhe Institute of Technology (2014), http://digbib.ubka.uni-karlsruhe.de/volltexte/1000041110
19. Goldreich, O., Micciancio, D., Safra, S., Seifert, J.: Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. Electronic Colloquium on Computational Complexity (ECCC) **6**(2) (1999), http://eccc.hpi-web.de/eccc-reports/1999/TR99-002/index.html
20. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Miller, G.L. (ed.) Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996. pp. 212–219. ACM (1996)
21. Grover, L.K.: How hard is to search an unsorted database containing n items? a quantum mechanical algorithm (1996), https://arxiv.org/pdf/quant-ph/9605043v1.pdf
22. Hammerschmidt, C.: Infineon preparing post-quantum cryptography for cars, infrastructure (2017), https://www.eenewseurope.com/news/infineon-preparing-post-quantum-cryptography-cars-infrastructure
23. Hashimoto, Y.: Multivariate Public Key Cryptosystems, pp. 17–42. Springer Singapore (01 2018)

24. Huelsing, A., Butin, D., Gazdag, S.L., Rijneveld, J., Mohaisen, A.: XMSS: eXtended Merkle Signature Scheme. RFC 8391 (2018), https://rfc-editor.org/rfc/rfc8391.txt
25. Hui, J.: Qc- what are qubits in quantum computing (2018), https://medium.com/@jonathan_hui/qc-what-are-qubits-in-quantum-computing-cdb3cb566595
26. Hülsing, A., Busold, C., Buchmann, J.: Forward Secure Siganures on Smart Card. In: International Conference on Selected Areas in Cryptography. pp. 66–80. 7707 (2013)
27. International Standard Organization: ISO 27005:2011: Information technology — Security techniques — information security risk management (2011)
28. Jao, D., Feo, L.D.: Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In: Yang, B. (ed.) Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7071, pp. 19–34. Springer (2011)
29. Kannwischer, M.J., Rijneveld, J., Schwabe, P., Stoffelen, K.: PQM4: Post-quantum crypto library for the ARM Cortex-M4 (2019), https://github.com/mupq/pqm4
30. Kohnfelder, L., Garg, P.: The threats to our products (1999), https://adam.shostack.org/microsoft/The-Threats-To-Our-Products.docx
31. Lautenbach, A., Islam, M.: D2 - security models. Tech. rep., HEAling Vulnerabilities to ENhance Software Security and Safety (2016)
32. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption", booktitle="advances in cryptology — eurocrypt '88. pp. 419–453. Springer Berlin Heidelberg (1988)
33. McElice, R.J.: A public-key cryptosystem based on algebraic coding theory. In: The Deep Space Network Progress Report. vol. 44, pp. 114–116 (1978)
34. Miura, H., Hashimoto, Y., Takagi, T.: Extended algorithm for solving underdefined multivariate quadratic equations. In: Gaborit, P. (ed.) Post-Quantum Cryptography. pp. 118–135. Springer Berlin Heidelberg (2013)
35. Mosca, M.: Cybersecurity in an era with quantum computers: will we be ready? Cryptology ePrint Archive, Report 2015/1075 (2015), https://eprint.iacr.org/2015/1075
36. Niederhagen, R., Waidner, M.: Practical post-quantum cryptography. Tech. rep., Fraunhofer SIT (2017), https://www.sit.fraunhofer.de/fileadmin/dokumente/studien_und_technical_reports/Practical.PostQuantum.Cryptography_WP_FraunhoferSIT.pdf
37. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. In: Problems of Control and Information Theory. vol. 15, pp. 19–34 (1986)
38. Persichetti, E.: Code-based key encapsulation from mceliece's cryptosystem. In: Blömer, J., Kotsireas, I.S., Kutsia, T., Simos, D.E. (eds.) Mathematical Aspects of Computer and Information Sciences. pp. 454–459. Springer International Publishing, Cham (2017)
39. SAE International: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems (February 2016)
40. Safecrypto-Consortium: Nist software performance tests (2018), https://www.safecrypto.eu/pqclounge/round-1-candidates/software-analysis-signatures/
41. Shor, P.W.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
42. Wolf, M., Gendrullis, T.: Design, implementation, and evaluation of a vehicular hardware security module. In: Kim, H. (ed.) Information Security and Cryptology - ICISC 2011. pp. 302–318. Springer Berlin Heidelberg (2012)