# Quantum-Safe Signing of Notification Messages in Intelligent Transport Systems

Sara Nikula[1(✉)], Kimmo Halunen[2,3], and Visa Vallivaara[1]

[1] VTT Technical Research Centre of Finland, Oulu, Finland
{sara.nikula,visa.vallivaara}@vtt.fi
[2] Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland
kimmo.halunen@oulu.fi
[3] Department of Military Technology, National Defence University, Helsinki, Finland

**Abstract.** In this work, we integrated three quantum-safe digital signature algorithms, CRYSTALS-Dilithium, FALCON and Rainbow, into notification messages used in intelligent transport systems. We evaluated the performance of the algorithms by measuring the time required to sign and verify messages, as well as the size of the signed messages, and compared the quantum-safe options to the elliptic curves currently accepted by the standards. Our results show that quantum-safe digital signature algorithms could be used for signing notification messages in intelligent transport systems, with only moderate changes to performance. The results also provide an evaluation of three quantum-safe digital signature algorithms' suitability for this purpose, thus helping to choose suitable algorithms when migrating intelligent transport systems towards quantum resistance.

**Keywords:** Post-quantum cryptography · Digital signature · Intelligent transport systems

## 1 Introduction

Intelligent transport systems utilize wireless communication in order to improve fluency and safety in traffic. In these systems, notification messages require digital signatures, which assure the origin and the authenticity of the message. According to current European standards, the messages are signed using elliptic curve cryptography [19].

Elliptic curves, defined as a set of x- and y-coordinates on a plane, can be used in public key cryptography, digital signatures and key exchange. The mathematical problem underlying these operations is elliptic curve discrete logarithm problem, which is known to be a hard problem for modern computers [5]. However, Shor's algorithm allows to solve this problem efficiently with a quantum computer [40]. Thus, signatures based on elliptic curves are not quantum-safe as they


J. Lin and Q. Tang (Eds.): AC3 2022, LNICST 448, pp. 11–25, 2022.
https://doi.org/10.1007/978-3-031-17081-2_2

could be forged with a powerful enough quantum computer. Even though this kind of quantum computers do not exist at the moment, they are being developed and quantum technology has taken some significant advances in the last decades [3]. To address this threat, the National Institute of Standards and Technology (NIST) initiated a call for proposals for quantum-safe cryptographic algorithms to be standardized in late 2016. The competition was divided into two classes, one for digital signature algorithms and the other for key encapsulation mechanism (KEM) algorithms [1]. In 2020, three digital signature algorithms were chosen for the third and final round of the competition: CRYSTALS-Dilithium and FALCON, based on lattice problems, and Rainbow, based on multivariate public key cryptography [29].

Cooperative Intelligent Transport Systems (C-ITS) refers to vehicles and road infrastructure automatically generating and sharing information, thus making traffic more fluent and safe [10]. For example, this can mean a vehicle notifying geographically nearby other vehicles about poor road conditions, or an emergency vehicle warning other road users as it is approaching [16]. As these messages can be sent either by vehicles or road infrastructure, we refer to the parties of this communication as C-ITS stations. In practice, On-Board Unit (OBU), a particular device manufactured for communication purposes, carries out this communication [39]. Several commercial producers manufacture and sell OBU devices, which often are based on ARM architecture [6,11,15].

Communication between several parties requires diverse specifications in order to operate. European standardization organizations prepare and publish standards and technical specifications regarding communication between C-ITS stations. Safety and reliability of this communication is ensured by technical specifications developed by ETSI (European Telecommunications Standards Institute). These documents specify digital signing of specific message types, as well as a public key infrastructure for maintaining certificates and verifying signatures [10,19,20].

CAM (Cooperative Awareness Message) and DENM (Decentralized Environmental Notification Message) are special message types defined in standards ETSI EN 302 637-2 [18] and ETSI EN 302 637-3 [16]. CAM messages are relevant for co-operation, communicating information such as direction and speed of the sending vehicle. DENM messages notify geographically nearby vehicles about unusual situations, such as road works, poor road conditions or wild animals detected on the road. Both of these message types are sent in a digitally signed form [16,18]. In the current standards, elliptic curves are the only accepted digital signature type. The standards accept three different elliptic curves: NIST P-256, brainpoolP256r1 and brainpoolP384r1 [19,26]. Alongside the signed message, the sender sends their certificate, which is digitally signed by the certificate authorities. The certificate contains information about the public key which is used to verify the signature [19].

In this work, these notification messages are combined with quantum-safe digital signatures. Replacing elliptic curve digital signatures with quantum-safe alternatives makes the communication more robust towards attacks of quantum

computers, thus mitigating the risk of forged signatures. The results section includes performance tests of CRYSTALS-Dilithium, FALCON and Rainbow, when used to sign notification messages of intelligent transport systems. This gives insights into the usability of quantum-safe digital signatures on this domain.

## 2 Previous Work

The field of post-quantum cryptography (PQC) that considers quantum-safe digital signature algorithms among other cryptographic primitives has been under active research since the threat against public-key cryptography by quantum computers has been known. There have been several proposals over the years, and covering the results of this field is out of scope of this paper. An interested reader can find more information about post-quantum cryptography in [8]. In this paper, we present relevant background and previous work related to the *implementation* of such algorithms, especially in the context of the ongoing standardisation effort.

Different applications of quantum-safe digital signature algorithms have been covered in previously published papers. In [41], the authors tested the suitability of quantum-safe digital signature algorithms with TLS 1.3 (Transport Layer Security). The results indicate that CRYSTALS-Dilithium and FALCON are suitable alternatives for this use. [28] showed that quantum-safe digital signature algorithms can be used to secure embedded devices. In [35], lattice-based quantum-safe digital signatures were used in industrial devices together with a X.509 certificate. These papers show that quantum-safe digital signatures are applicable in many different domains, but there are also some challenges that need to be addressed.

Intelligent transport is also a vast field of study with many results and research directions. In this section, we present some relevant previous work on the cybersecurity of intelligent transport systems and threats that it poses, which motivate also our research in this topic. There exist several ways for a malicious actor to cause disorder and potentially dangerous situations in traffic. A denial of service (DoS) attack could block the communication by sending numerous false messages, thus preventing the vehicles from processing authentic messages. Even a smaller amount of messages could cause disorder and slow down the traffic if, for example, a malicious vehicle pretended to be an emergency vehicle in an urgent duty [22]. Intentionally tampering safety-related messages can lead to dangerous situations by causing confusion about the real location and direction of nearly-operating vehicles. A malicious actor could also record and re-send messages sent by other vehicles [24].

The current technical specifications of ETSI mitigate the risk of tampered messages by digital signatures [19]. These are an efficient way to recognize forged messages as they can be used to verify the authenticity and the integrity of the message. However, as the current specification only accepts elliptic curve digital signatures, the era of quantum computers poses a risk to this communication.

This paper combines post-quantum digital signatures and cooperative intelligent transport systems, thus enhancing their security towards attacks of quantum computers.

A characteristic feature of communication between C-ITS stations is time-sensitivity. Critical messages must be delivered in a short enough time in order to be effective. Requirement for speed and security can lead to a contradiction: because of performance issues, excessive amounts of overhead bytes and processing should be avoided, but at the same time, some overhead is required to ensure the security of the communication [24]. This work aims to inspect the potential trade-off between security and efficiency when migrating to quantum-safe digital signatures.

ETSI is starting to migrate its standards towards quantum-resistancy and has its own technical committee for this purpose, CYBER QSC (Quantum Safe Cryptography) [14]. In 2021, CYBER QSC published two technical reports concerning digital signature and KEM algorithms in the NIST post-quantum cryptography standardization process [13]. Furthermore, CYBER QSC is preparing a report regarding migration of C-ITS use cases towards quantum-safety [14]. When conducting the performance tests presented in this paper, ETSI had not yet published these reports. Thus we have designed the tests and chosen the compared algorithms independently from these reports.

## 3 Test Program

In this work, we integrated three quantum-safe digital signature algorithms, CRYSTALS-Dilithium, FALCON and Rainbow, into the DENM message structures specified by ETSI, and used them to sign DENM messages quantum-safely. We evaluated the suitability of these algorithms by three different aspects: the time required to sign a message, the time required to verify a message, and the size of the signed message. On the grounds of these aspects, we assessed the suitability of these quantum-safe digital signature algorithms for this purpose. For comparison, we implemented the same functionalities with two different elliptic curves accepted by the current standard, NIST P-256 and brainpoolP256r1, and measured the same aspects when applying these. In this way we were able to assess how the migration to quantum-safe signatures would affect the fluency of the communication. According to the standards, CAM and DENM messages are sent using a similar signed message structure [16,18], and thus the achieved results can be generalized to concern CAM messages too.

### 3.1 Implementing the Digital Signature Algorithms

In its call for quantum-safe cryptography, NIST has defined five security levels for the cryptographic algorithms. The lowest level, security level 1, corresponds to 128 bits of security. The highest level corresponds to 256 bits of security, and the other levels lie somewhere in between. NIST recommended that the submissions would mainly concentrate on security levels 1–3 [30]. An elliptic curve using 256

bits long parameters offers a security level of 128 bits [5], thus corresponding to the security level 1 in this competition. All three quantum-safe digital signature algorithms included in this comparison offer reference implementations meeting several different security levels. In this work, we have utilized an implementation on security level 1 or the closest available level. With FALCON and Rainbow, this meant security level 1, and with CRYSTALS-Dilithium, security level 2. In this way, we compared the elliptic curve digital signature algorithms using 256-bit parameters with quantum-safe algorithms at the same security level. The third elliptic curve accepted by the current standard, brainpoolP384r1, was not included in the comparisons, because it meets a higher security level without offering security against quantum computers.

ETSI offers the message structures needed for a signed DENM message and the related structures in ASN.1 notation language in the attachments of the standardisation documents [16,19]. These structures do not include implementations of elliptic curve cryptography. In this work, we utilized elliptic curve digital signature algorithms from OpenSSL [32]. We chose OpenSSL because it is a widely known, well-documented and freely available cryptography library [33] and thus provides a good reference point for our quantum-safe implementations.

The three quantum-safe digital signature algorithms included in this comparison offer a reference implementation in C on their websites [12,21,38]. These implementations include functions for key generation, signing and verification. All three algorithms also offer an AVX2 optimized version, which improves performance by utilizing parallel processing [27]. However, some processors used in OBUs may not support this optimization. Processors based on ARM architecture use a different instruction set [2] and thus cannot benefit from AVX2 optimization. Thus, we compared only the standard reference implementations in C. While this might not give a fully truthful image of the performance of these algorithms, these results are hopefully better comparable across different processor architectures.

### 3.2 Performance Tests

In order to utilize quantum-safe digital signature algorithms, we adjusted the structures defined in ETSI's documents. In the current technical specifications, the signature structure allows a choice between three different elliptic curves [19,26]. We removed brainpoolP384r1 from the alternatives list, and added three quantum-safe alternatives: CRYSTALS-Dilithium, FALCON, and Rainbow. We modified the public key structure in the same way.

As stated in [24], real-time communication between C-ITS stations needs to be secure and efficient at the same time. In this work, we measured efficiency by three factors: how long it takes to sign a message, how long it takes to verify the signature and how many additional bytes need to be added to a message, contrasted to a situation where no digital signature is sent. These additional bytes consist of a digital signature and a certificate, which is needed to authenticate the message and verify the signature.

We implemented the tests described in this section in the C programming language. We used asn1c compiler [4] to obtain the needed message structures, defined in the documents [16,19], in C. Five different versions of the program included the same functionality, but using a different digital signature algorithm: CRYSTALS-Dilithium, FALCON, Rainbow, elliptic curve NIST P-256 or elliptic curve brainpoolP256r1. We conducted the tests with operating system Ubuntu 16.04 LTS and processor Intel Core i7-8665U, with base frequency of 1.90 GHz, and compiled the test programs with GCC 5.4.0. GCC offers several options for optimizing the code. In these tests, we used the optimization mentioned in the reference implementation files. With OpenSSL, CRYSTALS-Dilithium and Rainbow, this meant optimization -O3, and with FALCON, optimization -O2.

We decided that the implementation of a full certificate chain was out of scope for this research and thus we did not use the issuer and signature fields in the certificate. This somewhat shortens the certificate, but does not affect the comparison between algorithms.

In order to measure the speed of signing and verifying, we created a test program that created, signed and verified DENM messages. First, the program created a new pair of keys, and copied the public part of these keys in a certificate. Then, the program created a new DENM message, signed it and packaged it into a signed data structure together with the certificate. Now the program moved to the second part, opening and verifying the message. The program fetched the signature and the public key from the signed message and checked if it could successfully verify the signature with the public key. If not, the program notified the user about this and exited. If yes, it opened the DENM message and presented it to the user. When signing and verifying with elliptic curves, we called OpenSSL functions `ECDSA_sign` and `ECDSA_verify`. When using quantum-safe algorithms, we called functions `crypto_sign` and `crypto_sign_open`, defined in the reference implementations.

Fields of a DENM message have no significant impact on the signing speed as the message is hashed before signing it, as stated in the technical specification [19]. However, for the sake of credibility, we also filled in the necessary fields in the DENM message. A DENM message should include two mandatory fields including information about the message, such as the protocol version, information of the sending C-ITS station, location and generation time of the notification message. On top of these, the message can contain additional information about the observed situation [16]. We filled the protocol version field as specified in the standard, and the rest of the fields with imaginary information. In our imaginary DENM message, the identification number of the sending ITS-station was 123, the detection time was the current time obtained from the clock of the computer, the geographical coordinates were (66.5, 25.7) and the cause code for the observed event was 11, meaning “Animals on the road” [16]. As station type, depicting the type of the sending vehicle, we filled 5, which means a passenger car [17].

We carried out the performance measurements by executing the test program 5000 times in a row, each time collecting information about the time required

to sign and verify the message, and finally counting averages of these times. In the time measurements, we used the C command `clock`, which measures the elapsed time in microseconds. The program outputted no text when executing the performance tests, so that this would not affect the time measurements. In addition to this, we modified one version of the program to output information about the size of the message in bytes. The program divided the message into three parts: the signature, the certificate, and other data, and outputted the size of all three parts separately. This allowed us to compare the internal structures of the messages signed with the different algorithms. The size of the certificate is dependent on the size of the public key, which is copied into the certificate. Other data refers to all other data except the certificate and the signature, and in this setting, this portion was always 160 bytes long.

The functions provided by OpenSSL and the quantum-safe reference implementations were designed differently with regard to hashing the message. The quantum-safe signature functions included hashing the message; multivariate-based Rainbow uses OpenSSL implementation of SHA256 hash algorithm [37], whereas lattice-based CRYSTALS-Dilithium and FALCON require a specific type of extendable-output hash function [7,23] for conducting the mathematical operations included in the signing procedure. The OpenSSL functions contained no hashing but presumed that the input to the function would be the hash of the message. Thus, we needed to apply a hash function once before using the OpenSSL sign function. OpenSSL offers SHA256 hash function [34], which the standard approves [19,25], and we used this function when signing with OpenSSL functions. While conducting performance measurements, we counted the time required to hash the message into the signing times of elliptic curves.

## 4 Results

Results of the performance measurements are depicted in Figs. 1 and 2. The fastest algorithm in signing was elliptic curve NIST P-256 (0,06 ms), and the second fastest was Rainbow (0,16 ms). CRYSTALS-Dilithium (0,46 ms) was slightly faster than elliptic curve brainpoolP256r1 (0,55 ms). Evidently the slowest signer was FALCON (4,89 ms).

In verification, the fastest algorithm was Rainbow (0,029 ms), and the second fastest was FALCON (0,042 ms). NIST P-256 (0,078 ms) and CRYSTALS-Dilithium (0,113 ms) performed a bit slower. The slowest verifier was brainpoolP256r1 (0,39 ms). These results show that the times required to verify a message are not as diverse as those in signing; the difference between the slowest and the fastest verifier is 13-fold (Rainbow and brainpoolP256r1), whereas the difference between the slowest and the fastest signer is 81-fold (NIST P-256 and FALCON).

Figure 3 depicts the sizes of the signed messages in bytes. Elliptic curves yielded the smallest signed messages, both resulting to 343 bytes, whereas Rainbow yielded the largest message, 161938 bytes. Messages yielded by the lattice-based alternatives lie somewhere in between; a message signed with CRYSTALS-

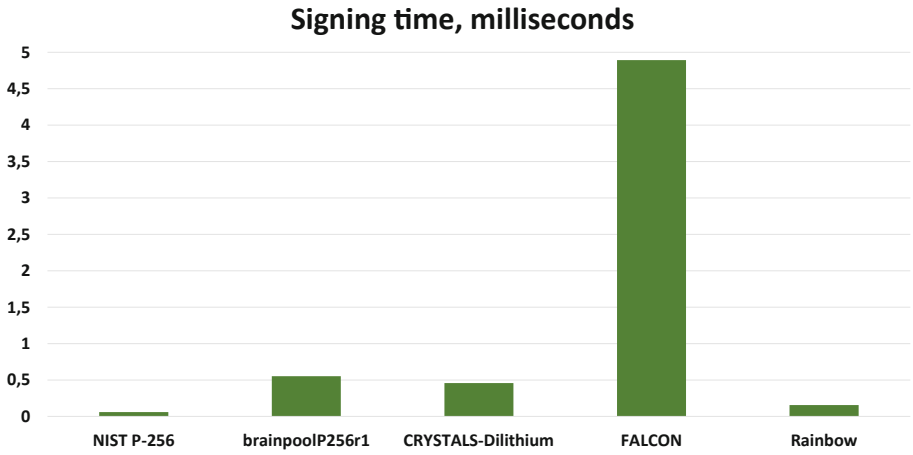


**Fig. 1.** Signing times of the compared digital signature algorithms.

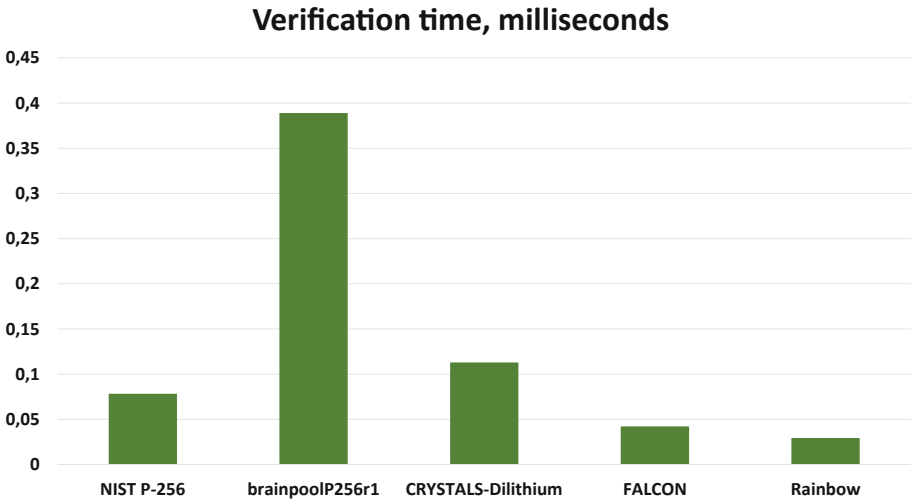


**Fig. 2.** Verification times of the compared digital signature algorithms.

Dilithium was 4004 bytes long and a message signed with FALCON was 1825 bytes long.

Figure 4 depicts the structures of the signed messages. Messages signed with elliptic curves had certificates (112 bytes) and signatures (71 bytes) of the same size, and these together took a bit over half of the whole message. CRYSTALS-Dilithium's signature was 2484 bytes and the certificate 1360 bytes. FALCON's signature was 720 bytes and the certificate 945 bytes long. Rainbow's signature took 130 bytes and the certificate 161648 bytes. In this setting, the fields for certificate issuer and signature of the certificate authority were removed, but as

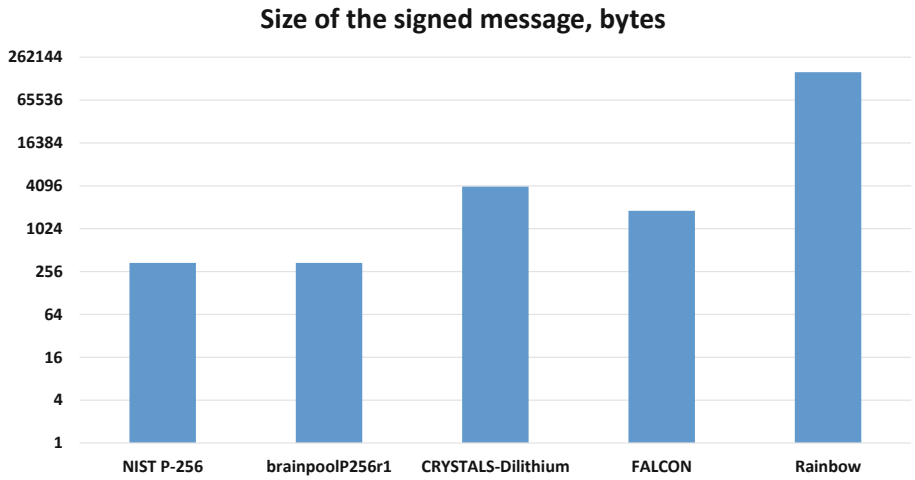


**Fig. 3.** Sizes of the signed messages yielded by the compared digital signature algorithms. Note that the scale is logarithmic.

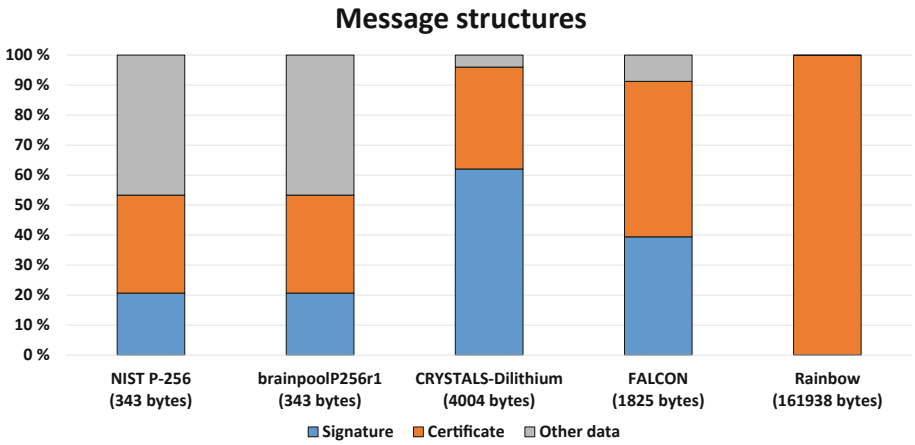


**Fig. 4.** Structures of the signed messages yielded by the compared digital signature algorithms.

mentioned, this has no significant effect on the comparison between the algorithms.

An important note regarding the message sizes is that the quantum-safe reference implementations were designed to make a copy of the original message alongside with signing. This was unnecessary with regard to the technical specifications, which supposed the original message to be elsewhere in the signed message structure. In this case, this message was 64 bytes long (two consecutive 32-byte outputs of SHA256 hash, as defined in the standard [19,25]), and thus

64 bytes can be reduced from the presented quantum-safely signed messages to obtain their length without excessive copies of the hashes. When taking this into account, Rainbow's signature would actually be shorter than that yielded with elliptic curves, only 66 bytes. Other proportions between the messages are not significantly changed by this extra 64 bytes.

## 5 Discussion

As can be seen from the signing times, the two elliptic curves perform quite differently in signing. Thus, migration to quantum-safe algorithms will have a different effect depending on if the original curve in use is NIST P-256, in which case any other algorithm in this comparison will slow down the signing, or brainpoolP256r1, in which case the quantum-safe alternatives can either speed up or slow down the signing. A noteworthy aspect in the verification times is that all quantum-safe algorithms are faster than brainpoolP256r1, and only CRYSTALS-Dilithium is slightly slower than NIST P-256. This implies that migration to the quantum-safe algorithms in C-ITS use will not be a problem with regard to verification performance. The results also show that the fastest algorithm in signing is not necessarily the fastest in verification. Rainbow was the fastest quantum-safe algorithm in both signing and verification. CRYSTALS-Dilithium seems to lie somewhere in between in both signing and verification, whereas FALCON performs notably better in verification than in signing. These factors need to be considered, when choosing the best alternative for C-ITS use. This requires more real world data on how the signatures are used by different participants in the C-ITS framework.

CRYSTALS-Dilithium and FALCON offer a possibility to reduce their key sizes by compression [7,21]. When using compression, only a seed value is stored in memory, and when needed, the actual key is calculated on the grounds of this value. CRYSTALS-Dilithium offers this possibility for its public key and FALCON for its private key. This would save memory space, or with regard to public keys, reduce the size of the certificate. However, compression means more real-time computing, which may not be desirable in real time applications such as many C-ITS use cases.

Of the lattice-based alternatives, FALCON produced shorter signed messages. CRYSTALS-Dilithium was notably faster than FALCON in signing, but slower in verification. Thus, which one of these alternatives would better suit this use, depends on which aspects one chooses emphasize: signing time, verification time or size of the signed message. A single C-ITS station probably executes verification more often than signing, because it can receive messages from several senders around it. On the other hand, the public key infrastructure defined by ETSI includes also certificate authorities, which sign certificates of other users on a regular basis. In this context, also signing time is a considerable feature. As proposed in [41], one possible option is to implement different parts of the public key infrastructure by combining different digital signature algorithms, in this way utilizing the most efficient algorithm for each part of the infrastructure.

FALCON's signing algorithm utilizes complex numbers, which the processor handles as floating-point numbers [23]. In case there is no floating-point unit (FPU) available, FALCON can perform notably slower in signing [35,36]. This effect did not apply in this work, as the processor used in the comparisons included a FPU. However, as also noted in [28,41], this may be a concern with regard to FALCON's portability to different processor architectures. With regard to varying architectures of the commercially available OBU devices, an ideal digital signature algorithm for C-ITS use would suit as many processor architectures as possible. This would refer to CRYSTALS-Dilithium being a safer option in this application domain. In our performance tests, FALCON was compiled with different optimization (-O2) than the other algorithms (-O3). According to our tests, running FALCON with optimization -O3 would not have significantly improved its performance in signing.

Multivariate-based Rainbow performed well in the performance tests, being faster than the lattice-based alternatives. However, because of a large public key, it produced very large messages. In addition, new cryptanalysis results [9] found during the third round in NIST PQC competition have revealed some security issues regarding Rainbow. Because of the questionable security of the multivariate-based schemes, NIST opened a possibility to send new general-purpose digital signature schemes which are not based on structured lattices to the competition [31]. Because of these issues, we do not believe that Rainbow would be an ideal alternative for C-ITS applications in its current form.

The submitters offered also two alternative implementations of Rainbow, Cyclic Rainbow and Compressed Rainbow. Cyclic-version would have decreased the size of the public key by over 50% [38], although it would still have been significantly larger than that of any other algorithm in this comparison. However, if Rainbow would otherwise turn out to be suitable for C-ITS use cases, using Cyclic-version would probably be beneficial.

The main contribution achieved by these results is increased knowledge of how suitable the different quantum-safe digital signature algorithms are for signing notification messages sent in intelligent transport systems. It is important to note that the message structures specified by ETSI are not restricted to notification messages, but also other message types are needed in communication between vehicles. Some of these messages need to be encrypted [19] and in these cases, creating and verifying digital signatures is not enough. Quantum-safe key exchange protocols for implementing these message types need to be evaluated against different criteria. These were out of the scope for this work and are left for future research. However, signed DENM and CAM messages are an important message type in C-ITS communication and thus we think that our performance tests provide relevant information about quantum-safe communication in C-ITS, even though all message types are not covered.

## 6 Future Research

NIST will choose one or more quantum-safe digital signature algorithms to be standardized in a few years. After that, it will be useful to test the winner algo-

rithm with the commercial OBUs available on the market, and possibly optimize the algorithm for this use. For some message types in C-ITS, ETSI's documents define symmetric encryption and encrypting the symmetric keys using elliptic curves [20]. It would be interesting to discover how the quantum-safe KEM algorithms in the NIST standardization contest perform in this use, how often these operations need to be performed and which KEM algorithm would be the best alternative for this purpose.

ETSI's technical specifications also define a complete certificate chain [20], which this work did not. Implementing the certificate chain quantum-safely is also an interesting object for further research. Empirically gathered statistics about how often signatures are created and verified would be beneficial in assessing the suitability of these quantum-safe digital signature algorithms. This statistics should also include signing and verification done by certificate authorities, which behave differently from common vehicles. These data would serve as a basis for choosing one or more optimal algorithms for the different stages of the certificate chain.

## 7 Summary

In the future, communication between vehicles will need to be secured against attacks of quantum computers. Thus, it is beneficial to have information about performance and potential usability issues of different quantum-safe digital signature alternatives with regard to this specific usage.

This paper presents an evaluation of the quantum-safe digital signature algorithms' suitability to be used by intelligent transport systems. We conducted tests based on ETSI's documents, defining a public key infrastructure and digital signing of messages. The current standard accepts elliptic curves for digital signing. However, quantum computers could potentially break these signatures. The test program implemented creating, signing and verification of a notification message, using either elliptic curves or one of the three quantum-safe digital signature algorithms: CRYSTALS-Dilithium, FALCON or Rainbow. We measured the performance of these algorithms in signing and verification, and measured the size of the message signed with the different algorithms.

The results show that the quantum-safe algorithms produce larger signed messages than elliptic curves, but their signing and verification speed is quite competitive. Based on these results, intelligent transport systems could use quantum-safe digital signature algorithms in their communication. Implementing a full certificate chain is one object for future research. To be able to better justify the choice of the digital signature algorithms, gathering statistics about how often messages are signed and verified in traffic would be beneficial. Furthermore, the C-ITS standards specify also other message types requiring different cryptographic operations. The suitable quantum-safe algorithms used with these message types need to be evaluated against different criteria.

**Acknowledgment.** This research was supported by PQC Finland project funded by Business Finland's Digital Trust program.

## References


1. Alagic, G., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD (2019). https://doi.org/10.6028/NIST.IR.8240
2. Arm Architecture: A Foundation for Computing Everywhere. https://www.arm.com/why-arm/architecture/cpu. Accessed 19 Aug 2021
3. Arute, F., et al.: Quantum supremacy using a programmable superconducting processor. Nature **574**, 505–510 (2019)
4. asn1c - ASN.1 Compiler. https://manpages.ubuntu.com/manpages/trusty/man1/asn1c.1.html. Accessed 27 July 2021
5. Aumasson, J.P.: Serious Cryptography: A Practical Introduction to Modern Encryption. No Starch Press, San Fransisco (2018)
6. OBU-301E Specification. https://www.unex.com.tw/sheet/OBU-301E.pdf. Accessed 10 Nov 2021
7. Bai, S., et al.: CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation (Version 3.1) (2021). https://pq-crystals.org/dilithium/data/dilithium-specification-round3-20210208.pdf. Accessed 30 July 2021
8. Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.): Post-Quantum Cryptography. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7d
9. Beullens, W.: Improved cryptanalysis of UOV and Rainbow. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 348–373. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_13
10. C-ITS Secure Communications. https://www.itsstandards.eu/highlighted-projects/c-its-secure-communications/. Accessed 28 July 2021
11. Powerful V2X Onboard Unit. https://www.commsignia.com/products/obu/. Accessed 10 Nov 2021
12. Dilithium. https://pq-crystals.org/dilithium/index.shtml. Accessed 19 July 2021
13. ETSI releases two Technical Reports to support US NIST standards for post-quantum cryptography. https://www.etsi.org/newsroom/news/1981-2021-10-etsi-releases-two-technical-reports-to-support-us-nist-standards-for-post-quantum-cryptography. Accessed 8 Nov 2021
14. Technical Committee (TC) CYBER (Cybersecurity) Activity Report 2020. https://www.etsi.org/committee-activity/activity-report-cyber. Accessed 29 Mar 2022
15. Ettifos On-Board Unit (OBU). https://www.ettifos.com/platforms. Accessed 10 Nov 2021
16. European Telecommunications Standards Institute: ETSI EN 302 637-3 V1.2.2 (2014). https://www.etsi.org/deliver/etsi_en/302600_302699/30263703/01.02.02_60/en_30263703v010202p.pdf
17. European Telecommunications Standards Institute: ETSI TS 102 894-2 V1.3.1 (2018). URL: https://www.etsi.org/deliver/etsi_ts/102800_102899/10289402/01.03.01_60/ts_10289402v010301p.pdfD
18. European Telecommunications Standards Institute: ETSI EN 302 637-2 V1.4.1 (2019). https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.04.01_60/en_30263702v010401p.pdf
19. European Telecommunications Standards Institute: ETSI TS 103 097 V1.4.1 (2020). https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/01.04.01_60/ts_103097v010401p.pdf

20. European Telecommunications Standards Institute: ETSI TS 102 941 V1.4.1 (2021). https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/01.04.01_60/ts_102941v010401p.pdf
21. FALCON - Fast-Fourier Lattice-based Compact Signatures over NTRU. https://falcon-sign.info/. Accessed 19 July 2021
22. Fernandes, B., Rufino, J., Alam, M., Ferreira, J.: Implementation and analysis of IEEE and ETSI security standards for vehicular communications. Mob. Netw. Appl. **23**(3), 469–478 (2018). https://doi.org/10.1007/s11036-018-1019-x
23. Fouque, P.A., et al.: FALCON: Fast-Fourier Lattice-based Compact Signatures over NTRU - Specification v1.2 (2020). https://falcon-sign.info/falcon.pdf. Accessed 19 July 2021
24. Hamida, E.B., Noura, H.N., Znaidi, W.: Security of cooperative intelligent transport systems: standards, threats analysis and cryptographic countermeasures. Electronics **4**, 380–423 (2015)
25. IEEE Vehicular Technology Society: IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages (2016). https://doi.org/10.1109/IEEESTD.2016.7426684
26. IEEE Vehicular Technology Society: IEEE Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages: Amendment 1 (2017). https://doi.org/10.1109/IEEESTD.2017.8065169
27. Overview: Intrinsics for Intel®Advanced Vector Extensions 2 (Intel®AVX2) Instructions. https://software.intel.com/content/www/us/en/develop/documentation/cpp-compiler-developer-guide-and-reference/top/compiler-reference/intrinsics/intrinsics-for-intel-advanced-vector-extensions-2/overview-intrinsics-for-intel-advanced-vector-extensions-2-intel-avx2-instructions.html. Accessed 28 July 2021
28. Marzougui, S., Krämer, J.: Post-quantum cryptography in embedded systems. In: ARES 2019: Proceedings of the 14th International Conference on Availability, Reliability and Security, pp. 1–7. Association for Computing Machinery (2019). https://doi.org/10.1145/3339252.3341475
29. PQC Standardization Process: Third Round Candidate Announcement. https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement. Accessed 28 July 2021
30. Security (Evaluation Criteria). https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/evaluation-criteria/security-(evaluation-criteria). Accessed 28 July 2021
31. Status Update on the 3rd Round. https://csrc.nist.gov/Presentations/2021/status-update-on-the-3rd-round. Accessed 10 Aug 2021
32. ECDSA_SIG_new. https://www.openssl.org/docs/man1.1.1/man3/ECDSA_SIG_get0_r.html. Accessed 6 Aug 2021
33. OpenSSL. https://www.openssl.org/. Accessed 27 July 2021
34. SHA256_Init. https://www.openssl.org/docs/man1.1.1/man3/SHA1.html. Accessed 2 Aug 2021
35. Paul, S., Scheible, P.: Towards post-quantum security for cyber-physical systems: integrating PQC into industrial M2M communication. In: Chen, L., Li, N., Liang, K., Schneider, S. (eds.) ESORICS 2020. LNCS, vol. 12309, pp. 295–316. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59013-0_15
36. Pornin, T.: New Efficient, Constant-Time Implementations of Falcon. Cryptology ePrint Archive, Report 2019/893 (2019). https://ia.cr/2019/893

37. GitHub - fast-crypto-lab/rainbow-submission-round2: Rainbow signature system for Round THREE submission. https://github.com/fast-crypto-lab/rainbow-submission-round2. Accessed 30 July 2021
38. Rainbow Signature. https://www.pqcrainbow.org/. Accessed 19 July 2021
39. Sedar, R., et al.: Standards-compliant multi-protocol on-board unit for the evaluation of connected and automated mobility services in multi-vendor environments. Sensors **21**(6), 2090 (2021). https://doi.org/10.3390/s21062090
40. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**(5), 1484–1509 (1997)
41. Sikeridis, D., Kampanakis, P., Devetsikiotis, M.: Post-Quantum Authentication in TLS 1.3: A Performance Study. International Association for Cryptologic Research (IACR) Cryptology ePrint Archive 2020 (2020)