2021 International Conference on Computer Communications and Networks (ICCCN) | 978-1-6654-1278-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCN52240.2021.9522179

# Performance Characterization of Post-Quantum Digital Certificates

Manohar Raavi , Pranav Chandramouli , Simeon Wuthier , Xiaobo Zhou , and Sang-Yoon Chang

Department of Computer Science, University of Colorado, Colorado Springs, Colorado, 80918, USA
{*mraavi,pchandra,swuthier,xzhou,schang2*}*@uccs.edu*

***Abstract*—Public Key Infrastructure (PKI) generates and distributes digital certificates to provide the root of trust for securing digital networking systems. To continue securing digital networking in the quantum era, PKI should transition to use quantum-resistant cryptographic algorithms. The cryptography community is developing quantum-resistant primitives/algorithms, studying, and analyzing them for cryptanalysis and improvements. National Institute of Standards and Technology (NIST) selected finalist algorithms for the post-quantum digital signature cipher standardization, which are Dilithium, Falcon, and Rainbow. We study and analyze the feasibility and the processing performance of these algorithms in memory/size and time/speed when used for PKI, including the key generation from the PKI end entities (e.g., a HTTPS/TLS server), the signing, and the certificate generation by the certificate authority within the PKI. The transition to post-quantum from the classical ciphers incur changes in the parameters in the PKI, for example, Rainbow I significantly increases the certificate size by 163 times when compared with RSA 3072. Nevertheless, we learn that the current X.509 supports the NIST post-quantum digital signature ciphers and that the ciphers can be modularly adapted for PKI. According to our empirical implementations-based study, the post-quantum ciphers can increase the certificate verification time cost compared to the current classical cipher and therefore the verification overheads require careful considerations when using the post-quantum-cipher-based certificates.**

***Index Terms*—Post-Quantum Cryptography, Quantum-Resistant Cryptography, Digital Certificate, PKI**

## I. Introduction

Public-key infrastructure (PKI) uses digital signature standards like Rivest-Shamir-Adleman (*RSA*) [1] and Elliptic Curve Digital Signature Algorithm (*ECDSA*) [2] to generate the digital certificates. The security of RSA relies on the computational hardness of solving the integer factorization problem and for ECDSA on solving the discrete logarithm problem. Shor's algorithm running on a quantum computer with powerful quantum computational capabilities can solve the integer factorization and discrete logarithm problems in polynomial time [3].

Due to the advancements in quantum computing threatening the mathematical primitives anchoring the security of the classical ciphers, there is an increasing need to transition to quantum-resistant PKI [4]. Cryptography communities are ramping up the efforts to standardize post-quantum primitives [5]. NIST's Post-Quantum Cryptography (PQC) standardization project is in the third round and has selected finalist algorithms to be standardized [6]. Three finalist digital signature algorithms include *Crystals-Dilithium (Dilithium)*, *Falcon*, and *Rainbow*. Dilithium and Falcon are lattice-based schemes while Rainbow is a multivariate-based scheme. NIST expects to standardize either Dilithium or Falcon while Rainbow is selected for diversity and future standardization considerations. The post-quantum digital signature ciphers, including Dilithium, Falcon, and Rainbow, are undergoing active research in analyzing their security and performance, and our research is motivated to inform these emerging cipher designs and applications.

PKI is critical in digital networking systems since it provides trust in the public key, which can be used for security protocols or for establishing symmetric keys. Because of its importance for digital and Internet networking, we study the effects of transitioning to the post-quantum algorithms in PKI and how it affects the digital networking systems' performances. While building on recent research literature studying post-quantum ciphers (Section VII), we distinguish our work by specifically focusing on the PKI integration of the post-quantum ciphers and conducting an empirical implementations-based study for our analyses of the NIST post-quantum ciphers. More specifically, we study and analyze the feasibility and the performances of the NIST PQC standardization project's finalist digital signature algorithms when integrated into PKI. Our work focuses on computational efficiency for the following reasons. First, transitioning to post-quantum PKI implementations affect computing the most. Second, we generalize the computational overheads caused by the post-quantum algorithms across the PKI architecture and implementation scenarios. We provide more information about the PKI architecture and implementation scenarios in Section II to better define the scope of our digital signature application for PKI and certificates.

**Contributions** We conduct an implement-based empirical study of the post-quantum ciphers on a classical computer, as the ciphers are designed for classical computers but are resistant against quantum attackers. We first test the feasibility of the PQC ciphers for X.509 PKI, including the Rainbow I classic cipher increasing the public-key and the certificate size by 411 and 163 times, respectively, when compared to that of RSA 3072. We measure and analyze the implementation performance of the NIST PQC standardization project's finalist digital signature algorithms when executing the PKI functions including Key-Pair Generation, Certificate Signing Request

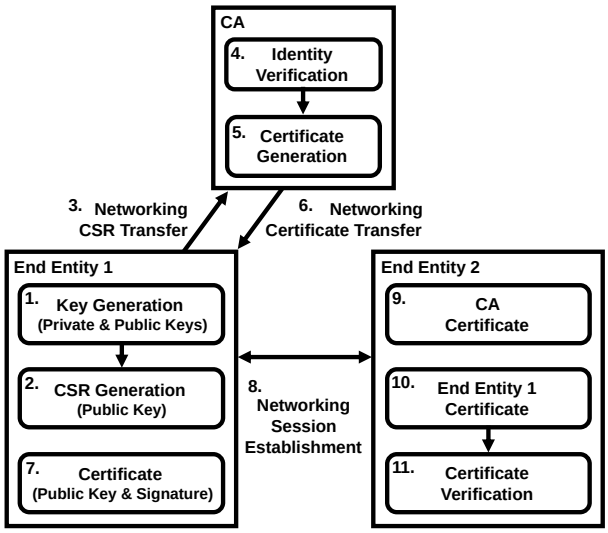



Fig. 1: PKI process focusing on the networking/computing between the end entity (the beneficiary of the PKI service) and the CA (PKI service provider). The PKI backend, such as that for registration, is not included in this diagram.

(CSR) Generation, Certificate Generation, and Certificate Verification. We analyze the results in comparison with the classical (RSA and ECDSA) and the hybrid digital signature algorithms combining a classical and a PQC algorithm for defense by depth. Furthermore, we analyze the storage requirements for storing CSR, and End-Entity certificate of PKI using the post-quantum and hybrid algorithms.

**Organization** The rest of the paper is organized as follows: Section II reviews the background of PKI and introduces the NIST PQC standardization project's finalist digital signature algorithms. Section III describes our approaches on computing and memory focus, PQC ciphers, security levels, and experimental setup while Section IV dives into the feasibility of PQC ciphers in the X.509 standard and analyzes the memory overheads. Section V analyzes the results and Section VI summarizes the implications of our results, takeaways, and application discussions. Section VII discusses related work in post-quantum PKI and Section VIII concludes our work.

## II. Background of PKI and PQC Ciphers

This section presents the overview of PKI, introduces the NIST PQC standardization project, and the finalist digital signature algorithms.

### A. Background of PKI

Public-key infrastructure (PKI) uses certificate profiles to authenticate end devices [7]. X.509 certificate format based on hierarchical design is widely in use compared to the OpenPGP certificates [8] based on the web of trust. Organization's security requirements and trust models in place help in designing different PKI architectures [9], [10]. Transitioning to post-quantum PKI involves the use of post-quantum digital signature algorithms to generate post-quantum certificates, which affect mostly the intra-host computing latency independent of the variations in the PKI architecture and model.

Figure 1 shows the PKI certification process. In Step 1, the Key-Pair Generation for the public-private key pair is done locally for the subject end entity, End-Entity 1. The private key does not leave the local node, while the public key gets shared with the others including the CA for the certification and later to the End-Entity 2 for communicating data. In Step 2, End-Entity 1 creates a Certificate Signing Request (CSR) using the public key and sends it to the CA in Step 3. In Step 4, the CA verifies the identity of End-Entity 1 and generates a certificate in Step 5. This certificate contains the public key of the End-Entity 1 and a signature generated by the CA using its private key. In Step 6, the certificate is transferred to End-Entity 1. In Step 7, End-Entity 1 verifies and configures the certificate for future communications. In Step 8, the certificate is transferred from End-Entity 1 to End-Entity 2 for authentication. The End-Entity 2 gets the CA and End-Entity 1 certificates in Steps 9 and 10 respectively for verification. In Step 11, End-Entity 2 verifies the certificate of End-Entity 1. This verification includes the use of CA's public key from the CA certificate to verify the signature on the End-Entity 1 certificate. Once verification is successful, a session is established between End-Entity 1 and End-Entity 2 for communicating data.

To put the PKI process into more concrete context and better explain the terms in Figure 1, we provide an example in the Internet and web security and more specifically for securing and providing the root of trust for the web servers, which is one of the popular applications of PKI. When a user laptop accesses google.com, End-Entity 1 is google.com while End-Entity 2 is the user/laptop. However, before the user accesses google.com (Step 8 in Figure 1), google.com interacts with the CA, such as Google Trust Services (GTS) CA 101, for Steps 1-7 in Figure 1. The interactions and networking between End-Entity 1 (the subject of the digital certificate and google.com in this example) and the CA (GTS CA 101) being offline, i.e., before google.com and the user/laptop communicates, is a strength of the CA. The networking between End Entity 1 and End Entity 2 does not rely on real-time CA availability. For example, if the CA-End-Entity-1 transaction were done online or on-the-fly and the centralized CA were busy, then the networking between End-Entity 1 and End-Entity 2 would also need to wait. The PKI process also helps with the scalability, since the google.com's certificate signed by GTS CA 101 can be shared with the user/laptop as well as any other End Entities wishing to access google.com. The certification verification (Step 11 in Figure 1) leverages the End-Entity 2's a priori trust with the CA, for example, given a list of trusted CA's, End-Entity 2 checks that the End-Entity 1's certificate is signed by a CA in that list.

The functions described in this section and shown in Figure 1 are the essential functions for the PKI process. In addition to its popular applications to provide the root of trust for authenticating the web servers for the more conventional internet applications, PKI and digital certificate management are applied for ad hoc networking where the data networking

TABLE I: Security Level 1 or 2 Algorithms

| Algorithm Type | NIST Security Level 1 or 2 [5] |
|---|---|
| Classical | RSA 3072 and P256 (ECDSA *secp256k1*). |
| PQC-Dilithium | Dilithium2 and Dilithium2+AES. |
| PQC-Falcon | Falcon 512 |
| PQC-Rainbow | Rainbow I Classic, Rainbow I Circumzenithal, and Rainbow I Compressed. |
| Hybrid with RSA | RSA3072+Dilithium2, RSA3072+Falcon512, RSA3072+Rainbow I Classic, RSA3072+Rainbow I Circumzenithal, and RSA3072+Rainbow I Compressed. |
| Hybrid with ECC | P256+Dilithium2, P256+Falcon512, P256+Rainbow I Classic, P256+Rainbow I Circumzenithal, and P256+Rainbow I Compressed. |
| **Security Level 1** equivalent to performing key search attack on AES192 **Level 2** equivalent to performing collision search attack on SHA256 | |

TABLE II: Security Level 3 Algorithms.

| Algorithm Type | NIST Security Level 3 [5] |
|---|---|
| Classical | P384 (ECDSA *secp384r1*) |
| PQC-Dilithium | Dilithium3 and Dilithium3+AES. |
| PQC-Falcon | N/A |
| PQC-Rainbow | Rainbow III Classic, Rainbow III Circumzenithal, and Rainbow III Compressed. |
| Hybrid with RSA | N/A |
| Hybrid with ECC | P384+Dilithium3, P384+Rainbow III Classic, P384+Rainbow Circumzenithal, and P384+Rainbow Compressed |
| **Security Level 3** equivalent to performing key search attack on AES192 | |

between the end entities do not go through the traditional internet core domain networking, e.g., Security Credential Management System (SCMS) for vehicular/V2X/VANET networking [11], [12], [13]. Our work is applicable in such ad hoc networking contexts in principle because the PKI designs for such applications are often more complex due to the requirements on the end entities and the additional security/privacy objectives, which build on the PKI functions targeted for our analyses. However, in this research, we focus on the NIST PQC Finalist algorithms designed for general digital networking, described in greater detail in Section II-B. Targeting embedded or sensor/IoT nodes and exploring the NIST Alternate PQC ciphers (NIST Alternate is a different class from the NIST Finalist for longer-term cipher analyses and targeted applications) are beyond the scope of this work.

**Focus of Our Work** Our work mainly focuses on computing (as opposed to the networking transmission) and more specifically on Step 1 Key-Pair Generation, Step 2 CSR Generation, Step 5 Certificate Generation, and Step 11 Certificate Verification in Figure 1. As discussed in Section III, transitioning to post-quantum digital signature algorithms affect these steps while having a significantly smaller effect in the other steps for the networking transmissions; for example, the packet transmission delay in the networking channel has minimal changes depending on the cipher algorithms.

### *B. Post-Quantum Digital Signature Ciphers*

NIST is currently in the third round of the standardization process and announced three finalist digital signature algorithm candidates. Two of the three finalist algorithms Dilithium and Falcon are based on lattice-based cryptography and the third algorithm is Rainbow based on the multivariate signature scheme. Lattice-based cryptography has strong security proofs that include NP-hard problems [14] such as Short Vector Problems (SVP), Closed Vector Problems (CVP), and Short Independent Vector Problems (SIVP) [15].

TABLE III: Security Level 5 Algorithms.

| Algorithm Type | NIST Security Level 5 [5] |
|---|---|
| Classical | P521 (ECDSA *secp521r1*) |
| PQC-Dilithium | Dilithium5 and Dilithium5+AES. |
| PQC-Falcon | Falcon 1024 |
| PQC-Rainbow | Rainbow V Classic, Rainbow V Circumzenithal, and Rainbow V Compressed. |
| Hybrid with RSA | N/A |
| Hybrid with ECC | P521+Dilithium5, P521+Falcon1024, P521+Rainbow V Classic, P521+Rainbow V Circumzenithal, and P521+Rainbow V Compressed. |
| **Security Level 5** equivalent to performing key search attack on AES256 | |

**Dilithium** Dilithium introduces Dilithium 2, Dilithium 3, and Dilithium 5 which correspond to NIST levels 2, 3, and 5 of the post-quantum security categories, respectively [6]. The design of Dilithium is based on the "Fiat-Shamir with Aborts" approach, SHAKE for its hashing algorithm and other security schemes in lattice-based cryptography [16]. For the NIST round 3 submissions, Dilithium also introduced Dilithium 2, Dilithium 3, and Dilithium 5 versions that use AES rather than SHAKE to expand the public key matrix and to show the efficiency of their algorithm as current hardware is more optimized for AES [17].

**Falcon** Falcon stands for the acronym, Fast Fourier lattice-based compact signature scheme over a N-th Degree Truncated Polynomial Ring (NTRU). It introduces Falcon 512 and Falcon 1024 corresponding to NIST levels 1 and 5 of the post-quantum security categories, respectively. Falcon's security scheme is based on Gentry, Peikert and Vaikuntanathan (GPV), NTRU lattices, Fast Fourier sampling, and other security schemes in lattice-based cryptography [18].

Multivariate-based cryptography has security proofs that include solving multivariate quadratic equations, multivariate polynomials over finite fields which is an NP-hard problem.

**Rainbow** Rainbow is based on an Oil-Vinegar signature scheme [19]. Rainbow introduces three families: Rainbow I, Rainbow III, and Rainbow V which correspond to NIST levels 1, 3, and 5 of the post-quantum security categories, respectively. Each family also has multiple variants: classic, circumzenithal, and compressed [20]. Rainbow circumzenithal computes the central Rainbow map from the public key and Rainbow Compressed key technique generates the central key map during signature generation reducing the private key size but increasing the execution time .

## III. Our Approach and Analyses Methodology

We empirically study the feasibility and the performance of the PQC ciphers when used for PKI and digital certificate management. In this section, we discuss the scope of our

analyses focusing on the computing functions of PKI and the NIST PQC finalist ciphers as well as the hybrid ciphers.

**Computing and Memory Focus** Our analyses focus on the computing and the memory/size overhead incurred by the post-quantum algorithms as opposed to networking. The transition to the post-quantum PKI for the post-quantum era affects the computing more than the steps for transmitting/delivering the packets, i.e., the transmission delay in the networking channel. More specifically, these correspond to the Key-Pair Generation (Step 1 in Figure 1), CSR Generation (Step 2 in Figure 1), Certificate Generation (Step 5 in Figure 1), and Certificate Verification (including Step 11 in Figure 1).

**PQC Ciphers: Classical vs. PQC vs. Hybrid** We compare the performance of the post-quantum algorithms (the NIST Finalist digital signature cipher algorithms), the classical algorithms (RSA and ECDSA), and the hybrid-scheme combining both the post-quantum and classical. Our classification targets to help understand the computational overheads introduced by selecting a post-quantum or a hybrid algorithm. The hybrid scheme is proposed for security by depth so that there is an additional layer of protection even if highly effective cryptanalysis is discovered and threatens the post-quantum ciphers [21], as cryptanalysis of the ciphers is currently in active research. Tables I-III list the post-quantum and hybrid-algorithms we used in our experimentation and analyses. The first column lists the algorithm family or type and the next column lists the respective security level algorithm.

**PQC Ciphers: Security Levels** Algorithms are listed based on the claimed security levels. Each of these security levels indicates the minimum required computational resources to perform a successful brute-force key search attack on a symmetric block cipher or performing a successful brute-force collision attack on a hash function [5]. Tables I-III also provide additional details on symmetric block ciphers and hash functions used as references. We select the possible hybrid algorithm combinations from *OpenSSL* (*OpenSSL 1.1.1*) implementation. It supports RSA 3072 and P256 (ECDSA with a 256-bit key and also known as *secp256k1*) for Level 1 hybrid combinations, P384 (ECDSA with a 384-bit key and also known as *secp384r1*) for Level 3 hybrid combinations, and P521 (ECDSA with 521-bit key and also known as *secp521r1*) for Level 5 hybrid combinations. For example, to create a level 5 secure hybrid algorithm, we used one classical algorithm and one post-quantum algorithm when considered individually treated as level 5 secure algorithms. We used all possible hybrid combinations in our analyses and listed them according to their security levels in Table I-III. Our comparison analyses are based on the security levels, i.e., we compare the cipher options between those providing equal security levels.

**Implementation** We implement the ciphers from *liboqs* and *OpenSSL* (*OpenSSL 1.1.1*) by *Open Quantum Safe* [22]. The libraries are deployed on a machine with four processing cores and 6 GBs of RAM on a computer equipped with an AMD Ryzen 7 1700x 8-core 16-thread processor at 3.4 GHz base processor frequency, and 32 GBs of RAM. Our certificates are based on X.509 [7], and we use TLS and TCP/IP to transfer certificates and CSR's. We run the experiment 1,000 times each for Key-Pair Generation, CSR Generation, Certificate Generation, and Certificate Verification.

**Measurements** We study and measure the post-quantum cipher incorporations to the PKI protocol steps, which include both the cipher algorithms and the PKI functions. Measuring the time cost overheads of the PKI functions separately (while disabling the ciphers) yield that the time overheads for Key-Pair Generation, CSR Generation, Certificate Generation, and Certificate Verification are 2.79ms, 0.92ms, 0.94ms, and 2.76ms, respectively. Our performance measurements in Section IV include these overheads excluding the ciphers. We omit the performance analyses focusing on the cipher algorithms themselves and refer to Section VII and our previous work [23] for those work which studied and analyzed only the cipher algorithms.

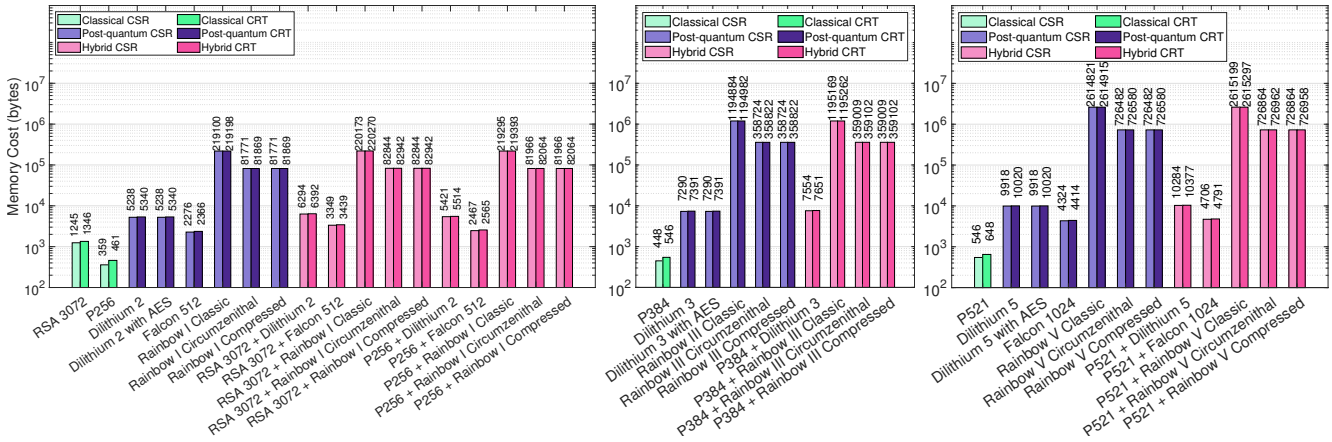
**Generality of Results and Analyses Focus on Level 5** Our experiments consist of 14 algorithms in security level 1, 4 algorithms from security level 2, 10 algorithms from security level 3, and 12 algorithms from security level 5. Since the cross-cipher comparisons within the security levels show similar behavior, we generally focus our analyses on the Level 5 ciphers with the highest security while showing all the experimental results including those for security levels 1, 2, and 3 ciphers.

## IV. Feasibility and Memory Size Analyses

In this section, we discuss the feasibility of post-quantum and hybrid algorithms in the X.509 standard. We also provide the memory size overheads caused by post-quantum and hybrid algorithms in comparison with the classical algorithms in respective security levels.

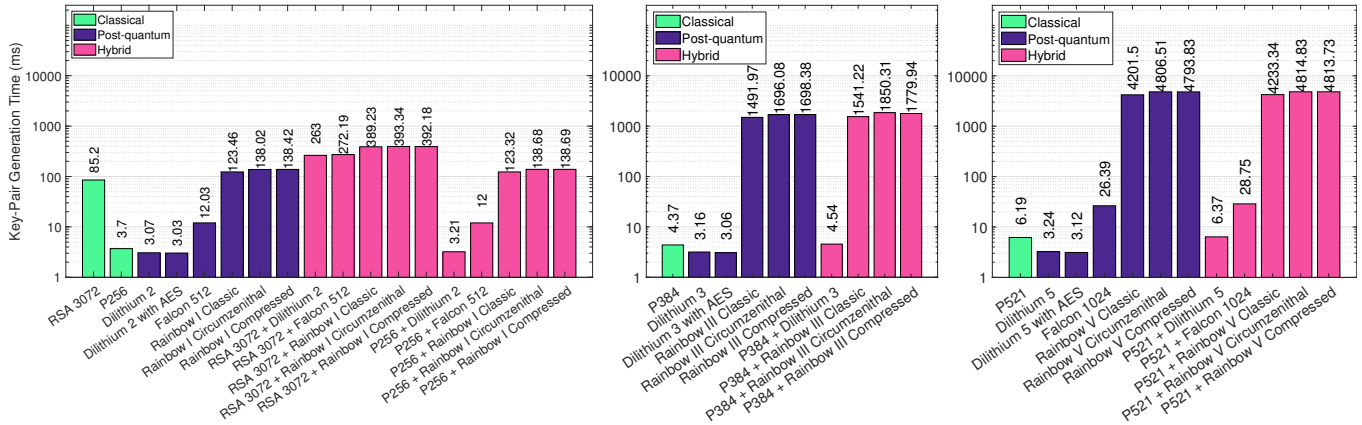
### A. Feasibility

While the application-layer cryptographic ciphers can be developed and applied separately to the X.509 PKI in principle, we test the feasibility of the PQC ciphers because of the parameter changes incurred by the PQC ciphers. More specifically, the Rainbow incurring greater sizes in public keys causes bigger certificates. For example, Rainbow I classic produces public keys 411 times bigger than RSA 3072 causing the certificates 163 times bigger compared to RSA 3072. Because there is no specific limitation to the size of the X.509 certificate standard, all of the ciphers are supported. However, *OpenSSL* implementation imposes restrictions on the X.509 certificate size causing issues in certificate generation for a few of the NIST PQC Standardization project's finalist digital signature algorithms [21]. By default, only two out of the nine Rainbow variant algorithms are enabled. *OpenSSL* provides a template to enable the additional algorithms and limits the number of active algorithms at any time to 64 [22]. We make changes to the *oqs-template/generate.yml* file and enable the remaining seven algorithms by setting the *enable* value to *true* from *false*. We recompile to generate certificates using all post-quantum and hybrid algorithms.




(a) Security levels 1 and 2 (b) Security level 3 (c) Security levels 5

Fig. 2: The file size comparison of Certificate Signing Request (CSR) and end-entity Certificate ("CRT" in the figure) for Classical, Post-Quantum, and Hybrid algorithms, respective to their security levels. Vertical axis are aligned for sub-figures.




(a) Security levels 1 and 2 (b) Security level 3 (c) Security levels 5

Fig. 3: Key-Pair Generation Time for Classical, Post-Quantum, and Hybrid-Algorithms.

### *B. File Size Comparison*

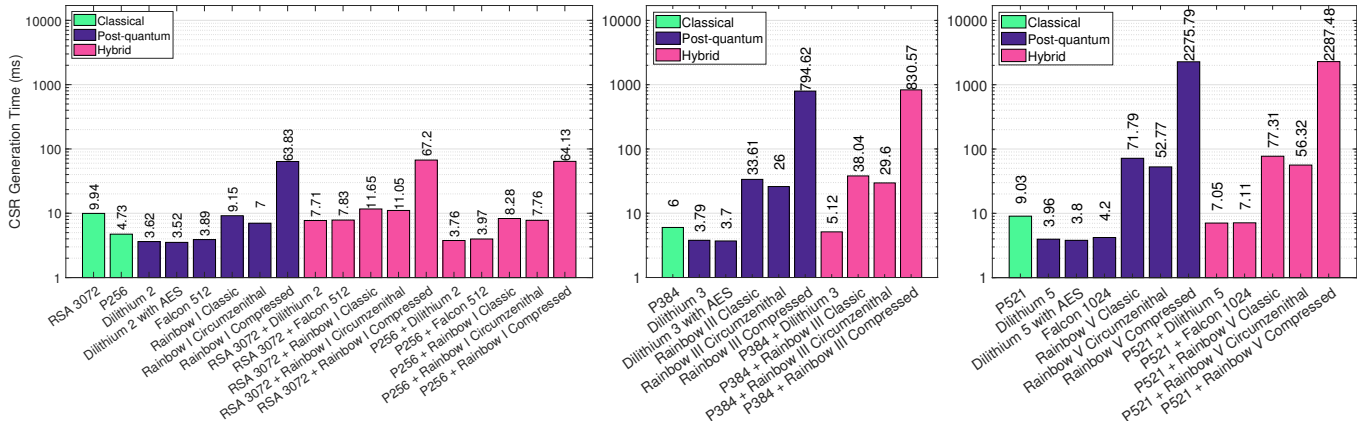
Figure 2 plots the file sizes for CSR and End-Entity certificate generated by the classical, post-quantum, and hybrid algorithms listed in Tables I-III. Figures 2a, 2b, and 2c separate the file sizes based on the security levels 1, 3, and 5 respectively. ECDSA specific P256, P384, and P521 have the lowest CSR and Certificate sizes compared to any other algorithms at their respective security levels. As mentioned in Section III, we focus our analysis on security level 5 algorithms. From Figure 2c, Transitioning to post-quantum and hybrid algorithms increase the minimum certificate size by 6.81 times using Falcon 1024 and 7.39 times using P521+Falcon 1024 at security level 5. Dilithium 5 certificate is 15.46 times and Rainbow V classic certificate is 4035.36 times bigger than the ECDSA P521 certificate size. In post-quantum algorithms, Dilithium 5 and Rainbow V classic certificates are 2.27 and 592.41 times bigger than Falcon 1024 certificate size respectively. P521+Dilithium 5 certificate is only 0.03 times bigger than Dilithium 5 certificate and p521+Falcon 1024 certificate is only 0.08 times bigger than Falcon 1024.

For memory-constrained devices, Falcon512/Dilithium 3/Falcon 1024 are suitable post-quantum algorithms at security levels 1/3/5 due to low memory overheads. In the case of hybrid algorithms, P256+Falcon 512/P384+Dilithium 3/P521+Falcon 1024 are suitable options. The followings are the key observations from our analyses:

- Falcon post-quantum and hybrid variants produce certificates of the lowest sizes at security levels 1 and 5 compared to any other post-quantum and hybrid algorithms.
- At security level 3 Falcon doesn't have a proposed algorithm and Dilithium variants turn out to be effective.
- Due to their huge public key sizes, Rainbow variants produce the biggest CSR and certificates causing the highest memory overheads across all the security levels.

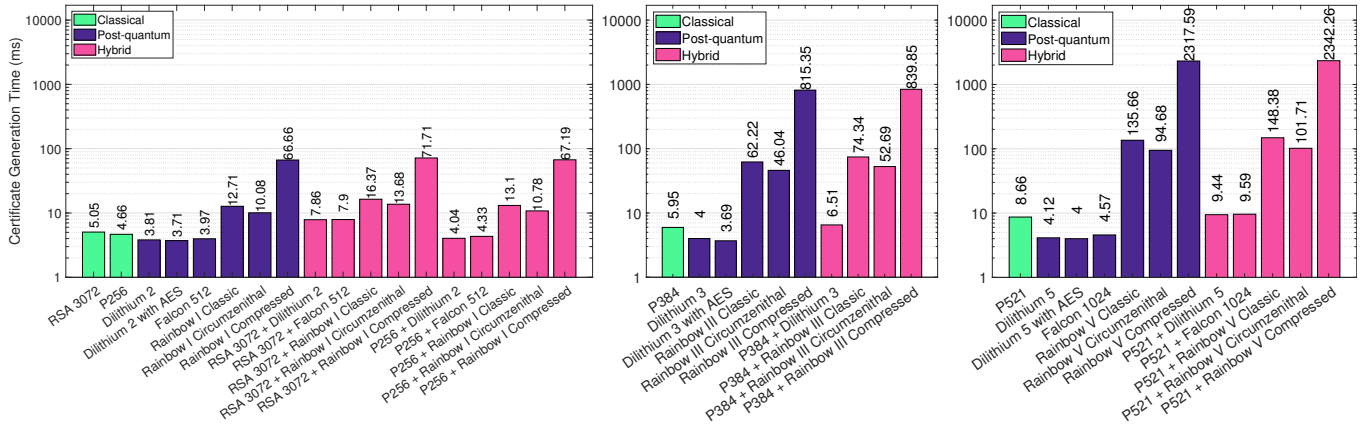
## V. Computational Performance Analyses for Post-Quantum PKI

In this section, we present the results of our implementation measuring the computational overheads caused by the clas-




(a) Security levels 1 and 2 (b) Security level 3 (c) Security levels 5

Fig. 4: CSR Generation Time for Classical, Post-Quantum, and Hybrid-Algorithms.




(a) Security levels 1 and 2 (b) Security level 3 (c) Security levels 5

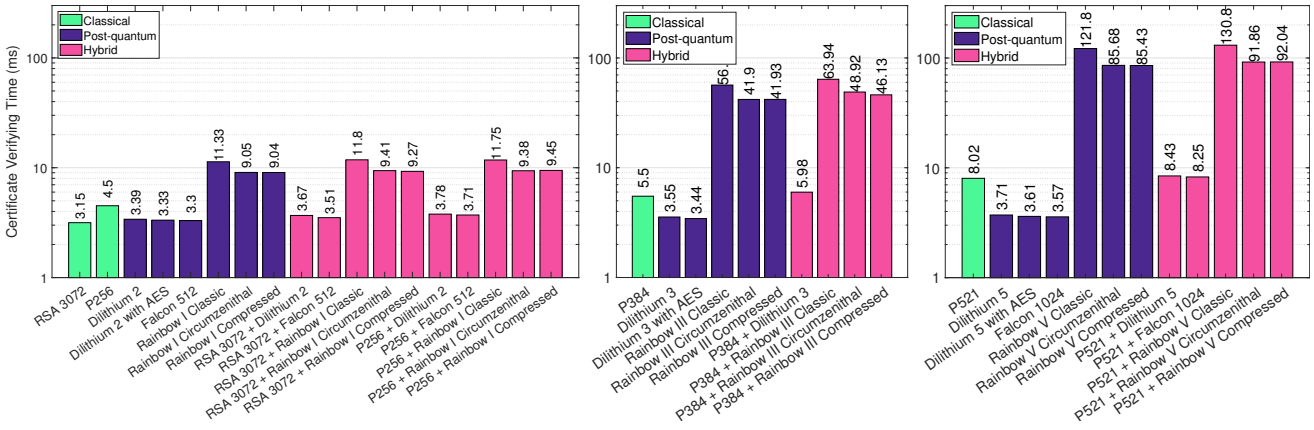
Fig. 5: Certificate Generation Time for Classical, Post-Quantum, and Hybrid-Algorithms.

sical, post-quantum, and hybrid digital signature algorithms. We build on the approach and procedures described in Section III, measure and analyze the time costs for Key-Pair Generation (Section V-A), CSR Generation (Section V-B), Certificate Generation (Section V-C), and Certificate Verification (Section V-D) for classical, post-quantum, and hybrid digital signature algorithms listed in Tables I-III.

### *A. Key-Pair Generation at End Entity*

Figure 3 plots the average time taken for Key-Pair Generation in classical, post-quantum, and hybrid algorithms, separated by their security levels. We find that the behavior in terms of time taken stays consistent with respect to algorithm type when varying security levels. For example, we find that the Key-Pair Generation time for ECDSA algorithms is consistently slower than RSA, Falcon, and Rainbow, while Dilithium remains faster than ECDSA across every security level. From Figure 3c, Dilithium 5 and Dilithium 5 with AES are 47.65% and 49.6% faster in Key-Pair Generation than P521. Falcon 1024 is 326% slower in Key-Pair Generation than P521. The Key-Pair generation times for all Rainbow algorithms are at least 20 times slower than the non-Rainbow algorithms. When comparing Rainbow V with the P521 hybrid versions of Rainbow V, there is a time difference of 31.84ms, 8.32ms, and 19.9ms for Classic, Circumzenithal, and Compressed, respectively. While the Rainbow V hybrid algorithms are slower than post-quantum Rainbow V algorithms, the Key-Pair Generation time differences are all under 32ms. Dilithium 5 and Falcon 1024 hybrids using P521 are only 3.13ms and 2.37ms slower compared to Dilithium 5 and Falcon 1024 respectively. We find that the hybrid cases bring low additional overheads, relative to the post-quantum counterparts. The followings are the key observations:

- Dilithium variants are the fastest post-quantum and hybrid algorithms in Key-pair Generation across all security levels, with the Dilithium AES variant being the fastest.
- Transitioning to Dilithium 2 and Falcon 512 from RSA 3072 helps in reducing the Key-Pair Generation times by 28.11 and 7.08 times.
- Rainbow algorithms and their hybrid combinations are slowest in Key-Pair Generation across all security levels. Rainbow is at least 1, 2, and 3 orders of magnitude




(a) Security levels 1 and 2 (b) Security level 3 (c) Security levels 5

Fig. 6: Certificate Verification Time for Classical, Post-Quantum, and Hybrid-Algorithms.

slower than Dilithium, in security levels 1 and 2, 3, and 5 respectively.

### *B. CSR Generation at End Entity*

The Certificate Signing Request (CSR) generation is a critical step in certificate generation at the end entity, as it enables using the public key and other identification information for the generation of a certificate. Figure 4 plots the average time taken for CSR Generation in classical, post-quantum, and hybrid algorithms separated by their security levels. From Figure 4c, Dilithium 5 with AES and Falcon 1024 reduces the CSR Generation times by 2.37 and 2.15 than P521. Rainbow variants CSR Generations time costs are larger by a minimum of 7.95 times and a maximum of 252.02 times than P521. The followings are the key observations:

- Dilithium variants are the fastest post-quantum and hybrid algorithms in CSR Generation times across all the security levels.
- Post-Quantum and hybrid variants of Dilithium and Falcon outperform the classical algorithms in CSR Generation times at all security levels.
- Rainbow variants are the slowest across all security levels. Rainbow Compressed is 1 order of magnitude slower than Dilithium in security levels 1 and 2, and is 2 orders of magnitude slower in security levels 3 and 5.

### *C. Certificate Generation at CA*

Figure 5 plots the average time taken for Certificate Generation in classical, post-quantum, and hybrid algorithms separated by their security levels. From Figure 5c, Dilithium 5 with AES and Falcon 1024 reduces the Certificate Generation costs by 53.81% and 47.22% than P521 respectively. Dilithium 5 is 10.9% and Dilithium 5 with AES is 14.25% faster than Falcon 1024 in Certificate Generation. P521+Dilithium 5 and P521+Falcon 1024 show similar Certificate Generation times, with Dilithium variant 0.15ms (1.64%) faster. P521+Dilithium 5 and P521+Falcon 1024 are 43.7% and 47.63% slower than their non-hybrid counterparts Dilithium 5 and Falcon 1024, respectively. The followings are the key observations:

- Post-quantum Dilithium and Falcon outperform classical algorithms in Certificate Generation at all security levels.
- Dilithium and Falcon hybrid variants with P521 are competitive and the fastest hybrid algorithms in Certificate Generation time costs at security level 5.
- Rainbow algorithms take longer times for Certificate Generation at all security levels. For example, at security level 5, Rainbow variants take 993% longer than P521.

### *D. Certificate Verification at All Interested Nodes*

Certificate verification is an important part of the analysis, as it has a high impact on the transition towards PQC. While Sections V-A, V-B, and V-C handle the generation components of PKI, certificate verification is executed every time that a certificate needs to be used. Thus, for many applications, minimizing the certificate verification time cost greatly reduces the computational overheads.

Figure 6 plots the average times taken for Certificate Verification in classical, post-quantum, and hybrid algorithms separated by their security levels. From Figure 6c, Transitioning to Dilithium 5, Dilithium 5 with AES, and Falcon 1024 from P521 reduces the Certification Verification costs by 53.74%, 54.98%, and 55.48%, respectively. Dilithium 5 takes 3.92% and Dilithium 5 with AES takes 1.12% longer in Certificate Verification than Falcon 1024. P521+Dilithium 5 takes 131.09% longer than Dilithium 5 and P521+Falcon 1024 takes 127.22% longer than Falcon 1024 in Certification Verification times. Rainbow V variants take a minimum of 2292.99% longer in Certificate Verification costs than Falcon 1024. The followings are the key observations:

- At security level 1, RSA 3072 has the lowest Certificate Verification costs. Dilithium 2 and Falcon 512 incur 7.61% and 4.76% longer in Certificate Verification costs than RSA 3072.
- In post-quantum algorithms, Falcon 512 and Falcon 1024 have the lowest Certificate Verification costs at security

| Algorithm | $A_1$ | $A_2$ | $A_3$ | $A_4$ | $A_5$ | $A_6$ | $A_7$ | $A_8$ | $A_9$ | $A_{10}$ | $A_{11}$ | $A_{12}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Generated (kBps) | 301651 | 289684 | 89762 | 78552 | 604.9 | 2317 | 35.41 | 76984 | 75742 | 69.93 | 75.22 | 2.05 |
| Verified (kBps) | 326041 | 321539 | 99424 | 100381 | 673.7 | 2561 | 960.6 | 86170 | 88049 | 79.33 | 83.28 | 52.05 |

TABLE IV: Certificate generation and verification in kilobytes per second (kBps) for Level 5 algorithms $A_1$ through $A_{12}$, as arranged in Figure 6c. $A_1$: P521, $A_2$: Dilithium 5, $A_3$: Dilithium 5 with AES, $A_4$: Falcon 1024, $A_5$: Rainbow V Classic, $A_6$: Rainbow V Circumzenithal, $A_7$: Rainbow V Compressed, $A_8$: P521 + Dilithium 5, $A_9$: P521 + Falcon 1024, $A_10$: P521 + Rainbow V Classic, $A_11$: P521 + Rainbow V Circumzenithal, $A_12$: P521 + Rainbow V Compressed.

levels 1 and 5. Even in the case of hybrid algorithms, Falcon variants have the lowest Certification Verification costs.
- Post-Quantum Dilithium and Falcon variant algorithms outperform ECDSA algorithms in Certificate Verification costs at all security levels.
- Rainbow variants take longer time than any other algorithms at respective security levels.

In addition to the time cost, we consider the rate at which certificates are generated and verified. This is done to provide insight into how long the algorithms spend processing each byte. Table IV compares each algorithm at security level 5 in terms of bytes per millisecond, where a larger number means that the algorithm spends less time on each byte. We find that while Rainbow V Compressed is slow in certificate generation, with its P521 variant being the slowest, Rainbow V Compressed and its P521 hybrid are 27.13 and 25.39 times faster in verification, respectfully. The followings are the key observations:

- RSA 3072, Dilithium, and Falcon verify certificates fastest and are faster than ECDSA at every security level.
- Rainbow Classic verifies faster than all other Rainbow variants.
- Rainbow V Compressed has the greatest increase in verified bytes per second when comparing against generated bytes per second.

## VI. Takeaways and Application Discussions

In this section, we discuss the key takeaways from our analyses of the feasibility and the memory performances (Section IV) and of the computational performances (Section V). We also discuss the application implications of our results.

The post-quantum ciphers as well as the hybrid ciphers are feasible and supported by X.509 certificates (Section IV-A). Transitioning to a post-quantum or hybrid algorithm introduces significant memory overheads, and the Falcon family of post-quantum and hybrid algorithms are recommended for implementations prioritizing storage overheads (Section IV-B).

Post-quantum PKI introduces varying computational performances in processing time overhead depending on the cipher selection (Section V). Dilithium algorithms have the lowest Key-Pair Generation time costs, CSR generation time costs, and Certificate Generation time costs compared to any other classical, post-quantum, or hybrid algorithms at their respective security levels (Sections V-A-V-C). Similarly, Falcon variant algorithms have the lowest Certification Verification time costs compared to any other post-quantum or hybrid algorithms (Section V-D). When comparing hybrid algorithms using one classical algorithm from the ECDSA family (P256 and P384) and one post-quantum algorithm from Dilithium/Falcon families, there are similar costs as post-quantum algorithms. We also find that Rainbow variants incur heavier computational costs compared to any algorithm across all functions and all security levels.

Due to the varying performances depending on the post-quantum ciphers, we make different recommendations depending on the application requirements. Dilithium family of algorithms are recommended for time-sensitive implementations for their competitive performance across all functions. For example, web-server/client scenarios requiring effective page load times can take advantage of faster execution times of Dilithium family of algorithms. Falcon family of algorithms are the most efficient post-quantum/hybrid algorithms in applications where Certificate Verification is critical. For example, applications such as blockchain rely on certificate verification of incoming transactions for every relay node as the networking when broadcasting, whereas the certificate generation is only executed when a transaction is created. In such situations, verification is a higher priority than a signature generation. For applications seeking to be robust against the dynamic cryptanalysis research and be conservative in security, the hybrid schemes are better alternatives than post-quantum algorithms. P256-Dilithium 2 incurs low computational overhead and is suitable for time-sensitive applications, while P256-Falcon 512 is suitable for applications prioritizing low storage overheads.

## VII. Related Work

**Post-Quantum PKI** Previous research studies post-quantum PKI. Similar to our feasibility study, previous research had concerns from the increasing size in the public keys and certificates and conducted feasibility/viability analyses [24], [21]. When X.509 is used for TLS 1.3, the post-quantum signatures are greater than the maximum allowed certificate size by TLS 1.3 and are thus incompatible [21]. Our research identifies that the restriction is not on the X.509 itself but rather on the TLS 1.3 use of X.509 certificate, i.e., TLS 1.3 can restrict the certificate size and interfere with the post-quantum feasibility. This research informed and improved the cryptographic implementations and libraries, including the *Open Quantum Safe* project [22].

Other research in post-quantum PKI studies the hybrid ciphers, including developing the approaches for hybrid ciphers in certificates [25] (which we include in our study) as well as

building it based on the qTESLA algorithm [26] (no longer considered for NIST standardization).

**Post-Quantum Beyond PKI** While the previous focuses on PKI, other research prepares for the post-quantum transition by analyzing the post-quantum digital signature ciphers themselves [23], [27] or the incorporation and application to other digital networking protocols, such as for TLS 1.3 [28], [29], [30], blockchain [27], and embedded networking [31], [32].

## VIII. Conclusion

This paper characterizes the performance of the NIST finalist digital signature algorithms when used for PKI and digital certificate. We compare the cipher algorithms between the classical (RSA and ECDSA), the post-quantum (Dilithium, Falcon, and Rainbow), and the hybrid schemes (combining the classical and post-quantum ciphers). We test the feasibility of the post-quantum transition for the X.509 certificates and analyze the computing overheads for the distinct PKI functions of Key-Pair Generation, Certificate Signing Request Generation, Certificate Generation, and Certificate Verification as well as the storage overheads. Our measurements focus on the algorithm computing measurements and can be generalized across different PKI architectures and implementations. We analyze our experimental results and make recommendations to inform the post-quantum cipher R&D and to encourage PKI developers and practitioners to prepare for the quantum era by transitioning to the post-quantum primitives and ciphers.

## Acknowledgment

This research is based upon work supported by Colorado State Bill 18-086 and by the National Science Foundation under Grant No. 1922410.

## References

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[2] W. J. Caelli, E. P. Dawson, and S. A. Rea, "Pki, elliptic curve cryptography, and digital signatures," *Computers & Security*, vol. 18, no. 1, pp. 47–66, 1999.

[3] 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.

[4] N. Bindel, U. Herath, M. McKague, and D. Stebila, "Transitioning to a quantum-resistant public key infrastructure," in *International Workshop on Post-Quantum Cryptography*. Springer, 2017, pp. 384–405.

[5] "Nist-call for proposals," https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf, Last accessed 9 Mar 2021.

[6] G. Alagic, J. Alperin-Sheriff, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, D. Moody, R. Peralta *et al.*, "Status report on the second round of the nist post-quantum cryptography standardization process," *NIST, Tech. Rep., July*, 2020.

[7] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. T. Polk *et al.*, "Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile." *RFC*, vol. 5280, pp. 1–151, 2008.

[8] H. Finney, L. Donnerhacke, J. Callas, R. L. Thayer, and D. Shaw, "OpenPGP Message Format," RFC 4880, Nov. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc4880.txt

[9] D. R. Kuhn, V. Hu, W. T. Polk, and S.-j. H. Chang, "Sp 800-32. introduction to public key technology and the federal pki infrastructure," 2001.

[10] M. Omar, Y. Challal, and A. Bouabdallah, "Certification-based trust models in mobile ad hoc networks: A survey and taxonomy," *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 268–286, 2012.

[11] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, "A security credential management system for v2v communications," in *2013 IEEE Vehicular Networking Conference*, 2013, pp. 1–8.

[12] C. Chen, S. Chang, Y. Hu, and Y. Chen, "Protecting vehicular networks privacy in the presence of a single adversarial authority," in *2017 IEEE Conference on Communications and Network Security (CNS)*, 2017, pp. 1–9.

[13] B. Brecht, D. Therriault, A. Weimerskirch, W. Whyte, V. Kumar, T. Hehn, and R. Goudy, "A security credential management system for v2x communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3850–3871, 2018.

[14] O. Regev, "Lattice-based cryptography," in *Annual International Cryptology Conference*. Springer, 2006, pp. 131–141.

[15] H. Nejatollahi, N. Dutt, S. Ray, F. Regazzoni, I. Banerjee, and R. Cammarota, "Post-quantum lattice-based cryptography implementations: A survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, pp. 1–41, 2019.

[16] 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*, pp. 238–268, 2018.

[17] "Crystals-dilithium," https://pq-crystals.org/dilithium/, Last accessed 14 Mar 2021.

[18] 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*, 2018.

[19] "Rainbow-website," https://www.pqcrainbow.org/, Last accessed 5 Mar 2021.

[20] "Rainbow-nist-submission," https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Rainbow-Round3.zip, Last accessed 1 Sept 2020.

[21] E. Crockett, C. Paquin, and D. Stebila, "Prototyping post-quantum and hybrid key exchange and authentication in tls and ssh." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 858, 2019.

[22] D. Stebila and M. Mosca, "Post-quantum key exchange for the internet and the open quantum safe project," in *International Conference on Selected Areas in Cryptography*. Springer, 2016, pp. 14–37.

[23] M. Raavi, S. Wuthier, P. Chandramouli, Y. Balytskyi, X. Zhou, and S.-Y. Chang, "Security comparisons and performance analyses of post-quantum signature algorithms," *International Conference on Applied Cryptography and Network Security (ACNS)*, 2021.

[24] P. Kampanakis, P. Panburana, E. Daw, and D. Van Geest, "The viability of post-quantum x. 509 certificates." *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 63, 2018.

[25] N. Bindel, J. Braun, L. Gladiator, T. Stöckert, and J. Wirth, "X. 509-compliant hybrid certificates for the post-quantum transition," *Journal of Open Source Software*, vol. 4, no. 40, p. 1606, 2019.

[26] G. Pradel and C. J. Mitchell, "Post-quantum certificates for electronic travel documents," 2019.

[27] T. M. Fernández-Caramés and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks," *IEEE Access*, vol. 8, pp. 21 091–21 116, 2020.

[28] C. Paquin, D. Stebila, and G. Tamvada, "Benchmarking post-quantum cryptography in tls," in *International Conference on Post-Quantum Cryptography*. Springer, 2020, pp. 72–91.

[29] P. Schwabe, D. Stebila, and T. Wiggers, "Post-quantum tls without handshake signatures," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020, pp. 1461–1480.

[30] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, "Assessing the overhead of post-quantum cryptography in tls 1.3 and ssh," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, 2020, pp. 149–156.

[31] K. Bürstinghaus-Steinbach, C. Krauß, R. Niederhagen, and M. Schneider, "Post-quantum tls on embedded systems: Integrating and evaluating kyber and sphincs+ with mbed tls," in *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020, pp. 841–852.

[32] K. Basu, D. Soni, M. Nabeel, and R. Karri, "Nist post-quantum cryptography-a hardware evaluation study." *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 47, 2019.