Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms

Manohar Raavi[1], Simeon Wuthier[1], Pranav Chandramouli[1], Yaroslav Balytskyi[2], Xiaobo Zhou[1], and Sang-Yoon Chang[1]

[1] University of Colorado, Colorado Springs, USA
Department of Computer Science
[2] University of Colorado, Colorado Springs, USA
Department of Physics and Energy Science
{mraavi,swuthier,pchandra,ybalytsk,xzhou,schang2}@uccs.edu

Abstract. Quantum computing challenges the computational hardness assumptions anchoring the security of public-key ciphers, such as the prime factorization and the discrete logarithm problem. To prepare for the quantum era and withstand the attacks equipped with quantum computing, the security and cryptography communities are designing new quantum-resistant public-key ciphers. National Institute of Standards and Technology (NIST) is collecting and standardizing the post-quantum ciphers, similarly to its past involvements in establishing DES and AES as symmetric cipher standards. The NIST finalist algorithms for public-key signatures are *Dilithium*, *Falcon*, and *Rainbow*. Finding common ground to compare these algorithms can be difficult because of their design, the underlying computational hardness assumptions (lattice based vs. multivariate based), and the different metrics used for security strength analyses in the previous research (qubits vs. quantum gates). We overcome such challenges and compare the security and the performances of the finalist post-quantum ciphers of *Dilithium*, *Falcon*, and *Rainbow*. For security comparison analyses, we advance the prior literature by using the depth-width cost for quantum circuits (DW cost) to measure the security strengths and by analyzing the security in Universal Quantum Gate Model and with Quantum Annealing. For performance analyses, we compare the algorithms' computational loads in the execution time as well as the communication costs and implementation overheads when integrated with Transport Layer Security (TLS) and Transmission Control Protocol (TCP)/Internet Protocol (IP). Our work presents a security comparison and performance analysis as well as the trade-off analysis to inform the post-quantum cipher design and standardization to protect computing and networking in the post-quantum era.

Keywords: Quantum-Resistant Cryptography, Post-Quantum Cryptography, Quantum Computing, Public-Key Cipher, Digital Signature Algorithms, Security Analysis, Performance Analysis

1 Introduction

Public-key digital signatures provide authentication and integrity protection and are critical in securing digital systems. The security of the most used present-day digital signature standards like Rivest-Shamir-Adleman (RSA) [44] and Elliptic Curve Digital Signature Algorithm (ECDSA) [14] are based on the computational hardness problems such as prime factorization and discrete logarithm problem. Shor's polynomial-time algorithm effectively solve these problems when equipped with a powerful quantum computer [45]. Securing digital communication against attackers that have access to the quantum computing resources [39] requires new public-key cryptographic algorithms which can withstand such quantum attackers.

Recent advancements in quantum computing and quantum computers (Section 2) yields a need for transitioning to post-quantum cryptography (quantum-resistant cryptography). This involves identifying the relevant hardness problems and designing and constructing the quantum-resistant algorithms for securing digital communications.

National Institute of Science and Technology (NIST) launched the Post-Quantum Cryptography (PQC) standardization project to establish and standardize quantum-resistant algorithms. NIST has a track record of preparing for the impending cryptanalysis and breaks on cryptographic ciphers and is preparing for the post-quantum era before the emergence of the practical quantum computer implementations capable of breaking the current systems. NIST's involvement in cryptography has global and lasting impacts on digital systems, as demonstrated by their involvement in standardizing DES in the 1970's and AES in the late 1990's. The standardization process starts with an open public call that lists the requirements of the algorithms. All the submission algorithms are openly published and subjected to analyses and, after the analyses, an algorithm is selected for standardization. PQC standardization project follows the same process and requested the interested parties to provide submissions for quantum-resistant cryptographic candidates.

In this paper, we study the security and performances of the third round digital signature candidate algorithms from NIST's PQC standardization project. The digital signature algorithms include three finalist candidates: *Crystals-Dilithium* (*Dilithium*), *Falcon*, and *Rainbow*, design principles and hardness problems underlying these digital signature algorithms come from Lattice-based and multivariate-based cryptography (Section 3). The researchers designed and developed these cipher schemes and algorithms separately, including the security analyses, which makes the cross-cipher comparison analyses challenging. Our work addresses such a gap and provides a comparison analyses between the different cipher families/schemes and the algorithms within the families to inform the security communities at this time of selecting the PQC cipher standard[3].

[3] Our work has been shared with NIST.

Contributions We compare the PQC digital signature cipher schemes in both security and performances. Our security analyses include two contributions. First, we adopt a model for visual representations to compare the size-security trade-offs of digital signature algorithms (Section 4). These include the security offered by the digital signature algorithms with respect to public key length and signature length. Second, we analyze the security of digital signature candidates based on DW cost and with and without quantum annealing (Section 5). We also make the following two contributions in performance analyses. First, we analyze the implementation overheads in the execution time and its scalability in message sizes of the individual signature algorithms for key-pair generation, signature generation, and signature verification (Section 6). Second, we analyze the communication/handshake overheads on TLS 1.3 and TCP/IPv4 connection when integrating the signature algorithms (Section 7).

Methodologies And Approaches We use both theory and empirical measurements in our paper. We take a theoretical approach to analyze the security against quantum cryptanalysis, since practical quantum computers supporting a sufficient number of quantum bits (qubits) to implement the theoretical cryptanalysis attacks are currently unavailable and under active research and development. In Section 4, we build on the previous analyses on the NIST finalist PQC schemes and adopt the visualization model proposed by Bernstein [12] to show the size-security trade-offs of the digital signature algorithms. In Section 5, we build on the quantum physics theory to compare and analyze the PQC schemes, which previously have been analyzed separately using different metrics of qubits and quantum gates, challenging the inter-scheme comparison. We adopt the quantum circuit DW cost combining qubits and quantum gate costs [34] to compare the security of the algorithms in the Universal Quantum Gate Model and the model with Quantum Annealing.

For the performance analyses of the PQC schemes, we implement prototypes and empirically measure performances between the schemes. While the post-quantum era prepares for adversaries equipped with quantum computing, the PQC cipher schemes are designed to be implemented on classical computers to defend the common user. We build on the algorithm implementations from *liboqs* by *Open Quantum Safe* [47] and deployed it on a virtual machine with 6 processing cores and 6 *GB* of RAM on a computer equipped with a 16-core 32-thread AMD Ryzen 9 3950X processor with 3.5 GHz processor frequency, and 32 *GB* RAM. While the absolute performance costs vary depending on the computer platform and its hardware specifications, we focus on the comparison results in this paper so that our results and insights are applicable to classical non-quantum computers beyond our platform. We compare the performance costs of the algorithms by themselves in Section 6 and when integrated with TLS 1.3 and TCP/IP in Section 7.

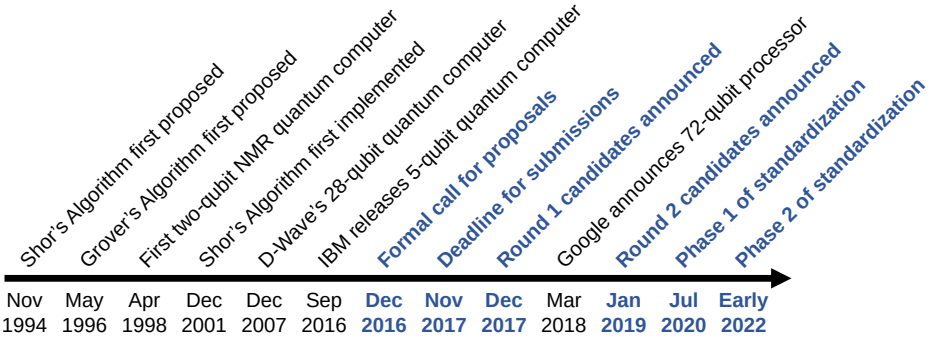


Fig. 1: PQC history and the events leading up to NIST's standardization. Highlighted in blue are the events specifically involving NIST.

2 Background: PQC History and NIST

We describe the PQC history and the NIST involvement to motivate our work in this section, and Figure 1 shows the timeline of events regarding cryptography and quantum computing. Cryptography provides the backbone to secure the digital systems in our society. While the practical quantum computers currently only support a small number of qubits and at the proof-of-concept stages, the theoretical algorithms building on quantum computing emerged to expedite the solving of the computational hardness problems anchoring the security of the current public-key ciphers. Shor's algorithm [45], invented in 1994, provides a polynomial-time algorithm in quantum computing for solving the prime factorization and discrete log problem, threatening the security of the public-key ciphers such as RSA and Diffie-Hellman Key Exchange. Grover's algorithm [31] in 1996 expedites the brute-force search of the general problems, such as the original database search problem and hash collision finding. The first successful implementation of quantum searching was performed, in 1998, on a two-qubit Nuclear Magnetic Resonance (NMR) quantum computer. It used Grover's algorithm to search for a system that has four states [19]. Followed by Grover's implementation, in 2001, a seven qubit NMR quantum computer [48] used Shor's algorithm to find prime factors for the number 15.

Inspired by the extraordinary opportunities in quantum computing, major tech giants started research into quantum computers. The first of such kind was D-Wave Systems, whose quantum computing capabilities are based on quantum annealing. In 2019, D-Wave unveiled a 5000 qubit processor [49]. On the other hand, International Business Machines (IBM) and Google follow the universal quantum gate model. IBM showed significant progress from 5 qubit processor [9] and reached a 53 qubit processor. IBM's quantum experience provides access to up to 32 qubit processor quantum computers for registered users at free of cost [16]. Google announced a 72 qubit quantum processor [35]. With steady growth in practical quantum computers and raising concerns in cryptogra-

Table 1: NIST security categories, where X is the MAXDEPTH

Security Category	**Reference Algorithm**	**Classical Bit Cost**	**Qubit Security**	**Circuit Size to Break the Algorithm [4]**
1	AES 128	128 (key search)	64 (Grover [31])	$2^{170}/X$ quantum gates or 2^{143} classical gates
2	SHA3-256	128 (collision)	85 (Brassard [13])	2^{146} classical gates
3	AES 192	192 (key search)	96 (Grover [31])	$2^{233}/X$ quantum gates or 2^{207} classical gates
4	SHA3-384	192 (collsion)	128 (Brassard [13])	2^{210} classical gates
5	AES 256	256 (key search)	128 (Grover [31])	$2^{298}/X$ quantum gates or 2^{272} classical gates
6	SHA3-512	256 (collision)	170 (Brassard [13])	2^{274} classical gates

phy, NIST initiated PQC standardization project. NIST requested nominations in Dec 2016 for public-key post-quantum cryptographic algorithms [4]. NIST's PQC standardization project aims to replace the present recommended digital signature standard of RSA, and ECDSA [27] with post-quantum algorithms. Instead of measuring the security in bits, all these post-quantum algorithms are referenced with security categories defined by NIST (Table 1). Each of these categories sets the minimum required computational resources to break well known symmetric block cipher or hash functions. Breaking a symmetric block cipher indicates a successful brute-force key search attack, and breaking a hash function means a successful brute-force collision attack. Computational resources can be restricted by a new parameter, defined by NIST, called MAXDEPTH. It can be used to limit the quantum attacks to fixed running time or circuit depth. PQC standardization is a multi-year process and involves multiple rounds of analyses and scrutiny for the maturity of cipher design before the standardization.

In response to the open public call for the PQC standardization proposal [4], NIST received 82 submissions, and only 69 of the submissions satisfied the minimum required conditions. The first-round of the PQC standardization project started in Dec 2017, selected 26 candidate algorithms. With fewer algorithms to analyze, the second-round of the PQC standardization project started in Jan 2019 and selected 15 algorithms for third-round [5]. These 15 algorithms are categorized as seven finalist candidates and eight alternate candidates (Section 3). Seven finalists candidate algorithms include four public-key encryption/KEM mechanisms and three digital signature schemes. Eight alternate candidate algorithms include five public-key encryption/KEM mechanisms and three digital signature schemes. At-most one of the finalist candidate algorithms in each of the public-key encryption/KEM and digital signature standard schemes are expected to be standards at the end of the PQC standardization project. Started in July 2020, the PQC standardization project aims to complete the analysis by early 2022 and confirm the candidates for standardization.

We focus on the digital signatures in this paper, while NIST solicits and plans to standardize both key exchange ciphers/key encapsulation mechanisms

Table 2: Signature algorithms descriptions (the last column indicates the color code used in our paper)

Algorithm	Scheme	NIST Status	Security Level	Reference	Color
Dilithium	Lattice-based	Finalist	1 (AES128) 2 (SHA256) 3 (AES192)	[23]	Blue
Falcon	Lattice-based	Finalist	1 (AES128) 5 (AES256)	[28]	Red
Rainbow	Multivariate	Finalist	1 (AES128) 3 (AES192) 5 (AES256)	[22]	Green
GeMSS	Multivariate	Alternate	1 (AES128) 3 (AES192) 5 (AES256)	[15]	Magenta

and digital signature ciphers. The current Round 3 includes three finalists and three alternate schemes for the digital signatures. At most one of the finalist schemes is expected to be standardized in Round 3, which is planned to occur in 2022. Our work focuses on the NIST finalist candidates for PQC signature schemes/algorithms in: *Dilithium*, *Falcon*, and *Rainbow*.

3 PQC Signature Schemes

The current third-round PQC standardization is planned to undergo public and researchers scrutiny/analyses until 2022 and includes three digital signatures schemes for its finalists: *Dilithium*, *Falcon*, and *Rainbow*. These families of algorithms can be categorized into two different schemes, listed in Table 2, based on the hardness problems on which they rely. These include Lattice-based signature schemes (for *Dilithium* and *Falcon*) in Section 3.1 and Multivariate-based signature schemes (for *Rainbow*) in Section 3.2. We also include the description for *GeMSS* (the only other Multivariate-based signature scheme which got selected as an alternate scheme by NIST) in order to compare it with the security of *Rainbow* in Section 4.

Within each algorithm families (*Dilithium*, *Falcon*, and *Rainbow*), there are multiple algorithms depending on the parameter choices for the security level control. The parameters affect the public key length and the private key length as well as the signature length for the PQC signature algorithms, and the security strengths increase as the length increase (using greater number/options to yield greater entropy), as we study in Section 4. The public key and the signature lengths for the different algorithms are also listed in the horizontal axis in Figure 2. For example, *Dilithium 2* has a public key length of 897 Bytes, *Dilithium 3* with 1472 Bytes, and *Dilithium 4* with 1760 Bytes. Thus, *Dilithium 4* is designed to have greater security strength than *Dilithium 3*, and *Dilithium 3* greater than *Dilithium 2*. This section describes the overview of the PQC schemes without the scheme-specific details, such as the actual parameters/variables to control for the different algorithms within the family; we refer the interested readers for the scheme-specific design details to the design documents for *Dilithium* [23], *Falcon* [28], and *Rainbow* [22].

3.1 Lattice-based signature schemes

Lattice-based signature schemes are based on a set of points in n-dimensional space with periodic structure [42]. The security of lattice-based cryptography comes from the use of NP-hard problems such as i) Short Vector Problems (SVP) which involves finding the shortest non-zero vector, ii) Closed Vector Problems (CVP) which involves finding the shortest vector, iii) Learning With Errors (LWE) which is computationally intensive as it requires a linear function over a finite ring of given samples, iv) Short Integer Solutions (SIS) which is based on Ajtai's theorem where if polynomial-time algorithm A solves the SIS problem, then there exists an Algorithm B that can solve the Short Vector Integer Problem (SVIP), v) Learning With Rounding (LWR) is the non-rounding variant of LWE. LWR is more efficient than LWE as it removes LWE's complex randomization elements [40]. *Dilithium* and *Falcon* are the two finalist lattice-based signatures schemes in the NIST third-round standardization process. Both are categorized as finalist candidate algorithms, and one of the two is expected to be a digital signature standard at the end of the PQC standardization process in 2022-2024 [3].

Dilithium *Dilithium* is one of the two lattice-based signature algorithms in the third round. *Dilithium* relies on Fiat-Shamir and Aborts framework and SVP's for its security [23]. *Dilithium* introduces three algorithms in *Dilithium 2*, *Dilithium 3*, and *Dilithium 4* correspond 1, 2, and 3 of NIST's post-quantum security categories respectively.

Falcon *Falcon* is Fast Fourier lattice-based compact signatures over N-th Degree Truncated Polynomial Ring (NTRU) [28], and one of the two lattice-based signature algorithms of the third round. *Falcon* relies on the NTRU for key generation, encryption and decryption data, Short Integer Problems (SIS), Floating-Point arithmetic, and gaussian sampling floating-point arithmetic for its security. *Falcon 512* and *Falcon 1024* algorithms correspond to 1 and 5 of NIST's post-quantum security strengths respectively.

3.2 Multivariate-based signature schemes

Multivariate-based signature schemes are known as an unbalanced Oil-Vinegar (UOV) system which is the process of hiding quadratic equations in n unknowns or Oil and $v = n$ unknowns called "vinegar" in a finite field k [36], and it is based on solving quadratic equations over finite fields, making it an NP-hard problem. The security of the signature scheme is based on the number of variables and the field size, which leads to large key sizes. *Rainbow* is the only finalist from multivariate-based signature schemes.

Rainbow *Rainbow*, a Multivariate signature scheme, is the only finalist Multivariate candidate of the third round. *Rainbow* relies on binary field L over

Unbalanced Oil and Vinegar (LUOV) and raises it to a bigger field K for its security [24]. The *Rainbow* family includes the following algorithms: *Rainbow Ia, Rainbow IIIc,* and *Rainbow Vc* where *I*, *III*, and *V* correspond to 1, 3, and 5 of NIST's post-quantum security strength, respectively. *Rainbow* has variants of cyclic and compressed algorithms.

GeMSS A Great Multivariate Short Signature (*GeMSS*) is one of the alternate candidates in the third round. *GeMSS* relies on Hidden Field Equations cryptosystem (HFE) to achieve its level of security and efficiency. *GeMSS* algorithm variants are *GeMSS 128*, *GeMSS 192*, *GeMSS 256* where 128, 192 and 256 correspond to 1, 3, and 5 of NIST's post-quantum security strengths respectively. *GeMSS* has variants of *BlueGeMSS* and *RedGeMSS*.

4 Security Analysis Using Visualization Model

In this section, we analyze the size-security trade-offs of NIST finalist algorithms. We build our analyses on the visualization model developed to compare cipher designs [12] and the individual analyses of the NIST finalist cipher designs, including the individual cipher developer's security analyses and the threats/attack discoveries and research targeting the individual cipher schemes.

Lattice-based algorithms analyze the cipher security strength against quantum attackers (the attacker's cost in breaking the ciphers) in qubit cost, and multivariate-based algorithms analyze the security strength in quantum gates cost. For multivariate-based algorithms, we include the comparison with *GeMSS* for *Rainbow*; *GeMSS* is the only other multivariate-based family selected by NIST to advance in the third round but it is selected as an Alternate as opposed to Finalist. The metrics of qubit and quantum gate to measure the security strength are relatively new and depend on the quantum computer physics and hardware architecture, which are in active research and development.

In this section, we separately analyze the size-security trade-offs of the lattice-based schemes using the attacker's security cost in qubits and the size-security trade-offs of the multivariate schemes using the attacker's cost in quantum gates. However, in Section 5, unlike the previous research approach analyzing the PQC ciphers individually, we compare them together by using the quantum circuit's depth-width cost (DW cost) which incorporates both qubit and quantum gate.

4.1 Metrics: Qubits and Quantum Gates

In classical computing, the state of a particular bit is always known. In the quantum case, before the measurement is done, the state of a qubit is unknown. Although qubit resembles a classical bit *after the measurement*, it takes two possible values and additionally can exploit the interference effects; *before the measurement*, a qubit can be in a *superposition* of these two states described by the wave function. The classical brute force attacker searches blindly and cannot distinguish if some particular value is closer or further from the key

being searched and while checking if the key is found may get only the binary result of "true" or "false". In the quantum case, a brute force attacker always has an overlap of the wave function of the current state of qubits and the key, and this overlap is bigger if some particular state of qubits is closer to the key being searched. Even if the cipher's structure is unknown for the attacker, interference helps to move in the direction towards the key enabling a faster search.

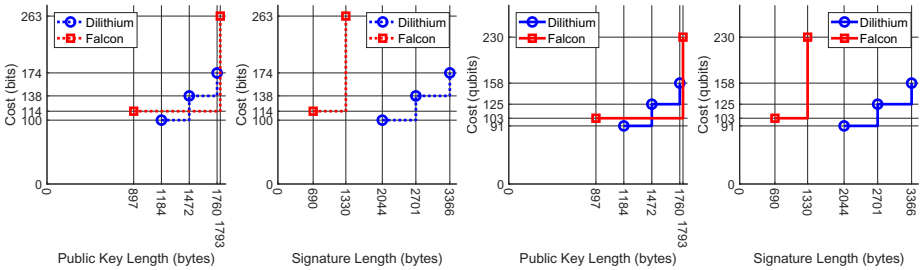
Qubit Cost A qubit cost indicates the number of qubits the attacker needs to break the cipher assuming a sufficient number of gates. For *Dilithium*, qubit cost represents the cost of solving SVP which is exponential to Block-Korkine-Zolotarev (BKZ) algorithm's block-size (b) [23] For *Falcon*, core SVP hardness indicates the cost for one call to SVP oracle in dimension b [28], [8].

Quantum Gate Cost Quantum gate cost indicates the minimum number of logical quantum gates required to perform a successful attack assuming a sufficient number of qubits. For Rainbow, quantum gate cost indicates the minimum number of logical gates required to perform key recovery attacks including MinRank, HighRank, UOV, and RBS attacks [22]. Against Rainbow, the number of quantum gates (#Gates) can be measured as follows where q is the Galois Field's Order: #Gates = #Field Multiplications $\cdot(2 \cdot \log_2(q)^2 + \log_2(q))$.

4.2 Lattice-based signature schemes security and length trade-off

Figure 2 shows the classical and quantum security cost trade-offs with respect to the public key length and the signature length of lattice-based signature algorithms. Public key length and signature length are important parameters controlling the trade-off between security strength and communication overhead in digital communications. For example, when Alice sends a signed message to Bob, Bob uses Alice's public key and signature to verify the message. Public key and signature are the overheads in the communication. Many real-world applications use the same private/public key-pair to sign/verify multiple messages than using One Time Pad/Ephemeral Keys resulting in the transfer of signatures more often than public-keys. Low signature size and public key lengths help in reducing communication costs. In Figure 2, an ideal algorithm will have a small communication cost (left in the plot) and provide strong security (top).

Dilithium and *Falcon*, the two lattice-based NIST finalists schemes, display the security vs. overhead/length trade-off, and the security costs measured in (classical) bits or qubits increase as the public key length or the signature length increase. Figure 2a analyzes the classical security cost to the public key length trade-offs. For additional 33 Bytes of public key length from *Dilithium* at 1760 Bytes, *Falcon* offers 89 bits more security cost. Figure 2b analyzes the classical security cost to the signature length trade-offs. For less than half the signature size of the *Dilithium* at 2044 Bytes, the classical security cost of the *Falcon* at 690 Bytes is 14 bits more. *Falcon* signature size at its highest security cost is 714 Bytes less to *Dilithium* at its lowest security. Figure 2c analyzes the



(a) Security in bits vs. public key length (b) Security in bits vs. signature length (c) Security in qubits vs. public key length (d) Security cost vs. signature length

Fig. 2: Security costs for lattice-based signature candidates. The dotted line shows the classical bit security level (the left two figures), whereas the solid line shows the quantum bit security level (the right two figures). The vertical scales are consistent across the figures.

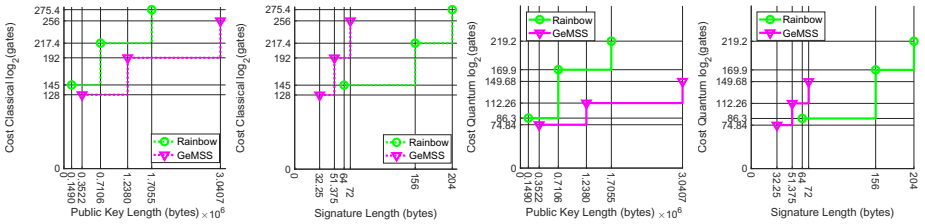
quantum security cost to the public key length trade-offs. For an additional 33 Bytes of public key length from *Dilithium* at 1760 Bytes, *Falcon* offers 72 qubits more quantum security cost. Figure 2d analyzes the quantum security cost to the signature length. For less than half the signature size of the *Dilithium* at 2044 Bytes, the quantum security cost of the *Falcon* at 690 Bytes is 12 qubits more. *Falcon* is close to ideal reference position of top-left corner compared to *Dilithium*.

For applications targeting security levels 2 and 3, *Falcon* doesn't have parameter sets and yet it is advantageous to use *Falcon 1024* security level 5 parameter set as it provides better security for less overhead compared to *Dilithium* parameter sets.

4.3 Multivariate-based signatures security and length trade-off

Figure 3 shows the classical and quantum security trade-offs with respect to the public key length and the signature length of multivariate-based signature algorithms. *Rainbow* and *GeMSS* security costs are measured in both classical gates and quantum gates, and the security costs increase as the public key length and signature length increases. Multivariate-based signature algorithms generate larger public keys and small signatures compared to the lattice-based schemes in Section 4.2.

Figure 3a analyzes the classical gate cost to the public key length trade-offs. Horizontal axis of the plot has Bytes of order 10^6 for huge public key lengths. For less than half the public key length of *GeMSS* at 0.3522×10^6 Bytes, *Rainbow* at 0.1490×10^6 Bytes requires 2^{17} more classical gates. Figure 3b analyzes the classical gate cost to the signature length trade-offs. *Rainbow* generates signature length of two to three times than that of *GeMSS* signature size for respective security categories. Figure 3c analyzes the quantum gate cost to the



(a) Security in classical gates vs. public key length

(b) Security in classical gates vs. signature length

(c) Security in quantum gates vs. public key length

(d) Security in quantum gates vs. signature length

Fig. 3: Security costs for multivariate-based signature candidates. The dotted line shows the classical gate security level (the left two figures), whereas the solid line shows the quantum gate security level (the right two figures). The vertical scales are in $\log_2(\#\text{Gates})$ and are consistent across the figures.

public key length trade-offs. For less than half the public key length of *GeMSS* at 0.3522×10^6 Bytes, *Rainbow* at 0.1490×10^6 Bytes requires 2^{12} more quantum gates. Figure 3d analyzes the quantum gate cost to the signature length trade-offs. Overall, *Rainbow* has smaller key lengths and greater security costs compared to *GeMSS*. Our analysis shows that with the current cryptanalysis research and knowledge, *Rainbow* provides a greater length- or bit-efficient scheme over *GeMSS*.

5 Security Analysis Between Dilithium, Falcon, and Rainbow

In this section, we present a security analysis of PQC algorithms with respect to qubit, quantum gate, and DW cost metrics working in the framework of the universal quantum gate model and quantum annealing against a cryptanalyst using Grover's algorithm (discussed in Section 2). Even though the computational hardness problems differ between the lattice-based (*Dilithium* and *Falcon*) vs. multivariate-based (*Rainbow*), Grover's algorithm-based attacker enables us to compare the security costs for its generality which expedites the search/brute-force process.

5.1 Universal Quantum Gate and Quantum Annealing

We provide a brief overview in this section about the universal quantum gate model, Clifford+T gates, and quantum annealing to provide the background of our inter-scheme security comparison analysis. While the universal quantum gate model is applicable to all schemes, including the three NIST finalist schemes we compare, quantum annealing is only applicable to the lattice-based schemes of *Dilithium* and *Falcon*.

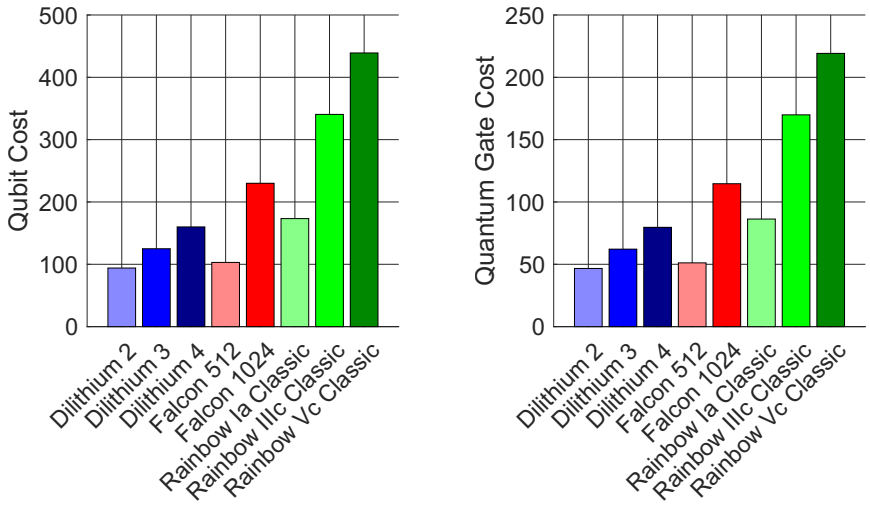
The *universal quantum gate model* consists of gates connected by wires [21]. Quantum gates are the basic building blocks that perform operations on a small number of qubits. Clifford+T gates are used as the universal underlying fault-tolerant logical quantum gate set [30]. Wires represent the motion of a carrier which encodes information both logically and physically. External inputs and outputs are provided by sources and sinks respectively. The structure of the ciphers implies the methods which can be used to break them. The security of *Dilithium* and *Falcon* is based on the hardness of finding the shortest vectors on the lattice, while the security of *Rainbow* is based on the hardness of solving multivariate polynomials over a finite field. Finding shortest vectors can be reformulated as a problem of finding a minimum, which can potentially be solved by *quantum annealing* [26, 32] which is designed to find a minimum of the cost function while being less demanding in terms of quantum error correction. Of course, finding a minimum is also possible by the universal quantum gate model [25]. However, in addition to qubits, its implementation would be complicated by the need for a large number of gates.

Quantum annealing works in the following way: the system is initialized in the ground state of the Hamiltonian H_{init} which we assume to be simple to implement. The minimum which we are looking for corresponds to the ground state of another Hamiltonian, H_{key}. Over the time τ, we change the Hamiltonian such that $H_{init} \rightarrow H_{key}$. Adiabatic theorem [10, 18, 38, 43] guarantees that if the time τ is sufficiently large and the change is sufficiently slow (adiabatical), then the system will end up in the ground state of the Hamiltonian H_{key} which corresponds to the key being searched. As we show in Section 5.4, the combination of *quantum annealing* and *universal gate model* provides significant advantages for the attacker.

5.2 Metric for All Ciphers: DW Cost to Combine Qubits and Quantum Gates

The security analyses of the PQC schemes in Section 4 and in the prior research literature are separate in the security strength metrics used, which makes them difficult to compare. The security costs (the attacker effort/requirement to break the ciphers) of *Dilithium* and *Falcon* are given in qubits, while the security of *Rainbow* is measured in terms of quantum gates, as discussed in Section 4. In this section, we introduce and apply the depth-width cost (DW cost) to our post-quantum cipher analyses to enable the inter-scheme comparison, in contrast to the previous research. The number of RAM operations needed to execute a quantum circuit is proportional to the DW cost metric which incorporates both the cryptanalyst cost factors in qubits and quantum gates and assumes an active error correction [34][4].

[4] In addition to DW metrics, Jaques and Schanck [34] introduce G cost metrics for self-correcting quantum memory. However, self-correcting quantum memory is not available even theoretically. The two dimensional toric code [37], is not thermally stable [6]. Even though in a non-physical case of four spatial dimensions, it is ther-



(a) Security Cost Comparison in qubits (b) Security Comparison in quantum gates

Fig. 4: Security cost comparison across signature candidates

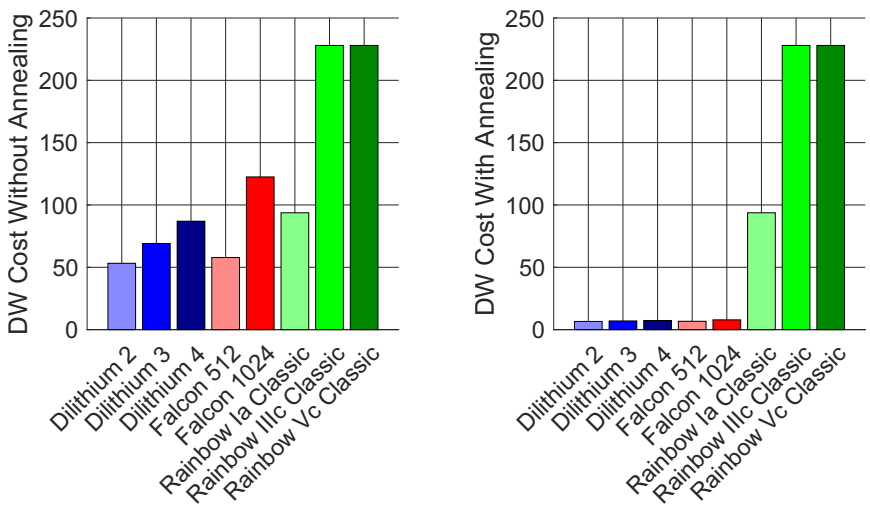
Definition 1. *A logical Clifford+T quantum circuit having D gates in-depth, W qubits in width, and consisting of an arbitrary number of gates is assigned a DW cost of $\theta(DW)$ RAM operations.*

From Definition 1, DW cost incorporates both qubits and quantum gates and thus enables the comparison between the NIST digital signature finalists.

5.3 Security Analysis in Universal Quantum Gate Model

We provide estimates assuming an adversary executing Grover's search algorithm [31] for the key search in a universal quantum circuit model, introduced in Section 4.1. If there are N key options among which one is used by the authorized sender and receiver, the adversary needs on average $\frac{\pi\sqrt{N}}{4}$ gates and $\log_2(N)$ cubits. Therefore, G quantum gates corresponds to $W = 2 \times \log_2(\frac{4}{\pi}G)$ qubits. Figure 4a compares the cryptanalyst costs of the PQC algorithms in qubits while Figure 4b does so in quantum gates. From the security analyses, *Rainbow* is much more expensive for a cryptanalyst to attack than *Dilithium* and *Falcon* in both gates and in qubits. For example, in terms of both qubits and quantum gates, *Rainbow Vc Classic* is 93% more expensive than *Dilithium 4* and 63% more expensive than *Falcon 1024*. The DW cost in the universal gate model alone is provided in Figure 5a. Due to the practical relevance of quantum annealing, we consider its impact in Section 5.4.

mally stable [7], it remains an open question if it is possible to implement it in three dimensions in which we live. Therefore, for our purposes, we use conservative DW cost metrics.



(a) DW cost, quantum universal gate model without annealing

(b) DW cost, quantum universal gate model with annealing

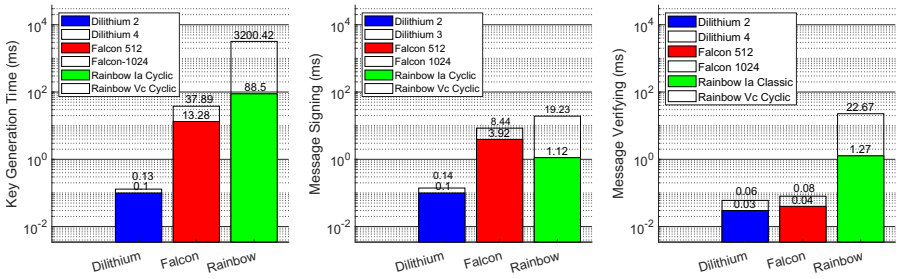
Fig. 5: Security cost comparison across signature candidates in DW cost without annealing and DW cost with annealing.

5.4 Security Analysis with Quantum Annealing

Quantum annealing is much more restricted than the universal quantum gate model; for example, it cannot execute Shor's algorithm. However, since it relies on adiabatic Hamiltonian evolution rather than on the gates, it's much cheaper in terms of the DW metrics. Figure 5b shows our analysis results. If quantum annealing is used to break *Dilithium* and *Falcon* by finding the minimum of the length of the vector on a lattice (quantum annealing is not applicable to *Rainbow*), it can significantly expedite the cryptanalysis effort for those lattice-based ciphers, and the gap in the security cost between *Rainbow* and the others grow even further. Even without quantum annealing, as seen in Section 5.3, *Rainbow Vc Classics* costs more for the adversary to break than *Dilithium* and *Falcon*. In DW cost, *Rainbow Vc Classic* is 188% more expensive than *Dilithium 4* and 187% more expensive than *Falcon 1024*. Our analyses based on a cryptanalyst using Grover's algorithm shows that *Rainbow* has the greatest security cost (the most computational effort for the adversary) with or without quantum annealing (applicable to the lattice-based schemes of *Dilithium* and *Falcon*).

6 Performance Analysis of Cipher Algorithms

In this section, we analyze the execution times of each finalist algorithm for key-pair generation, signature generation (signing), and signature verification



(a) Key generation time (b) Message signing time (c) Message verifying time

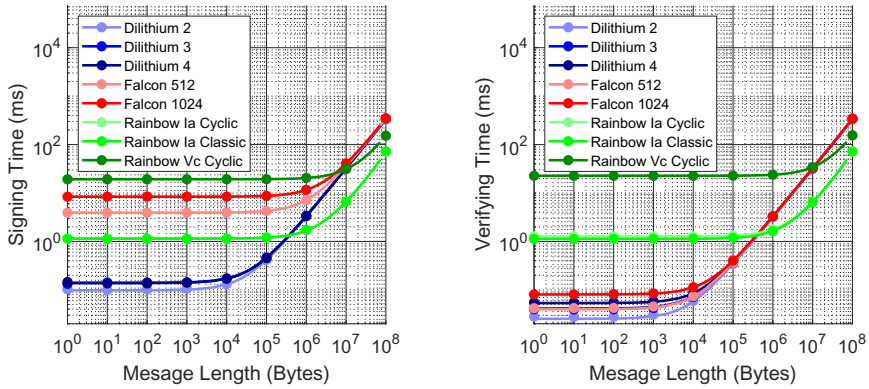
Fig. 6: Performance analysis of the fastest (colored bar) and slowest (outlined bar) signature candidates from each family across the three signature phases, with a message length of 100 Bytes.

(verifying). We use the *liboqs* library to analyze the performance of the algorithms. Using our benchmark software, each of the algorithms is sampled to calculate the average time duration for key-pair generation, signing a message, and verifying the messages. We also vary the message lengths with random data to measure the algorithm performances with respect to the message scalability. Each data point is averaged over 1000 runs, and the results are plotted.

Algorithm Performances Figure 6 shows the computing costs for every algorithm of each family. While we experiment with all algorithms, our presentation/plot focuses on the slowest from each algorithm family (outlined bar) and the fastest from each algorithm family to show the performance span across *Dilithium*(D), *Falcon*(F), and *Rainbow*(R).

We analyze the performances within each PQC family of algorithms in order to compare the performances between the algorithms once the scheme/family is selected, for example, an application chooses a PQC family or NIST selects the standard PQC signature. We introduce the *intra-family performance ratio*, R_i where i specifies the algorithm family, i.e., $i \in \{D, F, R\}$. For example, R_D is the ratio between the execution time for the slowest *Dilithium* algorithm and the time for the quickest *Dilithium* algorithm. By definition, $R_i > 1, \forall i$. By dividing the slowest algorithm's duration with the fastest for each family of algorithms, the key generation data (Figure 6a) shows that $R_D = 1.3$, $R_F = 2.85$, and $R_R = 36.16$. For the message signing phase (Figure 6b), $R_D = 1.4$, $R_F = 2.15$, and $R_R = 17.17$. For message verification (Figure 6c), $R_D = 2$, $R_F = 2$, and $R_R = 17.85$. This makes it clear that *Dilithium*, when compared to *Falcon* and *Rainbow*, shows optimal behavior with regards to the time taken to run the algorithms from each family.

While the intra-family comparison analysis helps in understanding how each algorithm in a family performs independently, an inter-family comparison analysis gives additional insights when compared with algorithms from other families. For key generation, our data shows that *Dilithium* is between 102.15 and 32004.2



(a) Signing time, varying message lengths (b) Verifying time, varying message lengths

Fig. 7: Performance analysis for the fastest and slowest candidates from each family. The horizontal axis is in logarithmic scale. The dots show the discrete experimental data, and the solid lines are our LoBF estimates.

times faster than the other finalist algorithms. For message signing, *Dilithium* is between 8 and 192.3 times faster than the other family of finalist algorithms. For message verification, *Dilithium* is up to 755.67 times faster than the other algorithms, but only 0.01 milliseconds faster than *Falcon*. From our intra-family and inter-family comparisons, we conclude that *Dilithium* is the fastest algorithm across all algorithms. We extend the analysis to algorithm scalability by using variable message lengths, to compare how the performance changes when the input size increases.

Message Scalability Performances We measure the signing and verifying performance of the PQC algorithms when the message length varies from 1 Byte to 100 MB in order to analyze the message scalability. Figure 7 shows the fastest and slowest algorithms in each family so that the algorithms within a cipher family have performances between the two algorithms. Using our results in Figure 7, we estimate the line of best fit (LoBF) based on minimizing the mean squared error to enable the analyses with greater precision. Table 3 includes the LoBF estimations. Our results from Figure 7a show that *Dilithium 2* is the optimal algorithm for signing, until the message length reaches the intersection with *Rainbow Ia Cyclic*, at 387578 Bytes. Therefore if the message length exceeds 387578 Bytes, *Rainbow Ia Cyclic* becomes the fastest signing algorithm instead of *Dilithium 2*. For verifying (Figure 7b), *Dilithium 2* is the fastest algorithm until a message length reaches 461478 Bytes, and for messages of greater length, *Rainbow Ia Classic* has the best verifying time.

Application Dependency The choice of the PQC cipher algorithm depends on its application since the cipher application determines the usage frequencies of

Table 3: The Line of Best Fit (LoBF) estimations for the signing time and verifying time in milliseconds, where x is the message length in Bytes

Algorithm	**LoBF (signing time)**	**LoBF (verifying time)**
Dilithium 2	$1.845 \times 10^{-15}x^2 + 3.215 \times 10^{-6}x + 0.09799$	$1.865 \times 10^{-15}x^2 + 3.213 \times 10^{-6}x + 0.02525$
Dilithium 3	$1.767 \times 10^{-15}x^2 + 3.225 \times 10^{-6}x + 0.1398$	$1.815 \times 10^{-15}x^2 + 3.221 \times 10^{-6}x + 0.04121$
Dilithium 4	$1.865 \times 10^{-15}x^2 + 3.216 \times 10^{-6}x + 0.1384$	$1.902 \times 10^{-15}x^2 + 3.211 \times 10^{-6}x + 0.05348$
Falcon 512	$1.783 \times 10^{-15}x^2 + 3.233 \times 10^{-6}x + 3.923$	$1.848 \times 10^{-15}x^2 + 3.225 \times 10^{-6}x + 0.03987$
Falcon 1024	$1.859 \times 10^{-15}x^2 + 3.216 \times 10^{-6}x + 8.4$	$1.867 \times 10^{-15}x^2 + 3.215 \times 10^{-6}x + 0.08054$
Rainbow Ia Cyclic	$1.802 \times 10^{-15}x^2 + 5.265 \times 10^{-7}x + 1.14$	$1.915 \times 10^{-15}x^2 + 5.135 \times 10^{-7}x + 1.271$
Rainbow Ia Classic	$1.792 \times 10^{-15}x^2 + 5.311 \times 10^{-7}x + 1.146$	$1.939 \times 10^{-15}x^2 + 5.143 \times 10^{-7}x + 1.129$
Rainbow Vc Cyclic	$1.463 \times 10^{-15}x^2 + 1.181 \times 10^{-6}x + 19.23$	$1.915 \times 10^{-15}x^2 + 1.135 \times 10^{-6}x + 22.67$

the signing vs. verifying and the message payload size. The frequency discrepancy of signing to verifying a message varies significantly according to the cipher applications For example, in cryptocurrency applications, once a transaction is created, it gets signed a single time. In contrast, as the transaction propagates across the network, every node verifies that the message is genuine [29], thus the signing-to-verifying ratio is close to zero in this case. We aim to prioritize the algorithm with fast message verification. In addition, the message payload size provided by the application requirement affects the performance prioritization and the cipher selection. Our results show that *Dilithium 2* is the most execution-time-efficient if the application message length is short and *Rainbow Ia Classic* for verifying if the message length is large.

7 Performance Analysis with Integration with TCP/IP and TLS

In this section, we analyze the communication/handshake overhead of the PQC algorithms at packet level when integrated with TLS 1.3 and TCP/IPv4. The handshake connection involves multiple transmissions between the client and the server, where the client initiates the connection by sending a *client hello* packet to the server and the server responds with *server hello* carrying the certificate signed by the Certificate Authority (CA), which contains the public key (post-quantum) of the server and the signature (post-quantum). The client then verifies the signature and sends a finished message to the server indicating the end of the handshake. After a successful handshake, application data is securely transferred. Our analysis focuses on packet-level overhead as opposed to the broader networking overhead between the two hosts. We use local virtual machines loaded with *OQS-OpenSSL_1_1_1* [47] acting as a client and server to establish a TLS 1.3 connection using post-quantum digital signature algorithms. We use *tcpdump* [33] to capture the TLS & TCP/IP handshake packets and *Wireshark* [2] to collect the packet data. More details about the experimental setup are provided below. We establish the TLS 1.3 connection 1000 times for the experimental samples and run and compare the performances with the RSA (not quantum-resistant) to provide a reference.

Table 4: TLS performance with different digital signature algorithms. The CPU samples have a confidence interval to 95%. The *Rainbow* algorithms were unable to complete the TLS connection due size limitations, and are therefore marked with asterisks (*).

Algorithm	**Time(ms)**	**CPU ±CI**	**CS(B)**	**TSS(B)**	**SHP**	**SHS(B)**	**HP**	**CS/SHS**	**SHP/HP**
Dilithium 2	3.12	23.90% ±2.25%	4700	5875	2	5743	10	0.818	0.2
Dilithium 3	3.13	27.33% ±2.17%	5900	7477	2	7345	10	0.803	0.2
Dilithium 4	3.21	31.29% ±0.51%	7200	9161	3	8963	12	0.803	0.25
Falcon 512	6.91	29.84% ±0.47%	2400	2726	1	2660	8	0.902	0.125
Falcon 1024	11.49	31.43% ±0.61%	4400	4923	2	4791	10	0.918	0.2
Rainbow Ia Cyclic*	7.53	26.04% ±1.39%	204700						
Rainbow Ia Classic*	6.51	31.78% ±0.82%	204700						
Rainbow Vc Cyclic*	75.52	32.45% ±0.73%	2300000						
RSA 2048	3.46	17.16% ±0.58%	1000	1319	1	1253	8	0.798	0.115

Table 4 shows the algorithms, time, CPU usage, Certificate Size (CS), total TCP Segment Size (TSS), number of Server Hello Packets (SHP), Server Hello Size (SHS), and number of Handshake Packets (HP). Time refers to the average handshake time elapsed for a connection, CPU usage represents the highest percentage logged while connection establishment, CS provides the size of the certificate generated using the algorithms listed. We observed that Server Hello with large certificates uses TCP segmentation. SHP represents the number of packets used to transfer the Server Hello message. TSS provides the total data transferred by TCP segments. SHS represents the total Server Hello size in bytes that contains the certificate and TLS extensions. HP represents the total number of packets used to establish the connection (counted to client finished message).

The implementation of *Rainbow* into TLS 1.3 fails due to excessively large certificate sizes, logged using *tcpdump*, that are responsible for overflowing the TCP window and thus causing errors. By default, the X.509 certificate size in TLS 1.3 has a limit of $2^{24} - 1$ Bytes [46]. *Rainbow*'s certificate size exceeds the X.509 limit and therefore causes the TLS 1.3 connection to fail.

Packet-Level Handshake Analysis: Time Overhead This section analyzes the handshake overheads in the unit of average connection time, and processing (CPU utilization). Our results in Table 4 show that *Dilithium 2* outperforms the other quantum-resistant algorithms in average handshake time. The average handshake time for *Falcon 512* is 2.21 times more than *Dilithium 2*. *Dilithium 2* is the most efficient in CPU usage, it is 30.92% more efficient than *Dilithium 4*, and 24.85% more efficient than *Falcon 512*. *Falcon 1024*'s CPU usage is 5.33% more than *Falcon 512*.

Packet-Level Handshake Analysis: Certificate Size and Server Hello This section analyzes the overhead caused by the CS and SHS on the connection handshake. *Falcon 512*'s CS (2400 B) is 0.51 times smaller than *Dilithium 2* (4700 B). *Falcon 512* is the only post-quantum algorithm that transfers its Server Hello message in a single packet and is suitable for devices capable of handling

only small certificate sizes due to small buffer sizes. We compute the fraction that the CS takes up inside the Server Hello with $\frac{\text{CS}}{\text{SHS}}$. This provides the additional extension overhead that each algorithm enforces on the TLS connection. *Falcon 1024* has $\frac{\text{CS}}{\text{SHS}}$ percentage of 92%, causing only 8% overhead. Except for *Falcon 512*, all other post-quantum algorithms use TCP segmentation for their Server Hello message, indicating the additional overhead within the handshake. Fraction $\frac{\text{SHP}}{\text{HP}}$ implicates the effect of post-quantum certificate carrying Server Hello on the handshake. A handshake using *Falcon 512* is composed of 12.5% Server Hello, while the other post-quantum algorithms have a percentage of 20% or more.

PQC Algorithm Choices with TLS Integration From our analysis, *Dilithium 2* has the fastest connection time among the PQC cipher algorithms, and even outperforms RSA in some ways. *Falcon 512* is a better alternative to the *Dilithium* family for its low CS and similar packet overheads to RSA. Multivariate-based algorithms are not suitable for TLS implementations since the large CS's are bigger than the CS limit for TLS 1.3.

8 Takeaways and Discussions

We analyze the PQC ciphers in security and performances in this paper and summarize our choices and recommendations based on the analyses in this section. In security costs, *Falcon 512/1024* incurs the most computational effort against a quantum-equipped cryptanalyst among the lattice-based algorithms in qubits (Section 4.2). The multivariate-based schemes are also compared but in quantum gates. Comparing the finalist scheme of *Rainbow* with the alternate scheme of *GeMSS* according to the security categories defined by NIST, *Rainbow Ia* incurs 2^{12} greater security costs in quantum gates than $GeMSS$ 128; *Rainbow IIIc* incurs 2^{67} greater quantum gates than $GeMSS$ 192; and *Rainbow Vc* 2^{70} greater security costs than $GeMSS$ 256 (Section 4.3). In our inter-scheme comparisons, *Rainbow Vc* has 188% and 187% greater security costs in DW cost against an adversary compared to *Dilithium 4* and *Falcon 1024*, respectively, when quantum annealing is enabled in the universal quantum gate model (Section 5). Our performance analyses focusing on the algorithms only (Section 6) yield that *Dilithium 2* is the quickest for signing for messages shorter than 390 kB and *Rainbow Ia Cyclic* is the quickest for signing messages longer than 390 kB; *Dilithium 2* is the quickest for verifying messages shorter than 460 kB and *Rainbow Ia Classic* for verifying messages longer than 460 kB. When the PQC algorithm is integrated with TLS and TCP/IP (Section 7), *Dilithium 2* is the best both in connection handshake time and in CPU processing, while *Falcon 512* has the shortest certificate size affecting the payload and memory size.

The choice of the PQC digital signature algorithm depends on both the PQC application requirements and the R&D in quantum computing and cryptanalysis. Our PQC recommendations are based on standard practices in networking security. For example, the key exchange occurs more sporadically than the

message communication frequency[5], and therefore we prioritize the recommendations based on the signature length as opposed to the public key length in Section 4.2. Other choices depend on the application domains and properties, including the application-layer message size affecting the efficiency performance comparison in the PQC ciphers in Section 6 and the signing frequency vs. verifying frequency providing different prioritization between the efficiency performances of signing vs. verifying in Section 6. For example, the cryptocurrency blockchain utilizing digital signatures for the transaction integrity would prioritize the verifying efficiency (since, for every signing to generate a transaction, numerous miners would verify the signature/transaction) and has a message input size less than 460 kB (e.g., Bitcoin has the transaction sizes in the order of hundreds to thousands of Bytes), so the performance-focused PQC cipher choice would be *Dilithium 2*. For networking-constrained applications, the algorithm can also be chosen in order to minimize the number of transmitted packets, which depends on the algorithm's certificate size and the networking protocol's packet specification including the field length. For example, if the PQC cipher is used for TLS and TCP/IP, then *Falcon 512* would be the choice as studied in Section 7

Our security analyses depend on the state-of-the-art R&D in quantum computing, including quantum physics (quantum annealing), quantum computing model (universal quantum gate model), and quantum cryptanalysis (the cryptanalysis on lattice-based and multivariate-based schemes as well as Grover's' algorithm). These fields are dynamically evolving in research. In fact, the NIST's finalist selection in advance of standardization is designed to facilitate cryptanalysis on *Dilithium*, *Falcon*, and *Rainbow*. While this paper focuses on the current state of the art, the R&D advancement will affect our future analyses and the PQC algorithm recommendations.

9 Related Work in PQC Analyses

Section 3 describes the background to our research including the post-quantum digital signature algorithms, and Section 2 explains the history of the post-quantum cryptography and NIST's involvement. In this section, we discuss more related work to our research, including the post-quantum algorithm performance studies, cryptanalysis, and security studies.

Post-Quantum Signature Algorithms Over the past few years, the interests in post-quantum cryptography have significantly increased, and various standardization authorities initiated projects to develop new quantum-resistant cryptography [1, 17]. Numerous submissions and candidate designs underwent extensive analyses. Previous research to that end analyzed the security of the PQC digital signature algorithms by themselves, e.g., *Dilithium* [23], *Falcon* [28],

[5] If they are comparable, using one-time pad can be an option for information-theoretic security resistant against (quantum-)computationally capable adversaries.

and *Rainbow* [22]. We discuss this in greater detail as we introduce the post-quantum algorithms in Section 3. Our work provides inter-scheme comparison analyses by incorporating the DW cost (combining qubits and quantum gates) and by introducing a cryptanalyst attacker using Grover's algorithm. Our security analyses also incorporates the state of the art in quantum computing and, more specifically, universal quantum gate and quantum annealing.

Performance analysis in TLS and TCP/IP The authors in [46] provided a performance study on the post-quantum digital signature algorithm candidates of NIST's PQC standardization project. The few selected parameters of seven out of the nine algorithms in the second-round were integrated with TLS 1.3 and analyzed for networking latency for respective algorithms. The Authors proposed a scheme to use different post-quantum algorithms at Certificate Authority (CA) and Intermediate Certificate Authority (ICA) to improve the overall handshake speed and throughput. Our work is comparable to theirs in that it includes the performance analyses when the PQC algorithms are integrated with TLS. However, our work provides the performance analyses with finer granularity at the packet level, enabling richer analyses (e.g., analyzing the required number of packet transmissions to capture the relationship between the certificate size and the protocol's segmentation). Furthermore, we limit our analyses on the NIST finalist schemes for sharper focus and include the security analyses.

Basu et al. conducted a hardware evaluation study on the signature candidates, including Dilithium, in [11]. Out of the three signature algorithms they analyzed, only *Dilithium* advanced to the third-round of NIST's PQC standardization project. Based on implementations on Field Programmable Gate Arrays (FPGA) and Application-Specific Integrated Circuit (ASIC), their analysis recommends the use of *Dilithium* in server implementations with low latency.

Cryptanalysis and Security Analyses Lattice-based cryptanalysis is provided in [20] and [41]. Authors in [20] provided a software toolkit named Sage 9.0, to perform side-channel attacks on lattice-based cryptography. They also proposed a cryptanalysis framework that can take advantage of side information or hints to perform lattice reduction attacks. Their analysis shows a significant cost reduction in performing cryptanalysis that utilizes the hints. In [41], the authors performed cryptanalysis based on skip-addition fault attacks. They made use of the determinism in the signature algorithm and inject a single fault targeting the signing operation. A portion of the secret key was extracted and used by the proposed forgery algorithm to generate signatures. Their analyses included the skip-addition attacks on *Dilithium* and zero-cost mitigation solutions.

10 Conclusion

This paper presents security comparisons and performance analyses of NIST finalist post-quantum digital signature candidate ciphers. In our security comparison, we use a visualization model to analyze the trade-off between the

key/signature size vs. security. We also analyze and compare the security strengths across different schemes (based on lattice-based vs. multivariate-based cipher designs) by building our analyses on the state of the art research (including DW cost, Grover's algorithm, universal quantum gate model, and quantum annealing). Moreover, we analyze the performances of the NIST finalist PQC digital signature schemes for key generation, signing, and verifying signatures. To measure the PQC implementation costs and the overheads in the communication, time, and processing, we also integrate the PQC algorithms with TLS and TCP/IP. Our paper includes discussions and recommendations and intends to facilitate further research in the PQC ciphers/cryptanalysis and aid the standardization in order to better secure the digital systems in the emerging era of quantum computing.

Acknowledgment

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

References

1. ETSI-Standards. https://www.etsi.org, Last accessed 1 Sept 2020
2. Wireshark tool. https://www.wireshark.org, Last accessed 1 Sept 2020
3. Workshops and timeline. https://csrc.nist.gov/projects/post-quantum-cryptography/workshops-and-timeline, Last accessed 1 Sept 2020
4. Alagic, G., Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.K., Miller, C., Moody, D., Peralta, R., et al.: Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology (2019)
5. Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Kelsey, J., Liu, Y.K., Miller, C., Moody, D., Peralta, R., et al.: Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process. NIST, Tech. Rep., July (2020)
6. Alicki, R., Fannes, M., Horodecki, M.: On thermalization in Kitaev's 2D model. Journal of Physics A: Mathematical and Theoretical **42**(6), 065303 (2009)
7. Alicki, R., Horodecki, M., Horodecki, P., Horodecki, R.: On thermal stability of topological qubit in Kitaev's 4D model. Open Systems & Information Dynamics **17**(01), 1–20 (2010)
8. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange—a new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343 (2016)
9. Alsina, D., Latorre, J.I.: Experimental test of Mermin inequalities on a five-qubit quantum computer. Physical Review A **94**(1), 012314 (2016)
10. Amin, M.H.: Consistency of the adiabatic theorem. Physical review letters **102**(22), 220401 (2009)
11. Basu, K., Soni, D., Nabeel, M., Karri, R.: NIST Post-Quantum Cryptography-A Hardware Evaluation Study. IACR Cryptol. ePrint Arch. p. 47 (2019)

12. Bernstein, D.J.: Visualizing size-security tradeoffs for lattice-based encryption. IACR Cryptol. ePrint Arch. p. 655 (2019)
13. Brassard, G., Høyer, P., Tapp, A.: Quantum cryptanalysis of hash and claw-free functions. ACM Sigact News **28**(2), 14–19 (1997)
14. Caelli, W.J., Dawson, E.P., Rea, S.A.: PKI, elliptic curve cryptography, and digital signatures. Computers & Security **18**(1), 47–66 (1999)
15. Casanova, A., Faugere, J.C., Macario-Rat, G., Patarin, J., Perret, L., Ryckeghem, J.: Gemss: A great multivariate short signature. Submission to NIST (2017)
16. Castelvecchi, D.: IBM's quantum cloud computer goes commercial. Nature News **543**(7644), 159 (2017)
17. Chen, L., Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography, vol. 12. US Department of Commerce, National Institute of Standards and Technology (2016)
18. Cheung, D., Høyer, P., Wiebe, N.: Improved error bounds for the adiabatic approximation. Journal of Physics A: Mathematical and Theoretical **44**(41), 415302 (2011)
19. Chuang, I.L., Gershenfeld, N., Kubinec, M.: Experimental implementation of fast quantum searching. Physical review letters **80**(15), 3408 (1998)
20. Dachman-Soled, D., Ducas, L., Gong, H., Rossi, M.: LWE with Side Information: Attacks and Concrete Security Estimation. IACR Cryptol. ePrint Arch. **2020**, 292 (2020)
21. Deutsch, D.E.: Quantum computational networks. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **425**(1868), 73–90 (1989)
22. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: International Conference on Applied Cryptography and Network Security. pp. 164–175. Springer (2005)
23. Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals-dilithium: A lattice-based digital signature scheme. IACR Transactions on Cryptographic Hardware and Embedded Systems pp. 238–268 (2018)
24. Duong, D.H., Tran, H.T.N., et al.: Choosing subfields for LUOV and lifting fields for rainbow. IET Information Security **14**(2), 196–201 (2020)
25. Durr, C., Hoyer, P.: A quantum algorithm for finding the minimum. arXiv preprint quant-ph/9607014 (1996)
26. Finnila, A.B., Gomez, M., Sebenik, C., Stenson, C., Doll, J.D.: Quantum annealing: a new method for minimizing multidimensional functions. Chemical physics letters **219**(5-6), 343–348 (1994)
27. FIPS, P.: 186-4: Federal information processing standards publication. Digital Signature Standard (DSS). Information Technology Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD pp. 20899–8900 (2013)
28. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z.: Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST's post-quantum cryptography standardization process (2018)
29. Gao, Y.L., Chen, X.B., Chen, Y.L., Sun, Y., Niu, X.X., Yang, Y.X.: A secure cryptocurrency scheme based on post-quantum blockchain. IEEE Access **6**, 27205–27213 (2018)
30. Gottesman, D.: Theory of fault-tolerant quantum computation. Physical Review A **57**(1), 127 (1998)
31. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996)

32. Hauke, P., Katzgraber, H.G., Lechner, W., Nishimori, H., Oliver, W.D.: Perspectives of quantum annealing: Methods and implementations. Reports on Progress in Physics **83**(5), 054401 (2020)
33. Jacobson, V., Leres, C., McCanne, S.: TCPDUMP public repository (2003), http://www.tcpdump.org
34. Jaques, S., Schanck, J.M.: Quantum cryptanalysis in the RAM model: Claw-finding attacks on SIKE. In: Annual International Cryptology Conference. pp. 32–61. Springer (2019)
35. Kelly, J.: A preview of Bristlecone, Google's new quantum processor. Google Research Blog **5** (2018)
36. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 206–222. Springer (1999)
37. Kitaev, A.Y.: Fault-tolerant quantum computation by anyons. Annals of Physics **303**(1), 2–30 (2003)
38. Messiah, A.: Quantum Mechanics: Translated [from the French] by J. Potter. North-Holland (1962)
39. Moses, T.: Quantum computing and cryptography. Entrust Inc., January (2009)
40. Nejatollahi, H., Dutt, N., Ray, S., Regazzoni, F., Banerjee, I., Cammarota, R.: Post-quantum lattice-based cryptography implementations: A survey. ACM Computing Surveys (CSUR) **51**(6), 1–41 (2019)
41. Ravi, P., Jhanwar, M.P., Howe, J., Chattopadhyay, A., Bhasin, S.: Exploiting determinism in lattice-based signatures: practical fault attacks on pqm4 implementations of NIST candidates. In: Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security. pp. 427–440 (2019)
42. Regev, O.: Lattice-based cryptography. In: Annual International Cryptology Conference. pp. 131–141. Springer (2006)
43. Rezakhani, A., Kuo, W.J., Hamma, A., Lidar, D., Zanardi, P.: Quantum adiabatic brachistochrone. Physical review letters **103**(8), 080502 (2009)
44. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
45. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)
46. Sikeridis, D., Kampanakis, P., Devetsikiotis, M.: Post-Quantum Authentication in TLS 1.3: A Performance Study. IACR Cryptol. ePrint Arch. p. 71 (2020)
47. Stebila, D., Mosca, M.: Post-quantum key exchange for the internet and the open quantum safe project. In: International Conference on Selected Areas in Cryptography. pp. 14–37. Springer (2016)
48. Vandersypen, L.M., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H., Chuang, I.L.: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance. Nature **414**(6866), 883–887 (2001)
49. Wheatley, M.: D-Wave debuts new 5,000-qubit quantum computer (Sep 2019), https://siliconangle.com/2019/09/24/d-wave-debuts-new-5000-qubit-quantum-computer