Performance Evaluation of TLS 1.3 Handshake on Resource-Constrained Devices Using NIST's Third Round Post-Quantum Key Encapsulation Mechanisms and Digital Signatures

Sultan Sarıbaş
Department of Computer Engineering
Abdullah Gül University
Kayseri, Türkiye
saribassultan@gmail.com

Samet Tonyalı
Department of Software Engineering
Gümüşhane University
Gümüşhane, Türkiye
samet.tonyali@gumushane.edu.tr

2022 7th International Conference on Computer Science and Engineering (UBMK) | 978-1-6654-7010-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/UBMK55850.2022.9919545

***Abstract*— Towards the end of the 20. century, quantum computing came into sight by the main effect of Shor's Algorithm. While this algorithm offers a solution for the factorization problem, which makes it exponentially faster, it alongside becomes a critical threat for the encryption schemes since mostly their security mechanisms rely on the difficulty of integer factorization or discrete log problems. Despite it is not known when advanced quantum computers will show up, yet, once it is developed, most existing cryptography will be rendered useless, which means all existing information security will be vulnerable. Owing to this risky situation, The National Institute of Standards and Technology (NIST) launched post-quantum cryptography (PQC) standardization process for the development of PQC schemes. In this paper, we have chosen three key encapsulation mechanisms and two digital signature algorithms with different parameter sets from the round three submissions. We measured their TLS 1.3 handshake performance using two resource-constrained devices and compared it to that of classical encryption and digital signature schemes. Experiment results showed that post-quantum algorithms come with an extra message overhead while their handshake delay values are promising.**

Keywords*— post-quantum cryptography, shor's algorithm, key encapsulation mechanism, digital signature, openssl, tls, kyber, ntru, saber, dilithium, falcon

I. INTRODUCTION

In the early 90s, especially in academia, it was considered that quantum computers could work faster than classical computers. However, there was not much budget and motivation to build a quantum computer until Peter Shor developed a quantum algorithm called Shor's Algorithm in 1994, which can factor huge numbers in polynomial time (logN) on quantum computers [1].

Prior to this, most public-key cryptography schemes had been thought of secure. However, as Aumasson [2] commented, these schemes rely on either of two basic problems, i.e., integer factorization and discrete logarithm problems. These problems are anchor points of the security of widely used public-key cryptographic encryption schemes. In conjunction with this, when Shor's algorithm was developed, it started to become a threat to the security of these schemes such as RSA, elliptic curve cryptography, and ElGamal since the algorithm can theoretically make it easy to solve the problems which these schemes rely on. Thus, Shor's algorithm and correspondingly quantum computers attracted big attention depending on these significant threats [3].

As Wright [5] noted, the first quantum processor was developed in 1998. It was a 2-qubits Quantum computer that implements a search algorithm called Grover's algorithm developed by Lov Grover. Table I lists several quantum processors that are developed and manufactured at the present time. According to the list, the leading companies in developing quantum processors are IBM, Google, Intel, and Rigetti. At the very moment, IBM has a quantum processor with the highest number of qubits (127 qubits) which is called the IBM Quantum Eagle processor. Moreover, IBM is the first company that offers a cloud-based quantum computing system. In IBM Cloud, there are several smaller processors besides the large ones. Gambetta [6] reported that these smaller processors will lead them to release a 433 qubits Quantum system in 2022, which is called IBM Quantum Osprey. In addition, they are planning to launch a 1,121-qubit IBM Quantum Condor processor in 2023.

To protect Internet communication against quantum attacks, we can utilize some other public-key cryptography schemes that do not rely on the hardness of the integer factorization or discrete logarithm problems. In this manner, this type of cryptography is called post-quantum cryptography (PQC). In other words, PQC is the secure cryptographic system that cybersecurity specialists are competing to develop for being protected from quantum threats, while computer scientists compete to develop the advanced quantum computers. Although these PQC solutions are already being developed by computer science experts, these algorithms need to be standardized. Therefore, the National Institute of Standards and Technology (NIST) took a step toward this purpose by initiating a process.

Consequently, NIST launched a Post-Quantum Cryptography Standardization contest in August 2016. They accepted Public-key Encryption, Key-establishment Algorithms, and Digital Signature Algorithms for this contest until November 30, 2017 [7]. In this paper, three algorithms from Public-key Encryption with a total of ten different parameters and two algorithms from Digital Signatures with a total of five different parameters have been chosen from

978-1-6654-7010-0/22/$31.00 ©2022 IEEE

Round-3 Submissions of the NIST contest to run these public-key encryption algorithms with each digital signature and measure their performance. The digital signature algorithms are Crystals-Dilithium (2,3,5) and Falcon (512,1024). The public-key encryption algorithms are Crystals-Kyber (512,768,1024), NTRU (HPS-2048-509, HPS-2048-677, HPS-4096-821, HRSS-701), and Saber (LightSaber, Saber, Firesaber). These measurements were made by using Wireshark for packet capturing, and two Raspberry Pi devices were used for communication.

TABLE I. SEVERAL ADVANCED QUANTUM PROCESSORS

Manufacturer	Processor Name	Number of Qubits	Released Date
IBM	IBM Quantum Eagle processor	127qb	Nov, 2021 [8]
Google	Bristlecone	72qb	2017 [9]
IBM	IBM Quantum Hummingbird processor	65qb	Sep, 2020 [6]
Google	Sycamore	54qb	2018 [10]
Intel	Tangle Lake	49qb	Jan, 2018 [11]
Rigetti	Aspen-10	32qb	Nov, 2021 [12]
Rigetti	Aspen-9	32qb	Feb, 2021 [12]
IBM	IBM Quantum Falcon processors	27qb	2019 [6]
Google	Foxtail	22qb	2016 [13]
Intel	17-Qubit Superconducting Chip	17qb	Oct, 2017 [14]

The rest of the paper is organized as follows. Section II describes the existing post-quantum cryptography algorithms that are candidates in the NIST contest. In Section III, the performance of these algorithms that are measured for this project will be compared. Finally, in Section IV, the conclusion, and open issues will be covered.

II. BACKGROUND

It is clarified earlier that Shor's algorithm is a significant factor in developing quantum computers and consequently post-quantum cryptography. Stebila and Mosca [15] pointed out that once large-scale quantum computers come into existence, the attackers, who record the present network traffic and communications, will be able to break their security and decrypt them. Therefore, this situation and the concerns about the security mentioned in Section-I generate the motivation for urgently developing post-quantum cryptography, also known as quantum-resistant or quantum-safe cryptography.

However, the large and reliable quantum computers do not exist yet in the present time. An advanced quantum computer that would be able to break the existing security schemes needs thousands or millions of qubits [16]. The most advanced quantum computer has 127 qubits today. According to Aumasson's [2] foresight, “There is no need to panic though – the quantum computers that will break RSA won’t come this year, very likely not this decade and probably not this century”. Nevertheless, it is not known when such a computer can be built. Nowadays, there are remarkable developments and goals in this field. Thus, necessary countermeasures should be taken against this possible threat.

Moreover, the cryptography experts work on a backup plan. It is indicated in this paper that NIST launched a competition in order to standardize post-quantum algorithms. The algorithms that are selected as third-round candidates were announced at the third PQC Standardization Conference held on June 09, 2021. Open Quantum Safe provides these algorithms as an open project on their GitHub repository. Open Quantum Safe (OQS) is an open-source project that aims to support the development of quantum-resistant (post-quantum) cryptography [15].

Ultimately, the OQS-OpenSSL GitHub repository [17] has been cloned to two Raspberry Pi 4 tools supported by Ubuntu operating system. These Raspberry Pi tools have been communicated by using public-key and digital signature algorithms that are chosen from third-round candidates of NIST's competition and each of their performance was measured for this study. These algorithms and their features are as follows:

A. Key Encapsulation Mechanisms

1) Kyber

Kyber is a key encapsulation mechanism (KEM) that has different parameter sets with different security levels. Kyber's security relies on the hardness of the learning-with-errors (LWE) problem. To be more precise, the security level of Kyber-512, Kyber-768, and Kyber-1024 is nearly equivalent to that of AES-128, AES-192, and AES-256, respectively. [18]. The claimed NIST level, the public key size, secret key size, and cipher text size are given in Table II.

TABLE II. KYBER PARAMETER SET AND PERFORMANCE OVERVIEW

Parameter Set	512	768	1024
Claimed NIST level	1	3	5
Public key size (bytes)	800	1184	1568
Secret key size (bytes)	1632	2400	3168
Cipher text size (bytes)	768	1088	1568

2) NTRU

NTRU is a key encapsulation mechanism (KEM) that has different parameter sets with different security levels. It is specified in [19], NTRU is a Hofstein, Pipher, and Silverman's NTRU encryption scheme based KEM, and NTRU’s security relies on the mix of reduction two modulo numbers and polynomial algebra [20]. The claimed NIST level, the public key size, secret key size, and cipher text size are given in detail in Table III.

TABLE III. NTRU PARAMETER SET AND PERFORMANCE OVERVIEW

Parameter Set	HPS-2048-509	HPS-2048-677	HRSS-701	HPS-4096-821
Claimed NIST level	1	3	3	5
Public key size (bytes)	699	930	1138	1230
Secret key size (bytes)	935	1234	1450	1590
Cipher text size (bytes)	699	930	1138	1230

3) Saber

Saber is a key encapsulation mechanism (KEM) that has different parameter sets with different security levels. Saber's security relies on the hardness of the Module Learning With Rounding problem (MLWR). To be more precise, the security level of LightSABER, SABER, and FireSABER is nearly equivalent to that of AES-128, AES-192, and AES-256, respectively [21]. The claimed NIST level, the public key size, secret key size, and cipher text size are given in Table IV.

TABLE IV. SABER PARAMETER SET AND PERFORMANCE OVERVIEW

Parameter Set	LightSaber	Saber	Firesaber
Claimed NIST level	1	3	5
Public key size (bytes)	672	992	1312
Secret key size (bytes)	1568	2304	3040
Cipher text size (bytes)	736	1088	1472

B. Digital Signatures Algorithms

1) Dilithium

Dilithium is a digital signature scheme that has different parameter sets with different security levels. Dilithium's security relies on the hardness of lattice problems over module lattices. Moreover, the Dilithium3 parameter set can achieve more than 128 bits of security against all known quantum and classical attacks [22]. The claimed NIST level, the public key size, secret key size, and cipher-text size are given in Table V.

TABLE V. DILITHIUM PARAMETER SET AND PERFORMANCE OVERVIEW

Parameter Set	2	3	5
Claimed NIST level	1	3	5
Public key size (bytes)	1312	1952	2592
Secret key size (bytes)	2528	4000	4864
Cipher text size (bytes)	2420	3293	4595

2) Falcon

Falcon is a Fast-Fourier lattice-based digital signature scheme that has different parameter sets with different security levels. Falcon's security relies on the hardness of the short integer solution problem (SIS) over NTRU lattices [23]. The claimed NIST level, the public key size, secret key size, and cipher-text size are given in Table VI.

TABLE VI. FALCON PARAMETER SET AND PERFORMANCE OVERVIEW

Parameter Set	512	1024
Claimed NIST level	1	5
Public key size (bytes)	897	1793
Secret key size (bytes)	1281	2305
Cipher text size (bytes)	690	1330

III. PERFORMANCE COMPARISON OF THE ALGORITHMS

A. Experimental Setup

1) Hardware

In our experiments, we used two Raspberry Pi 4 devices with their touch screens and an Ethernet switch. We connected these Raspberry Pi devices to the same local network via Ethernet cables over the switch.

Each Raspberry Pi device has a 64-bit quad-core ARM Cortex-A72 CPU processor with a 1.5GHz clock frequency. They have 4GB RAM and a gigabit Ethernet (gigabit per second) port.

As the network switch device, we used the TP-LINK TL-SF1008D model. This device supports 10/100 Mb/s transmission speed in Half Duplex mode, and 20/200 Mb/s transmission speed in Full-Duplex mode. It has a 2MB buffer size.

2) Software

We installed Ubuntu 20.04 64-bit operating system on each Raspberry Pi device. We cloned the open-quantum-safe/openssl repository which is forked from OpenSSL's official GitHub repository. The open-quantum-safe/openssl repository, which is published on GitHub, has the post-quantum key-exchange and digital signature algorithms different from the OpenSSL project. Also, we used Wireshark software, which is a packet analyzer, for capturing the packets during the OpenSSL key exchange of Raspberry Pi devices. Lastly, we used a Linux terminal for operating the actions such as running the server side, starting the conversation, etc.

The key exchange and digital signature algorithms for classical algorithms used in this study are given in Table VII. From these classical algorithms, prime256v1 is a 256-bit prime field Weierstrass elliptic curve algorithm, secp384r1 (p384) is a 384-bit prime field Weierstrass elliptic curve algorithm, and secp521r1 (p521) is 521-bit prime field Weierstrass elliptic curve algorithm. Among these algorithms prime256v1 belongs to the ANSI X9.62 standard which provides generating verifiably random elliptic curves, whereas secp384r1 and secp521r1 belong to the Elliptic Curve Cryptography (SECG) standard which provides a method for generating verifiably random domain parameters . From digital signature algorithms, RSA-3072 relies on the principle of the prime factorization method, whereas other algorithms rely on the principle of mathematical representation of elliptic curves [24].

TABLE VII. DIGITAL SIGNATURE AND KEY EXCHANGE ALGORITHMS USED IN CLASSICAL ALGORITHMS TEST CASES

	Key Exchange Algorithms	Digital Signature
Case-1	prime256v1	RSA-3072
Case-2	secp384r1	p384
Case-3	secp521r1	p521

B. Performance Metrics

For this study, we will be using three performance metrics which are handshake delay, transferred data size, and throughput value. Handshake delay is the metric that we called the execution time of handshaking between server and client. Transferred data size is the total data size that is transferred during handshaking. Lastly, the throughput values were calculated by dividing the total transmitted data size (megabits) by handshake delay (seconds). This calculation was made by converting the data size values to megabits, and ultimately getting Mbps(megabit/second) values.

C. Results and Discussion

In this work, post-quantum algorithms, and classical algorithms were compared in terms of transferred data size, handshake delay, and throughput. Each key encapsulation mechanism (KEM) algorithm was executed with each digital signature algorithm for this study. To form a baseline for the comparison activities, we measured the performance of three pairs of key exchange and digital signature algorithms as shown in Table VII.

The calculated throughput values, handshake delay, and transferred data size metrics will be compared for each case. It is clearly observed that the lower throughput values and the lower handshake delay mean better performance since when throughput or handshake delay is low, it will provide accommodation for other network traffic.

Fig. 1, 2, and 3 show the handshake delay, the transferred data size, and the throughput values, respectively, for both post-quantum and classical algorithms. In these figures, post-quantum algorithms are grouped by KEM algorithms. The values in the x-axis are the KEM algorithms whereas the values indicated in the series with colors show the digital signature algorithms. Additionally, only Case-1, Case-2, and Case-3 values are the classical algorithm cases.

As depicted in Fig. 1, the handshake delay is generally more in classical algorithms than in post-quantum algorithms. All post-quantum algorithm cases have a handshaking delay smaller than 0.05 seconds, only except for the Falcon digital signatures cases, whereas all classical algorithm cases have a handshake delay larger than 0.06 seconds. This situation basically shows the feasibility of post-quantum algorithms in terms of running time. Moreover, Fig. 1 also shows that Dilithium-2 generally works faster than the other digital signature algorithms, which is followed by Dilithium-3, Dilithium-5, Falcon-512, and Falcon-1024, respectively. Furthermore, the pair of Dilithium-2 and lightsaber is the fastest working case among all other post-quantum algorithm cases.

Even though the handshake delay is generally lower in post-quantum algorithms, Fig. 2 shows that transferred data size is mostly larger in post-quantum algorithms than in classical algorithms. The transferred data size is between 3 and 8 kilobytes in the classical algorithm. Conversely, the transferred data size ranges from 6 to 15 kilobytes in post-quantum algorithms. It is observed that the pair of Dilithium-2 and ntru_hrss701 has the largest transferred data size with more than 14 kilobytes. In contrast, the pair of Dilithium-5 and lightsaber has the smallest transferred data size with around 6 kilobytes. Also, generally, Dilithium-5 has lower transferred data size values as shown in Fig. 2.

It can be clearly seen in Fig. 3 that the classical algorithms produce far less throughput than the post-quantum algorithms do. All the classical algorithm cases produce a throughput of lower than 1Mbps. Also, Case-3 outperforms the other two cases where the throughput produced is only 0.3 Mbps. However, the post-quantum algorithm pairs produce a throughput of more than 1Mbps or just around 1Mbps. Particularly, the Falcon-1024 digital signature cases show the successful development of post-quantum algorithms. The performance of these cases is too close to that of the classical algorithm cases. Among all the post-quantum algorithms, the pair of Falcon-1024 and ntru_hps2048509 has the best performance, which is about 0.91 Mbps. In contrast, the pair of Dilithium-2 and ntru_hps4096821 has the worst performance which is about 6.8 Mbps. It seems it is hard to determine which KEM algorithm performs better since it varies with the digital signature. Still, it is obvious that Falcon-1024 has the best performance among the digital signature algorithms tested.

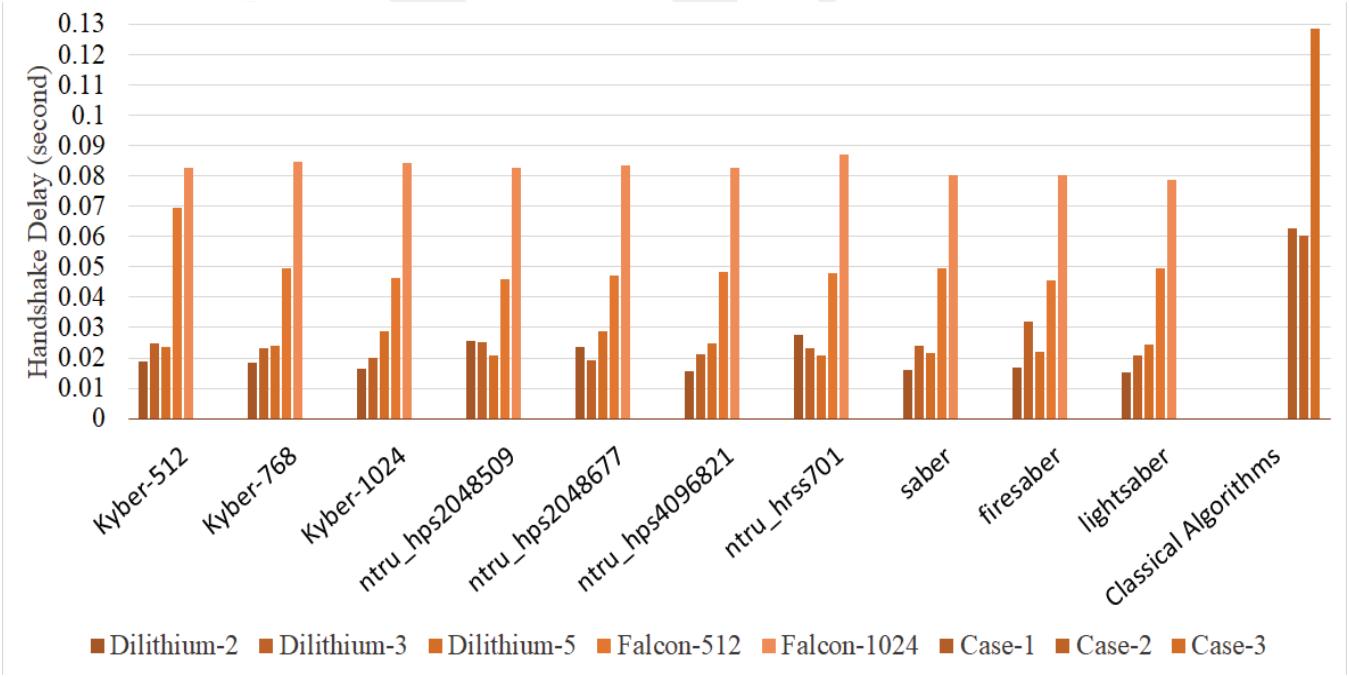

Fig. 1. Handshake Delay Values for Both Post-Quantum (Grouped by KEM Algorithms) and Classical Algorithms

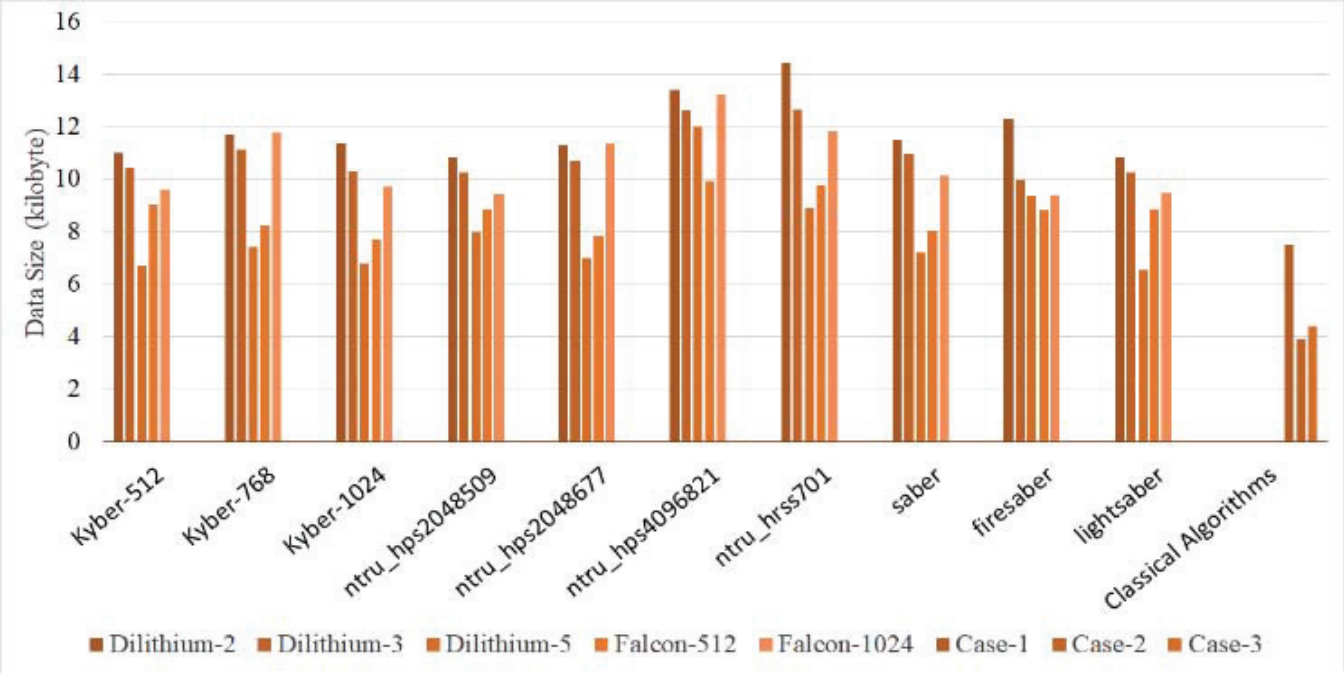

Fig. 2. Transferred Data Size for Both Post-Quantum (grouped by KEM algorithms) and Classical Algorithms

IV. Conclusion and Future Work

In this paper, we explained the threats posed by Shor's algorithm to existing public-key encryption schemes in the case of the development of a scaled quantum processor. Also, we measured the handshake delay, the throughput, and the transferred data size of NIST's third round post-quantum key encapsulation mechanisms and digital signatures and compared their performance to that of classical encryption algorithms.

As a result, post-quantum cryptography is not quite better than classical cryptography algorithms, yet, according to their throughput values, but still there is a successful development, especially in terms of handshake delay. Through experiments, it is shown that post-quantum algorithms are promising encryption schemes. However, they should be improved and widely accepted before large-scale quantum computers arrive, to prevent the threat of breaking existing encryption schemes.

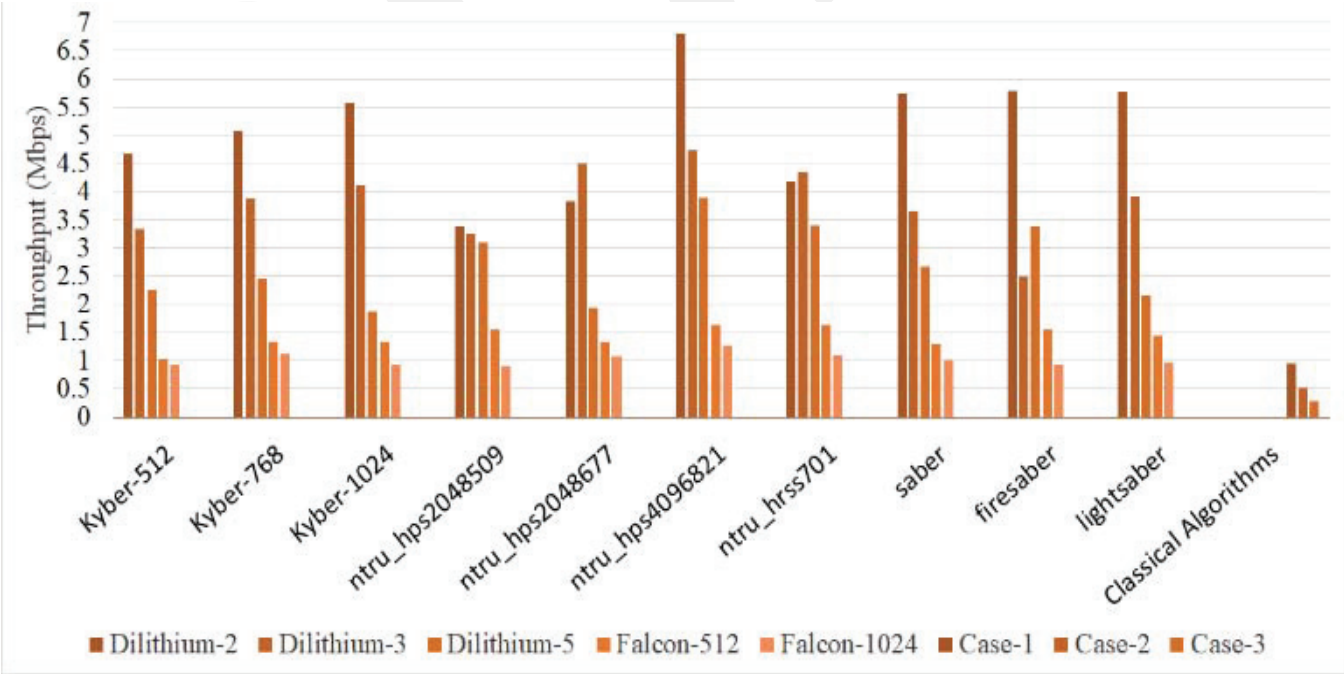

Fig. 3. Throughput Values for Both Post-Quantum (grouped by KEM Algorithms) and Classical Algorithms

REFERENCES

[1] M. Hayward, "Quantum Computing and Shor's Algorithm," *Herokuapp.com*, 2015. https://quantum-algorithms.herokuapp.com/299/paper/index.html

[2] J.-P. Aumasson, "The impact of quantum computing on cryptography," *Computer Fraud & Security*, vol. 2017, no. 6, pp. 8–11, Jun. 2017, doi: 10.1016/s1361-3723(17)30051-9.

[3] "Shor's factoring algorithm | Quantiki," *Quantiki.org*, 2015. https://www.quantiki.org/wiki/shors-factoring-algorithm

[4] V. Bhatia and K. R. Ramkumar, "An Efficient Quantum Computing Technique for Cracking RSA Using Shor's Algorithm," *2020 IEEE 5th International Conference on Computing Communication Automation (ICCCA)*, Oct. 2020, 10.1109/iccca49541.2020.9250806. doi:

[5] M. A. Wright, "The Impact of Quantum Computing on Cryptography," *Network Security*, vol. 2000, no. 9, pp. 13–15, Sep. 2000, doi: 10.1016/s1353-4858(00)09027-9.

[6] J. Gambetta, "IBM's roadmap for scaling quantum technology | IBM Research Blog," *IBM Research Blog*, Sep. 15, 2020. https://research.ibm.com/blog/ibm-quantum-roadmap

[7] NIST, "Post-Quantum Cryptography | CSRC," *Nist.gov*, 2017. https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions

[8] J. Chow, O. Dial, and J. Gambetta, "IBM Quantum breaks the 100-qubit processor barrier | IBM Research Blog," *IBM Research Blog*, Nov. 16, 2021. https://research.ibm.com/blog/127-qubit-quantum-processor-eagle

[9] Google AI Blog, "A Preview of Bristlecone, Google's New Quantum Processor," *Google AI Blog*, Mar. 05, 2018. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html

[10] Google AI Blog, "Quantum Supremacy Using a Programmable Superconducting Processor," *Google AI Blog*, Oct. 23, 2019. https://ai.googleblog.com/2019/10/quantum-supremacy-using-programmable.html

[11] Intel, "Intel Newsroom," *Intel Newsroom*, Jan. 08, 2018. https://newsroom.intel.com/news/intel-advances-quantum-neuromorphic-computing-research/#gs.ucrwjo

[12] Rigetti, "Building scalable, innovative quantum systems," *Rigetti Computing*, 2017. https://www.rigetti.com/what-we-build

[13] K. Kissel, "An Update on Google's Quantum Computing Initiative," Nov. 2018. Accessed: Mar. 19, 2022. [Online]. Available: https://indico.cern.ch/event/719844/contributions/3058028/attachments/1746692/2828167/CERN_2018_Quantum_Talk_Kissell.pdf

[14] Intel, "Intel Newsroom," *Intel Newsroom*, Oct. 10, 2017. https://newsroom.intel.com/news/intel-delivers-17-qubit-superconducting-chip-advanced-packaging-qutech/

[15 Stebila and M. Mosca, "Post-quantum Key Exchange for the Internet and the Open Quantum Safe Project," in *Selected Areas in Cryptography–SAC 2016*, Jul. 2017, vol. 10532, pp. 14–37. doi: 10.1007/978-3-319-69453-5 2.

[16] M. Mosca, "Cybersecurity in an era with quantum computers: will we be ready?," *Cryptology ePrint Archive*, 2015. https://eprint.iacr.org/2015/1075

[17] open-quantum-safe, "open-quantum-safe/openssl: Fork of OpenSSL that includes prototype quantum-resistant algorithms and ciphersuites based on liboqs," *GitHub*, Jun. 23, 2022. https://github.com/open-quantum-safe/openssl

[18] Crystals, "Kyber," *Pq-crystals.org*, Dec. 2020. https://pq-crystals.org/kyber/index.shtml

[19] C. Chen *et al.*, "NTRU Algorithm Specications And Supporting Documentation," 2019. [Online]. Available: https://ntru.org/f/ntru-20190330.pdf

[20] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," *Algorithmic Number Theory - ANTS-III*, vol. 1423, pp. 267–288, 1998, doi: 10.1007/bfb0054868.

[21] Saber, "SABER: MLWR-based KEM," *Kuleuven.be*, 2019. https://www.esat.kuleuven.be/cosic/pqcrypto/saber/index.html

[22] Crystals, "Dilithium," *Pq-crystals.org*, Feb. 16, 2021. https://pq-crystals.org/dilithium/index.shtml

[23] Falcon, "Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU," *falcon-sign.info*, 2017. https://falcon-sign.info/

[24] J. Jancar and V. Sedlacek, "Standart Curve Database," *Neuromancer.sk*, 2020. https://neuromancer.sk/std/