QUIC Protocol with Post-Quantum Authentication

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

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

Abstract. Post-quantum ciphers (PQC) are designed to replace the current public-key ciphers which are vulnerable against the quantum-equipped adversaries, e.g., RSA. We study the incorporation of the PQC algorithms into the QUIC and TCP/TLS networking protocols and analyze the performances and overheads in authentication and connection establishment. To distinguish from previous research, we focus on the newer QUIC networking protocol while comparing it with TCP/TLS. The QUIC protocol builds on UDP and its superiority over TCP/TLS is highlighted by the quicker and lower-overhead connection establishments. QUIC is thus gaining wider deployment, including its planned standardization for HTTP/3. We implement and experiment in local networking environment which provides greater analyzability and control. We compare QUIC vs. TCP/TLS when using PQC and measure the handshake overhead in time duration while varying both the PQC security strength and the networking conditions. Our results show that the PQC overhead increases with the PQC cipher security strength (the key and signature sizes) and as the network condition worsens (greater occurrences of packet dropping). Comparing between the PQC and the classical cipher with comparable security strengths, the PQC ciphers outperform RSA in the handshake time duration; both Dilithium 2 and Falcon 512 handshakes are quicker than RSA 3072.

Keywords: QUIC, TCP, TLS, Post-Quantum Cryptography, Digital Signatures

1 Introduction

Since the initialization and standardization of Hypertext Transfer Protocol (HTTP), the internet has seen a rising amount of web traffic over the years, driving the need for scalability and optimization. Over 70% of internet traffic [3] and 60% of internet connections [17] are secure HTTP (version 1 or 2) using Transmission Control Protocol with Transportation Layer Security (TCP/TLS) for transport layer communication. TCP/TLS's head-of-the-line blocking where a packet drop in a stream blocks all other streams limits its capabilities. Quick UDP Internet

Connections (QUIC) transport protocol designed by Google [12] and recently standardized by Internet Engineering Task Force (IETF) [10] is gaining popularity for removing head-of-the-line blocking as well as adding a plethora of new features for scalability and optimization. Google's implementations show that QUIC has 8% faster website search responses and 18% reduced re-buffer rates for YouTube over TCP/TLS [12]. QUIC carries more than 7% of internet traffic and is replacing TCP/TLS across major applications [12]. Works are in progress to standardize and replace TCP/TLS with QUIC as the primary transport for upcoming HTTP Version 3 (HTTP/3) [6]. Once HTTP/3 is standardized, majority of internet traffic (≈70%) will be transported using QUIC. QUIC provides confidentiality and integrity within the authentication scheme for connections to ensure the security of the data packets.

Cryptographic ciphers are widely used in networking protocols. For example, well-known protocols like TLS [15], SSH [5], IPsec [9], etc, use digital signature cipher algorithms like Rivest-Shamir-Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) with X.509 certificates for authentication of end-devices. Security of the protocols using RSA relies on integer factorization problem and ECDSA on discrete logarithm problem.

Recent advancements in quantum computing and Shor's algorithm (capable of solving integer factorization and discrete logarithm problem in polynomial time assuming quantum computer) cause a need to design and develop new post-quantum ciphers (PQC). National Institute of Science and Technology (NIST) launched a PQC standardization project [2], in December 2016, to identify and standardize cipher algorithms that can withstand the growing quantum threats. In August 2022, NIST PQC standardization project finished its third round [4] and selected algorithms for standardization. NIST selected the digital signature algorithms which are lattice-based (Dilithium and Falcon) and hash-based (SPHINCS$^+$). In our paper, we focus on the lattice-based schemes of Dilithium and Falcon as opposed to the hash-based algorithm of SPHINCS$^+$ because SPHINCS$^+$ produces long signatures which can challenge its deployment to many applications. To defend against the future quantum adversaries to protect the authenticity, the networking protocols (QUIC, TCP/TLS, SSH, IPsec) should transition from the classical ciphers to the PQC ciphers.

2 Background of QUIC and NIST PQC

2.1 QUIC Protocol

In contrast to TCP/TLS which has clear distinction between the OSI layers by design, the QUIC networking protocol has multi-layer connection that combines application and transport layers. QUIC builds on the faster UDP protocol. To add reliability to UDP, QUIC utilizes data fields for the connection state/identification in the application layer in the OSI model and a combination of cryptographic and transport-layer handshakes in the transport layer. QUIC's multi-layer connection helps in combining and negotiating both cryptographic

and transport parameters during a handshake. In addition to the above differences with TCP/TLS, QUIC supports and requires clients in addition to servers to use error codes in application protocol negotiation failures. QUIC does not use TLS end-of-early data messages to signal key changes and it also does not need TLS middle-box compatibility mode that adds 32 byte legacy session id value in client and server hello messages.

2.2 NIST PQC Ciphers

We focus on the NIST standardization PQC cipher algorithms due to NIST's strong influence in standardizing cipher algorithms which impacts their future use in digital security, as demonstrated by the DES standardization in the 1970s and the AES standardization in the 1990s (popularly used globally in our current days). NIST PQC standardization project finished its third round, and the selected lattice-based digital signature algorithms are Dilithium and Falcon.

Dilithium Crystals Dilithium uses "Fiat-Shamir with Aborts" approach, SHAKE or AES for its hashing algorithm. Dilithium introduces Dilithium 2, Dilithium 3 and Dilithium 5 which correspond to NIST post-quantum security levels 2, 3, and 5, respectively [7].

Falcon Falcon stands for the acronym, Fast Fourier lattice-based compact signature scheme over a N-th Degree Truncated Polynomial Ring (NTRU). Falcon's security scheme is based on Gentry, Peikert and Vaikuntanathan (GPV), NTRU lattices, Fast Fourier sampling. Falcon introduces Falcon 512 and Falcon 1024 which correspond to NIST post-quantum security level 1 and 5 respectively [8].

3 Performance Analysis

In this section, First, we provide details of our experimentation and implementation in Section 3.1. Second, in Section 3.2, we analyze the TCP/TLS and QUIC connection establishment overheads induced by the PQC (Dilithium and Falcon) under no artificial network drop (D=0). Later, in Subsection 3.3, we analyze the behavior of both TCP/TLS and QUIC connections under different artificial packet dropping ($D \neq 0$) scenarios.

3.1 Implementation and Experimentation

We compare the NIST selected PQC ciphers described in Section 2.2 with the classical cipher RSA, which cipher is selected based on its popularity and since it supports digital signatures (it also supports key exchange and encryption for confidentiality). We implement the ciphers using Open-Quantum-Safe OpenSSL 1.1.1 [18]. We focus on our experiments based on the server and the client implementations on virtual machines (VM) on a single physical machine in this paper due to the following two reasons. First, the VM-based implementation enables a sharper focus on the comparative analyses of QUIC vs. TCP/TLS and excludes the other noise/random factors, such as the networking latency

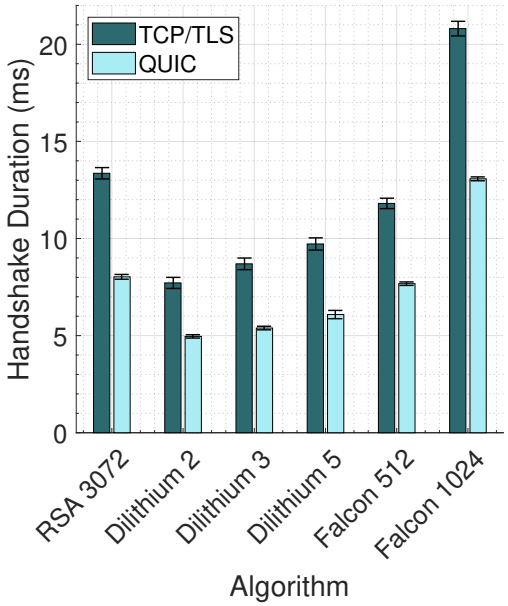


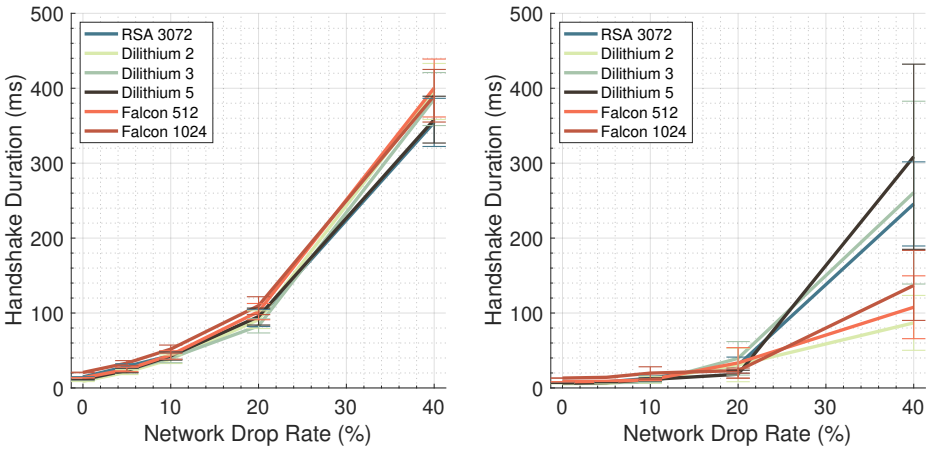
Fig. 1: The handshake duration while varying the PQC algorithms. The plot includes average values and the 95% confidence intervals.

variations between the local client and the remote server. Second, it enables the networking control between two machines, including enabling the networking condition simulation such as varying network drop rates and network delays.

Our virtual machine setup uses Ubuntu 18.04 with each VM containing 8 cores and 16 GB of RAM on an AMD Ryzen 9 3960x 24-core 48-thread processor with a base processor frequency of 3.8 GHz, and 64 GB of RAM. For the network control and simulation, we use the traffic control (*tc*) queuing discipline (*qdisc*) network emulator (*netem*) that selectively controls packets to be en-queued and modified when sent and received from the client machine. For networking, we implement the TCP/TLS 1.3 with Open-Quantum-Safe BoringSSL [18] and QUIC with LiteSpeed QUIC (*lsquic*) [1]. These protocols require multiple machines and, more specifically, a client to initiate the handshakes and a server to respond to TLS/QUIC connection requests. We run our experiments to establish 1,000 TCP/TLS and QUIC connections authenticated using PQC algorithms as well as classical RSA 3072. We use Python 3.9.7 for experimental automation and use *tcpdump* for packet capturing.

3.2 QUIC and TCP/TLS Performance

Experimental Design Transitioning to PQC authentication impact the handshake duration between the client and the server. We test the performance of both TCP/TLS and QUIC connections and analyze their overheads with PQC authentication.



(a) TCP/TLS packet dropping (b) QUIC packet dropping

Fig. 2: The handshake time duration while varying the packet drop rate D.

Experimental Results Fig. 1 plots the handshake duration for TCP/TLS and QUIC connections authenticated by classical RSA and NIST selected digital signatures for standardization. Our results show that, except for Falcon 1024, transitioning to PQC speeds up the handshake duration and connection establishment of both TCP/TLS and QUIC protocols. At comparable security of level 1, QUIC authentication with Dilithium 2 and Falcon 512 is 39.48% and 6.07% faster compared to RSA while it is 42.24% and 11.61% faster for TCP/TLS, respectively. Using Dilithium for authentication, QUIC is at least 35.40% faster than TCP/TLS. Using Falcon 1024 increases the handshake duration of QUIC by 69.46% and TCP/TLS by 76.27% compared to Falcon 512. Using Falcon 512 and Falcon 1024, QUIC is 34.49% and 37.01% faster when compared to TCP/TLS, respectively. Overall, Dilithium algorithms are most efficient and cause low handshake duration compared to RSA and Falcon in both QUIC and TCP/TLS protocols.

3.3 Performance with Packet Dropping (D)

Experimental Design QUIC is designed to overcome TCP/TLS head of the line problem as discussed in Section 2. Our experiment targets to test the performance of QUIC when the network is lossy where not all the packets sent from the sender don't reach the receiver. We conduct lossy network experiment varying the droprate, $D \in \langle 0, 5, 10, 20, 40 \rangle$, which is the percentage of packet dropped by the network in uniform distribution. For example, when $D = 10$ network drops 10% of the packets during the connection.

Experimental Results Fig. 2a and Fig. 2b plots the average TCP/TLS and QUIC handshake times under varying network drop rates. As the drop rate increases, QUIC connection overheads increases but still are lower than TCP/TLS

overheads with both the RSA and post-quantum algorithms for authentication. Using classical algorithm RSA, QUIC consistently outperforms TCP/TLS by 69% with $D = 5$ (i.e. dropping 5% traffic), 71.2% with $D = 10$, 65.7% with $D = 20$, and 31.3% with $D = 40$.

Using post-quantum authentication with Dilithium 2, QUIC is faster by 73.2% with $D = 5$, 70.8% with $D = 10$, 67.4% with $D = 20$, and 78.5% with $D = 40$ than TCP/TLS connections. When using post-quantum authentication with Dilithium 5, QUIC consistently outperforms TCP/TLS by 69.1% when $D = 5$ (0.1% improvement from RSA), 75.2% when $D = 10$, 81.6% when $D = 20$, and 16.3% when $D = 40$ (15% improvement from RSA). When using post-quantum authentication with Falcon 512, QUIC is 66.9% with $D = 5$, 75.8% with $D = 10$, 67.6% with $D = 20$, and 73.5% with $D = 40$ faster than TCP/TLS connections. QUIC remains the top performer throughout the varying D and is preferable over TCP/TLS.

4 Related Work

Related to our research are previous research works comparing QUIC vs. TCP/TLS and incorporating PQC on those protocols. Previous research compared the transport capabilities of QUIC and TCP protocols under different networking scenarios [19,16]. Yu et al. in [19] conducted an experimental study to evaluate the performance of QUIC and TCP protocols when competing for resources to deliver the application data. Their study shows that QUIC performs better than TCP in lossy networks and has no major advantage in loss-free networks. Seufert et al. [16] conducted a study to explore the application-level Quality of Experience (QoE) benefits of QUIC over TCP. Their works conclude that there are no QoE benefits from QUIC than TCP with respect to web browsing or video streaming unless there are bandwidth limitations. Our work studies the PQC integration on these networking protocols.

Other relevant research investigated the performance of the TCP/TLS using post-quantum authentication [11,17,14,13]. Kampanakis et al. in [11] investigated the viability of using post-quantum certificates in protocols including TLS and QUIC. They emulated large certificates by generating the certificates matching the sizes of the RSA keys (8192 and 16384 bits) and merging multiple certificates for certificate chains, as opposed to actually implementing the classical and PQC ciphers. They tested the protocol capabilities in handling such huge certificates chains up to 135 KB. Their emulation results show that TLS and QUIC can handle huge post-quantum certificates with minor implementation modifications. Sikeridis et al. [17] studied the throughput performance of TLS 1.3 when using post-quantum algorithms. Their results show that transitioning to post-quantum authentication in TLS induces latency overhead compared to classical algorithms. Our work implements the PQC ciphers in software and includes the analyses for both QUIC and TCP/TLS, including the comparisons between the two protocols. Our work also shows that the PQC ciphers in Dilithium and Falcon have smaller overheads in time duration than RSA when

using security strength level 1 (Dilithium 2 vs. RSA-3072 and Falcon-512 vs. RSA 3072 in Section 3.2).

Our current work builds on our previous works of individual PQC algorithm performances [14] and PQC performance when integrated with PKI [13]. However, this paper focuses on understanding the behavior of QUIC when integrated with PQC authentication and compares its performance to that of TCP/TLS.

5 Conclusion

This paper analyzes the overheads of post-quantum authentication and connection establishments in the QUIC networking protocol. We compare the performance of QUIC and TCP/TLS in handshake duration times. We implement the protocols and the algorithms and analyze the behaviors under local environment that enables the control in networking, including the packet loss. Our implementation-based experimental results show that the connection overhead in handshake and PQC-based authentication increase with the cipher's security strength and with the deteriorating networking conditions. Our analyses results show that the PQC overheads in the handshake duration increases with the PQC cipher security strength (longer key and signature sizes) and as the network connection worsens (greater occurrences of packet dropping). The PQC ciphers also outperform RSA in the handshake time duration; both Dilithium 2 and Falcon 512 handshake is quicker than RSA-3072 while all of these algorithms are comparable in its security strength (security level 1).

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 1922410 and by a grant from the U.S. Civilian Research & Development Foundation (CRDF Global).

References

1. lsquic github. `https://github.com/litespeedtech/lsquic`, Last accessed 18 Apr 2022
2. Nist-call for proposals. `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf`, Last accessed 18 Apr 2022
3. Percentage of https (tls) encrypted traffic on the internet ? `https://etherealmind.com/percentage-of-https-tls-encrypted-traffic-on-the-internet/`, Last accessed 10 Nov 2021
4. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., Lichtinger, J., Miller, C., Moody, D., Peralta, R., et al.: Status report on the third round of the nist post-quantum cryptography standardization process. US Department of Commerce, NIST (2022)

5. denis bider: Use of RSA Keys with SHA-256 and SHA-512 in the Secure Shell (SSH) Protocol. RFC 8332 (Mar 2018). https://doi.org/10.17487/RFC8332, https://www.rfc-editor.org/info/rfc8332
6. Bishop, M.: Hypertext Transfer Protocol Version 3 (HTTP/3). Internet-Draft draft-ietf-quic-http-34, Internet Engineering Task Force (Feb 2021), https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34, work in Progress
7. 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)
8. 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)
9. Frankel, S., Krishnan, S.: IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap. RFC 6071 (Feb 2011). https://doi.org/10.17487/RFC6071, https://www.rfc-editor.org/info/rfc6071
10. Iyengar, J., Thomson, M.: QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000 (May 2021). https://doi.org/10.17487/RFC9000, https://rfc-editor.org/rfc/rfc9000.txt
11. Kampanakis, P., Panburana, P., Daw, E., Van Geest, D.: The viability of post-quantum x. 509 certificates. IACR Cryptol. ePrint Arch. **2018**, 63 (2018)
12. Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., Yang, F., Kouranov, F., Swett, I., Iyengar, J., et al.: The quic transport protocol: Design and internet-scale deployment. In: Proceedings of the conference of the ACM special interest group on data communication. pp. 183–196 (2017)
13. Raavi, M., Chandramouli, P., Wuthier, S., Zhou, X., Chang, S.Y.: Performance characterization of post-quantum digital certificates. In: 2021 International Conference on Computer Communications and Networks (ICCCN). pp. 1–9. IEEE (2021)
14. Raavi, M., Wuthier, S., Chandramouli, P., Balytskyi, Y., Zhou, X., Chang, S.Y.: Security comparisons and performance analyses of post-quantum signature algorithms. In: International Conference on Applied Cryptography and Network Security. pp. 424–447. Springer (2021)
15. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018). https://doi.org/10.17487/RFC8446, https://www.rfc-editor.org/info/rfc8446
16. Seufert, M., Schatz, R., Wehner, N., Gardlo, B., Casas, P.: Is quic becoming the new tcp? on the potential impact of a new protocol on networked multimedia qoe. In: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX). pp. 1–6. IEEE (2019)
17. Sikeridis, D., Kampanakis, P., Devetsikiotis, M.: Post-quantum authentication in tls 1.3: A performance study. IACR Cryptol. ePrint Arch. **2020**, 71 (2020)
18. 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)
19. Yu, Y., Xu, M., Yang, Y.: When quic meets tcp: An experimental study. In: 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC). pp. 1–8. IEEE (2017)