Migrating Applications to Post-Quantum Cryptography: Beyond Algorithm Replacement

Alexandre Augusto Giron [a]
[1]*Informatics and Statistics Department*
Federal University of Santa Catarina (UFSC), Florianópolis-SC, Brazil
[2]*Federal University of Technology – Parana (UTFPR), Brazil*
alexandregiron@utfpr.edu.br

Keywords: Post-Quantum Cryptography (PQC), Hybrid PQC, Network Security.

Abstract: Post-Quantum Cryptography (PQC) defines cryptographic algorithms designed to resist the advent of the quantum computer. Most public-key cryptosystems today are vulnerable to quantum attackers, so a global-scale transition to PQC is expected. As a result, several entities foment efforts in PQC standardization, research, development, creation of Work Groups (WGs), and issuing adoption recommendations. However, there is a long road to broad PQC adoption in practice. This position paper motivates ongoing and future research on this topic. It describes why migrating to PQC is necessary and gathers evidence that the "hybrid mode" can help the migration process. Finally, it stresses that there are risks yet to be considered by the literature. Quantum-safe protocols are being evaluated, but more attention (and awareness) is needed for the software and protocols at the application layer. Lastly, this position paper gives further recommendations for a smother PQC migration.

1 INTRODUCTION

The Internet Society (IS) (Society, 2023) is a global initiative that often expresses the benefits of an open, accessible, cryptographically secure Internet. IS strives against opposition trying to weaken cryptography mechanisms on the Internet. Weakening the most effective protection mechanism would leave applications and systems vulnerable to an adverse scenario, such as mass surveillance. Therefore, web applications exchanging data through the Internet require a robust security protocol. Otherwise, they are susceptible to eavesdroppers, depending on how the applications implement cryptography mechanisms.

Internet infrastructure is composed of numerous types of geographically dispersed equipment, including adaptors, switches, and routers. This infrastructure allows attackers to store, inspect, capture, and manipulate transmission data. Strong cryptography in network protocols prevents attackers from disclosing and modifying transmitted data. Although strong cryptography is not permitted everywhere (Partners Digital, 2023), fortunately, most cryptography schemes today give Internet users sufficient security guarantees.

However, since Shor's algorithm (Shor, 1994), widely used Public-Key Cryptosystems are vulnerable to the Cryptographically Relevant Quantum Computer (CRQC) (Mosca and Piani, 2020). In the somewhat-near future, experts predict the CRQC's capability of breaking current cryptography schemes. As a result, vulnerable schemes leave the Internet insecure against such a quantum attacker. Regarding the attacker's capabilities, Bindel et al. (Bindel et al., 2019) define the *record-now-decrypt-later*, in which the attack starts today (or it is already started) by secretly capturing data in transit and storing it for decryption when a CRQC is available. Such a threat is worrisome and limits confidentiality on the Internet.

Researchers started addressing this issue with the so-called Post-Quantum Cryptography (PQC) (Bernstein and Lange, 2017). Also called quantum-safe or quantum-resistant, PQC is built with mathematical problems with no known solution by both quantum and classical computation. The purpose of PQC is to protect users of today's (classical) computers against attackers with quantum algorithm capabilities. Therefore, PQC enables solving the quantum threat by replacing vulnerable algorithms, thus protecting users even before the CRQC arrives.

[a] https://orcid.org/0000-0001-7668-7505

Although it looks like a simple substitution, the PQC migration is considered to be complex, non-trivial and time-consuming (Kampanakis and Lepoint, 2023; Joseph et al., 2022). The main reasons include the widespread usage of public-key cryptography, complex characteristics of Public-Key Infrastructures (PKIs), Hardware support requirements, compliance with regulations, and, more specifically, the confidence in the security of PQC schemes. PQC schemes do not have the same scrutiny and study level as classical schemes. In the same comparison, PQC schemes can significantly increase byte cost requirements.

At the time of this writing, several research efforts address the PQC migration challenges by evaluating it in network protocols and hardware platforms (Paquin et al., 2020; Sikeridis et al., 2020); and creating new strategies to accommodate PQC better (Schwabe et al., 2020). The NIST PQC standardization process is considered a leading effort, with plans to give PQC standards by 2024 (NIST, 2016). In addition, Working Groups (WGs) were created to study PQC in different Internet-related protocols, such as Transport Layer Security (TLS) and Certificate Management Protocol (CMP). However, as time passes, PQC migration needs additional attention and increased urgency. While the migration urgency is increased due to the *record-now-decrypt-later* threat, the time required to change several network protocols and implementations also contributes to this urgency.

Avoiding abrupt changes is ideal since confidence in PQC security has yet to be fully established. Therefore, experts recommend the "hybrid mode" for the PQC migration, where classical cryptography schemes are combined with PQC (Bindel et al., 2019). This combination is performed to keep security as long as one of the combined parts is secure. Using hybrids as the PQC migration strategy gives more time to assess PQC security and performance impacts before replacing classical schemes.

In this context, this work discusses about PQC migration strategies, including the hybrid mode, emphasizing the challenges and research gaps for PQC adoption. The contributions of this paper are:

- it emphasizes why carefully adopting PQC is necessary, discussing quantum threats and known hybrid mode strategies;
- it discusses challenges for further research, considering different PQC adoption approaches for applications;
- it gives insight about the lack of PQC awareness in application-layer protocols and applications, showing that, otherwise, the migration strategy can fail to mitigate quantum threats; and
- it gives takeaways for readers with PQC adoption recommendations, inviting further engagement regarding quantum threat awareness.

The text is organized as follows. Section 2 discusses why PQC migration is needed, motivated by quantum threats. Section 3 shows the recommended strategies for migration, its advantages and research challenges. Section 4 argues about additional risks in applications yet to be fully considered by the literature. Lastly, Section 5 gives final remarks and takeaways.

2 WHY MIGRATE TO PQC?

Considering that public-key cryptosystems are often used for authentication and Key Exchange (KEX), when vulnerable, they allow the following attacks:

- Impersonation: having access to the victim's private key sk in a digital signature system, the attacker can impersonate by signing messages with sk. If the private key of a web server is compromised, the attacker can create a "fake" server, and then every user will think that their connection is legitimate. The server's impersonation allows further attacks, such as disclosing user data and communications.
- Violate confidentiality: having access to the private key in a KEX process, the attacker obtains knowledge of symmetric encryption keys used in the user's communication. Therefore, the attacker can disclose the contents of the encrypted traffic.

In theory, a CRQC executes the Shor algorithm and gives the capability to a quantum attacker to recover a private key from the victim's public key. As of today, there is no publicly-known CRQC available. So, public-key cryptosystems used for authentication can not be exploited for impersonation until a CRQC arrives. Experts estimate that a CRQC will eventually be available, so such cryptosystems will have to be replaced (Mosca and Piani, 2020). Given the complexity related to authentication on the Internet, such as X.509 PKIs and the uncertainty of when a CRQC will be operational, applications and systems must be prepared in advance to prevent impersonations by quantum attackers.

In regards to KEX mechanisms, the quantum threat imposes additional concerns. KEX aids applications to provide confidentiality with symmetric encryption. For example, a typical KEX is the Elliptic-Curve Diffie-Hellman (ECDH), where the parties' public keys are exchanged in the communication channel. This exchange generates a shared secret that

is later used for deriving symmetric encryption keys. However, a quantum attacker could capture the whole KEX process and obtain the private keys from the exchanged public keys, thus allowing to generate the same symmetric keys. Therefore, attackers can exploit KEX mechanisms vulnerable to quantum computers to break confidentiality.

Vulnerable KEX mechanisms are worrisome because they are susceptible to a *record-now-decrypt-later* attack. Figure 1 illustrates the scenario where a quantum attacker is capturing encrypted-communicated data today, expecting to decrypt it in the future. Given that KEX is widely used in network protocols, such as TLS and SSH, quantum computers threaten the confidentiality of today's communications. Besides, Grover's quantum algorithm (Grover, 1996) is another threat to confidentiality. It weakens symmetric encryption algorithms by decreasing the key-space search in half, improving a brute-force attack. However, experts say Grover's algorithm is difficult to apply in practice. Additionally, a simple mitigation to Grover would be to double the security parameters of symmetric primitives, keeping the original security expectation.

The immediate solution to the quantum threat is a replacement of vulnerable algorithms by PQC. Sometimes called "PQC Drop-in replacement" or "PQC-only deployment", the vulnerable KEX and authentication mechanisms are replaced solely by PQC alternatives. Applications equipped with PQC can resist quantum threats, but there are still threats imposed by classical computation. For example, two promising PQC algorithms, SIKE and Rainbow, are now considered vulnerable to classical attacks (Castryck and Decru, 2022; Beullens, 2022). These examples suggest that migrating to PQC must be handled with care. In other words, a drop-in replacement of PQC should be done after the confidence in its security is well established. Instead, the hybrid mode is recommended for an early (and smother) adoption (Stebila and Mosca, 2016).

3 FIRST STEP: HYBRIDS

Hybrid PQC is an approach of adopting PQC in implementations, which supports Post-Quantum Cryptography (PQC) but maintains compatibility with the classical cryptography algorithms. In this work, the term Hybrid should not be confused with Hybrid Encryption (Kurosawa and Desmedt, 2004), a conjunction of symmetric and asymmetric cryptography. As mentioned before, the first reason for selecting the hybrid mode regards confidence in PQC security. Generally, the classical methods have higher confidence and years of utilization, both academically and by industry standards. Therefore, the hybrid mode is recommended, which means that both PQC and traditional algorithms are used in conjunction. Hence the security of the construction holds until at least one algorithm is not broken.

In practice, hybrids are being proposed as follows:

- Concatenation of KEX objects (Stebila et al., 2020): two (or more) KEX mechanisms execute in parallel, but the exchanged public keys (or ciphertexts) of the KEX parties are concatenated before sending. Each KEX will produce a shared secret concatenated prior to symmetric key-derivation. In this way, symmetric keys are produced with seeds from a classical and a PQC algorithm. An attacker would need to break each KEX to obtain the symmetric keys.
- Dual signatures: For authentication with digital signatures, the same data can be signed twice but using different signing keys (a PQC and a classical one). The verifier checks the two signatures for authenticating the data. Legacy implementations can be compatible but will check only the classical signature. Regarding the PKI infrastructure for authentication, there are three possibilities (Ounsworth and Pala, 2019; Ounsworth, 2023):
  - Composite hybrid: in this strategy, two (or more) cryptographic objects are concatenated, composing the hybrid. For example, a composite instance would concatenate two signatures or two public keys, one from PQC and the other

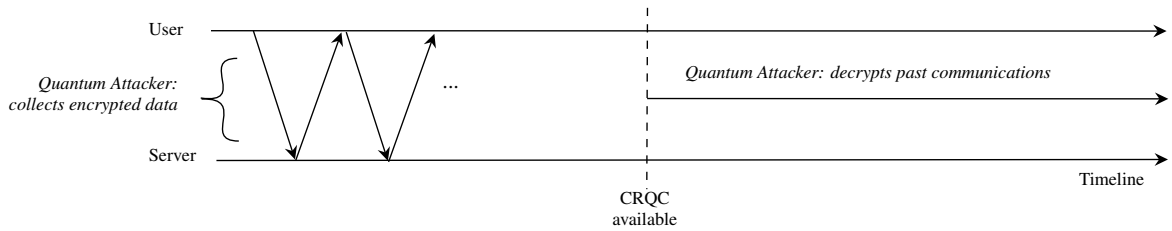

Figure 1: Record-now-decrypt-later attack timeline

from a classical algorithm. Composite is simple to implement in practice.
- "Catalyst Hybrid": similar to the composite, but applied to digital certificates, the PQC algorithm objects are added through X.509 extensions. Such extensions can be non-critical to avoid damaging legacy implementations.
- Parallel PKIs: in this mode, the implementation adds a second PKI (or more), which uses PQC algorithms only. Adding a second PKI probably incurs into a new set of certificates to be handled by the implementation (called certification paths or certificate chains).

3.1 Implementations and evaluations

The Open Quantum Safe (OQS) Project is a notorious effort to provide a cryptographic library for use by the community. In addition, OQS provides example implementations (such as OQS-OpenSSL) and programming language *bindings* for broad adoption. In their implementations, the hybrid mode is not only recommended but present. Other implementations followed the same strategy, such as the Bouncy Castle (Factor, 2023), CIRCL (Faz-Hernández and Kwiatkowski, 2019), and OpenSSH (OpenSSH, 2022), in this case, providing hybrid modes for KEX operations. Table 1 summarizes implementations and applications using hybrid modes. The industry's interest in hybrid modes is evident, considering the Google, Cisco, and Cloudflare experiments (Braithwaite, 2016; Westerbaan, 2021; Kampanakis, 2020), and Internet Engineering Task Force standardization drafts (Stebila et al., 2020). They focus on the hybrid KEX for PQC adoption.

Table 1: Popular cryptographic implementations and its PQC support features (if present).

Implementation	Release Version	PQC Support?	Hybrid mode?
OQS-OpenSSL	1.1.1	✓	Present
OpenSSL	3.1.0	✗	Not present
OpenSSH	9.0	✓	Default
Bouncy Castle	1.73	✓	Present
Wolf SSL	5.6.0	✓	Present
Mbed TLS	v3.4.0	✗	Not present
CIRCL	v1.3.2	✓	Present

Several reports suggest minor performance penalties when comparing hybrids to PQC-only replacements (Paquin et al., 2020; Braithwaite, 2016). For example, Sikeridis et al. (Sikeridis et al., 2020) experimented with hybrids in TLS and SSH protocols. Their work shows that the average latency of hybrids is less than 2% compared to PQ-only instances. Combining efficient elliptic curve operations with the PQC alternatives in a hybrid mode results in a good performance, considering computation time and byte costs.

3.2 PQC adoption challenges

Unfortunately, PQC significantly increases the sizes of cryptographic objects, such as public keys and signatures. Building a hybrid instance requires adding at least one PQC algorithm (called ingredient) to the classical scheme, so it increases the number of cryptographic objects being transmitted between parties, and it requires a cryptographic combiner that has to combine the algorithms securely (i.e., keeping the security properties of the combined ingredients) (Bindel et al., 2019).

With the rise of new computing environments, including 5G networks and Vehicular Networks (VNs) (Husain et al., 2019), adopting PQC would require additional efforts and specific analysis. Furthermore, due to their specific requirements, other challenges may appear when deploying hybrids in TLS for different applications. Using VNs as an example, they are critically affected by the cost of communication payload data and message headers (Husain et al., 2019). VNs connect On-Board-Units (OBUs) between cars, where hardware and software updates are complex due to their long-term lifetime. These particularities can impose additional challenges when deploying PQC (and hybrids) in such environments.

In practice, developers will face a decision concerning the PQC algorithms they will need in their applications. The NIST PQC standardization process can help in such a decision (NIST, 2016). For Key-Encapsulation Mechanisms (KEMs), NIST announced Kyber as a primary choice for KEM to be standardized until 2024. For Digital Signatures, NIST announced Dilithium as a primary choice but added Falcon and Sphincs+ as alternatives. Table 2 allows comparing sizes of selected PQC algorithms and classical algorithms. Noteworthy, other organizations are putting efforts in PQC: the standards being proposed by the IETF and the recommendations issued by European agencies such as ENISA, ETSI, BSI, and ANSSI (Ounsworth, 2023). Besides, NIST has recently started a migration project that would help the PQC adoption awareness (NIST, 2023).

Decreasing cryptographic payloads is beneficial for the performance of network applications. Still, the increased-size challenge remains open. For example, the NIST PQC process has an open call for new signature schemes with smaller sizes. Although hybrids share the same problem, Table 2 shows that adding the size of classical schemes to PQC does not increase the overall size significantly.

Table 2: Comparison of classical and some PQC schemes selected by the NIST process.

Algorithm Name	Parameter Set Name	Public Key size (bytes)	Ciphertext or Signature size	(S)ignature or (K)EM/KEX	Quantum-safe?
NIST P256	SECP256R1	64	64	K	✗
Kyber	KYBER512	800	768	K	✓
	KYBER768	1184	1088	K	✓
ECDSA	ECDSA_SECP256R1	64	64	S	✗
Falcon	FALCON-512	897	690	S	✓
Dilithium	DILITHIUM2	1312	2420	S	✓
Sphincs+	SPHINCS+-SHAKE256-128S-SIMPLE	32	7856	S	✓
	SPHINCS+-SHAKE256-128F-SIMPLE	32	17088	S	✓

In summary, this paper takes a position in favour of a hybrid strategy for the PQC migration for the following reasons.

- Non-disruptive transition to PQC: while an update in administrated devices, such as TLS servers, can be easier, user devices may not have the same care, time or awareness about the quantum threat to cryptography. Besides, the complexity of Public-Key Infrastructures does not allow a disruptive change without denying services to non-compatible users. The best-case scenario lies in both sides having the Hybrid PQC capabilities and negotiating their preferred algorithms. The negotiation process should allow backward compatibility to avoid denying access to services, depending on internal policies.
- Confidence in traditional cryptography: algorithms such RSA and DH has been studied for many years. Some PQC algorithms and their cryptographic assumptions are more recent compared to others. Users may have more confidence in traditional cryptography regarding its security. In addition, the quantum threat does not replace current threats. Users could keep the security provided by classical algorithms and combine it with PQC algorithms for better protection.
- Regulatory and industry compliance requirements: government and industry use cases of cryptography may have to obey specific regulations or compliance with published standards. For these use cases, it is impossible to switch to PQC, at least until new regulations occur.
- Level of scrutiny of implementations: regardless of being open-source, cryptographic implementations are subject to a wide range of attacks. Attackers can exploit the improper use of randomness, "unknown" backdoors, vulnerable padding procedures, and side-channel attacks. The responsible disclosure of such attacks helps to improve security. This scrutiny is different in level when comparing PQC with traditional cryptography, but this is expected to change with time. On the other hand, the wide variety of available cryptographic implementations indicates that this scrutiny can take many years.

4 APPLICATION-LEVEL RISKS

In preparation for the PQC migration, awareness of the quantum threats is essential. The reality is that several users, system administrators, and government agencies are getting concerned about the security of their applications against quantum computers (Mosca and Piani, 2020). Given the urgency of *store-now-decrypt-later* attacks, this need requires increased attention.

Although deploying hybrids is recommended, this work argues that it may not be enough for quantum-safe protection from the application's perspective. The main reason is due to the sensitive data that some applications have to manage. Such sensitive data could be further explored by quantum attackers, even if the application has already migrated to PQC. Below, Section 4.1 describes the risks for general applications and Section 4.2 discusses the risks for specific application-layer protocols.

4.1 Managing User data

Consider a common Internet-banking application *A* as an example. Consider a pre-quantum period, in which *A* communicates with users using classical cryptography (thus vulnerable to quantum computers), and a post-quantum period, in which *A* uses a quantum-safe communication infrastructure, i.e. using PQC. Note that the content users send to *A* includes confidential data, such as the bank account number and passwords for financial transactions. However, users might have communicated in the pre-quantum period, thus vulnerable to a *record-now-decrypt-later* attack.

Table 3: Application-layer risks under a record-now-decrypt-later threat (not exhaustive).

Application-layer Protocol/Utility	Specification	Secure Channel Provider	Sensitive Information	Risk Description
Basic HTTP Authentication	RFC 7617	TLS	User Credentials	Exchanged long-term credentials can allow access to server's resources after breaking TLS with a quantum computer
OAuth 2.0	RFC 6749	TLS	Refresh token	RFC leaves to implementations to explicitly define expiration time; an example of refresh token expiration time is one year (Restrepo, 2022). Attackers could obtain valid tokens exchanged with classical TLS.
OIDC/OAuth 2.0	(Sakimura et al., 2023)	TLS	ID Token, Refresh tokens	Similar to OAuth 2.0 (already pointed out by (Schardong et al., 2022))
Kerberos V5 (with kinit)	RFC 4120, RFC 4556	N/A	Renewal Ticket	In theory, ticket-granting tickets exchanged with classical cryptography combined with a long-lifetime ticket renewal policy (from 0 to 99,999 days) (Long et al., 2023) could be exploited by a quantum attacker.
Email Protocols	RFC 8314	TLS	User Credentials	RFC 8314 recommends TLS for IMAP, SMTP and other email protocols. Quantum attackers could exploit long-term user credentials exchanged with TLS.
WebRTC	(W3C, 2023)	DTLS	Authentication password	WebRTC specifies different authentication methods, if long-term passwords are used, a quantum attacker could recover the password after breaking the DTLS session.
Rsync over SSH	(Tridgell et al., 2022)	SSH	Server password	Rsync allows sharing files over SSH for security. A quantum attacker could decrypt SSH tunnels and recover exchanged rsync passwords.

Therefore, even after *A* has migrated to PQC, a quantum attacker can decrypt past communications and use the confidential data (e.g., passwords) for further attacks if such data is still valid. Long-term confidential data enables further interactions between the attacker and the application, in this case, performing financial transactions on the user's behalf.

The above example illustrates that long-term confidential information can be exploited after an application's PQC migration. For a complete migration to the post-quantum era, PQC algorithm replacement might not suffice if confidential (or long-term) user data can allow future interactions with the application. As a result, applications must manage user data considering what knowledge quantum adversaries can get. A "quantum risk assessment process" should include policies and security measures to protect against the quantum threat. For example, when migrating to PQC, the internet-banking application might have to manage (or enforce) user passwords using a quantum-safe channel or Out-Of-Band (OOB) mechanism.

4.2 Application-Layer Sensitive Data

Section 4.1 gives a generic example of an application. Below, this section describes the risks in concrete examples, in this case, application-layer protocols and software. Table 3 summarizes the risks under the *record-now-decrypt-later* attack. Each application requires a confidential channel, often provided by TLS. Since secure channel providers can be vulnerable to quantum threats, applications exchanging confidential data face different risks. Having these concrete examples, this position paper emphasizes that applications need PQC algorithms and risk analysis for the PQC migration.

Noteworthy, application-layer protocols do not need significant changes to accommodate PQC. In some cases, updating their TLS-based configurations (such as digital certificates) and implementations is enough to transit to PQC; however, it may not be enough to secure against the risks that come with *record-now-decrypt-later* attacks. Therefore, each application protocol might need specific analysis and further changes for complete protection. Considering the risks in Table 3, mitigation measures include: limiting authorizations and access token duration time, enforcing a policy for long-term confidential data usage, and revoking past actions performed with classical cryptography. The main drawback is that developing such measures in the application increases the PQC migration efforts (and time).

Considering SSH as an example, Table 1 showed that OpenSSH already deployed PQC in hybrid mode, thus mitigating quantum attacks. Therefore, quantum attackers could exploit applications on top of OpenSSH if (a) the application has not yet updated OpenSSH and the quantum attacker has captured the encrypted communication data; or (b) if the application has updated but the user data (e.g., passwords) have a long-term lifetime, so past communications that exchanged it makes the *record-now-decrypt-later* attack still valid for further exploitation. Although OpenSSH already supports hybrid, researchers noted the absence of a PQC working group (WG) for SSH (Kampanakis and Lepoint, 2023). This work corrob-

orates this need but expand it to application-layer protocols like those presented in Table 3.

5 TAKEAWAYS

In summary, this position paper discussed the need for migrating applications to PQC. It emphasized the PQC adoption challenges and argued about the hybrids as the recommended mechanism for an easier transition. Additional risks were discussed when applications had already migrated to PQC. In such a case, this work showed that applications will require specific risk analysis and implement security measures for complete protection against quantum threats.

Given the contextualization and PQC adoption open challenges identified in this work, this position paper issues the following takeaways.

- Consider hybrids as the recommended PQC migration strategy, given the favourable scenario regarding performance comparisons and security confidence in PQC. Remember that adopting PQC is still challenging for some applications, e.g., due to increased sizes. There is ongoing research to address these challenges.
- Take a specific approach for analyzing how is the best option for PQC migration. Note that it does not only need a PQC algorithm selection that best suits the application's needs. The PQC migration strategy should include risk analyses related to long-term confidential data and other information that quantum-capable attackers could explore. In summary, application-layer software needs no significant changes to accommodate PQC, except when dealing with confidential data exchanged through quantum-vulnerable protocols.
- Call for improving participation in PQC adoption Work Groups (WG), also creating new WGs, aiming at increasing the awareness of the quantum threats for general applications. Such WGs would foment new risk analyses for other protocols, such as the application-layer protocols that were not yet analyzed elsewhere.

Hybrid modes for the post-quantum transition may be a temporary approach. However, this does not necessarily mean that it will be a short period. On the contrary, Hybrid PQC can be present in network communications for an extended period, for as long as needed to gain full confidence in PQC security. Additionally, the awareness of the effects of quantum threats and how to mitigate them helps to build a secure post-quantum world.

ACKNOWLEDGEMENTS

The author would like to say thanks to Ricardo Custódio, the Federal University of Technology - Parana (UTFPR/Brazil) and the Technology Innovation Institute (TII/UAE) for their support.

REFERENCES

Bernstein, D. J. and Lange, T. (2017). Post-quantum cryptography. *Nature*, 549(7671):188–194.

Beullens, W. (2022). Breaking rainbow takes a weekend on a laptop. Cryptology ePrint Archive, Paper 2022/214. https://eprint.iacr.org/2022/214.

Bindel, N., Brendel, J., Fischlin, M., Goncalves, B., and Stebila, D. (2019). Hybrid key encapsulation mechanisms and authenticated key exchange. In Ding, J. and Steinwandt, R., editors, *Post-Quantum Cryptography*, pages 206–226, Cham. Springer International Publishing.

Braithwaite, M. (2016). Experimenting with post-quantum cryptography. https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html.

Castryck, W. and Decru, T. (2022). An efficient key recovery attack on sidh (preliminary version). Cryptology ePrint Archive, Paper 2022/975. https://eprint.iacr.org/2022/975.

Factor, K. (2023). Post-quantum hybrid cryptography in Bouncy Castle. https://doc.primekey.com/bouncycastle/post-quantum-hybrid-cryptography-in-bouncy-castle.

Faz-Hernández, A. and Kwiatkowski, K. (2019). *Introducing CIRCL: An Advanced Cryptographic Library*. Cloudflare. Available at https://github.com/cloudflare/circl. v1.3.2 Accessed Jan, 2023.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219.

Husain, S. S., Kunz, A., Prasad, A., Pateromichelakis, E., and Samdanis, K. (2019). Ultra-high reliable 5g v2x communications. *IEEE Communications Standards Magazine*, 3(2):46–52.

Joseph, D., Misoczki, R., Manzano, M., Tricot, J., Pinuaga, F. D., Lacombe, O., Leichenauer, S., Hidary, J., Venables, P., and Hansen, R. (2022). Transitioning organizations to post-quantum cryptography. *Nature*, 605(7909):237–243.

Kampanakis, P. (2020). Post-quantum tls 1.3 and ssh performance (preliminary results). https://blogs.cisco.com/security/tls-ssh-performance-pq-kem-auth.

Kampanakis, P. and Lepoint, T. (2023). Vision paper: Do we need to change some things? In Günther, F. and Hesse, J., editors, *Security Standardisation Research*, pages 78–102, Cham. Springer Nature Switzerland.

Kurosawa, K. and Desmedt, Y. (2004). A new paradigm of hybrid encryption scheme. In Franklin, M., editor,

Advances in Cryptology – CRYPTO 2004, pages 426–442, Berlin, Heidelberg. Springer Berlin Heidelberg.

Long, L., Mandalika, S., Simpson, D., Gorzelany, A. M., Hall, J., Bichsel, A., and Pamnani, V. (2023). Maximum lifetime for user ticket renewal. https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/maximum-lifetime-for-user-ticket-renewal.

Mosca, M. and Piani, M. (2020). Quantum threat timeline report 2020. Available at: https://globalriskinstitute.org/publications/quantum-threat-timeline-report-2020/. Accessed on 20.07.2021.

NIST (2016). Post-quantum cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography.

NIST (2023). Migration to post-quantum cryptography. https://www.nccoe.nist.gov/crypto-agility-considerations-migrating-post-quantum-cryptographic-algorithms.

OpenSSH (2022). Openssh 9.0 release notes. https://www.openssh.com/txt/release-9.0.

Ounsworth, M. (2023). PQC at the IETF. https://pkic.org/events/2023/post-quantum-cryptography-conference/pkic-pqcc-pqc-at-ietf-mike-ounsworth-entrust.pdf.

Ounsworth, M. and Pala, M. (2019). Composite keys and signatures for use in Internet PKI. https://datatracker.ietf.org/doc/html/draft-ounsworth-pq-composite-sigs-01. Internet-Draft.

Paquin, C., Stebila, D., and Tamvada, G. (2020). Benchmarking post-quantum cryptography in tls. In Ding, J. and Tillich, J.-P., editors, *Post-Quantum Cryptography*, pages 72–91, Cham. Springer International Publishing.

Partners Digital, G. (2023). World map of encryption laws and policies. https://www.gp-digital.org/world-map-of-encryption/.

Restrepo, R. (2022). Oauth 2.0 refresh token best practices. https://stateful.com/blog/oauth-refresh-token-best-practices.

Sakimura, N., Bradley, J., Jones, M. B., de Medeiros, B., and Mortimore, C. (2023). *OpenID Connect Core 1.0*. Available at https://openid.net/specs/openid-connect-core-1_0.html. Accessed March, 2023.

Schardong, F., Giron, A. A., Müller, F. L., and Custódio, R. (2022). Post-quantum electronic identity: Adapting openid connect and oauth 2.0 to the post-quantum era. In Beresford, A. R., Patra, A., and Bellini, E., editors, *Cryptology and Network Security*, pages 371–390, Cham. Springer International Publishing.

Schwabe, P., Stebila, D., and Wiggers, T. (2020). Post-quantum tls without handshake signatures. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, CCS '20, page 1461–1480, New York, NY, USA. Association for Computing Machinery.

Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134, Santa Fe, NM, USA. IEEE, IEEE.

Sikeridis, D., Kampanakis, P., and Devetsikiotis, M. (2020). Assessing the overhead of post-quantum cryptography in tls 1.3 and ssh. In *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pages 149–156, New York, NY, USA. Association for Computing Machinery.

Society, I. (2023). Internet Society. https://www.internetsociety.org/.

Stebila, D., Fluhrer, S., and Gueron, S. (2020). Hybrid key exchange in TLS 1.3. http://tools.ietf.org/html/draft-ietf-tls-hybrid-design-00. Internet-Draft.

Stebila, D. and Mosca, M. (2016). Post-quantum key exchange for the internet and the open quantum safe project. In *International Conference on Selected Areas in Cryptography*, pages 14–37. Springer.

Tridgell, A., Mackerras, P., and Davison, W. (2022). *rsync - a fast, versatile, remote (and local) file-copying tool*. Available at https://download.samba.org/pub/rsync/rsync.1. Accessed Apr, 2023.

W3C, W. (2023). *WebRTC: Real-Time Communication in Browsers*. Available at https://www.w3.org/TR/webrtc/. Accessed March, 2023.

Westerbaan, B. (2021). Sizing up post-quantum signatures. https://blog.cloudflare.com/sizing-up-post-quantum-signatures/.