Integrating post-quantum cryptography to UXP

Technical Report

Version 1.1

10.01.2023

ID D-2-499

Annotation

Mailing address:
Cybernetica AS
Mäealuse 2/1
12618 Tallinn
Estonia

Table of Contents

1 Introduction

It is likely that quantum computers will be able to break many of the public key cryptographic algorithms currently in use (RSA, ECDH, ECDSA) as quantum computers become more powerful and widespread. This presents a significant risk to the security of data exchange networks. An adversary with access to a quantum computer could potentially compromise the confidentiality and integrity of transmitted data.

To address this risk, it is important to consider integrating post-quantum cryptographic methods into the Unified eXchange Platform (UXP). In this document, we will discuss the various post-quantum cryptographic primitives that are currently available, and how they could be integrated into UXP to provide enhanced security. We will also examine the challenges and considerations involved in implementing post-quantum cryptography in UXP, and provide recommendations for how to effectively deploy these techniques in a production environment.

2 Post-quantum cryptography

It is hard to estimate when a quantum computer of sufficient power will be built. EvolutionQ published the Quantum Threat Timeline Report 2021 [1], where international quantum computing experts were asked a series of questions. In one of the questions, experts were asked to assess the likelihood of building a quantum computer sufficiently powerful that is able to break RSA-2048 in less than 24 hours. Their answers are illustrated in Figure 1.

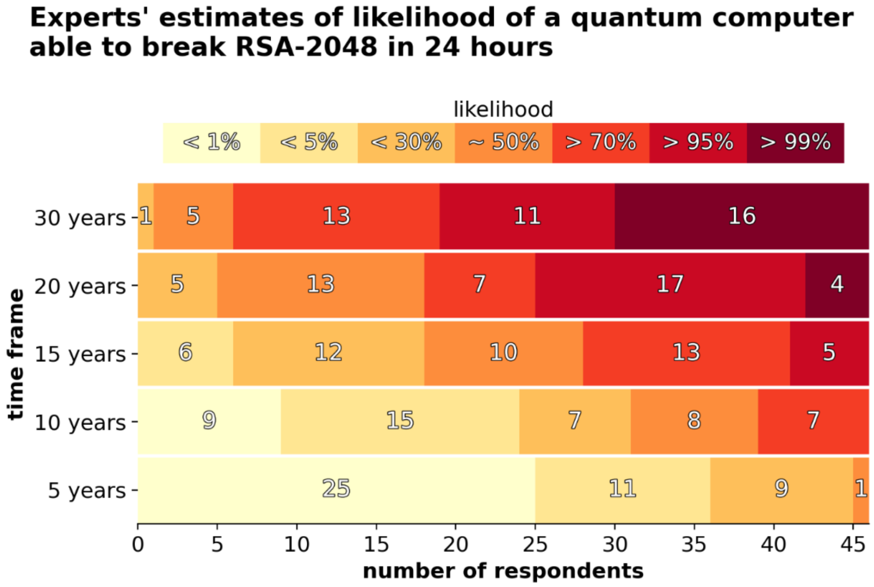

Figure 1. Likelihood of breaking RSA-2048—in less than 24 hours from [1]

As can be seen from the graph, roughly 90% of respondents believe that it is about 50% or more likely that we will have a such quantum computer in 20 years. About 60% of respondents think that it is about 50% or more likely that we will have a such quantum computer in 15 years [1]. It may seem that we have more than enough time to make our systems secure against quantum computer attacks. However, we should not forget that the migration time from traditional cryptography to post-quantum cryptography could take an extensive amount of time. Additionally, we may need to add some extensions or change the way how some protocols work in order to be compatible with post-quantum primitives. One of the biggest challenges when it comes to post-quantum cryptography is the increased size of keys, ciphertexts, and signatures that they generate. Thus, if we implement these schemes directly into protocols like TLS or DNSSEC, it is very likely that the messages that result from those schemes are not going to meet the size limitations of those protocols. It means that there will be fragmentation of messages and additional round-trips, which leads to latency and data traffic increase. Moreover, no single post-quantum signature scheme fits all use-case scenarios. While some have smaller signature sizes, others are easier to implement but have larger signature sizes.

2.1 Standardisation of post-quantum cryptography

In 2016, the US National Institute of Standards and Technology started the process of selecting key encapsulation mechanisms (KEM) and digital signatures that are resistant to quantum computer attacks [2]. In total 69 schemes were submitted to the first round of competition, and in July 2022 4 schemes were selected to be standardised. The following schemes will be standardised by NIST [3]:

Name	Type	Family
Crystals-Kyber [4]	KEM	Lattice-based
Crystals-Dilithium [5]	Signature	Lattice-based
Falcon [6]	Signature	Lattice-based
SPHINCS+ [7]	Signature	Hash-based

Table 1. Algorithms selected for NIST standardisation

Kyber is currently the only KEM that NIST will standardize. The overall performance of Kyber is excellent in terms of software, hardware, and a wide range of hybrid environments. In general, Kyber appears to provide a high level of security without sacrificing performance [3].

Dilithium will be standardised first and NIST proposes using it as a primary signature algorithm. Compared with other signature schemes that participated in the standardisation, it has the fastest key generation and signing algorithms. Additionally, it has a relatively simple implementation.

Due to the small size of public key and signature and fast verification, Falcon is a suitable choice in some network-constrained scenarios. However, key generation and signing algorithms in Falcon are complex to implement as they require floating-point arithmetic. Without using floating-point arithmetic, both algorithms are much slower.

In terms of security, the most conservative signature scheme selected to be standardised is Sphincs+. Its security is based only on the security of underlying symmetric primitives (hash functions). The basic primitives on which Sphincs+ is constructed are one-time signatures, few-time signatures, and Merkle trees. Due to the way it is constructed, Sphincs+ has a limit on the signatures that can be produced for a given public key. Sphincs+ has small public and private key sizes, but very big signatures compared to the other signature schemes.

Moreover, there will be another round for KEM [8] and one more call for digital signature schemes [9]. The main reason for that is that NIST wants to standardise KEM and signature scheme that are not based on structured lattices.

Additionally, there was a parallel process of standardising stateful hash-based signatures Leighton-Micali Signature (LMS) and eXtended Merkle Signature Scheme (XMSS) [10]. These signatures are also secure against quantum computer attacks since their security is based only on the security of the underlying hash function. Smaller signature and faster signing algorithms distinguish LMS and XMSS from SPHINCS+ (stateless hash-based signature). These signature schemes use one-time signatures, which means that a key pair can only be used once for signing. In order to construct more complex structures, XMSS and LMS use many one-time key pairs. To create a signature, the signer must keep a state indicating which one-time keys were already utilised. This could lead to problems in managing the state securely. Systems implementing stateful hash-based signatures should have protection against attacks in which the adversary powers off the device after the signature has been created but before the updated state is saved. Another rising question is how to create backups of the private key and the state.

It is often suggested to start using stateful hash-based signatures already now, for example for firmware updates. This is a suitable use case as there is one central signing party that creates signatures and keeps track of the state. Additionally, this use case does not require creating many signatures as firmware updates are infrequent.

The view on post-quantum KEMs of the German Federal Office for Information Security (BSI) is different from the NIST. Technical Guideline TR-02102-1 [11] proposes using FrodoKEM (based on structured lattices) and Classic McEliece (based on codes) for the protection of long-term secrets in a hybrid mode. In terms of security, these schemes are among the most conservative ones, and there is a belief that structured lattices are too new to have been studied and matured yet.

Similarly, the French National Agency for the Security of Information Systems (ANSII) encourages developers to start transitioning to the hybrid mode for systems that aim at offering long-lasting security as soon as possible [12]. In addition, ANSII believes that FrodoKEM is a very conservative candidate (in terms of security) and should be a viable option for applications that support such large keys.

Table 2 presents a comparison of different KEMs and digital signature schemes. All the numbers correspond to an instantiation of the scheme with parameters that guarantee NIST security level 5 (at least as hard to break as AES256). Benchmarking results are taken from [13], algorithms are compiled with all CPU extensions available but with code portability/distributability features switched on (Intel(R) Xeon(R) Platinum 8259CL CPU @ 2.50GHz). For KEMs table presents the sizes of public key (pk), private key (sk), and ciphertext (ctx) and the number of operations per second for key generation (gen), encapsulation (encaps), and decapsulation (decaps) algorithms. For signature schemes, the table presents the sizes of public key (pk), private key (sk), and signature (sig) and the number of operations per second for key generation (gen), signing (sign), and verification (ver) algorithms.

Name	Sizes (in bytes)	Efficiency (ops/sec)	Who suggests to use
Key Encapsulation Mechanisms			
Kyber1024	pk: 1 568 sk: 3 168 ctx: 1 568	gen: 36949.33 encaps: 32357.67 decaps: 41785.67	NIST (standard)
FrodoKEM-1344-SHAKE	pk: 21 520 sk: 43 088 ctx: 21 632	gen: 204.53 encaps: 190.35 decaps: 193.01	BSI ANSII
Classic McEliece-8192128	pk: 1 357 824 sk: 14 120 ctx: 240	gen: 1.47 encaps: 14101.67 decaps: 6723.00	BSI NIST (4th round)
Signature Schemes			
Dilithium5	pk: 2 592 sk: 4 864 sig: 4 595	gen: 9991.00 sign: 5224.33 ver: 10612.00	NIST (standard)
Falcon-1024	pk: 1 793 sk: 13 953 sig: 1 280	gen: 39.11 sign: 1403.33 ver: 8671.67	NIST (standard)
Sphincs$^+$-256f (fast)	pk: 64 sk: 128 sig: 49 856	gen: 226.36 sign: 10.66 ver: 225.77	NIST (standard)
Sphincs$^+$-256s (small)	pk: 64 sk: 128 sig: 29 792	gen: 13.82 sign: 1.13 ver: 438.85	NIST (standard)

Table 2. Comparison of post-quantum schemes

3 Prototyping and implementing post-quantum cryptography

In this section, we will describe how it is proposed to combine post-quantum and traditional cryptographic algorithms for both KEMs and signatures. We will highlight where post-quantum cryptography should be introduced and UXP and discuss challenges connected to those use cases. Moreover, we will give a brief overview of the current status of public key infrastructure (PKI) implementing post-quantum cryptography.

3.1 Hybrid mode

While migration to post-quantum cryptography happens and we are not yet confident enough in the security of post-quantum algorithms, we may need to use hybrid solutions. In other words, the preliminary solution is a combination of traditional and post-quantum schemes. It is quite straightforward to combine traditional key establishment with post-quantum KEM as follows:

1. A client sends a concatenation of values corresponding to the first message in the traditional key establishment (ECDH or KEM style) and post-quantum KEM. That is the public key for KEMs and the ephemeral keyshare for ECDH.
2. A server answers with ciphertext produced by encapsulation algorithm or ECDH ephemeral key share.
3. Shared secret is computed by concatenating shared secrets derived with traditional and post-quantum algorithms.

Only if both parties support post-quantum algorithms, the final shared secret contains a secret derived using a post-quantum algorithm. If one of the parties does not support post-quantum schemes, the shared secret will contain only part derived using traditional key establishment. The conception above is taken from a draft document that defines hybrid key exchange in TLS 1.3 [1]. Hybrid key establishment is illustrated in Figure 2.

[1] https://datatracker.ietf.org/doc/pdf/draft-stebila-tls-hybrid-design-03

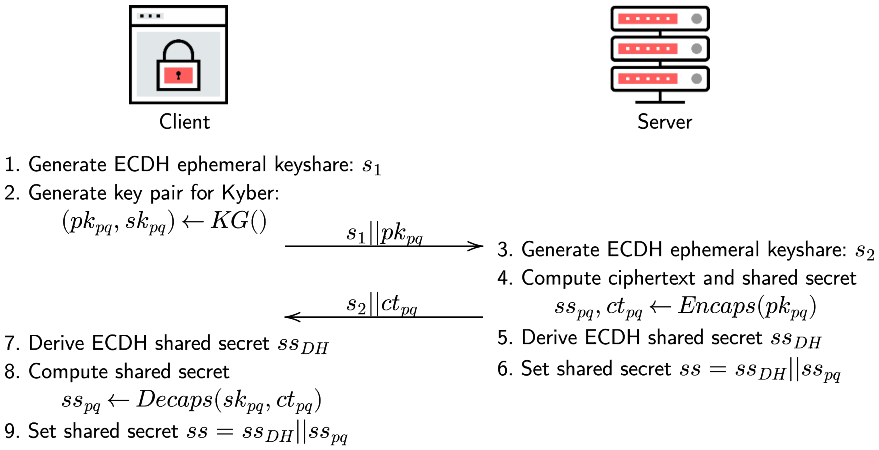
Client
Server
1. Generate ECDH ephemeral keyshare: s_1
2. Generate key pair for Kyber:
$(pk_{pq}, sk_{pq}) \leftarrow KG()$
$s_1||pk_{pq}$
3. Generate ECDH ephemeral keyshare: s_2
4. Compute ciphertext and shared secret
$ss_{pq}, ct_{pq} \leftarrow Encaps(pk_{pq})$
$s_2||ct_{pq}$
5. Derive ECDH shared secret ss_{DH}
6. Set shared secret $ss = ss_{DH}||ss_{pq}$
7. Derive ECDH shared secret ss_{DH}
8. Compute shared secret
$ss_{pq} \leftarrow Decaps(sk_{pq}, ct_{pq})$
9. Set shared secret $ss = ss_{DH}||ss_{pq}$

Figure 2. Hybrid key establishment

In contrast, the hybrid mode for signatures is more complex. A work by Bindel [14] describes several ways how signatures may be combined:

- *Concatenate* – place two independent signatures from two different schemes side-by-side;
- *Weak nesting* – first signature scheme just signs the message and the second signature scheme signs the signature from the first signature scheme;
- *Strong nesting* – first signature scheme just signs the message and the second signature scheme signs both the message and the signature from the first signature scheme;
- *Dual message combiner using nesting* – the first signature scheme signs the first message and the second signature scheme signs the first message, the signature from the first signature scheme and the second message (that might be somehow related to the first message).

The choice of method for combining signatures depends on the final use case.

3.2 Post-quantum cryptography in UXP

We identified several places where we need to introduce post-quantum algorithms in UXP:

- In TLS connections between security servers and external clients and corresponding certificates;
- To sign messages that are exchanged between security servers;
- To create global configuration signature.

Let us discuss each of the use cases separately. Since post-quantum signatures and KEMs are not yet standardised, they are not supported by TLS 1.3. NIST promised to publish standards for selected algorithms by the end of 2024. When it comes to post-quantum signatures in the TLS, there are several factors to keep in mind. One important distinction to make is between online and offline signatures. The signing speed is a crucial factor for online signatures that are created during a TLS handshake, whereas the size of the public key and signature are more important for offline signatures that are created well before the connection is established. The Dilithium signature scheme has the most efficient signing algorithm that suits online signatures, while Falcon is more suitable for offline signatures due to its smaller public key and signature size. However, using different signature schemes for different use cases can introduce challenges. For example, clients may need to support multiple signature algorithms to create and verify signatures, which could be difficult for devices with limited resources.

Hybrid signatures in the concatenation mode can be used to sign messages sent by the UXP security servers. This means that the size of messages sent over the network will increase since post-quantum signatures are much larger than the size of traditional signatures (RSA, ECDSA). On the bright side, the signing and verification time of lattice-based signatures is quite fast.

The global configuration signature can also be a hybrid of the traditional and lattice-based post-quantum scheme. Alternatively, for more conservative security it can be signed with a stateful hash-based signature (LMS or XMSS).

One key consideration when discussing post-quantum cryptography is how to handle public key certificates. Since the 2019 version of the ITU-T standard for X.509, it allows for the inclusion of alternative public key information, digital signature algorithms, and signatures within a public key certificate. These can be added to the certificate using the subjectAltPublicKeyInfo, altSignatureAlgorithm, and altSignatureValue extensions. This means that it is possible to include post-quantum public keys and signatures in certificates, allowing for parties who support post-quantum algorithms to use the alternative signature, while those that do not can still use the traditional signature [2]. However, it's important to keep in mind that adding post-quantum keys and signatures to a certificate increases its size, as these elements tend to be much larger than traditional ones.

3.3 Current status of PQ PKI

Due to the fact that the NIST standardization process is yet to be finished, there is no general guideline on how should we approach using post-quantum primitives in UXP. In this section, we present various solutions which could be used to benchmark post-quantum KEMs and digital signatures within UXP infrastructure.

3.3.1 DigiCert

DigiCert offers PQC toolkit [15] which enables the creation of hybrid TLS certificates based on the IETF draft for Multiple Public-Key Algorithm X.509 Certificates [16]. Their toolkit allows anyone to establish a local certificate chain and access the potential of deploying post-quantum hybrid TLS certificates, while still providing backward compatibility. At the moment of writing this report, the DigiCert PQC toolkit supports Dilithium and XMSS digital signatures.

3.3.2 Sectigo

Sectigo offers their Quantum Safe Kit [17] for creating quantum-safe digital certificates and hybrid digital certificates. Quantum safe certificates are generated using new post-quantum algorithms standardized by NIST. Conversely, hybrid digital certificates contain both traditional RSA and ECC keys and signatures as well as post-quantum keys and signatures.

3.3.3 Cloudflare

Cloudflare has conducted experiments incorporating post-quantum cryptographic primitives in the Transport Layer Security (TLS) 1.3 protocol [18]. Additionally, they have begun to incorporate post-quantum cryptography in their products. For example, Cloudflare Tunnel is a tool that enables users to connect to resources on Cloudflare without a public IP address and in a secure manner. It uses the Cloudflare network to create a connection between the user's device and the destination, allowing the user to access resources as if they were on the same network. Some of the connections established through Cloudflare Tunnel now utilize post-quantum cryptography to ensure security. Specifically, the connection from cloudflared to the Cloudflare network is now post-quantum secure.

3.3.4 Open Quantum Safe

The Open Quantum Safe project (OQS) is an open-source initiative that facilitates the development and testing of post-quantum cryptography. It maintains an open-source C library called *liboqs* which enables the integration of post-quantum cryptographic methods into existing protocols and applications, such as OpenSSL, BoringSSL, and OpenSSH. This project is widely used in research studies to assess the impact of post-quantum cryptography on various protocols and use cases. The *liboqs* library includes implementations of all post-quantum schemes that have been selected for standardization, as well as additional candidates from Round 3 of the NIST

[2] https://www.itu.int/rec/T-REC-X.509-201910-I/en

project. Some of the implementations come from the PQClean project [19], while others have been derived from reference or optimized versions submitted by teams to the NIST project.

The OQS-OpenSSL-1.1.1 package, which is actively maintained, provides post-quantum algorithms for use with TLS 1.3, X.509, and S/MIME. It supports both post-quantum-only and hybrid modes for KEMs and signatures. A fork of BoringSSL also provides support for post-quantum and hybrid key exchange, as well as post-quantum and hybrid public key authentication.

4 Summary and recommendations

Transitioning from traditional cryptography to post-quantum cryptography is not an easy one, as there are many challenges to overcome. It is important to recognize that one of the main challenges will be the increased size of keys, signatures, and ciphertexts, which may negatively affect how some of the commonly used protocols work. In spite of this, it would be beneficial to start testing and prototyping now if we want to identify which are the constraints of post-quantum cryptography in our use case. This way we will be able to deploy those algorithms whenever the need arises. Open Quantum Safe project's implementation of post-quantum cryptography already now allows starting testing hybrid modes in TLS 1.3, which can be run in the UXP security servers.

For key establishment in TLS, we recommend using Kyber1024 in hybrid mode (with elliptic curves) as it has reasonable performance and will be soon standardised by NIST. For TLS authentication we recommend using Dilithium5 in hybrid mode as it is the first post-quantum signature scheme that will get standardised. To sign messages that are exchanged between UXP security servers we recommend testing both Dilithium and Falcon in hybrid mode to see which one suits better. Falcon signatures are smaller and signature verification is faster, Dilithium has faster signing and it is easier to implement it securely.

For the global configuration signature, we also recommend testing two signature schemes. Firstly, stateful hash-based signatures seem to fit in this use case. Signatures are not often created by one party, who can maintain the state. Moreover, NIST already published their recommendation for stateful hash-based signature schemes [10] in 2020. Secondly, we recommend testing Falcon in hybrid mode as in this scenario verification speed is more important than signing speed.

Bibliography

[1] Marco Piani Michele Mosca. *2021 Quantum Threat Timeline Report: Global Risk Institute*. https://globalriskinstitute.org/publication/2021-quantum-threat-timeline-report-global-risk-institute-global-risk-institute/. Accessed: 2022-12-20. Jan. 2022.

[2] Information Technology Laboratory Computer Security Division. *Announcing Request for Nominations for Public-Key Post-Quantum Cryptographic Algorithms*. https://csrc.nist.gov/News/2016/Public-Key-Post-Quantum-Cryptographic-Algorithms. Accessed: 2022-12-20. Dec. 2016.

[3] Dustin Moody. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*. Tech. rep. National Institute of Standards and Technology, 2022. DOI: 10.6028/nist.ir.8413-upd1. URL: https://doi.org/10.6028/nist.ir.8413-upd1.

[4] Peter Schwabe et al. *CRYSTALS-Kyber Algorithm specifications and supporting documentation, Selected Algorithms of NIST's post-quantum cryptography standardization process*. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Accessed: 2022-12-08. Nov. 2022.

[5] Vadim Lyubashevsky et al. *CRYSTALS-DILITHIUM: Algorithm specifications and supporting documentation, Selected Algorithms of NIST's post-quantum cryptography standardization process*. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Accessed: 2022-12-08. Nov. 2022.

[6] Thomas Prest et al. *Falcon: Algorithm specifications and supporting documentation, Selected Algorithms of NIST's post-quantum cryptography standardization process*. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Accessed: 2022-12-08. Nov. 2022.

[7] Andreas Hulsing et al. *Sphincs+: Algorithm specifications and supporting documentation, Selected Algorithms of NIST's post-quantum cryptography standardization process*. https://csrc.nist.gov/Projects/post-quantum-cryptography/selected-algorithms-2022. Accessed: 2022-12-08. Nov. 2022.

[8] National Institute of Standards and Technology. *Round 4 Submissions*. https://csrc.nist.gov/Projects/post-quantum-cryptography/round-4-submissions. Accessed: 2022-12-20. Nov. 2022.

[9] National Institute of Standards and Technology. *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process*. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf. Accessed: 2022-12-20. Aug. 2022.

[10] National Institute of Standards and Technology. *Stateful Hash-Based Signatures*. https://csrc.nist.gov/projects/stateful-hash-based-signatures. Accessed: 2022-12-20. Oct. 2022.

[11] German Federal Office for Information Security. *BSI TR-02102-1: "Cryptographic Mechanisms: Recommendations and Key Lengths" Version: 2022-1*. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html. Accessed: 2022-12-20. Feb. 2022.

[12] Agence nationale de la sécurité des systèmes d'information. *ANSSI views on the Post-Quantum Cryptography transition*. https://www.ssi.gouv.fr/uploads/2022/01/anssi-technical_position_papers-post_quantum_cryptography_transition.pdf. Accessed: 2022-12-20. Mar. 2022.

[13] Open Quantum Safe Project. *OQS algorithm performance visualizations*. https://openquantumsafe org/benchmarking/visualization/speed_sig.html. Accessed: 2022-12-20.

[14] Nina Bindel et al. "Transitioning to a Quantum-Resistant Public Key Infrastructure". In: *Post-Quantum Cryptography - 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings*. Ed. by Tanja Lange and Tsuyoshi Takagi. Vol. 10346. Lecture Notes in Computer Science. Springer, 2017, pp. 384–405. DOI: 10.1007/978-3-319-59879-6_22. URL: https://doi.org/10.1007/978-3-319-59879-6%5C_22.

[15] digicert. *PQC toolkit setup guide*. https://docs.digicert.com/en/certcentral/certificate-tools/post-quantum-cryptography.html. Accessed: 2022-12-21.

[16] Alexander Truskovsky et al. *Multiple Public-Key Algorithm X.509 Certificates*. Internet-Draft draft-truskovsky-lamps-pq-hybrid-x509-01. Work in Progress. Internet Engineering Task Force, Aug. 2018. 24 pp. URL: https://datatracker.ietf.org/doc/draft-truskovsky-lamps-pq-hybrid-x509/01/.

[17] Sectigo. *Introducing Post Quantum Cryptography, Quantum Computers Will Change PKI*. https://sectigo.com/quantum-labs. Accessed: 2022-12-21.

[18] Bas Westerbaan. *Introducing post-quantum Cloudflare Tunnel*. https://blog.cloudflare.com/post-quantum-tunnel/. Accessed: 2022-12-21. Mar. 2022.

[19] Matthias J. Kannwischer et al. "Improving Software Quality in Cryptography Standardization Projects". In: *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*. IEEE, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010. URL: https://doi.org/10.1109/EuroSPW55150.2022.00010.