Post-Quantum Security Overview of the Public Key Infrastructure

Fruzsina Bene
Department of Information Systems
ELTE Eötvös Loránd University
1117 Budapest, Hungary
aoo2zv@inf.elte.hu
0000-0001-5062-0708

Attila Kiss
Department of Information Systems
ELTE Eötvös Loránd University,
1117 Budapest, Hungary
Department of Informatics, J. Selye University
945 01 Komárno, Slovakia
kiss@inf.elte.hu
0000-0001-8174-6194

***Abstract*—Recently, there has been an increasing focus on the investigation of quantum-safe solutions for a variety of applications. One of the pressing issues that needs to be made quantum secure is the TLS (Transport Layer Security) protocol. Proposals for its implementation have been discussed in several articles. The TLS protocol is based on PKI (Public Key Infrastructure). In addition, there are many other PKI applications that are used every day in both private and enterprise environments, so securing their use is essential. The methods currently developed to ensure adequate security will become obsolete with the advent of quantum computers. According to the Cloud Security Alliance, by around 2030, the performance of quantum computers will increase to the point where the risk of vulnerability of traditionally encrypted data will be very high. It is therefore important to make the right preparations in time to ensure that we can transform our solutions into quantum secure solutions by the time quantum computing becomes a real threat. In this paper, we present an analysis to this end, presenting quantum-safe solutions already in use and, in comparison, proposing new, well-performing solutions for a quantum-resistant PKI.**

***Index Terms*—cryptography, quantum computing, post-quantum cryptography, public key infrastructure, cybersecurity**

I. Introduction

The classical methods we currently rely on for PKI will no longer be secure when it comes to quantum computing. We need to be prepared and need to find new solutions such as integrating post-quantum cryptography (PQC) algorithms to make it quantum-resistant. The National Institute of Standards and Technology (NIST) initiated a standardization project for PQC algorithms with the aim for creating secure quantum-resistant cryptographic algorithms that can replace classical algorithms used in so many fields for authentication, secure communication and information transfer [5].

An important concept is that solutions using PKI should be cryptographically agile. Crypto-agility is defined as the ability of a security system to be able to rapidly switch between algorithms, cryptographic primitives, and other encryption mechanisms without the rest of the system's infrastructure being significantly affected by these changes [6]. With this in mind, we provide possible PQC solutions for PKI systems and take stock of some of the solutions already in use. This topic has been addressed in several previous articles (e.g. [1], [2], [3]). In our previous work, we have provided a comprehensive overview of existing methods for quantum-safe solutions to PKI and recommendations for future implementations [34].

In this paper, we aim to present additional solutions in more detail and discuss further details of PQC algorithms that have undergone several updates since their release. As this field is changing dynamically day by day, it is necessary to revisit from time to time the existing solutions and the requirements that need to be met. The algorithms used are also under constant scrutiny, as the security of PQC algorithms is not constant, attacks against them are constantly being developed and many algorithms previously considered secure are hacked over time. For this reason, the systems we use and which we trust need to be regularly monitored, and up-to-date security analyses need to be provided for this monitoring.

II. Public Key Infrastructure introduction

Today various data transfers play an important role in our everyday life and it is essential to do all of them in a secure way. It is important especially for organizations that they can provide secure internal communication and they make sure that all connections are made securely. To these problems public key infrastructure (PKI) offers a solution that manages encryption and secure authentication with creating and managing certificates and public keys [11]. PKI is responsible for creation, issuance, publication, management and revocation of public keys for digital signatures. PKI thus ensures that anyone using an open network can be clearly identified [9].

A. How does Public Key Infrastructure work?

PKI is not a single product or service, rather it is a set of policies, roles, procedures, hardware and software that provide the link between real communication parties (such as people, devices or vendors) and public keys [1]. Encryption keys

Cite as: F. Bene and A. Kiss, "Post-Quantum Security Overview of the Public Key Infrastructure", *Syst. Theor. Control Comput. J.*, vol. 3, no. 2, pp. 27–35, Dec. 2023.
DOI: 10.52846/stccj.2023.3.2.55

need to be assigned to identities so that the communication parties can always verify each other. To ensure this, PKI uses digital certificates. Certificates are basic digital documents that provide assurance of correspondence between a user and its public key [11].

Based on these, PKI can be defined as follows: a PKI is the basis of a pervasive security infrastructure whose services are implemented and delivered using public-key concepts and techniques [11].

PKI uses two types of cryptography algorithms to create certificates. Specifically, these are public key cryptography and digital signature algorithms. It uses public key cryptography (asymmetric cryptography) to generate public keys. The most popular mathematical techniques used to generate keys for public key cryptography implemented in PKI are Rivest-Shamir-Adleman (RSA) [13], Diffie-Hellman [12] and Elliptic curve cryptography (ECC) [14]. In addition it uses digital signature algorithms to generate signatures. These signature algorithms can be Digital Signature Algorithm (DSA) or Elliptic Curve Digital Signature Algorithm (ECDSA) [15] used with (Secure Hash Algorithm) SHA-2 or SHA-3 usually encrypted with RSA. The key sizes and hash functions used affect the security that a given implementation can provide.

There are several ways to demonstrate how PKI works. Depending on the application, there may be minor differences in the structure of the PKI. A variation is shown in Fig. 1, which shows the structure of a PKI used in digital authentication and communication, for example.

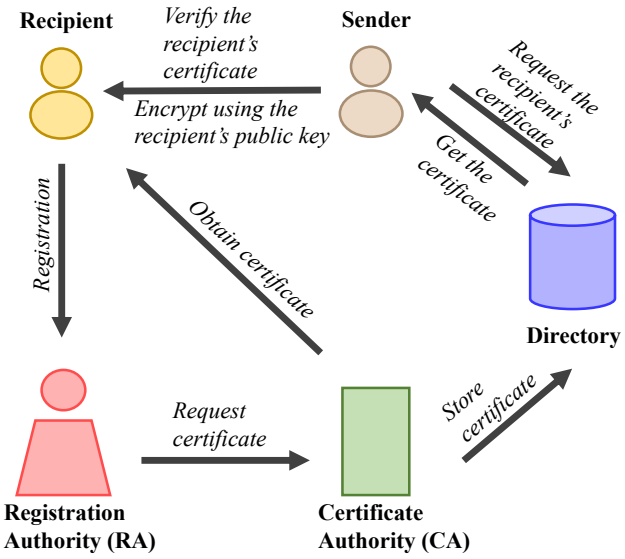

Fig. 1. Illustration of the structure of Public Key Infrastructure

First, the recipient requests a certificate with his public key from the Registration Authority (RA). The RA confirms the identity of the recipient and forwards the confirmation to the Certificate Authority (CA). The CA obtains the certificate for the recipient. The recipient can then digitally sign and identify himself with the certificate he has just received. Before the sender wants to send something to the recipient, it also retrieves the recipient's certificate. The certificates are stored in a directory, from where both the CA and the sender can retrieve them. After authenticating the recipient's certificate, the sender contacts the recipient using the recipient's public key for encryption.

Another PKI structure that can be used for digital signatures is as follows. A user applies to the Registration Authority for a certificate using his public key. The RA confirms the identity of the user to the Certification Authority, which then obtains the certificate. The user can then digitally sign a contract with his new certificate. The other party involved then verifies the identity of the user with a Validation Authority (VA). All parties obtain information about the issued certificates from the certification authority.

Depending on the specific features required by the application, the structure of the PKI may vary in small elements, but the basic structure and functionality, the basic authorities that make up the PKI, remain the same. To fully understand the reasons for these differences, we need to review the PKI applications, the different parties involved in the applications and the different requirements of these applications in each case.

B. Public Key Infrastructure Applications

PKI plays an important role in protecting many processes, such as communication or data transfer and all phases of product development and distribution in production environments. A lot of applications and connected devices are in use every day that require proper authentication and certification. For this reason, it is crucial to ensure the security of the techniques it uses [1].

One of the most important information security protocols based on PKI are used in the Secure Sockets Layer (SSL) and also in its modern version, Transport Layer Security (TLS). SSL/TLS certificates on websites made the move from Hypertext Transfer Protocol (HTTP) to Hypertext Transfer Protocol Secure (HTTPS) possible [10]. On websites using HTTP there was no verification step that could assure the user that they were communicating with the intended party. This made it easy to spoof or impersonate a website, which could be abused by cybercriminals. HTTPS was the solution. Certificates can be created for websites using PKI to ensure the authenticity of the site. This allows the user to check that the connection to the website is secure and even view the details of the certificate. Table. I gives examples of websites using the HTTPS protocol and the key generation algorithms and digital signature algorithms used on the websites.

PKI can also generate digital signatures for software. In an enterprise environment an important use of PKI is to provide

TABLE I
ALGORITHMS USED IN HTTPS CERTIFICATES

Website	Signature alg.	Public key
https://www.youtube.com/	SHA-256 with RSA	ECC
https://www.facebook.com/	SHA-256 with RSA	ECC
https://www.google.com/	SHA-256 with RSA	ECC
https://app.element.io/	X9.62ECDSA - SHA-256	ECC
https://github.com/	X9.62ECDSA - SHA-384	ECC
https://www.nist.gov/	SHA-256 with RSA	RSA (2048)
https://www.overleaf.com/	SHA-256 with RSA	RSA (2048)
https://www.reddit.com/	SHA-256 with RSA	RSA (2048)
https://account.proton.me/	SHA-256 with RSA	RSA (4096)
https://learn.microsoft.com/	SHA-384 with RSA	RSA (2048)
https://www.elte.hu/	SHA-384 with RSA	RSA (4096)

restricted access to enterprise intranets and VPNs. Email and data encryption is also a core PKI use case. Password-free Wifi access is secured by device ownership based authentication, which can be implemented with PKI.

PKI is also responsible for securing services such as signing documents, online shopping, transactions with credit cards, validating passports, encrypting documents and files and secure communication between IoT devices.

In every application the main goal is validation and to secure information transfer and communication.

III. QUANTUM COMPUTERS AND CRYPTOGRAPHY

To review the threats posed by quantum computers to cryptography, we first need to understand in detail the traditional cryptographic algorithms, how these algorithms work, and the mathematical problems on which their security is based.

A. Symmetric cryptography

In symmetric cryptography, the same key is used for the encryption and decryption process. This allows us to create simple and fast cryptographic methods. An important condition for symmetric cryptography is that the parties involved exchange keys in a secure way, because if the shared key is leaked to an eavesdropper, the encrypted message can be easily intercepted and modified. Popular symmetric key algorithms are Data Encryption Standard (DES) and Advanced Encryption Standard (AES). Asymmetric cryptography is a solution to the key distribution problem of symmetric cryptography [35].

B. Asymmetric cryptography

Asymmetric cryptography is also known as public key cryptography. This method requires two keys, a public key for encryption and a private key for decryption. The keys are mathematically related to each other to a certain extent, so they need to be much longer than in symmetric cryptography to ensure a sufficient level of security. Due to the complexity of the algorithms and the length of the keys, it will be slower than a symmetric algorithm.

The security of asymmetric cryptographic algorithms is based on complex computational and hard mathematical problems such as factorization of large prime numbers and the discrete logarithm problem [14] [35].

The most popular forms of public key cryptography are the RSA cryptosystem [13], ECC [14] and Diffie-Hellman key exchange [12]. These are also based upon hard mathematical problems. RSA is based on factoring large numbers. The scheme was invented in 1977 by Ronald Rivest, Adi Shamir, and Leonard Adleman. The algorithm is most often used to ensure key exchange, often with symmetric algorithms side by side, such as AES, where the symmetric algorithm performs the encryption and decryption [35].

ECC is based on computing discrete logarithms in groups of points on an elliptic curve defined over a finite field. Diffie-Hellman key exchange is one of the earliest practical examples of public key cryptography and the original implementation uses the multiplicative group of integers modulo n.

To solve these problems, we need to be able to access or compute the secret information for these algorithms, the cryptographic keys, which in general are large numbers. Without knowing this information, it is impossible to decode the encrypted data or create signatures so we can authenticate ourselves.

The hard mathematical problems that are in use in public key cryptography are in NP. The proof of problems in NP can be verified in polynomial time. The key distribution problem is computationally easy, for example the multiplication of two large numbers, but obtaining the private key as finding the prime factors of a given large number is considered to be computationally hard. Classical computers cannot solve these hard mathematical problems in reasonable time, however quantum computers have enough computational power to solve these hard problems much faster [32].

Quantum computers can solve difficult mathematical problems in seconds [16]. This would allow them to solve such problems as integer factorization and discrete logarithm problem much faster than their classical counterparts. The rise of quantum computers with sufficient computing power is therefore challenging the security of classical cryptographic algorithms [32]. Popular and widely deployed tools for cryptographic protection, which are now fully trusted, would be easily broken.

Shor introduced an efficient polynomial-time algorithm in 1994 [16] which was made for solving integer factorization and discrete logarithm problems. If a quantum computer with sufficient computing power could run Shor's algorithm, then it could be used to break public-key cryptography schemes, such as RSA, ECC and Diffie-Hellman key exchange.

In 1996 Grover introduced an algorithm which was made to search unsorted databases and can be run on quantum computers [36]. The algorithm can search for a specific record in an unsorted database containing N records and can

identify that record in $O(\sqrt{N})$ steps [36]. This algorithm has been used to break symmetric cryptographic algorithms [35] such as the Data Encryption Standard (DES). It also poses a threat to cryptographic hash functions.

C. Cryptographic hash functions

Cryptographic hash functions are used to transform data. A hash function is an operation that produces a fixed-length sequence of bits from a sequence of bits of arbitrary length. The hash value can be used, for example, to check the authenticity of data. An important concept is collision. The set of possible inputs to a hash function is infinite, and the set of possible hash values is finite. This follows from the fact that the input can be a sequence of bits of arbitrary length and the output can be of fixed size. Because of this, there is an infinite number of bit sequences with the same hash value, a phenomenon called collision. Cryptographic hash functions must satisfy the requirement that the hash function used is unidirectional, i.e. it must be algorithmically easy to compute the hash value of a message, while it must be algorithmically difficult to find a message with a given hash value for the hash value (preimage resistance). In addition, a good hash function must also provide collision resistance. This means that it should be algorithmically difficult to find two different messages with the same hash value.

Since hash functions work with fixed-length outputs, finding collisions can be done quickly on a computer with high computing power. It is like searching a large unsorted database [35]. Thus, using for example Grover's algorithm on quantum computers, attacks against hash functions could be performed in real time, which poses a real threat for hash functions with shorter output lengths, but for more complex hash functions with longer output lengths, even quantum computers cannot achieve a break within a reasonable time. Thus, SHA-2 and SHA-3 also provide a quantum-safe solution with longer outputs [35].

IV. Quantum threat for Public Key Infrastructure

The public key infrastructure uses asymmetric cryptography and digital signature algorithms to operate. The security of these relies on hard computational problems and hash functions. The threat of integer factorization and discrete logarithm problems being solved efficiently by quantum computers by Shor's algorithm and Grover's algorithm on symmetric cryptography and hash functions poses a serious threat to the PKI techniques in use today.

Currently used PKI schemes are mostly based on non-quantum-resistant cryptographic mechanisms. The most commonly implemented public key algorithms are RSA and ECC. These algorithms can provide high security against attacks made by traditional computers. However they can be easily broken with quantum algorithms (these are algorithms that rely on the existence of quantum computers). So the invention and deployment of quantum-proof techniques become significantly important for many computer and information systems.

V. Making Public Key Infrastructure quantum-safe

A. Post-Quantum Cryptography algorithms

In recent years, serious efforts have been made to develop cryptographic algorithms that can withstand quantum computer's threats. Different post-quantum cryptography (PQC) schemes were made such as hash-based, lattice-based, isogeny-based, code-based and multivariate cryptography schemes. A few years ago National Institutes of Standards and Technology (NIST) initiated a process to select quantum-resistant public-key cryptographic algorithms for standardization. The purpose of this process was to create new cryptography standards for digital signature, public-key encryption, and key-establishment algorithms that are capable of protecting sensitive data after the advent of quantum computers as well [5]. After the third round of evaluation and analysis of the candidates NIST announced the first selected algorithms to standardize. After that the Round 4 of the NIST PQC Standardization Process took place in summer of 2022 with following key exchange mechanisms still under consideration. In the fourth round only key exchange mechanisms have participated in, and there were no digital signature candidates remaining. NIST has therefore announced a new call for digital signature algorithms for the PQC standardization process. The call for submissions closed on June 1, 2023, and the analysis and selection of submitted candidates is expected to take several years.

This is a long and labor-intensive process, and in order to use these schemes in real solutions, we need to be up-to-date with the information about them.

B. Quantum-resistant solutions for Public Key Infrastructure

Post-quantum cryptographic (PQC) systems can provide a good solution against broken cryptosystems. So the final goal in the quantum-resistant PKI creation process is switching from classical asymmetric cryptographic algorithms to PQC algorithms. However this upgrade takes time, effort and resources. Thus, the first major step in ensuring the information and cybersecurity in the post-quantum era can be the use of hybrid digital certificates. In addition it also gives motivation for the creation and use of hybrid schemes, that the recently developed PQC techniques have not been studied long enough, successful attacks can be introduced at any time, and thus run the risk of being insecure.

In a hybrid solution a traditional algorithm and one or more post-quantum algorithms are used in parallel. This ensures that as long as one of them remains unbroken, confidentiality or authenticity can be ensured. Such hybrid solutions were already presented for PKI and some of them are also in

commercial use.

It is also important to consider that post-quantum algorithms have different storage and resource requirements than classical algorithms. They work with significantly larger private and public keys, which need to be stored and managed by the system in which they are implemented. In addition, since these schemes derive their security from mathematical and coding theory hard problems, their implementation is more computationally intensive and time consuming. There can be order of magnitude differences in their complexity. Some applications using PKI today would have to be very heavily redesigned to use PQC algorithms, other applications are ready for quantum-safe algorithms even in their current form.

The PKI used in the TLS protocol is particularly vulnerable to the threats of quantum computers, but also is well suited to quantum-proof algorithms and hybrid solutions to compensate for it. TLS makes it possible to easily integrate post-quantum algorithms even working with large keys, because TLS data structures allow certificates of size up to 2^{24} bytes [7]. Post-quantum key exchange has already been introduced in negotiation supporting Transport Layer protocols and they have already used methods in commercial. TLS is suitable for hybrid solutions and these methods can be easily adopted. The use of such a PQC key exchange can be developed to other PKI solutions as well.

One of the first projects to implement quantum-safe cryptographic solutions was OpenSSL, which is an open-source implementation of the TLS protocol. OpenSSL has hybrid solutions integrated in its operation and is under continuous development.

Since OpenSSL is an open-source project, there are multiple integration proposals for post-quantum authentication from different contributors. A well-designed solution is the integration of XMSS (eXtended Merkle Signature Scheme) [18] hash-based signature scheme into OpenSSL [8]. A few years before that Bos et al. integrated Ring-LWE (ring learning with errors) [17] based key exchange protocol as an additional cipher-suite into OpenSSL [4].

The Open Quantum Safe (OQS) is a project that has been running since 2017 and aims to support the development and prototyping of quantum-resistant cryptography. The project consists of two main parts, one is liboqs, which is an open source C library for quantum-resistant cryptographic algorithms, and the other one is the prototyping of prototype integrations into protocols and applications, including OpenSSL [19]. The integration of several quantum-resistant key exchange algorithms such as Ring-LWE, LWE, NTRU and SIDH (Supersingular Isogeny Diffie–Hellman), etc. has been demonstrated.

VI. Overview of algorithms for quantum-resistant Public Key Infrastructure

After three rounds of the NIST PQC standardization process, the first algorithms to be standardized have been selected. These included CRYSTALS-KYBER as a key encapsulation mechanism (KEM) and three digital signature algorithms, CRYSTALS-Dilithium, FALCON, and SPHINCS+. These schemes are all based on the hardness of lattice-based computational problems, except the SPHINCS+ scheme, which is a stateless hash-based signature scheme.

Four KEM algorithms were nominated for the fourth round of the NIST PQC standardization process, these were BIKE, Classic MCEliece, HQC and SIKE. These are algorithms that were not standardized after the third round, but showed enough promise to be subjected to further analysis and testing, with further modifications to allow teams to re-submit them.

It is therefore worth reviewing the most promising PQC algorithms currently available and take under consideration all the standardized candidates and the candidates from the fourth round.

Since our ultimate goal is to replace the traditional cryptographic algorithms used in PKI with quantum-safe algorithms, let's review a comparison between them to get a proper picture of the applicability of PQC algorithms to PKI.

For comparison, all the schemes selected for standardization and other Round 3 and Round 4 submissions have been selected, Public-key Encryption and Key-establishment Algorithms and Digital Signature Algorithms respectively.

For the evaluation of cryptographic schemes, NIST has defined security levels that the algorithms meet with different parameter sets. The levels are defined as described in Table. II.

TABLE II
THE SECURITY LEVELS OF NIST

Level	At least as hard to break as
Level 1	AES128
Level 2	SHA256
Level 3	AES192
Level 4	SHA384
Level 5	AES256

Besides the PQC algorithms, we should mention the eXtended Merkle Signature Scheme (XMSS), which is a hash-based digital signature system. XMSS uses a one-time signature scheme as its main building block. The scheme relies on cryptographic hash functions to provide strong security for cryptographic digital signatures without relying on the conjectured hardness of mathematical problems. The scheme remains safe even if the collision resistance of the underlying hash function is broken. The scheme is suitable for compact implementations, the implementation is relatively simple and is naturally resistant to side-channel attacks. Because it is hash-based, it can withstand known attacks

using quantum computers [44].

SIKE (Supersingular Isogeny Key Encapsulation) [37] is an isogeny-based key encapsulation scheme based on pseudo-random walks in supersingular isogeny graphs. SIKE was a promising candidate for standardization because it has small key and ciphertext sizes. However, further study of the scheme [38] [39] [40] showed that the algorithm is insecure. Because of this, the authors issued an announcement that the algorithm is insecure and should not be used, so we do not include it in our further analysis.

BIKE (Bit Flipping Key Encapsulation) is a code-based key encapsulation mechanism based on QC-MDPC (Quasi-Cyclic Moderate Density Parity-Check) codes. BIKE is built upon the Niederreiter framework [41], with some tweaks. It also applies the implicit-rejection version of Fujisaki-Okamoto transformation [42] for converting a δ-correct PKE into an IND-CCA (Indistinguishability under chosen ciphertext attack) KEM.

Its variants have different security levels. The version under review has security level 5, which is corresponding to the security of AES-256 (Advanced Encryption Standard) [20].

The first code-based public-key cryptosystem was introduced by McEliece in 1978. The public key specifies a random binary Goppa code and a ciphertext is a codeword plus random errors. The private key allows efficient decoding: extracting the codeword from the ciphertext, identifying and removing the errors [22]. The McEliece system was designed to be one-way (OW-CPA), which means that the codeword cannot efficiently be found from a ciphertext and public key, when the codeword is chosen randomly [22]. This KEM is built from Niederreiter's [41] dual version of McEliece's algorithm using binary Goppa codes. It is a KEM designed for IND-CCA2 (Indistinguishability under adaptive chosen ciphertext attack) security at a very high security level.

If the parameters are well chosen then it is effective against quantum computers as well. A variant with a smaller parameter set was chosen (mceliece348864), because with larger parameters the public key sizes are over one million bytes and the private keys are larger than thirteen thousand bytes. Managing keys of this size is a serious task.

HQC (Hamming Quasi-Cyclic) is a code-based public key encryption scheme as well. The KEM provides IND-CCA2 security. The main advantages of the scheme are relatively small public key size and efficient implementations based on classical decoding algorithms [23]. For the chosen variant, HQC-256 the ciphertext is generated deterministically from a seed of 256 bits [23].

CRYSTALS-Kyber is an IND-CCA2-secure key encapsulation mechanism. The security of Kyber is based on the hardness of solving the learning-with-errors problem in module lattices [24]. Lattices have strong security proofs based on worst-case hardness, the hard problems in lattice theory can provide the security for various cryptosystems and lattices have efficient implementations, therefore lattice-based cryptographic methods hold great potential for PQC algorithms [33]. The authors recommend using the Kyber-768 parameter set, which achieves more than 128 bits of security against all known classical and quantum attacks.

NTRU is also a lattice-based public-key cryptosystem. The chosen variant is NTRU-HPS, which uses fixed-weight sample spaces and allows several choices of q for each n, with parameters $q = 4096$ and $n = 821$. In the algorithm n is a fixed odd prime, and we are working over a multiplicative group of integers modulo n and q is a power of 2 [26].

CRYSTALS-Dilithium [29], Falcon [27] and SPHINCS+ [28] are the digital signatures that were selected to be standardized by NIST. All these schemes are based on the computational hardness of problems on lattices [5] [33], except the SPHINCS+ scheme, which is a stateless hash-based signature scheme.

CRYSTALS - Dilithium is strongly secure under chosen message attacks (IND-CCA security) and is based on the hardness of lattice problems over module lattices [29]. Dilithium has the smallest public key and signature size of any lattice-based signature scheme that only uses uniform sampling [29]. Following the recommendations of the scheme's authors, we choose the Dilithium3 parameter set for comparison, as it achieves more than 128 bits of security against all known classical and quantum attacks [29].

Falcon is based on the theoretical framework of Gentry, Peikert and Vaikuntanathan for lattice-based signature schemes [27]. This framework is made over NTRU lattices with a trapdoor sampler called "Fast Fourier sampling". The underlying hard problem is the short integer solution problem (SIS) over NTRU lattices [27]. Advantages of the scheme are shorter signature sizes than in any other lattice-based signature scheme with the same security guarantees although the public keys are around the same size. Another advantage is that the use of the fast Fourier sampling allows for very fast implementations [27].

SPHINCS+ is a stateless hash-based signature scheme. The design advances the original SPHINCS signature scheme with multiple improvements, specifically aimed at reducing signature size [28]. In the SPHINCS+ scheme there is an XMSS private key, which is the only secret seed included in the SPHINCS+ secret key. The advantages of the SPHINCS+ scheme include that it does not introduce a new intractability assumption, state-of-the-art attacks against it are easily analyzed, the key sizes are relatively small, there is an overlap with XMSS and the scheme reuses established building blocks. The security of SPHINCS+ is solely based on assumptions about the used hash function [28]. However

main disadvantages are big signature sizes and the speed of the algorithms. The selected variant is SPHINCS+-256s which provides level 5 security.

These algorithms were designed to provide security against attacks by both classical and quantum computers. The existing variants are the same algorithm with different parameterization. In general, the parameters are given in the name of the variant. The selected variants were chosen with the existing strongest or a generally strong parameterization. Wherever possible, sets of parameters for the selected variants address security level 5 as defined by NIST [30].

The data for the schemes in Table. III, Table. IV and Fig. 2 below was provided by the original documents made by the teams who developed the method of BIKE [21], Classic McEliece [22], HQC [23], NTRU [26], CRYSTALS-Kyber [25], CRYSTALS-Dilithium [29], Falcon [27] and SPHINCS+ [28].

Table. III contains the essential details of the algorithms such as the selected variant, the type and structure of the schemes.

The following abbreviations are used in the tables: asymm. - asymmetric, KE - key exchange, KEM - key encapsulation mechanism, KES - key encryption scheme DS - digital signature, key gen. - key generation, C - ciphertext, S - signature.

TABLE III
ESSENTIAL DETAILS OF THE ALGORITHMS

Algorithm	Variant	Type	Structure
Diffie-Hellman	-	asymm.	KE
RSA	-	asymm.	KE
ECC	-	asymm.	KE
BIKE	sec. level 5	code-based	KEM
Classic McEliece	mceliece348864	code-based	KES
HQC	hqc-256	code-based	KES
Kyber	Kyber-768	lattice-based	KEM
NTRU	ntruhps4096821	lattice-based	KEM
Dilithium	Dilithium3	lattice-based	DS
Falcon	Falcon-1024	lattice-based	DS
SPHINCS$^+$	SPHINCS$^+$-256s	hash-based	DS

Table. IV contains the data needed to compare the algorithms such as public key, private key and ciphertext or signature sizes.

Key sizes are given in bits (B) or bytes in all cases. Only bits are marked with a B, when the size is given in bytes, only the numbers are shown. The last column by digital signature algorithms contains the signature size and by public-key cryptosystems the size of the ciphertext.

It can be clearly seen from the table by comparing the key sizes that the keys of the PQC algorithms are in most cases much larger in size than the keys of conventional algorithms. It is also immediately apparent from the tables that the key sizes of the Classic McEliece scheme are orders of magnitude larger than those of the other algorithms. For this reason, it will not be included alongside the other Public-key Encryption / KEM schemes in the further aggregate analysis. We also can observe that there are significant differences even between the key sizes and signature sizes of the post quantum cryptographic algorithms under investigation.

TABLE IV
COMPARISON OF CLASSICAL AND PQC ALGORITHMS

Algorithm	Public key	Private key	C / S
Diffie-Hellman	2048 to 4096B	2048 to 4096B	-
RSA	2048 to 4096B	2048 to 4096B	-
ECC	256B / 384B	256B / 384B	-
BIKE	40973B	4640B	C: 41229B
Classic McEliece	261120	6492	C: 96
HQC	7245	72	C: 14485
Kyber	1184	2400	C: 1088
NTRU	1230	1592	C: 1230
Dilithium	1952	-	S: 3293
Falcon	1793	-	S: 1280
SPHINCS$^+$	64	128	S: 29792

A more visual comparison of the most promising Public-key Encryption / KEM algorithms listed is shown in Fig. 2. As can be seen, there are significant differences in key size and ciphertext size between the algorithms.

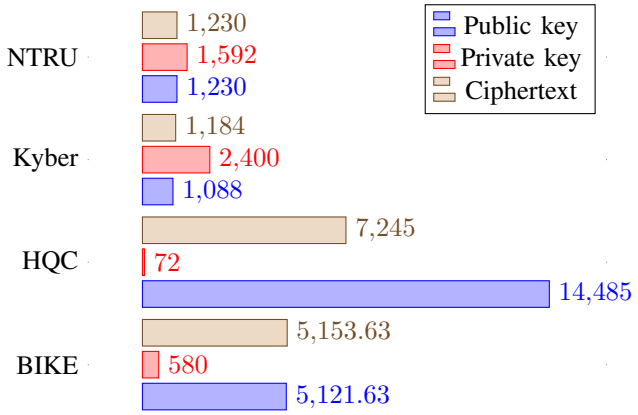

Fig. 2. PQC algorithms key sizes in bytes

VII. CONCLUSION

The presented variants of the algorithms are those that can provide the highest security with implementation at the lowest possible cost. However, they are all considered to be high cost in terms of memory usage. When implementing them into a real application, one can also consider their variants with different key sizes. These variants even with the smallest key sizes and lowest security level are also an improvement over the current conventional algorithms and can provide adequate security against quantum computers in the short term. However, in the longer term, it is worthwhile to develop

systems using PKI in such a way that even the most storage-consuming PQC algorithms can be integrated. To achieve this, the key size values shown in the table can provide some guidance.

Classic McEliece is the least suitable for implementing in PKI since the generated keys are of an outstandingly large size. Most solutions using PKI cannot afford the storing and managing keys which require this amount of memory. Nevertheless, Classic McEliece is a long-established and reliable scheme, and therefore there are applications where it has been already implemented, and where the regulations make its use highly recommended despite the costly implementation. This is a good example of how reliability is often worth prioritizing over cost-effectiveness.

If we are looking for an efficient and cost-effective solution, it can be said that Kyber can be one of the best solutions. Its reliability is also proven by the fact that it was the first to be selected for standardization within the NIST PQC standardization project. However, the NTRU and the Kyber algorithms are both particularly promising, as they both achieve level 5 security and their key sizes are relatively small, so that using them in a hybrid way, e. g. with RSA, does not require a significant increase in storage space, but results in a significantly more secure solution.

Among the digital signature algorithms, Falcon and Dilithium are the most promising candidates, SPHINCS+ is working with large keys and signature sizes and is not the fastest algorithm, thus losing out to the other two candidates.

Furthermore, when moving to quantum secure solutions, the integration of traditional hash-based schemes such as XMSS should be considered. These have the great advantage that they are suitable for compact implementations, are relatively simple to implement, and naturally resist known attacks using quantum computers.

A. Further security recommendations

PKI applications can be secured in many ways. In addition to using quantum-proof cryptographic algorithms instead of traditional ones, there are many other recommendations that can be considered by implementing PKI to mitigate the threat from quantum computing.

The use of certificates with X.509 format makes it easier to develop hybrid solutions. Certificates with X.509 format support an extensible schema for embedded data. This allows keys from different algorithms to be used in the same certificate. This is particularly useful when hybrid solutions are desired, as it allows the keys of a traditional and a PQC algorithm to be stored side by side [1].

Use of SHA-2 hash algorithms instead of SHA-1 algorithms also provides a higher level of security. There are multiple organizations that already have a policy for deprecating SHA-1 algorithms in SSL and code signing certificates, for example Microsoft.

Key lengths are also something that should be considered well. Long keys that comply with security standards should be used, but the amount of storage and resources that can be used must also be taken into account.

Applying universal security requirements for software can make it easier to use and update certificates and cryptographic algorithms, and lead to easier integration [1].

In addition keeping certificates and the algorithms they use up to date is also an important element of secure PKI applications. We should always think ahead when designing our systems and their security, as it is easier and faster to update the systems if we have the right conditions in place.

VIII. Future works

Several papers on how to make PKI quantum-safe have already been prepared, and the threats, algorithms and solutions have been analyzed in detail. However, practical solutions, such as integrating post-quantum algorithms into specific PKI applications, have been less developed. To ensure the security of PKI in the near future and in the era of quantum computers, more practical solutions need to be developed and introduced that can be easily adapted into real world PKI applications.

However, in addition to post-quantum cryptography, it is also possible to use encryption using quantum computers. Quantum computers themselves offer the possibility of creating new encryption methods that also provide protection against quantum computers. The use of these algorithms to complement or replace PKI may also be a worthwhile topic for investigation.

Acknowledgment

This research was supported by Project no. TKP2021-NVA-29 which has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme. This research was also supported by the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary (NKFIH-873-1/2020).

References

[1] Yunakovsky, Sergey E., et al. "Towards security recommendations for public-key infrastructures for production environments in the post-quantum era." *EPJ Quantum Technology* 8.1 (2021): 14.

[2] Raavi, Manohar, et al. "Performance characterization of post-quantum digital certificates." *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021.

[3] Bindel, Nina, et al. "Transitioning to a quantum-resistant public key infrastructure." *Post-Quantum Cryptography: 8th International Workshop*, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8. Springer International Publishing, 2017.

[4] Alkim, Erdem, et al. "Post-quantum key Exchange — A new hope." *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[5] Alagic, Gorjan, et al. "Status report on the third round of the NIST post-quantum cryptography standardization process." *US Department of Commerce*, NIST, 2022.

[6] Grote, Olaf, Andreas Ahrens, and César Benavente-Peces, "A review of post-quantum cryptography and crypto-agility strategies." *2019 International Interdisciplinary PhD Workshop (IIPhDW)*. IEEE, 2019.

[7] Bindel, Nina, et al. "Transitioning to a quantum-resistant public key infrastructure." *Post-Quantum Cryptography: 8th International Workshop*, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8. Springer International Publishing, 2017.

[8] Butin, Denis, Julian Wälde, and Johannes Buchmann, "Post-quantum authentication in OpenSSL with hash-based signatures." *2017 Tenth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 2017.

[9] Hunt, Ray, "PKI and digital certification infrastructure." *Proceedings. Ninth IEEE International Conference on Networks*, ICON 2001. IEEE, 2001.

[10] Davies, Joshua, "Implementing SSL/TLS using cryptography and PKI" *John Wiley and Sons*, 2011.

[11] Adams, Carlisle, and Steve Lloyd, "Understanding PKI: concepts, standards, and deployment considerations" *Addison-Wesley Professional*, 2003.

[12] Diffie, W. and Hellman, M. E., "New Directions in Cryptography" *IEEE Transactions on Information Theory*, 22 (1976), pp. 644-654.

[13] Rivest, Ronald L., Adi Shamir, and Leonard Adleman, "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM* 21.2 (1978): 120-126.

[14] Koblitz, Neal, Alfred Menezes, and Scott Vanstone, "The state of elliptic curve cryptography." *Designs, codes and cryptography* 19.2 (2000): 173-193.

[15] Johnson, Don, Alfred Menezes, and Scott Vanstone, "The elliptic curve digital signature algorithm (ECDSA)." *International journal of information security 1* (2001): 36-63.

[16] Shor, Peter W, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer." *SIAM review* 41.2 (1999): 303-332.

[17] Lyubashevsky, Vadim, Chris Peikert, and Oded Regev, "On ideal lattices and learning with errors over rings." *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010. Proceedings 29*. Springer Berlin Heidelberg, 2010.

[18] J. Buchmann, E. Dahmen, and A. Hülsing, "XMSS — A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions," *PQCrypto, ser. LNCS*, vol. 7071. Springer, pp. 117–129., 2011.

[19] Stebila, Douglas, and Michele Mosca, "Post-quantum key exchange for the internet and the open quantum safe project." *International Conference on Selected Areas in Cryptography*. Cham: Springer International Publishing, 2016.

[20] Aragon, Nicolas, et al. "BIKE: bit flipping key encapsulation", 2017.

[21] Aragon, Nicolas, et al. "BIKE: bit flipping key encapsulation - Round 4 Submission", 2022.

[22] Chou, Tung, et al. "Classic McEliece: conservative code-based cryptography", 2020.

[23] Melchor, Carlos Aguilar, et al. "Hamming quasi-cyclic (HQC)" *NIST PQC Round*, 2022.

[24] J. Bos et al. "CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM," *2018 IEEE European Symposium on Security and Privacy (EuroS&P), London, UK*, pp. 353-367, doi: 10.1109/EuroSP.2018.00032., 2018.

[25] J. Bos et al. "CRYSTALS-Kyber - Algorithm Specifications And Supporting Documentation (version 3.0)", 2020.

[26] Chen, Cong, et al. "NTRU-Algorithm specifications and supporting documentation (Round 3 Submission)" Tech. Rep. 2020.[Online]. Available: https://csrc. nist. gov/Projects/post-quantum-cryptography/round-3-submissions, 2020.

[27] Fouque, Pierre-Alain, et al. "Falcon: Fast-Fourier lattice-based compact signatures over NTRU." *Submission to the NIST's post-quantum cryptography standardization process*, 2020.

[28] Jean-Philippe Aumasson, Daniel J. Bernstein, et al. "SPHINCS+" *Submission to the NIST post-quantum project*, v.3.1, 2022.

[29] Lyubashevsky, Vadim, et al. "CRYSTALS-Dilithium: algorithm specifications and supporting documentation." *NIST Post-Quantum Cryptography Standardization Round 3*, 2020.

[30] NIST, "Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process.", 2016.

[31] Maurer, Ueli, "Abstract models of computation in cryptography." *Cryptography and Coding: 10th IMA International Conference, Cirencester, UK, December 19-21, 2005. Proceedings 10*. Springer Berlin Heidelberg, 2005.

[32] Aumasson, Jean-Philippe, "The impact of quantum computing on cryptography." *Computer Fraud & Security* 2017.6 (2017): 8-11., 2017.

[33] Regev, O., "Lattice-based cryptography" *In Advances in cryptology (CRYPTO)*, pages 131–141, 2006.

[34] Bene Fruzsina, and Attila Kiss, "Public Key Infrastructure in the Post-Quantum Era." *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)*. IEEE, 2023.

[35] Mavroeidis, Vasileios, et al. "The impact of quantum computing on present cryptography." *arXiv preprint* arXiv:1804.00200, 2018.

[36] Grover, Lov K, "A fast quantum mechanical algorithm for database search." *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996.

[37] Campagna, Matthew, et al. "Supersingular isogeny key encapsulation." 2019.

[38] Damien Robert, "Breaking SIDH in polynomial time" *Cryptology ePrint Archive*, Report 2022/1038, 2022.

[39] Luciano Maino and Chloe Martindale, "An attack on SIDH with arbitrary starting curve" *Cryptology ePrint Archive*, Report 2022/1026, 2022.

[40] Tomoki Moriya, "Masked-degree SIDH" Cryptology ePrint Archive, Report 2022/1019, 2022.

[41] Harald Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory" *Problems of Control and Information Theory*, 15(2):159–166, 1986.

[42] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz, "A modular analysis of the Fujisaki-Okamoto transformation" Theory of Cryptography Conference, pages 341–371. Springer, 2017.

[43] Dam, Duc-Thuan, et al. "A Survey of Post-Quantum Cryptography: Start of a New Race." *Cryptography* 7.3 (2023): 40.

[44] Buchmann, J., Dahmen, E., and A. Huelsing, "XMSS - A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions" *Lecture Notes in Computer Science*, Volume 7071, Post-Quantum Cryptography, doi: 10.1007/978-3-642-25405-5_8, 2011.