Post-Quantum Cryptography - An Overview of Standards, Protocols, and Practical Applications

Jörn-Marc Schmidt
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: joern-marc.schmidt@iu.org

Alexander Lawall
IU International University of Applied Sciences
Erfurt, Thüringen, Germany
e-mail: alexander.lawall@iu.org

***Abstract*—In cryptographic security, quantum computing poses a significant challenge to traditional cryptographic protocols. This study investigates the landscape of Post-Quantum Cryptography (PQC), focusing on the transition from theoretical underpinnings, over standardization efforts to practical implementations. The primary research question that guides this contribution is: What mechanisms can be implemented to safeguard applications and what efforts are under way by application providers and technology platforms? This question is answered by the current state of standards supporting PQC and the ongoing preparation efforts. Thereby, not only the standards for cryptographic algorithms, but also the protocols relying on them are considered. Furthermore, the status of (open-source) implementations is reviewed and roadmaps from companies / technology providers are discussed. Hence, this paper does not only discuss what a company can do to protect their applications but also takes the viewpoint of an end-user regarding the support of applications.**

Keywords-Post Quantum Cryptography (PQC); PQC Standards; PQC Implementations; PQC Libraries; Technology Roadmaps.

I. Introduction

This work is an extended version of *Theoretical and Practical Aspects in Identifying Gaps and Preparing for Post-Quantum Cryptography*, published at SECURWARE 2024 [1].

Quantum computers will influence many fields. They will improve biological and chemical simulations, can be applied for risk modeling, and improve solving of optimization problems. In addition to those constructive improvements, they have the potential to impact the security of cryptographic algorithms. Especially, asymmetric algorithms that rely on the hardness of factorization or the discrete logarithm problem cannot be considered secure when a Cryptographic Relevant Quantum Computer (CRQC) is available. Hence, use cases relying on such algorithms will be impacted by CRQCs. Moreover, even data transmitted today can be endangered by attackers recording the transmission and decrypting it as soon as CRQCs are available. This is referred to as harvest now and decrypt later attack.

Quantum computers also have the potential to impact symmetric cryptography. However, it is assumed that such attacks can be addressed by using larger keys for symmetric encryption [2]. For example, the National Security Agency (NSA) states that the Advanced Encryption Standard (AES) can be considered secure when used with 256-bit keys [3]. Hence, the remainder of this paper will focus on asymmetric cryptography.

This paper is organized as follows: It starts with a brief discussion of the current state of the art in Section II. Section III discusses the general preparation process and security protocols. Section IV summarizes the status of the standardization of new cryptographic algorithms, while Section V looks into the status of protocol standards. Libraries that support Post-Quantum Cryptography (PQC) algorithms, as a foundation for implementations, are presented in Section VI. Section VII discusses applications that are available for endusers and Section VIII discusses the possibilities to use quantum mechanisms to improve security. Finally, conclusions are drawn in Section IX.

II. State of the art

The challenge of ensuring Post-Quantum (PQ) security is already picked up by security researchers, developers, several government agencies, and companies. In order to drive the readiness of post-quantum cryptographic algorithms and their adoption in standard applications forward, many activities are underway. They include various working groups, like the Internet Engineering Task Force (IETF) working group *Post-Quantum Use In Protocols* [4], and the European Telecommunications Standards Institute (ETSI) *Quantum-Safe Cryptography (QSC)* working group [5]. Further activities are driven by various companies like Google [6], IBM [7], Microsoft [8], and Utimaco [9].

This paper provides an overview of those activities. Thereby, its focus is on use cases for asymmetric cryptography due to the expected high impact of CRQC on this type of algorithm. The paper highlights the status of standardization processes and the production-readiness of implementations. As such, it demonstrates what is currently done by different organizations, and gives guidance on what can be done today to protect own applications and data.

III. Use Cases of cryptography

The transition to post-quantum cryptography, given the widespread use of the algorithms, is a huge undertaking. As a first step, it is important to understand where susceptible algorithms are employed and how valuable the protected data is. Hence, for a company to prepare, a risk assessment of its application portfolio is required. The first step in such an endeavor is creating a cryptographic inventory, providing insights on where algorithms and protocols are used, together

with related parameters. Various tools can help creating an inventory [10].

Afterwards, a sound risk model that integrates into the company's risk management procedures is required. For the financial industry, for example, the Financial Services Information Sharing and Analysis Center (FS-ISAC) provides a white paper on modeling the risk [11]. This helps to create a profound strategy and to decide where the highest risks and the biggest benefits are expected. Finally, a maturity index helps judging and comparing where a company is on its journey to post-quantum security [12][13].

Generally speaking, data requires protection at rest, in transit, and in use.

Data at rest commonly relies on symmetric cryptography, where limited impact of quantum computers is expected. Solutions that employ asymmetric cryptography can make use of Key Encapsulation Mechanisms (KEMs) discussed in Section IV.

Encryption of data in use is not yet widely used. An available possibility is to rely on processor extensions like Intel Software Guard Extensions (SGX) [14] / Trust Domain Extensions (TDX) [15] or AMD Secure Encrypted Virtualization (SEV) [16]. Especially the attestation, i.e., proving that the protected environment is in a trustworthy state, relies on asymmetric cryptography. Solutions are discussed in [17].

In particular, when focusing on harvest now and decrypt later attack scenarios, security of encryption in transit against attacks with quantum computers is the most pressing scenario. In order to protect data in transit, it is possible to

- protect the underlying infrastructure by ensuring that the communication is PQ-secure. While this has large impact, it is restricted to endpoints that are in direct control; protecting the connections to end-users might not be possible. Commonly, protocols like IPsec and Media Access Control security (MACsec) are employed in such scenarios.
- ensure that the communication protocols are PQ-secure. Common protocols are Transport Layer Security (TLS) and Secure Shell (SSH). Both protocols allow to negotiate the used ciphers with a handshake. This enables using PQC whenever both parties support it without preventing non-PQC-secure communication in case one endpoint is not able to use such a cipher.
- encrypt the transferred message in a quantum-secure way. By using a method that ensures that the data is encapsulated with post-quantum cryptography, sound protection against adversaries can be achieved. This can be done either via standards suitable to the application, like Secure/Multipurpose Internet Mail Extensions (S/MIME) for emails/webpages, JavaScript Object (JSON) Signing and Encryption (JOSE)/Concise Binary Object Representation (CBOR) Object Signing and Encryption (COSE) for messages between applications and Pretty Good Privacy (PGP) for encrypting arbitrary data including files. Another option is to rely on self-defined, custom protocols, e.g., by employing implementations discussed in Section VI directly.
- rely on plattforms and services that use PQC for protecting data or at least have a clear roadmap regarding PQC-migration. Especially in cloud environments, it is not always required that platform-users implement security mechanisms themselves, but it is possible to rely on services delivered by the could provider.

A common requirement that is independent of the layer where data protection is applied, is ensuring as sound authentication of the communication partners and the authenticity of the data. Related methods are required as soon as a CRQC is available. Collecting data today, as in the harvest now and decrypt later scenario, does not represent a current threat. However, a lack of being ready in time will have devastating consequences as well, as an adversary can impersonate every identity that is not protected and forge any non-PQC signature. While details of different protocols on how to achieve a secure authentication vary, many make use of certificates issued by a Public Key Infrastructure (PKI). In essence, issuing a PQ-secure certificate requires a PQ-secure signature algorithm. However, there are many processes around a secure PKI and different ways of integrating a PQ-secure signature into a certificate. As it can be the foundation for critical processes like TLS authentication, for re-signing documents, including contracts, and for secure authentication of devices, it is a critical aspect of the PQ-migration as well.

The different options that are discussed in Section V are shown in Figure 1. In addition, a few examples for end-user applications are discussed in Section VII.

IV. The Quest for new Cryptographic Algorithms

The basis of all protocols and building blocks is quantum-secure algorithms. Hence, it is essential to develop and standardize new (asymmetric) cryptographic algorithms to replace the current ones.

A key activity in this regard was launched by National Institute of Standards and Technology (NIST) end of 2016. The NIST issued a call for papers for new post-quantum cryptographic algorithms [18]. Out of 69 initial submissions, three were selected to become Federal Information Processing Standards (FIPS). The following documents have recently (at the time writing this paper) been finalized:

- FIPS 203, Module-Lattice-Based Key-Encapsulation Mechanism Standard (ML-KEM), based on Cryptographic Suite for Algebraic Lattices (CRYSTALS)-Kyber [19]
- FIPS 204, Module-Lattice-Based Digital Signature Standard (ML-DSA), based on CRYSTALS-Dilithium [20]
- FIPS 205, Stateless Hash-Based Digital Signature Standard (SLH-DSA), based on SPHINCS+ (for practical stateless hash-based signatures) [21]

Moreover, the process is continuing with a fourth round. The remaining candidates are the Key-Encapsulation Mechanisms (KEMs) Bit Flipping Key Encapsulation (BIKE), Classic McEliece, Hamming Quasi-Cyclic (HQC), and Supersingular Isogeny Key Encapsulation (SIKE). As there is no algorithm for digital signatures left from the initial submissions, NIST

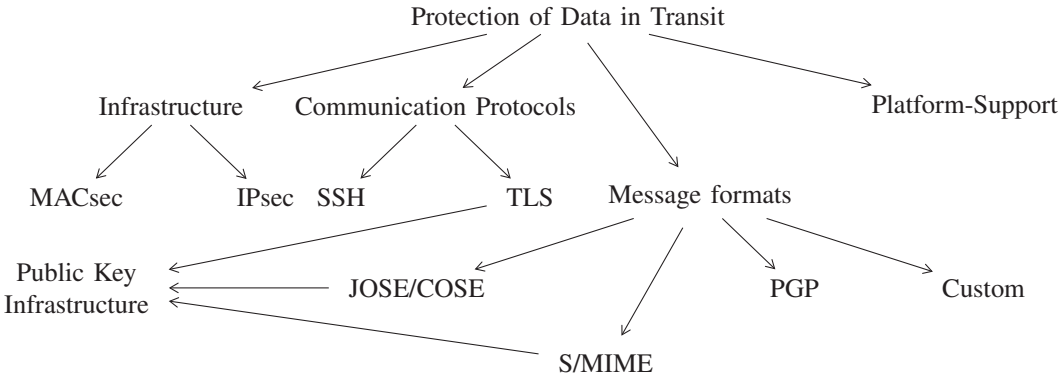

Figure 1. Overview of protocols used in different scenarios to protect data in transit.

launched another Call-for-Proposals on *Post-Quantum Cryptography: Digital Signature Schemes*, which is currently in the second round [22]. Hence, despite there are NIST standards already finalized, further algorithms are under consideration.

Naturally, the NIST process and its contributions from researchers all over the world are closely followed by government agencies from other nations.

The British National Cyber Security Center (NCSC) published a white paper recommending the use of the NIST standards or the hash-based signatures Leighton-Micali Hash-Based Signatures (LMS) or eXtended Merkle Signature Scheme (XMSS) [23].

In terms of post-quantum algorithms, the German Federal Office for Information Security / Bundesamt für Sicherheit in der Informationstechnik (BSI) recommends in its technical policy TR-02102-1 Version 2025-01 using FrodoKEM, Classic McElice or ML-KEM as a post-quantum cryptographic algorithm for encryption/key-agreement [24]. It mentions FrodoKEM and Classic McElice as a more conservative choice compared to the ML-KEM that is standardized by NIST. While FrodoKEM is not planned to be part of a NIST standard, its specification was submitted to International Organization for Standardization (ISO) for standardization [25].

For digital signatures, the policy recommends Merkle-Signatures, in detail XMSS or LMS, including Multi-Tree-Variants as described in [26] in addition to the NIST algorithms ML-DSA and SLH-DSA. The NIST algorithms should be used in the *hedged* version.

In general, the policy recommends combining a PQC approach and a classical one. The combination needs to ensure staying secure, as long as one of the used schemes is secure. Hash-based signatures are an exception in case they are properly implemented, i.e., they do not require a hybrid approach.

In contrast to the German BSI, the French Cybersecurity Agency (ANSSI) states in their PQC position paper, that the *ANSSI traditionally does not provide any closed list of recommended algorithms in order to avoid proscribing innovative state-of-the-art algorithms that could be well-suited for some particular use cases* [27]. However, a list of post-quantum algorithms together with recommendations is given. For KEM, they include ML-KEM and FrodoKEM. The list of digital signature algorithms contains ML-DSA, Falcon (FN-DSA), XMSS/LMS and SLH-DSA. In terms of combining PQC and classical algorithms, the ANSSI states their alignment with the position of the BSI recommending a hybrid approach.

Overall, the process of standardization results in the publication of various recommendations and draft standards. The analysis, including research on secure implementations, is still ongoing, leading to new attacks, cf. [28]. Despite the NIST is driving the most prominent competition, the government bodies of UK, Germany, and France are basically in line with the recommendations and have not announced any plans for running another competition.

While there is a focus on the most frequently used primitives, i.e., key encapsulation and digital signatures, several other, not that widely used primitives require consideration regarding PQ-security as well. A collection of such primitives with their PQC-status is given in [29]. The collection includes, e.g., schemes for authentication, protecting backups, and ensuring privacy used by Android, Apple, and Chrome. This list shows that a majority of the schemes are not PQ-secure. In particular, many of them rely on different versions of a Password-Authenticated Key Exchange (PAKE) scheme. While common PAKE schemes used today are not PQ-ready, there is a proposal of a generic construction to derive a PAKE from a KEM [30]. However, it is criticized not being fully PQ-secure [31]. Note that most of those schemes do not rely on standards but were designed for their specific purpose. Hence, it is up to the company to come up with PQC versions of their solutions.

Concluding, the current state, especially in a hybrid setting with a classic algorithm, provides a solid foundation for building and implementing protocols and further post-quantum secure solutions. However, when it comes to non-standard solutions, there are still many open questions and various schemes to adapt to provide PQ-security.

V. Protocols

In addition to developing and standardizing quantum-secure algorithms, protocol standards need to be adapted. Table I given an overview of the infrastructure and communication protocols and their actual status regarding PQC support.

TABLE I
OVERVIEW OF INFRASTRUCTURE AND COMMUNICATION PROTOCOLS, THEIR PQC SUPPORT, AND RELATED IMPLEMENTATIONS.

Protocol	Standard	PQC Support	Implementation
MACsec	IEEE Standard [32]	relies on symmetric cryptographic primitives	
IPsec	RFC 6071 [33]	RFC 8784 [34], Internet Darft [35]	Cisco IOS XE [36], Junos OS [37], strongSwan [38]
TLS	RFC 8446 [39]	Internet Draft [40]	Botan [41], WolfSSL [42], rustls [43], Open Quantum Safe* [44] Applications: Google experiments [6], Cloudflare [45]
SSH	RFC 4251 [46]	Internet Draft [47]	OpenSSH [48], Amazon implementation [49], Open Quantum Safe* [44]

* Use in production is not recommended by the project/developers.

A. Infrastructure

Common communication protocols to connect hosts to networks in a secure fashion or to establish a secure connection between networks are MACsec [32] and IPsec [33].

1) MACsec: As MACsec relies only on symmetric algorithms during the key agreement, using a 256-bit key is sufficient for post-quantum security. In addition, it is important to ensure that the key distribution is quantum-secure. Especially, since the session keys do not provide forward secrecy, i.e., a compromise of the long-term key material affects past session keys [50].

2) IPsec: For Internet Protocol Security (IPsec), Request For Comments (RFC) 8784 [34] defines a method to use pre-shared keys to achieve post-quantum security. The RFC is already supported by several products, like Ciscos IOS XE [36], strongSwan [38] and Junos OS [37]. This provides a viable solution already today. Potential adoptions of PQC for the Internet Key Exchange Protocol Version 2 (IKEv2) are in draft status. For example, [35] specifies a Hybrid Key Exchange with ML-KEM.

B. Communication Protocols

Common communication protocols include Transport Layer Security (TLS) and Secure Shell (SSH).

1) Transport Layer Security (TLS): The Transport Layer Security (TLS) protocol allows a secure end-to-end connection between applications. Various research has been conducted on how to best integrate post-quantum cryptography in the actual version of the protocol, TLS 1.3, and related performance, e.g., [51][52][53][54][55]. For TLS 1.3, a draft specifies a hybrid use of algorithms [40]. This ensures that the connections remain secure even if used algorithms are broken.

Version 3.7.0 of the Botan library [41] enables a hybrid key exchange per default using x25519/ML-KEM-768 and adds support for ML-KEM key exchange in a non-hybrid mode. WolfCrypt, the underlying library of WolfSSL also supports ML-KEM, ML-DSA, SPHINCS+, and stateful hash-based signatures. They are integrated as Kyber and Elliptic Curve Cryptography (ECC)/Kyber hybrid codepoints, as well as Dilithium signature algorithms [42]. Since version 0.23.22, rustls supports a hybrid PQ key exchange, using ECC and ML-KEM [43].

Other implementations of the draft and non-hybrid PQC key exchange methods are provided by the Open Quantum Safe project [44] in form of an OpenSSLv3 provider and an integration into a BoringSSL fork. However, those two implementations should not be considered *production quality* according to the project.

Note that a recent IETF draft states that TLS 1.2 will not be further enhanced, which implies, it will not support PQC, despite TLS 1.2 is still widespread [56].

Further experiments on challenges when using PQC-TLS at a large scale were conducted by Google [6]. Their tests revealed incompatibilities in network products that will be fixed via firmware updates. Similar PQC-support is enabled by Cloudflare [45], targeting support of all outbound connections by March 2024. This can be used with browsers supporting the hybrid cipher suite consisting of X25519 and Kyber-768, like Chrome, where it has been enabled since version 116 [57]. Moreover, Cloudflare provides real-time data on the percentage of their Hypertext Transfer Protocol Secure (HTTPS) connections that utilize PQC [58]. At the of time writing this paper, around 33% of the HTTPS connections are using PQC.

Hence, a draft standard and implementations are available. Some widespread experiments have been conducted successfully and first rollouts are taking place. Standardized support of PQC for TLS 1.3 is expected to build on the released NIST standards.

2) Secure Shell (SSH): Secure Shell (SSH) is a protocol for secure execution of remote commands. A very prominent implementation is OpenSSH, which is part of many major Linux distributions. OpenSSH made a hybrid key exchange method that combines ML-KEM with an Elliptic-Curve Diffie–Hellman (ECDH) key exchange default in version 9.9 [48]. The implementation relies on an individual submission [59], which has been replaced by an Internet Draft [47]. The same mechanism is implemented and used by Amazon Web Services (AWS) [49]. Amazon states in the PQC roadmap to adopt ML-KEM for SSH as soon as *as standards bodies such as the IETF publish implementation guidance for those protocols* [60]. The Open

Quantum Safe project [44] also provides an experimental implementation supporting PQC based on an OpenSSH-Fork.

Overall, with OpenSSL, that uses a hybrid approach per default, and the AWS implementation, there are real-world possibilities for PQC key-agreement, despite there being no final standard yet.

C. Message Security

On the message layer, the application can choose to encrypt/sign the transferred data, depending on the use case. Potential solutions include JOSE/COSE for sharing data between applications, S/MIME for mail/web pages and PGP for arbitrary data, including file exchange.

1) JOSE/COSE: JSON and CBOR are formats for data exchange between applications. The related signing and encryption standards are JOSE and COSE. For COSE, hash-based signatures are defined in RFC 8778 [61]. An active IETF drafts exists to support Dilithium [62]. In addition to this working group draft, other individual drafts have been submitted to the IETF as well.

2) S/MIME: The S/MIME standard [63] mandates the support of RSA-based and ECC-based ciphers for signing and encryption. Preparing the standard for the quantum-age is part of the *Limited Additional Mechanisms for PKIX and SMIME (lamps)* working group charter [64]. Nevertheless, the possibility of integrating PQC-ciphers into the mail client Thunderbird is briefly discussed in [65], and a demo integration was done by the MTG AG [66].

3) PGP: The options for using post-quantum ciphers in PGP were analyzed by Wussler [67], leading to an IETF draft [68]. A former version of this draft was formally analyzed by Tran et al. [69].

While there is work underway for all three standards, there is still a lack of practical implementations and experiments that will lead to solutions that can be used in production environments.

D. Platform-Support

Especially with the increasing use of cloud computing, it is important to take a look at the security foundations of the cloud providers and service platforms in general. Google Cloud Platforms (GCP) protects its internal communication with a protocol called Application Layer Transport Security (ALTS), which already employs post-quantum cryptography to protect against harvest now and decrypt later attacks [70]. Similar to Google, Meta is using a hybrid PQC implementation for most internal communications, protecting it against attacks recording traffic today and decrypting it with quantum computers when possible [71].

Amazon published a roadmap for the transition of Amazon Web Services (AWS) towards PQC, mentioning that their libraries, including those used for HTTPS-based endpoints, support PQC and hence, also allow customers testing the impact of PQC [60].

Microsoft launched a project called Quantum Safe [72] and participate [73] in the Open Quantum Safe project [44]. Furthermore, the integration of PQC in Microsoft libraries is ongoing [74].

E. Public Key Infrastructures (PKIs) and Certificates

Public Key Infrastructures (PKIs) are essential for ensuring trust in the digital world. Ranging from communication protocols to digitally signed documents - a reliable PKI is required to ensure the identity of the counterpart. For trustworthy certificates in the presence of quantum computers, the whole chain, starting with the root certificate must be quantum-secure.

The draft [80] defines a composite certificate combining ML-DSA with traditional signature algorithms. This solution ensures that the certificate remains secure even in case one of the algorithms is broken. A similar approach is used for KEM solutions [81] in the context of PKI-related profiles and protocols like Cryptographic Message Syntax (CMS) [82] and Public Key Infrastructure for X.509 (PKIX).

Various drafts are already published to be ready to proceed now the NIST standards are finalized. They include certificates using ML-KEM [75], and Dilithium [76].

The draft [83] relies on the Stateful Hash-Based Signature Schemes (S-HBS) [26], Hierarchical Signature System (HSS), eXtended Merkle Signature Scheme (XMSS) [84], and XMSSˆMT, a multi-tree variant of XMSS and provides algorithm identifiers for X.509 PKIs. While their security is well understood, those signature schemes come with the drawback that they can only create a limited amount of signatures, and it is required to maintain a state to remain secure.

During the transition phase, it is important that also legacy systems that might not support post-quantum cryptography can verify a certificate with classic algorithms. The specifications above cannot be used in such a scenario, as they require the verifying system process PQC signatures. A possible approach in the transition scenario is using related certificates, as laid out in the draft specifications [78] and the individual submission [79]. The impact of hybrid certificates on current implementations was investigated in [85]. The authors concluded the certificates can be processed by the tested solutions without or with minor modifications.

Another option is specified by the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) [77], namely to include an alternative signature in a certificate. This allows clients that are not capable of processing PQC algorithms to ignore this signature, while others can benefit from it. However, the drawback of this approach is the increased certificate size for all consuming entities. Table II provides and overview of the different options.

When it comes to commercial products, PKI solution vendors are working towards addressing the upcoming challenges, preparing examples [86], offering experimental suites [87][9] or solutions [88].

Despite various activities that are underway, neither the majority of the standardization work nor the related implementations have been concluded yet. As especially the root

TABLE II
OVERVIEW DIFFERENT WAYS HOW CERTIFICATES CAN SUPPORT PQ ALGORITHMS.

Certificate Type	Standard	Purpose	Contains
PQC Certificates	Internet Drafts [75][76]	supports only PQC-enabled system	PQC keys and signatures
Hybrid Certificates	ITU-T Specification [77], Internet Drafts [78][79]	can be verified by PQC-enabled and legacy systems; allow a transition phase	PQC and non-PQC keys and signatures
Composite Certificates	Internet Draft [80]	verification requires PQC-enabled system; remain secure in cases the non-PQC or the PQC algorithm breaks	PQC and non-PQC keys and signatures

certificates are commonly valid for several years, it is important to plan their replacement together with a sound transition approach.

VI. FOUNDATIONS AND LIBRARIES

Together with research and standardization of PQC algorithms, their implementation is progressing. A popular project to support *the transition to quantum-resistant cryptography* is Open Quantum Safe [44]. It is part of the Linux Foundation's Post-Quantum Cryptography Alliance. Its main working items are a C library for post-quantum algorithms, called liboqs, and prototype integration into protocols and applications.

Currently, liboqs supports the standards ML-KEM, ML-DSA, XMSS, LMS, as well as Falcon and SPHINCS+. Furthermore, the NIST round 4 candidates Classic McEliece, BIKE and HQC, as well as, FrodoKEM and NTRU-Prime. The project provides several language wrappers to allow using it for example in C++, JAVA, Go, and Python. However, the project page does recommend refraining from using the library in production environments, as it has not undergone a thorough audit/analysis process yet.

Another library is libpqcrypto [90], that provides C-reference implementations for 19 algorithms with different parameter sets and comes with a Python API. The project team warns that there might be security problems in the library, either due to issues of the cryptographic primitives themselves or due to software bugs.

PQClean [91][95] aims at providing standalone C implementations of the NIST PQC algorithms. The project states that the library is suited for research purposes and suggest an use-case-specific assessment for any other use of the library. The JavaScript library pqc.js makes use of PQClean.

The Cloudflare Interoperable, Reusable Cryptographic Library (CIRCL) [93] provides several Go implementations of cryptographic primitives, including PQC. It comes with the cautious note that it is provided as-is and parts are experimental.

A popular library that provides PQC support is Bouncy Castle for Java and C# [94]. Its implementation includes ML-KEM, ML-DSA, SLH-DSA, LMS, and XMSS algorithms. Further algorithms are implemented as well, but project states that those non-standardized algorithms can be used for experiments and hybrid scenarios, but not for long-term protection. Further libraries that have already been mentioned in the context of TLS are Botan [41] and wolfCrypt [42]. An overview of implementations is given in Table III.

Overall, there are two aspects to consider about using PQC algorithms today: (1) First standards have recently been finalized and the security research is ongoing. They also do not have the benefit of a long history of intensive security research that current standards possess. Therefore, the BSI, recommends using the current PQC algorithms in a hybrid mode. (2) In addition to the security of the algorithms, quality [95] and security of its implementations are important. This includes sufficient quality assurance and auditing to prevent vulnerabilities and security bugs as well as resistance against potential side-channel attacks like [96]–[98].

VII. APPLICATIONS

In addition to developments on standards and implementations that are directly related to those standards, there are also products and solutions that move forward in supporting PQC.

A. Messengers

Messengers are widely used on mobile phones to exchange information. Very common apps are WhatsApp, Signal, Threema, Telegram, and Apples iMessage. Key functionality of all apps is, in addition to various different ways of exchanging messages, protecting the security and privacy of its users. Hence, they all provide end-to-end encryption for the exchanged messages [99]–[101][1]. Note that WhatsApp and Signal rely on the same protocol [102].

Recent versions of Signal support a post-quantum secure key agreement protocol, called PQXDH [103]. Apple's iMessage protocol PQ3 also relies on PQC [104]. In addition to PQXDH, PQ3 also performs PQC rekeying. However, both protocols do not provide a quantum-secure authentication method yet.

B. Blockchain

Blockchain technology enables a decentralized trust model with various applications. Those include cryptocurrencies such as Bitcoin [105], health applications [106], and blockchain-backed logistics [107]. Given the data stored in and protected by different blockchains, it is important to ensure that neither the data a chain contains can be manipulated using quantum attacks

[1]Note that for Telegram, end-to-end encryption is only enabled for secret chats, not per default.

TABLE III
OVERVIEW OF SELECTED LIBRARIES SUPPORTING POST-QUANTUM ALGORITHMS.

Name	Language	Supported Algorithms	Experimental
liboqs [89]	C	ML-KEM, ML-DSA, XMSS, LMS, Kyber, Dilithum, Falcon, SPHINCS+, Classic McEliece, BIKE, HQC, FrodoKEM, NTRU-Prime. CROSS, MAYO	Y
libpqcrypto [90]	Python, C	Classic McEliece, Dilithium, Kyber, FrodoKEM, NTRU Prime, SPHINCS+ (and more)	Y
PQClean [91]	C	Kyber, HQC, Classic MCEliece, Dilithium, Falcon, SPHINCS+	Y
pqc.js [92]	JavaScript	uses PQClean	Y
CIRCL [93]	Go	ML-KEM, X-Wing, Kyber, Frodo, CSIDH, ML-DSA, Dilithium	Y
Bouncy Caste [94]	JAVA, C#	ML-KEM, ML-DSA, SLH-DSA, LMS, XMSS*	
Botan [41]	C++	ML-DSA, SLH-DSA, HSS/LMS, XMSS	
wolfSSL/wolfCrypt [42]	C	ML-KEM, ML-DSA, SPHINCS+, LMS/HSS, XMSS	

* Further PQC-algorithms are implemented, but according to the documentation unsuitable for long-term use.

nor future transactions can be forged. The key primitives that a blockchain uses are hash functions and public-key cryptography. A good overview on the impact and potential solutions is given in [108]. In addition to the scientific analysis like [109]–[111], there are products that already focus on post-quantum secure blockchains using post-quantum cryptography [112] or combine PQC with quantum key distributions technologies [113]. In contrast, making the popular system Ethereum post-quantum secure, is on the roadmap, but is stated ongoing research [114]. For Bitcoin, an experimental branch, Bitcoin Post-Quantum (BPQ), exists [115].

VIII. QUANTUM KEY DISTRIBUTION

Quantum mechanisms cannot only be used to attack cryptographic primitives. It is also possible to use them for protecting digital communication. In essence, quantum mechanisms allow exchanging key material in a secure way. This is called Quantum Key Distribution (QKD). The shared keys allow establishing a secure data transmission channel, e.g., via symmetric algorithms.

In contrast to post-quantum cryptography, using QKD requires specific hardware, like encryptors provided by Fraunhofer [116] or ID Quantique [117]. As it is relying on quantum effects, there are expected to be very secure systems. However, neither the German BSI [118] nor the United States National Security Agency (NSA) [119] consider QKD a priority. The BSI mentions that the technology is not yet mature enough in terms of security and only suitable for some niche use cases. The NSA highlights cost efficiency and better maintainability of PQC compared to QKD. Both recommend focusing on PQC.

IX. CONCLUSION AND FUTURE WORK

Quantum computers endanger the security of cryptographic algorithms. Especially asymmetric algorithms are affected. This requires new algorithms as well as updated standards to make use of those new algorithms. Various efforts from research over standardization to implementation are currently under way to address this challenge. This paper started by looking at possibilities to secure the underlying network infrastructure. As IPsec and MACsec can rely on secret-key cryptography, the remaining challenge is secure key management.

In order to achieve end-to-end security, SSH can be used with post-quantum security, e.g., via OpenSSH, first rollouts of PQC TLS implementations are taking place. Standards for message encryption are still at a comparably early stage. However, libraries, especially BouncyCastle for JAVA and C#, wolfCrypto for C and Botan for C++, provide algorithms that can already integrated into applications; given the required expert knowledge is available. Several applications like messengers, services like Blockchain and cloud platforms are moving or have already completed important steps towards supporting PQC. The use of quantum technologies to exchange key material, called QKD, is another area of ongoing research, especially suited for niche applications.

Overall, the transition will require thorough planning. This paper highlighted where first steps can be done already today. Depending on the use case, hybrid approaches can protect against quantum attacks while preventing risks due to attacks on comparably new PQC algorithms. Furthermore, becoming crypto-agile, in the sense that algorithms can be exchanged easily, will not only help in addressing the current PQC challenge, but also reduce the effort of future transitions of cryptographic algorithms.

REFERENCES

[1] J. Schmidt and A. Lawall, “Theoretical and practical aspects in identifying gaps and preparing for post-quantum cryptography”, in *Proceedings of the Eighteenth International Conference on Emerging Security Information, Systems and Technologies (SECURWARE), 2024*, Nice, France: IARIA, Nov. 2024, pp. 36–42, ISBN: 978-1-68558-206-7.

[2] U.S. Department of Homeland Security, *Post-quantum cryptography - frequently asked questions*, https://www.dhs.gov/sites/default/files/publications/post_quantum_cryptography_faq_3_seals_october_2021_508.pdf, retrieved: May, 2025.

[3] National Security Agency, *The Commercial National Security Algorithm Suite 2.0 and Quantum Computing FAQ*, https://media.defense.gov/2022/Sep/07/2003071836/-1/-1/1/CSI_CNSA_2.0_FAQ_.PDF, retrieved: May, 2025.

[4] P. E. Hoffman and S. Celi (WG chairs), *Post-Quantum Use In Protocols (pquip)*, https://datatracker.ietf.org/wg/pquip/about/, retrieved: May, 2025.

[5] ETSI, *Quantum-Safe Cryptography (QSC)*, https://www.etsi.org/technologies/quantum-safe-cryptography, retrieved: May, 2025.

[6] Google, *How Google is preparing for a post-quantum world*, https://cloud.google.com/blog/products/identity-security/how-google-is-preparing-for-a-post-quantum-world, retrieved: May, 2025.

[7] IBM, *Make the world quantum safe*, https://www.ibm.com/quantum/quantum-safe, retrieved: May, 2025.

[8] Microsoft, *Post-quantum cryptography*, https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/, retrieved: May, 2025.

[9] Utimaco, *Post Quantum Cryptography*, https://utimaco.com/solutions/applications/post-quantum-cryptography, retrieved: May, 2025.

[10] ETSI, *ETSI TR 103 619 V1.1.1 (2020-07) - CYBER; Migration strategies and recommendations to Quantum Safe schemes*, https://www.etsi.org/deliver/etsi_tr/103600_103699/103619/01.01.01_60/tr_103619v010101p.pdf.

[11] Post-Quantum Cryptography (PQC) Working Group, "Risk model technical paper", FS-ISAC, Tech. Rep., 2023, available at https://www.fsisac.com/hubfs/Knowledge/PQC/RiskModel.pdf, retrieved: May, 2025.

[12] T. Patterson, *Moving toward a Quantum Security Maturity Index*, Presentation at Post-Quantum Cryptography Conference 2023, available at https://pkic.org/events/2023/pqc-conference-amsterdam-nl/pkic-pqcc_tom-patterson_accenture_moving-toward-a-quantum-security-maturity-index.pdf, retrieved: May, 2025.

[13] DigiCert, *Post-Quantum Cryptography (PQC) Maturity Model*, https://www.digicert.com/resources/post-quantum-cryptography-maturity-model.pdf, retrieved: May, 2025.

[14] Intel, *Intel® Software Guard Extensions (Intel®SGX)*, https://www.intel.de/content/www/de/de/products/docs/accelerator-engines/software-guard-extensions.html, retrieved: May, 2025.

[15] Intel, *Intel® Trust Domain Extensions (Intel® TDX)*, https://www.intel.com/content/www/us/en/developer/tools/trust-domain-extensions/overview.html, retrieved: May, 2025.

[16] AMD, *AMD Secure Encrypted Virtualization (SEV)*, https://www.amd.com/de/developer/sev.html, retrieved: May, 2025.

[17] G. Caruso, "Post-quantum algorithms support in Trusted Execution Environment", Available at https://webthesis.biblio.polito.it/31076/, Ph.D. dissertation, Politecnico di Torino, 2024.

[18] NIST - Computer Security Resource Center, *Post-Quantum Cryptography PQC - Call for Proposals*, https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/Call-for-Proposals, retrieved: May, 2025.

[19] NIST, "Module-lattice-based key-encapsulation mechanism standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 203, 2024. DOI: 10.6028/NIST.FIPS.203.

[20] NIST, "Module-lattice-based digital signature standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 204, 2024. DOI: 10.6028/NIST.FIPS.204.

[21] NIST, "Stateless hash-based digital signature standard", U.S. Department of Commerce, Washington, D.C., Tech. Rep. Federal Information Processing Standards Publication (FIPS) 205, 2024. DOI: 10.6028/NIST.FIPS.205.

[22] NIST, *Post-Quantum Cryptography: Additional Digital Signature Schemes*, https://csrc.nist.gov/projects/pqc-dig-sig/round-2-additional-signatures, retrieved: May, 2025.

[23] National Cyber Security Center, *Next steps in preparing for post-quantum cryptography*, https://www.ncsc.gov.uk/whitepaper/next-steps-preparing-for-post-quantum-cryptography, retrieved: May, 2025.

[24] Federal Office for Information Security / Bundesamt für Sicherheit in der Informationstechnik (BSI), *Kryptographische Verfahren: Empfehlungen und Schlüssellängen*, https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR02102/BSI-TR-02102.pdf, Technische Richtlinie, Bonn, Deutschland, 2025.

[25] E. Alkim et al., *FrodoKEM - Practical quantum-secure key encapsulation from generic lattices*, "https://frodokem.org/, retrieved: May, 2025.

[26] D. Cooper *et al.*, "Recommendation for Stateful Hash-Based Signature Schemes", Special Publication (NIST SP), National Institute of Standards and Technology, U.S. Department of Commerce, Tech. Rep., 2020. DOI: https://doi.org/10.6028/NIST.SP.800-208.

[27] ANSSI, *ANSSI views on the Post-Quantum Cryptography transition (2023 follow up)*, https://cyber.gouv.fr/sites/default/files/document/follow_up_position_paper_on_post_quantum_cryptography.pdf, retrieved: May, 2025.

[28] P. Ravi, D. Jap, S. Bhasin, and A. Chattopadhyay, "Invited paper: Machine learning based blind side-channel attacks on pqc-based kems - A case study of kyber KEM", in *IEEE/ACM International Conference on Computer Aided Design, ICCAD 2023, San Francisco, CA, USA, October 28 - Nov. 2, 2023*, IEEE, 2023, pp. 1–7. DOI: 10.1109/ICCAD57390.2023.10323721.

[29] B. Westerbaan et al., *Fancy cryptography in the wild*, https://github.com/fancy-cryptography/fancy-cryptography, retrieved: May, 2025.

[30] H. Beguinet, C. Chevalier, D. Pointcheval, T. Ricosset, and M. Rossi, "GeT a CAKE: Generic Transformations from Key Encaspulation Mechanisms to Password Authenticated Key Exchanges", in *Applied Cryptography and Network Security*, M. Tibouchi and X. Wang, Eds., Cham: Springer Nature Switzerland, 2023, pp. 516–538, ISBN: 978-3-031-33491-7.

[31] N. Alnahawi, J. Alperin-Sheriff, D. Apon, and A. Wiesmaier, *NICE-PAKE: On the security of KEM-based PAKE constructions without ideal ciphers*, Cryptology ePrint Archive, Paper 2024/1957, available at https://eprint.iacr.org/2024/1957, 2024.

[32] M. Seaman, *IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security*.

[33] S. Frankel and S. Krishnan, *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*, RFC 6071, Feb. 2011. DOI: 10.17487/RFC6071.

[34] S. Fluhrer, P. Kampanakis, D. McGrew, and V. Smyslov, *Mixing Preshared Keys in the Internet Key Exchange Protocol Version 2 (IKEv2) for Post-quantum Security*, RFC 8784, Jun. 2020. DOI: 10.17487/RFC8784.

[35] P. Kampanakis and G. Ravago, "Post-quantum Hybrid Key Exchange with ML-KEM in the Internet Key Exchange Protocol Version 2 (IKEv2)", Internet Engineering Task Force, Internet-Draft draft-ietf-ipsecme-ikev2-mlkem-00, May 2025, Work in Progress, 10 pp.

[36] CISCO, *Security and VPN Configuration Guide, Cisco IOS XE 17.x*, https://www.cisco.com/c/en/us/td/docs/routers/ios/config/17-x/sec-vpn/b-security-vpn/m-sec-cfg-quantum-encryption-ppk.html, retrieved: May, 2025.

[37] Juniper Networks, *Release Notes: Junos OS Release 22.4R1*, https://www.juniper.net/documentation/us/en/software/junos/release-notes/22.4/junos-release-notes-22.4r1/index.html, retrieved: May, 2025.

[38] strongSwan, *IPsec and Related Standards*, https://docs.strongswan.org/docs/latest/features/ietf.html, retrieved: May, 2025.

[39] E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.3*, RFC 8446, Aug. 2018. DOI: 10.17487/RFC8446.

[40] D. Stebila, S. Fluhrer, and S. Gueron, "Hybrid key exchange in TLS 1.3", Internet Engineering Task Force, Internet-Draft draft-ietf-tls-hybrid-design-12, Jan. 2025, Work in Progress, 24 pp.

[41] Botan, *Botan - Release Notes*, https://botan.randombit.net/news.html, retrieved: May, 2025.

[42] wolfSSL, *wolfSSL Support for Post-Quantum*, https://www.wolfssl.com/products/wolfcrypt-post-quantum/, retrieved: May, 2025.

[43] J. Birr-Pixton et al., *rustls releases*, https://github.com/rustls/rustls/releases, retrieved: May, 2025.

[44] *Open Quantum Safe Project*, https://openquantumsafe.org/, retrieved: May, 2025.

[45] W. Evans, B. Westerbaan, C. Patton, P. Wu, and V. Gonçalves, *Post-quantum cryptography goes GA*, https://blog.cloudflare.com/post-quantum-cryptography-ga/, retrieved: May, 2025.

[46] C. M. Lonvick and T. Ylonen, *The Secure Shell (SSH) Protocol Architecture*, RFC 4251, Jan. 2006. DOI: 10.17487/RFC4251.

[47] P. Kampanakis, D. Stebila, and T. Hansen, "PQ/T Hybrid Key Exchange in SSH", Internet Engineering Task Force, Internet-Draft draft-ietf-sshm-mlkem-hybrid-kex-02, Apr. 2025, Work in Progress, 14 pp.

[48] OpenSSH, *Openssh 9.9 release notes*, https://www.openssh.com/txt/release-9.9, retrieved: May, 2025.

[49] AWS Security Blog, *Post-quantum hybrid SFTP file transfers using AWS Transfer Family*, https://aws.amazon.com/de/blogs/security/post-quantum-hybrid-sftp-file-transfers-using-aws-transfer-family, retrieved: May, 2025.

[50] ETSI, *ETSI TR 103 617 V1.1.1 (2018-09) - Quantum-Safe Virtual Private Networks*, https://www.etsi.org/deliver/etsi_tr/103600_103699/103617/01.01.01_60/tr_103617v010101p.pdf.

[51] J. I. E. Pablos, M. E. Marriaga, and A. P. d. Pozo, "Design and Implementation of a Post-Quantum Group Authenticated Key Exchange Protocol With the LibOQS Library: A Comparative Performance Analysis From Classic McEliece, Kyber, NTRU, and Saber", *IEEE Access*, vol. 10, pp. 120 951–120 983, 2022. DOI: 10.1109/ACCESS.2022.3222389.

[52] J. Henrich, A. Heinemann, A. Wiesmaier, and N. Schmitt, "Performance Impact of PQC KEMs on TLS 1.3 Under Varying Network Characteristics", in *Information Security*, E. Athanasopoulos and B. Mennink, Eds., Cham: Springer Nature Switzerland, 2023, pp. 267–287, ISBN: 978-3-031-49187-0. DOI: 10.1007/978-3-031-49187-0_14.

[53] S. Paul, Y. Kuzovkova, N. Lahr, and R. Niederhagen, "Mixed Certificate Chains for the Transition to Post-Quantum Authentication in TLS 1.3", in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS '22, Nagasaki, Japan: Association for Computing Machinery, 2022, pp. 727–740, ISBN: 9781450391405. DOI: 10.1145/3488932.3497755.

[54] C. R. Garcia, A. C. Aguilera, J. J. V. Olmos, I. T. Monroy, and S. Rommel, "Quantum-Resistant TLS 1.3: A Hybrid Solution Combining Classical, Quantum and Post-Quantum Cryptography", in *2023 IEEE 28th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2023, pp. 246–251. DOI: 10.1109/CAMAD59638.2023.10478407.

[55] M. Sosnowski *et al.*, "The Performance of Post-Quantum TLS 1.3", in *Companion of the 19th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT 2023, Paris, France: Association for Computing Machinery, 2023, pp. 19–27, ISBN: 9798400704079. DOI: 10.1145/3624354.3630585.

[56] R. Salz and N. Aviram, "TLS 1.2 is in Feature Freeze", Internet Engineering Task Force, Internet-Draft draft-ietf-tls-tls12-frozen-08, Apr. 2025, Work in Progress, 6 pp.

[57] D. O'Brien, *Protecting Chrome Traffic with Hybrid Kyber KEM*, https://blog.chromium.org/2023/08/protecting-chrome-traffic-with-hybrid.html, retrieved: May, 2025.

[58] Cloudflare Radar, *Adoption & Usage - Post-Quantum Encryption Adoption*, https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption, retrieved: May, 2025.

[59] P. Kampanakis, D. Stebila, and T. Hansen, "PQ/T Hybrid Key Exchange in SSH", Internet Engineering Task Force, Internet-Draft draft-kampanakis-curdle-ssh-pq-ke-03, Work in Progress, 14 pp.

[60] M. Campagna, M. Goldsborough, and P. O'Donnell, *AWS post-quantum cryptography migration plan*, https://aws.amazon.com/de/blogs/security/aws-post-quantum-cryptography-migration-plan/, retrieved: May, 2025.

[61] R. Housley, *Use of the HSS/LMS Hash-Based Signature Algorithm with CBOR Object Signing and Encryption (COSE)*, RFC 8778, Apr. 2020. DOI: 10.17487/RFC8778.

[62] M. Prorock, O. Steele, R. Misoczki, M. Osborne, and C. Cloostermans, "ML-DSA for JOSE and COSE", Internet Engineering Task Force, Internet-Draft draft-ietf-cose-dilithium-06, Apr. 2025, Work in Progress, 19 pp.

[63] J. Schaad, B. C. Ramsdell, and S. Turner, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 4.0 Message Specification*, RFC 8551, Apr. 2019. DOI: 10.17487/RFC8551.

[64] R. Housley and T. Hollebeek (WG Chairs), *Limited Additional Mechanisms for PKIX and SMIME (lamps)*, https://datatracker.ietf.org/wg/lamps/about/, retrieved: May, 2025.

[65] C. Döberl et al., "Quantum-resistant End-to-End Secure Messaging and Email Communication", in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ser. ARES '23, Benevento, Italy: Association for Computing Machinery, 2023, pp. 1–8, ISBN: 9798400707728. DOI: 10.1145/3600160.3605049.

[66] MTG AG, *PQC Anwendungen jetzt testen!*, https://www.mtg.de/de/post-quantum-kryptografie/pqc-demo/#PQC_Testanwendungen, retrieved: May, 2025.

[67] A. Wussler, "Post-Quantum cryptography in OpenPGP", M.S. thesis, Wien, 2023. DOI: 10.34726/hss.2023.106226.

[68] S. Kousidis, J. Roth, F. Strenzke, and A. Wussler, "Post-Quantum Cryptography in OpenPGP", Internet Engineering Task Force, Internet-Draft draft-ietf-openpgp-pqc-07, Feb. 2025, Work in Progress, 107 pp.

[69] D. D. Tran, K. Ogata, and S. Escobar, "A formal analysis of OpenPGP's post-quantum public-key algorithm extension", in *Proceedings of the 2nd International Workshop on Formal Analysis and Verification of Post-Quantum Cryptographic Protocols (FAVPQC), 2023*, Brisbane, Australia: JAIST Press, 2023, pp. 22–35.

[70] S. Kölbl, R. Misoczki, and S. Schmieg, *Securing tomorrow today: Why Google now protects its internal communications from quantum threats*, https://cloud.google.com/blog/products/identity-security/why-google-now-uses-post-quantum-cryptography-for-internal-comms, retrieved: May, 2025.

[71] S. Lin *et al.*, *Post-quantum readiness for TLS at Meta*, https://engineering.fb.com/2024/05/22/security/post-quantum-readiness-tls-pqr-meta/, retrieved: May, 2025.

[72] Microsoft, *Quantum-safe overview*, https://quantum.microsoft.com/en-us/vision/quantum-cryptography-overview, retrieved: May, 2025.

[73] Microsoft Research, *Post-quantum cryptography*, https://www.microsoft.com/en-us/research/project/post-quantum-cryptography/, retrieved: May, 2025.

[74] A. Thipsay, *Microsoft's quantum-resistant cryptography is here*, https://techcommunity.microsoft.com/blog/microsoftsecurityandcompliance/microsofts-quantum-resistant-cryptography-is-here/4238780, retrieved: May, 2025.

[75] S. Turner, P. Kampanakis, J. Massimo, and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM)", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-kyber-certificates-10, Apr. 2025, Work in Progress, 71 pp.

[76] J. Massimo, P. Kampanakis, S. Turner, and B. Westerbaan, "Internet X.509 Public Key Infrastructure - Algorithm Identifiers for the Module-Lattice-Based Digital Signature Algorithm (ML-DSA)", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-dilithium-certificates-09, May 2025, Work in Progress, 89 pp.

[77] Telecommunication Statdardization Sector of ITU, *Directory Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks*, Series X - Data networks and open system communication, Oct. 2019.

[78] A. Becker, R. Guthrie, and M. J. Jenkins, "Related Certificates for Use in Multiple Authentications within a Protocol", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-cert-binding-for-multi-auth-06, Dec. 2024, Work in Progress, 14 pp.

[79] C. Bonnell, J. Gray, D. Hook, T. Okubo, and M. Ounsworth, "A Mechanism for Encoding Differences in Paired Certificates", Internet Engineering Task Force, Internet-Draft draft-bonnell-lamps-chameleon-certs-06, Apr. 2025, Work in Progress, 52 pp.

[80] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, and S. Fluhrer, "Composite ML-DSA for use in X.509 Public Key Infrastructure and CMS", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-pq-composite-sigs-04, Mar. 2025, Work in Progress, 82 pp.

[81] M. Ounsworth, J. Gray, M. Pala, J. Klaußner, and S. Fluhrer, "Composite ML-KEM for use in X.509 Public Key Infrastructure and CMS", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-pq-composite-kem-06, Mar. 2025, Work in Progress, 66 pp.

[82] R. Housley, J. Gray, and T. Okubo, *Using Key Encapsulation Mechanism (KEM) Algorithms in the Cryptographic Message Syntax (CMS)*, RFC 9629, Aug. 2024. DOI: 10.17487/RFC9629.

[83] D. V. Geest, K. Bashiri, S. Fluhrer, S.-L. Gazdag, and S. Kousidis, "Use of the HSS and XMSS Hash-Based Signature Algorithms in Internet X.509 Public Key Infrastructure", Internet Engineering Task Force, Internet-Draft draft-ietf-lamps-x509-shbs-13, Dec. 2024, Work in Progress, 35 pp.

[84] A. Huelsing, D. Butin, S.-L. Gazdag, J. Rijneveld, and A. Mohaisen, *XMSS: eXtended Merkle Signature Scheme*, RFC 8391, May 2018. DOI: 10.17487/RFC8391.

[85] J. Fan et al., "Impact of post-quantum hybrid certificates on PKI, common libraries, and protocols", *International Journal of Security and Networks*, vol. 16, no. 3, pp. 200–211, 2021. DOI: 10.1504/IJSN.2021.117887.

[86] GlobalSign, *Post Quantum Computing - Future-proofing digital trust with safe certificates*, https://www.globalsign.com/en/post-quantum-computing, retrieved: May, 2025.

[87] Keyfactor, *Post-Quantum Cryptography Keys and Signatures*, https://doc.primekey.com/ejbca/ejbca-operations/ejbca-ca-concept-guide/certificate-authority-overview/post-quantum-cryptography-keys-and-signatures, retrieved: May, 2025.

[88] Entrust, *Post-Quantum Cryptography*, https://www.entrust.com/solutions/post-quantum-cryptography, retrieved: May, 2025.

[89] Open Quantum Safe, *Liboqs*, https://github.com/open-quantum-safe/liboqs, retrieved: May, 2025.

[90] PQCRYPTO, *Libpqcrypto*, https://libpqcrypto.org/, retrieved: May, 2025.

[91] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, "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*, Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010.

[92] *pqc.js*, https://github.com/Dashlane/pqc.js/, retrieved: May, 2025.

[93] A. Faz-Hernandez and K. Kwiatkowski, *Introducing CIRCL: An Advanced Cryptographic Library*, Available at https://github.com/cloudflare/circl. v1.6.1 Accessed May, 2025, Cloudflare, Jun. 2019.

[94] *Bouncy Castle*, https://www.bouncycastle.org/, retrieved: May, 2025.

[95] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, "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*, Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010.

[96] P. Ravi, A. Chattopadhyay, J. P. D'Anvers, and A. Baksi, "Side-channel and Fault-injection attacks over Lattice-based Post-quantum Schemes (Kyber, Dilithium): Survey and New Results", *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 1–54, Mar. 2024, ISSN: 1539-9087. DOI: 10.1145/3603170.

[97] C. Mujdei et al., "Side-channel Analysis of Lattice-based Post-quantum Cryptography: Exploiting Polynomial Multiplication", *ACM Trans. Embed. Comput. Syst.*, vol. 23, no. 2, pp. 1–23, Mar. 2024, ISSN: 1539-9087. DOI: 10.1145/3569420.

[98] A. T. Hoang et al., "Deep Learning Enhanced Side Channel Analysis on CRYSTALS-Kyber", in *2024 25th International Symposium on Quality Electronic Design (ISQED)*, 2024, pp. 1–8. DOI: 10.1109/ISQED60706.2024.10528674.

[99] Telegram, *Mtproto mobile protocol*, https://core.telegram.org/mtproto, retrieved: May, 2025.

[100] Threema, *Threema cryptography whitepaper*, https://digi77.com/software/public/threema_cryptography_whitepaper.pdf, retrieved: May, 2025.

[101] Signal, *Technical information*, https://signal.org/docs/, retrieved: May, 2025.

[102] M. Marlinspike, *Whatsapp's signal protocol integration is now complete*, https://signal.org/blog/whatsapp-complete/, retrieved: May, 2025.

[103] E. Kret and R. Schmidt, *The pqxdh key agreement protocol*, https://signal.org/docs/specifications/pqxdh/, retrieved: May, 2025.

[104] Apple Security Engineering and Architecture (SEAR), *iMessage with PQ3: the new state of the art in quantum-secure messaging at scale*, https://security.apple.com/blog/imessage-pq3/, retrieved: May, 2025.

[105] A. M. Antonopoulos, *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*, 1st. O'Reilly Media, Inc., 2014, ISBN: 1449374042.

[106] novo nordisk, *Blockchain x clinical trials*, https://techlife.novonordisk.com/cases/epid, retrieved: May, 2025.

[107] DHL Trend Research, *Blockchain in logistics*, https://www.dhl.com/content/dam/dhl/global/core/documents/pdf/glo-core-blockchain-trend-report.pdf, retrieved: May, 2025.

[108] T. M. Fernández-Caramès and P. Fraga-Lamas, "Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks", *IEEE Access*, vol. 8, pp. 21 091–21 116, 2020. DOI: 10.1109/ACCESS.2020.2968985.

[109] N. K. Parida, C. Jatoth, V. D. Reddy, M. M. Hussain, and J. Faizi, "Post-quantum distributed ledger technology: A systematic survey", *Scientific Reports*, 2023. DOI: 10.1038/s41598-023-47331-1.

[110] D. A. Bard, J. J. Kearney, and C. A. Perez-Delgado, "Quantum advantage on proof of work", *Array*, vol. 15, p. 100 225, 2022, ISSN: 2590-0056. DOI: https://doi.org/10.1016/j.array.2022.100225.

[111] H. Gharavi, J. Granjal, and E. Monteiro, "Post-quantum blockchain security for the internet of things: Survey and research directions", *IEEE Communications Surveys & Tutorials*, vol. 26, no. 3, pp. 1748–1774, 2024. DOI: 10.1109/COMST.2024.3355222.

[112] Algorand, *Leading on post-quantum technology*, https://algorand.co/technology/post-quantum, retrieved: May, 2025.

[113] M. Misiaszek-Schreyner, Ł. Kujawski, M. Kosik, P. Kulicki, and M. Sopek, *The QSB, quantum secured blockchain, a whitepaper*, https://www.quantumblockchains.io/wp-content/uploads/2023/06/QBCK_WhitePaper.pdf, retrieved: May, 2025.

[114] Ethereum, *Future-proofing Ethereum*, https://ethereum.org/en/roadmap/future-proofing/, retrieved: May, 2025.

[115] N. Anhao, *Bitcoin Post-Quantum*, https://bitcoinpq.org/download/bitcoinpq-whitepaper-english.pdf, retrieved: May, 2025.

[116] Fraunhofer HHI, *Quantum Key Distribution System for Future-Proof Security*, https://www.hhi.fraunhofer.de/en/departments/pn/research-groups/free-space-optical-systems/quantum-key-distribution-1.html, retrieved: May, 2025.

[117] ID Quantique, *Providing the ultimate, long-term data protection in a post-quantum world*, https://www.idquantique.com/quantum-safe-security/products/#quantum_key_distribution, retrieved: May, 2025.

[118] French Cybersecurity Agency (ANSSI), Federal Office for Information Security (BSI), Netherlands National Communications Security Agency (NLNCSA), Swedish National Communications Security Authority, Swedish Armed Forces, *Position Paper on Quantum Key Distribution*, https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Quantum_Positionspapier.pdf, retrieved: May, 2025.

[119] National Security Agency/Central Security Service, *Quantum Key Distribution (QKD) and Quantum Cryptography (QC)*, https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/, retrieved: May, 2025.