Received December 24, 2019, accepted January 18, 2020, date of publication January 23, 2020, date of current version February 4, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2968985*

# Towards Post-Quantum Blockchain: A Review on Blockchain Cryptography Resistant to Quantum Computing Attacks

**TIAGO M. FERNÁNDEZ-CARAMÉS[1], (Senior Member, IEEE), AND PAULA FRAGA-LAMAS[1], (Member, IEEE)**

[1]Department of Computer Engineering, Faculty of Computer Science, Centro de Investigación CITIC, Universidade da Coruña, 15071 Coruña, Spain

Corresponding authors: Tiago M. Fernández-Caramés (tiago.fernandez@udc.es) and Paula Fraga-Lamas (paula.fraga@udc.es)

This work was supported in part by the Xunta de Galicia under Grant ED431G2019/01, in part by the Agencia Estatal de Investigación of Spain under Grant TEC2016-75067-C4-1-R and Grant RED2018-102668-T, and in part by the European Regional Development Fund (ERDF) Funds of the EU (AEI/FEDER, UE).

**ABSTRACT** Blockchain and other Distributed Ledger Technologies (DLTs) have evolved significantly in the last years and their use has been suggested for numerous applications due to their ability to provide transparency, redundancy and accountability. In the case of blockchain, such characteristics are provided through public-key cryptography and hash functions. However, the fast progress of quantum computing has opened the possibility of performing attacks based on Grover's and Shor's algorithms in the near future. Such algorithms threaten both public-key cryptography and hash functions, forcing to redesign blockchains to make use of cryptosystems that withstand quantum attacks, thus creating which are known as post-quantum, quantum-proof, quantum-safe or quantum-resistant cryptosystems. For such a purpose, this article first studies current state of the art on post-quantum cryptosystems and how they can be applied to blockchains and DLTs. Moreover, the most relevant post-quantum blockchain systems are studied, as well as their main challenges. Furthermore, extensive comparisons are provided on the characteristics and performance of the most promising post-quantum public-key encryption and digital signature schemes for blockchains. Thus, this article seeks to provide a broad view and useful guidelines on post-quantum blockchain security to future blockchain researchers and developers.

**INDEX TERMS** Blockchain, blockchain security, DLT, post-quantum, quantum-safe, quantum-resistant, quantum computing, cryptography, cryptosystem, cybersecurity.

## I. INTRODUCTION

Blockchain is a technology that was born with the cryptocurrency Bitcoin [1] and that is able to provide secure communications, data privacy, resilience and transparency [2]. A blockchain acts as a distributed ledger based on a chain of data blocks linked by hashes that allow for sharing information among peers that do not necessarily trust each other, thus providing a solution for the double-spending problem [3]–[5]. Such features have popularized blockchain in the last years and it has already been suggested as a key technology for different applications related to smart health [6], measuring systems [7], logistics [8], [9], e-voting [10] or smart factories [11], [12].

Blockchain users interact securely with the blockchain by leveraging public-key/asymmetric cryptography, which is essential for authenticating transactions. Hash functions are also key in a blockchain, since they allow for generating digital signatures and for linking the blocks of a blockchain. The problem is that both public-key cryptosystems and hash functions are threatened by the evolution of quantum computers. In the case of public-key cryptosystems, secure transaction data may be recovered fast by future quantum computing attacks. Such attacks impact the most popular public-key algorithms, including RSA (Rivest, Shamir, Adleman) [13], ECDSA (Elliptic Curve Digital Signature Algorithm) [14], [15], ECDH (Elliptic Curve Diffie-Hellman) [16] or DSA (Digital Signature Algorithm) [17], which can be broken in polynomial-time with Shor's algorithm [18] on a sufficiently powerful quantum computer. Moreover, quantum computers can make use of Grover's algorithm [19] to

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

accelerate the generation of hashes, which enables recreating the entire blockchain. Furthermore, Grover's algorithm may be adapted to detect hash collisions, which can be used to replace blocks of a blockchain while preserving its integrity.

This article analyzes how to evolve blockchain cryptography (i.e., its public-key security algorithms and hash functions) so that it can resist quantum computing attacks based on Grover's and Shor's algorithms, thus deriving into the creation of post-quantum blockchains. To guide researchers on the development of such a kind of blockchains, this article first provides a broad view on the current state of the art of post-quantum cryptosystems. Specifically, the most relevant post-quantum cryptosystems for blockchains are analyzed, as well as their main challenges. Furthermore, extensive comparisons are provided on the characteristics and performance of the most promising post-quantum public-key encryption and digital signature schemes.

The rest of this article is structured as follows. Section II describes the essential concepts related to blockchain and to its security primitives. Section III studies the impact of quantum attacks on blockchain public-key security schemes and on the most popular hash functions. In addition, Section III enumerates the most relevant post-quantum initiatives, emphasizing the ones related to blockchain and indicating the main features that a blockchain post-quantum scheme would need to provide. Section IV reviews the main types of post-quantum public-key and digital signature schemes, and analyzes their application to blockchain. Section V studies the performance of the most promising post-quantum cryptosystems when running them on hardware that can be used by blockchain nodes. Section VI details the main blockchain proposals that have already considered the use of post-quantum schemes. Section VII indicates the most significant challenges currently posed by post-quantum blockchain schemes and points at different paths to be followed by future researchers and developers. Finally, Section VIII summarizes the most relevant findings of this review article and Section IX is dedicated to conclusions.

## II. BLOCKCHAIN BASICS AND CRYPTOGRAPHIC PRIMITIVES

### A. TERMINOLOGY AND KEY CONCEPTS

Before starting to review the state of the art on post-quantum blockchains (i.e., on blockchains whose cryptosystems can resist quantum computing attacks), it is necessary to introduce several basic concepts, since some of the terminology may vary in the literature from one author to another.

It is first important to note that the concept of blockchain has evolved significantly since its original definition for Bitcoin [1]. In fact, researchers are still discussing the different elements that a blockchain has to contemplate to be actually considered a blockchain. The most common definition of blockchain is the one given in the Introduction of this article: it is a public ledger that stores data (e.g., transaction information, an event log) that are shared among multiple entities that do not necessarily trust each other. Every transaction on the blockchain is verified and stored by following a consensus protocol. Once a transaction is stored, ideally, it cannot be removed from the blockchain without making a significant computational effort.

A blockchain node is a computational entity able to perform operations on the blockchain. It is common to distinguish between regular blockchain nodes, which only interact with the blockchain, and full nodes, which have a copy of the blockchain and contribute to it by validating transactions. A blockchain miner is a third type of node that is present in many blockchains and whose contribution is essential during blockchain transaction validations: to carry out the validation, they perform certain actions following a consensus protocol. There are many consensus protocols [20], being some of the most popular Proof-of-Work (PoW) (used by Bitcoin), the variants of the Byzantine Fault Tolerance (BFT) methods [21] or Proof-of-Stake (PoS).

The concept of smart contract is also relevant: it is a piece of code stored on the blockchain that can be executed autonomously. Smart contracts can be used to automate certain tasks depending on the state of the blockchain and in other external data sources called oracles [22].

The previously introduced concepts have contributed to the success of blockchain and to its main security features:

- Decentralization. If one node of the blockchain is attacked or shut down, its information keeps on being available from the other blockchain nodes.
- Data privacy and integrity. Blockchain uses public-key cryptography and hash functions for providing data privacy, integrity and authentication.
- Data immutability. Once a transaction is stored on the blockchain, it is not possible to make further modifications on it (the only exception is blockchain forks [2], which require to reach a consensus among the entities that participate on the blockchain).

A detailed description on the inner workings of the previously mentioned blockchain components and algorithms is out of the scope of this paper, but the interested reader can find further information in [2], [22]–[28].

### B. BLOCKCHAIN SECURITY PRIMITIVES

The security features provided by blockchain are essentially sustained by public-key/asymmetric cryptography and hash functions, whose role in blockchain security is detailed in the next subsections.

#### 1) PUBLIC-KEY CRYPTOGRAPHY

A blockchain usually makes use of public-key cryptosystems for securing information exchanges between parties by authenticating transactions through digital signatures. During the signature process, the signer signs with a private key, while the public key, which is shared publicly, is used to verify that the signature is valid. Thus, when a signing algorithm is secure, it is guaranteed that only the person with a private key could have generated certain signature. For instance, Bitcoin uses ECDSA signatures with the Koblitz curve secp256k1, which depends on a private key for signing messages and on the corresponding public key for checking the signature.

Public-key cryptography is also essential for the so-called wallets, which are private key containers that store files and simple data. Thus, in a blockchain system each user has a wallet that is associated with at least a public address (usually a hash of the user public key) and a private key that the user needs for signing transactions. For instance, in blockchains like Bitcoin every transaction ends up being 'sent' to the public address of the receiver and is signed with the private key of the sender. In order to spend bitcoins, their owner has to demonstrate the ownership of a private key. To verify the authenticity of the received currency, every entity that receives bitcoins verifies its digital signature by using the public key of the sender.

#### 2) HASH FUNCTIONS

Hash functions like SHA-256 or Scrypt are commonly used by blockchains because they are easy to check, but really difficult to forge, thus allowing the generation of digital signatures that blockchain users need to authenticate themselves or their data transactions in front of others.

Hash functions are also used by blockchains to link their blocks (i.e., groups of transactions that are considered to occur at the same time instant). Such blocks are linked in chronological order, containing each block the hash of the previous block. It is straightforward to hash a block of a blockchain, but some blockchains like Bitcoin restrict block hashing to make it meet a specific mathematical condition (e.g., the hash should contain a number of leading zeros [1]), which slows down block addition.

Finally, it is worth mentioning that hash functions are used in blockchains for generating user addresses (i.e., user public/private keys) or for shortening the size of public addresses [29], [30].

## III. FROM PRE-QUANTUM TO POST-QUANTUM BLOCKCHAIN

### A. BLOCKCHAIN PUBLIC-KEY SECURITY

It must be first noted that public-key cryptosystems strength against classical computing attacks has been traditionally estimated through the so-called bits-of-security level. Such a level is defined as the effort required by a classical computer to perform a brute-force attack. For instance, an asymmetric cryptosystem has a 1024-bit security when the effort required to attack it with a classical computer is similar to the one needed to carry out a brute-force attack on a 1024-bit cryptographic key. As a reference, Table 1 indicates the security level of some of the most popular symmetric and asymmetric cryptosystems.

**TABLE 1. Reference security levels for popular symmetric and asymmetric cryptosystems (source: [31]).**

| Security Level | Symmetric Cryptosystem Key Size | RSA Key Size | ECDSA Curve Key Size |
|---|---|---|---|
| 80 | 2TDEA (112 bits) | 1024 bits | prime192v1 (192 bits) |
| 112 | 3TDEA (168 bits) | 2048 bits | secp224r1 (224 bits) |
| 128 | AES-128 (128 bits) | 3072 bits | secp256r1 (256 bits) |
| 192 | AES-192 (192 bits) | 7680 bits | secp384r1 (384 bits) |

The cost of breaking current 80-bit security cryptosystems with classical computers is estimated to be between tens of thousands and hundreds of millions of dollars. In the case of 112-bit cryptosystems, they are considered to be secure to classical computing attacks for the next 30 to 40 years [32]. However, researchers have determined that 160-bit elliptic curves can be broken with a 1000-qubit quantum computer, while 1024-bit RSA would need roughly 2,000 qubits [33]. Such a threat affects not only cryptosystems that rely on integer factorization (e.g., RSA) or elliptic curves (e.g., ECDSA, ECDH), but also others based on problems like the discrete logarithm problem [34], which can be solved fast through Shor's algorithm.

As of writing, powerful quantum computers are not available: the most powerful quantum computer (claimed by IonQ) has only 79 qubits and even technologically-advanced organizations like the U.S. National Security Agency (NSA) seem to have not made significant progress on large quantum computers [35]. However, it is estimated that in the next 20 years such a kind of computers will be functional enough to be able to break easily current strong public-key cryptosystems [36]. In fact, organizations like the NSA have already warned on the impact of quantum computing on IT products and recommended increasing the ECC (Elliptic Curve Cryptography) security level of certain cryptographic suites [34]. Although some researchers have speculated on the real reasons behind such an NSA announcement [37], long-term public-key cryptography seems to be threatened and developers need to prepare current blockchains for the post-quantum era.

Table 2 indicates the main characteristics of the most relevant public-key cryptosystems that are affected by the quantum threat. The Table also includes the characteristics of other relevant cryptosystems that will be broken or that will be impacted significantly by quantum attacks related to Shor's and Grover's algorithms.

### B. HASH FUNCTION SECURITY

In contrast to public-key cryptosystems, traditional hash functions are considered to be able to withstand quantum attacks since it seems unlikely the development of quantum algorithms for NP-hard problems [38]. Although new hash functions have been recently proposed by academics to resist quantum attacks [39], it is usually recommended to increase the output size of traditional hash functions.

**TABLE 2.** Main blockchain and popular cryptosystems impacted by the quantum threat.

| Algorithm | Main Affected Blockchains/DLTs | Function | Pre-Quantum Security Level | Estimated Post-Quantum Security Level | Key Size | Hash/Signature Size |
|---|---|---|---|---|---|---|
| SHA-256 | Bitcoin, Ethereum, Dash, Litecoin, Zcash, Monero, Ripple, NXT, Byteball | Hash function | 256 bits | 128 bits (Grover) | - | 256 bits |
| Ethash (Keccak-256, Keccak-512) | Ethereum | Hash function | 256/512 bits | 128/256 bits (Grover) | - | 256/512 bits |
| Scrypt | Litecoin, NXT | Hash function | 256 bits | 128 bits (Grover) | - | 256 bits |
| RIPEMD160 | Bitcoin, Ethereum, Litecoin, Monero, Ripple, Bytecoin | Hash function | 160 bits | 80 bits (Grover) | - | 160 bits |
| Keccak-256 | Monero, Bytecoin | Hash function | 256 bits | 128 bits (Grover) | - | 256 bits |
| Keccak-384 | IOTA | Hash function | 384 bits | 192 bits (Grover) | - | 384 bits |
| ECDSA | Bitcoin, Ethereum, Dash, Litecoin, Zcash, Ripple, Byteball | Signature | 128 bits | Broken (Shor) | 256 bits | 520 bits |
| RSA-1024 | - | Signature, Encryption | 80 bits | Broken (Shor) | 1024 bits | 1024 bits |
| RSA-2048 | - | Signature, Encryption | 112 bits | Broken (Shor) | 2048 bits | 2048 bits |
| RSA-3072 | - | Signature, Encryption | 128 bits | Broken (Shor) | 3072 bits | 3072 bits |
| DSA-3072 | - | Signature | 128 bits | Broken (Shor) | 3072 bits | - |
| SHA-3 256 | - | Hash function | 256 bits | 128 bits (Grover) | - | 256 bits |
| AES-128 | - | Symmetric Encryption | 128 bits | 64 bits (Grover) | 128 bits | - |
| AES-256 | - | Symmetric Encryption | 256 bits | 128 bits (Grover) | 256 bits | - |

This recommendation is related to quantum attacks that can make use of Grover's algorithm to accelerate brute force attacks by a quadratic factor [36]. Specifically, Grover's algorithm can be used in two ways to attack a blockchain:

- First, to search for hash collisions and then replace entire blockchain blocks. For instance, in the specific case of the work described in [41], it is proposed to use Grover's algorithm to find collisions in hash functions, concluding that a hash function would have to output 3*n bits to provide a n-bit security level. Such a conclusion means that many current hash functions would not be valid for the post-quantum era, while others like SHA-2 or SHA-3 will have to increase their output size.
- Second, Grover's algorithm can be used to accelerate mining in blockchains like Bitcoin (i.e., it is able to speed up the generation of nonces), which would allow for recreating entire blockchains fast, thus undermining their integrity.

In addition, quantum attacks through Shor's algorithm also impact hash functions: if a blockchain hash function is broken, someone with a powerful enough quantum computer may use Shor's algorithm to forge digital signatures, to impersonate blockchain users and to steal their digital assets.

As a reference, Table 2 includes the main characteristics of the most popular hash functions that are currently used by relevant blockchains and indicates the impact of quantum computing on their security level.

### C. POST-QUANTUM BLOCKCHAIN INITIATIVES

Post-quantum cryptography is currently a hot topic that has been addressed by research projects (e.g., PQCrypto [42], SAFEcrypto [43], CryptoMathCREST [44] or PROMETHEUS [45]), standardization initiatives [46]–[53] and workshops [54]–[56], which obtained relevant results [57]–[59] and produced interesting reports [32], [60]–[65]. Among the previously mentioned initiatives, it is worth noting the NIST call for proposals for post-quantum public-key cryptosystems [66], which is currently in its second round [67] and which is expected to deliver the first standard drafts between 2022 and 2024.

Although the previous projects and initiatives generated very valuable results, they were not explicitly focused on post-quantum blockchains. However, there have been specific post-quantum initiatives related to the most popular blockchains. For instance, Bitcoin Post-Quantum is an experimental branch of Bitcoin's main blockchain that uses a post-quantum digital signature scheme [68]. Another example is Ethereum 3.0, which plans to include quantum-resistant components like zk-STARKs (Zero-Knowledge Scalable Transparent ARguments of Knowledge) [69]. Other blockchain platforms like Abelian [70] have suggested using lattice-based post-quantum cryptosystems to prevent quantum attacks, while certain blockchains such as Corda are experimenting with post-quantum algorithms like SPHINCS [71].

### D. IDEAL CHARACTERISTICS OF BLOCKCHAIN POST-QUANTUM SCHEMES

In order to be efficient, a post-quantum cryptosystem would need to provide blockchains with the following main features:

- Small key sizes. The devices that interact with a blockchain need to ideally make use of small public and private keys in order to reduce the required storage space. In addition, small keys involve less complex computational operations when managing them. This is especially important for blockchains that require the interaction of Internet of Things (IoT) end-devices, which are usually constrained in terms of storage and computational power. It is worth indicating that IoT, like other emerging technologies (e.g., deep learning [72]), has experienced a significant growth in the last years [73]–[77], but IoT devices still face some important challenges, mainly regarding security [78]–[82], which are limiting to some extent its jointly use with blockchain and its widespread adoption.
- Small signature and hash length. A blockchain essentially stores data transactions, including user signatures and data/block hashes. Therefore, if signature/hash length increases, blockchain size will also increase as well.
- Fast execution. Post-quantum schemes need to be as fast as possible in order to allow a blockchain to process a large amount of transactions per second. Moreover, a fast execution usually involves low computational complexity, which is necessary to not to exclude resource-constrained devices from blockchain transactions.
- Low computational complexity. This feature is related to a fast execution, but it is important to note that a fast execution with certain hardware does not imply that the post-quantum cryptosystem is computationally simple. For instance, some schemes can be executed fast in Intel microprocessors that make use of the Advanced Vector Extensions 2 (AVX2) instruction set, but the same schemes may be qualified as slow when executed on ARM-based microcontrollers. Therefore, it is necessary to look for a trade-off between computational complexity, execution time and supported hardware devices.
- Low energy consumption. Some blockchains like Bitcoin are considered to be power hungry mainly due to the energy required to execute its consensus protocol. There are other factors that impact power consumption, like the used hardware, the amount of performed communications transactions and, obviously, the implemented security schemes, which can draw a relevant amount of current due to the complexity of the performed operations [83], [84].

## IV. POST-QUANTUM CRYPTOSYSTEMS FOR BLOCKCHAIN

There are four main types of post-quantum cryptosystems and a fifth kind that actually mixes both pre-quantum and post-quantum cryptosystems. The following subsections analyze the potential application of such schemes for the implementation of encryption/decryption mechanisms and for signing blockchain transactions.

A detailed description on the algorithms cited in the next subsections is out of the scope of this article, but the interested reader can consult the specific references cited throughout the text and books like [85], which provide a wide but comprehensive description of the most popular post-quantum cryptosystems.

As a summary, the five different types of post-quantum cryptosystems are depicted in Figure 1 together with examples of encryption and digital signature scheme implementations.

### A. PUBLIC-KEY POST-QUANTUM CRYPTOSYSTEMS

#### 1) CODE-BASED CRYPTOSYSTEMS

They are essentially based on the theory that supports error-correction codes. For instance, McEliece's cryptosystem is an example of code-based cryptosystem [86] that dates back from the 70s and whose security is based on the syndrome decoding problem [87]. McEliece's scheme provides fast encryption and relatively fast decryption, which is an advantage for performing rapid blockchain transactions. However, McEliece's cryptosystem requires to store and perform operations with large matrices that act as public and private keys. Such matrices usually occupy between 100 kilobytes and several megabytes, which may be a restriction when resource-constrained devices are involved. To tackle this issue, future researchers will have to study matrix compression techniques, as well as the use of different codes (e.g., Low-Density Parity-Check (LDPC) codes, Quasi-Cyclic Low-Rank Parity-Check (QC-LRPC) codes) and specific coding techniques [88].

As a reference, Table 3 compares the main characteristics of the public-key code-based post-quantum encryption cryptosystems that passed to the second round of the NIST call. There are other post-quantum cryptosystems [89], but the NIST second-round candidates are specially interesting due to their standardization chances and because they have been already thoroughly analyzed by the cryptographic community.

It is important to note that the parameters of the algorithms compared in Table 3 can be adjusted according to the required security and thus key size and performance may vary among them. Specifically, the cryptosystems of the Table were selected with the objective of comparing the ones with the smallest key sizes that provided the main quantum security levels demanded by NIST (128, 192 and 256 bits). The same criteria were applied for the selection of the algorithms compared in the rest of this article.

As it can be observed in Table 3, the evaluated code-based cryptographic schemes provide between 128 and 256 bits of classical security, but such a level is reduced significantly in terms of quantum security. Regarding the compared

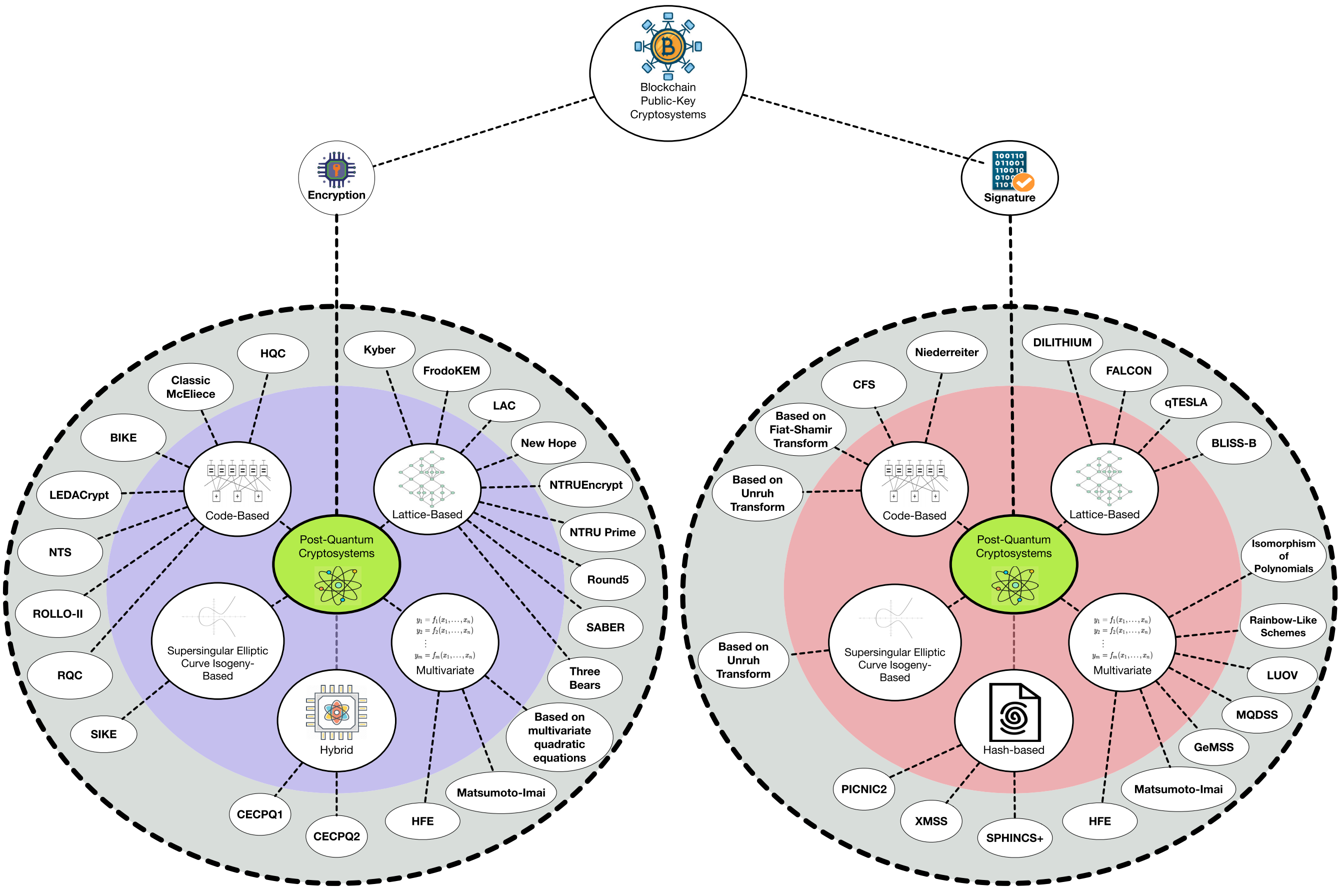

**FIGURE 1.** Post-quantum public-key cryptosystem taxonomy and main practical implementations.

public/private key sizes, they range between very small sizes (320 bits, for the private keys of ROLLO-II and RQC) and up to 15.5 KB (for the public key of the highest security level of HQC). On average, even when making use of compression techniques, the size of code-based scheme keys is clearly larger than the one required by current ECDSA and RSA-based encryption systems.

It is worth pointing out that in the case of HQC two key sizes are indicated: the one inside parentheses is related to the use of a seed expander. However, note that during the execution of the algorithm an expanded key will consume the amount of memory indicated outside the parentheses and will also need to perform the expansion operation, which slows down the execution of the algorithm.

Overall, among the schemes compared in Table 3, it seems that RQC-II provides the best trade-off between security and key size, although it is not among the fastest post-quantum schemes (the performance of the algorithms in Table 3 is analyzed later in Section V).

### 2) MULTIVARIATE-BASED CRYPTOSYSTEMS

Multivariate-based schemes rely on the complexity of solving systems of multivariate equations, which have been demonstrated to be NP-hard or NP-complete [85]. Despite their resistance to quantum attacks, it is necessary further research for improving their decryption speed (due to the involved ‘‘guess work’’) and to reduce their large key size and ciphertext overhead [90].

Currently, some of the most promising multivariate-based schemes are the ones based on the use of square matrices with random quadratic polynomials, the cryptosystems derived from Matsumoto-Imai’s algorithm and the schemes that rely on Hidden Field Equations (HFE) [91]–[93].

### 3) LATTICE-BASED CRYPTOSYSTEMS

This kind of cryptographic schemes are based on lattices, which are sets of points in n-dimensional spaces with a periodic structure. Lattice-based security schemes rely on the presumed hardness of lattice problems like the Shortest Vector Problem (SVP), which is an NP-hard problem whose objective is to find the shortest non-zero vector within a lattice. There are other similar lattice-related problems like the Closest Vector Problem (CVP) or the Shortest Independent Vectors Problem (SIVP) [94], which nowadays cannot be solved efficiently through quantum computers.

Lattice-based schemes provide implementations that allow for speeding up blockchain user transactions since they are often computationally simple, so they can be executed fast

**TABLE 3.** Post-quantum code-based public-key encryption schemes that passed to the second round of the NIST call.

| Cryptosystem | Subtype | Claimed Quantum Security | Claimed Classical Security | Public Key Size (bits) | Private Key Size (bits) | Key References |
|---|---|---|---|---|---|---|
| BIKE-1 Level 1 | QC-MDPC McEliece | - | 128 bits | 20,326 | 2,130 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-1 Level 3 | QC-MDPC McEliece | - | 192 bits | 39,706 | 3,090 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-1 Level 5 | QC-MDPC McEliece | - | 256 bits | 65,498 | 4,384 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-2 Level 1 | QC-MDPC Niederreiter | - | 128 bits | 10,163 | 2,130 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-2 Level 3 | QC-MDPC Niederreiter | - | 192 bits | 19,853 | 3,090 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-2 Level 5 | QC-MDPC Niederreiter | - | 256 bits | 32,749 | 4,384 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-3 Level 1 | QC-MDPC Ouroboros | - | 128 bits | 22,054 | 2,010 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-3 Level 3 | QC-MDPC Niederreiter | - | 192 bits | 43,366 | 2,970 | [95], [96], [97], [98], [99], [100], [101] |
| BIKE-3 Level 5 | QC-MDPC Niederreiter | - | 256 bits | 72,262 | 4,256 | [95], [96], [97], [98], [99], [100], [101] |
| Classic McEliece (mceliece8192128) | Niederreiter's dual version using binary Goppa codes | - | 256 bits | 10,862,592 | 112,640 | [102], [103], [104], [105] |
| HQC Level 1 (hqc-128-1) | Quasi-Cyclic and BCH codes | 64 bits | 128 bits | 49,360 (25,000) | 2,016 (320) | [106], [107], [108] |
| HQC Level 3 (hqc-192-1) | Quasi-Cyclic and BCH codes | 96 bits | 192 bits | 87,344 (43,992) | 3,232 (320) | [106], [107], [108] |
| HQC Level 5 (hqc-256-1) | Quasi-Cyclic and BCH codes | 128 bits | 256 bits | 127,184 (63,912) | 4,256 (320) | [106], [107], [108] |
| LEDACrypt KEM Level 1 (for two circulant blocks) | QC-LDPC Niederreiter | - | 128 bits | 14,976 | 3,616 (192) | [109], [110], [111] |
| LEDACrypt KEM Level 3 (for two circulant blocks) | QC-LDPC Niederreiter | - | 192 bits | 25,728 | 5,152 (256) | [109], [110], [111] |
| LEDACrypt KEM Level 5 (for two circulant blocks) | QC-LDPC Niederreiter | - | 256 bits | 36,928 | 6,112 (320) | [109], [110], [111] |
| NTS-KEM Level 1 | Based on McEliece and Niederreiter | 64 bits | 128 bits | 2,555,904 | 73,984 | [112], [113] |
| NTS-KEM Level 3 | Based on McEliece and Niederreiter | 96 bits | 192 bits | 7,438,080 | 140,448 | [112], [113] |
| NTS-KEM Level 5 | Based on McEliece and Niederreiter | 128 bits | 256 bits | 11,357,632 | 159,376 | [112], [113] |
| ROLLO-II 128 | Based on rank metric codes with LRPC codes | - | 128 bits | 12,368 | 320 | [114], [115] |
| ROLLO-II 192 | Based on rank metric codes with LRPC codes | - | 192 bits | 16,160 | 320 | [114], [115] |
| ROLLO-II 256 | Based on rank metric codes with LRPC codes | - | 256 bits | 19,944 | 320 | [114], [115] |
| RQC-I | Based on Rank Quasi-Cyclic codes | - | 128 bits | 6,824 | 320 | [108], [116], [117] |
| RQC-II | Based on Rank Quasi-Cyclic codes | - | 192 bits | 11,128 | 320 | [108], [116], [117] |
| RQC-III | Based on Rank Quasi-Cyclic codes | - | 256 bits | 18,272 | 320 | [108], [116], [117] |

and in an efficient way. However, like it occurs with other post-quantum schemes, lattice-based implementations need to store and make use of large keys, and involve large ciphertext overheads. For example, lattice-based schemes like NTRU [118] or NewHope [119] often require to manage keys in the order of a few thousand bits.

As of writing, the most promising lattice-based cryptosystems are based on polynomial algebra [118], [120], [121] and on the Learning With Errors (LWE) problem and its variants (e.g., LP-LWE (Lindner-Peikert LWE) or Ring-LWE [122], [123]).

Table 4 compares the public-key lattice cryptosystems that passed to the second round of the NIST call. As it can be observed in the Table, the included schemes provide a classical security between 128 and 368 bits and a quantum security between 84 and 300 bits, so their complexity differs significantly depending on the algorithm and on the provided security level. Key size also fluctuates remarkably: from the 128-bit private key of the IoT version of Round5, to the 344,704-bit private key of FrodoKEM-1344. As it was previously mentioned for the code-based encryption schemes, seed expanders can be used to compress keys. The lattice-based cryptosystems that use seed expanders are shown in Table 4 by indicating two key sizes (the key size required when using a seed expander is inside parentheses).

Among the cryptosystems compared in Table 4 that provide a roughly 100-bit quantum security level, it seems that Round5 KEM IoT is the one with the smallest keys and, as it will be later observed in Section V, it provides a fast execution.

#### 4) SUPERSINGULAR ELLIPTIC CURVE ISOGENY CRYPTOSYSTEMS

These schemes are based on the isogeny protocol for ordinary elliptic curves presented in [124], but enhanced to withstand the quantum attack detailed in [125]. There are different promising post-quantum cryptosystems of this type [126], [127], whose key size is usually in the order of a few thousand bits [128].

Only one isogeny-based public-key encryption scheme passed to the second round of the NIST call: SIKE [129], [130]. SIKE is based on pseudo-random walks in supersingular isogeny graphs. A good reference of SIKE key sizes is SIKEp434, which, for a 128-bit level of classical security, makes use of a 2640-bit public-key and a 2992-bit private key.

#### 5) HYBRID CRYPTOSYSTEMS

Hybrid schemes seem to be next step towards post-quantum security, since they merge pre-quantum and post-quantum cryptosystems with the objective of protecting the exchanged data both from quantum attacks and from attacks against the used post-quantum schemes, whose security is currently being evaluated by industry and academia.

This kind of cryptosystems have been tested by Google [152], which merged New Hope [119] with an ECC-based Diffie-Hellman key agreement scheme named X25519. A second version of the hybrid scheme (CECPQ2) is currently being tested: it merges X25519 with instantiations of NTRU (HRSS (Hülsing, Rijneveld, Schanck, Schwabe) and SXY (Saito, Xagawa, Yamakawa)).

Although these schemes look promising, it must be noted that they involve implementing two complex cryptosystems, which require significant computational resources and more energy consumption. Therefore, future developers of hybrid post-quantum cryptosystems for blockchains will have to look for a trade-off between security, computational complexity and resource consumption. In addition, developers will have to address the large payload problem that arises with this kind of cryptosystems when providing Transport Layer Security (TLS) communications (such a problem is due to the required public-key and ciphertext sizes).

### B. POST-QUANTUM SIGNING ALGORITHMS

#### 1) CODE-BASED CRYPTOSYSTEMS

Different post-quantum code-based signing algorithms have been proposed in the past. Some of the most relevant subtypes of this kind of cryptosystems are based on the schemes from Niederreiter [153] and CFS (Courtois, Finiasz, Sendrier) [154], which are really similar to McEliece's cryptosystem. The signatures of such schemes are short in length and can be verified really fast, but, as it occurs with traditional McEliece's cryptosystems, the use of large key sizes requires significant computational resources and, as a consequence, signature generation may become inefficient.

Other code-based signing algorithms have been proposed in the literature, such as identification protocols related to the application of Fiat-Shamir transformation [155], which in some cases outperform cryptosystems like CFS [156]. Nonetheless, it must be noted that, Fiat-Shamir signatures are not known to be completely secure against quantum attacks [157] (only under certain circumstances [158]), so alternatives like the Unruh transformation should be considered [157].

#### 2) MULTIVARIATE-BASED CRYPTOSYSTEMS

In this kind of signature schemes the public key is generated through a trapdoor function that acts as private key. This fact usually derives into large public keys, but very small signatures [85].

Some of the most popular multivariate-based schemes rely on Matsumoto-Imai's algorithm, on Isomorphism of Polynomials (IP) [159] or on variants of HFE, which are able to generate signatures with a size comparable to the currently used RSA or ECC-based signatures [160]. Other relevant multivariate-based digital signature schemes have been proposed, like the ones based on pseudo-random multivariate quadratic equations [161] or on Rainbow-like signing schemes (e.g., TTS [162], TRMS [163] or Rainbow [164]). Nonetheless, such cryptosystems need to be further improved

**TABLE 4.** Post-quantum lattice-based public-key encryption schemes that passed to the second round of the NIST call.

| Cryptosystem | Subtype | Claimed Quantum Security | Claimed Classical Security | Public Key Size (bits) | Private Key Size (bits) | Key References |
|---|---|---|---|---|---|---|
| CRYSTALS Kyber-512 | Based on solving the LWE problem with Module Lattices | 100 bits | 128 bits | 6,400 | 13,056 (256) | [131], [132], [133], [134] |
| CRYSTALS Kyber-512 90s | Based on solving the LWE problem with Module Lattices | 100 bits | 128 bits | 6,400 | 13,056 (256) | [131], [132], [133], [134] |
| CRYSTALS Kyber-768 | Based on solving the LWE problem with Module Lattices | 164 bits | 192 bits | 9,472 | 19,200 (256) | [131], [132], [133], [134] |
| CRYSTALS Kyber-768 90s | Based on solving the LWE problem with Module Lattices | 164 bits | 192 bits | 9,472 | 19,200 (256) | [131], [132], [133], [134] |
| CRYSTALS Kyber-1024 | Based on solving the LWE problem with Module Lattices | 230 bits | 256 bits | 12,544 | 25,344 (256) | [131], [132], [133], [134] |
| CRYSTALS Kyber-1024 90s | Based on solving the LWE problem with Module Lattices | 230 bits | 256 bits | 12,544 | 25,344 (256) | [131], [132], [133], [134] |
| FrodoKEM-640 AES | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 128 bits | 76,928 | 159,104 | [122], [135], [136], [137] |
| FrodoKEM-640 SHAKE | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 128 bits | 76,928 | 159,104 | [122], [135], [136], [137] |
| FrodoKEM-976 AES | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 192 bits | 125,056 | 250,368 | [122], [135], [136], [137] |
| FrodoKEM-976 SHAKE | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 192 bits | 125,056 | 250,368 | [122], [135], [136], [137] |
| FrodoKEM-1344 AES | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 256 bits | 172,160 | 344,704 | [122], [135], [136], [137] |
| FrodoKEM-1344 SHAKE | Based on solving the LWE problem with generic “algebraically unstructured” lattices | - | 256 bits | 172,160 | 344,704 | [122], [135], [136], [137] |
| LAC-128 (CCA) | Based on solving Ring-LWE | - | 128 bits | 4,352 | 8,448 | [138] |
| LAC-192 (CCA) | Based on solving Ring-LWE | - | 192 bits | 8,448 | 16,640 | [138] |
| LAC-256 (CCA) | Based on solving Ring-LWE | - | 256 bits | 8,448 | 16,640 | [138] |
| NewHope-512 (CCA) | Based on solving Ring-LWE | 101 bits | 128 bits | 7,424 | 15,104 | [119], [139], [140], [141] |
| NewHope-1024 (CCA) | Based on solving Ring-LWE | 233 bits | 256 bits | 14,592 | 29,440 | [119], [139], [140], [141] |
| NTRUEncrypt (ntruhrss701) | Based on solving LWE/Ring-LWE | 128-192 bits | - | 9,104 | 11,616 | [118], [142], [143] |
| NTRUEncrypt (ntruhps2048677) | Based on solving LWE/Ring-LWE | 128-192 bits | - | 7,448 | 9,880 | [118], [142], [143] |
| NTRUEncrypt (ntruhps4096821) | Based on solving LWE/Ring-LWE | 192-256 bits | - | 9,840 | 12,736 | [118], [142], [143] |
| NTRU Prime (sntrup4591761) | Based on solving Ring-LWE | 139-208 bits | 153-368 bits | 9,744 | 12,800 | [144], [145], [146] |
| NTRU Prime (ntrulpr4591761) | Based on solving Ring-LWE | 140-210 bits | 155-364 bits | 8,376 | 9,904 | [144], [145], [146] |
| Round5 KEM IoT | Based on General Learning with Rounding (GLWR) | 88-101 bits | 96-202 bits | 2,736 | 128 | [147], [148] |
| SABER KEM (LightSABER) | Based on Module Learning With Rounding problem (Mod-LWR) | 114-153 bits | 125-169 bits | 5,376 | 12,544 ( 7,936) | [149], [150] |
| SABER KEM | Based on Module Learning With Rounding problem (Mod-LWR) | 185-226 bits | 203-244 bits | 7,936 | 18,432 (10,752) | [149], [150] |
| SABER KEM (FireSABER) | Based on Module Learning With Rounding problem (Mod-LWR) | 257-308 bits | 283-338 bits | 10,496 | 24,320 (14,080) | [149], [150] |
| Three Bears (BabyBear CCA) | Based on integer module learning with errors (I-MLWE) | 140-180 bits | 154-190 bits | 6,432 | 320 | [151] |
| Three Bears (MamaBear CCA) | Based on integer module learning with errors (I-MLWE) | 213-228 bits | 235-241 bits | 9,552 | 320 | [151] |
| Three Bears (PapaBear CCA) | Based on integer module learning with errors (I-MLWE) | 285-300 bits | 314-317 bits | 12,672 | 320 | [151] |

in terms of key size, since they usually require several tens of thousands of bytes per key.

Table 5 compares the main characteristics of the digital signature schemes that passed to the second round of the NIST call. In such a Table, for schemes like Rainbow, the values inside parentheses indicate the length of the compressed keys. As it can be observed, among the compared multivariate-based cryptosystems, MQDSS provides really small keys, but the sizes of its signatures are among the largest in the comparison. In contrast, the rest of the compared multivariate-based schemes require several kilobytes for each key, but they produce short signatures (with a length between 239 and 1,632 bits).

### 3) LATTICE-BASED CRYPTOSYSTEMS

Among the different lattice-based signature schemes described in the literature, the ones based on Short Integer Solution (SIS) [165] seem to be promising due to their reduced key size. According to some performance analyses, BLISS-B (Bimodal Lattice Signatures B), which relies on the hardness of the SIS problem, provides one of the best performances for lattice-based signing cryptosystems, being on a par with RSA and ECDSA [166]. However, note that the original BLISS [167] was attacked in 2016 under specific conditions through a side-channel attack [168], while its variant BLISS-B is also susceptible to cache attacks that are able to recover the secret signing key after 6,000 signature generations [169].

Besides BLISS, there are in the literature other lattice-based signature schemes that rely on the SIS problem but that were devised specifically for blockchains [170]. Researchers have also developed lattice-based blind signature schemes [171], which were introduced by David Chaum in the early 80s for creating an untraceable payment system [172]. For instance, a lattice-based blind signature scheme is detailed in [173], which was specifically conceived for providing user anonymity and untraceability in distributed blockchain-based applications for IoT.

Finally, it is worth mentioning the lattice-based signature schemes presented in [174], [175]. Specifically, in [174] the authors propose a cryptosystem whose public and private keys are generated through Bonsai Trees [176]. Regarding the work in [175], it presents a lattice-based signature scheme optimized for embedded systems, which, for a 100-bit security level, makes use of a public key of 12,000 bits and a private key of 2,000 bits, and generates signatures of 9,000 bits. This latter scheme, due to its simplicity and efficiency, was selected as signature algorithm for blockchain-related developments like QChain [177], a post-quantum decentralized system for managing public-key encryption.

Table 5 allows for comparing the main characteristics of the lattice-based schemes that passed to the second round of the NIST call. As it can be observed, lattice-based signature schemes require keys whose size is in general smaller than the one needed by multivariate-based schemes, but the generated signatures are slightly larger. Among the compared lattice-based cryptosystems, FALCON makes use of the smallest key sizes and signature lengths. Other schemes like qTESLA are fast (as it will be later observed in Section V), but their major drawback is their large key sizes [192].

### 4) SUPERSINGULAR ELLIPTIC CURVE ISOGENY CRYPTOSYSTEMS

It is possible to use supersingular elliptic curve isogenies for creating post-quantum digital signature schemes [193], but there are not in the literature many of such schemes and they still suffer from poor performance. For instance, in [194] the authors present different signature schemes based on isogeny problems and on the Unruh transform, which makes use of small key sizes and relatively efficient signing and verification algorithms. Another signature scheme based on the Unruh transform is presented in [195], which, for a 128-bit quantum security level, makes use of a 336-byte public key and a 48-byte private key, but it generates 122,880-byte signatures (even when using compression techniques). Therefore, it is necessary to address key size issues when implementing isogeny-based cryptosystems and Supersingular Isogeny Diffie-Hellman (SIDH), especially in the case of resource-constrained devices, which need to use key compression techniques that often involve computationally intensive steps [196], [197].

### 5) HASH-BASED SIGNATURE SCHEMES

The security of these schemes depends on the security of the underlying hash function instead of on the hardness of a mathematical problem. This kind of schemes date back from the late 70s, when Lamport proposed a signature scheme based on a one-way function [198]. Currently, variants of eXtended Merkle Signature Scheme (XMSS) [199] like XMSS-T and SPHINCS [200] are considered promising hash-based signature schemes for the post-quantum era that derive from the Merkle tree scheme described in [201].

However, some researchers consider XMSS and SPHINCS to be impractical for blockchain applications due to their performance [202], so alternatives have been suggested. For example, XMSS has been adapted to blockchain by making use of a single authentication path instead of a tree, while using one-time and limited keys in order to preserve anonymity and minimize user tracking [203]. Other authors [202] proposed substituting XMSS with XNYSS (eXtended Naor-Yung Signature Scheme), a signature scheme that combines a hash-based one-time signature scheme with Naor-Yung chains, which allow for creating chains of related signatures [204].

## V. PERFORMANCE COMPARISON OF POTENTIAL BLOCKCHAIN POST-QUANTUM CRYPTOSYSTEMS

### A. PUBLIC-KEY ENCRYPTION SCHEMES

Tables 6 and 7 compare the post-quantum public-key encryption cryptosystems previously mentioned in Section IV when executed on hardware that can run both a regular blockchain

**TABLE 5.** Post-quantum digital signature schemes that passed to the second round of the NIST call.

| Algorithm | Type | Subtype | Claimed Quantum Security | Public-Key Size | Private Key Size | Signature Size | Key References |
|---|---|---|---|---|---|---|---|
| DILITHIUM 1280x1024 SHAKE (recommended) | Lattice-based | Based on the "Fiat-Shamir with Aborts" technique | 128 bits | 1,472 bytes | - | 2,701 bytes | [178] |
| DILITHIUM 1280x1024 AES (recommended) | Lattice-based | Based on the "Fiat-Shamir with Aborts" technique | 128 bits | 1,472 bytes | - | 2,701 bytes | [178] |
| FALCON-512 | Lattice-based | Based on SIS over NTRU lattices and Fast Fourier sampling | 103 bits | 897 bytes | 1,314.56 (32) bytes | 657.38 bytes | [179], [180] |
| FALCON-1024 | Lattice-based | Based on SIS over NTRU lattices and Fast Fourier sampling | 230 bits | 1,793 bytes | 2,546.62 (32) bytes | 1273.31 bytes | [179], [180] |
| GeMSS 128 | Multivariate-based | Built on HFEv- | 128 bits | 352.19 KB | 13.44 KB | 258 bits | [181], [182] |
| GeMSS 192 | Multivariate-based | Built on HFEv- | 192 bits | 1,237.964 KB | 34.07 KB | 411 bits | [181], [182] |
| GeMSS 256 | Multivariate-based | Built on HFEv- | 256 bits | 3,040.70 KB | 75.89 KB | 576 bits | [181], [182] |
| LUOV Level 1 (Chacha8) | Multivariate-based | Based on Unbalanced Oil and Vinegar (UOV) | 128 bits | 11.5 KB | 32 bytes | 239 bytes | [183], [184] |
| LUOV Level 3 (Chacha8) | Multivariate-based | Based on Unbalanced Oil and Vinegar (UOV) | 192 bits | 35.4 KB | 32 bytes | 337 bytes | [183], [184] |
| LUOV Level 5 (Chacha8) | Multivariate-based | Based on Unbalanced Oil and Vinegar (UOV) | 256 bits | 82.0 KB | 32 bytes | 440 bytes | [183], [184] |
| MQDSS 31-48 | Multivariate-based | Based on the 5-pass SSH (Sakumoto, Shirai, and Hiwatari) identification scheme | 64-128 bits | 46 bytes | 16 bytes | 20,854 bytes | [185], [186] |
| MQDSS 31-64 | Multivariate-based | Based on the 5-pass SSH (Sakumoto, Shirai, and Hiwatari) identification scheme | 96-192 bits | 64 bytes | 24 bytes | 43,728 bytes | [185], [186] |
| PICNIC2 L1-FS | Hash-based | Relies on Zero-Knowledge Proofs | 128 bits | 32 bytes | 16 bytes | 13,802 bytes (max) | [187] |
| PICNIC2 L3-FS | Hash-based | Relies on Zero-Knowledge Proofs | 192 bits | 48 bytes | 24 bytes | 29,750 bytes (max) | [187] |
| PICNIC2 L5-FS | Hash-based | Relies on Zero-Knowledge Proofs | 256 bits | 64 bytes | 32 bytes | 54,732 bytes (max) | [187] |
| qTESLA-p-I | Lattice-based | Based on RLWE | 128 bits | 14,880 bytes | 5,184 bytes | 2,592 bytes | [188] |
| qTESLA-p-III | Lattice-based | Based on RLWE | 192 bits | 38,432 bytes | 12,352 bytes | 5,664 bytes | [188] |
| Rainbow Ia | Multivariate-based | - | 128 bits | 149 bytes (58.1) KB | 93 KB | 512 bits | [189] |
| Rainbow IIIc | Multivariate-based | - | 192 bits | 710.6 (206.7) KB | 511.4 KB | 1,248 bits | [189] |
| Rainbow Vc | Multivariate-based | - | 256 bits | 1,705.5 (491.9) KB | 1,227.1 KB | 1,632 bits | [189] |
| SPHINCS+ -SHAKE256-128f-simple | Hash-based | Stateless signature scheme | 128 bits | 32 bytes | 64 bytes | 16,976 bytes | [190], [191] |
| SPHINCS+ -SHAKE256-192f-simple | Hash-based | Stateless signature scheme | 192 bits | 48 bytes | 96 bytes | 35,664 bytes | [190], [191] |
| SPHINCS+ -SHAKE256-256f-simple | Hash-based | Stateless signature scheme | 256 bits | 64 bytes | 128 bytes | 49,216 bytes | [190], [191] |
| SPHINCS+ -SHA-256-128f-simple | Hash-based | Stateless signature scheme | 128 bits | 32 bytes | 64 bytes | 16,976 bytes | [190], [191] |
| SPHINCS+ -SHA-256-192f-simple | Hash-based | Stateless signature scheme | 192 bits | 48 bytes | 96 bytes | 35,664 bytes | [190], [191] |
| SPHINCS+ -SHA-256-256f-simple | Hash-based | Stateless signature scheme | 256 bits | 64 bytes | 128 bytes | 49,216 bytes | [190], [191] |
| SPHINCS+-Haraka-128f-simple | Hash-based | Stateless signature scheme | 128 bits | 32 bytes | 64 bytes | 16,976 bytes | [190], [191] |
| SPHINCS+-Haraka-192f-simple | Hash-based | Stateless signature scheme | 192 bits | 48 bytes | 96 bytes | 35,664 bytes | [190], [191] |
| SPHINCS+-Haraka-256f-simple | Hash-based | Stateless signature scheme | 256 bits | 64 bytes | 128 bytes | 49,216 bytes | [190], [191] |

**TABLE 6.** Performance comparison of post-quantum encryption algorithms for blockchain nodes (part 1).

| References | Cryptosystem | Claimed Classical Security | Performance Evaluation Hardware | Key Generation (#Cycles) | Encapsulation (#Cycles) | Decapsulation (#Cycles) |
|---|---|---|---|---|---|---|
| [95] | BIKE-1 Level 1 | 128 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 730,025 | 689,193 | 2,901,203 |
| [95] | BIKE-1 Level 3 | 192 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 1,709,921 | 1,850,425 | 7,666,855 |
| [95] | BIKE-1 Level 5 | 256 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 2,986,647 | 3,023,816 | 17,483,906 |
| [95] | BIKE-2 Level 1 | 128 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 6,383,408 | 281,755 | 2,674,115 |
| [95] | BIKE-2 Level 3 | 192 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 22,205,901 | 710,970 | 7,114,241 |
| [95] | BIKE-2 Level 5 | 256 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 58,806,046 | 1,201,161 | 16,385,956 |
| [95] | BIKE-3 Level 1 | 128 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 433,258 | 575,237 | 3,437,956 |
| [95] | BIKE-3 Level 3 | 192 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 1,100,372 | 1,460,866 | 7,732,167 |
| [95] | BIKE-3 Level 5 | 256 bits | Intel Core i5-6260U @ 1.80 GHz, 32 GB of RAM | 2,300,332 | 3,257,675 | 18,047,493 |
| [102] | Classic McEliece (mceliece8192128) | 256 bits | Intel Xeon E3-1220 v3 @ 3.10 GHz | ∼4,675,000,000 | ∼296,000 | ∼458,000 |
| [131] | CRYSTALS Kyber-512 | 128 bits | Intel Core i7-4770K @ 3.5 GHz | 118,044 | 161,440 | 190,206 (≈ 279, 150) |
| [131] | CRYSTALS Kyber-512 90s | 128 bits | Intel Core i7-4770K @ 3.5 GHz | 232,368 | 285,336 | 313,452 (≈ 436, 088) |
| [131] | CRYSTALS Kyber-768 | 192 bits | Intel Core i7-4770K @ 3.5 GHz | 217,728 | 272,254 | 315,976 (≈ 469, 008) |
| [131] | CRYSTALS Kyber-768 90s | 192 bits | Intel Core i7-4770K @ 3.5 GHz | 451,018 | 514,088 | 556,972 (≈ 758, 934) |
| [131] | CRYSTALS Kyber-1024 | 256 bits | Intel Core i7-4770K @ 3.5 GHz | 331,418 | 396,928 | 451,096 (≈ 667, 596) |
| [131] | CRYSTALS Kyber-1024 90s | 256 bits | Intel Core i7-4770K @ 3.5 GHz | 735,382 | 810,398 | 860,272 (≈ 1, 148, 394) |
| [135] | FrodoKEM-640 AES | 128 bits | Intel Core i7-6700 @ 3.4 GHz | 1,384,000 | 1,858,000 | 1,749,000 |
| [135] | FrodoKEM-640 SHAKE | 128 bits | Intel Core i7-6700 @ 3.4 GHz | 7,626,000 | 8,362,000 | 8,248,000 |
| [135] | FrodoKEM-976 AES | 192 bits | Intel Core i7-6700 @ 3.4 GHz | 2,820,000 | 3,559,000 | 3,400,000 |
| [135] | FrodoKEM-976 SHAKE | 192 bits | Intel Core i7-6700 @ 3.4 GHz | 16,841,000 | 18,077,000 | 17,925,000 |
| [135] | FrodoKEM-1344 AES | 256 bits | Intel Core i7-6700 @ 3.4 GHz | 4,756,000 | 5,981,000 | 5,748,000 |
| [135] | FrodoKEM-1344 SHAKE | 256 bits | Intel Core i7-6700 @ 3.4 GHz | 30,301,000 | 32,611,000 | 32,387,000 |
| [106] | HQC Level 1 (hqc-128-1) | 128 bits | Intel Core i7- 7820X CPU @ 3.6 GHz, 16 GB of RAM | 110,000 | 190,000 | 310,000 |
| [106] | HQC Level 3 (hqc-192-1) | 192 bits | Intel Core i7- 7820X CPU @ 3.6 GHz , 16 GB of RAM | 190,000 | 330,000 | 510,000 |
| [106] | HQC Level 5 (hqc-256-1) | 256 bits | Intel Core i7- 7820X CPU @ 3.6 GHz , 16 GB of RAM | 270,000 | 470,000 | 690,000 |
| [138] | LAC-128 (CCA) | 128 bits | Intel Core-i7-4770S @ 3.10 GHz, 7.6 GB of RAM | 90,411 | 160,314 | 216,957 |
| [138] | LAC-192 (CCA) | 192 bits | Intel Core-i7-4770S @ 3.10 GHz, 7.6 GB of RAM | 281,324 | 421,439 | 647,030 |
| [138] | LAC-256 (CCA) | 256 bits | Intel Core-i7-4770S @ 3.10 GHz, 7.6 GB of RAM | 267,831 | 526,915 | 874,742 |

**TABLE 7.** Performance comparison of post-quantum encryption algorithms for blockchain nodes (part 2).

| References | Cryptosystem | Claimed Classical Security | Performance Evaluation Hardware | Key Generation (#Cycles) | Encapsulation (#Cycles) | Decapsulation (#Cycles) |
|---|---|---|---|---|---|---|
| [109] | LEDACrypt KEM Level 1 (for two circulant blocks) | 128 bits | Intel i5-6600 @ 3.6 GHz | - | - | - |
| [109] | LEDACrypt KEM Level 3 (for two circulant blocks) | 192 bits | Intel i5-6600 @ 3.6 GHz | - | - | - |
| [109] | LEDACrypt KEM Level 5 (for two circulant blocks) | 256 bits | Intel i5-6600 @ 3.6 GHz | - | - | - |
| [139] | NewHope-512 (CCA) | 128 bits | Intel Core i7-4770K @ 3.5 GHz | 117,128 | 180,648 | 206,244 |
| [139] | NewHope-1024 (CCA) | 256 bits | Intel Core i7-4770K @ 3.5 GHz | 244,944 | 377,092 | 437,056 |
| [142] | NTRUEncrypt (ntruhrss701) | 128/192 bits | Intel Core i7-4770K @ 3.5 GHz | 23,302,424 | 1,256,210 | 3,642,966 |
| [142] | NTRUEncrypt (ntruhps2048677) | 128/192 bits | Intel Core i7-4770K @ 3.5 GHz | 21,833,048 | 1,313,454 | 3,399,726 |
| [142] | NTRUEncrypt (ntruhps4096821) | 192/256 bits | Intel Core i7-4770K @ 3.5 GHz | 31,835,958 | 1,856,936 | 4,920,436 |
| [144] | NTRU Prime (sntrup4591761) | 153-368 bits | Intel Xeon E3-1275 v3 @ 3.5 GHz | 940,852 | 44,788 | 93,676 |
| [144] | NTRU Prime (ntrulpr4591761) | 155-364 bits | Intel Xeon E3-1275 v3 @ 3.5 GHz | 44,948 | 81,144 | 113,708 |
| [112] | NTS-KEM Level 1 | 128 bits | 16-core server with Intel Xeon E5-2667 v2 @ 3.3 GHz, 256 GB of RAM | 39,388,653 | 124,528 | 650,116 |
| [112] | NTS-KEM Level 3 | 192 bits | 16-core server with Intel Xeon E5-2667 v2 @ 3.3 GHz, 256 GB of RAM | 125,672,723 | 396,513 | 1,181,373 |
| [112] | NTS-KEM Level 5 | 256 bits | 16-core server with Intel Xeon E5-2667 v2 @ 3.3 GHz, 256 GB of RAM | 229,357,286 | 532,168 | 2,500,475 |
| [114] | ROLLO-II 128 | 128 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 9,620,000 | 1,520,000 | 4,960,000 |
| [114] | ROLLO-II 192 | 192 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 11,040,000 | 2,000,000 | 6,520,000 |
| [114] | ROLLO-II 256 | 256 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 11,410,000 | 2,390,000 | 7,940,000 |
| [147] | Round5 KEM IoT | 96-202 bits | MacBook Pro 15.1 with Intel Core i7 @ 2.6 GHz | 56,300 | 97,900 | 59,500 |
| [116] | RQC-I | 128 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 700,000 | 1,300,000 | 6,660,000 |
| [116] | RQC-II | 192 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 1,120,000 | 2,180,000 | 14,680,000 |
| [116] | RQC-III | 256 bits | Intel Core i7-7820X @ 3.6 GHz, 16 GB of RAM | 1,820,000 | 3,550,000 | 23,200,000 |
| [149] | SABER KEM (LightSABER) | 125-169 bits | Intel Core i5-7200U @ 2.50 GHz | 85,474 | 108,927 | 119,868 |
| [149] | SABER KEM | 203-244 bits | Intel Core i5-7200U @ 2.50 GHz | 163,333 | 196,705 | 215,733 |
| [149] | SABER KEM (FireSABER) | 283-338 bits | Intel Core i5-7200U @ 2.50 GHz | 259,504 | 308,277 | 341,654 |
| [129] | SIKE (SIKEp434) | 128 bits | Intel Core i7-6700 @ 3.4 GHz | 1,047,991,000 | 1,482,681,000 | 1,790,304,000 |
| [151] | Three Bears (BabyBear CCA) | 154-190 bits | Intel Core i3-6100U @ 2.3 GHz | 41,000 | 60,000 | 101,000 |
| [151] | Three Bears (MamaBear CCA) | 235-241 bits | Intel Core i3-6100U @ 2.3 GHz | 79,000 | 96,000 | 156,000 |
| [151] | Three Bears (PapaBear CCA) | 314-317 bits | Intel Core i3-6100U @ 2.3 GHz | 118,000 | 145,000 | 211,000 |

node (i.e., a node that only interacts with the blockchain) or a full blockchain node (i.e., a node that stores and updates periodically a copy of the blockchain and that is able to validate blockchain transactions).

For the sake of fairness, all the evaluation microprocessors indicated in Tables 6 and 7 are based on Intel x64 architecture and had Turbo Boost and Hyper-Threading features disabled. Since the version of the Intel microprocessor varies among the compared cryptosystems, the obtained results should be analyzed considering the differences in microprocessor performance. To carry out such an analysis in a fair way, Table 9 shows the most relevant characteristics of each microprocessor whose performance is referenced in this article. Thus, Table 9 compares the different clock frequencies, the main target platforms (i.e., laptop, server or desktop), the microprocessor typical energy consumption (indicated as Thermal Design Power (TDP)) and the estimated performance (making use of the Passmark CPU benchmarks [205]). In addition, also for the sake of fairness, Tables 6 to 8 compare the obtained performance results on the number of required execution cycles, which means that they have been normalized by taking the specific microprocessor clock frequency into account.

Specifically, Tables 6 and 7 indicate the number of cycles required by each microprocessor for key generation, encapsulation/encryption and decapsulation/decryption. The cycles required by LEDACrypt are not included because in their NIST second-round documentation it is only indicated the total algorithm execution time instead of the number of cycles. For CRYSTALS-Kyber, Table 6 indicates inside the parentheses the estimated number of cycles for the case when key generation is included in the decapsulation process (to avoid having to store expanded private keys).

In order to show in a clear and fast way to the reader which algorithms perform the better on the hardware platforms indicated in Tables 6 and 7 (i.e., without normalizing the performance differences related to the use of different clock frequencies), Figure 2 shows a bar chart of the average execution times of the algorithms listed in such Tables 6 and 7. As it can be observed, the lightest versions of schemes like NTRU Prime, Three Bears and SABER are really fast. However, it is important to note that, while Three Bears and SABER were evaluated in low-power microprocessors for laptops, the results obtained for NTRU Prime were obtained when ran on an Intel Xeon processor, which is a powerful microprocessor for servers.

In contrast, SIKE is the overall slowest scheme among the ones compared, while a cryptosystem like Classic McEliece suffers from a really slow key generation in spite of obtaining reduced decapsulation/decryption and encapsulation/encryption times. Nonetheless, it must be indicated that such slow schemes may be optimized for certain computational architectures and thus provide smallest execution times. In addition, post-quantum schemes, once publicly shared, evolve fast, so new implementations may be released in the future with the objective of reducing their computational complexity and, as a consequence, the required execution time.

### B. DIGITAL SIGNATURE SCHEME PERFORMANCE

Table 8 compares the performance of post-quantum digital signature algorithms that passed to the second round of the NIST call. The following considerations should be taken into account regarding the information shown in the Table:

- In the case of FALCON, the authors measured its performance in terms of spent time instead of cycles. This is related to the fact that the processor used by the researchers implemented dynamic frequency scaling based on load and temperature, which derived into measurements that vary up to 15% [180].
- For Rainbow, the values inside the parentheses indicate the performance of the key-compressed version, which, as it can be observed, requires much more computational effort than the regular version due to the involved decompression process.
- Most cryptosystems have been evaluated after optimizing them for AVX2, a 256-bit instruction set provided by Intel. The only exception is SPHINCS+ performance for the HARAKA version, whose optimized version was implemented to take advantage of the AES-NI instruction set.

Figure 3 shows through a bar chart the average execution times for the post-quantum cryptosystems listed in Table 8. Like in the case of the results obtained for the post-quantum encryption schemes, it is worth noting that the compared execution times were obtained in similar but not identical hardware platforms, so performance differences should be considered just as estimations. In addition, the following aspects should be taken into account regarding Figure 3:

- The obtained results are sorted by the sum of the three compared times, which is an estimation of the overall speed of each algorithm.
- FALCON is not included since there are no data for the three compared parameters.
- Besides post-quantum cryptosystems, the time required by two comparable pre-quantum schemes have been included as a reference: ECDSA (P-256) and RSA-3072. The execution times shown in the Figure for such implementations were obtained from [166], where the author used the libstrongswan library, which acted as an openssl wrapper for RSA and ECDSA, and whose measurements were performed on a laptop with an Intel Core i7-3610QM CPU at 2.30 GHz.
- The obtained results show that, as it was expected, the AVX2/AES-NI optimizations are clearly faster than the reference versions.
- The fastest schemes are DILITHIUM and the lightest versions of LUOV, qTESLA, MQDSS and Rainbow.

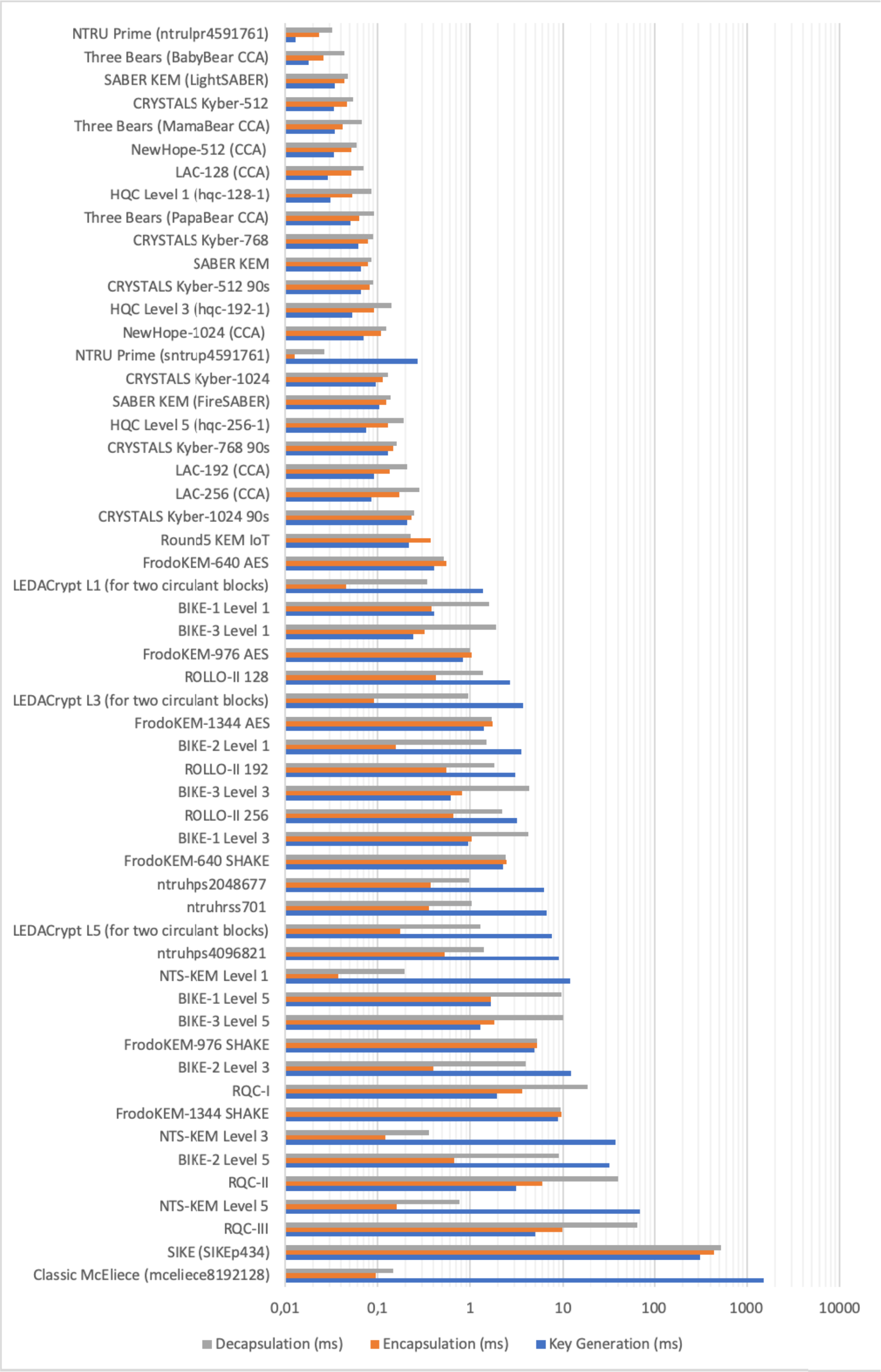

**FIGURE 2.** Comparison of the average execution times (in milliseconds) of NIST call second round public-key encryption schemes.

**TABLE 8.** Performance comparison of post-quantum digital signature algorithms that passed to the second round of the NIST call.

| | | Performance (Reference Implementation) | | | Performance (Optimized AVX2 Implementation) | | |
|---|---|---|---|---|---|---|---|
| Algorithm | Evaluation Platform | Key Generation (#Cycles) | Signing (#Cycles) | Verification (#Cycles) | Key Generation (#Cycles) | Signing (#Cycles) | Verification (#Cycles) |
| DILITHIUM 1280x1024 SHAKE (recommended) | Intel Core-i7 6600U (Skylake) CPU @ 2.6 GHz | 371,083 | 1,562,215 | 375,708 | 156,777 | 437,638 | 155,784 |
| DILITHIUM 1280x1024 AES (recommended) | Intel Core-i7 6600U (Skylake) CPU @ 2.6 GHz | - | - | - | 99,907 | 350,465 | 109,782 |
| FALCON-512 | Intel Core i7-6567U @ 3.3 GHz | 7.26 ms | - | - | - | - | - |
| FALCON-1024 | Intel Core i7-6567U @ 3.3 GHz | 21.63 ms | - | - | - | - | - |
| GeMSS 128 | Intel Core i7-6600U CPU @ 2.60 GHz (Skylake) | - | - | - | 36,800,000 | 529,000,000 | 84,600,000 |
| GeMSS 192 | Intel Core i7-6600U CPU @ 2.60 GHz (Skylake) | - | - | - | 167,000,000 | 1,720,000,000 | 233,000,000 |
| GeMSS 256 | Intel Core i7-6600U CPU @ 2.60 GHz (Skylake) | - | - | - | 508,000,000 | 2,830,000,000 | 550,000,000 |
| LUOV Level 1 (Chacha8) | Intel Core i5-8250U CPU @ 1.60 GHz | - | - | - | 1,100,000 | 224,000 | 49,000 |
| LUOV Level 3 (Chacha8) | Intel Core i5-8250U CPU @ 1.60 GHz | - | - | - | 4,600,000 | 643,000 | 152,000 |
| LUOV Level 5 (Chacha8) | Intel Core i5-8250U CPU @ 1.60 GHz | - | - | - | 9,700,000 | 1,100,000 | 331,000 |
| MQDSS 31-48 | Intel Core i7-4770K CPU @ 3.5 GHz | 1,192,984 | 26,630,590 | 19,840,136 | 1,074,644 | 3,816,106 | 2,551,270 |
| MQDSS 31-64 | Intel Core i7-4770K CPU @ 3.5 GHz | 2,767,384 | 85,268,712 | 62,306,098 | 2,491,050 | 9,047,148 | 6,132,948 |
| PICNIC2 L1-FS | Intel Core i7-4790 CPU @ 3.60 GHz | 149,749 | 3,066,663,719 | 1,857,340,295 | 21,026 | 229,947,918 | 100,546,772 |
| PICNIC2 L3-FS | Intel Core i7-4790 CPU @ 3.60 GHz | 362,481 | 10,190,171,124 | 5,537,696,230 | 20,160 | 657,944,759 | 223,785,326 |
| PICNIC2 L5-FS | Intel Core i7-4790 CPU @ 3.60 GHz | 691,790 | 25,488,037,138 | 12,943,455,830 | 35,716 | 1,346,724,260 | 387,637,876 |
| qTESLA-p-I | Intel Core i7-6700 (Skylake) @ 3.4 GHz | 2,316,200 | 2,324,900 | 671,400 | - | - | - |
| qTESLA-p-III | Intel Core i7-6700 (Skylake) @ 3.4 GHz | 13,726,600 | 6,284,600 | 1,830,400 | - | - | - |
| Rainbow Ia | Reference: Intel Xeon CPU E3-1225 v5 @ 3.30 GHz (Skylake),<br>AVX2: Intel Xeon CPU E3-1275 v5 @ 3.60 GHz (Skylake) | 35,000,000 (40,200,000) | 402,000 (20,200,000) | 155,000 (3,440,000) | 8,290,000 (9,280,000) | 67,700 (6,410,000) | 21,700 (3,370,000) |
| Rainbow IIIc | Reference: Intel Xeon CPU E3-1225 v5 @ 3.30 GHz (Skylake),<br>AVX2: Intel Xeon CPU E3-1275 v5 @ 3.60 GHz (Skylake) | 340,000,000 (402,000,000) | 1,700,000 (217,000,000) | 1,640,000 (19,400,000) | 94,800,000 (110,000,000) | 588,000 (61,800,000) | 114,000 (17,800,000) |
| Rainbow Vc | Reference: Intel Xeon CPU E3-1225 v5 @ 3.30 GHz (Skylake),<br>AVX2: Intel Xeon CPU E3-1275 v5 @ 3.60 GHz (Skylake) | 757,000,000 (879,000,000) | 3,640,000 (469,000,000) | 2,390,000 (45,400,000) | 126,000,000 (137,000,000) | 755,000 (87,200,000) | 197,000 (4,300,0000) |
| SPHINCS+ -SHAKE256-128f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 10,829,190 | 350,847,594 | 13,922,112 | 3,909,682 | 133,452,230 | 9,468,278 |
| SPHINCS+ -SHAKE256-192f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 15,192,014 | 645,965,282 | 21,943,196 | 6,303,298 | 171,354,532 | 14,758,202 |
| SPHINCS+ -SHAKE256-256f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 74,279,484 | 902,307,648 | 21,261,734 | 16,898,344 | 416,398,690 | 15,383,888 |
| SPHINCS+ -SHA-256-128f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 15,426,726 | 693,497,446 | 13,449,776 | 3,257,486 | 116,197,711 | 6,094,962 |
| SPHINCS+ -SHA-256-192f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 21,274,744 | 464,737,100 | 20,803,660 | 2,280,172 | 140,223,132 | 9,723,976 |
| SPHINCS+ -SHA-256-256f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 71,620,636 | 1,092,969,048 | 22,716,202 | 5,594,338 | 145,433,610 | 9,384,544 |
| SPHINCS+-Haraka-128f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 21,556,006 | 378,800,946 | 13,712,542 | 654,294 | 25,178,368 | 1,333,172 |
| SPHINCS+-Haraka-192f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 19,985,722 | 484,198,114 | 44,676,162 | 2,317,102 | 58,491,132 | 3,714,942 |
| SPHINCS+-Haraka-256f-simple | Intel Core i7-4770K CPU @ 3.5 GHz | 82,842,862 | 1,046,811,244 | 20,879,946 | 2,510,894 | 65,870,866 | 1,949,510 |

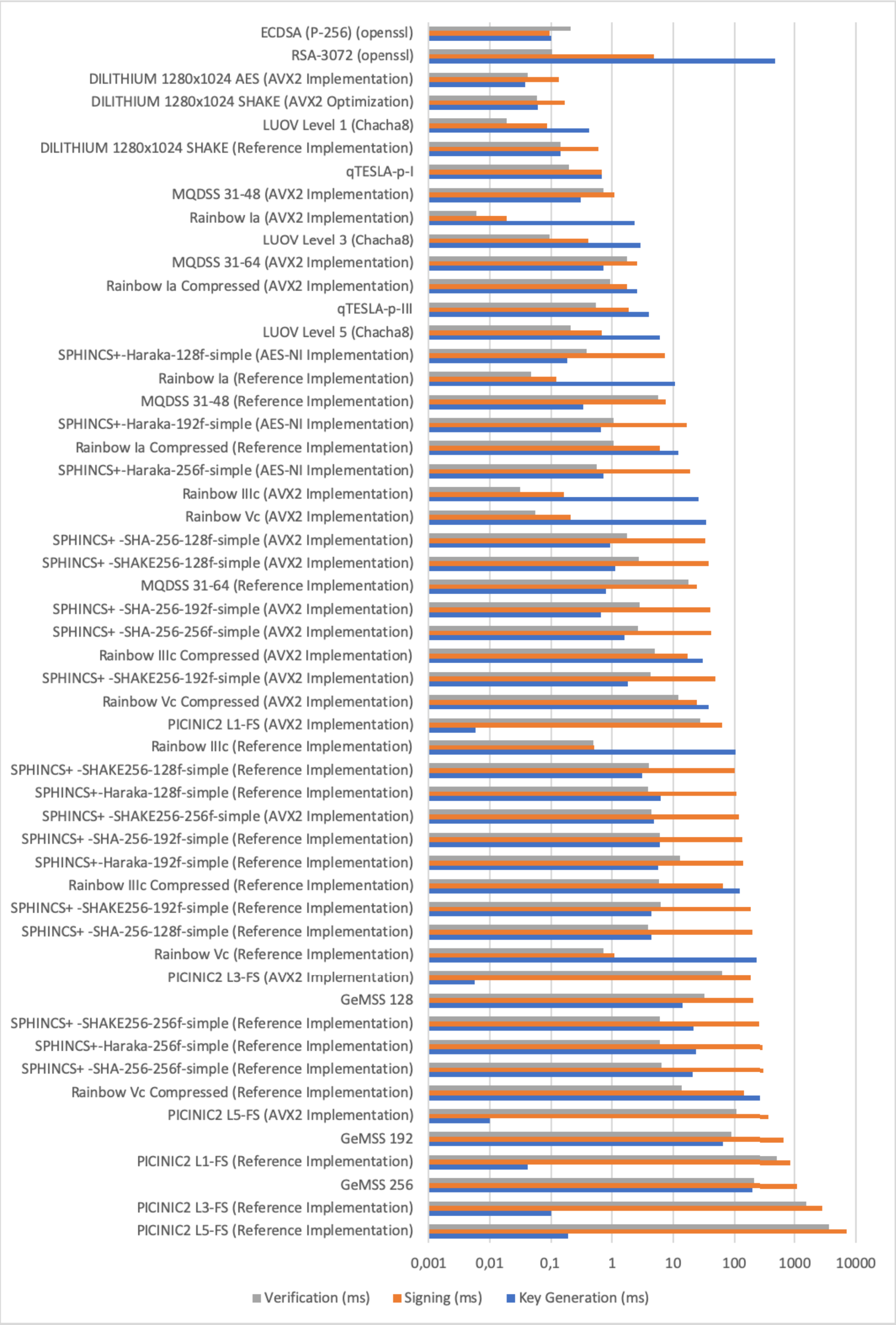

**FIGURE 3.** Comparison of the average execution times (in milliseconds) of NIST call second round digital signature schemes.

**TABLE 9.** Specifications of Intel microprocessors used for evaluating potential blockchain post-quantum algorithms.

| Microprocessor | Clock Frequency | Market Segment | Microarchitecture | Typical TDP | Release Data | Passmark Average Mark | Passmark Single Thread Rating |
|---|---|---|---|---|---|---|---|
| Intel Core i3-6100U | 2.3 GHz | Laptop | Skylake | 15 W | Q4 2015 | 3,603 | 1,302 |
| Intel Core i5-6260U | 1.8 GHz | Laptop | Skylake | 15 W | Q4 2015 | 4,362 | 1,593 |
| Intel Core i5-7200U | 2.5 GHz | Laptop | Kaby Lake | 15 W | Q4 2016 | 4,602 | 1,722 |
| Intel Core i7-6600U | 2.6 GHz | Laptop | Skylake | 15 W | Q3 2015 | 4,802 | 1,805 |
| Intel Core i7-6567U | 3.3 GHz | Laptop | Skylake | 28 W | Q3 2015 | 5,582 | 1,984 |
| Intel Xeon E3-1220 v3 | 3.1 GHz | Server | Haswell | 80 W | Q1 2011 | 7,163 | 1,945 |
| Intel Core i5-8250U | 1.6 GHz | Laptop | Kaby Lake | 15 W | Q3 2017 | 7,645 | 1,926 |
| Intel Core i5-6600 | 3.3 GHz | Desktop | Skylake | 65 W | Q2 2015 | 7,778 | 2,095 |
| Intel Xeon E3-1225 v5 | 3.3-3.7 GHz | Server | Skylake | 80 W | Q4 2015 | 7,829 | 1,969 |
| Intel Core-i7-4770S | 3.1 GHz | Desktop | Haswell | 65 W | Q2 2013 | 9,348 | 2,174 |
| Intel Xeon E3-1275 v3 | 3.5 GHz | Server | Haswell | 95 W | Q2 2013 | 9,915 | 2,214 |
| Intel Core i7-4790 | 3.6 GHz | Desktop | Haswell | 84 W | Q2 2014 | 9,989 | 2,283 |
| Intel Core i7-6700 | 3.4 GHz | Desktop | Skylake | 65 W | Q2 2015 | 10,003 | 2,154 |
| Intel Core i7-4770K | 3.5 GHz | Desktop | Haswell | 84 W | Q2 2013 | 10,075 | 2,250 |
| Intel Core i7-7820X | 3.6 GHz | Desktop | Skylake | 140 W | Q2 2017 | 18,489 | 2,401 |
| Intel Xeon E5-2667 v2 | 3.3 GHz | Server | Sandy Bridge | 130 W | Q1 2014 | 22,568 | 2,023 |

Overall, the AVX2 optimization of DILITHIUM seems to be, in terms of execution time, the most promising post-quantum digital signature scheme, since it obtains very similar results to ECDSA-256. Unfortunately, DILITHIUM key sizes are larger than the ones used by ECDSA-256, so researchers should focus on developing new approaches to reduce them.

- The slowest cryptosystems are the most secure versions of PICNIC2, GeMSS, Rainbow and SPHINCS. In the case of PICNIC2, its lack of speed is due to slow verification and signing processes. Regarding GeMSS, Rainbow and SPHINCS, their execution speed is impacted significantly by the amount of time devoted to key generation.

## VI. POST-QUANTUM BLOCKCHAIN PROPOSALS

Different authors have already proposed post-quantum blockchains or modifications of current blockchains to tackle the quantum threat [206]–[208]. For instance, in [209] it is proposed a framework aimed at sharing sensitive industrial data in public distributed networks. Such a framework is able to work with Inter-Planetary File System (IPFS) and Ethereum, and implements Diffie-Hellman Key Exchange on SIDH. Ethereum is also modified in [210], but with the multivariate-based cryptosystem Rainbow, whose performance is compared in the cited article with the current Ethereum version (based on ECDSA).

In the case of [211], the authors propose to improve Bitcoin (which uses the Koblitz curve secp256k1 and SHA-256 during the ECDSA signature process) with TESLA# [212], which makes use of BLAKE2 [213] and SHA-3 [214]. It is also worth mentioning the work in [10], where it is presented a blockchain-based transparent e-voting protocol that makes use of Niederreiter's code-based cryptosystem to proof the system against quantum attacks.

Other authors have suggested the implementation of quantum-safe blockchains [39], [215]. For example, in [215] the researchers present a quantum-safe transaction authentication scheme based on lattice-based cryptography and provide a standard transaction model to prevent quantum attacks. Similarly, in [39] a lattice-based signature scheme is proposed for developing a post-quantum blockchain that can be used to implement a cryptocurrency.

Commercial blockchains have also analyzed and addressed the impact of quantum computers. DLTs like IOTA's Tangle [40] claim to be more resistant than Bitcoin to quantum attacks that affect processes like nonce search [216]. In addition, IOTA has the advantage of being based on one-time hash-based signatures (Winternitz signatures) instead of on ECC. Furthermore, IOTA is expected to make use of ternary hardware (instead of traditional binary hardware) that will implement a new hash function called CURL-P, which is currently being audited. Finally, it is worth mentioning that there are other blockchains that have been devised to replace Bitcoin in the post-quantum era, like Quantum-Resistant Ledger [217], which replaces secp256k1 with XMSS.

## VII. MAIN CHALLENGES AND FUTURE RESEARCH TOPICS IN POST-QUANTUM BLOCKCHAIN

### A. QUANTUM COMPUTING FAST EVOLUTION

Quantum computing is currently a hot topic that has attracted a lot of attention from academia and industry. As a consequence, it is possible that new attacks will be developed against the post-quantum cryptosystems mentioned in this article, so researchers will have to pay attention to the quantum computing scene and its advances.

### B. TRANSITION FROM PRE-QUANTUM TO POST-QUANTUM BLOCKCHAIN

The transition from pre-quantum to post-quantum blockchains requires to think carefully the involved steps. For such a purpose, different researchers have devised methods. For instance, in [218] the authors propose a scheme to extend the validity of past blockchain blocks when the security of a hash function or of the digital signatures is compromised. However, the transition scheme may actually imply a hard-fork of the blockchain, but, to avoid it, a soft-fork mechanism may be implemented [219]. Another mechanism is proposed in [220], where it is presented a simple commit-delay-reveal protocol that enables blockchain users to move in a secure way funds from pre-quantum Bitcoin to a version that implements a post-quantum digital signature scheme.

### C. LARGE KEY AND SIGNATURE SIZES

In general, post-quantum cryptosystems require to use keys whose size is much larger than current public-key cryptosystems (usually between 128 and 4,096 bits).

In the case of digital signature cryptosystems, there are schemes like the ones based on supersingular isogenies that seem promising in terms of key size, but they produce large signatures and its performance is poor in comparison to other cryptosystems. For instance, as it was previously mentioned in Section IV-B.4, the scheme detailed in [195], for a 128-bit quantum security level, makes use of 2,688-bit public keys and 384-bit private keys, but it produces signatures of 120 KB, which is a problem for structures like blockchains that have to store massive amounts of such signatures. Similarly, hash-based schemes have a relatively small public/private key size, but their signatures often exceed 40 KB [60]. In contrast, some multivariate-based are able to provide short signatures, but the keys used for generating and verifying such signatures can occupy several kilobytes. Regarding lattice-based schemes, there are versions of DILITHIUM that are really fast, but whose key size is roughly 1,500 bytes and their signature length occupies 2,701 bytes.

With respect to post-quantum public-key encryption cryptosystems, certain optimized versions of schemes like Round5 seem promising, since their performance is good enough for most current blockchain node hardware, while keeping key size low (2,736 bits for the public key and only 128 bits for the private key). Nonetheless, more research is still needed in post-quantum schemes in order to provide a good trade-off between key sizes and security for blockchains.

### D. SLOW KEY GENERATION

In order to increase security, some post-quantum schemes limit the number of messages signed with the same key. As a consequence, it is necessary to generate new keys continuously, which involves dedicating computational resources and slowing down certain blockchain processes. Therefore, blockchain developers will have to determine how to adjust such key generation mechanisms to optimize the blockchain efficiency.

### E. COMPUTATIONAL AND ENERGY EFFICIENCY

As it can be concluded from the comparisons shown in Sections IV and V, some post-quantum schemes require a significant execution time, storage and computational resources. Such needs often derive into increased energy consumption, so future developers will have to look for novel approaches to optimize cryptosystems in order to maximize their computational and energy efficiency, and, as a consequence, the efficiency of the overall blockchain.

### F. STANDARDIZATION

As it was mentioned in Section III-C, multiple initiatives are currently analyzing post-quantum cryptosystems in order to standardize them. Since this is an ongoing effort, the researchers that look for guaranteeing blockchain compatibility will have to monitor the post-quantum scene and avoid the risk of using non-standard, discarded or broken schemes.

### G. BLOCKCHAIN HARDWARE UNSUITABILITY

Some computationally intensive post-quantum cryptosystems may not be suitable for certain hardware that is currently used for implementing blockchain nodes. Therefore, post-quantum schemes should provide a trade-off between security and computational complexity so that not to restrict the potential hardware that may interact with the blockchain.

### H. LARGE CIPHERTEXT OVERHEADS

Certain cryptosystems generate large overheads that may impact the performance of a blockchain. To tackle this issue, future post-quantum developers will have to minimize ciphertext overhead and consider potential compression techniques.

### I. QUANTUM BLOCKCHAIN

Besides the use of cryptosystems to transition from pre-quantum to post-quantum blockchain, several researchers proposed quantum-computing based blockchains [221]–[223]. For instance, in [224] and [225], the authors propose to migrate Bitcoin to quantum computers, while others described how to accelerate mining by modifying Grover's algorithm [226]. Moreover, some authors have already suggested using quantum cryptography to implement smart contracts [227]. Furthermore, more research is necessary on key establishment physics-based methods that are collectively known as Quantum-Key Distribution (QKD) [61].

## VIII. KEY FINDINGS

After the thorough literature analysis carried out in this article, the following conclusions can be drawn:

- After revising the literature, it was found no previous paper that provides a broad view on the importance and application of post-quantum blockchain as it is provided in this article. Although there are other reviews that addressed the impact of quantum computing on blockchain, they were essentially focused on giving generic recommendations for quantum-proofing blockchain [60] or on specific fields [228]. Moreover, it was found no other review that included the following main contributions together:

  - A detailed analysis on the impact of quantum attacks on blockchain public-key cryptosystems and hash functions.
  - A review on the most relevant post-quantum blockchain projects and standardization initiatives.
  - A detailed analysis on the characteristics of the main types of post-quantum encryption and digital signature schemes that can potentially be applied to blockchain.
  - Thorough comparisons on the performance of the most promising post-quantum blockchain cryptosystems.
  - A summary on the main post-quantum blockchain challenges and future trends that will provide a guide for future researchers and developers.

- Although there have been large projects on post-quantum security, it was not found any large academic initiative on the application of such a kind of security to blockchain.
- Nowadays, there are no post-quantum blockchain algorithms that provide, at the same time, small key size, short signature/hash sizes, fast execution, low computational complexity and low energy consumption. Such factors are especially critical for resource-constrained embedded devices like the ones used in the Internet of Things [228].
- Most of the post-quantum cryptosystems whose performance was compared in this article are currently being analyzed by the cryptographic community with the objective of selecting the most appropriate to be standardized through the NIST public call. Therefore, future developers should monitor the news and reports from NIST before selecting a specific post-quantum algorithm.
- It is not straightforward to choose a blockchain post-quantum cryptosystem. Future developers will have to take such a decision based on their blockchain node hardware, on the available resources (i.e., memory, speed), on the required blockchain node performance and on the necessary security level. For such a purpose, the tables provided throughout this article can be a very useful guide to estimate which may be the most promising candidates. Nonetheless, it has to be emphasized that the results provided in this article are related to specific hardware platforms, so performance will vary significantly when implemented and optimized for other hardware.
- Regarding the specific implementations compared in this article, the following general assessments can be stated on their application to blockchain:
  - Coded-based cryptosystems make use of large keys whose management and operation require a relevant amount of computational resources. More research is necessary on key compression techniques and on the use of certain types of codes and coding techniques.
  - Lattice-based cryptosystems also need to be enhanced in terms of key size, but it can be stated that they are currently some of the most promising candidates for implementing schemes for post-quantum blockchains. In fact, the comparisons performed in this article have shown that lattice-based algorithms Three Bears and SABER are really fast, even when executed on low-power microprocessors for laptops. In addition, a scheme like Round5 KEM IoT seems appropriate for being executed in most current blockchain node hardware and in many applications that do not require very high security. Furthermore, lattice-based digital signature cryptosystems have already been suggested and tested in different practical blockchain implementations [170], [173], [177] and, according to the comparisons shown in this article, certain optimized versions of DILITHIUM and qTESLA are among the fastest ones.
  - Multivariate-based public-key cryptosystems still need to be improved to increase decryption speed and to decrease key size. However, it should be noted that some multivariate-based signature algorithms optimized for the AVX2 instruction set (i.e., LUOV, MQDSS and Rainbow) are clearly faster than most of the compared digital signature cryptosystems.
  - Hybrid schemes like the ones tested by Google (CECPQ1 and CECPQ2) seem to be the next step prior to the actual implementation of pure post-quantum blockchains, but they require to make use of hardware able to handle at the same time two advanced security mechanisms and large payloads.
  - Super-singular elliptic-curve isogenie cryptosystems based on the Unruh transform seem promising, but still need to be optimized to decrease their signature size.
  - Hash-based digital signature cryptosystems have in general poor performance, but some researchers have suggested new faster algorithms that seem to be practical for blockchain [202], [203].
- It is necessary to study further how to enhance blockchain security by adding certain features that have been barely used in non-academic blockchain developments and validate their security in the post-quantum era. Some of such features are:

  - Aggregate signatures. They allow for generating a unique signature from several of them. This concept is attractive for blockchain, since it enables faster verification and reduces storage and bandwidth [229].
  - Ring signatures. They allow for specifying a set of possible signers without revealing who of them actually produced a signature [230]. Some researchers have already suggested quantum-resistant lattice-based schemes to secure ring signatures [231]–[233] and applied them in blockchain developments [234].
  - Identity-Base Encryption (IBE). It enables a sender and a receiver to communicate without exchanging public or private keys. For such a purpose, a trusted third-party is used as a middle-man between the sender and the receiver to generate private keys, which are sent to the receiver upon request. The scheme has been also generalized as Identity-based Broadcast Encryption (IBBE), which is able to manage multiple receivers instead of only one. IBE and IBBE are interesting for closed groups of users like private blockchains [235] and there are already implementations [236] (even for embedded systems [237]), but their need for a trusted third-party seems to be in conflict with the concept of public blockchain, whose existence is precisely justified by the lack of trust.
  - Secret sharing. It consists in dividing a piece of sensitive information into multiple parts that are distributed among diverse participants and which can be reconstructed by using a minimum number of parts [238]. For instance, in [8] it is introduced a private-key distribution method to help recover lost private keys that is based in secret sharing and in network protocols that guarantee the security of secret share transmission. Another example can be found in [239], where the authors use secret sharing to distribute transaction data securely among peers in a blockchain.
  - Homomorphic encryption. It enables third-party services to process a transaction without revealing unencrypted data to them [240], [241]. This kind of encryption has been already proposed to enhance the Bitcoin protocol [242], [243] and for blockchain-based IoT systems [244].
  - Zero-Knowledge Proofs. This kind of proofs validate a statement without revealing any secret related to it [245]. There is a specific type of these proofs called Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) that is aimed at reducing the complexity and the size of the proof [246]. However, it is necessary to design zk-SNARKs to make use of post-quantum cryptosystems or to take advantage of new post-quantum schemes like zk-STARKs [247]. In addition, it is possible to make use of SNAGS (Succinct Non-Interactive Arguments), whose quantum-resistivity is still being studied by the research community [248].
  - Secure Multi-Party Computation (SMPC). SMPC allows the parties involved in a blockchain to act together, but in a way that a single party does not have access to all the information, thus preventing secret data leaks. An example of the use of SMPC on a blockchain is Enigma [249], which first stores hashes on a blockchain and then the related data on an SMPC network that divides them into multiple pieces that are spread among different nodes.
- Although the analyses carried out in this article are focused on blockchain, since other DLTs work in a similar way, it is quite straightforward to apply to them the provided recommendations and extracted conclusions. Thus, such recommendations and conclusions could be extrapolated to DLTs based on Directed Acyclic Graphs (DAGs) (e.g., IOTA [40], Byteball [250]) or on Hashgraphs (e.g., Swirlds [251]). However, researchers still need to evaluate thoroughly DLT implementations that have already claimed to be better prepared for the post-quantum era than certain blockchains (e.g., IOTA, Quantum-Resistant Ledger [217]).

## IX. CONCLUSION

The recent progress on quantum computing has sparked interest in researchers and developers that work with DLTs like blockchain, where public-key cryptography and hash functions are essential. This article analyzed the impact of quantum-computing attacks (based on Grover's and Shor's algorithms) on blockchain and studied how to apply post-quantum cryptosystems to mitigate such attacks. For such a purpose, the most relevant post-quantum schemes were reviewed and their application to blockchain was analyzed, as well as their main challenges. In addition, extensive comparisons were provided on the characteristics and performance of the most promising post-quantum public-key encryption and digital-signature schemes. Thus, this article gives a broad view and insights on the quantum threat on blockchain, and provides useful guidelines for the researchers and developers of the next-generation of quantum-resistant blockchains.

## REFERENCES

[1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System.* Accessed: Nov. 2, 2019. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] M. Swan, *Blockchain: Blueprint for a New Economy*, 1st ed. Newton, MA, USA: O'Reilly Media, 2015.

[3] G. Zyskind and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, San Jose, CA, USA, May 2015, pp. 180–184.

[4] P. Giungato, R. Rana, A. Tarabella, and C. Tricase, "Current trends in sustainability of bitcoins and related blockchain technology," *Sustainability*, vol. 9, no. 12, p. 2214, Nov. 2017.

[5] C. Decker and R. Wattenhofer, "Information propagation in the bitcoin network," in *Proc. IEEE P2P*, Trento, Italy, Sep. 2013, pp. 1–10.

[6] T. M. Fernández-Caramés, I. Froiz-Míguez, O. Blanco-Novoa, and P. Fraga-Lamas, "Enabling the Internet of mobile crowdsourcing health things: A mobile fog computing, blockchain and IoT based continuous glucose monitoring system for diabetes mellitus research and care," *Sensors*, vol. 19, no. 15, p. 3319, Jul. 2019.

[7] W. S. Melo, A. Bessani, N. Neves, A. O. Santin, and L. F. R. C. Carmo, "Using blockchains to implement distributed measuring systems," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 5, pp. 1503–1514, May 2019.

[8] F. Xiong, R. Xiao, W. Ren, R. Zheng, and J. Jiang, "A key protection scheme based on secret sharing for blockchain-based construction supply chain system," *IEEE Access*, vol. 7, pp. 126773–126786, 2019.

[9] T. M. Fernández-Caramés, O. Blanco-Novoa, I. Froiz-Míguez, and P. Fraga-Lamas, "Towards an autonomous industry 4.0 warehouse: A UAV and blockchain-based system for inventory and traceability applications in big data-driven supply chain management," *Sensors*, vol. 19, no. 10, p. 2394, May 2019.

[10] S. Gao, D. Zheng, R. Guo, C. Jing, and C. Hu, "An anti-quantum E-voting protocol in blockchain with audit function," *IEEE Access*, vol. 7, pp. 115304–115316, 2019.

[11] T. M. Fernandez-Carames and P. Fraga-Lamas, "A review on the application of blockchain to the next generation of cybersecure industry 4.0 smart factories," *IEEE Access*, vol. 7, pp. 45201–45218, 2019.

[12] P. Fraga-Lamas and T. M. Fernandez-Carames, "A review on blockchain technologies for an advanced and cyber-resilient automotive industry," *IEEE Access*, vol. 7, pp. 17578–17598, 2019.

[13] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 26, no. 1, pp. 96–99, Jan. 1983.

[14] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics Comput.*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[15] V. S. Miller, "Use of elliptic curves in cryptography," in *Proc. Adv. Cryptol.*, in Lecture Notes in Computer Science, vol. 218, Aug. 1985, pp. 417–426.

[16] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[17] *Digital Signature Standard (DSS)*, Standard FIPS 186-2, NIST, Jan. 2000.

[18] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[19] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, Philadelphia, PA, USA, May 1996, pp. 212–219.

[20] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[21] M. Castro and B. Liskov, "Practical Byzantine fault tolerance and proactive recovery," *ACM Trans. Comput. Syst.*, vol. 20, no. 4, pp. 398–461, Nov. 2002.

[22] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[23] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.

[24] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, "An overview of blockchain technology: Architecture, consensus, and future trends," in *Proc. IEEE Int. Congr. Big Data (BigData Congr.)*, Honolulu, HI, USA, Jun. 2017, pp. 557–564.

[25] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and G. Das, "Everything you wanted to know about the blockchain: Its promise, components, processes, and problems," *IEEE Consum. Electron. Mag.*, vol. 7, no. 4, pp. 6–14, Jul. 2018.

[26] E. Karafiloski and A. Mishev, "Blockchain solutions for big data challenges: A literature review," in *Proc. IEEE Int. Conf. Smart Technol.*, Ohrid, Macedonia, Jul. 2017, pp. 763–768.

[27] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," in *Proc. IEEE Technol. Eng. Manage. Conf. (TEMSCON)*, Santa Clara, CA, USA, Jun. 2017, pp. 137–141.

[28] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—A systematic review," *PLoS ONE*, vol. 11, no. 10, Oct. 2016, Art. no. e0163477.

[29] M. Raikwar, D. Gligoroski, and K. Kralevska, "SoK of used cryptography in blockchain," Cryptol. ePrint Arch., Tech. Rep. 2019/735, Sep. 2019.

[30] L. Wang, X. Shen, J. Li, J. Shao, and Y. Yang, "Cryptographic primitives in blockchains," *J. Netw. Comput. Appl.*, vol. 127, pp. 43–58, Feb. 2019.

[31] E. Barker (NIST), "Recommendation for key management Part 1: General (revision 4)," U.S. Dept. Commerce, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Special Publication 800-57, Jan. 2016.

[32] *National Institute of Standards and Technology (NIST)*, document NIST-IR-8105 (draft), Report on Post-Quantum Cryptography, Apr. 2016.

[33] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *Quantum Inf. Comput.*, vol. 3, no. 4, pp. 317–344, 2003.

[34] *CNSS Advisory Memorandum Information Assurance 02-15: Use of Public Standards for the Secure Sharing of Information among National Security Systems*, NSS, Fort Meade, MD, USA, Jul. 2015.

[35] J. Cartwright, "NSA keys into quantum computing," *Phys. World*, vol. 27, no. 2, pp. 6–7, Feb. 2014.

[36] M. Mosca, "Cybersecurity in an era with quantum computers: Will we be ready?" *IEEE Secur. Privacy*, vol. 16, no. 5, pp. 38–41, Sep. 2018.

[37] N. Koblitz and A. Menezes, "A riddle wrapped in an enigma," *IEEE Security Privacy*, vol. 14, no. 6, pp. 34–42, Dec. 2016.

[38] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1510–1523, Oct. 1997.

[39] S. Krendelev and P. Sazonova, "Parametric hash function resistant to attack by quantum computer," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Poznan, Poland, Sep. 2018, pp. 387–390.

[40] *IOTA's*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.iota.org

[41] G. Brassard, P. Høyer, and A. Tapp, "Quantum cryptanalysis of hash and claw-free functions," in *Proc. Latin Amer. Symp. Theor. Informat.*, Valdivia, Chile, Mar. 2006, pp. 163–169.

[42] *PQCRYPTO Project*. Accessed: Nov. 2, 2019. [Online]. Available: https:pqcrypto.eu.org

[43] *SAFECrypto Project*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.safecrypto.eu

[44] *CryptoMathCREST Project*. Accessed: Nov. 2, 2019. [Online]. Available: https://cryptomath-crest.jp/english

[45] *PROMETHEUS*. Accessed: Nov. 2, 2019. [Online]. Available: http://prometheuscrypt.gforge.inria.fr

[46] *ETSI Technical Committee Cyber Working Group on Quantum-Safe Cryptography*. Accessed: Nov. 2, 2019. [Online]. Available: https://portal.etsi.org/TBSiteMap/CYBER/CYBERQSCToR.aspx

[47] *Crypto Forum Research Group*. Accessed: Nov. 2, 2019. [Online]. Available: https://irtf.org/cfrg

[48] *Quantum-Safe Hybrid (QSH) Ciphersuite for Transport Layer Security (TLS) Version 1.3*, document Internet-Draft draft-whyte-qsh-tls, IETF, Oct. 2016.

[49] *XMSS: Extended Merkle Signature Scheme*, document RFC 8391, IETF. Accessed: Nov. 2, 2019. [Online]. Available: https://datatracker.ietf.org/doc/rfc8391/

[50] *Leighton-Micali Hash-Based Signatures*, document RFC 8554, IETF, Accessed: Nov. 2, 2019. [Online]. Available: https://datatracker.ietf.org/doc/rfc8554/

[51] *ISO/IEC JTC 1/SC 27 (Working Group on IT Security Techniques)*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.iso.org/committee/45306.html

[52] *IEEE Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices*, IEEE Standard 1363.1-2008, Mar. 2009.

[53] *ANSI. X9.98-2010 (R2017): Lattice-Based Polynomial Public Key Establishment Algorithm for the Financial Services Industry*, Standard ANSI ASC X9, Feb. 2017.

[54] *ETSI/IQC 2018 Quantum Safe Workshop*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.etsi.org/news-events/events/1296-etsi-iqc-quantum-safe-workshop-2018

[55] *NIST's Workshop Cybersecurity a Post-Quantum World*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.nist.gov/news-events/events/2015/04/workshop-cybersecurity-post-quantum-world

[56] *NIST's Announcement of the First Post-Quantum Cryptography Standardization Conference*. Accessed: Nov. 2, 2019. [Online]. Available: https://csrc.nist.gov/events/2018/first-pqc-standardization-conference

[57] *PQCrypto Contributions*. Accessed: Nov. 2, 2019. [Online]. Available: https://cordis.europa.eu/project/rcn/194347/results

[58] *SAFECrypto Contributions*. Accessed: Nov. 2, 2019. [Online]. Available: https://cordis.europa.eu/project/rcn/194240/results/en

[59] *PROMETHEUS Publications*. Accessed: Nov. 2, 2019. [Online]. Available: http://prometheuscrypt.gforge.inria.fr/articles.html

[60] V. Gheorghiu, S. Gorbunov, M. Mosca, B. Munson, and W. Paper, "Quantum proofing the blockchain," Blockchain Res. Inst., Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep., Nov. 2017. [Online]. Available: https://evolutionq.com/assets/mosca_quantum-proofing-the-blockchain_blockchain-research-institute.pdf

[61] M. Lucamarini, A. Shields, R. Alléaume, C. Chunnilall, I. P. Degiovanni, M. Gramegna, A. Hasekioglu, B. Huttner, R. Kumar, A. Lord, and N. Lütkenhaus, "Implementation Security of Quantum Cryptography; Introduction, challenges, solutions," ETSI, Sophia Antipolis, France, White Paper 27, Jul. 2018.

[62] ETSI Technical Committee Cyber Working Group on Quantum-Safe Cryptography. *Quantum-Safe Cryptography (QSC); Limits to Quantum Computing Applied to Symmetric Key Sizes*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/006/01.01.01_60/gr_QSC006v010101p.pdf

[63] ETSI Technical Committee Cyber Working Group on Quantum-Safe Cryptography. *Quantum-Safe Cryptography; Quantum-Safe Threat Assessment*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/004/01.01.01_60/gr_QSC004v010101p.pdf

[64] *IETF Internet-Draft on the Transition From Classical to Post-Quantum Cryptography*. Accessed: Nov. 2, 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-hoffman-c2pq

[65] *ASC X9 IR01-2019—Quantum Computing Risks to the Financial Services Industry*, Standard ANSI ASC X9, Feb. 2019.

[66] *Announcement of the NIST's Call for Proposals for Public-Key Post-Quantum Cryptography Algorithms*. Accessed: Nov. 2, 2019. [Online]. Available: https://bit.ly/2hKONFb

[67] *NIST's Second Round Announcement Call for Proposals Post-Quantum cryptosystems*. Accessed: Nov. 2, 2019. [Online]. Available: https://csrc.nist.gov/news/2019/pqc-standardization-process-2nd-round-candidates

[68] *Bitcoin Post-Quantum*. Accessed: Nov. 2, 2019. [Online]. Available: https://bitcoinpq.org

[69] *Ethereum's Official Roadmap*. Accessed: Nov. 2, 2019. [Online]. Available: https://github.com/ethereum/wiki/wiki/Sharding-roadmap

[70] *Abelian*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.abelianfoundation.org

[71] *Corda's Supported Security Suites*. Accessed: Nov. 2, 2019. [Online]. Available: https://docs.corda.net/cipher-suites.html

[72] P. Fraga-Lamas, L. Ramos, V. Mondéjar-Guerra, and T. M. Fernández-Caramés, "A review on IoT deep learning UAV systems for autonomous obstacle detection and collision avoidance," *Remote Sens.*, vol. 11, no. 18, p. 2144, Sep. 2019.

[73] T. Fernández-Caramés and P. Fraga-Lamas, "Towards The Internet-of-smart-clothing: A review on IoT wearables and garments for creating intelligent connected E-textiles," *Electronics*, vol. 7, no. 12, p. 405, Dec. 2018.

[74] D. Hernández-Rojas, T. Fernández-Caramés, P. Fraga-Lamas, and C. Escudero, "Design and practical evaluation of a family of lightweight protocols for heterogeneous sensing through BLE beacons in IoT telemetry applications," *Sensors*, vol. 18, no. 2, p. 57, Dec. 2017.

[75] M. Suárez-Albela, T. M. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "A practical evaluation of a high-security energy-efficient gateway for IoT fog computing applications," *Sensors*, vol. 17, no. 9, p. 1978, Aug. 2017.

[76] P. Fraga-Lamas, M. Celaya-Echarri, P. Lopez-Iturri, L. Castedo, L. Azpilicueta, E. Aguirre, M. Suárez-Albela, F. Falcone, and T. M. Fernández-Caramés, "Design and experimental validation of a LoRaWAN fog computing based architecture for IoT enabled smart campus applications," *Sensors*, vol. 19, no. 15, p. 3287, Jul. 2019.

[77] O. Blanco-Novoa, T. Fernández-Caramés, P. Fraga-Lamas, and L. Castedo, "A cost-effective IoT system for monitoring indoor radon gas concentration," *Sensors*, vol. 18, no. 7, p. 2198, Jul. 2018.

[78] S. Ebrahimi, S. Bayat-Sarmadi, and H. Mosanaei-Boorani, "Post-quantum cryptoprocessors optimized for edge and resource-constrained devices in IoT," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5500–5507, Jun. 2019.

[79] T. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and L. Castedo, "Reverse engineering and security evaluation of commercial tags for RFID-based IoT applications," *Sensors*, vol. 17, no. 12, p. 28, Dec. 2016.

[80] P. Fraga-Lamas and T. M. Fernández-Caramés, "Reverse engineering the communications protocol of an RFID public transportation card," in *Proc. IEEE Int. Conf. RFID*, Phoenix, AZ, USA, May 2017, pp. 30–35.

[81] T. M. Fernández-Caramés, P. Fraga-Lamas, M. Suárez-Albela, and L. Castedo, "A methodology for evaluating security in commercial RFID systems," in *Radio Frequency Identification*, P. C. Crepaldi and T. C. Pimenta, Eds., 1st ed. Rijeka, Croatia: InTech, 2016.

[82] Z. Liu, K.-K.-R. Choo, and J. Grossschadl, "Securing edge devices in the post-quantum Internet of Things using lattice-based cryptography," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 158–162, Feb. 2018.

[83] M. Suárez-Albela, P. Fraga-Lamas, and T. Fernández-Caramés, "A practical evaluation on RSA and ECC-based cipher suites for IoT high-security energy-efficient fog and mist computing devices," *Sensors*, vol. 18, no. 11, p. 3868, Nov. 2018.

[84] M. Suárez-Albela, P. Fraga-Lamas, L. Castedo, and T. Fernández-Caramés, "Clock frequency impact on the performance of high-security cryptographic cipher suites for energy-efficient resource-constrained IoT devices," *Sensors*, vol. 19, no. 1, p. 15, Dec. 2018.

[85] D. J. Bernstein, J. Buchman, and E. Dahmen, *Post-Quantum Cryptography*. Berlin, Germany: Springer-Verlag, 2009.

[86] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," Deep Space Netw. Prog. Rep., Tech. Rep. DSN PR 42-44, Jan./Feb. 1978, pp. 114–116.

[87] E. Berlekamp, R. Mceliece, and H. Van Tilborg, "On the inherent intractability of certain coding problems (Corresp.)," *IEEE Trans. Inf. Theory*, vol. 24, no. 3, pp. 384–386, May 1978.

[88] W. Lee, J.-S. No, and Y.-S. Kim, "Punctured Reed–Muller code-based McEliece cryptosystems," *IET Commun.*, vol. 11, no. 10, pp. 1543–1548, Jul. 2017.

[89] *Quantum-Safe Cryptography (QSC); Quantum-Safe Algorithmic Framework*, document ETSI GR QSC 001 V1.1.1, ETSI, Sophia Antipolis, France, 2017. Jul. 2016. [Online]. Available: https://www.etsi.org/deliver/etsi_gr/QSC/001_099/001/01.01.01_60/gr_QSC001v010101p.pdf

[90] A. Petzoldt, S. Bulygin, and J. Buchmann, "Selecting parameters for the rainbow signature scheme," in *Proc. PQCrypto*, Darmstadt, Germany, May 2010, pp. 218–240.

[91] J. Ding, A. Petzoldt, and L.-C. Wang, "The cubic simple matrix encryption scheme," in *Proc. PQCrypto*, Waterloo, ON, Canada, Oct. 2014, pp. 76–87.

[92] J. Ding, "A new variant of the Matsumoto-Imai cryptosystem through perturbation," in *Proc. Int. Workshop Public Key Cryptogr.*, Singapore, Mar. 2004, pp. 305–318.

[93] J. Ding and D. Schmidt, "Cryptanalysis of HFEv and internal perturbation of HFE," in *Proc. Int. Workshop Public Key Cryptogr.*, Les Diablerets, Switzerland, Jan. 2005, pp. 288–301.

[94] J. Blömer and S. Naewe, "Sampling methods for shortest vectors, closest vectors and successive minima," in *Proc. Int. Colloq. Automata, Lang., Program.*, Wrocław, Poland, Jul. 2007, pp. 65–77.

[95] *BIKE Suite*. Accessed: Nov. 2, 2019. [Online]. Available: https://bikesuite.org

[96] *BIKE's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://bikesuite.org/files/BIKE.pdf

[97] P. S. L. M. Barreto, S. Gueron, T. Güneysu, R. Misoczki, E. Persichetti, N. Sendrier, and J.-P. Tillich, "CAKE: Code-based algorithm for key encapsulation," in *Proc. 16th IMA Int. Conf. Cryptogr. Coding*, Oxford, U.K., Dec. 2017, pp. 207–226.

[98] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. L. S. M. Barreto, "MDPC-McEliece: New McEliece variants from moderate density parity-check codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Istambul, Turkey, Jul. 2013, pp. 2069–2073.

[99] D. Micciancio, "Improving lattice based cryptosystems using the hermite normal form," in *Proc. Int. Cryptogr. Lattices Conf.*, Providence, RI, USA, Mar. 2001, pp. 126–145.

[100] P.-L. Cayrel, G. Hoffmann, and E. Persichetti, "Efficient implementation of a CCA2-secure variant of McEliece using generalized Srivastava codes," in *Proc. PKC*, Darmstadt, Germany, May 2012, pp. 138–155.

[101] J.-C. Deneuville, P. Gaborit, and G. Zémor, "Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory," in *Proc. PQCRYPTO*, Utrecht, The Netherlands, Jun. 2017, pp. 18–34.

[102] *Classical McEliece*. Accessed: Nov. 2, 2019. [Online]. Available: https://classic.mceliece.org

[103] *Classical McEliece Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://classic.mceliece.org/nist/mceliece-20171129.pdf

[104] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Proc. PQCRYPTO*, Cincinnati, OH, USA, Oct. 2008, pp. 31–46.

[105] A. Canteaut and N. Sendrier, "Cryptanalysis of the original McEliece cryptosystem," in *Proc. ASIACRYPT*, Beijing, China, Oct. 1998, pp. 187–199.

[106] *HQC*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-hqc.org

[107] *HQC Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-hqc.org/doc/hqc-specification_2019-04-10.pdf

[108] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zemor, "Efficient encryption from random quasi-cyclic codes," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3927–3943, May 2018.

[109] *LEADCrypt*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.ledacrypt.org/LEDAcrypt/

[110] *LEDACrypt's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.ledacrypt.org/documents/LEDAcrypt_spec_latest.pdf

[111] H. Jiang, Z. Zhang, and Z. Ma, "Tighter security proofs for generic key encapsulation mechanism in the quantum random oracle model," in *Proc. PQCrypto*, Chongqing, China, May 2019, pp. 227–248.

[112] *NTS-KEM* Accessed: Nov. 2, 2019. [Online]. Available: https://nts-kem.io

[113] *NTS-KEM's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://drive.google.com/file/d/1qPsXhK_oXJ88M1ec6pRbvvRKaCMQZfsc/view

[114] *ROLLO*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-rollo.org

[115] *ROLLO's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-rollo.org/doc/rollo-specification_2019-04-10.pdf

[116] *RQC*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-rqc.org

[117] *RQC's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://pqc-rqc.org/doc/rqc-specification_2019-04-10.pdf

[118] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Proc. 3rd Int. Symp. Algorithmic Number Theory*, Portland, OR, USA, Jun. 1998, pp. 267–288.

[119] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—A new hope," in *Proc. USENIX Secur. Symp.*, Aug. 2016, pp. 327–343.

[120] D. Stehlé and R. Steinfeld, "Making NTRU as secure as worst-case problems over ideal lattices," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Tallinn, Estonia, May 2011, pp. 27–47.

[121] G. S. Aujla, R. Chaudhary, K. Kaur, S. Garg, N. Kumar, and R. Ranjan, "SAFE: SDN-assisted framework for edge–cloud interplay in secure healthcare ecosystem," *IEEE Trans. Ind. Inf.*, vol. 15, no. 1, pp. 469–480, Jan. 2019.

[122] R. Lindner and C. Peikert, "Better key sizes (and attacks) for LWE-based encryption," in *Proc. Cryptographers' Track RSA Conf.*, San Francisco, CA, USA, Feb. 2011, pp. 319–339.

[123] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," in *Proc. EUROCRYPT*, Athens, Greece, May 2013, pp. 35–54.

[124] A. Rostovtsev and A. Stolbunov, "Public-key cryptosystem based on isogenies," Cryptol. ePrint Arch., Tech. Rep. 2006/145, 2006.

[125] A. Childs, D. Jao, and V. Soukharev, "Constructing elliptic curve isogenies in quantum subexponential time," *J. Math. Cryptol.*, vol. 8, no. 1, pp. 1–29, Jan. 2014.

[126] J.-F. Biasse, D. Jao, and A. Sankar, *A Quantum Algorithm for Computing Isogenies Between Supersingular Elliptic Curves* (Lecture Notes in Computer Science), vol. 8885. Cham, Switzerland: Springer, 2014, pp. 428–442.

[127] L. De Feo, D. Jao, and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," *J. Math. Cryptol.*, vol. 8, no. 3, pp. 209–247, Jun. 2014.

[128] C. Costello, P. Longa, and M. Naehrig, "Efficient algorithms for supersingular isogeny Diffie–Hellman," Cryptol. ePrint Arch., Tech. Rep. 2016/413, 2016.

[129] *SIKE*. Accessed: Nov. 2, 2019. [Online]. Available: https://sike.org

[130] *SIKE's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://sike.org/files/SIDH-spec.pdf

[131] *Kyber*. Accessed: Nov. 2, 2019. [Online]. Available: https://pq-crystals.org/kyber/index.shtml

[132] *Kyber Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://pq-crystals.org/kyber/data/kyber-specification-round2.pdf

[133] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, and D. Stehlée, "CRYSTALS–Kyber: A CCA-secure module-lattice-based KEM," in *Proc. IEEE Eur. Symp. Secur. Privacy*, London, U.K., Apr. 2018, pp. 353–367.

[134] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *Des., Codes Cryptogr.*, vol. 75, no. 3, pp. 565–599, Jun. 2015.

[135] *FrodoKEM*. Accessed: Nov. 2, 2019. [Online]. Available: https://frodokem.org

[136] *FrodoKEM's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://frodokem.org/files/FrodoKEM-specification-20190702.pdf

[137] J. W. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila, "Frodo: Take off the ring! Practical, quantum-secure key exchange from LWE," in *Proc. ACM CCS*, Vienna, Austria, Oct. 2016, pp. 1006–1018.

[138] *LAC's NIST Submission Package*. Accessed: Nov. 2, 2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/LAC.zip

[139] *New Hope*. Accessed: Nov. 2, 2019. [Online]. Available: https://newhopecrypto.org

[140] *New Hope's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://newhopecrypto.org/data/NewHope_2019_07_10.pdf

[141] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "NewHope without reconciliation," Cryptol. ePrint Arch., Tech. Rep. 2016/1157, 2016.

[142] *NTRU*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.onboardsecurity.com/nist-post-quantum-crypto-submission

[143] J. Hoffstein, J. Pipher, J. M. Schanck, J. H. Silverman, W. Whyte, and Z. Zhang, "Choosing parameters for NTRUEncrypt," in *Proc. RSA Conf.*, San Francisco, CA, USA, Feb. 2017, pp. 3–18.

[144] *NTRU Prime*. Accessed: Nov. 2, 2019. [Online]. Available: https://ntruprime.cr.yp.to

[145] *NTRU Prime's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://ntruprime.cr.yp.to/nist/ntruprime-20190330.pdf

[146] D. J. Bernstein, C. Chuengsatiansup, T. Lange, and C. van Vredendaal, "NTRU prime: Reducing attack surface at low cost," in *Proc. SAC*, Ottawa, ON, Canada, Aug. 2017.

[147] *Round5*. Accessed: Nov. 2, 2019. [Online]. Available: https://round5.org

[148] *Round5's Documentation for the Second Round of the NIST Call*. Accessed: Nov. 2, 2019. [Online]. Available: https://round5.org/Supporting_Documentation/Round5_Submission.pdf

[149] *SABER*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/saber/

[150] J.-P. D'Anvers, A. K. S. S. Roy, and F. Vercauteren, "Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM," in *Proc. Africacrypt*, Marrakesh, Morocco, May 2018, pp. 282–305.

[151] *Official Source Forge Repository Three Bears*. Accessed: Nov. 2, 2019. [Online]. Available: https://sourceforge.net/projects/threebears/

[152] *Google Blog Google's Experiments With a Hybrid Cryptosystem*. Accessed: Nov. 2, 2019. [Online]. Available: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

[153] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Problems Control Inf. Theory*, vol. 15, no. 2, pp. 159–166, 1986.

[154] N. T. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Gold Coast, QLD, Australia, Dec. 2001, pp. 157–174.

[155] M. Abdalla, J. H. An, M. Bellare, and C. Namprempre, "From identification to signatures via the fiat-Shamir transform: Minimizing assumptions for security and forward-security," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Amsterdam, The Netherlands, Apr./May 2002.

[156] S. M. E. Y. Alaoui, P. L. Cayrel, R. El Bansarkhani, and G. Hoffmann, "Code-based identification and signature schemes in software," in *Proc. Int. Conf. Availability, Rel. Secur.*, Regensburg, Germany, Sep. 2013, pp. 122–136.

[157] D. Unruh, "Non-interactive zero-knowledge proofs in the quantum random oracle model," in *Proc. EUROCRYPT*, Sofia, Bulgaria, Apr. 2015, pp. 755–784.

[158] D. Unruh, "Post-quantum security of fiat-Shamir," in *Proc. ASIACRYPT*, Hong Kong, Nov./Dec. 2017, pp. 65–95.

[159] J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms," in *Proc. EUROCRYPT*, Saragossa, Spain, May 1996, pp. 33–48.

[160] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, and J. Ding, "Design principles for HFEv-based multivariate signature schemes," in *Proc. Int. Conf. Adv. Cryptol.*, Auckland, New Zealand, Nov./Dec. 2015, pp. 311–334.

[161] N. T. Curtois, "On multivariate signature-only public key cryptosystems," in *Proc. IACR Cryptol. ePrint Arch.*, Apr. 2001, pp. 1–24.

[162] J.-M. Chen and B.-Y. Yang, "Tame transformation signatures with topsy-turvy hashes," in *Proc. IWAP*, Taipei, Taiwan, Oct. 2002, pp. 1–8.

[163] L.-C. Wang, Y.-H. Hu, F. Lai, C.-Y. Chou, and B.-Y. Yang, "Tractable rational map signature," in *Proc. Int. Workshop Public Key Cryptogr.*, Les Diablerets, Switzerland, Jan. 2005, pp. 23–36.

[164] J. Ding, B.-Y. Yang, C.-H. O. Chen, M.-S. Chen, and C. M. Cheng, "New differential-algebraic attacks and reparametrization of rainbow," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, New York, NY, USA, Jun. 2008, pp. 242–257.

[165] M. Ajtai, "Generating hard instances of lattice problems," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, Philadelphia, PA, USA, May 1996, pp. 99–108.

[166] M. Sjöberg, "Post-quantum algorithms for digital signing in public key infrastructures," M.S. thesis, School Comput. Sci. Commun., KTH Roy. Inst. Technol., Stockholm, Sweden, Jun. 2017. [Online]. Available: https://www.primekey.com/wp-content/uploads/2017/08/post-quantum-algorithms-forpki.pdf

[167] L. Ducas, A. Durmus, T. Lepoint, and V. Lyubashevsky, "Lattice signatures and bimodal Gaussians," in *Proc. Annu. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 2013, pp. 40–56.

[168] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, Gauss, and reload—A cache attack on the bliss lattice-based signature scheme," in *Proc. CHES*, Santa Barbara, CA, USA, Aug. 2016, pp. 323–345.

[169] P. Pessl, L. G. Bruinderink, and Y. Yarom, "To BLISS-B or not to be—Attacking strong Swan's implementation of post-quantum signatures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Dallas, TX, USA, Oct./Nov. 2017, pp. 1843–1855.

[170] Y.-L. Gao, X.-B. Chen, Y.-L. Chen, Y. Sun, X.-X. Niu, and Y.-X. Yang, "A secure cryptocurrency scheme based on post-quantum blockchain," *IEEE Access*, vol. 6, pp. 27205–27213, 2018.

[171] P. Zhang, H. Jiang, Z. Zheng, P. Hu, and Q. Xu, "A new post-quantum blind signature from lattice assumptions," *IEEE Access*, vol. 6, pp. 27251–27258, 2018.

[172] D. Chaum, "Blind signatures for untraceable payments," in *Proc. CRYPTO*, Aug. 1982, pp. 199–203.

[173] C. Li, G. Xu, Y. Chen, H. Ahmad, and J. Li, "A new anti-quantum proxy blind signature for blockchain-enabled Internet of Things," *Comput., Mater. Continua*, vol. 61, no. 2, pp. 711–726, Jan. 2019.

[174] C.-Y. Li, X.-B. Chen, Y.-L. Chen, Y.-Y. Hou, and J. Li, "A new lattice-based signature scheme in post-quantum blockchain network," *IEEE Access*, vol. 7, pp. 2026–2033, 2019.

[175] T. Güneysu, V. Lyubashevsky, and T. Pöppelmann, "Practical lattice-based cryptography: A signature scheme for embedded systems," in *Proc. CHES*, Leuven, Belgium, Sep. 2012, pp. 530–547.

[176] D. Cash, D. Hofheinz, E. Kiltz, and C. Peikert, "Bonsai trees, or how to delegate a lattice basis," in *Proc. EUROCRYPT*, French Rivera, France, May/Jun. 2010, pp. 523–552.

[177] H. An and K. Kim, "QChain: Quantum-resistant and decentralized PKI using blockchain," in *Proc. SCIS*, Niigata, Japan, Jan. 2018, pp. 1–8.

[178] *Dilithium*. Accessed: Nov. 2, 2019. [Online]. Available: https://pq-crystals.org/dilithium/index.shtml

[179] *FALCON*. Accessed: Nov. 2, 2019. [Online]. Available: https://falcon-sign.info

[180] *FALCON NIST Call Specification*. Accessed: Nov. 2, 2019. [Online]. Available: https://falcon-sign.info/falcon.pdf

[181] *GeMSS*. Accessed: Nov. 2, 2019. [Online]. Available: https://www-polsys.lip6.fr/Links/NIST/GeMSS.html

[182] *GeMSS NIST Call Specification*. Accessed: Nov. 2, 2019. [Online]. Available: https://www-polsys.lip6.fr/Links/NIST/GeMSS_specification.pdf

[183] *LUOV*. Accessed: Nov. 2, 2019. [Online]. Available: https://www.esat.kuleuven.be/cosic/pqcrypto/luov/

[184] *LUOV Official GitHub Repository*. Accessed: Nov. 2, 2019. [Online]. Available: https://github.com/WardBeullens/LUOV

[185] *MQDSS*. Accessed: Nov. 2, 2019. [Online]. Available: http://mqdss.org

[186] *MQDSS NIST Call Specification*. Accessed: Nov. 2, 2019. [Online]. Available: http://mqdss.org/files/MQDSS_Ver2.pdf

[187] *PICNIC*. Accessed: Nov. 2, 2019. [Online]. Available: https://microsoft.github.io/Picnic/

[188] *qTESLA*. Accessed: Nov. 2, 2019. [Online]. Available: https://qtesla.org

[189] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. ACNS*, New York, NY, USA, Jun. 2005, pp. 164–175.

[190] *SPHINCS+*. Accessed: Nov. 2, 2019. [Online]. Available: https://sphincs.org

[191] *SPHINCS+ NIST Call Specification*. Accessed: Nov. 2, 2019. [Online]. Available: https://sphincs.org/data/sphincs+-round2-specification.pdf

[192] R. Campbell, "Evaluation of post-quantum distributed ledger cryptography," *J. Brit. Blockchain Assoc.*, vol. 2, no. 1, p. 7679, Mar. 2019.

[193] X. Sun, H. Tian, and Y. Wang, "Toward quantum-resistant strong designated verifier signature from isogenies," in *Proc. 4th Int. Conf. Intell. Netw. Collaborative Syst.*, Bucharest, Romania, Sep. 2012, pp. 292–296.

[194] S. D. Galbraith, C. Petit, and J. Silva, "Identification protocols and signature schemes based on supersingular isogeny problems," in *Proc. ASIACRYPT*, Hong Kong, Dec. 2017, pp. 3–33.

[195] Y. Yoo, R. Azarderakhsh, A. Jalali, D. Jao, and V. Soukharev, "A post-quantum digital signature scheme based on supersingular isogenies," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, Sliema, Malta, Apr. 2017, pp. 163–181.

[196] C. Costello, D. Jao, P. Longa, M. Naehrig, J. Renes, and D. Urbanik, "Efficient compression of SIDH public keys," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, Paris, France, Apr. 2017, pp. 679–706.

[197] R. Azarderakhsh, D. Jao, K. Kalach, B. Koziel, and C. Leonardi, "Key compression for isogeny-based cryptosystems," in *Proc. 3rd ACM Int. Workshop ASIA Public-Key Cryptogr.*, Xi'an, China, May/Jun. 2016, pp. 1–10.

[198] L. Lamport, "Constructing digital signatures from a one-way function," SRI Int., Palo Alto, CA, USA, Tech. Rep. CSL-98, 1979.

[199] J. Buchmann, E. Dahmen, and A. Hülsing, "XMSS-a practical forward secure signature scheme based on minimal security assumptions," in *Proc. PQCrypto*, Taipei, Taiwan, Nov./Dec. 2011, pp. 117–129.

[200] D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-O'Hearn, "Sphincs: Practical stateless hash-based signatures," in *Proc. EUROCRYPT*, Sofia, Bulgaria, Apr. 2015, pp. 368–397.

[201] R. C. Merkle, "A certified digital signature," in *Proc. EUROCRYPT*, 1989, pp. 218–238.

[202] W. van der Linde, "Post-quantum blockchain using one-time signature chains," M.S. thesis, Dept. Comput. Sci., Radboud Univ., Nijmegen, The Netherlands, Aug. 2018.

[203] K. Chalkias, J. Brown, M. Hearn, T. Lillehagen, I. Nitto, and T. Schroeter, "Blockchained post-quantum signatures," in *Proc. IEEE iThings, GreenCom, CPSCom SmarData*, Halifax, NS, Canada, Jul./Aug. 2018, pp. 1196–1203.

[204] M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," in *Proc. 21st Annu. ACM Symp. Theory Comput.*, Seattle, WA, USA, May 1989, pp. 33–43.

[205] *Passmark CPU Benchmarks Main*. Accessed: Dec. 22, 2019. [Online]. Available: https://www.cpubenchmark.net

[206] M. Baldi, P. Santini, and G. Cancellieri, "Post-quantum cryptography based on codes: State of the art and open challenges," in *Proc. AEIT Int. Annu. Conf.*, Cagliari, Italy, Sep. 2017, pp. 1–6.

[207] Y. Qassim, M. E. Magaña, and A. Yavuz, "Post-quantum hybrid security mechanism for MIMO systems," in *Proc. Int. Conf. Comput., Netw. Commun.*, Santa Clara, CA, USA, Jan. 2017, pp. 684–689.

[208] V. Clupek, L. Malina, and V. Zeman, "Secure digital archiving in post-quantum era," in *Proc. 38th Int. Conf. Telecommun. Signal Process.*, Prague, Czech Republic, Jul. 2015, pp. 622–626.

[209] J. D. Preece and J. M. Easton, "Towards encrypting industrial data on public distributed networks," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Seattle, WA, USA, Dec. 2018, pp. 4540–4544.

[210] R. Shen, H. Xiang, X. Zhang, and B. Cai, "Application and implementation of multivariate public key cryptosystem in blockchain," in *Proc. CollaborateCom*, London, U.K., Aug. 2019, pp. 419–428.

[211] M. C. Semmouni, A. Nitaj, and M. Belkasmi, "Bitcoin security with post quantum cryptography," in *Proc. NETYS*, Marrakech, Morocco, Jun. 2019, pp. 281–288.

[212] P. S. Barreto, P. Longa, M. Naehrig, J. E. Ricardini, and G. Zanon, "Sharper ring-LWE signatures," Cryptol. ePrint Arch., Tech. Rep. 2016/1026, Nov. 2016.

[213] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, smaller, fast as MD5," in *Proc. ACNS*, Banff, AB, Canada, Jun. 2013, pp. 281–288.

[214] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," NIST, Gaithersburg, MD, USA, Tech. Rep. NIST FIPS-202, Aug. 2015.

[215] W. Yin, Q. Wen, W. Li, H. Zhang, and Z. Jin, "An anti-quantum transaction authentication approach in blockchain," *IEEE Access*, vol. 6, pp. 5393–5401, 2018.

[216] S. Popov, "The tangle, version 1.4.3," White Paper, Apr. 2018. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

[217] *The QRL. QRL—The Quantum Resistant Ledger*. Accessed: Nov. 2, 2019. [Online]. Available: https://theqrl.org/

[218] M. Sato and S. Matsuo, "Long-term public blockchain: Resilience against compromise of underlying cryptography," in *Proc. Int. Conf. Comput. Commun. Netw.*, Vancouver, BC, Canada, Jul./Aug. 2017.

[219] F. Chen, Z. Liu, Y. Long, Z. Liu, and N. Ding, "Secure scheme against compromised hash in proof-of-work blockchain," in *Proc. NSS*, Hong Kong, Aug. 2018, pp. 1–15.

[220] I. Stewart, D. Ilie, A. Zamyatin, S. Werner, M. F. Torshizi, and W. J. Knottenbelt, "Committing to quantum resistance: A slow defence for Bitcoin against a fast quantum computing attack," *Roy. Soc. Open Sci.*, vol. 5, no. 6, Jun. 2018, Art. no. 180410.

[221] K. Ikeda, "Security and privacy of blockchain and quantum computation," *Adv. Comput.*, vol. 111, pp. 199–228, May 2018.

[222] D. Rajan and M. Visser, "Quantum blockchain using entanglement in time," *Quantum Rep.*, vol. 1, no. 1, pp. 3–11, Apr. 2019.

[223] X. Sun, M. Sopek, Q. Wang, and P. Kulicki, "Towards quantum-secured permissioned blockchain: Signature, consensus, and logic," *Entropy*, vol. 21, no. 9, p. 887, Sep. 2019.

[224] J. Jogenfors, "Quantum bitcoin: An anonymous, distributed, and secure currency secured by the no-cloning theorem of quantum mechanics," in *Proc. IEEE ICBC*, Seoul, South Korea, May 2019, pp. 245–252.

[225] K. Ikeda, "qBitcoin: A peer-to-peer quantum cash system," in *Proc. Adv. Intell. Syst. Comput.*, Nov. 2018, pp. 763–771.

[226] F. M. Ablayev, D. A. Bulychkov, D. A. Sapaev, and A. V. Vasiliev, "Quantum-assisted blockchain," *Lobachevskii J. Math.*, vol. 39, no. 7, pp. 957–960, Sep. 2018.

[227] A. Coladangelo, "Smart contracts meet quantum cryptography," Jul. 2019, *arXiv:1902.05214*. [Online]. Available: https://arxiv.org/abs/1902.05214

[228] T. M. Fernández-Caramés, "From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things," *IEEE Internet Things J.*, to be published.

[229] Y. Zhao, "Aggregation of Gamma-signatures and applications to bitcoin," Cryptol. ePrint Arch., Tech. Rep. 2018/414, Dec. 2018.

[230] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Advances in Cryptology—ASIACRYPT* (Lecture Notes in Computer Science), vol. 2248, C. Boyd, Ed. Berlin, Germany: Springer-Verlag, 2001, pp. 1–14. [Online]. Available: https://link.springer.com/chapter/10.1007/3-540-45682-1_32

[231] Z. Liu, K. Nguyen, G. Yang, and H. Wang, "A lattice-based linkable ring signature supporting stealth addresses," in *Proc. ESORISCS*, Luxembourg City, Luxembourg, Sep. 2019, pp. 726–746.

[232] H. Zhang, F. Zhang, H. Tian, and M. H. Au, "Anonymous post-quantum cryptocash," in *Proc. FC*, Nieuwpoort, Curaçao, Feb. 2018, pp. 461–479.

[233] W. A. A. Torres, R. Steinfeld, A. Sakzad, J. K. Liu, V. Kuchta, N. Bhattacharjee, M. H. Au, and J. Cheng, "Post-Quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1.0)," in *Proc. ACISP*, Wollongong, NSW, Australia, Jul. 2018, pp. 558–576.

[234] *CryptoNote's*. Accessed: Nov. 2, 2019. [Online]. Available: https://cryptonote.org

[235] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 1984, pp. 47–53.

[236] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. CRYPTO*, Santa Barbara, CA, USA, Aug. 2001, pp. 213–229.

[237] T. Güneysu and T. Oder, "Towards lightweight Identity-Based Encryption for the post-quantum-secure Internet of Things," in *Proc. 18th Int. Symp. Qual. Electron. Design*, Santa Clara, CA, USA, Mar. 2017, pp. 319–324.

[238] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[239] R. K. Raman and L. R. Varshney, "Distributed storage meets secret sharing on the blockchain," in *Proc. ITA*, San Diego, CA, USA, Feb. 2018, pp. 1–6.

[240] C. Moore, M. O'Neill, E. O'Sullivan, Y. Doröz, and B. Sunar, "Practical homomorphic encryption: A survey," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Melbourne, VIC, Australia, Jun. 2014, pp. 2792–2795.

[241] H. Hayouni and M. Hamdi, "Secure data aggregation with homomorphic primitives in wireless sensor networks: A critical survey and open research issues," in *Proc. IEEE 13th Int. Conf. Netw., Sens., Control (ICNSC)*, Mexico City, Mexico, Apr. 2016, pp. 28–30.

[242] B. F. França. (Apr. 2015). *Homomorphic Mini-Blockchain Scheme*. Accessed: Nov. 2, 2019. [Online]. Available: http://cryptonite.info/files/HMBC.pdf

[243] D. Lukianov. (Dec. 2015). *Compact Confidential Transactions for Bitcoin*. Accessed: Nov. 2, 2019. [Online]. Available: http://voxelsoft.com/dev/cct.pdf

[244] L. Zhou, L. Wang, Y. Sun, and P. Lv, "BeeKeeper: A blockchain-based IoT system with secure storage and homomorphic computation," *IEEE Access*, vol. 6, pp. 43472–43488, 2018.

[245] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *J. Cryptol.*, vol. 7, no. 1, pp. 1–32, Dec. 1994.

[246] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von Neumann architecture," in *Proc. USENIX Secur. Symp.*, San Diego, CA, USA, Aug. 2014, pp. 781–796.

[247] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," IACR Cryptol. ePrint Arch., White Paper 1845, Mar. 2018. [Online]. Available: https://eprint.iacr.org/2018/046.pdf

[248] A. Chiesa, P. Manohar, and N. Spooner, "Succinct arguments in the quantum random oracle model," Cryptol. ePrint Arch., Tech. Rep. 2019/834, Oct. 2019.

[249] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," White Paper, Jun. 2015. [Online]. Available: https://arxiv.org/abs/1506.03471

[250] *Byteball's*. Accessed: Nov. 2, 2019. [Online]. Available: https://obyte.org

[251] L. Baird, "Overview of swirlds hashgraph," White Paper, May 2016. [Online]. Available: https://www.swirlds.com/downloads/Overview-of-Swirlds-Hashgraph.pdf

**TIAGO M. FERNÁNDEZ-CARAMÉS** (Senior Member, IEEE) received the M.Sc. degree and the Ph.D. degree in computer science from the University of A Coruña (UDC), Spain, in 2005 and 2011, respectively. He has been an Assistant Professor of electronic technology with UDC, Spain, since 2016. Since 2005, he has been with the Department of Computer Engineering, UDC, through different predoctoral scholarships from 2005 to 2009 and as an Interim Professor from 2007 to 2016. His current research interests include IoT/IIoT systems, RFID, wireless sensor networks, augmented reality, embedded systems, and blockchain, as well as the different technologies involved in the Industry 4.0 paradigm. In such fields, he has contributed to 40 papers for conferences, 35 articles for JCR-indexed journals, and two book chapters. Due to his expertise in the previously mentioned fields, he has acted as Peer Reviewer and Guest Editor for different top-ranked journals, and as Project Reviewer for national research bodies from Austria, Croatia, and Argentina.

**PAULA FRAGA-LAMAS** (Member, IEEE) received the M.Sc. degree in computer engineering from the University of A Coruña (UDC), in 2009, and the M.Sc. and Ph.D. degrees in the joint mobile network information and communication technologies program from five Spanish universities, University of the Basque Country, University of Cantabria, University of Zaragoza, University of Oviedo, and University of A Coruña, in 2011 and 2017, respectively. She holds an MBA and postgraduate studies in business innovation management (Jean Monnet Chair in European Industrial Economics, UDC), and Corporate Social Responsibility (CSR) and social innovation (Inditex-UDC Chair of Sustainability). Since 2009, she has been with the Group of Electronic Technology and Communications (GTEC), Department of Computer Engineering (UDC). She has over 60 contributions in indexed international journals, conferences, and book chapters, and holds four patents. She has also been participating in over 30 research projects funded by the regional and national government as well as Research and Development contracts with private companies. She is actively involved in many professional and editorial activities, acting as reviewer of more than 35 international journals, advisory board member, topic/guest editor of top-ranked journals and TPC member of international conferences. Her current research interests include mission-critical scenarios, Industry 4.0, Internet of Things (IoT), Cyber-Physical Systems (CPS), Augmented Reality (AR), fog and edge computing, blockchain and Distributed Ledger Technology (DLT), and cybersecurity.