Bitcoin Post-Quantum

Noah Anhao
noahanhao@bitcoinpq.org

1. Introduction

The security of the decentralized digital currency, Bitcoin, is based on the Elliptic Curve Digital Signature Algorithm (ECDSA) and the Proof of Work (PoW) algorithm. ECDSA allows authorization of the payer. PoW prevents a payer from paying with the same money more than once.

Bitcoins are transferred between users through transactions from one address to another, the address being is a hash of a recipient's public key or script. The public key of the sender is revealed when the transaction is distributed over the Bitcoin network. The owning of bitcoins amounts to the ownership of the private key of the corresponding address. The digital signature, which is put on the transaction, is the proof of ownership.

The cryptographic strength of the ECDSA is based on the complexity of the discrete logarithm problem in the group of elliptic curve points (ECDLP), which is solved in exponential time; i.e., it is difficult for a classical computer. Shor's quantum algorithm for calculation of the discrete logarithms in finite fields allows calculation of the private ECDSA key through the public key in polynomial time; this is significantly easier. This threatens addresses with a non-zero balance that were previously used for spending and the transactions that have not yet been included in a block. When the transaction is distributed over the network, the public key is disclosed. This gives a window for attack before the transaction is included in the next block to calculate the private key using a quantum computer and forge a new transaction with a valid digital signature.

Fortunately, there are problems that are suitable for digital signature schemes and are sufficiently complex for both classical and quantum computers. Searching for the preimage of the hash applies to these problems. Grover's quantum algorithm for searching an unsorted database allows solving the problem related to the hash preimage searching in time of order the square root of the classical time, which is a great acceleration. However, the complexity of the problem remains exponential. Thus, the necessary security level is achieved by using a hash function of the appropriate length. This allows the hash-based cryptographic systems to be considered as quantum-safe. According to the PQCRYPTO recommendations (The European Consortium of Universities and Companies for Post-Quantum Cryptography Issues), the extended Merkle signature scheme (XMSS) should be used as a quantum-safe digital signature because it combines high security, acceptable key generation time and the size of the signature, in contrast with the other post-quantum digital signature algorithms offered by the scientific community.

Bitcoin Post-Quantum (BPQ) is an experimental branch of Bitcoin's main blockchain using quantum-safe digital signatures. In the future, the experience of BPQ may be useful for the introduction of quantum-safe cryptography to the main branch of Bitcoin. Furthermore,

BPQ serves the current needs for a back-up blockchain in the event of a sudden leap in technological development that could compromise the safety of the most generic cryptocurrencies.

There is mounting evidence that quantum computers will become powerful enough to crack popular cryptographic schemes in the foreseeable future, even though it is impossible to accurately predict when it will happen.

Outline of facts demonstrating the timeliness of our development:

- In 2015, the US National Security Agency (NSA) announced the plans for the transition to post-quantum cryptographic algorithms: "Unfortunately, the growth of elliptic curve use has bumped up against the fact of continued progress in the research on quantum computing, which has made it clear that elliptic curve cryptography is not the long-term solution many once hoped it would be." [1];

- In 2016, IBM provided [2] the first cloud-based quantum computer, IBM Q, with five qubits, accessible to anyone who wants to practice quantum programming;

- In 2016, Intel engineers announced [3] the work on a quantum processor with millions of qubits;

- In 2016, Google Chrome developers implemented [4] the post-quantum key exchange algorithm New Hope and in 2017 Google predicted [5] the commercialization of quantum technology within the next five years;

- In April 2018, the developers implemented the post-quantum algorithm of the XMSS digital signature for the OpenSSH 7.7 update [6];

- In June 2018, Microsoft added [7] post-quantum key exchange algorithms and signatures to their OpenVPN fork.

2. Quantum Computing

At a deeper level of reality, each particle of matter is not just a dot in the state space, but it is a "cloud" of its possible states. For example, when a single photon passes through a screen with two slits of the width of about the photon's own wavelength, it passes through both slits simultaneously. Then the different variants of the photon, which are moving along different trajectories, interact among themselves, as evidenced by the interference pattern on the projection screen, that appears from multiple repetition of the experiment [8]. The particle is in a *superposition* of its possible states.

A wave function set in the state space fully describes the quantum object. The instrumental meaning of the wave function is that the square of its modulus is equal to the probability of detecting a particle in a certain state after measurement. The measuring of a quantum state leads to the splitting of possible ways of the evolution of the quantum system as separate streams of reality in which the measured value, the state of the device, and of the observer itself, are specified. Therefore, a measurement always gives a concrete value with some probability.

Since the state can be measured, it is possible to select a suitable quantum object and its characteristic for the measurement, agree on bit encoding 0 and 1 and use the quantum state as the information carrier. The quantum object can be put into the state of superposition in a predictable manner, which simultaneously encodes 0 in one part of the reality variants and 1 in the other, resulting in the abstract mathematical concept of a qubit (q-bit, quantum bit) without any binding to a particular physical implementation. The algorithms that are executed on quantum computers are sequential transformations of qubit states. Algebraically it appears that the unitary matrices of quantum gates are multiplied by the state vectors of qubits. In all variants of reality, therefore, a certain amount of useful computation with its input data takes place and the combination of intermediate computational results is carried out using interference, as shown in the Figure 1. The last operation of a quantum algorithm is usually the measuring of qubit states.

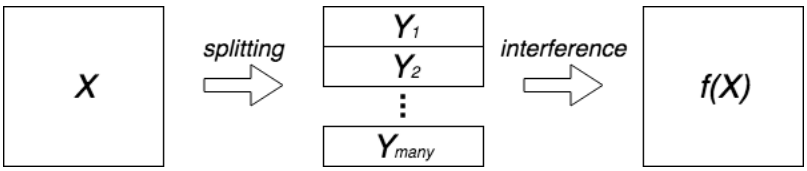

Figure 1. A typical quantum computation. $Y_1 \ldots Y_{many}$ are intermediate results that depend on the input X. All of them are needed to compute the output $f(X)$ efficiently. [9]

As Deutsch showed [10], the universal quantum computer can simulate various physical systems, real and theoretical, which are beyond the scope of the universal Turing machine. The universal quantum computer can then make it possible to simplify the solution of certain types of problems that cannot be proved quickly in the framework of classical computation. This simplification relates to the problems of cryptology in particular.

In 1994, Shor [11] proposed quantum algorithms for discrete logarithm in the group of elliptic curve points and in the factorization of numbers. In 2001, IBM [12] demonstrated the efficiency of Shor's algorithm, decomposing the number 15 into multipliers 3 and 5 on a 7-qubit quantum computer. According to Proos' and Zalka's assessment [13], the restoration of an ECDSA 256-bits private key requires approximately 1500 qubits and $6*10^9$ ($\sim 2^{32}$) operations. According to Microsoft experts [14], 2330 qubits and $1.26*10^{11}$ ($\sim 2^{36}$) operations are required.

The secp256k1 elliptic curve is used for signing Bitcoin transactions and provides a 128-bit classical security level. Soon Bitcoin will use the Schnorr signature scheme, which is also based on the complexity of the discrete logarithm problem; also not quantum safe.

Bitcoin's blockchain analysis demonstrated [15] that as of June 2018, about 19% of the addresses, where 36% of the bitcoins are stored, hold public keys open, making these funds vulnerable to quantum attack in the first place. The remaining bitcoins are located at one-time addresses and, therefore, are less at risk of attack because their public keys are disclosed only at the time of the spending, meaning the attacker has little time to attempt a double spend attack before the transaction is included in the block. Until then, the public keys are hashed under the 160-bit hash of RIPEMD160(SHA2-256(pubkey)) for P2(W)PKH and P2SH addresses and under the 256-bit hash of SHA2-256(pubkey) for P2WSH addresses.

Grover's quantum algorithm [16] for searching unsorted databases allows one to solve the problem related to the hash preimage search in time of order the square root of the classical time, which is great acceleration. However, the complexity of the problem remains exponential. Therefore, the post-quantum security level of a 160-bit hash function RIPEMD160 is 80 bits, so the bitcoins stored at one-time addresses are relatively safe compared with those located at addresses with disclosed public keys. To obtain the desired post-quantum security level (at least 128 bits), a hash function of the appropriate length must be selected. The 256-bit hashes used in P2WSH addresses provide a sufficient 128-bit post-quantum security level until the corresponding public keys are disclosed. It is worth mentioning that in the new P2WSH addresses a 256-bit hash function was implemented instead of the 160-bit hash for another purpose — to provide 128-bit collision resistance against birthday attacks when using multi-signatures.

3. Post-Quantum Cryptography

A post-quantum cryptosystem is one that is secure against a quantum attack. The known algorithms of post-quantum cryptography with a public key are based on various approaches [17], including hash functions, supersingular elliptic curves isogenies, error-correcting codes, lattices and multivariate quadratic equations.

The security level of a hash-based digital signature is reduced to the first and second preimage resistance of a hash function. On a classic computer the recovery of n bits hash is confined to a brute force search (complexity $O(2^n)$), whereas on a quantum computer it is confined to the Grover's search (complexity $O(2^{n/2})$). Therefore, we can choose a suitable hash function that provides the necessary classical and post-quantum security level. Other

approaches to post-quantum cryptography are based on mathematical problems that are considered quantum-resistant, but perhaps can be solved on a classical computer in consequence of a breakthrough in the development of mathematics.

According to the PQCRYPTO recommendations [18], the digital signature scheme XMSS [19] with parameters from RFC 8391 [20] is used to achieve 128-bit post-quantum cryptography in Bitcoin Post-Quantum. The drawback of the XMSS scheme is that it can generate a limited number of signatures, which depends on the height of the Merkle tree used.

In addition to XMSS, the PQCRYPTO recommendations include the SPHINCS [21] scheme, which has no restrictions on the number of signatures. However, for a 128-bit security level, the size of such a signature is about 41 kB, which is too expensive to use in a blockchain and is more suitable for authentication systems. The size of the XMSS signature is about 2.5 kB, which is quite large compared with the ECDSA signature (71 bytes), and requires an increase in the block size; however, this is the most appropriate option for post-quantum signature in a blockchain.

a. Winternitz One-Time Signature (W-OTS+)

An important cryptographic primitive is the One-Time Signature (OTS), which involves the one-time use of a key for signing a message. Signing two different messages with one key is unsafe. Bitcoin Post-Quantum uses a variant of the Winternitz One-Time Signature Scheme (W-OTS+), described by Hülsing [22] and diagrammed in the Figure 2.

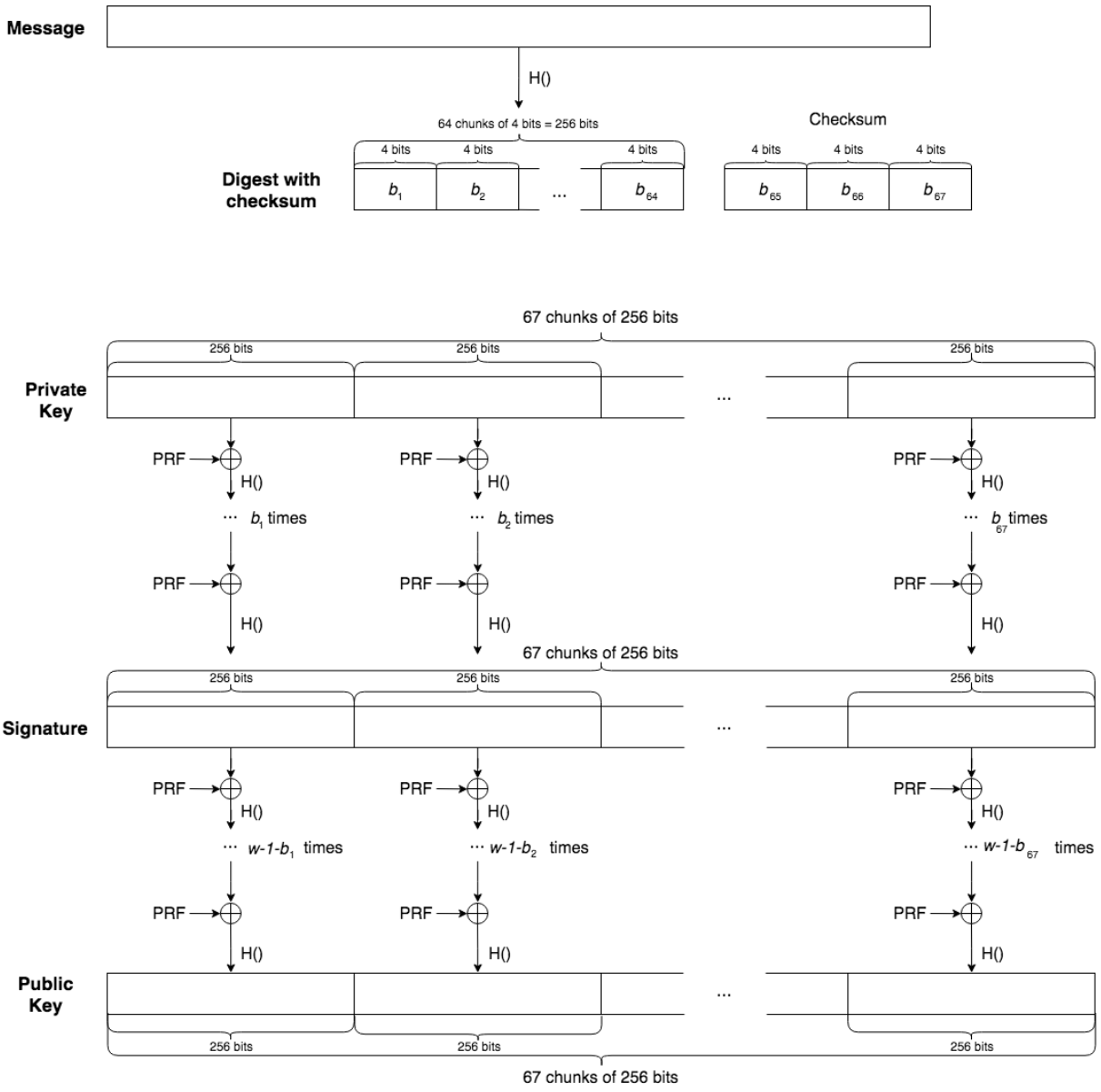

Figure 2. Scheme of W-OTS+ (n = 32, m = 256, w = 16)

Bitcoin Post-Quantum uses W-OTS+ with parameters n = 32, m = 256 and w = 16, where n is the security parameter that specifies byte length of the hash used in the signing iterations; m is the bit length of the message digest to be signed, and w is the Winternitz parameter that determines time–memory tradeoff.

A private key is a set of randomly generated 256-bit numbers. Their quantity ($l = l_1 + l_2$) is determined by the parameters m and w:

$$\ell_1 = \left\lceil \frac{m}{\log(w)} \right\rceil, \quad \ell_2 = \left\lfloor \frac{\log(\ell_1(w-1))}{\log(w)} \right\rfloor + 1, \quad \ell = \ell_1 + \ell_2.$$

In our case, $l = l_1 + l_2 = 64 + 3 = 67$.

To calculate the public key, each of these sixty seven 256-bit numbers is XORed with bit masks returned by the pseudo-random function (PRF) and hashed $w - 1 = 15$ times.

SHAKE-128(256 bits) is used as a hash function and PRF. PRF bit masks are generated from a random seed value.

For signing a message, a 256-bit message hash and a 12-bit checksum should be calculated. The hash of the message with the checksum (together 268 bits) is divided into sixty seven 4-bit numbers $b_1, b_2, \ldots, b_{67}$. Each of the 67 private key numbers is hashed b_i times; the resulting array of sixty seven 256-bit numbers being the signature.

Each of the 67 signature numbers is hashed 15 − bi times and is checked with the corresponding public key values for verification.

To prevent the private key from being disclosed, the checksum is added to the hash of the message to be signed if the hash of the message consists entirely of zeroes.

b. eXtended Merkle Signature Scheme

The eXtended Merkle Signature Scheme (XMSS) [19] allows assembling a set of one-time public keys W-OTS+ into one public key. This means that XMSS enables the public key to be used multiple times.

To build a XMSS tree of height h, a seed-value is generated randomly, from which the 2^h W-OTS+ private keys and corresponding public keys are produced. From each W-OTS+ public key, which consists of l numbers, a binary L-tree is built. The public key numbers are the leaves of the L-tree. Because l is not necessarily a power of 2 (as was shown above, in our case, $l = 67$), the tree node, which does not have a pair on the right, rises a level higher until it becomes the right pair for the other node as shown in the Figure 3.

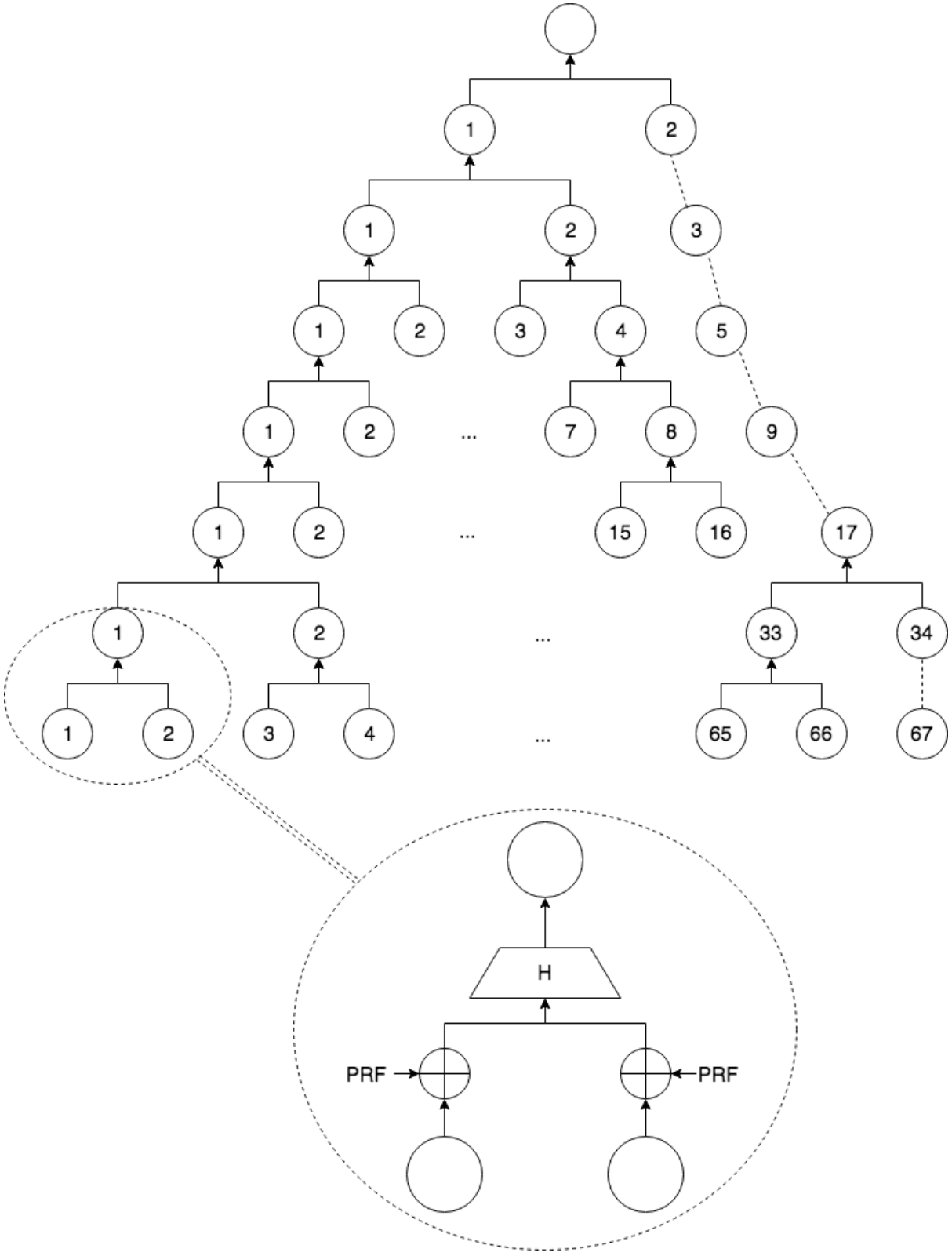

Figure 3. L-tree construction scheme

The root of each 2^h L -trees becomes a leaf of the Merkle tree as shown in the Figure 4. The root of the Merkle tree becomes the XMSS public key.

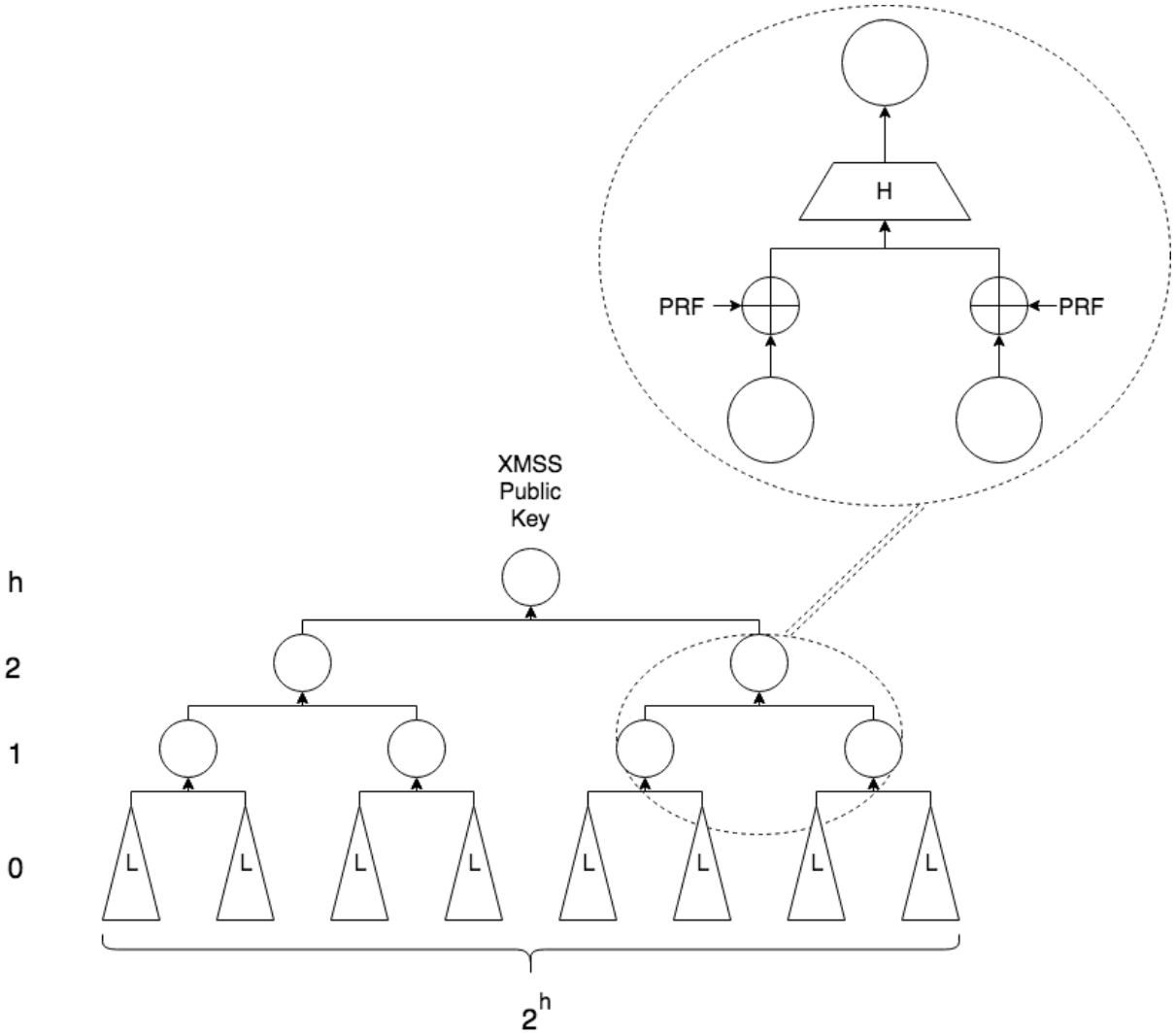

Figure 4. XMSS tree construction scheme

The first unused W-OTS+ key is used upon signing of the transaction. The XMSS signature contains the W-OTS+ signature and an authentication path and denotes the sequence of tree nodes required to calculate the root of the Merkle tree.

For instance, assume that the node (0,5) is used to sign the transaction as it is shown in the Figure 5. For the verifier to calculate the node (1,3), the node (0,6) is added to the authentication path. Next, in order to determine the node (2,2) we need the node (1,4). Similarly, for the calculation of the root (3,1) we need the node (2,1). The authentication path, therefore, consists of the nodes (0,6), (1,4) and (2,1) and, together with the W-OTS+ signature, allows verifying the signature by calculating the root of the Merkle tree and checking it with the XMSS public key.

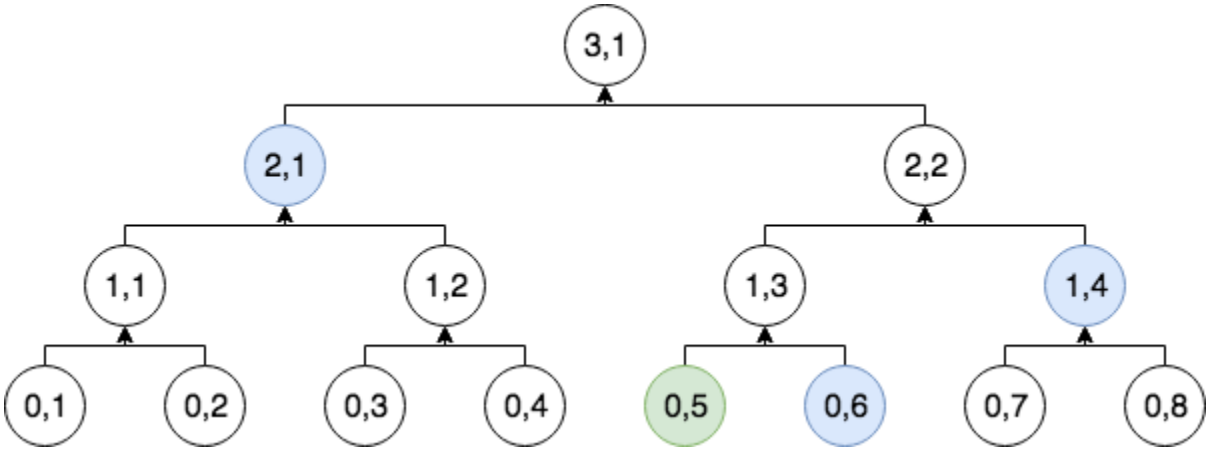

Figure 5. Authentication path

The required number of signatures is obtained by using XMSS trees of different heights *h*. BPQ applies parameters from RFC 8391 [19]:

Name	Function	n	w	len	h
XMSS-SHAKE_10_256	SHAKE128(256 bits)	32	16	67	10
XMSS-SHAKE_16_256	SHAKE128(256 bits)	32	16	67	16
XMSS-SHAKE_20_256	SHAKE128(256 bits)	32	16	67	20

The XMSS tree with height $h = 10$ allows $2^{10} = 1024$ transactions to be signed, with height $h = 16$, $2^{16} = 65536$ transactions; and with height $h = 20$, $2^{20} = 1048576$ transactions. The larger the tree the longer the generation time. On a modern computer creating a tree with a height $h = 10$ takes less than a second, $h = 16$ less than a minute, and $h = 20$ about 10 minutes.

4. Post-Quantum Bitcoin Fork

a. Blockchain Hard Fork

As a part of the existing consensus, only those transactions that have valid ECDSA signatures are accepted in Bitcoin. Meanwhile, blocks that contain transactions signed by XMSS will be rejected by Bitcoin nodes.

Bitcoin's Post-Quantum consensus operates in the following way: the network rules are identical to the Bitcoin network before the block height 555,000; the new rules will apply starting from this block. According to these rules, the support of quantum-safe XMSS signatures is implemented, the block size is increased and the mining algorithm is changed.

The owners of bitcoins at the time of the hard fork automatically receive the same number of coins in the BPQ blockchain. To protect their coins from hacking with the help of a

quantum computer, they will need to create new keys for XMSS and perform a transaction from their old addresses to their new ones. Transactions to the addresses of the old type are not supported by standard software. Although such transactions can be created manually and accepted by the network, the outputs of such transactions cannot be spent by using ECDSA.

Approximately one year after the launch of the main BPQ network, the support for old elliptic curve digital signatures will be completely disabled. The coins, which by that time will not be protected from quantum attack by transfer to the quantum-safe addresses, will be burned. Therefore, the previously lost keys will not be compromised in the BPQ blockchain. The award for mining will be increased so that the final emission of coins will be equal to 21,000,000.

The Bitcoin Post-Quantum codebase is forked from Bitcoin Core 0.16.0 and includes support for SegWit. Thus, there is no transaction malleability problem in BPQ and it is ready for the Lightning Network.

b. Quantum-Safe Addresses

Bitcoin Post-Quantum introduces modified bech32 addresses like these:

MAINNET: pq1p7mpu6tdds2q08dwvwn0srr886jej4suupqzkedn5sv48wgg5anmsprg42s

TESTNET: tq1peewnl23ptz9c0askjems7adpat6f7h9qa8l38h0th23lnyw7v0uqkthefn

REGTEST: pqrt1pnfn3yvfr4lenlpk09zmrcktw6eu0qt849cunxc0f8lnts8znv9sq0alx6w

By analogy to Bitcoin's P2WSH addresses, the quantum-safe BPQ address consists of a human-readable part "pq", separator "1", a witness program (value of the witness version is set to "1") in the modified base32 encoding, and checksum.

The P2(W)PKH addresses are not used intentionally. The money from quantum-safe addresses can be spent by scripts of all types - both requiring a single signature ([pubkey] OP_CHECKSIG) and multi-signature (*m* [pubkey$_1$] ... [pubkey$_n$] *n* OP_CHECKMULTISIG), or by any other. The identical look of the addresses for all types of scripts gives an additional advantage — obfuscation of the functional purpose of the address.

To obtain the balance in the Bitcoin Post-Quantum blockchain, users who own coins in Bitcoin's main blockchain at the time of the fork must generate an XMSS key and the corresponding pq-address. Afterwards, they should enter their old ECDSA keys into the wallet program and make the first transaction to the new address from their old addresses. Transactions from old addresses to old addresses are not supported.

Attention! For the security reasons, it is strongly recommended that you transfer the money from the keys in the main Bitcoin blockchain to addresses that are managed by other keys before you enter the old ECDSA keys into the wallet program.

c. Block Size And Weight

All BPQ transactions are SegWit-transactions (except of transactions from legacy Bitcoin addresses). Quantum-safe signatures are large in comparison with ECDSA signatures; thus, the witness scale factor is increased from 1:4 to 1:32 so as to ensure network efficiency remains the same as in Bitcoin at the current average number and configuration of transactions, and the block weight limit increases from 4 M to 32 M weight units by increasing the witness part while the limit of the non-witness part remains 1 MB. This leads to blocks with 32 M weight units being about 16 MB in size.

d. Consensus Algorithm

Proof of Work (PoW) in Bitcoin consists of a search of the random value of the special field (nonce) of the block header until the double SHA2-256 hash of the block header appears to be less than some target number. The target number is automatically calculated every 2,016 blocks based on the previous value and timestamp of the blocks so that a further search at the current hash rate lasts about 10 minutes.

An attack on Bitcoin's PoW seeks to take over the majority of the hash power. An attack of this nature would allow the attacker to perform a double spending. This means that the money will be spent in the current blockchain, and at the same time the calculation and publishing of an alternative chain of sufficient length, by which the money can be sent to another address, will occur. According to the rules of consensus, the network participants recognize a longer chain of blocks as valid.

Grover's quantum algorithm provides a quadratic acceleration in comparison with the classical search for a suitable nonce value for the mining task, which makes the production of quantum devices for block producing promising. This in turn leads to the centralization of computing power and increases the probability of a successful 51% attack [23] to an even greater extent than it is today.

As Bernstein [24] has demonstrated, a quantum computer is economically much less efficient than the classical one for parallel searching for hash function collisions. For example, shortening of the operating time of the quantum algorithm from $O(2^{n/2})$ to $O(2^{n/3})$ requires $O(2^{n/3})$ independent modules, whereas only $O(2^{n/6})$ modules are required for the classical computer to achieve the same acceleration. Thus, quantum computing devices will not help with the collision search problem. This is unlike the problem of search for a preimage, for which quantum computing devices provide quadratic acceleration.

The Equihash PoW algorithm [25], which is used in the Zcash cryptocurrency, is based on the generalized birthday problem. The problem of *Equihash*(n, k, d) involves searching for 2^k numbers $(i_1, i_2, \ldots, i_{2^k})$ of length $(n/(k+1))+1$ bits. The two conditions should be fulfilled (H() – is the hash function Blake2b):

$$H(i_1) \oplus H(i_2) \oplus \ldots \oplus H(i_2^k) = 0 \text{ and}$$

$$H(i_1 \,||\, i_2 \,||\, \ldots \,||\, i_2^k) \text{ starts with } d \text{ zero bits}$$

Originally conceived as ASIC-resistant, Equihash is quantum-resistant in the sense that the classical devices for its implementation are much more cost-effective than quantum devices; thus, it excludes the possibility of concentration of the large computing power in the hands of an attacker with access to quantum computing devices.

At the same time, centralization of classical computing power, the common problem of all Proof of Work algorithms, remains. Experience shows that no attempt to create ASIC-resistant algorithms has been successful. The algorithms Ethash, Cryptonight and Equihash, which were developed to serve this purpose, were eventually implemented as application-specific integrated circuits. For any kind of calculation, it is possible to create specialized hardware that will be tailored to a particular task and cope with it more effectively than general purpose devices (CPU or GPU).

The cryptocurrencies most vulnerable to 51% attack are those with a small computing power of the network, which copy the mining algorithm from another cryptocurrency with larger computing power. Using the example of *Equihash*(n = 200, k = 9), in order to perform a double-spending attack, any small Zcash mining pool, which has several percent of the total network computing power, can, unnoticed for the pool members, quickly switch the part of the computing power to mining blocks in another blockchain with the same algorithm, but with a sufficiently lower computing power. Bitcoin Gold and Zencash were attacked this way in 2018, and, most likely, other cryptocurrencies with a small computing power, which are using the popular algorithms *Equihash*(n = 200, k = 9) and *Equihash*(n = 144, k = 5) will also be attacked.

For the proof of work BPQ uses Equihash with parameters n = 96, k = 3 (which are unique at the present time) as a computational puzzle and hash function SHA2-256 for its difficulty adjusting.

5. Privacy

Privacy is an obligatory feature of money. Nobody has the right to know how much money you have in your wallet, or when, to whom, and for what you pay. This information should be private until you consider it necessary to disclose it to someone.

Bitcoin's initial orientation to the anonymity of public keys and their regular change to achieve privacy [26] did not justify itself because the connections between the used public keys are visible in the blockchain and are easy to analyze.

Some cryptocurrencies try to solve this problem by using non-interactive zero-knowledge proof schemes such as zk-SNARKs. The proposed schemes of confidential transactions and MimbleWimble are also of interest. However, as these solutions are not quantum-safe, they cannot be considered safe at all. In addition, it is important to understand

that all anonymous transactions that are done today with the help of quantum-unsafe algorithms, will be easily disclosed by a quantum computer in the future and thus cannot be considered fully anonymous today.

In subsequent protocol updates, Bitcoin Post-Quantum will use quantum-safe non-interactive zero-knowledge proofs like ZKB++/Picnic [27] and zk-STARKs [28] to achieve privacy.

6. References

[1] Cryptography Today
http://web.archive.org/web/20151116013105/https://www.nsa.gov/ia/programs/suiteb_cryptography/index.shtml

[2] IBM Q Experience https://quantumexperience.ng.bluemix.net/qx/experience

[3] Intel Bets It Can Turn Everyday Silicon into Quantum Computing's Wonder Material
https://www.technologyreview.com/s/603165/intel-bets-it-can-turn-everyday-silicon-into-quantum-computings-wonder-material/

[4] Experimenting with Post-Quantum Cryptography
https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

[5] Commercialize quantum technologies in five years
https://www.nature.com/news/commercialize-quantum-technologies-in-five-years-1.21583

[6] OpenSSH 7.7 adds experimental support for PQC XMSS keys
https://www.openssh.com/txt/release-7.7

[7] Microsoft Adds Post-Quantum Cryptography to an OpenVPN Fork
https://www.bleepingcomputer.com/news/microsoft/microsoft-adds-post-quantum-cryptography-to-an-openvpn-fork/

[8] D. Deutsch. Fabric of reality. Chapter 2 "Shadows"
https://www.amazon.com/Fabric-Reality-Parallel-Universes-Implications-ebook/dp/B005KGJX8E/

[9] D. Deutsch. The Beginning of Infinity. Explanations that Transform the World. Chapter 11 "The Multiverse"
https://www.amazon.com/Beginning-Infinity-Explanations-Transform-World-ebook/dp/B005DXR5ZC/

[10] D. Deutsch. Quantum theory, the Church-Turing principle and the universal quantum computer. https://people.eecs.berkeley.edu/~christos/classics/Deutsch_quantum_theory.pdf

[11] P. Shor. Algorithms for Quantum Computation: Discrete Log and Factoring http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.4485&rep=rep1&type=pdf

[12] L. Vandersypen, M. Steffen, G. Breyta, C. Yannoni, M. Sherwood, I. Chuang. IBM Almaden Research Center. Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance http://cryptome.org/shor-nature.pdf

[13] J. Proos, C. Zalka. Department of Combinatorics and Optimization University of Waterloo. Shor's discrete logarithm quantum algorithm for elliptic curves. https://arxiv.org/pdf/quant-ph/0301141v2.pdf

[14] M. Roetteler, M. Naehrig, K. Svore, K. Lauter. Microsoft Research, USA. Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms https://eprint.iacr.org/2017/598.pdf

[15] How many bitcoins are vulnerable to a hypothetical quantum attack? https://medium.com/@sashagnip/how-many-bitcoins-are-vulnerable-to-a-hypothetical-quantum-attack-3e59e4172e8

[16] L. Grover. A fast quantum mechanical algorithm for database search. https://arxiv.org/pdf/quant-ph/9605043.pdf

[17] Post-quantum cryptography. http://pqcrypto.org/

[18] PQCRYPTO. Initial recommendations of long-term secure post-quantum systems. http://pqcrypto.eu.org/docs/initial-recommendations.pdf

[19] J. Buchmann, E. Dahmen, A. Hülsing. XMSS – A Practical Forward Secure Signature Scheme based on Minimal Security Assumptions. https://eprint.iacr.org/2011/484.pdf

[20] A. Hülsing, D. Butin, S. Gazdag, J. Rijneveld, A. Mohaisen. XMSS: eXtended Merkle Signature Scheme. RFC 8391. https://datatracker.ietf.org/doc/rfc8391/?include_text=1

[21] SPHINCS: practical stateless hash-based signatures http://sphincs.cr.yp.to/

[22] A. Hülsing. W-OTS+ Shorter Signatures for Hash-Based Signature Schemes. https://huelsing.files.wordpress.com/2013/05/wotsspr.pdf

[23] Blockchain: how a 51% attack works (double spend attack) https://medium.com/coinmonks/what-is-a-51-attack-or-double-spend-attack-aa108db63474

[24] D. Bernstein. Cost analysis of hash collisions: Will quantum computers make SHARCS obsolete? https://cr.yp.to/hash/collisioncost-20090517.pdf

[25] A. Biryukov, D. Khovratovich. Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem. https://www.cryptolux.org/images/b/b9/Equihash.pdf

[26] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Section 10 "Privacy" https://bitcoin.org/bitcoin.pdf

[27] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, G. Zaverucha. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives. https://eprint.iacr.org/2017/279.pdf

[28] E. Ben-Sasson, I. Bentov, Y. Horesh, M. Riabzev. Scalable, transparent, and post-quantum secure computational integrity. https://eprint.iacr.org/2018/046.pdf