International Conference on Innovative Research in Engineering, Management and Sciences
International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 4 | Issue 9 | ISSN : 2456-3307

Enhancement of Signature Schemes for Heightening Security in Blockchain

Prof M ShanmugamShoba*[1], Dr. Rekha B Venkatapur[2]
*[1]Senior Assistant Professor, Information Science & Engineering Department, NHCE, Bangalore, India
[2]Professor & Head, CSE, KSIT, Bengaluru, India
reach2shobhamahe@gmail.com/mshanmugams@newhorizonindia.edu[1]

ABSTRACT

Blockchain has become one of the most pioneering technologies, with the rise of Bitcoin, blockchain which is the core technology of Bitcoin has received increasing attention. There are multiple signature scheme based on digital signature schemes that supports making signatures on many different messages generated by many different users, the size of the signature could be shortened by compressing multiple signatures into a single signature. Based on the blockchain architecture and existing Merkle tree based signature schemes, In this paper, an analysis of how to enhance the signature schemes to secure the transactions on blockchain based on extensible post-quantum (PQ) resistant digital signature scheme best suited to blockchain and distributed ledger technologies is proposed.

I. INTRODUCTION

Recent advances in quantum computing and the threat this poses to classical cryptography has increased the interest in PQ research. More specifically, due to Shor's algorithm [1], a quantum computer could easily factor a big integer in polynomial time, thus effectively break RSA. Implementations of Shor's algorithm can also solve discrete logarithms and render today's digital signatures, such as DSA, ECDSA and EdDSA, useless[2].

The race to build quantum computers has already begun and companies like Google, Microsoft, IBM, D-Wave and Intel are at the forefront. That being said, we have yet to build a computer with the thousands of stable qubits that are required to make classical public key cryptography obsolete. However, there is significant progress in the field and some optimistic predictions estimate that a large quantum computer capable of breaking asymmetric cryptography might be available in the next10to20years[3],[4].

The security impact of breaking public key cryptography would be tremendous, as almost everything from HTTPS, VPN and PKI in general, is basing their authentication, key exchange and digital signatures on the security of RSA or Elliptic Curve Cryptography (ECC). Blockchains would be hit equally hard resulting in broken keys that hold coins/assets, and would perhaps be one of the most affected sectors because there is economic incentive for hackers to get access to blockchain accounts anonymously.

To address the concern of compromised keys, PQ cryptog- raphy is dealing with the design and evaluation of systems that will survive the quantum supremacy. An enhancedsolution is a modified version of the hash-based XMSS [5] family of schemes. It practically makes use of a single authentication path; thus, it is a chain and not a tree and it mainly focuses on {one and limited}-time keys, which is usually the most applicable to blockchains as we want to preserve anonymity and minimisetracking.

Compared to existing schemes, the approach outperforms limited-time schemes when required to sign only once or afew times. Unlike one-time schemes (OTS), this schemes provide a fallback mechanism to easily support many-time signatures. Moreover, the underlying logic of a "blockchained" authentication path could be applied to convert any existing hash-based scheme to a {one and/or few}-time optimise done. To our knowledge, this is the first signature scheme that can utilise an existing blockchainor graph structure to reduce the signature cost to one OTS, even when we plan to sign many times. This makes existing many-time stateful signature schemes obsolete for blockchain applications. Moreover, the scheme is solely based on hash functions and that no special math theory is required for its implementation makes it a promising candidate for existing or new blockchain applications, and for low-end devices, such as in IoT applications, where hashing operations are already implemented and sometimeshardware optimized.

II. HASH FUNCTION IN BLOCKCHAINS

A cryptographic hash function is an algorithm that maps data ofarbitrary size to a fixed size string. Two security requirements namedone-wayness and collision-resistance are usually required for hash functions. The former ensures that the underlying hash function is notinvertible, while the latter implies that it is not easy to find two inputshaving the same hash value. For a hash function withn-bit length out-put, the complexities of breaking one wayness and finding a collisionare respectively bounded by O(2n)brute force attack and O(2n/2)birth day attack. Therefore, for ensuring at least 80-bit security, the output length of hash functions should be at least 160 bits. The most popular hash function used in blockchains is SHA256, which is one of the algorithms from a family of cryptographic hash functions named SHA (Secure Hash Algorithms). SHA is a U.S. Federal Information Processing Standard, and most of the algorithms in this family, including SHA0 (published in 1993), SHA1 (published in1995), SHA2 (published in 2001) are designed by the United StatesNational Security Agency (NSA). While SHA3 (published in 2014) isoriginal from Keccak proposed by (Bertoni et al., 2010), and only thepadding method is modified by the National Institute of Standards andTechnology(NIST).To satisfythe current security requirement, SHA2 and SHA3 are recommendedfor using in blockchains and cryptocurrencies.

III. DIGITAL SIGNATURES

Besides the hash function, the digital signature is another inevitablecryptographic primitive in blockchains. The concept of the digital signature was put forward by Diffie and Hellman in 1976 when they openedthe gate of public key cryptography (Diffie and Hellman, 1976). As abasic primitive of cryptography, digital signature is used for ensuringthe source authentication (Lin et al., 2018), source non-repudiation andintegrity. The standard security of the digital signature is existentialunforgeability against adaptively chosen messages attacks (EUF-CMA),which guarantees that the adversarycannot forge a valid signature on anew message, even if it can access the signing oracle that could providethe signing service.ECDSA (Certicom-Research, 2000)andEdDSA(Bernstein et al.,2011) are the two digital signature schemes frequently used inblockchains. In principle, both of them are based on the hardness ofthe elliptic curve version of discrete logarithm problem. ECDSA worksover a general elliptic curve and now is used in Bitcoin and Ethereum,whileEdDSA works over a (twisted) Edward curve and now is usedin Naïve coin and

Monero. The Edward curve is a plane model of anelliptic curve and has better efficiency and security than a generalelliptic curve. Thus, it has been already selected as the next ellipticcurve generation of TLS by Internet Research Professional WorkingGroup.

IV. SPECIAL SIGNATURE PRIMITIVES FOR BLOCKCHAINS

To enhance the privacy and anonymity of transactions, someadvanced signature primitives such as ring signature and multi-signature are also widely applied in blockchains

4.1 Ring signatures

Anonymity is always required in information systems (Shen et al.,2018), especially in the e-cash system. However, Bitcoin can only provide pseudonymity due to the linkability of transactions. Therefore,many new alternative cryptocurrencies have been proposed to addressthis problem. From a perspective of cryptography, there are many kindsof signatures for achieving anonymity, such as blind signature (Chaum,1982), ring signature (Rivest et al., 2001), group signature (ChaumandvanHeyst, 1991)andDC-nets(Chaum, 1988). However, only ring signature and its variants have been used in blockchains for anonymity.The concept of ring signature was proposed in 2001 by Rivest,Shamir and Tauman (Rivest et al., 2001).Onecanusearing signature scheme to sign messages on behalf of a group including him-self/herself without revealing himself/herself, while he/she can compose this group without other group members' permission. Besidesthe existential unforgeability, the unconditional anonymity is anotherimportant security requirement for ring signature. This new propertycan be divided into two sub-properties: untraceability and unlinkability. The former means that one cannot identify the signer, while thelatter says that no one can decide whether two signatures are generatedby the same signer. The unconditional anonymity is a strong security notion that would be a double edged sword: On one hand, it providesperfect privacy protection towards individual signing behavior. On theother hand, it could be abused for some illegal purpose such as washtrading. Therefore, some restrictions on anonymity should be taken intoconsideration. In fact, even ten years before the concept of ring signature, Chaum (Chaum and van Heyst, 1991) proposed the concept ofgroup signature, which allows a group member to anonymously sign amessage on behalf of the group, with the restriction that a designatedgroup manager is able to identify the signer whenever it is necessary.One of the main differences between group signature and ring signaturelies in that the ring structure is an ad hoc group that can be formed in anon the-fly manner, while the group structure is formed under the control of the group manager. Furthermore, anyone who wants to adjointhe group has to at first perform a registration process — either onlineor offline.In 2004, Liu et al. (Liu et al., 2004) proposed a linkable spontaneous anonymousgroup(LSAG) signature scheme, which is essentiallyalinkable ringsignature considering the spontaneous group formationand no group manager (Sun et al., 2017). Recently, Liu et al.'s ideawas adopted by Back (Back, 2015) in designing Ring-Coin with necessary improvements in efficiency. Along with another line, Fujisaki etal. (Fujisaki and Suzuki, 2007) in 2007 extended the concept of ring signature into the so called traceable ring signature by adding an issue related tag into the signature. In this case, anyone in the ring, pretending to be another person to sign the same message, would face the riskof revealing his/her identity immediately. This idea was adopted to prevent double-spending and now becomes the basis of CryptoNote (vanSaberhagen, 2013) with a slight modification.However, either CryptoNote or Ring-Coin suffers from the possibleattack based on the observation and analysis of the amounts sent ina given transaction (Noether, 2015). To hide amounts for any transaction, Maxwell (Maxwell, 2017) proposed the concept of the confidentialtransaction by using homomorphic commitment protocol.

Shortly afterward, Noether (Noether, 2015) offered a modification to the Moneroprotocol by coupling three techniques: Maxwell's confidential transaction, ring signature and multilayered linkable spontaneous, anonymousgroup signature (MLSAG). Noether's idea is now named as Ring Confidential Transactions for Monero (RingCT for short).In 2017, Sun et al. (Sun et al., 2017) proposed a non-trivial upgradedversion towards RingCT, named as RingCT 2.0. Besides the rigorous formalization of the syntax of RingCT and formal security models, RingCT2.0 also applies some well-known cryptographic primitives, includingPedersen commitment, the accumulator with one-way domain and sig-nature of knowledge, to obtain the significant storage and communication cost saving. More specifically, the signature size is reducedfromO(nm)to O(m),wherenandmare the numbers of groups andaccounts in one group, respectively. In other words, the transaction size in RingCT 2.0 is independent of the number of groups in the ring, andthis enables each block to process more transactions

4.2 One-time (ring) signatures

Lamport in 1979 (Lamport, 1979) proposed the concept of one-timesignature (OTS), where the signing key can be usedsecurely but onlyonce, and the signing key would be revealed if it is used twice or more.OTS is frequently used as a building block in constructions of encryptions and authenticated key agreements.By combining the ideas of OTS and ring signature, Saberhagen (vanSaberhagen, 2013) proposed a new signature scheme where the privatekey can be used only once for signing on behalf of a group. Supposethat Bob ' s public key is(A,B), and Alice wants to send a payment toBob. Then, Alice can pick a random numberr ∈ 𝔽qand compute thetransaction public keyRand the destination keyPas follows:

R=rGandP=Ḩs(rA)G+B,

Where Ḩs : {0,1}∗ → 𝔽qis a cryptographic hash function andGis thepublic base point of the elliptic curveE(𝔽q). Then, Bob can locate Alice'spayment via checking every past transaction on the blockchain with hisprivate key pairs(a,b) to see ifP=Ḩs(aR)G+Bholds. After locating Alice's payment, Bob can recover the corresponding one-time private key x=Ḩs(aR)+b(6)and spend this output at any time by signing a transaction withx.

4.3 Borromean (ring) signatures

Another interesting primitive related to ring signature andblockchain is the so-called Borromean (ring) signature (BRS), proposedby Maxwell and Poelstra in 2015 (Maxwell and Poelstra, 2015). Poelstra(Poelstra, 2017) claimed that BRS is now used in Elements (Element,2015), Liquid (Liquid) and Monero. Moreover, all of those projects arenow being transited from the BRS-based range proofs to Bulletproofs(Bünz et al., 2017).In an abstract view, a ring signature is nothing than a signature thatthe signer knows one of secret keys foragivengroup,sayx1 ∨ x2 ∨ • • • ∨ xn,while a Borromean ring signature extends this idea to the scenariowhere the signer knows one of secrets foreach given group,say(x1∨x2∨ • • •)∧(y1∨y2 ∨ • • •)∧ • • • ∧(z1∨z2∨ • • •).Apparently, this idea gains the capability to express knowledge of anymonotone boolean function of the signing keys (Maxwell and Poelstra,2015).Although the primitive of attribute-based signature (ABS) (Majietal., 2011) can also realize theformula by considering signing keysx1,x2,...,y1,y2,...,z1,z2,···as attributes and modeling the signingcapability as a tree-like access structure corresponding ,thereexists an essential difference between ABS and BRS. In particular, ABSfocuses on who can generate a valid signature, while BRS focuses onhow to aggregate multiple ring signatures anonymously. That is, thevalidations of all involved ring signatures in a BRS scheme are intertwined. If one of the ring signatures involved in the joint Borromeansignature is invalid, then the entiresignature is invalid, and you cannottell which one is invalid (Poelstra, 2017). This is the very reason forthe name Borromean ring signature. In topological,Borromean ringsis astyle of interlocking rings such that each ring goes through each otherring (Poelstra, 2017;Cromwell et al., 1998).The construction of BRS scheme in reference (Maxwell and Poel-stra, 2015) is based on an elegant combination of several efficient techniques, including Schnoor authentication(Schnorr, 1991), AOS ring sig-nature (Abe et al., 2002), and the newly developed "halfchameleon hash" and "multiple chameleon hash". Interested readers are suggestedto refer (Maxwell and Poelstra, 2015)formoredetails.

4.4 Multi-signatures The primitive of multi

Signatureallowsasinglesignaturetoworkas several ordinary signatures on the same message. One of the critical requirements of multi-signature is that the single signature has thesame size as one regular signature. This primitive was introduced by(Itakura and Nakamura, 1983) in 1983 and has been studied over thepast decades (Ohta and Okamoto, 1999;Okamoto, 1988;Boldyreva,2002;Micali et al., 2001).Very recently, ZILLIQA team (Zilliqa) proposed the next gener-ation high throughput blockchain platform by using an EC-Schnorr multi-signature protocol as one of its innovative ingredients. Morespecifically, the protocol in ZILLIQA consists of the followingsteps:

•The standard Schnorr signature scheme (Schnorr, 1991) is instantiated over the elliptic curve specified bysecp256k1(Certicom-Re-search, 2000).

• The above EC-Schnorr signature scheme for a single user is extendedto an EC-Schnorr multi-signature scheme for multiple users based onthe idea in reference (Micali et al., 2001)

.•The above EC-Schnorr multi-signature is tweaked for PBFT (practical Byzantine fault tolerance) settings, where the message isrequired to be properly signed by at least23n+1 nodes in the committee.

V. HASH-BASED POST-QUANTUM DIGITALSIGNATURES

5.1One-TimeSignatures

Hash-based signature schemes have been documented in the literature since 1979, thanks to the LamportOTS scheme .The logic behind Lamport's scheme is straightforward,the signer generates pairs of random values per bit required to be signed and these pairs form the private key. The public key is formed by the hashes of those values. To sign a message, the signer reads the message bitwise and presents one value from each secret pair depending on the bit value. The verifier can then validate that the hashes of all the secret values are equal tothecorrespondinghashvaluesinthepublickey.

Although Lamport OTS hash computations are considered fast, key and signature sizes are relatively large. For instance, if SHA256 is used as the underlying hash function, the public key consists of 512 hashed outputs of 256 bits each (one hash-pair per bit), while the signature consists of 256 secret values (256 bits each). If we aggregate the above, the key and signature consist of 24.5 kB. Similarly, if SHA512 is applied, about 98 kB are required.

Further enhancements to the original algorithm, reduce the key size significantly. At present, the WOTSalgorithmanditsvariantsareconsideredsomeoft he most efficient key and signature compression methods, while Bleichenbacher and Maurer's graph-based scheme attempts to achieve the best possible efficiency in terms of signature size and number of hash function evaluations per bit of the message.

As a note, one of the main differences between OTS approaches lies in the security assumptions of requiring (or not) collision resistant hash functions and the use ofextra bitmasks. Currently, WOTS-T, proven to be secure in the QROM model, is considered one the the most promising candidates from the WOTSfamily, because only one extra seed value is required along with the public key to compute the required bitmasks, while its security is not affected by the birthday paradox and it also introduces keying of all hash functioncallstopreventmulti-targetsecondpre-imageattacks. The latter results in shorter public keys and hash-outputsizes.

5.2Few- and Many-TimeSignatures

Although there exist multiple methods to turn a one-time into a multi-time signature scheme ,a popular approach is to use Merkle authentication trees by fixingbeforehand the total number of signatures which will ever be produced. Using Merkle trees, the total number of signatures which can be issued is defined at key generation. The main benefit is its short signature output and fast verification, while the drawbacks are the relatively expensive key generation time and the fact that they are stateful. Figure 1 depicts a 4-time (atmaximum)Merkletreesignaturescheme.

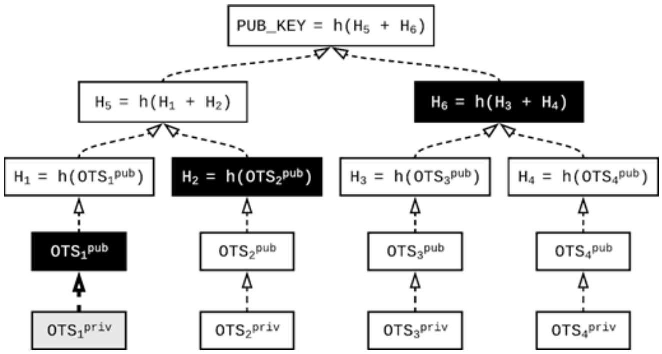

Fig. 1:Few-time Merkle tree signature scheme able to sign four messages in total. Dark nodes represent the authentication path required if we sign with OTS 1.

Moving to stateless few-time signatures requires extra complexity and larger signature outputs. HORS (and its extension HORST) is currently the one used in the majority of many-time stateless signature schemes, such as SPHINCS .

Many-time hash-based schemes can be constructed by combining the above {one and few} time constructions and they are grouped into two categories, stateful (e.g., XMSS, LMS) and stateless (e.g., SPHINCS, SPHINCS+, Gravity, Simpira, Haraka). Stateful schemes typically produce shorter signatures, but they need a mechanism to keep state (what paths/keys have already been used).

On the other hand, stateless schemes start with a moderately large Merkle tree or tree-layers at the top, but instead of using OTS signatures at the bottom, they use a few-time signature method. The latter allows them to pick indices randomly and thus no path-state tracking is required. The downsideto stateless schemes is their signature size; for instance, in SPHINCS-256eachsignatureis41kBlong.

It is highlighted that the distinction between few- and many- time hash-based signature schemes is not always clear. In the literature, few-time usually refers to stateless schemes, such as BiBa, HORS and HORST, for which practical parameters allow multiple signing operations, but not enough signatures to be considered in many real-world applications. On the other hand, many-time schemes can be configured to allow highly interactive environments to reuse the same key- pair for many years. The authors of Gravity SPHINCS claim that 1 trillion (240) signatures is a reasonable upper bound, whilst SPHINCS-256 allows for a maximum of 1 quadrillion (250) signatures. In practice, one can parameterize amany-timeschemetosupportjustafeworseveralsignatures.

5.3SpeedandSecurityofHashFunctions

The underlying hash algorithm is of obvious importance to the overall security of the proposed scheme. Several factors influence the choice of algorithm, including speed, security level and availability; e.g., what hardware features can be leveraged to improve the runtime performance, and what implementations are available in existing, well-reviewed cryptography libraries.

The first thing to establish, however, is whether the al- gorithm is resilient to PQ attacks. The SHA-2 and SHA-3 algorithms support multiple digest sizes, namely 224, 256, 384 and 512 bits [36], [37]. We observe that by leveraging the improved search speed provided by Grover's algorithm, collision resistance can be reduced from a half to a third of the chosendigestsize.Consequently,inthepresenceoflarge-scale quantum computers, 384-bit versions of SHA-2 and SHA-3 would provide 128 bits of security against collisions, whereas the256-bitversionswouldonlyoffer85bits.

Further, we observe that quantum pre-image attacks on 256- bit versions of SHA-2 and SHA-3 can be realised by 2153.8 and 2146.5 surface code cycles, respectively[38].

Asaresultofthesetwoobservations,SHA256isconsidered unsuitable for use in schemes basing their security on hash collision resistance, but it is still secure otherwise. It should also be mentioned that

PQ algorithms have fundamentally worse price-performance ratio than the classical vanOorschot-Wiener hash-collision circuits, even under optimistic assumptionsregardingthespeedofquantumcomputers[39].

From performance measurements presented in eBACS [40], we have evaluated the relative performance of SHA-2, SHA-3 and BLAKE2 on general-purpose CPUs. We have deliberately chosen an Intel, an AMD and an ARM processor to cover typical desktop and mobile units.

As can be seen from Table 1, the number of cycles per byte decreases with the size of the input. This is expected due to the small input sizes in this comparison and the block-wise operation mode of the hash functions. The rate of decrease naturally flattens out as the input grows beyond the blocksize.

It should also be noted that the different versions of SHA-3 generally performs worse than their SHA-2 counterparts. One of the reasons for this is the fact that SHA-1 and SHA-2 have better hardware support from modern processors, e.g., through instruction set extensions like the Intel® SHA Extensions.

Note that, despite not offering protection against length extension attacks, SHA-2 offers similar bit-level security to SHA-3. Typically, hash-based PQ schemes, including BPQS(Blockchain Post Quantum Signature), are not prone to such attacks and therefore, we consider SHA- 2 to be a better alternative due to the performance benefits it offers.

If performance is of importance, one can also consider employing the less supported BLAKE2b [41] algorithm. We highlight, however, the lack of wide-spread library support compared to the aforementioned algorithms.

Table 1: Performance Metrics for SHA-2, SHA-3 and Blake

Input Size	**Measurements of Hash Functions**[a] Cycles / Byte (relative to SHA2-256 on 8 byte input)								
	SHA-2			**SHA-3**			**BLAKE2**[b]		
	Intel	AMD	ARM	Intel	AMD	ARM[c]	Intel	AMD	ARM
256-bit utput									
8	1.00	0.19	2.99	3.48	2.89	6.78	0.47	0.38	2.30
64	0.24	0.04	0.54	0.46	0.38	0.85	0.05	0.04	0.28
576	0.13	0.02	0.20	0.25	0.21	0.40	0.05	0.04	0.15
512-bit utput									
8	1.49	1.12	5.72	3.58	3.00	6.79	0.53	0.46	3.23
64	0.19	0.14	0.71	0.48	0.40	0.85	0.07	0.05	0.41
576	0.09	0.05	0.30	0.43	0.36	0.71	0.03	0.03	0.14

[a]Based on numbers reported by ECRYPT II in eBACS [40].

Intel - amd64, genji122,supercop-20171020

AMD - amd64, genji262,supercop-20171020

ARM - armeabi, odroid,supercop-20160806

[b]BLAKE2s with 32-bit words, 10 rounds, and 256-bit output; BLAKE2b with 64-bit words, 12 rounds, and 512-bit output.

[c]No data for SHA-3; numbers are for keccakc512/1024 with 256- and 512-bit output sizes, respectively. These are the Keccak team's final submissions for SHA-3-256 and SHA-3-512.

VI. BLOCKCHAINED POST-QUANTUM SIGNATURES TAILORED TO ONE-TIME KEYS

Most if not all few-time hash-based signature schemesmake use of Merkle trees. The maximum number of messages a basic Merkle tree signature scheme can sign is 2h, where h is the height of the tree. Also, all leaves (keys) should be computed during key generation in order to form the root. Due to the above, to construct a tree of height h = 40, key generation would be considered impractical, because we need to compute 240OTS keys and each OTS key internally requires many hash invocations (i.e., 512 hash invocations with LamportOTS or 67 for WOTS (w = 16) when using SHA256). The trick to keeping key generation time practical, while allowing for a large number of signatures is to use a multi-leveltree.

BPQS is a simplified single-chain variant of the XMSS family protocols which are literally an extension of the basic Merkle tree signature scheme (see Figure 1). BPQS can theoretically sign many times, but its design focuses on short and fast one-time signatures with the extra option to re-sign if and when needed. The above requirement is what a typical blockchain or DLT requires, as the use of one-time keys is recommended to preserve anonymity. However, a lot of things can go wrong, e.g., a transaction might not go through or there might be a fork in the chain, in which case one should be able to sign more than one time without compromising security or freezing assets.

An additional, surprising benefit of BPQS is that it is alsoan ideal candidate for the opposite requirement; signing multiple times with the same key. This interesting property is due to the underlying graph-structure of blockchain and DLT systems that effectively allow many-time signatures at a minimal cost comparedtootherhash-basedPQsolutions.Thisworksbyreferencing the block (or transaction) in which the same BPQS key has been used in the past. In short, only a small part of the new signature is required to be submitted and the rest of the path will be delegated to the previous transaction this key was used to sign. The latter enables us to complete the full path to the advertised root BPQS key. Actually, because previous transactions are verified on the ledger already, verifiers do not even need to validate the rest of the path, as it was inherently verified in the past. This characteristic makes BPQS very useful for notary-based DLTs, such as Cordaand Fabric, as the notary nodes normally sign transactions with the same knownkey.

6.1BPQSScheme

BPQS requires an underlying OTS scheme. Although any OTS solution could in theory be applied, our scheme shares logic with the XMSS protocol family, hence the selection of the WOTS variant, use of L-Trees and generation of bitmasks (blinding masks) define the security assumptions and proofs,similarlytoXMSS,itsmulti-levelversionXMSSMTand XMSS-T [27]. Also, according to, collision resistance is actually cheaper using quantum algorithms, and thus similarly to the Gravity SPHINCS scheme, bitmasks and L-Trees might beomitted.

One could state that BPQS is a subset of XMSS tailored to fast first-time signatures. The main difference is that XMSS overcomes the limitation to one message per key by usinghash trees which reduce the authenticity of many OTS verification keys to one public XMSS root key. In contrast, BPQS utilises a chain of small 2-leaf Merkle trees. Geometrically, XMSS grows in both width and height (see Figure 1), while BPQS grows on chain height only (see Figure 2). All in all,we stress that BPQS is a generic blockchained construction,where blocks are “tiny” Merkle trees, meaning that it can be parameterised according to the requirements of the application. In case blinding masks are applied, their deterministic generation should follow the same logic with the corresponding XMSS familyscheme.

There are 2 basic building blocks of BPQS:

- BPQS-FEW, which strictly supports few-time signatures andisdepictedinFigure2(left),
- BPQS-EXT, which theoretically can be extended to sup- portmany-timesignatures,seeFigure2(right).

In BPQS-FEW, all keys are precomputed during key gen- eration, the penalty for each extra signature is just 1 extrahash output, but it cannot be extended to practically support “unlimited”signatures.

On the other hand, BPQS-EXT initially requires only two OTS keys and in contrast to BPQS-FEW, the left leaf in each 2-leaf Merkle tree is an OTS fallback key that can be used to sign the next signature block when required. Unfortunately, the extensibility property comes with the cost of requiring one extra WOTS key per newsignature.

The full BPQS scheme combines both BPQS-FEW and BPQS-EXT in a way where the last leaf in the chain of BPQS- FEW is a BPQS-EXT fallback key. This trick allows us toconvert the few-time variant to a many-time one. Actually, BPQS-EXT can be considered a special case of BPQS, in which there is no initial BPQS-FEW chain.

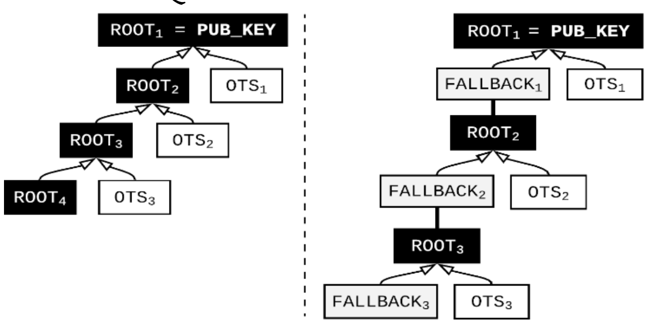

Fig. 2: BPQS-FEW (left), a few-time signature scheme. BPQS-EXT (right), a linearly extensible many-time signature scheme.

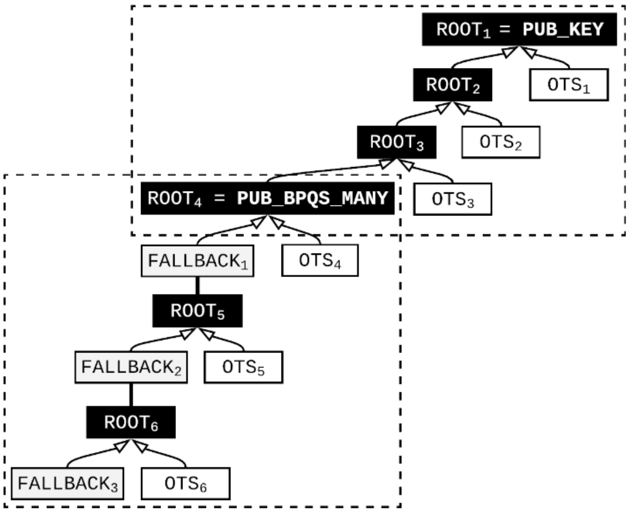

Fig. 3: Full BPQS protocol, with a height-3 BPQS-FEW top-level and a BPQS-EXT fallback key to allow for extensibility.

Parameters for BPQS include:

- the WOTS variant used (e.g., WOTS-T[27]),
- theWinternitz parameter (e.g., w = 16), which defines the base at which the initial hash is interpreted. Similarly to XMSS [5], w defines the actual size of each WOTS chain, which in turn affects signature size. Note that there is no consistency on the interpretation of the Winternitzparameter in the literature. For instance, in LMS [45] it is defined as 2^w and thus $w_{BPQS} = 16 = 2^4$ would be equivalent to $w_{LMS} = 4$,
- theunderlyinghashfunction(i.e.,SHA384),
- thenumberofprecomputedOTSkeys,meaningtheinitial height (e.g., h =4).

6.2BPQSMixed

The extensibility property of BPQS enables various custom constructions. BPQS can be used as a building block toconvert any hash-based signature scheme into a {first or few} time optimised one. For instance, in Figure 4, BPQS-FEW is used for the first (shorter) signatures and then it fallbacks to another PQ scheme. Although in the depicted approach the key-pair of the fallback (other) PQ scheme should be a-priori known and precomputed, one could use the BPQS-EXT in a similar fashion, so that this is not necessarily a requirement and the "other" PQ key will be generated only after the few-time signatures are exhausted. Moreover, if the "other" PQ scheme is stateless, such as SPHINCS, the final protocol is literally a "startstatefulthengostateless"scheme.

It should be emphasized that the "other" PQ scheme might be another BPQS scheme, so one could eventually create a chain of different BPQS schemes. The latter would result in shorter signatures versus just extending it with BPQS-EXT each time.

With regards to the "Other PQ Key Params" shown in Figure 4, it is important that some schemes are required to publish bitmasks (or a seed in XMSS-T [27]) as part of the initial advertised public key. Otherwise, it would allow an adversary to select the seed/bitmask in a forgery. However, if BPQS uses a hash function with a bigger output (e.g., SHA384 or SHA512) this might not be necessary, because the provided security-level against potential quantum collision attackswouldstillbeenoughtopreventsuchattacks.

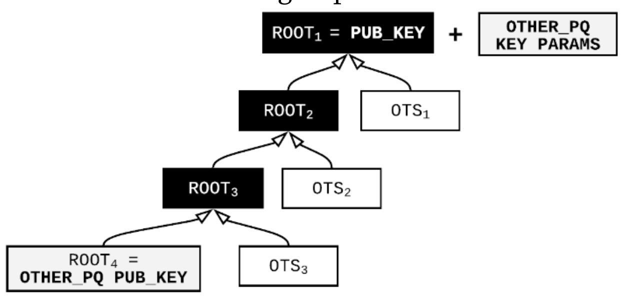

Fig. 4: A versatile BPQS protocol (BPQS-VERS1), with a BPQS- FEW top-level of height 3, in which the last root is the public key ofanotherPQscheme,suchasXMSSMT[44]orSPHINCS+[32].

6.3Combined PQSchemes

As already mentioned, BPQS can fallback to another PQ scheme whenever required. By applying a similar logic, Figure5showsvariouscustommodelsforcombiningmultiplePQ schemes into one. The approach is very simple, but allows for very useful constructions, such as a "Stateful and Stateless" scheme in Figures 5 A and B, or a "Stateful with Stateless Fallback" scheme in Figure 5 C. The latter provides a solution to clustered environments in which multiple nodes require consensus over signature states, but a fallback mechanism is a prerequisite for the system to stay functional if consensus fails for anyreason.

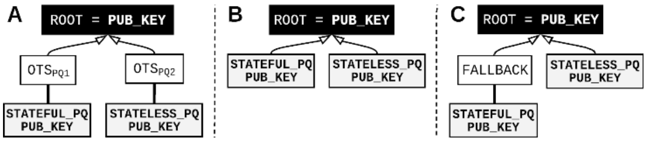

Fig. 5: Various recommended designs using a parallel BPQS logic to combine multiple schemes into one concrete PQ solution. Note that if the underlying schemes require extra parameters, such as bitmasks, these should be published along with the root public key, similarly to Figure4.

The three depicted approaches offer different flexibility when it comes to:

1. Choosing a balance between key generation time and size ofsignature,
2. Deciding whether to allow for picking of the underlying algorithms at a latertime.

Forinstance,optionBrequiresbothPQkeystobegenerated to form the to-be-advertised combined public key, whilst option A is practically a BPQS-EXT that will be used to sign the "upcoming" PQ schemes. Along the same lines, option C is a combination of A and B, but the left PQ scheme is not required to be a-priori selected and computed. Note that one could even combine two different stateful or stateless schemes together, e.g., if needed for compatibility purposes, such as when using the same key in two different blockchains, one supporting the original SPHINCS-256 and the other supporting a variation of it (or its standardised version when this becomesavailable).

VII. CONCLUSION

In this work, we introduced BPQS and its extensions to support {one and few}-time optimised post-quantum signa- tures. We have also presented the security challenges that blockchains and DLTs will soon face and why pure OTS schemes are not recommended as a quantum-resistant replace- ment. As shown, BPQS compares favourably even against conventional non-quantum schemes such as RSA, ECDSA and EdDSA, while it provides more reliable quantum-security estimates because of its rooting in a secure cryptographic hash function.

Among others, the main features of the BPQS protocol are:

- shorter signatures, and faster key generation, signing and verificationtimesthantheXMSS[5]andSPHINCS [23] family PQ protocols when signing for one or few times, which is usually preferred in blockchain systems to preserveanonymity,
- it is computationally comparable to non-quantum schemes. One can take advantage of the easy-to-apply multiple hash-chain WOTS parallelisation and caching to providealmostinstantsigningandfasterverification,
- its extensibility property allows for many-timesignatures, while it can also easily be customised, so it can fallback toanothermany-timeschemeifandwhenrequired,
- when used in blockchain and DLT applications, it can take advantage of the underlying chain/graph structure by referencing a previous transaction, in which the same key is reused. This could effectively mean that each new BPQS signature simply requires the effort of an OTS scheme, because the rest of the signature path to the root isintheledgeralreadyandcanbeomitted,
- it could be used as a building block to implement novel PQ schemes such as a simultaneously "Statefuland Stateless" scheme, which might benefit clustered envi- ronments, where nodes can fallback to stateless schemes when consensus is lost. Additionally, such schemes can be used for forward and backward compatibility purposes or when requiring to reuse a key between two indepen- dent and incompatibleblockchains.

The main drawback of the original BPQS protocol is that the size of its signature output increases linearly with the number of signatures. However, one can mitigate this by using a combined PQ approach or by utilising existing graph struc- tures in blockchain applications. All in all, the customisation, caching and extensibility properties of BPQS make it an ideal candidate for blockchains and it could serve as a bridging protocolbetweenstateless,statefulandotherPQscheme s.

VIII. REFERENCES

[1] P. W. Shor, “Algorithms for quantum computation: discrete logarithms andfactoring”,35thFOCS,pp.124-134,1994.

[2] J. Proos, and C. Zalka, “Shor’s discrete logarithm quantum algorithm for elliptic curves”, Quantum Information & Computation, v.3 i.4,2003.

[3] M. Mosca, “Cybersecurity in an era with quantum computers: will we be ready?”, QCrypt,2015.

[4] “The Quantum Countdown. Quantum Computing And The Future OfSmart Ledger Encryption”, Long Finance, http://longfinance.net/DF/ Quantum_Countdown.pdf, February2018.

[5] J. Buchmann, E. Dahmen, and A. Hülsing, “XMSS – A Practical Forward Secure Signature Scheme Based on Minimal Security Assumptions”,PQCrypto2011:Post-QuantumCryptography,pp.117-129,2011.

[6] J. Kelly, “A Preview of Bristlecone, Google’s New Quantum Proces- sor,” Google Research Blog, https://research.googleblog.com/2018/03/a- preview-of-bristlecone-googles-new.html, March2018.

[7] D-Wave, “Temporal Defense Systems Purchases the First D- Wave 2000Q Quantum Computer”, D-Wave Press Release, https://www.dwavesys.com/press-releases/temporal-defense-systems- purchases-first-d-wave-2000q-quantum-computer, January2017.

[8] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, “Defining and Detecting Quantum Speedup”,Sciencevol.345,issue6195,pp.420-424,July2014.

[9] L. K. Grover, “Afast quantum mechanical algorithm for database search”, STOC,1996.

[10] R. Anderson, and R. Brady, “Why quantum computing is hard-and quantumcryptographyisnotprovablysecure”,arXiv:1301.7351,2013.

[11] “CECPQ1 post-quantum cipher suite,” Wikipedia article, https://en. wikipedia.org/wiki/CECPQ1,2016.

[12] “The Post-Quantum PKI Test server”, http://test-pqpki.com/,2018.

[13] L. Chen, S. Jordan, Y-K. Liu, D. Moody, R. Peralta, R. Perlner, and D. Smith-Tone, “NISTIR 8105 Report on Post-Quantum Cryptography”, NIST,2016.

[14] NIST, “Post-Quantum Cryptography Standardization”,https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization, 2017.

[15] “Quantum Safe Cryptography and Security – An introduction, bene- fits, enablers and challenges”, ETSI, http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf,2015.

[16] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, “Scalable, transparent, and post-quantum secure computational integrity”, IACR CryptologyePrintArchive:Report2018/046,2018.

[17] T. Ruffing, P. Moreno-Sanchez, and A. Kate, “CoinShuffle: Practical DecentralizedCoinMixingforBitcoin”,ESORICS,2014.

[18] P. Waterland, “The QRL Whitepaper”, https://theqrl.org/whitepaper/ QRL_whitepaper.pdf,2011.

[19] A. Hülsing, “W-OTS+ – Shorter Signatures for Hash-Based Signature Schemes”,IACRCryptologyePrintArchive:Report2017/965,2017.

[20] S. Popov, “The Tangle”, https://iota.org/IOTA_Whitepaper.pdf,2017.

[21] “How bad is reusing an address?”,IOTA forum, https://forum.iota.org/ t/how-bad-is-reusing-an-address/1277,2017.

[22] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, “Corda: An Introduc- tion”, https://docs.corda.net/_static/corda-introductory-whitepaper.pdf, 2016.

[23] 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”, EUROCRYPT 2015, pp. 368-397,2015.

[24] L. Lamport, “Constructing digital signatures from a one-way function”, TechnicalReportCSL98,SRIInternational,1979.

[25] R. C. Merkle, “A Digital Signature Based on a Conventional Encryption Function”, CRYPTO 1987 pp. 369-378, 1987.

[26] D. Bleichenbacher, and U. Maurer, “On the efficiency of One-Time Digital Signatures”, ASIACRYPT,1996.

[27] A. Hülsing, J. Rijneveld, and F. Song, “Mitigating Multi-Target Attacks inHash-basedSignatures”,PKC2016pp.387-416,2016.

[28] A. Perrig, “The BiBa one-time signature and broadcast authentication protocol”, 8th ACM Conference on Computer and Communication Security, pp. 28-37,2001.

[29] L. Reyzin, and N. Reyzin, “Better than BiBa: Short One-time Signatures withFastSigningandVerifying”,ACISP2002,pp.144-153,2002.

[30] J.Buchmann, E. Dahmen, E. Klintsevich, K. Okeya, and C. Vuil- laume. “Merkle Signatures with Virtually Unlimited Signature Capac- ity”, ACNS,2007.

[31] P. Kampanakis, and S. Fluhrer, “LMS vs XMSS: Comparion of two Hash-Based Signature Standards”, IACR Cryptology ePrintArchive: Report 2017/349,2017.

[32] D.J.Bernstein,C.Dobraunig,M.Eichlseder,S.Fluhrer,S-L.Gazdag,

[33] A. Hülsing, P. Kampanakis, S. Kölbl,

[34] J-P. Aumasson, and G. Endignoux, “Improving Stateless Hash-Based Signatures”,IACRCryptologyePrintArchive:Report2017/933,2017.

[35] S. Gueron, and N. Mouha “SPHINCS-Simpira: Fast Stateless Hash- based Signatures with Post-quantum Security”, IACR Cryptology ePrint Archive: Report 2017/645,2017.

[36] S. Kölbl, M. Lauridsen, F. Mendel, and C. Rechberger, “Haraka v2 – Efficient Short-Input Hashing for Post-Quantum Applications”, IACRCryptologyePrintArchive:Report2016/098,2016.

[37] Federal Information Processing Standards Publication 180-4, “Secure Hash Standard (SHS)”, Information Technology Laboratory, National Institute of Standards and Technology, March2012.

[38] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, “The KECCAK SHA-3 submission, Version 3”,2011.

[39] M. Amy, O. Di Matteo, V. Gheorghiu, M. Mosca, A. Parent, J. Schanck, “Estimating the Cost of Generic Quantum Pre-image Attacks on SHA-2 and SHA-3”, IACR Cryptology ePrint Archive: Report 2016/992,2016.

[40] D. J. Bernstein, “Cost analysis of hash collisions: Will quantum com-putersmakeSHARCSobsolete?”,SHARCS2009,2009.

[41] “Measurements of hash functions, indexed by machine“, eBACS: EN- CRYPT Benchmarking of Cryptographic Systems, http://bench.cr.yp.to/ results-hash.html, accessed: 21 February2018.

[42] M-J. Saarinen, and J-P. Aumasson, “The BLAKE2 Cryptographic Hash and Message Authentication Code (MAC): IETF RFC 7693.”, Internet Engineering Task Force. DOI: 10.17487/RFC7693, 2015.

[43] “IOTA ERROR: PRIVATE KEY REUSE DETECTED“, IOTA github, https://github.com/iotaledger/wallet/issues/928,2018.

[44] C.Cachin,“ArchitectureoftheHyperledgerBlockchainFabric”,https://www.zurich.ibm.com/dccl/papers/cachin_dccl.pdf, 2016.

[45] A. Hülsing, S-L. Gazdag, D. Butin, and J. Buchmann,“Hash-based Signatures: An Outline for a New Standard”, NIST Workshop on Cybersecurity in a Post-Quantum World, 2015.

[46] D. McGrew, and M. Curcio. “Hash-Based Signatures”, https:// datatracker.ietf.org/doc/draft-mcgrew-hash-sigs, accessed: April2018.

[47] “BPQS library”, https://github.com/corda/bpqs, accessed: April 2018.

[48] “BouncyCastleCryptoAPIs”,v2.1.1,release:1.59,2017.

[49] “EdDSA-Java”, v0.2.0,https://github.com/str4d/ed25519-java,2018.