QCW

Quantum Crypto Wrapper

01 Quantum Inc.

Technical White Paper
V3.3
(September, 2025)

Disclaimer

This White Paper may contain forward-looking statements, including statements regarding the cryptographic technologies (the "Technology") offered by 01 Quantum Inc. (the "Company"). These forward-looking statements involve known and unknown risks, uncertainties and other factors which may cause the actual results, performance or achievements of the Technology to be materially different from any future results, performance or achievements expressed or implied by such forward-looking statements.

A number of factors could cause actual results to differ materially from those in the forward-looking statements, including, but not limited to, rapid changing in the field of computer hardware and software, competition, changes in technology and government policies. In light of the significant uncertainties inherent in the forward-looking statements included herein, the inclusion of such information should not be regarded as a representation by the Company as facts.

The Company believes that the expectations reflected in these forward-looking statements are reasonable; however, no assurance can be given that these expectations will prove to be correct and such forward-looking statements included in this presentation should not be relied upon. In addition, these forward-looking statements relate to the date on which they are made. The Company disclaims any intention or obligation to update or revise any forward-looking statements, whether as a result of new information, future events or otherwise.

This White Paper is for information purposes only. The Company does not guarantee the accuracy of the conclusions reached in this White Paper, and the White Paper is provided "as is" with no representations and warranties, express or implied, whatsoever. All warranties are expressly disclaimed. The Company expressly disclaims all liability for and damages of any kind arising out of the use, reference to, or reliance on any information contained in this White Paper, even if advised of the possibility of such damages. In no event will the Company be liable to any person or entity for any direct, indirect, special or consequential damages for the use of, reference to, or reliance on this White Paper or any of the content contained herein.

Recipients are specifically notified as follows:

No offer of securities: This White Paper does not constitute a prospectus nor offer document of any sort and is not intended to constitute an offer or solicitation of securities or any other investment or other product in any jurisdiction. Nothing in this White Paper is an offer to sell, or the solicitation of an offer to buy, any tokens. The Company is publishing this White Paper solely to receive feedback and comments from the public. If and when the Company and its partners offer for sale any tokens (or a Simple Agreement for Future Tokens), it will do so through definitive offering documents, including a disclosure document and risk factors.

No representations: Nothing in this White Paper should be treated or read as a guarantee or promise of how the Company's business or the tokens will develop or of the utility or value of the tokens. No representations or warranties have been made to the recipient or its advisers as to the accuracy or completeness of the information, statements, opinions or matters (express or implied) arising out of, contained in or derived from this White Paper or any omission from this document or of any other written or oral information or opinions provided now or in the future to any interested party or their advisers. No representation or warranty is given as to the achievement or reasonableness of any plans, future projections or prospects and nothing in this document is or should be relied upon as a promise or representation as to the future. Any statements about future events are based solely on the Company's analysis of the issues described in this White Paper. That analysis may prove to be incorrect. To the fullest extent, all liability for any loss or damage of whatsoever kind (whether foreseeable or not) which may arise from any person acting on any information and opinions contained in this White Paper or any information which is made available in connection with any further enquiries, notwithstanding any negligence, default or lack of care, is disclaimed.

TABLE OF CONTENTS

ABSTRACT

Today, technology has truly become part of our lives. We use the Internet, we all share in the cloud, and we create and store data. It is hard to imagine what the world would be like without technology. The more we rely on technology, the more imperative it is to protect ourselves from cyber-attacks.

To many, cybercrime is a distant event which appears on headline news affecting only the well-known names. However, over the past few years, things have changed dramatically. Recent statistics remind us that cybercrimes are closer than we think: with over 60% of small to medium sized businesses being targeted by cybersecurity attacks and over 80% of customers' data that could be compromised in an attack[1]. This is in addition to cyberattacks on individuals which have largely gone unreported. These statistics and experience reflect the challenge we face as we live through a period of unparalleled digital change embracing digital assets, mobile, Internet of Things, Artificial Intelligence and cloud computing which together result in multi-faceted cyber-attack opportunities. These risks are going to increase as quantum computing becomes more accessible. With a quantum computer's extraordinary computation power, what would take a conventional computer over 150 years to decode, may only take seconds, rendering most encryption obsolete and yet quantum computing is no longer fictitious.

The first line of attack by quantum hackers is likely to be related to financial assets as there are financial gains to be reaped. We believe the lowest hanging fruit for quantum hackers in the financial world will be cryptocurrencies. This is because the distributed nature of its technology that makes it safe in the classical world of computing will be the same factor that makes it vulnerable to quantum attacks. The digital signature of Elliptic Curve Cryptography (ECC)[2] used by virtually all cryptocurrencies (Bitcoin, Ethereum, Solana, HyperLiquid, Avalanche, etc.) are quantum-vulnerable. Security in cryptocurrencies, since its inception, has been relying on 1) a digital signature to guarantee the trustworthiness of the transactions; 2) the private key used to sign a transaction cannot be reverse-engineered back from its public key within the normal life span of a human being. This bedrock of trust will be broken when the underlying cryptographic technology is vulnerable to the hackers equipped with a powerful enough quantum computer.

This White Paper proposes a ground breaking solution to provide crucial quantum security to cryptocurrencies through the use of quantum-safe validation and quantum-safe wallets. Quantum safety solutions can be added to conventional blockchain transactions by executing zero-knowledge proofs to ensure both quantum-safe processing of addresses and signatures as well as the legitimacy of the transactions for incorporation on the chain.

[1] Data obtained from "Small Business Cybersecurity Statistics You Should Know" by strongdm Feb 1, 2024 https://www.strongdm.com/blog/small-business-cyber-security-statistics

[2] https://en.wikipedia.org/wiki/Elliptic-curve_cryptography

INTRODUCTION

This section describes the concepts of this quantum-safe solution in a simplified form that will be further elaborated in the Quantum-Safe Blockchain Technology section.

The security of Distributed Ledger Technology (DLT) such as blockchain networks can be enhanced using quantum-safe (QS) mechanisms. For example, existing blockchains can be secured without altering the core structure or operations of the blockchain by using QS mechanisms for the address and/or signature of one or more endpoints, thereby protecting the integrity of data entering the blockchain.

QS mechanism can be carried out by a conventional validator who acts as the gate keeper making sure transactions are valid between the parties by performing verification of zero-knowledge proofs written by the party who initiates the transaction and eventually causing the transaction to be recorded on a distributed ledger while on the other hand the entity is able to verify the registration of the transaction on the ledger. For example, the witness proving QS transactions may perform QS cryptographic operations which are beyond the scope of current blockchain systems to ensure that the identity of one or more parties have not been spoofed by someone using quantum technology.

The party who initiates the transaction can then provide secure proof of the QS validation to the entity via a secure signature. The secure signature can include both conventional and QS signatures. The ledger registration entity can be a group of blockchain miners, for example, or an entity executing a smart contract using smart contract data provided by the transaction initiator (a.k.a. the payer).

While the payer or the conventional validator can process large QS addresses or QS signatures of parties to provide resistance to quantum computing attacks, an underlying ledger recording the transaction may not need to record the large QS addresses and QS signatures. For example, a blockchain can instead record a hash of a QS address and/or a hash of a QS signature of one or more parties to the transaction. Specifically, to ensure minimum additional storage requirements in existing blockchains, for example, a QS address can be represented in the blockchain by a hash value of the QS address instead of the actual QS address. Similarly, a QS signature can be represented in the blockchain by a hash value of the QS address instead of the actual QS signature. The association between the hash value and the actual data can be guaranteed and verified by the mechanism of zero-knowledge proofs written by the payer and verified by the conventional validator. One example of such witness proofs mechanism is zero-knowledge proof (ZKP).

As described from another angle, the size of a QS address and QS signature are typically much larger than the conventional addresses and conventional signature respectively. Our solution makes use of zero-knowledge proofs to accommodate these large QS addresses and QS signatures to provide resistance to quantum computing attacks without suffering the data size limitation imposed by the conventional blockchain. This minimizes the impact on the underlying ledger when recording the transaction as it may not need to record the large QS addresses.

It is noted that the benefits of using zero-knowledge proofs mechanism can be realized in a wide variety of distributed ledger systems via smart contract mechanisms. Interfaces to existing cryptocurrency and non-fungible token blockchains, for example, can be adapted to take advantage of QS mechanisms without alterations to existing blockchains per se.

Blockchain security and fairness can be enhanced by integrating Quantum-Safe Verifiable Random Numbers (QS-VRNs) into stake consensus mechanisms. This can be achieved through the use of QS cryptographic techniques, such as Quantum Random Number Generator (QRNG), lattice-based, hash-based, or code-based cryptographic primitives, to generate and verify random numbers that are unpredictable and tamper-proof even against quantum adversaries.

For example, a secure seed for a QS-VRNs can be generated using a QS entropy source by a trusted validator. A QS cryptographic function, such as a lattice-based Verifiable Random Function (VRF) or a hash-based Probability Random Number Generator (PRNG) can derive the VRN from the seed. A QS witness proof, e.g., a zero-knowledge proof or lattice-based signature can be created alongside the VRN. The VRN and its proof can then be published on-chain or sent to verifiers.

Any entity can validate the proof using a QS cryptographic algorithm to ensure fairness. A trusted verifier can check the proof using a secure key without revealing sensitive data. If valid, the number is accepted as QS, fair, and tamper-proof.

QS-VRNs can also strengthen Proof-of-Stake (PoS) and leader election mechanisms by replacing traditional pseudorandom functions with QS-VRNs, thereby ensuring that validators and block proposers are selected in a way that remains secure even in the post-quantum world of computing. Furthermore, these verifiable random numbers can be audited on-chain using zero-knowledge proof mechanism such as Zero-Knowledge Proofs (ZKPs) or other cryptographic commitments, providing transparency without compromising security.

This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter. Furthermore, the claimed subject matter is not limited to limitations that solve any or all disadvantages noted in any part of this disclosure.

THE VISION

Our vision is to deliver a post-quantum blockchain that is powered by the best-in-class Post-Quantum Cryptography (PQC) to withstand the power of quantum computers. This allows the world of cryptocurrencies to be protected against the inevitable arrival of Q-Day, the day when the power of commercially available quantum computers surpasses the minimum requirement[3] to break the cryptographic algorithm the Internet is currently relying on, namely RSA (Rivest-Shamir-Adleman) and ECC.

Our mission is to extend the state-of-the-art blockchain reliability into the post-quantum world of computing. Our post-quantum blockchain will not only be secure against the power of quantum computers, but also retain the features, flexibility and performance of its pre-quantum ancestor. Some examples of features are smart contract capability, high transaction-per-second, low gas fee, NFT and Web3 support, etc.

[3] According to cryptanalysis experts, the minimum requirement to break RSA and ECC cryptography are 4000 qubits.

We envision a decentralized, secure and scalable network operated and governed by the community that uses it. The computational resources of this post-quantum blockchain can be scaled up to meet all demands for growth.

While the vision is ambitious, it can be achieved by our profound experience in PQC, blockchain technology, cybersecurity, as well as our innovative team of developers with a combined 150+ years of software development experience. A very successful POC was completed several years ago using the open sources of Solana[4]. To facilitate a commercial launch, we will build a pilot token on the HyperLiquid chain via a HyperEVM compatible QS wallets for users, APIs for various platforms. In particular, this white paper will highlight the novel post-quantum method to warrant quantum-safety on classical computers as well as preventing attacks from both the traditional world of computing as well as the upcoming quantum computers.

After the initial commercial availability, new updated versions of our QS blockchain are expected to be released for continuous improvement in all aspects including, but not limited to, PQC, general security, features, performance, etc.

QUANTUM THREAT

Traditional cryptography such as RSA encryption relies on the computational difficulty of factoring large prime numbers. The security of RSA encryption is based on the fact that it's very difficult and time-consuming for classical computers to factorize a large semi-prime number into its prime factors when the number is sufficiently large (for example, a product of two large prime numbers). This is called a mathematical "trapdoor" that is easy going one way, but extremely difficult the opposite way.

For example, a small semi-prime number of 21 will take a split second even for a human to factor into the original two prime numbers that make it up, which are 3 and 7. Raising the bar slightly to 221 will likely take much longer to factor into the original 13 and 17. A very large semi-prime number will take more than 150 years for supercomputer today to factor into the two original large prime numbers. Since the brute forcing effort is longer than the average life span of a human being, it is considered to be safe and has been the bedrock in protecting the digital world for over 40 years. This protection has been extended to cryptocurrencies which rely on this traditional cryptography.

[4] https://fr.wikipedia.org/wiki/Solana_(blockchain)

Quantum computers have the potential to crack RSA, which is a widely used encryption algorithm in secure communication protocols and systems, due to their ability to perform certain types of calculations much faster than classical computers.

Quantum computers exploit the principles of quantum mechanics to perform certain calculations much more efficiently than classical computers. Most notably, Shor's Algorithm[5], developed by Peter Shor in 1994, demonstrated that a quantum computer could factorize large numbers exponentially faster than the best-known classical algorithms. Shor's Algorithm, which is designed to run on a Quantum computer, is the process of period-finding which is done using Quantum Fourier Transform (QFT). The QFT can be used to determine the period of a function f(x). QFT processing can be done efficiently on a quantum computer because all of the experiments can be run at once in superposition, with bad experiments deteriorating from destructive interference effects and the good experiments dominating from constructive interference effects. Once the period-finding mechanism of the QFT becomes available, it can be exploited to find patterns in the mathematical structure of the number being factored. While not yet a commodity item, quantum computers will at least be available via the cloud in the foreseeable future.

Shor's algorithm exploits two key properties of quantum computing:

1. Super-positioning[6]: Quantum computers can perform multiple calculations simultaneously by leveraging superposition, allowing them to explore multiple potential solutions to a problem in parallel.
2. Entanglement: Quantum computers can use entanglement to correlate the outcomes of different quantum computations, enhancing their computational power.

Using these properties, Shor's algorithm can factorize large numbers into their prime factors much faster than the best classical algorithms known today. As a result, if and when large-scale, fault-tolerant quantum computers become a reality, they could potentially break RSA encryption, which relies on the difficulty of factoring large numbers into their primes.

THE ARRIVAL OF THE QUANTUM ERA

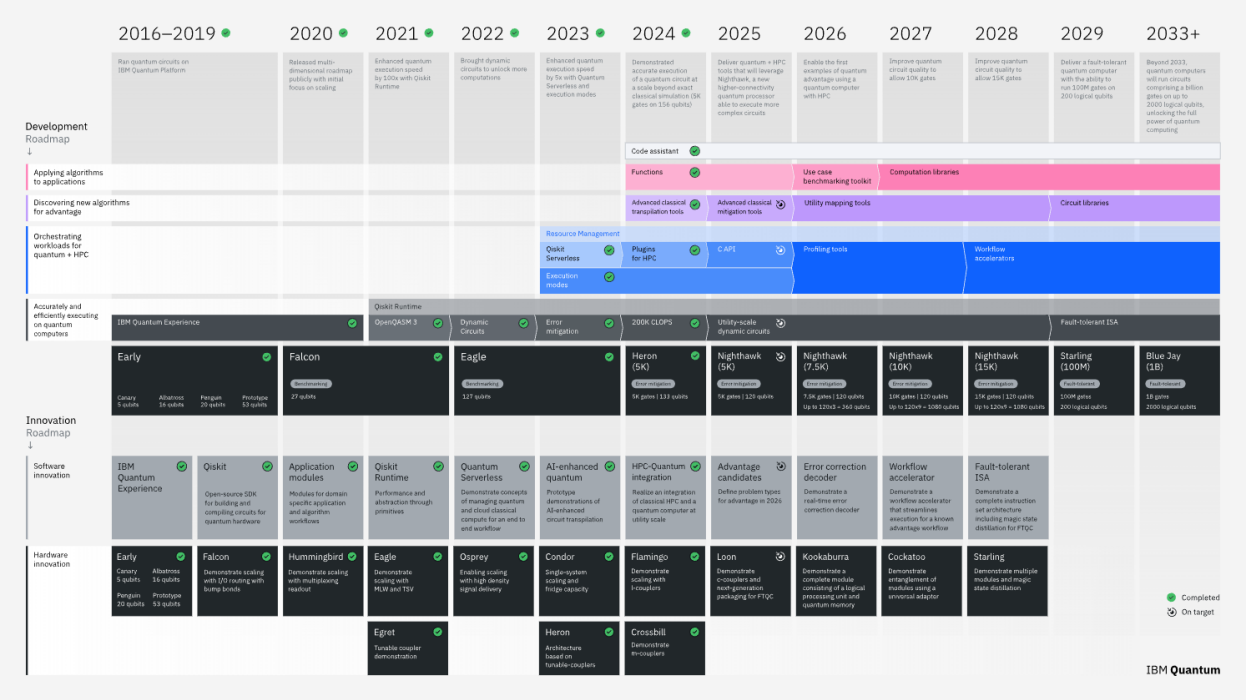
Quantum computers have been in development since the early 1980s. Quantum computing theory was first introduced as a concept by Richard Feynman. There are many quantum computer vendors in the western world, namely IBM, Google, Microsoft, Honeywell, D-Wave, Quantinuum, IonQ, Xanadu, etc. IBM as an example, have a very transparent roadmap[7] of their quantum computer commercial deliveries. They were the first major player to deliver the world's first commercial quantum computer back in 2019. That was a 27 quantum-bit or "qubit", "quantum toy"', similar to the PCs we had in the 1970s. Please note that there is no reliable information of the quantum computer development progress in closed worlds like Russia, China, North Korea, Iran, etc.

[5] https://en.wikipedia.org/wiki/Shor%27s_algorithm

[6] Computers work with information in the form of bits as a "1" or a "0" at any one time. If we send a question to a computer it has to proceed with orderly, linear fashion to find the answer. Quantum computers adopt superposition rules, its bits can be 1 or 0, or 0 and 1 at the same time. In this superposed state, a quantum bit exists as two equally probably possibilities at the same time so when a single quantum bit can be in two states at the same time, it can perform two calculations at the same time. Two quantum bits could perform four simultaneous calculations, three quantum bits could perform eight; and so on… creating exponentially increased calculation power.

[7] https://www.forbes.com/sites/moorinsights/2022/05/18/ibms-newest-quantum-computing-roadmap-unveils-four-new-quantum-processors-and-future-plans-for-a-quantum-supercomputer/?sh=3495394f7ebd

As of the date of this White Paper, IBM had fulfilled their roadmap delivering their 1,121-qubit version in December 2023, breaking the 1,000-qubit barrier. They are expecting to deliver a new version Loon sometime in 2025 followed by Kookaburra in 2026, and eventually a full-scale fault-tolerant version by 2029.



Most importantly, IBM is just one of the many quantum computer vendors in this fierce competition. Google, for example, in December 2024, announced that their latest quantum computing technologies with error correction. This gives quiet qubits which is a major breakthrough in quantum computing technology. In February, 2025 Microsoft announced their "Majorana 1" topological core technology allowing up to 1m qubits to be integrated into one single chip. The combination of Google and Microsoft announcement essentially had put the world's attention on the arrival of Q-Day, the day when commercially available quantum computers have the power to crack RSA and/or ECC.

The power of quantum computing is a game changer for mankind as it allows computations of things that were impossible in the pre-quantum world of computing. Significant advancements will be achieved in the areas of cosmology analysis, chemical reaction simulation, AI machine learning, DNA analysis, etc. Unfortunately, with any game changing invention, there is a dark side. The dark side of quantum computing is its ability to be the destructor of the present, modern asymmetric cryptography. The application of quantum computing to cryptography has become one of the most promising possibilities and has attracted a lot of scientific attention, research[8] and resources. Various simulations and experiments have been conducted to compress the brute force time. With such focus, it is not surprising to see that recent developments on period finding mechanisms is making the quantum application to cryptography more of a reality than a theoretical application.

[8] Vasileios Mavroeidis, Kamer Vishi, Mateusz D. Zych, Audun Jøsang, "The Impact of Quantum Computing on Present Cryptography", March 2018. https://arxiv.org/pdf/1804.00200.pdf

Q-DAY AND ITS PREVENTION

Cryptanalysis expert believes 4,000-qubits is the minimum barrier to cross into Q-Day. As of December 2023, Q-Day has not yet arrived as the commercially available quantum computer manufactured by IBM was only 1,121-qubits. However, the various announcement by quantum computer vendors in early 2025 has accelerated the expectation of Q-Day. More worrisome is that there are many vendors from the closed world such as China and Russia where they do not announce their roadmap and tend to release products on surprise-basis. Let alone the national level technology is usually 3-5 years ahead of the commercial level.

The U.S. National Institute of Standards and Technology (NIST[9]) has been a world leader for PQC research since 2016. NIST is a government agency in the US with the mission to promote American innovation and industrial competitiveness. One of the most important functions of their mission is to study and give non-binding recommendations/standardization for cryptographic algorithms. According to NIST, "regardless of whether we can estimate the exact time of the arrival of the quantum computing era, we must begin now to prepare our information security systems to be able to resist quantum computing". It also recognizes that a large international community has emerged to address the issue of information security in a quantum computing future. Efforts to develop quantum-safe technologies are intensifying, reflecting the urgency and determination for cybersecurity to win in this quantum race.

To counter this threat, researchers are actively working on developing quantum-safe cryptographic algorithms that will remain secure even in the presence of powerful quantum computers. These efforts are aimed at ensuring the security of sensitive information in the era of quantum computing.

NIST endorsement is a crucial factor for public trust as it does not only guarantee effectiveness of the cryptographic algorithms, but also ensures no secret backdoor. Since 2017, NIST has been studying Post-Quantum Cryptography (PQC) with 82 submissions from various universities and study groups globally. As of the date of this paper, NIST has announced 5 PQC recommendations and expected more to come. The race between Q-Day and NIST recommendations indicate, unfortunately, a catastrophe is likely on the horizon. The main reason is that it takes time to implement quantum-safety. Since cryptography is essentially the bedrock of any modern application, it is extremely complicated to change the huge multi-layer of application built on top of the cryptographic bedrock. It can take up to 2-3 years for some complicated environment. Apparently, the rule of thumb indicates that it could be a global cyber disaster if Q-Day arrives before 2028. The problem is compounded in the world of cryptocurrencies because the public key is always available on the public chain which is music to the ears of hackers who perform Harvest-Now-Decrypt-Later (HNDL) attacks.

PQC experts from the Company demonstrated their tremendous foresight and their ability to predict the NIST recommendations many years ahead of NIST's timeline and spearheaded the proof-of-concept (PoC) project for quantum-safe cryptocurrencies several years ago. The PQC technology implemented in the PoC project on Solana was eventually the Federal Information Processing Standards (FIPS-204) initial recommendation published by NIST in August 2023. This early mover advantage has provided the Company with a significant lead while others can only begin their work after NIST has published their final recommendations in late 2024.

[9] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, Daniel Smith-Tone, "Report on Post-Quantum Cryptography made by U.S. National Institute of Standard and Technology", April 2016.
https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.8105.pdf

Q-DAY EFFECTS ON CRYPTOCURRENCIES

There has always been a myth that blockchain is virtually unhackable due to its distributed nature and the consensus mechanism. Cybersecurity experts wrote countless numbers of papers about how to further protect the blockchain mechanism such as strengthening the prevention of an entity or individual getting control over half or more of a network's hashrate ("51% attacks"). Recent studies also include discussions about whether it is a problem if quantum computers, AI machine learning, or a combination of both are being used to perform 51% attacks. We believe while these are all good discussions, people are missing the main point. It is like continuously discussing how to strengthen the lock of a safe, but totally forgetting about an opening at the back of the safe. This "opening" is the ECC digital signature, which is the bedrock of the original blockchain technology proposed by Satoshi Nakamoto[10], whoever that is.

The fundamental concept of cryptocurrencies is based on Public Key Infrastructure (PKI) whereby the public key is the "account name" while the private key is the "authorization". After Q-Day, the private key of a non-quantum-safe crypto wallet can be easily forged by reverse engineering it back from the public key. If the private key can no longer be trusted, the whole world of cryptocurrencies could be totally destroyed because hackers can sign-in and spend the digital money of the victims as well as validate double spending, etc.

From a practical standpoint, after Q-Day, the value of any digital assets remaining in a wallet that is not quantum-safe can potentially be reduced to $0 in a short period of time. This is a huge trillion-dollar crisis that cannot be resolved overnight. Our commercial goal is to provide a quantum-safe version tokens as well as quantum-safe wrapped tokens for an existing token that is quantum-vulnerable. The pilot launch will be a quantum-safe native token on HyperLiquid with an achievable roadmap to extend to the existing cryptocurrencies such as Ethereum, Bitcoin, etc. The Company expects to offer a "quantum crypto harbor" before the arrival of Q-Day, allowing enough time for crypto holders to park their tokens in a quantum-safe wrapped counterpart.

[10] https://en.wikipedia.org/wiki/Satoshi_Nakamoto

QUANTUM-SAFE BLOCKCHAIN TECHNOLOGY

This disclosure relates generally to providing distributed ledger operations with resistance to quantum computing attacks. The techniques described herein can be used, for example, to enhance the security of existing blockchain systems so that these blockchain operations will be safe in from attacks from quantum computers. Techniques are described for operating a scalable and fault-tolerant system for transaction validation that employs post-quantum cryptographic (PQC) technologies in conjunction with ZKP mechanisms). Transactions can be signed using PQC algorithms, making them resistant to emerging quantum-computing threats. Zero-knowledge proof mechanisms allow the payer side to ascertain the correctness of transactions without revealing critical data, thus protecting privacy. A traditional validator leverages these proofs to ensure consistency and validity of the transaction such as the payer's QS signature. As a result, this provides a robust horizontal scaling capability.

The development of blockchain technology has marked a new era in the world of computing. Blockchain involves distributing a database over multiple computers, as opposed to using a single central database. This is also known as Distributed Ledger Technology ("DLT") but is generally being referred to as "blockchain" by the public. DLT takes cyber-security to a new height by requiring that any new block of data proposed for inclusion in the database be not only digitally signed by an authorized node who has proposed the block of data, but also that the new block includes a hash value of the previous block of data. In a certain sense, the resulting chain of data blocks is like how DNA works in humans: the DNA of each new individual has a signature of the parents. The complexity of the linkage makes it extraordinarily unlikely that there will be any doubt as to the continuity of the chain. This makes data entry a one-way street. Each block is a permanent link in the chain. It cannot be removed or edited. Any correction must be in the form of a new, additional block of data.

DLT further requires the chain of data blocks be replicated among numerous computers using a self-correcting mechanism. A consensus among the numerous computers, e.g., a simple majority, is required to legitimize a new block. For malicious activities to be successful, the malicious activity must simultaneously attack many nodes so that a fake transaction appears to be legitimate to a majority of the numerous computers. Otherwise, the "minority fake transaction" will be over-written by the self-correcting mechanism of the DLT.

Today most DLT/blockchain networks use traditional cryptography, such as elliptic curves and hashing mechanisms, to protect the integrity of the transactions. This is true for most permissioned and permissionless blockchains, and for cryptocurrencies such as Bitcoin, Ethereum, Solana, HyperLiquid, etc. The combination of blockchain consensus innovation and traditional cryptography is often considered to provide most secure platform for cyber-security that is practically feasible. However, concern has been raised by the prospect of the ability quantum computers which can be able to break traditional cryptography such as elliptic curve protections.

For example, while other mechanisms in DLT may not be immediately threatened by the power of quantum computers, the digital signature part is theoretically vulnerable to quantum computing attack. DLT normally uses asymmetric cryptography such as elliptic curve to sign a transaction for the party who initiates a transaction. This is to guarantee the authenticity of the initiating party. This is like the traditional financial transactions where someone needs to sign a paper in front of the bank officials to initiate a money transfer. The initiator of a transaction at a bank can be asked to present government-issued identification to the bank officials. In addition, the validators who witness the transaction, e.g., the

bank officials, need to sign and guarantee the transaction before the transaction can be added into the official ledger. In DLT today, offers of identification and signatures are done electronically, and the official ledger is a distributed electronic ledger which is confirmed by consensus among the numerous computing nodes receiving copies of the distributed ledger.

Unfortunately, the security of DLT can break down in a post-quantum world of computing when a digital signature cannot be trusted, e.g., because a signature using traditional cryptography such as elliptic curve could easily be forged using a quantum computer. In general, asymmetric cryptography techniques such as elliptic curve use a public key and private key pair for encryption/decryption and signature/verification process. The public key is used to encrypt, and the private key is used to decrypt. The private key is used to sign, and the public key is used to verify. In DLT, e.g., for cryptocurrencies, when person Alice wants to send some "coins" to Bob, Alice creates a transaction record and signs the record using Alice's private key. Then a validator/miner verifies Alice's signature using Alice's public key. The validator/miner can also check other constraints, e.g., making sure that Alice has enough "coins" to fulfill the transaction. Then the validator signs the transaction, e.g., using a key of the validator, and adds the transaction to the blockchain.

What if the private key of Alice or the key of the validator could be forged? If so, a malicious user could, for example, impersonate Alice by creating a transaction to transfer "coins" from A to the malicious user's address. The integrity of the chain is then destroyed by introducing falsified information from an endpoint. The blockchain record would still be permanent, but it would contain false information.

In the classical world of computing, it is virtually impossible to forge the private key. The word "virtually" is used because in cryptography there is no such thing as being absolutely uncrackable. The strength of a cryptography pertains to, in practical terms, how long it would take to find a solution by "brute force" by trying all possible combinations. Today, it takes over 150 years for a traditional supercomputer to "brute force" reverse engineer the private key from the public key. Therefore, the use of private keys and public keys is currently considered to be safe because the "brute force" time required is longer than the average lifespan of a human.

However, in theory, quantum computers have the potential to disrupt this scenario. Quantum computing is a mechanism originally proposed by scientists such as Paul Benioff and Richard Feynman in the early 1980s. It is based on quantum-mechanical phenomena such as superposition and entanglement so that computational steps can be carried out simultaneously, rather than sequentially as done on traditional digital computers. Over the years, several algorithms have been accepted as being capable of cracking the private and public key relationship by having the ability to reverse engineer the private key back from the public key.

In one of the most well-known examples, in 1994 Peter Shor showed that theoretically a quantum computer (if someone can ever successfully build one) would be able to factor large number in polynomial time. Therefore, it would possibly break the public/private key mechanism. Shor's Algorithm is designed to run on a quantum computer. Basically, Shor's Algorithm is a process of period-finding, which is done by the Quantum Fourier Transform (QFT), which takes some function f(x) and figures out the period of the function. QFT can be done efficiently on a quantum computer because it can have all the experiments running at once in superposition, with bad experiments deteriorating from destructive interference effects and the good experiments dominating from constructive interference effects. The rest of Shor's Algorithm is entirely a classical algorithm. Once we have the period-finding mechanism of the QFT, we can exploit it to find patterns in the mathematical structure of the number we are trying to factor.

In recent years, advancement in science has allowed the development of quantum computers. There are many quantum computer vendors such as, but not limited to, IBM, Google, Honeywell, DWave, etc. For example, in December 2023, IBM broke the 1000 qubits barrier and released their 1121 qubit version. In December 2024, Google announced their 105 qubits version with effective error correction. Although it can still be years or decades before they become commodity items, some level of quantum computer is already available via the cloud today.

As discussed above, the weakest link in DLT when facing the threat of quantum computers is the digital signature. In other words, to make DLT become quantum-safe, the most imminent requirement is to replace the digital signing mechanism with a post-quantum cryptography, i.e., a QS mechanism, which ensures reliability of the signatures of the transaction initiators as well as signatures of the validators.

Today, several post-quantum cryptography algorithms are available. Any one of these, or their equivalent, are suitable for demonstration purpose. In this disclosure, we describe, inter alia, how to apply post-quantum cryptography algorithms to an existing DLT to achieve quantum safety.

The quantum-safe concern today is addressed in the world of post-quantum cryptography studies. In general, it is believed that at some point all the blockchains will be replaced and new blocks and transactions will be formed using a post-quantum cryptography algorithm via a "hard-fork," i.e., a sudden shift to the use of quantum-safe mechanisms for the chain itself. In such a scenario, all the post-fork transactions will become quantum-safe, but all the pre-fork unconsumed transactions will be vulnerable.

However, these approaches have limitations. Some of the biggest problems of post-quantum cryptography are the sizes of key pairs and the sizes of signatures. They are typically 20x-30x the size of traditional cryptography such as that of elliptic curve. In practice, this means that QS DLT will consume more storage. Using cryptocurrency as an example, when a user tries to initiate a transaction, a fee needs to be included to put the transaction into the distributed ledger (the chain). The amount of fee depends on the complexity of the transaction and, more importantly, how much data the user wants to store within the blockchain. In other words, if we build a post-quantum blockchain by simply replacing its crypto by a post-quantum cryptography algorithm, every transaction will potentially cost 20x-30x more. This effectively raises the practical aspect of usability of such post-quantum blockchain.

This is in addition to the technical limitation (if any) a DLT can have imposed in its internal structure such as transaction size, address size, etc. The address in a blockchain can be incompatible to accommodate the size of the address in the post-quantum cryptography algorithm. A key challenge addressed in this disclosure is how to implement quantum-safety in DLT without suffering any substantial size limitation, fee inflation, internal structural limitation, or degradation in throughput.

EXAMPLE SOLUTIONS

Herein, examples are non-limiting, being intended to provide sufficient guidance to the types of solutions made possible by the techniques described below, as will be appreciated by the practitioner of ordinary skill of the art. For the sake of brevity, examples herein generally assume a two-party transaction between a payer and a payee involving of an amount of currency, such as with the payment of cyber coins or other tokens, where the parties wish the transaction to be documented and/or effected via recording the transaction on a public global blockchain. However, it will be appreciated that the techniques described herein are generally applicable to a wide variety of transactions, such as transactions and contracts involving a single party, two parties, or multiple parties, that pertain to any type of assets to be altered and/or exchanged subject to wide variety of terms and conditions. The payment of electronic coins or other tokens is merely a simple and perhaps most common example. Similarly, transactions can be recorded on a variety of distributed ledgers such as public blockchains, private blockchains, and other cryptographic ledger technologies.

Herein a distinction is generally drawn between newer quantum-safe (QS) encryption technologies versus conventional encryption technologies used in the past, such as public-private key pairs and blockchain blockhashes. Techniques are described to efficiently leverage a mixture of QS technology and conventional technology to provide QS security to systems without the use of QS computations at every step. Like the separation of computationally intense blockhash generation from less cumbersome verification of resulting blockhashes, here various hashes, zero-knowledge proofs, and Merkle trees are used to provide non-quantum system components with the benefits of separate computationally intense QS operations.

Zero-knowledge proofs can be used in several ways. For example, they can serve as gateways and/or computational services so that endpoints themselves do not need to process large QS public keys and QS signatures. The use and verification of zero-knowledge proofs can be facilitated via smart contracts. Many well-known smart contract formats are in use today. New smart contract formats can be augmented for quantum safety in several ways.

A smart contract is a custom program that exists on a blockchain. When a smart contract is being deployed, it is deployed along with the hash of the program itself. When a blockchain participant (e.g., a user) wants to submit a new transaction, the transaction record typically consists of: (1) the hash of the smart contract program, whereby the actual smart contract code can be resolved from the initialization block when the contract was deployed; (2) a method name; (3) input data; and (4) output data. One of the tasks of a blockchain validator/miner is to take the hash, method, and input date and run the methods to check that the result is identical to the output for a transaction to be valid. Smart contracts are deterministic. The main requirement is that the smart contract execution should be independent of the hardware to produce the same results. Practically, this is normally achieved by using a Virtual Machine, e.g., HyperEVM in HyperLiquid and BPF in Solana. As a result, smart contracts can strictly limit how developers implement them, such as requiring the use of custom libraries, custom cryptography, etc. Further, smart contracts can be executed via mechanisms with limited computational and/or memory resources.

QUANTUM-SAFE TRANSACTION OVERVIEW

Figure 1 illustrates an example architecture incorporating several useful features which can be used together, separately, or in a variety of sub-combinations. In the example of Figure 1, a smart contract paradigm is used to coordinate the data shared, the constraints tested, and the security employed by transaction originators, smart contract executors, miners, and/or blockchains. It will be appreciated that similar data can be exchanged among entities, like those illustrated in Figure 1, without the use of smart contracts. However, smart contracts are a convenient way to structure information destined for recordation on a blockchain, and for regulating operations performed on the data to ensure, e.g., minimum standards for verification and cryptological security. With or without smart contracts, quantum safety can be integrated into conventional blockchain endpoints without burdening the endpoints with QS computations or burdening the blockchain with large QS data sizes.

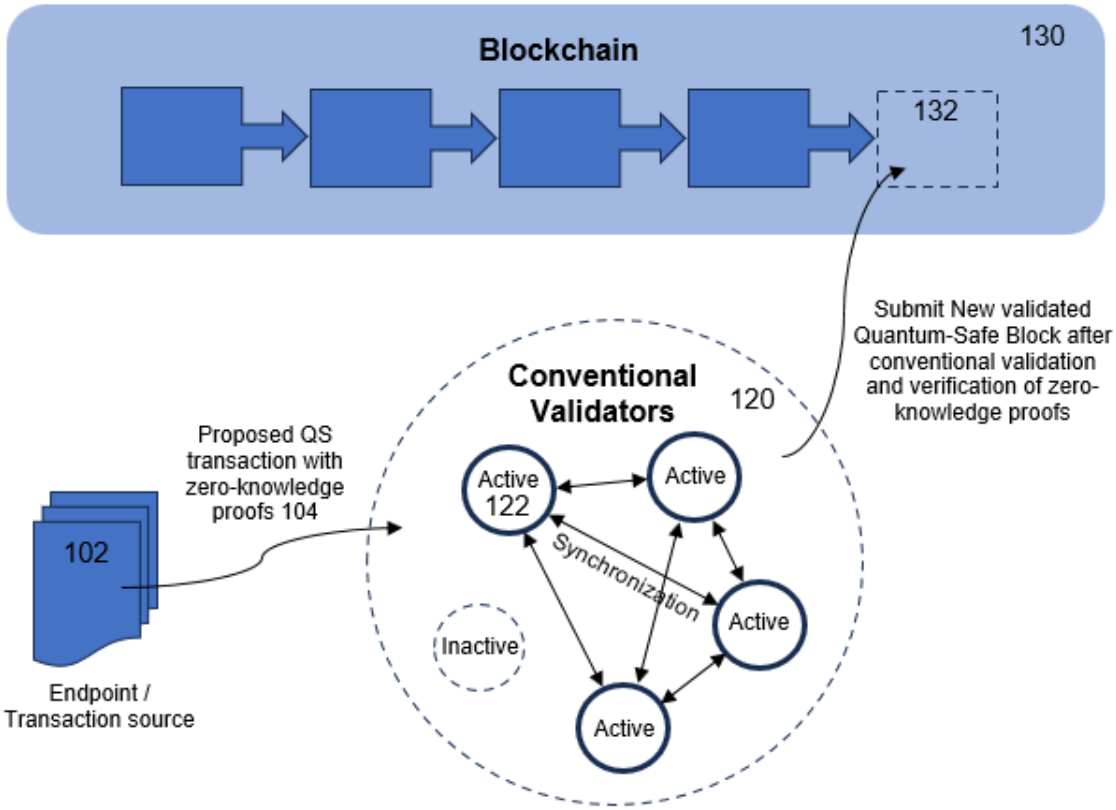


Figure 1

In Figure 1 an endpoint 102, e.g., a personal device used by a cryptocurrency payer, proposes a transaction 104 in accordance with a smart contract deployed on the blockchain 130. Endpoint 102 writes a number of zero-knowledge proofs attesting multiple important aspects of the transaction. These include: (1) proof that the QS signature of the payer has been verified; (2) a QS address (e.g., a hash value) of the payer is correctly associated with the QS public key of the initiating entity, (e.g., a payer); (3) proof that the QS signatures of the payer is correctly associated with the hash written in the transaction; (4) proof that the amount/nature of the item being transferred is valid; (5) a Merkle tree for a block including the transaction and other transactions is constructed properly with a publicly visible root.

For a transaction to be recorded the blockchain, a payer and a payee will normally be identified by a conventional address on the blockchain. For quantum safety, each of the payer and payee will additionally each get a QS public key. As compared to conventional public keys, these can be very large and therefore unwieldy for storage directly onto the conventional blockchain. For all QS operations described herein, a hash of a QS public key of each party can serve as a QS address for the party, and a transaction can bear a QS signature of a party that is created by, for example, encrypting the transaction with the QS private key of the party.

Within the network of validators 120, an active validator 122 receives the proposed transaction 104. The smart contract transaction data is then interpreted and executed by the validator 122 based on the record 114, the requirements of the of smart contract, and/or other information available to the validator 122. The work done by smart contract executor 122 includes processing information related to the quantum safety measures. This work can be much less computationally intensive than the work done by the endpoint 104, and operate on much smaller pieces of information. Confirming the zero-knowledge proofs, for example, can require much less effort than was required to create the zero-knowledge proofs. The manner in which the zero-knowledge proofs and/or other aspects of the contract are verified at the validator 122 can be controlled by information contained in the smart contract transaction data itself and/or other known to the validator 122 for processing contracts of this type.

Advantages of the approach illustrated in Figure 1 can be understood when considered in contrast to other approaches. For example, an alternative is to augment smart contract transaction data themselves with QS signatures. However, such signatures are not supported natively by today's blockchains. QS signatures are significantly larger than conventional cryptographic signatures. Most blockchains limit the transaction size, e.g., to 1KB. QS signatures simply do not fit in such a limited transaction record size. Hence, quantum safety cannot be provided to convention blockchains by migrating to the larger key pairs with the size needed to defend against quantum attack. Instead, mechanism of zero-knowledge proof is being used to avoid the need to actually storing the QS signatures and QS public key in smart contracts. The zero-knowledge proof which is a small data (e.g. 256 bytes) is being stored in every transaction before submitting to the validators/miners of the blockchain.

The quantum-safe operations described herein can be achieved in several ways. In all the mentioned QS operations, hash values can be used to represent the original QS public keys, and the original QS signatures used in smart contracts. Many hashing algorithms, such as SHA256 and SHA512 have already been proven to be quantum-safe. So quantum-safe security can be retained.

Zero-knowledge proof mechanisms can be used to attest to a number of important aspects of the transaction such as, but not limited to: (a) correctly associate a QS address with the QS public key of a party; and (b) correctly associate the hash value of a QS signature with an original QS signature, for example, without having to query or looking up in any table/database.

Signatures can be handled in a number of ways. For example, in Figure 1 the entity that initiates the transaction, endpoint 102, creates an initial QS transaction 104 that includes a QS signature of the entity that can be used to authenticate the transaction. The actual QS signature and its hash, as well as the associated actual QS public key and its hash, are included in the transaction data 104. The endpoint 102 writes a number of zero-knowledge proofs attesting multiple important aspects of the transaction. These include: (1) proof that the QS signature of the payer has been verified; (2) a QS address (e.g., a hash value) of the payer is correctly associated with the QS public key of the initiating entity, (e.g., a payer); (3) proof that the QS signatures of the payer is correctly associated with the hash written in the transaction; (4) proof that the amount/nature of the item being transferred is valid; (5) a Merkle tree for a block including the transaction and other transactions is constructed properly with a publicly visible root.

As previously described, the payer 102 passes the QS transaction 104 together with the zero-knowledge proofs to the validators 120. The validators 120 perform the conventional process of validating the transaction and verifying the zero-knowledge proofs before submitting as a new block 132 of the underlying blockchain 130. In other words, if the verification of the zero-knowledge proofs has failed, the transaction 104 will be discarded.

QUANTUM-SAFE TRANSACTION FLOW

Figure 2 is a call flow of an example of a QS process for registering a transaction in a conventional blockchain using zero-knowledge proofs and a smart contract. As discussed before, a smart contract is not strictly required but is convenient. For Figure 2, the architecture of Figure 1 can be assumed for the establishment of the smart contract on the chain.

In Figure 2, a payer 202 wants to pay an amount to a payee 201. At 210 the payer 202 receives conventional address and an optional QS address identifying the payee 201.

At 212, the payer 202 creates a transaction for the payment in the form of a populated transaction data. In practice, there are many options for what to include in the transaction data for a proposed QS transaction. However, in this example it is assumed that, for acceptance to enter into the underlying conventional blockchain, the transaction data must contain at least conventional addresses and a conventional signature of the payer. Further, certain QS information should be included in the blockchain record. The QS information that will be contained in the blockchain record should be in compact hashed form. Therefore, for purposes of the present example, it is assumed that such transaction data for the proposed transaction presented to the conventional validator by the originating endpoint, here payer 202, will include that information or the derivative of them.

In Figure 2, at 212 the payer 202 puts the conventional and optional QS address of the payee 201 into the transaction data. The payer 202 also indicates in the transaction data the nature, terms, and conditions, if any are required. For example, the transaction in Figure 2 is a simple payment of an amount of a cryptocurrency. Alternatively, the transaction could pertain to the transfer of a different asset, or the change of status of an asset, right, or privilege, for example.

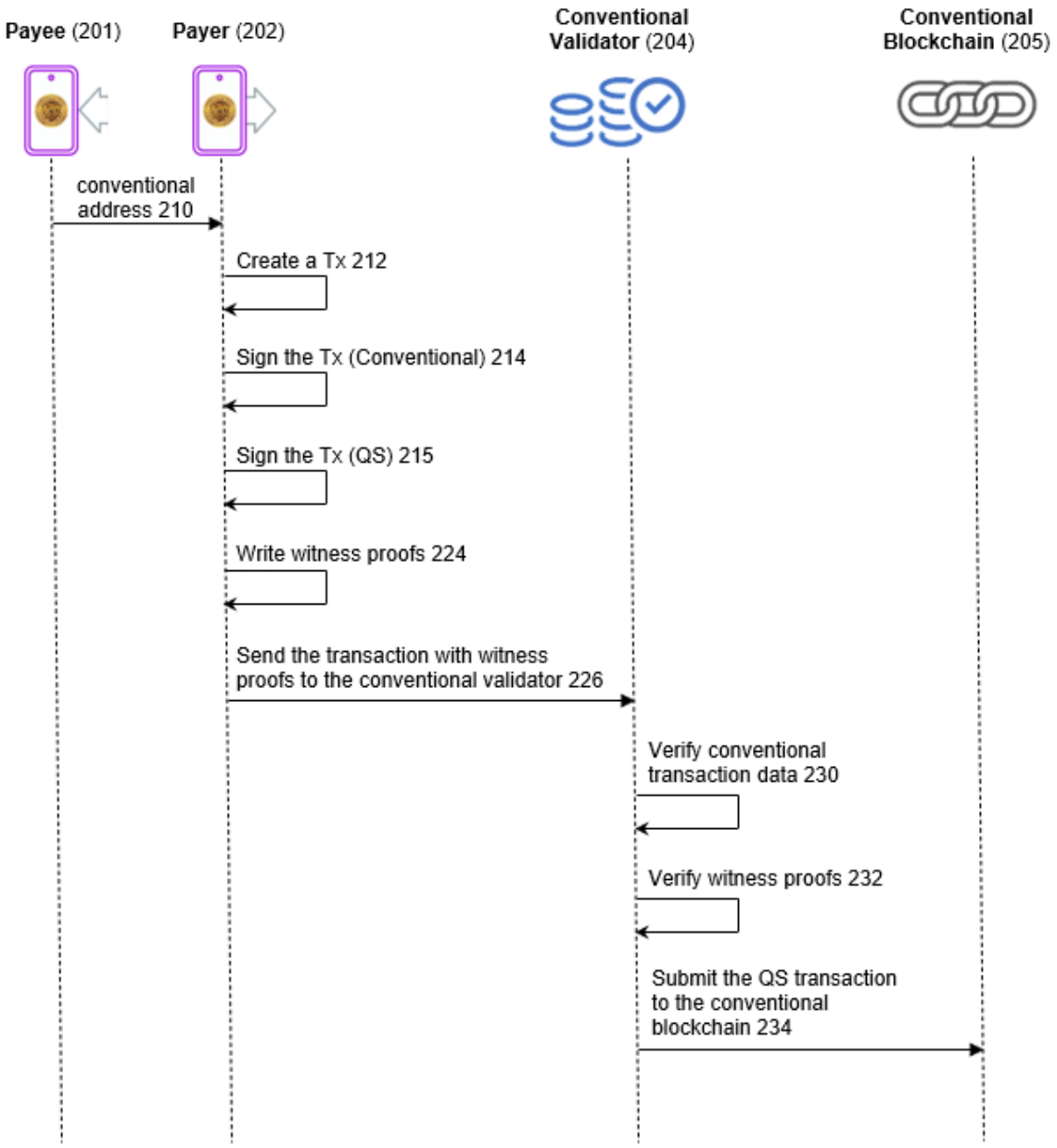


Figure 2

At the 214, the payer 202 signs the transaction with the conventional signature 214 as well as the QS signatures 215 of the payer 202. In addition, the payer 202 also writes a number of zero-knowledge proofs 224 attesting to multiple important aspects of the transaction such as, but not limited to, proofs that: (1) the QS signature of the payer 202 has been verified; the (2) the QS address of the payer 202 (which here is the hash value of the QS public key of the payer) 202 is correctly associated with the QS public key of the payer 202; (3) QS signatures of the payer 202 is correctly associated with the hash written in the transaction; (4) the amount being transferred and/or other transaction terms are valid. Further, (5) a

Merkle tree for a block of multiple transactions, wherein the Merkle tree is constructed properly with a publicly visible root. The result can be smart contract transaction data like the data record described herein in reference to Figure 3.

The payer then sends the smart contract transaction data 226 to a conventional validator 204, which at 230 verifies the conventional transaction information, and at 232 verifies the QS zero-knowledge proofs included in the information arriving at 226. At 234, if the validator 204 is satisfied that the proposed transaction meets all necessary constraints, e.g., as to QS security, authenticity, funds availability, etc. as well as successfully verifying the zero-knowledge proofs written in 224, the validator 204 will submit the QS transaction to the conventional blockchain 234.

For the processing at 232, the conventional validator 204 does not need to perform that same kinds of computations that are performed at 224 by the payer 202. Rather, in Figure 2 the smart contract is intended to be executed via mechanisms with limited computational and/or memory resources. Since the conventional validator 204 needs only to check the proofs, rather than creating them, the specific QS cryptography being utilized in the flow can be unknown to not only the smart contracts but also the conventional validator 204. With the proofs, even with limited resources, the conventional validator 204 can confirm the legitimacy of the transaction.

CRITICAL TRANSACTION DATA

Figure 3 illustrates an example record 302 of data for a proposed transaction to be created by the Payer prior to submission to a conventional validator for validation and entry of the transaction onto a conventional blockchain. In practice, the content of record can vary greatly as required by conventional and/or QS terms stipulated by a smart contract registered on the conventional blockchain for registering a type of QS secured transaction, and/or terms enumerated in the proposed transaction. In Figure 3, record 302 includes a set 310 of conventional transaction data, a set 340 of QS transaction data, and a set 360 of zero-knowledge proofs.

The conventional transaction data 310 includes basic transaction data 312, which can include header items like a memo description of the transaction, parties, the asset at issue, an amount, and terms. The conventional transaction data 310 also includes a conventional address 314 of the payer, a conventional address 316 of the payee, and a conventional signature 318 of the payer for the proposed transaction. The conventional payer address 314 can be a conventional public key associated with the payer or a hash value of the payer's public key. Similarly, the conventional payee address 316 can be a conventional public key associated with the payee or a hash value of the payee's public key.

The conventional transaction data 310 also includes a blockhash 330 identifying a prior transaction on the blockchain to which the present proposed transaction is related. For example, block hash 330 can identify a smart contract under which the proposed transaction is to be governed.

QS transaction data 340 that will be used in the proposed transaction is included in record 302. The payer and the payee each have a conventional address and an optional QS address. The conventional payer 314 and payee addresses 316 are included in the conventional transaction data 310. The hash value of the payer's QS address 342 will be stored in the QS transaction data 340. In addition, the hash value of the payer's QS signature 344 will also be stored in the QS transaction data 340.

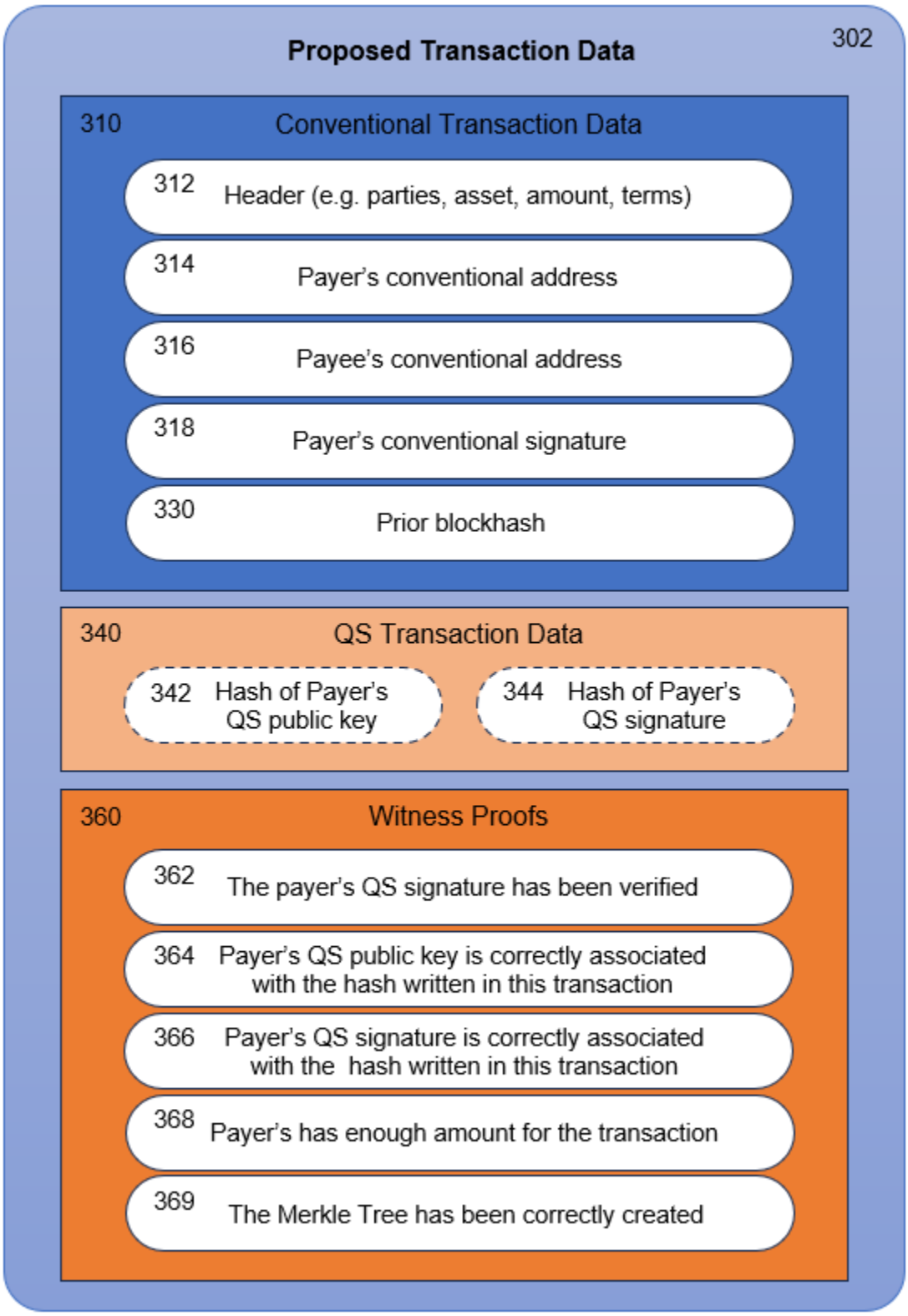

Proposed Transaction Data
302
310
Conventional Transaction Data
312
Header (e.g. parties, asset, amount, terms)
314
Payer's conventional address
316
Payee's conventional address
318
Payer's conventional signature
330
Prior blockhash
340
QS Transaction Data
342
Hash of Payer's QS public key
344
Hash of Payer's QS signature
360
Witness Proofs
362
The payer's QS signature has been verified
364
Payer's QS public key is correctly associated with the hash written in this transaction
366
Payer's QS signature is correctly associated with the hash written in this transaction
368
Payer's has enough amount for the transaction
369
The Merkle Tree has been correctly created

Figure 3

Optionally QS addresses that are hash value addresses of the QS public keys can be used to avoid burdening conventional operations with processing the QS public keys, which are very large compared to conventional addresses. The QS transaction data 340 can even include optional quantum-safety instructions. For example, data received from a payer device for a proposed transaction can include an explicit reference to a smart contract to be used in registering the transaction on the blockchain, e.g., in addition to, or instead of, providing a block hash 330 that identifies a governing smart contract. Special instructions can be provided, for example, on how to validate the transaction at the smart contract level, for processing of the QS signatures, and for verifying the validity of the transaction by confirming the various zero-knowledge proofs to ensure that everything is valid. The instructions can include information for smart contract processing whereby, for example, the smart contract type can be used for non-quantum-safe contracts as well. In this example, the QS transaction data 340 are all coming from the payer.

Similarly, different items of information may be included in different quantum-safe smart contracts depending on which information is desired to be available on the chain. For example, it may suffice to identify the payer only by the conventional address on the chain, rather than including the quantum-safe address of the payer. The reader will appreciate that Figure 3 illustrates one of the many ways to apply the teachings herein for using a smart contract to work with QS transactions in a conventional blockchain.

In the example of Figure 3, various zero-knowledge proofs are written using the appropriate witness key. A typical example of the witness proving mechanism is via zero-knowledge proof (ZKP). These zero-knowledge proofs include: (1) proof that the QS signature of the payer has been verified 362; (2) a QS address (e.g., a hash value) of the payer is correctly associated with the QS public key of the initiating entity 364, (e.g., a payer); (3) proof that the QS signatures of the payer is correctly associated with the hash written in the transaction 366; (4) proof that the amount/nature of the item being transferred is valid 368; (5) a Merkle tree for a block including the transaction and other transactions is constructed properly with a publicly visible root 369.

The inclusion of proof that the amount/nature of the item being transferred is valid is for the purpose of eliminating race attacks whereby 2 conflicting transactions are proposed at virtually the same time for malicious purposes. The whole zero-knowledge proofs are to be verified by the conventional validator during the validation process in addition to the validation of all the conventional items.

A race attack in general refers to a situation where a malicious actor attempts to double-spend or overspend by rapidly submitting multiple transactions in parallel, all signed with the same Falcon key, targeting the same account or balance. The goal is to get two (or more) conflicting transactions accepted by the network before the system can recognize the state change from the first one.

A typical example act of race attack is as followed: The attacker prepares two transactions both signed by a valid QS signature from the same wallet while both attempting to spend from the same wallet balance. These 2 transactions are submitted simultaneously (or with minimal delay) to different validators, or even to the same validator in quick. If there is no mechanism tying each transaction to the correct prior state, both can potentially be accepted into mempool, and one may even get processed depending on timing and load — especially in a speculative or parallel-execution model. This creates a risk where both transactions appear valid independently, but only one should actually be accepted based on the true state. If the zero-knowledge proof does not explicitly validate that the input balance is correct and sufficient at the time of signing, and it was tied to a known state (e.g., a Merkle root or block state hash).

Then the proof can appear valid even if the balance has already been spent by a parallel transaction. In essence, the zero-knowledge proofs become stateless — and the validator can't distinguish whether the balance has already been consumed when processing the second transaction.

The zero-knowledge proof generated by the payer in our mechanism includes the expected state root (e.g., Merkle root of the balance tree or a balance commitment), which is a proof that the input amount is sufficient. Optionally, a commitment to the prior blockhash or account version. This means the conventional validator will only accept the zero-knowledge proof if the proof can be verified cryptographically and the current state matches the prior state committed in the proof. Practically, if transaction #1 is accepted and the balance is reduced, then the state root changes. Then transaction #2's zero-knowledge proof will fail to get verified because the balance no longer matches the witness it is proving against. This creates zero-knowledge proofs enforced consistency, effectively turning the zero-knowledge proof into a state-aware guardrail.

In other words, including amount validity in the zero-knowledge proof ensures that only the first transaction matching the true state can succeed — all others will zero-knowledge proof verification will fail because they're no longer proving against the correct state.

ROLL-UP MECHANISM

Throughput in a blockchain can be significantly improved using a roll-up mechanism. While the roll-up mechanism is NOT the existing feature of the conventional validator, it can be an important upgrade feature to be adopted by a conventional validator to reduce on-chain writes and improve throughput of the overall chain operation significantly. Figure 4 illustrates how the QS mechanism described here does not prevent a conventional validation mechanism to include roll-up by batching multiple QS transactions and commits a single Merkle root representing the updated state. Each transaction is still signed individually with the QS signature by the payer for authenticity followed by writing the zero-knowledge proofs to guarantee the accuracy, correctness, validity of the QS transaction.

Logistically, a conventional validator can process multiple QS transactions and verifies their zero-knowledge proofs, and produces a single state transition proof (e.g., a Merkle root) that is then submitted to the underlying blockchain a summarized result.

Figure 4 illustrates an example set of data processed by a conventional validator for recording on a set of transactions on a conventional blockchain. In Figure 4, the conventional validator creates a record 402 pertaining to a group of QS validated transactions 302^1 through 302^n. The conventional validator verifies the zero-knowledge proof of each individual transaction and batch all those successful ones within the same transaction block.

In practice, a conventional validator adopted this roll-up mechanism may group any number of transactions satisfying the zero-knowledge proofing mechanism onward for conventional validation and recording on a blockchain.

In the case that multiple transactions 302^1 through 302^n sent together in a bundle like record 402, common data can be kept separately in each of records 302^1 through 302^n, or consolidated as economies permit. For example, if all the transactions pertain to a single payer, payee, and/or prior blockhash, that

information can be stored in record 402 more compactly, rather than enumerated separately in each of 302^1 through 302^n.

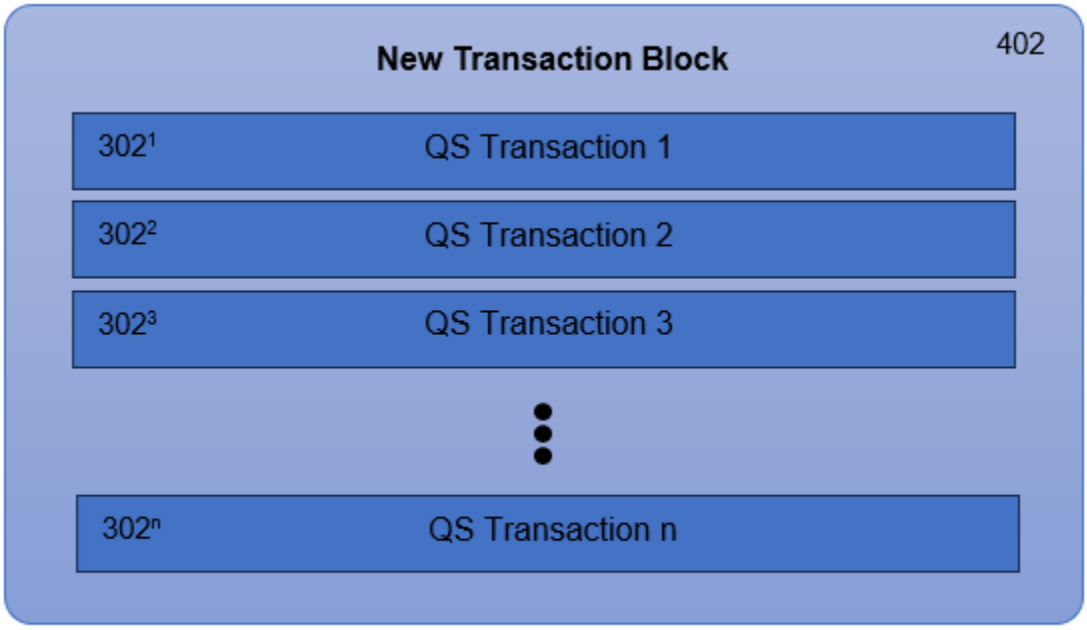


Figure 4

Similar to what is mentioned earlier in this paper, for clarity, in addition to the zero-knowledge proofs provided for each transaction within records 302^1 through 302^n it should also include, but not limited to, proof that the QS signature of the payer has been verified; proof that the QS public key of the payer is correctly associated with the corresponding hash value written in the transaction; proof that the QS signature of the payer has been verified; and a proof that the terms of transaction have been verified (e.g. as to the asset been transferred, the amount, availability of funds, etc.); and proof that the Merkle tree created is constructed properly with a publicly visible root for recording the block of transactions in a next block on the underlying blockchain.

OVERALL FLOWCHART

Figure 5 is a flow chart of an example process incorporating security features described in relation to Figures 1-4. In Figure 5 a QS transaction is processed by a payer which creates zero-knowledge proofs of QS validations. The zero-knowledge proofs enable smart contracts operating a conventional blockchain to confirm QS security and validity of the transaction.

At step 502 of Figure 5, a payer obtains from a payee a conventional address of the payee and an optional QS address of the payee, where the QS address of the payee is a hash of a QS public key of the payee. Typically, when a new transaction is created, e.g., when the payer wants to pay (or transfer) something to the payee, the payee informs the payer its conventional address and an optionally QS address. In practice, the payer can obtain the necessary addresses in several ways. It may not be required for a

particular smart contract, for example, to obtain a QS signature of the payee, or even to record the optional QS address of the payee on a conventional blockchain. Nonetheless, for added security, QS addresses and/or QS signatures of all parties should be preferred or required by the smart contract.

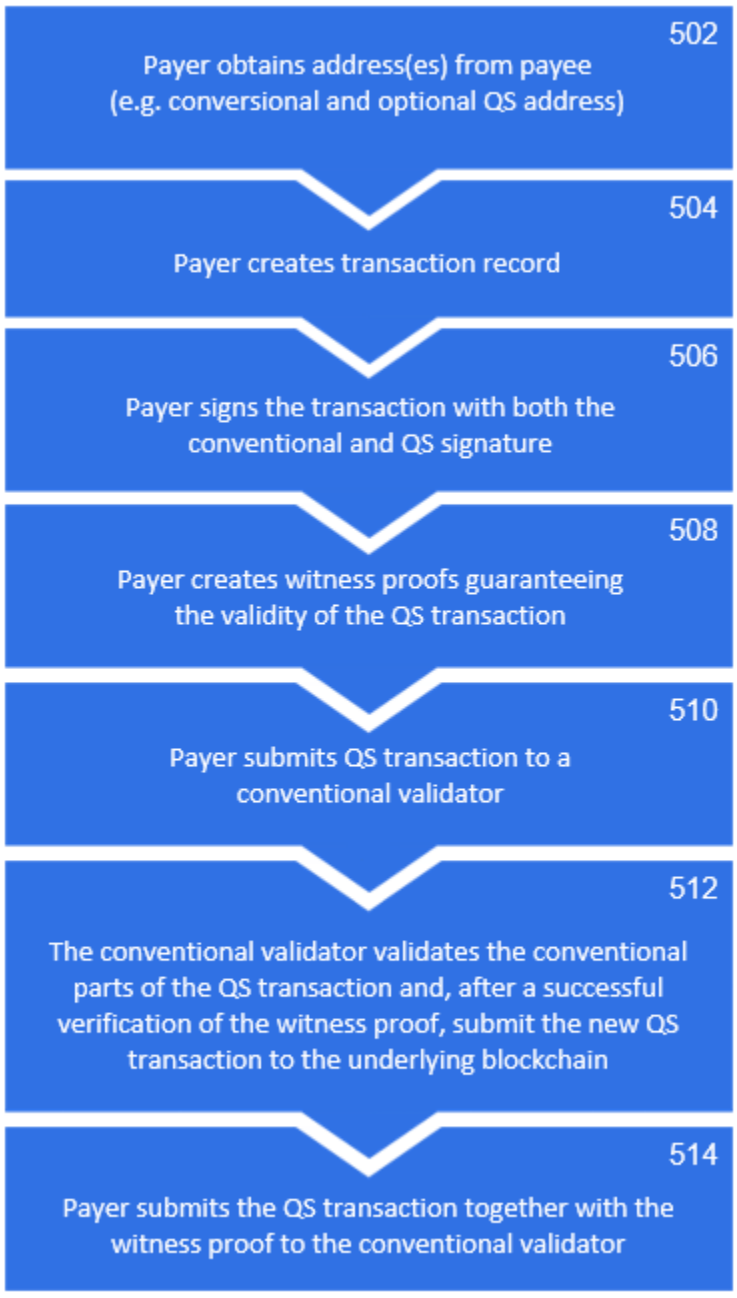


Figure 5

In step 504, the payer creates a transaction record that includes the conventional address and optionally QS address of the payee. The transaction record can include basic information such as the amount or item to be transferred, conventional addresses or other identifiers of the parties, and the type of smart

contract to be used for recording the transaction. In this example, the payer also includes in the transaction record the hash value of the payer's QS public key.

The payer signs the transaction with both conventional and QS signatures 506. Both the conventional signature and the hash value of the QS signature can be included in the transaction record whereby the zero-knowledge proofs will be recorded in the transaction that the QS signature of the payer has been verified and that the QS signature of the payer is correctly associated with the corresponding hash value written in the transaction. Zero-knowledge proofs created in 508 include (1) proof that the QS signature of the payer has been verified; (2) proof that the QS address of the payer is correctly associated with the QS public key of the payer; (3) proof that the QS signatures of the payer is correctly associated the hash written in the transaction; (4) proof that the amount being transferred is valid; and (5) a Merkle tree for the block of transactions constructed properly with a publicly visible root for a recording block on the chain.

In step 510 the payer sends the transaction to a conventional validator. Selection of a validator can be achieved by many well-known methods such as a staking mechanism, etc.

Alternatively, for example, the payer can send the transaction record to a smart contract execution entity, and that entity can send the transaction record to the conventional validator for processing prior to executing a smart contract based on the transaction record.

In step 512, the conventional validator can test various constraints of the proposed transaction, such as the status of the parties, availability of funds or items for transfer, validity of the QS signature of the payer by verifying the zero-knowledge proofs written in 508, etc.

If the conventional validator determines in step 512 that the conventional part of the QS transaction is valid as well as having successfully verified the zero-knowledge proofs written in 508, it can submit the QS transaction to the conventional blockchain.

In many cases, as proposed earlier in an optional roll-up mechanism a conventional validator can adopt to significantly enhance the throughput of the blockchain operation: one or more zero-knowledge proofs can be generated once a group of transactions being validated and submitted together. For example, it will not be necessary to repeat the work of proving the correctness of the conventional address and the optional QS address of a payer for each transaction in the block that has the same payer. The conventional validator can provide a single signature for all the transactions, and one Merkle tree can be used for all the transactions in a block submitted by various payers.

The smart contract can test any number of constraints of the proposed transaction and/or perform QS validation of the proffered transaction. Testing of constraints can include, for example, checking the transaction for consistency, e.g., that funds are not double spent, etc. This can be accomplished by using various known methods, such as building a spending Merkle tree and signing the transaction with both a conventional/QS signature along with verifying all the necessary zero-knowledge proofs before submitting the transaction to the underlying blockchain.

Once the requirements of the smart contract are satisfied, in step 514 a completed record smart contract transaction data is submitted to the blockchain. Here the conventional validation process of the underlying blockchain will be followed. The QS zero-knowledge proof mechanisms included in the transaction record provide for verification of quantum safety within the conventional blockchain, whereby

the conventional blockchain provides secure public storage for the transaction between parties as well as assurances that quantum security of the transaction has been achieved.

It should be appreciated that many variations of the processes described are possible, including alterations in which entities include what data in which interaction.

For example, the applications used by payers and payees may exist within the conventional validator network, where the users' wallets are stored in the underlying blockchain network. Alternatively, the conventional validators may deal exclusively with transactions originating outside of any application under their control. Moreover, the conventional validators may deal with a mixture of proprietary wallets and transactions originating outside of the network or the optional roll-up mechanism described earlier.

Payers and payees may be identified in various ways on the blockchain record, and not necessarily by hashes of their QS addresses, for example. QS addresses, QS signatures, and other QS data may be passed directly or via lookup tables, and tables may be indexed by hashes of stored items or by other rubrics.

PERFORMANCE

Since the L1 quantum-safe blockchain in this paper is chain independent, it can be any chain that supports smart control. For example, HyperLiquid and Solana which are using a proof-of-stake (more precisely, a proof-of-history) consensus model. They follow the same sharding mechanism in the original L1 chain. Both HyperLiquid and Solana are famous for their throughput measure in terms of transactions-per-second (TPS). Even though quantum-safe digital signature is conceived to be slower than that of ECC, TPS of the quantum-safe Solana transactions are believed to be 100% the same as the underlying L1 chain since the quantum-safe signature is performed at the payer end. This means the process is essentially decentralized, which is running "off-the-chain". As described herein, the optional roll-up mechanism can be applied to significantly speed up the throughput. It is an ongoing measure to explore ways to increase TPS include. In addition, verification of zero-knowledge proofs at the conventional validation process is a very fast operation as compared to that of the writing of zero-knowledge proofs by the payers.

The quantum-safe blockchain will continue to optimize individual validator performance, as well as experimenting with scaling techniques that add more validators to the network. Both directions have distinct trade-offs. Any blockchain with parallel execution capabilities can support additional concurrency by requiring more powerful hardware or even structuring each validator as a cluster of individual machines. However, there are practical limits to the number of global validators based on the cost and complexity for validator operators. The rise and popularity of serverless databases in cloud services exemplify how a small number of entities can efficiently deploy and maintain these types of complex distributed systems.

CONCLUSION

The quantum-safe tokens mark a revolution in the post-quantum digital asset industry with the ability to not only protect against the arrival of Q-Day, but also retain the features, flexibility and performance of its pre-quantum ancestor. Some examples of features are smart contract capability, high transaction-per-second, low gas fee, NFT and Web3 support, etc.

We envision a decentralized, secure, and scalable network operated and governed by the community that uses it. The computational resources of this post-quantum blockchain can be scaled up to meet all demands for growth.

From a practical standpoint, after Q-Day the value of any digital asset remaining in a wallet that is not quantum-safe can potentially be reduced to virtually $0 in a short period of time. This is a significant, trillion-dollar crisis that cannot be resolved overnight. Our technical goal is to provide to our partners a quantum-safe token minted on an existing L1 chain without altering any of the internal structure of the chain. The pilot project with our partner is to launching a quantum-safe native token minting on HyperLiquid. Our plan is to offer this as a "quantum crypto harbor" before the arrival of Q-Day allowing enough time for crypto holders to park their tokens into a quantum-safe version. For the crypto traders, it is an opportunity for crypto arbitrage as the price of quantum-safe tokens should rise and the price of all non-quantum safe tokens should fall when the threat of quantum computer hacks becomes more widely recognized.

About 01 Quantum Inc.

Established in 1992, 01 Quantum is always at the forefront of technology. Its latest innovation is on cyber security with its latest development focused on Post-Quantum Cryptography (PQC). Our patented invention PQC, together with PQC selected by NIST, are designed to operate on today's conventional computer systems to safeguard against potential cyberattacks from quantum computers. Our technology has been designed to transform today's cyber security in a way that is safe against future attacks from the world of quantum computers. Examples of vertical applications are emails/files encryption, digital signatures, blockchain security, remote access/VPN, password management, credit card security, cloud storage, artificial intelligence, IoT and web site security.

INDEX