Peer-to-Peer Networking and Applications
https://doi.org/10.1007/s12083-021-01152-z

A blockchain-based quantum-secure reporting protocol

Saeed Banaeian Far[1] · Maryam Rajabzadeh Asaar[1]

Received: 24 November 2020 / Accepted: 2 April 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract
The reporting systems are needed to design so that the whistleblower's privacy, report confidentiality, and report integrity should be under-consideration. Additionally, it is expected that the approved report will be accessible publicly and not changed. We believe that blockchain technology is the best choice for reporting systems' infrastructure since it provides a transparent and immutable database. This paper presents the first blockchain-based quantum-secure reporting protocol (QS−RP) using multivariate public key cryptography (MPKC). In the QS−RP, a fast verification mechanism is applied, which makes use of the Merkle technique. The QS−RP provides confidentiality to the selectively secure multi-key (C-SSMK) and unforgeability to selectively secure multi-key (UF-SSMK). Additionally, the QS−RP provides several new features such as report confidentiality before report generation, user/whistleblower privacy, and report integrity. The most important feature of the QS−RP is providing the whistleblower's privacy and report confidentiality against quantum computers. Analysis of the security of the QS−RP indicated the mentioned claims in the random oracle model (ROM). Finally, the QS−RP is compared with other blockchain-based reporting protocols. The comparison shows the QS−RP provides more security features than other reporting protocols, and the performance analysis's results show that it is 90% faster in the execution time on the user side, and it is 66% efficient in the communication overhead in compared to other blockchain-based reporting protocols. Moreover, the QS−RP has no on-chain overhead for whistleblowers.

Keywords Anonymous reporting · Blockchain · Multivariate public key cryptography · Quantum-secure

1 Introduction

The blockchain concept was first presented in 2008 by S. Nakomoto under the Bitcoin project [1]. Blockchain technology is a distributed ledger or a distributed database with several features such as being transparent, immutable, and open-source [2, 3].

The blockchain consists of linked blocks through a chain of hash functions used as a distributed database where each network user can have a copy [4]. It can be said that the blockchain technology got its main idea from error propagation since each change in the previous block(s) can change all the following blocks' output. These features have caused the blockchain technology to be an immutable and transparent database since all network users have access to it, and each change in the recorded data or previous block(s) appears in the next one, continuing up until the current one. After a few years, researchers and developers found that in addition to the financial application [5–7], blockchain technology can be applied in other fields such as in controlling users' access to recorded data [8], vehicular networks [9–11], electronic health records [12, 13], industrial internet of things [14], cellular communication [15], and energy systems [16]. Another field where blockchain technology can be applied as an immutable database is in systems which have been designed to collect reports [17, 18]. The need to have a transparent and immutable database is sensed here since reports are always made against one or many people. Despite this, people affected by the sent reports try to destroy, change, delete or steal those reports. We believe that the blockchain is the best choice to store reports since they will be kept immutable, and everyone can gain access to them. In recent years, blockchain-based reporting protocols have been a case that researchers and developers have focused on, due to the fact that cyberspace users have always wanted to know about everything around them. They want to ensure that the recorded information will not be changed by existing authorities, privileged insiders, or service providers. As mentioned above, we believe that blockchain technology can be applied in reporting systems since all

✉ Maryam Rajabzadeh Asaar
asaar@srbiau.ac.ir

Saeed Banaeian Far
saeed.banaeian@srbiau.ac.ir

[1] Department of Electrical and Computer Engineering, Science and Research Branch Islamic Azad University, Tehran, Iran

Published online: 01 May 2021

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

network members demand access to submitted reports. However, in this type of system, the whistleblowers' privacy is an issue under-consideration. That is, whistleblowers want to be under protection after their reporting in such a way that no one can trace them. Additionally, there should be some incentives (e.g., assigning a reward to the whistleblower) to motivate network users to send reports. Another challenge is the fact that, on paying rewards, the whistleblower's privacy is at risk. Such security challenges accordingly require more attention from researchers and developers.

In 1994, P. Shor presented a quantum-based algorithm to solve two families of NP-hard problems designed based on the hardness of number theory problems (integer factorization problem such as RSA and QR, and discrete logarithm problem such as ElGamal and ECC) in a polynomial time [20]. As a consequence of this, schemes designed based on the hardness of the number theory problems currently in use would be broken in the post-quantum age, the users of which would have no privacy. After the Shor's algorithm, researchers and developers tried to find quantum-secure methods (or methods resist against quantum computers) on which they could design cryptographic primitives. All four found methods (lattice-based cryptography, code-based cryptography, multivariate cryptography, and hash-based signature [21]) had been presented before the presentation of the Shor's algorithm. In those years (after 1994), researchers proposed quantum-secure security protocols and schemes. Post-quantum cryptography aims to develop systems compatible with current computers while cooperating with existing processors and communication protocols, creating security against quantum computers. Therefore, researchers and developers are now ready to deal with quantum computers using post-quantum cryptography. Almost all post-quantum cryptographic primitives have now been designed. However, some of today's recently-designed security protocol challenges have no direct version for the post-quantum age. For example, there is no blockchain-based reporting protocol that preserves the whistleblower's privacy. As a result, at the post-quantum age, the whistleblowers' privacy will be at risk. Thus, there is a need for the existence of a transparent and immutable reporting system which works based on traditional computers that have security against quantum computers. Then, users in the current age, acting as whistleblowers, could send their reports through a quantum-secure protocol and ensure that their privacy will be kept at all times.

Contribution: In this study, the first blockchain-based quantum-secure reporting protocol (QS−RP) is presented by applying multivariate public key cryptography (MPKC) and a secure one-way hash function. The QS−RP is a fast and efficient reporting protocol; it provides features such as

i) report confidentiality before the generation (to prevent publishing invalid reports in public, no one, even the CA, recovers the sent report before the report generation process),
ii) user anonymity (the whistleblower is anonymous and untraceable),
iii) report integrity (the submitted report is the same one sent by the whistleblower),
iv) individual verifiability for report integrity (only the whistleblower using its real identity can verify whether or not the submitted report on the blockchain has kept its integrity),
v) efficiency (the fast cryptographic primitives and schemes are used in the QS−RP), and
vi) security against insider adversaries (no insider adversary can create disorder in the QS−RP).

To design the QS−RP, the following methods were applied:

- A secure hash function was applied to generate the user/whistleblower's pseudonym, efficiency, as well as the report's integrity checking.
- A proof-of-presence method based on the Merkle tree technique [22] was used to implement a fast-verification method for verifying the threshold number of auditors who want to cooperate. Accordingly, if malicious auditors create a disorder, or if some auditors do not cooperate, the QS−RP will work.
- The concept of proof-of-trust (PoT) [23, 24] was used to share parts of the private key among all auditors, so that more reliable auditors get more valuable keys.

Finally, it was shown that the QS−RP provides C-SSMK-security and UF-SSMK-security in the random oracle model (ROM). Additionally, it was found that the QS−RP is more efficient and secure than other recently-proposed reporting protocols and provides more security features. The performance analysis shows that the QS−RP, on the user side, is 90 % faster in the execution time (for 10 users), and in comparison to ring signature-based reporting protocols, it is 66% efficient than them in the communication overhead.

Organization: In Section 2, we present an overview of some related works on reporting protocols, while it also secure transaction protocols. In Section 3, we describe the needed preliminaries of this paper. In Section 4, we present the QS−RP in detail and analyze it. In Sections 5 and 6, we compare the presented QS−RP with other blockchain-based reporting schemes and present the conclusion of the paper, respectively.

2 Related Works

Reporting protocols have been proposed in various fields such as mobile ad hoc networks, wireless sensor networks, and vehicular ad hoc networks. Nevertheless, blockchain-based

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

reporting protocols are a recently-proposed concept. Reporting protocols can be divided into two categories, including centralized and blockchain-based protocols (it should be noted that there is only a limited number of reporting protocols designed which are based on blockchain).

2.1 Centralized Protocols

Stumpf *et al.* proposed an integrity reporting protocol to provide a secure remote attestation for preventing masquerading attacks [25]. The authors applied the Diffie-Hellman key exchange protocol [26] and the RSA cryptosystem [27] in their presented protocol in such a way that applying the key exchange protocol is the main idea that makes their protocol secure against masquerading attacks.

In 2009, Choi *et al.* presented the *ASR* protocol that used random nodes (to create random and unpredictable links) to forward a report between two destinations [28]. Through this method, each malicious node cannot modify its reported behavior based upon the monitoring point. Therefore, the report's integrity and authenticity are preserved through the use of random and multiple links.

The *SinkTrail* protocol was presented by Liu *et al.* in 2011 [29]; a proactive reporting method suitable for wireless sensor networks was provided in the *SinkTrail* protocol. The *SinkTrail* protocol solved the sink node's inability to move freely in a distributed area for times when the pre-calculated paths are not applicable. The authors focused on *i*) dynamically adapting to various terrestrial changes, and *ii*) routing and forwarding data packets. In the last two discussed protocols (*ASR* [28] and *SinkTrail* [29]), the authors focused on transferring reports between two destinations through two different ways. As they have centralized architectures, they paid no attention to features such as report immutability (after receiving and recording) and transparency.

Tripp Barba *et al.*, Kamel *et al.*, and Li *et al.* presented three reporting protocols that were designed for vehicular networks/electronic vehicles [30–32].

Tripp Barba *et al.*'s protocol [30] allows drivers to report traffic accidents anonymously to avoid personal and professional repercussions. This study aims to propose a new collaborative protocol for enforcing anonymity in multi-hop VANETs, closely inspired by the well-known Crowds protocol. Their protocol is dependent on a forwarding probability that determines whether the next forwarding step in message routing is random. An important item in their protocol is to resist against multi-hop lossy wireless networks. Li *et al.* presented the *Lynx* protocol which provides anonymous real-time reporting among electric vehicles [31]. In Kamel *et al.*'s protocol [32], vehicles have the ability to send other vehicles' misbehavior report to the road-side unit (RSU). Upon report approval, the public-key infrastructure revokes the under-report vehicle. In 2017, Li *et. al* proposed an anonymous data reporting strategy based on a blind signature that motivates users to send misbehavior reports [33]. The presented strategy suggested a system for ensuring anonymous data reporting.

2.2 Blockchain-based Protocols

In 2018, Buldas *et al.* presented a hash-based server-supported digital signature scheme. They suggested that the concept of blockchain authenticated data structures, and they presented a blockchain design that could have independent values [34]. The fresh signatures' keys are generated by supporting the server using a hash tree and a hash chain. They claimed that there was no need to have a fully trusted server in their scheme. The proposed hash-based signature can be applied in the blockchain application since it is efficient.

In 2018, Kiktenko *et al.* proposed several solutions to solve challenges of post-quantum blockchains [35]. To provide security against quantum computers, they combined the original byzantine fault tolerance state-machine replication without using digital signatures, a quantum key distribution (QKD), and the Toeplitz hashing method.

Another scheme presented in 2018 applied a lattice-based one-time linkable ring signature to deal with quantum computers [36]. The ring signature is used here to provide user anonymity.

In 2019, three blockchain-based protocols were proposed to send reports by anonymous network users [17, 18, 37]. To preserve the users' privacy in the mentioned protocols, the authors applied a ring signature and a zero-knowledge proof (ZKP), respectively. In the three mentioned protocols, the currently-in-use (not post-quantum) cryptographic primitives were applied.

The authors in *Reportcoin* [17] applied a ring signature to make users who want to report a misbehavior anonymous.

A ZKP scheme is applied in ARS-PS [18] to make the assigned reputation confidential until proof.

The BB2AR's authors [37] have acted like a *Reportcoin*, using a ring signature for user anonymity. The method they used to give anonymous rewarding is based on [19] that presented the concept of blockchain-based anonymous rewarding for the first time.

In 2019, F. Esgin *et al.* proposed a new lattice-based ring signature for confidential transactions [38]. They also presented a formal definition for RingCT-like protocols and an extractable commitment scheme based on the lattice.

In 2020, Shahid *et al.* presented a distributed ledger using a hash-based signature that could be adapted with the blockchain technology and implemented on IoT [39]. In their network architecture, two layers of perception (the layer where devices create the hash-based signature) and communication (the layer where the peer-to-peer network is) are assumed.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

3 Preliminaries

The preliminaries of this paper are presented in this section, and we list the used notations in Table 1.

3.1 One-Way Hash Function

The one-way hash function is defined as a map $h: \{0,1\}^* \rightarrow \{0,1\}^l$ (l is assumed as a constant length for output, e.g., 256 or 512) that there is a probabilistic polynomial-time (PPT) algorithm to calculate $h(r)$ where $r \in k$ and k is a finite field. But no PPT algorithm can find r for given $h(r)$ [40]. Additionally, no PPT algorithm can find two random values r and r' ($r \neq r'$) such that $h(r) = h(r')$. The secure one-way hash function provides a security against quantum computers [21], and the advantage of a PPT adversary $\mathcal{A}$, who has an access to a quantum computer, to find r for given $h(r)$ is calculated as:

$$ADV_{\mathcal{A}}^{hash} = Pr[\mathcal{A}(h(r)) = r \cup \mathcal{A}(r) = r' | h(r) = h(r')] < \varepsilon$$

Table 1 The List of Notations

Notation	Description
$\mathcal{A}$	Adversary who can execute the Shor's algorithm
ACA	Auditor in Class A
ACB	Auditor in Class B
$\mathcal{C}$	Challenger
CA	Central authority
F	Map of public key
F^{new}	New public key
$\hat{F}(.)$	MPKC-based encryption function
$h(.)$	Secure one-way hash function
ID_i	ith user's identity
$KP1$	Key pool 1
$KP2$	Key pool 2
L_1, L_2	Original private keys
L_1^{new}, L_2^{new}	New private keys
$l_{1,i}$ and $l_{2,i}$	Assigned part of original private keys to ith auditor
NA	Number of auditors
NPK	Number of private keys
PK_i	ith part of private key
PID_i	ith user's pseudonym
s	Secret key of CA
TH	Threshold number of trusted auditors to verify
U_i	ith user
x	Plaintext (report)
y	Ciphertext (private report)
$\|\|$	Concatenate operation

Definition 1 The secure one-way hash function algorithm is secure against quantum computers, and $\mathcal{A}$, who has access to a quantum computer, can learn nothing about r on having $h(r)$, and it cannot find two random values r and r' such that $h(r) = h(r')$.

3.2 Merkle Tree

The Merkle tree is defined as a 2-to-1 hash function shown as $h: \{0,1\}^{2k} \rightarrow \{0,1\}^k$ and formalized as $input_{level\_i} \leftarrow h(output_{left\_level\_i-1} \| output_{right\_level\_i-1})$. We show the Merkle tree structure in Fig. 1 (for more details refer to [22]).

The Merkle tree is applied as a mechanism to prove whether or not a presentation. This mechanism is called proof-of-presence (PoP), and we define PoP algorithm as $(\{1,0\}) \leftarrow$ PoP(Parts of private key, TH, Merkle root). The parts of the private key, threshold TH, and Merkle tree root (the mentioned symbols are named due to the paper's notations) are given PoP algorithm and returns 1 if the parts of the private key are more than the determined TH. Else, it returns 0.

3.3 Multivariate Cryptography

Multivariate cryptography has been known since 1980 as a fast encryption and signature algorithms. In multivariate cryptography, m sets of the polynomial are defined, and the multivariate cryptography algorithm calculates all output sets [41, 42] based on the hardness of the multivariate quadratic polynomial system (MQ problem [43]). In the following, we describe MPKC's general definition in more detail [43, 44].

In the MPKC, m and n are given as two positive integers and the finite field k is defined as $k = GF(q)$. The map $\hat{F}: k^n \rightarrow k^m$ consists three invertible maps that are defined as $\hat{F} = L_1 \circ F \circ L_2$ where $L_1: k^m \rightarrow k^m$, $F: k^n \rightarrow k^m$, and $L_2: k^n \rightarrow k^n$. We show used elements of the MPKC below.

- *Keys:* The public key in the MPKC is defined as a map of F over a finite field k, and private keys are L_1 and L_2.
- *Encryption:* The ciphertext $y \in k^m$ is computed as $y = F(x)$ where $x \in k^n$ is a plaintext, and we can write $x = L_2^{-1} \circ F^{-1} \circ L_1^{-1}(y)$ as the decryption map.
- *Signature:* The pair $(x, y) \in k^n \times k^m$ is given as a signature, and the signature is verified if $y = \hat{F}(x)$.

Security against quantum computers: As mentioned before, MPKC provides security against quantum computers [21]. The two definitions below generally show the advantages of $\mathcal{A}$, that has access to a quantum computer, to break message confidentiality for an MPKC-based encryption algorithm and unforgeability for an MPKC-based signature algorithm.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

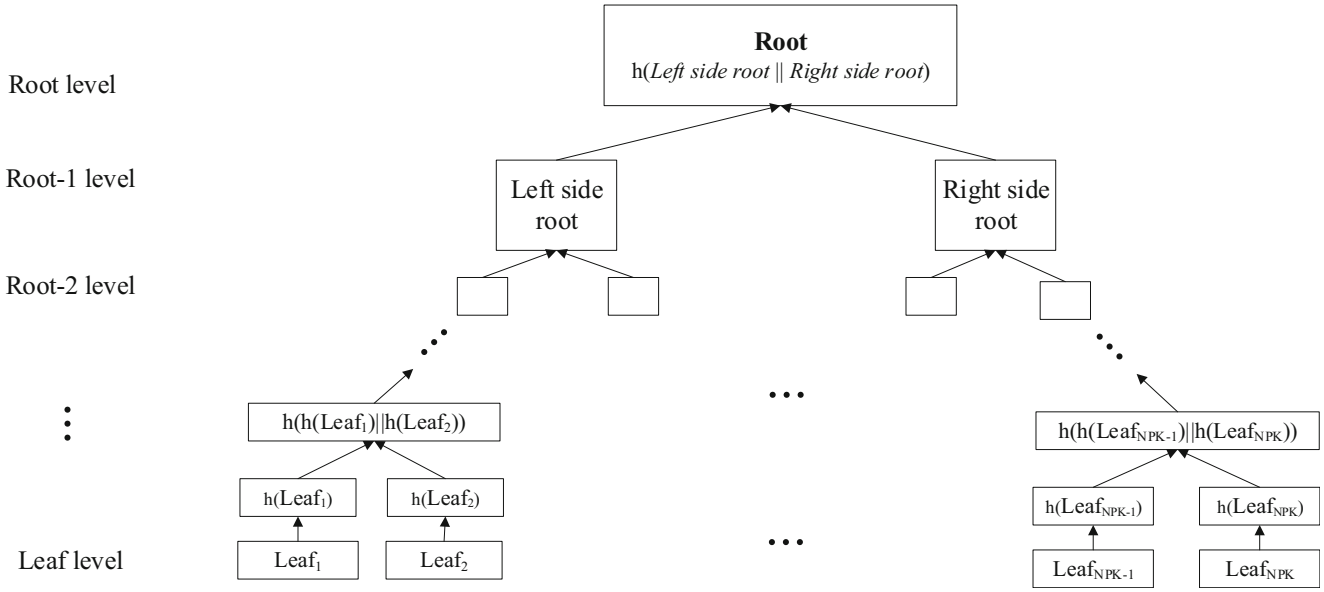

Fig. 1 The Merkle Tree Structure

– *Confidentiality:* There is a PPT algorithm to calculate the cipher $y = F(x)$. However, no PPT algorithm can obtain the message x for given the cipher y. Therefore, the MPKC-based encryption algorithm is quantum-secure, and the advantage of a PPT $\mathcal{A}$, that has access to a quantum computer to break the message confidentiality of the MPKC-based encryption algorithm for given the cipher y is calculated as:

$$ADV_{\mathcal{A}}^{Confidentiality} = Pr[\mathcal{A}(\dagger) = \S|\dagger]$$

Definition 2 The MPKC-based encryption algorithm provides message confidentiality against quantum computers, and $\mathcal{A}$, who has access to a quantum computer, cannot find the content of the encrypted message $y = \hat{F}(x)$.

- *Unforgeability:* Similarly, on having the private key, there is a PPT algorithm to sign the message x. However, no PPT algorithm can create a valid signature on the message x if access to private keys is denied. The advantage of $\mathcal{A}$, that has access to a quantum computer to forge the MPKC-based signature algorithm is calculated as:

$$ADV_{\mathcal{A}}^{Unforgeability} = Pr[\mathcal{A}(\S) = (\dagger, \S)|\S]$$

Definition 3 The MPKC-based signature algorithm provides unforgeability against quantum computers, and $\mathcal{A}$, that has access to a quantum computer, cannot create a valid signature pair (x, y) if it only has public parameters.

3.4 Problem Definition

As aforementioned, the presented blockchain-based reporting protocols are state-of-art issues to be applied in different societies. However, the whistleblower privacy, report integrity, and the content of the sent report, which its validity has not been approved, should be kept. It has been proved that reporting protocols [17, 18, 37] designed based on the number-theory-based problems' hardness will provide no security for their users in the post-quantum age, and the whistleblower's privacy will be compromised. Additionally, it is probable that malicious insiders, who are affected by the sent report, try to change, steal, or delete the sent report.

This paper presents the first blockchain-based reporting protocol using post-quantum cryptography through MPKC such that provides report integrity, report confidentiality before approval, report accessibility, and whistleblower's privacy in the post-quantum age. Blockchain technology prepares immutability for submitted reports.

Accordingly, the below listed problems, as the main problems, are defined and solves in this study:

i) The main problem that this paper focuses on is the whistleblowers' security in reporting protocols in the post-quantum age.
ii) The sent report should also be kept confidential before approval, and adversaries or privileged insiders should be able to learn nothing about the sent report's content.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

iii) The other existing problem in reporting protocols, which has been solved in this study, is to delete, change, or damage registered reports by the mentioned entities.

This paper presents the first blockchain-based quantum-secure reporting protocol adapted to current computers, and the three mentioned problems, namely the whistleblowers' privacy in the post-quantum age and avoiding report tampering are solved. For more clarification, we present a practical example in the following:

Imagine a company that wants to implement a reporting system to find its malicious staff. The company owner allows staff to report all misbehavior they see. Nevertheless, there is a concern of getting bad feedback for the whistleblowers (staff who reports the misbehavior they see) or changing whistleblowers' sent reports by privileged staff. Applying a blockchain-based reporting system that provides user/staff privacy at all times is the best choice for the company owner to remove staff concerns. Upon having the aforementioned system, staffs rely on the company, and misbehaviors around the company are thus decreased to a great extent.

It is suggested that this type of system that decreases misbehavior and increases reliability in society be applied in other parts of society as well.

3.5 The Quantum-Secure Reporting Protocol Framework

In this section, we define the QS−RP Framework.

3.5.1 Network Model

We show the QS−RP's system model in Fig. 2 and describe details of the QS−RP's system model in the following:

- **Algorithms:** There are some required algorithms to present the QS−RP's system model; they will be defined in the following:

- ($params$) ← Setup(1^{λ}): The security parameter λ is given to Setup algorithm, and the set of system parameters $params$ is returned.
- (F, L_1, L_2) ← KeyGen ($params$): The set of system public parameters $params$ is given to KeyGen algorithm, and the pair of public-private keys are returned.
- ($\{ACA, ACB\}$) ← PoT(auditor's ID): The auditor's identity is given as an input, and PoT algorithm returns the class of auditor due to a proof-of trust (PoT) mechanism [23, 24].
- ($KP1$, $KP2$) ← KeyAssig(L_1, L_2): The generated system private keys are given to KeyAssig algorithm, and KeyAssig algorithm divides them into NPK parts. The private key's divided parts are categorized into two key pools $KP1$ and $KP2$, where $|KP1| = |KP2|$. Assigning each part of the private key to each auditor is done with the help of PoT algorithm.
- (PID_i, $pour_i$, $mint_i$) ← Transac(ID_i, x, F, F^{new}, L_1^{new}, L_2^{new}, t): The Transac algorithm takes the user identity ID_i, the report x, the system public key F, a new set of public-private keys based on MPKC (F^{new} and L_1^{new}, L_2^{new}), and the current time t. It returns tuples of the report x, includes the user pseudonym PID_i and two ciphers $pour_i$ and $mint_i$.
- (H_i, Y_i t_{new}) ← ProofGen($pour_i$, part of private key): The ProofGen algorithm takes $pour_i$ and a part of private key, related to the auditor, and returns the verifying proof H_i, Y_i t_{new}.
- ((x, y)) ← RepGen(Verifying proof, L_1, L_2): The RepGen algorithm takes the verifying proof (H_i, Y_i t_{new}) tuples of the report x and returns the signed report x by the system private keys L_1, L_2 and submits it on the blockchain if 1 was returned by PoP algorithm.
- (1, 0) ← CVerif((x, y), F): The CVerif algorithm takes the signed report (x, y) and system public key F and returns 1 if the signed report is valid. Else, it returns 0.

It is assumed that the number of auditors NA is larger than the number of the divided private key NPK ($NA \gg NPK$ and all parts of the divided keys are assigned to all auditors randomly[1].

- **Entities:** Four entities exist in the QS−RP system model; they are defined below:

- *Blockchain:* The blockchain is assumed as a distributed, immutable, and transparent database and is responsible for queries.
- *Central authority (CA):* The CA is assumed as a fully-trusted party who initializes the system (QS−RP), verifies the report's generated proofs, and signs the proved report and submits it on the blockchain.
- *User (whistleblower):* The user is assumed as a regular network member who could work as a whistleblower and sends a private report to approval and submit on to blockchain.
- *Auditor:* Auditors are assumed as semi-trusted parties who can cooperate in the "proof generation" process. It is probable that several malicious auditors are present

[1] There is a probability that each part of the private key is given to several auditors. However, no one tries to find who has an equal part of the private key similar to itself since this knowledge provides no advantage for the auditor who finds that.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

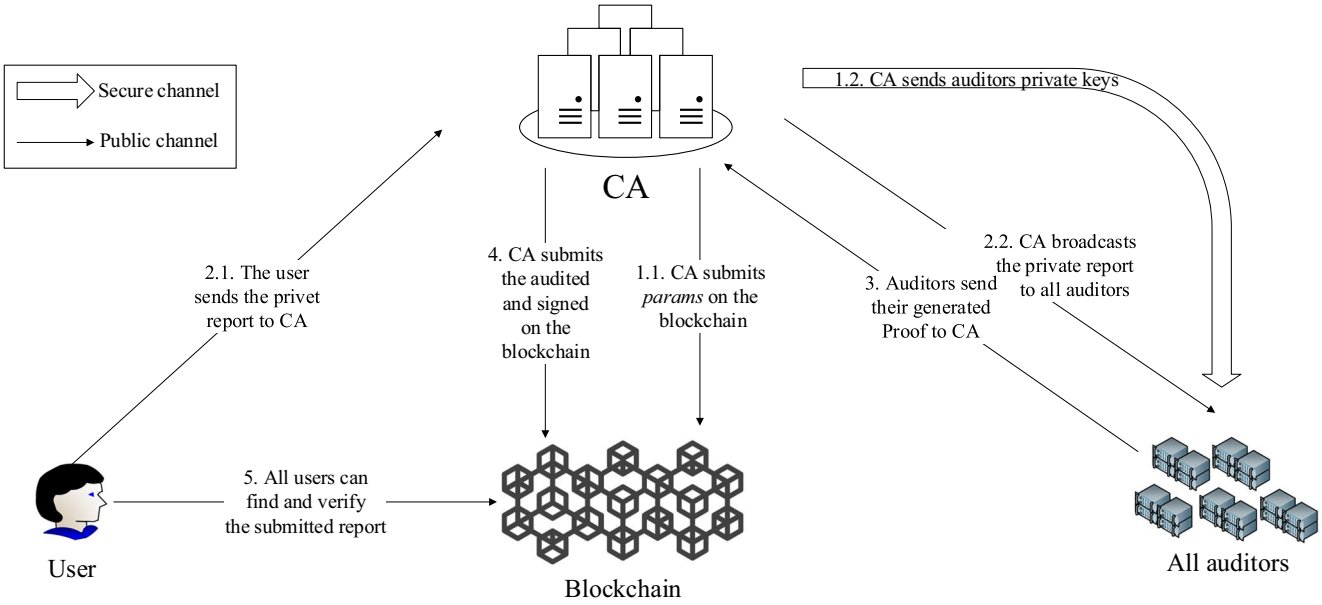

Fig. 2 System Model of the QS-RP

among them and want to create disorder in the "proof generation" process.

Remark 1 It seems that there is a conflict between the decentralized architecture of the blockchain and an entity called CA. However, according to the protocols discussed in Section 2.2, in most blockchain-based protocols, some entities handle systems. To name a few examples, we can mention those called Server (see [34], Page 5), Verifier (see [17], Page 6), IDM (see [18], Page 2), and Authority (see [37], Page 4).

Therefore, it is required that, in addition to the blockchain as the distributed database, there needs to be an entity for initialing the QS-RP, verifying the received proofs, and assigning and sending the whistleblower's reward.

– **System model:** According to Fig. 2 and defined algorithms, the QS-RP is defined in five phases as below:

1. (*params*) $\leftarrow$ *Setup*(λ): This phase includes two steps:

1.1. At first, Setup, KeyGen, KeyAssig, and PoP algorithms are executed by CA using the security parameter λ, and the system public parameters *params* is submitted on the blockchain as the system genesis block.

1.2. The CA gives divided parts of the private keys to auditors through a secure channel.

2. (*Confidential tuples of the report*) $\leftarrow$ *Transact*(*Report*): This phase includes two steps:

2.1 A user who wants to submit a report on the blockchain executes Transac algorithm to create its pseudonym and parts of confidential tuples of the report (an encrypted report and hashed report are generated and prepared to be sent). It then, through a public channel, sends the generated tuples of the report tuples and its pseudonym to CA.

1. Upon receiving report tuples, CA creates a new Merkle tree related to the received tuples and broadcasts required tuples among auditors to verify[2].

3. (*Verifying proof set*) $\leftarrow$ *Proof generation(Required tuples):* After the broadcasting, auditors who want to join the proof generation process (they could give up the received report tuples and avoid cooperating in the "proof generation" process) execute ProofGen algorithm to get the verifying proof set. They then send the given verifying proof to CA through a public channel (it means they agree to decrypt and submit the report on the blockchain). Then, CA decrypts the private/encrypted report.

4. (Signed report) $\leftarrow$ *Report generation(Verifying proof set):* The CA executes RepGen algorithm to sign and

[2] Maybe it is easier if the user broadcasts the confidential report among all auditors, but in this case, *i*) the user has to consume a lot of energy, and *ii*) CA cannot create the original Merkle tree for checking.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

submit the verified report set and the user's reward on the blockchain if PoP algorithm, described in Section 3.2, returns 1 (the threshold number of auditors *TH* are present in the proof generation process) and the verified report is valid and not repetitive.

5. $(\{0,1\}) \leftarrow$ *CVerif(Signed report):* Finally, the user/whistleblower can gain its reward, and all other users can find the submitted report on the blockchain and centrally verify it using CVerif algorithm.

3.5.2 Design Goals

In the following, we list goals that the QS−RP should be achieved.

The random oracle as a simulator is applied to make queries for oracles.

1. **Report confidentiality:** The sent report should be kept confidential, and no user in the network should know the content of the sent report. The QS−RP provides *report confidentiality* if no PPT $\mathcal{A}$ can win in *Game 1* designed as below.

– *Setup:* The challenger $\mathcal{C}$ calls $\texttt{Setup}(1^{\lambda})$ and gives *params* to $\mathcal{A}$, and keeps the private key secure.
– *Experiment:* The $\mathcal{A}$ submits polynomially bounded numbers of queries to ProofGen oracle (*Proof generation* phase of the QS−RP includes two steps, generating verifying proofs and decryption of the sent private report, in this game, the decryption of the sent report is $\mathcal{A}$'s ideal) and collects all pairs of plaintext-ciphertext as responses. Then, $\mathcal{A}$ stores all received responses from ProofGen oracle.
– *Challenge:* The $\mathcal{C}$ selects a challenge c and executes Transact algorithm on it. Then, $\mathcal{C}$ gives output of Transact algorithm to $\mathcal{A}$.
– *Guess:* The $\mathcal{A}$ guesses a value $c^* = c$ such that $\texttt{Transact}(c^*) = \texttt{Transact}(c)$.

The $\mathcal{A}$ wins *Game 1* if it guesses the valid value for c^* with a non-negligible probability.

Definition 4 The QS−RP provides the *report confidentiality* if $\mathcal{A}$ wins *Game 1* with a negligible probability.

2. **Confidential to selectively-secure multi-key (C-SSMK):** Like *Game 1*, the report should be confidential. In this game, we assume that some malicious auditors cooperate in obtaining the private report. In *Game 2*, we assume that $\mathcal{A}$ has an access to *KP*2.

– *Setup:* The $\mathcal{C}$ calls $\texttt{Setup}(1^{\lambda})$, KeyGen(*params*), PoT(*ID*), and $\texttt{KeyAssig}(L_1, L_2)$, and gives *params*, and *KP*2 to $\mathcal{A}$, and keeps *KP*1 secure.
– *Experiment:* This phase is the same *Experiment* phase described in *Game 1*. But $\mathcal{A}$ has an access to *KP*2.
– *Challenge:* This phase is the same *Challenge* phase described in *Game 1*.
– *Guess:* This phase is the same *Guess* phase described in *Game 1*.

The $\mathcal{A}$ wins *Game 2* if it guesses the valid value for c^* with a non-negligible probability for given *KP*2.

Definition 5 The QS−RP is C-SSMK if $\mathcal{A}$ wins *Game 2* with a negligible advantage against $\mathcal{C}$.

3. **Ungorgable to selectively-secure multi-key (UF-SSMK):** Malicious auditors should not be able to forge the used MPKC-based signature or submit an invalid report on the blockchain instead of CA if they cooperate. The QS−RP is UF-SSMK if $\mathcal{A}$ fails in *Game 3* written below.

– *Setup:* This phase is the same *Setup* phase described in *Game 2*.
– *Experiment:* There are two approaches to submits an invalid report instead of a valid report on the blockchain; *first,* forging the used MPKC-based signature algorithm, and *second,* CA executes the used MPKC-based signature algorithm if PoP algorithm returns 1. To do both, $\mathcal{A}$ works as below:

– For the *first* approach, $\mathcal{A}$ submits polynomially bounded numbers of queries to RepGen oracle and collects responses (the main algorithm executed in *Report generation* phase is an MPKC-based signature, and in submitting queries to RepGen oracle, the output of the signature oracle is $\mathcal{A}$'s ideal).
– For the *second* approach, $\mathcal{A}$ submits polynomially bounded numbers of queries to proofGen oracle and stores received verifying proofs as responses (in *Game 1* the decryption was $\mathcal{A}$'s ideal, but in this game, $\mathcal{A}$ needs the generated verifying proofs).

The $\mathcal{A}$ sends received verifying proofs to $\mathcal{C}$, and stores received signature samples.

– *Challenge:* The $\mathcal{C}$ first executes PoP algorithm, and it then gives a random challenge c to $\mathcal{A}$.
– *Guess:* The $\mathcal{A}$ has to create a valid signature (y_c^*, c) on the challenge c.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

According to the two mentioned approaches, $\mathcal{A}$ wins *Game 3* if *i*) it guesses the valid signature y_c^* on the challenge c with a non-negligible probability such that $\hat{F}(c) = y_c^*$, or *ii*) PoP algorithm executed by CA returns 1.

Definition 6 The QS−RP is UF-SSMK if $\mathcal{A}$ wins *Game 3* with a negligible advantage against $\mathcal{C}$.

4. **Privacy:** We believe that the QS−RP provides a privacy if it supports the following features.

- *Report confidentiality before report generation:* To prevent the publishing of an invalid report in public networks, the report should be kept confidential before report generation.
- *User untraceability:* No one should be able to find a link between a user who sent two (or more) reports.
- *User anonymity:* The user who sent a report (whistleblower) should be under protection. The anonymity contains pseudonymity and untraceability. That means no one find the whistleblower's real identity, and no one find a link between submitted reports and the anonymous whistleblower.
- *Secure address:* The assigned reward to the whistleblower should be secure, and no one steal/gain it.

Definition 7 The QS−RP provides privacy includes report confidentiality before report generation, report integrity, user untraceability, user anonymity, and secure address.

5. **Security against insider attack:** The $\mathcal{A}$ can be present among auditors who have parts of the private key (*KP*2). The $\mathcal{A}$ should not be able to create a disorder in the QS−RP.
6. **Security against man in the middle (MitM) attack:** The $\mathcal{A}$ can be present on the public channel, and it wants to make a disorder in the proof generation process.
7. **Security against online attack:** The QS−RP is vulnerable to an online attack if $\mathcal{A}$ who present between an auditor and CA can recover the encrypted message by eavesdropping on the public channel.
8. **Security against offline attack:** The QS−RP is vulnerable to an offline attack if The $\mathcal{A}$ can change the submitted report.

Definition 8 The QS−RP provides a security against common attacks includes insider attack, MitM attack, online attack, and offline attack.

9. **Report integrity:** If a received message will be the same message before a correspondence, it is ensured that a message had kept its integrity.

Definition 9 The QS−RP provides report integrity and the submitted report on the blockchain by CA is precisely the same report sent by the whistleblower.

4 The QS−RP

This section presents the detail of the QS−RP and its analysis.

4.1 The Protocol

In the following, we describe the detail of the QS−RP, and for more clarification, the QS−RS is depicted in Fig 3.

4.1.1 Setup Phase

This phase includes two steps; we show them in Table 2 (Algorithm 1) and describe in the following:

- **Step 1:** To initialize the QS−RP system, Setup algorithm is executed by CA. In Setup algorithm, security parameter λ is given and the set of system public parameter $params = \{algs, m, n, k, h(.), \Delta t, TH\}$ is returned where $algs = \{$Transact, ProofGen, RepGen, CVerif$\}$, and Δt is the validity time of each proof generation process. The CA submits the set of QS−RP public parameters *params* as the genesis block on the blockchain. Then, *params* is given to KeyGen algorithm by CA, and the public key F and the private keys set (L_1, L_2) are returned by KeyGen. The CA publishes the public key F and keeps the private key (L_1, L_2) secure. The CA then executes PoT and KeyAssig, and private keys set (L_1, L_2) are divided into *NPK* parts as $L_1 = \{l_{1,1}, ..., l_{1,NPK}\}$ where $L_1 = \Sigma_i^{NPK} l_{1,i}$, and $L_2 = \{l_{2,1}, ..., l_{2,NPK}\}$ where $L_2 = \Sigma_i^{NPK} l_{2,i}$.
- **Step 2:** Then, CA sets two key pools $KP1 = \{l_{1,1}, l_{2,1}, ..., l_{1,NPK/2}, l_{2,NPK/2}\}$ and $KP2 = \{l_{1,NPK/2+1}, l_{2,NPK/2+1}, ..., l_{1,NPK}, l_{2,NPK}\}$. The CA assigns randomly each part of the private key to *ACA* auditors from *KP*1 and assigns randomly private key of *ACB* auditors from *KP*2 to them using KeyAssig algorithm. The CA sends each auditor's private key set $PK_i = \{l_{1,i}, l_{2,i}, i\}$ to it through a secure channel, and stores all PK_i sets secure.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

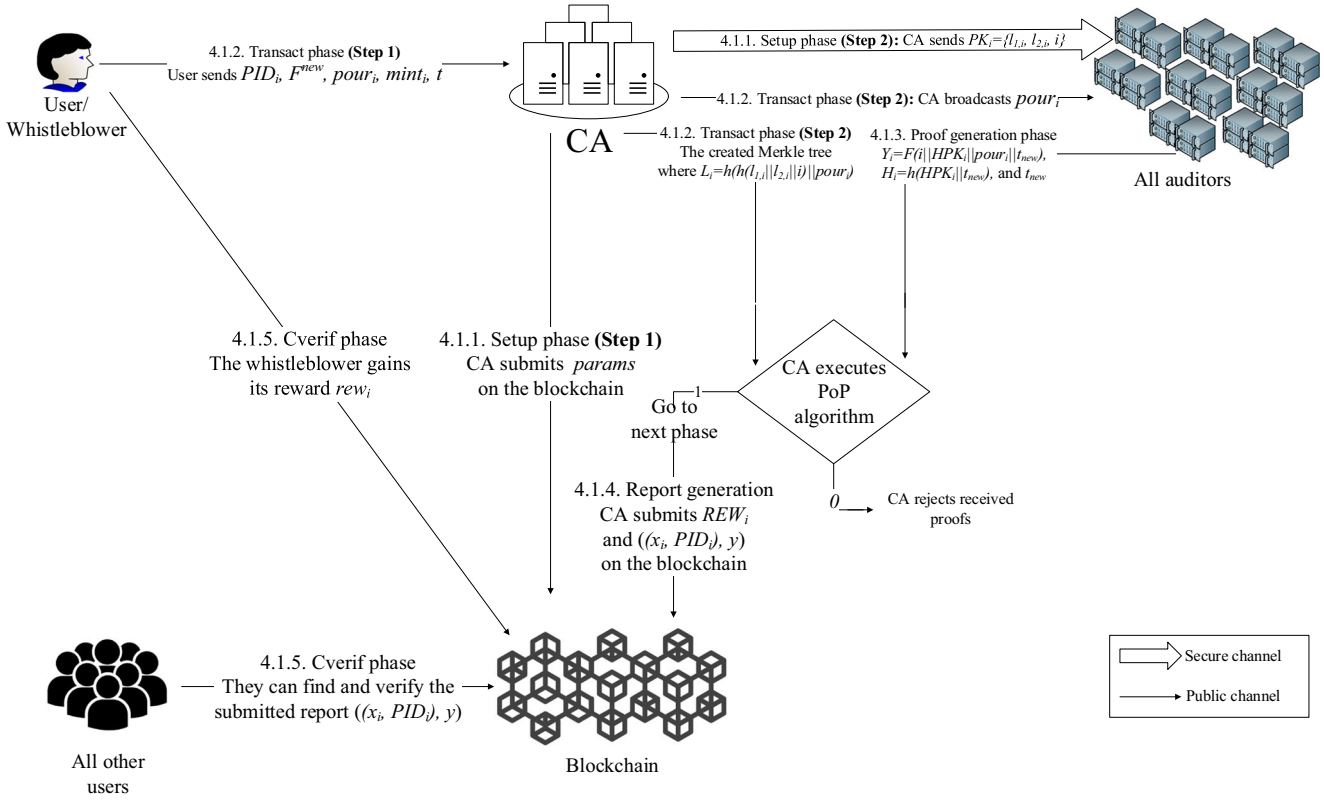

Fig. 3 The Quantum-Secure Reporting Protocol (QS−RP)

4.1.2 Transact Phase

This phase includes two steps; we show them in Table 3 (Algorithm 2) and describe in the following:

Table 2 Setup Phase

Algorithm 1: QS-RP setup

Step 1

(CA side)

1. $(params) \leftarrow Setup(\lambda)$
2. The *params* is submitted on the blockchain.
3. $(F, L_1, L_2) \leftarrow$ `KeyGen` (*params*)
4. $(\{ACA, ACB\}) \leftarrow$ `PoT`(auditor's *ID*)
5. $(KP1, KP2) \leftarrow$ `KeyAssig`(L_1, L_2)
6. $L_1 \leftarrow \{l_{1,1}, ..., l_{1,NPK}\}$
7. $L_2 \leftarrow \{l_{2,1}, ..., l_{2,NPK}\}$

Step 2

(CA PK_i Auditors)

8. $KP1 \leftarrow \{l_{1,1}, l_{2,1}, ..., l_{1,NPK/2}, l_{2,NPK/2}\}$
9. $KP2 \leftarrow \{l_{1,NPK/2+1}, l_{2,NPK/2+1}, ..., l_{1,NPK}, l_{2,NPK}\}$
10. $l_{i,1} \leftarrow kp1_i \in_R KP1$
11. $l_{i,2} \leftarrow kp2_i \in_R KP2$
12. $PK_i \leftarrow \{l_{1,i}, l_{2,i}, i\}$

– **Step 1:** The user U_i, as a whistleblower, generates a temporary set of the pair public-private keys ((F^{new}), (L_1^{new}, L_2^{new})) based on the MPKC. The `Transact` algorithm is then executed by U_i for taking ID_i, x_i, t, F, F^{new}, L_1^{new}, and L_2^{new} as inputs where x_i is the report and t is a time stamp. The `Transact` algorithm returns generated private report tuples $PID_i = h(ID_i||x_i||t)$, $pour_i = \hat{F}(L_1^{new} || L_2^{new} || PID_i)$, and $mint_i = \hat{F}^{new}(x_i || PID_i || t)$ where PID_i is the user's pseudonym and $pour_i$ and $mint_i$ are two ciphers.

Table 3 Transact Phase

Algorithm 2: Transact phase

Step 1

(U_i $PID_i, F^{new}, pour_i, mint_i, t$ CA)

1. Select (F^{new}), (L_1^{new}, L_2^{new})
2. $PID_i = h(ID_i||x_i||t)$
3. $pour_i = \hat{F}(L_1^{new} || L_2^{new} || PID_i)$
4. $mint_i = \hat{F}^{new}(x_i || PID_i || t)$

Step 2

(CA $pour_i$ Auditors' group)

For $i = 1 : NPK$

5. $L_i = h(h(l_{1,i}||l_{2,i}||i)||pour_i)$

end *For*

6. Create the original Merkle tree

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Then, U_i sends its pseudonym PID_i, new public key F^{new}, two ciphers $pour_i$ and $mint_i$, and t to CA through a public channel.

- **Step 2:** On receiving PID_i, F^{new}, $pour_i$, $mint_i$, and t, CA stores them as temporary parameters and creates Merkle tree by setting $L_i = h(h(l_{1,i} \| l_{2,i} \| i) \| pour_i)$ as the ith Mrkle tree's leaf. Then, CA broadcasts $pour_i$ among all auditors to verify.

4.1.3 Proof Generation Phase

As shown in Table 4 (Algorithm 3), each auditor, who wants to join the proof generation process, executes `ProofGen` algorithm to generate the verifying proof $Y_i = \hat{F}(i \| HPK_i \| pour_i \| t_{new})$. Then, the ith auditor sends the generated verifying proof $H_i = h(HPK_i \| t_{new})$, Y_i, and t_{new} to CA through a public channel.

According to Algorithm 3 shown in Table 4, upon receiving H_i, Y_i, and t_{new}, CA decrypts Y_i and sets $PK'_i = \mathrm{h}(\mathrm{HPK_i} \| \mathrm{pour_i})$ as the ith Merkle tree's leaf if $|t_{new} - t_{current}|$, and $H_i = h(HPK_i \| t_{new})$. It means that the auditor is authenticated. The CA then creates a new Merkle tree using PK'_i s and calculates $L_1^{new} \| L_2^{new} \| PID_i = L_2^{-1} \circ F^{-1} \circ L_1^{-1}(pour_i)$ and verifies $L_1^{new} \| L_2^{new} \| PID_i$ if `PoP` algorithm returns 1.

The CA recovers the report x_i by calculating $x_i \| PID_i \| t = L_2^{-1,new} \circ F^{-1,new} \circ L_1^{-1new}(mint_i)$. The report x_i is verified if the recovered t is the same t stored in *Transaction phase* as a temporary parameter, and $x_i \| PID_i \| t = L_2^{-1,new} \circ F^{-1,new} \circ L_1^{-1,new}(mint_i)$.

Table 4 Proof Generation Phase

Algorithm 3: Proof generation phase

(Auditors H_i, Y_i, t_{new} CA)
// Auditor side
1. $t_{new} \leftarrow$ Current time
2. $Y_i \leftarrow \hat{F}(i \| HPK_i \| pour_i \| t_{new})$
3. $H_i \leftarrow h(HPK_i \| t_{new})$
// CA side
4. If $|t_{new} - t_{current}|$ and $H_i = h(HPK_i \| t_{new})$ Go to 5
Else, terminate
For $i = 1:$ all received H_i, Y_i, t_{new}
5. $PK'_i = \mathrm{h}(\mathrm{HPK_i} \| \mathrm{pour_i})$
end *For*
6. Create a new Merkle tree
7. If (0) $\leftarrow$ `PoP`(PK'_i, *TH*, Original Merkle root) Go to 8
Else, terminate
8. $L_1^{new} \| L_2^{new} \| PID_i = L_2^{-1} \circ F^{-1} \circ L_1^{-1}(pour_i)$
9. $x_i \| PID_i \| t = L_2^{-1,new} \circ F^{-1,new} \circ L_1^{-1new}(mint_i)$

4.1.4 Report Generation Phase

The CA executes `RepGen` algorithm to calculate $Sign_{L_1,L_2}(x_i \| PID_i) = ((x_i \| PID_i), y)$ and $Sign_{L_1,L_2}(rew_i) = REW_i$ where rew_i is the assigned reward to the whistleblower if the verified report is valid and is not repetitive. Then, CA submits $((x_i \| PID_i), y)$ on the blockchain, and sends REW_i to the PID_i 's one-time public key/ address F^{new}.

Finally, CA removes stored parameters PID_i, F^{new}, L_1^{new}, L_2^{new}, $pour_i$, $mint_i$, and t from its temporary memory.

4.1.5 CVerif Phase

Each U_i can find the report set $((x_i \| PID_i), y)$ on the blockchain and can verify it if `CVerif` algorithm returns 1.

The user U_i, who sent the valid report (whistlrblower), can gain its reward using L_1^{new}, L_2^{new} from its one-time public key/ address F^{new}. Especially, it can verify the report's integrity if $PID_i = h(ID_i \| x_i \| t)$.

4.2 Analysis

This section shows in the ROM the `QS-RP` achieves the mentioned goals described in Section 3.5.2.

4.2.1 Report Confidentiality

We show the `QS-RP` provides report confidentiality.

Theorem 1 The `QS-RP` provides report confidentiality if the used one-way hash function is quantum-secure, and the used MPKC-based encryption algorithm provides report confidentiality against quantum computers.

Proof: According to *Game 1* described in Section 3.5.2, we show the `QS-RP` provides the report confidentiality.

- *Setup:* On having the security parameter λ, $\mathcal{C}$ executes *Setup* phase of the `QS-RP`, and gives *params* to $\mathcal{A}$.
- *Experiment:* The $\mathcal{A}$ submits polynomially bounded numbers of ciphers *mint* as queries to `ProofGen` oracle and collects all returned responses. For each sent query, $x \| ID \| t$ is returned to $\mathcal{A}$ by `ProofGen` oracle (submitting queries to `ProofGen` is equivalent to submitting queries to the decryption algorithm). The $\mathcal{A}$ stores all received responses.
- *Challenge:* The $\mathcal{C}$ selects a random challenge c and calculates $h_c = h(c)$ and $y_c = \hat{F}(c)$, and gives h_c and y_c to $\mathcal{A}$.
- *Guess:* The $\mathcal{A}$ wins if it guesses a valid value c^* such that $h(c^*) = h_c$ and $\hat{F}(c^*) = y_c$.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

To break report confidentiality of the QS−RP, $\mathcal{A}$ has to obtain the hidden report x_i from $PID_i = h(ID_i||x_i||t)$ or decrypt *min* $t_i = \hat{F}^{new}(x_i||PID_i||t)$. This process is equivalent to guess the valid value for c^* in *Game 1*. Therefore, to achieve that, $\mathcal{A}$ has to find a PPT algorithm to calculate $h^{-1}(h_c)$ or $L^{-1} \circ F^{-1} \circ L^{-1}(y_c)$. The $\mathcal{A}$ has ability to find the report x_i and break the report confidentiality of the QS−RP if it can find c^*.

As aforementioned, the used one-way hash function and the used MPKC-based encryption algorithm are quantum-secure. Therefore, the winning condition is not satisfied, and the advantage of $\mathcal{A}$ in *Game 1* against $\mathcal{C}$ is calculated as $ADV_{\mathcal{A}}^{Game1} = Pr[\mathcal{A}(\sqrt{}^{\dashv\nabla\dashv\Updownarrow\int}, y_c, h_c) = c \,|params, y_c, h_c]$.

Then, $\mathcal{A}$ fails in *Game 1*, and the QS−RP provides the report confidentiality since the used one-way hash function and the used MPKC-based encryption algorithm are quantum-secure.

4.2.2 C-SSMK

The sent report x should be confidential if some auditors cooperate.

Theorem 2 The QS−RP is *C-SSMK* if the used MPKC-based encryption algorithm provides the message confidentiality against quantum computers.

Proof We prove this theorem due to *Game 2* described in Section 3.5.2 to show that the QS−RP is C-SSMK, and $\mathcal{A}$ has a negligible advantage to win *Game 2*.

- *Setup:* The $\mathcal{C}$ executes *Setup*, KeyGen, and KeyAssig algorithms and gives *params* and $KP2 = \{l_{1,NPK/2+1}, l_{2,NPK/2+1}, ..., l_{1,NPK}, l_{2,NPK}\}$ to $\mathcal{A}$. The $\mathcal{C}$ keeps $KP1 = \{l_{1,1}, l_{2,1}, ..., l_{1,NPK/2}, l_{2,NPK/2}\}$ secure.
- *Experiment:* In this experiment, $\mathcal{A}$ works similar to *Experiment* phase in the proof of Theorem 1, and it submits polynomially bounded numbers of ciphers *mint* and identities ID_i as queries to ProofGen oracle and collects returned responses.

For each sent query, $x||ID||t$ is returned to $\mathcal{A}$ by ProofGen oracle.

The $\mathcal{A}$ stores all received responses.

// The *Challenge* step and *Guess* step of this proof are same steps in the proof of Theorem 1, but in this proof, $\mathcal{A}$ has $KP2 = \{l_{1,NPK/2+1}, l_{2,NPK/2+1}, ..., l_{1,NPK}, l_{2,NPK}\}$.

Having polynomially bounded numbers of y_c and c in *Game 2* is equivalent to have polynomially bounded numbers of *mint* and $x_i||PID_i||t$ in the QS−RP. Therefore, $\mathcal{A}$ has ability to decrypt *mint* in the QS−RP if it can decrypt y_c in *Game 2*, and the QS−RP is not C-SSMK if $\mathcal{A}$ wins *Game 2*.

As $\mathcal{A}$ has only the key pool $KP2$, and like *Game 1* it has to find a PPT algorithm to calculate $L^{-1} \circ F^{-1} \circ L^{-1}(y_c)$, furthermore, $KP2$ cannot get more help to it. Therefore, the advantage of $\mathcal{A}$ in *Game 2* is calculated as $ADV_{\mathcal{A}}^{Game2} = Pr[\mathcal{A}(\sqrt{}^{\dashv\nabla\dashv\Updownarrow\int}, y_c, h_c, KP2) = c \,|params, y_c, h_c, KP2]$.

Then, $\mathcal{A}$ fails in *Game 2*, and the QS−RP is C-SSMK since the used MPKC-based encryption algorithm is quantum-secure.

4.2.3 UF-SSMK

We show the QS−RP is UF-SSMK, and an invalid report is not submitted on the blockchain if malicious auditors cooperate.

Theorem 3 The QS−RP is UF-SSMK if the used MPKC-based signature algorithm provides unforgability against quantum computers.

Proof In this proof, the feature of UF-SSMK is proved. But the proof of this theorem is similar to the proof of Theorem 2. The proof of Theorem 2 shows the report confidentiality in encryption (see *Game 2*), and this proof shows the unforgeability in the applied MPKC-based signature.

- *Setup:* The $\mathcal{C}$ executes *Setup*, KeyGen, and KeyAssig algorithms and gives *params* and $KP2$ to $\mathcal{A}$. The $\mathcal{C}$ keeps $KP1$ secure.
- *Experiment:* Algorithms are executed in *Report generation* phase of the QS−RP are PoP and the MPKC-based signature. Therefore, submitting queries to RepGen oracle refers to submitting queries to an MPKC-based signature oracle. Regarding two approaches described in Section 3.5.2 for *Game 3*, $\mathcal{A}$ submits polynomially bounded numbers of queries to RepGen and ProofGen oracles, and it collects responses that generated as below:

- For each query RepGen oracles returns pair $((x, PID), y)$.
- The ProofGen oracles returns verifying proof Y_i, H_i, and t.

The $\mathcal{A}$ sends all received verifying proof Y_i, H_i, and t_i to $\mathcal{C}$, and stores received signature pairs $((x, PID), y)$.

- *Challenge:* The $\mathcal{C}$ executes PoP algorithm using received Y_i, H_i, and t from $\mathcal{A}$. Then, $\mathcal{C}$ gives the challenge c to $\mathcal{A}$ and asks it to create a valid signature on c.
- *Guess:* The $\mathcal{A}$ guesses a signature pair (c, y_c).

For the *first* approach, $\mathcal{A}$ cannot forge the used MPKC-based signature since it has not other parts of the private from $KP1$, and the used MPKC-based signature is quantum-secure.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

For the *second* approach, PoP algorithm returns 1 since lower than 50% of valid key parts are given to it as inputs.

Therefore, due to *params*, PoP algorithm, and $KP2$, the advantage of $\mathcal{A}$ against $\mathcal{C}$ in *Game 3* described in Section 3.5.2 is calculated as $ADV_{\mathcal{A}}^{Game3} = Pr[\mathcal{A}(\sqrt{}, c, KP2) = (y_c, c) \parallel 1 \leftarrow PoP\ |params, c, KP2]$.

Then, $\mathcal{A}$ fails in *Game 3*, and the QS−RP is UF-SSMK since the used MPKC-based signature algorithm is quantum-secure.

4.2.4 Privacy

In the following, we show that the QS−RP provides all defined features that provide the privacy.

- **Report confidentiality before report generation:** The report x_i cannot be recovered before report generation since decryption keys (L_1^{new}, L_2^{new}) are encrypted using the system's public key F. To recover them, auditors who have all parts of the system's private keys (auditor in both groups *ACA* and *ACB*) have to cooperate in the proof generation process to decrypt $mint_i$. Therefore, $\mathcal{A}$ or malicious auditor(s), who only has/have $KP2$, cannot recover the report x_i before the report generation process, and the report x_i be confidential before the report generation process since CA is assumed as a fully trusted-party and the MPKC-based encryption algorithm is quantum-secure.
- **User untraceability:** The $\mathcal{A}$ can find no link between a user if it had sent two different reports x_i and x'_i since there is no link between two outputs of a one-way hash function in such a way no link can be found between two pseudonyms $PID_i = h(ID_i \| x_i \| t)$ and $PID'_i = h(ID_i \| x'_i \| t')$. Therefore, user/ whistleblower U_i is untraceable since the used one-way hash function is quantum-secure, and no PPT $\mathcal{A}$, who has access to a quantum computer can find a collision for the used quantum-secure one-way hash function.
- **User anonymity:** As aforementioned all users can find the report set $(x_i \| PID_i, y)$ where $PID_i = h(U_i \| x_i \| t)$, and no one can obtain the real identity of the user U_i since the one-way hash function is quantum-secure, and no PPT $\mathcal{A}$, who has access to a quantum computer can find a the argument of hash function on having its output.
- **Secure address:** The reward rew_i is gained by someone who has the one-time private key (L_1^{new}, L_2^{new}). Therefore, no one except the user who has the private keys can gain the reward rew_i from the one-time public key/address $\hat{F}^{new}$ since the used MPKC cryptosystem is quantum-secure.

4.2.5 Security Against Common Attacks

We show that the QS−RP provides the security against common attacks such as insider attack, MitM attack, online attack, offline attack.

- **Insider attack:** In this attack, $\mathcal{A}$ has an access to *params*, $KP2$, and the set of $\{PK_i \| L_{1,i} \| L_{2,i} \| i\}$ where $NPK/2 + 1$. It tries to make a disorder in the process of decryption, signature, or PoP algorithm such that one of them return 0 We show that the used algorithms in the QS−RP return 1 if $\mathcal{A}$ be present among auditors in indexes of $NPK/2 + 1$ to NPK (we assumed in Section 3.5.2 Part 5, $\mathcal{A}$ can be present among auditor's group as malicious auditor). To make this attack, for $\mathcal{A}$ with the index of $NPK/2 + 1$, it calculates invalid verifying proofs $Y_i^* = \hat{F}(i \| HPK_i \| y^* \| t_{new})$ and $H_i = h(HPK_i \| t_{new})$. The $\mathcal{A}$ then sends generated values Y_i^*, H_i, and t_{new} to $\mathcal{C}$. On receiving all verifying proofs (valid and invalid), CA executes decryption, signature, and PoP algorithms. The $\mathcal{A}$ wins if one of decryption, signature, or PoP algorithms returns 0. It means that $L_2^{-1} \circ F^{-1} \circ L_1^{-1}(y) = \perp$, $sign_{L_1,L_2}(c) \neq y$, or (0) $\leftarrow$ PoP(PK_1 , ..., PK_{NPK}, TH, Merkle tree). As the key pool $KP2$ includes 50% of the private key's parts and the majority of received verifying proofs are valid, $\mathcal{A}$ cannot make the disorder successfully on mentioned algorithms. Therefore, the advantage of $\mathcal{A}$ is calculated as $ADV_{\mathcal{A}}^{InsiderAttack} = Pr[\mathcal{A} \parallel 0 \leftarrow \text{signature} \parallel 0 \leftarrow \text{decryption} \parallel \perp \leftarrow PoP | KP2, \{PK_i \| L_{1,i} \| L_{2,i} \| i\}\ |_{NPK/2+1}$.
- **MitM attack:** Like insider attack, $\mathcal{A}$ tries to make a disorder in the process of decryption, signature, or PoP algorithm such that one of the mentioned algorithms returns 0. But $\mathcal{A}$, as an outsider adversary, only has access to *params*. The $\mathcal{A}$ sends fake values $Y_i^* = \hat{F}(i^* \| HPK_i^* \| y^* \| t_{new})$, $HPK_i^* = h(HPK_i^* \| i^*)$, and t_{new} to CA through a public channel. The $\mathcal{A}$ wins if one of decryption, signature, or PoP algorithms returns 0. Like insider attack, the advantage of $\mathcal{A}$ is calculated as $ADV_{\mathcal{A}}^{MitM} = Pr \times [\mathcal{A} \parallel 0 \leftarrow \text{signature} \parallel 0 \leftarrow \text{decryption} \parallel 0 \leftarrow PoP] = Pr[0 \leftarrow PoP]$.
- **Online attack:** We show that the QS−RP is secure against the online attack, and no PPT $\mathcal{A}$ can recover the report x_i before the report generation process (this proof is similar to proofs of Theorems 1 and 2, but $\mathcal{A}$ is assumed on the existing public channel between auditors and CA). There is a need to have new private keys (L_1^{new}, L_2^{new}) to recover the report x_i on having $mint_i$. However, the set of new private keys (L_1^{new}, L_2^{new}) can be recovered after the proof generation process. Therefore, $\mathcal{A}$ cannot recover the set $x_i \| PID_i \| t$ since the used MPKC-based encryption algorithm is quantum-secure.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

- **Offline attack:** We show that the QS−RP is secure against the offline attack, and no one can change the submitted report set $((x_i \| PID_i), y)$. To change the submitted report set $((x_i \| PID_i), y)$, there are two approaches as follows:

  1. Regarding the immutability as the main feature of the blockchain that is selected as the QS−RP's database, all submitted transactions cannot be changed.
  2. If immutability is not assumed as the blockchain's feature, $\mathcal{A}$ has to forge the submitted signature pair $((x_i \| PID_i), y)$. But it cannot forge the applied MPKC-based signature algorithm since the used MPKC-based signature is quantum-secure.

Therefore, no one can change the submitted report set $((x_i \| PID_i), y)$.

4.2.6 Report integrity

Before the report generation, CA verifies the integrity of the report x_i if the recovered t is the same t stored in *Transact* phase, and if $mint_i = \hat{F}^{new}(x_i \| PID_i \| t)$. Moreover, the user U_i verifies whether or not its sent report x_i kept the integrity if $PID_i = h(ID_i \| x_i \| t)$. Therefore, the submitted report on the blockchain is the report sent by U_i; and U_i understand the fraud if x'_i is submitted on the blockchain instead of x_i.

5 Comparison and Evaluation

In this section, the QS−RP is compared with other related protocols that were proposed in recent years. It should be noted that there are limited blockchain-based reporting protocols and limited protocols related/similar to the QS−RP to compare with. Additionally, most of the centralized reporting protocols are focused on the routing methods and do not have goals similar to those of the present study, and it is therefore not possible to have more protocols in this section.

5.1 Feature

The general and security features of the QS−RP are compared in this section.

5.1.1 General

At first, an overview of some of the schemes described in Section 2 will be presented as in Table 6, some of which provided user anonymity such as [17, 18, 37, 38] and report confidentiality such as the one providing anonymous reputation [18], or the two schemes providing confidential transaction using ringCT [36, 38], which our scheme supports. The acronyms and notation related to this section are listed in Table 5. In this study, the three schemes [17, 18, 37] will be compared with the QS−RP since they have more similarities to the QS−RP where *Reportcoin* [17], BB2AR [37], and ARS-PS [18] provide user anonymity, and the ARS-PS scheme provides report confidentiality prior to confirmation using a ZKP.

5.1.2 Security

In this section, the security features of the QS−RP are compared with those of the three other blockchain-based protocols including *Reportcoin* and BB2AR as two reporting protocols [17, 37] and ARS-PS as an anonymous reputation protocol [18]. This comparison is shown in Table 7.

As discussed in Section 4.2, Table 7 shows that the QS−RP provides all required security features defined for a reporting protocol. In the following, the other items of Table 7 will be discussed.

- *Quantum-secure:* Both the two reporting protocols (*Reportcoin* and BB2AR) and the anonymous reputation protocol (ARS-PS) do not provide security against quantum computers since they apply cryptographic primitives designed based on the hardness of number-theory-based problems.
- *Whistleblower anonymity:* According to the proven security of the used ring signatures in *Reportcoin* and BB2AR and the security of the used randomizable signature and ZKP scheme in ARS-PS, they all provide provable whistleblower anonymity.
- *Report confidentiality:* Since it is clear that a typical ring signature does not provide message confidentiality, *Reportcoin* and BB2AR do not support report confidentiality. However, in ARS-PS, the used ZKP scheme proves the value of reputation (we assume that the reputation in ARS-PS is a report sent by anonymous users) without revealing it.
- *Secure address:* This security feature is not checkable in *Reportcoin* and ARS-PS since the rewarding policy is not assumed. Nevertheless, a provable method to provide a secure address is presented in BB2AR.
- *Insider attack security:* According to the assumption in this paper (see Section 3.5.2, Part 5), insider attack is implemented by network users who should cooperate in the reporting process (while they not cooperate). Therefore, the insider attacks are implementable in *reportcoin* and BB2AR, and ARS-PS provides security against this attack.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 5 The List of Acronyms and Notations

Acronym and Notation	Description
AC	All-time confidential (before confirmation)
EC	Elliptic curve
Enc	Encryption
H	Hash function
Lat	Lattice
MPKC	Multivariate public key cryptography
MT	Merkle tree
PN	Pseudonym
PQ	Post-quantum
QKD	Quantum key distribution
QS	Quantum-secure
RCT	RingCT
RS	Ring signature
S	Signature
SS	Secret sharing
ZKP	Zero-knowledge proof
$Cost_{Pair}$	Cost of pairing operation
$Cost_P$	Cost of power operation
$Cost_M$	Cost of multiplication operation
$Cost_{MPKC}$	Cost of MPKC-based encryption algorithm
$Cost_H$	Cost of hash function
n	Number of users in ring signature ($= 10$)
T_{Pair}	Execution time of bilinear pairing operation
T_P	Execution time of power/exponentiation operation
T_M	Execution time of multiplication operation
T_H	Execution time of hash function
✓	Provides the feature
×	Does not provide the feature
-	Is not checkable

5.2 Performance

In this section, the QS−RP's performance is evaluated and compared with other protocols. It should be said, the main effective item it the QS−RP's performance is that it works independently from the number of users n (a ring signature is typically used for user anonymity which its computational cost and communications overhead are depended on the number of users n). Additionally, the QS−RP does not use heavy cryptographic primitives like ZKP.

Table 6 The Comparison of Features

Scheme ⇒ Item ⇓	Reportcoin 2019 [17]	ARS-PS 2019 [18]	BB2AR 2019 [37]	H-based S 2018 [34]	PQ blockchain 2018 [35]	lat-based one-time RS 2018 [36]	*MatRiCT* 2019 [38]	PQ distributed ledger 2020 [39]	QS−RP (our scheme)
QS primitive	-	-	-	H	QKD	Lat	Lat	H	MPKC
Authentication technique	RS	S	RS	S	QKD	S	RS	S	MT
User anonymity technique	RS	ZKP	RS	-	-	-	RS	-	PN
Message/Report confidentiality technique	-	ZKP	-	-	-	AC/ RCT	AC/ RCT	-	AC/ Enc

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Table 7 The Security Comparison

Scheme ⇒ / Item ⇓	Reportcoin 2019 [17]	BB2AR 2019 [37]	ARS-PS 2019 [18]	QS−RP
Quantum-secure	×	×	×	✓
Whistleblower anonymity	✓	✓	✓	✓
Report confidentiality	×	×	✓	✓
C-SSMK	-	-	-	✓
Secure address	-	✓	-	✓
Insider attack security	×	×	✓	✓

5.2.1 User Cost

In the following, we show the cost of the phases that make an overhead to users in the QS−RP and the three mentioned protocols in Table 8 and describe below that those processes are done by users.

- *Reportcoin* 2019 [17]: In *Reportcoin*, a ring signature is applied to provide user anonymity (a typical ring signature does not give the message confidentiality). Each user has to create the ring signature by having a list of its neighbors' public keys, and there is a need to have the mentioned list to verify the ring signature. The overhead of creating and verifying the ring signature is related to the length of the list of public keys. Therefore, a higher overhead in anonymization and checking report integrity (verification of the ring signature) is caused by a larger public keys size.
- BB2AR 2019 [37]: The BB2AR scheme is similar to *Reportcoin* since it applies a ring signature to provide user anonymity in its reporting schemes. However, in the BB2AR protocol, the network verifier checks the received report's integrity and validity. Therefore, no overhead is imposed on network users since they do not check it.
- ARS-PS 2019 [18]: Users can purchase through an anonymous channel such as *Zerocash* [5] in the ARS-PS scheme. After that, the rate (we assume it as a private report) related to the retailer is submitted using an anonymous token by ZKP. Then other users who want to purchase somethings from the same retailer can find the submitted rating token related to the mentioned retailer and verify the rate.
- QS−RP: In the presented QS−RP, the whistleblower's pseudonym (user U_i) is related to the sent report, and only the whistleblower can verify the report integrity. The QS−RP imposes a lower overhead than the three other protocols to its users. Each user calculates a hash function just once to anonymize and check the report integrity (we ignore the cost of hash functions since their cost is low). Also, it computes two encrypted reports using two rounds of executing the encryption algorithm based on MPKC.

Table 9 Comparison of Communication Overhead for Whistleblower (parameter)

Scheme ⇒ / Item ⇓	*Reportcoin* 2019 [17]	BB2AR 2019 [37]	ARS-PS 2019 [18]	QS−RP
Off-chain Overhead	$4+n$	$5+n$	17	5
On-chain Overhead	$4+2n$	0	0	0

5.2.2 Communication

The whistleblower's communication overhead includes on-chain and off-chain overhead comparison as shown in Table 9. In the following, Table 9 will be discussed briefly.

- *Off-chain overhead:* To send a report, the whistleblower has to send some parameters such as ring signature tuples, ZKP tuples, or a confidential report through a public channel. According to the used schemes in discussed protocols, whistleblowers in *Reportcoin* and BB2AR send the created ring signatures' tuples which the corresponded tuples are related to the number of user in the used ring signature (n). The costumer in ARS-PS

Table 8 The Comparison of User Side Cost

Scheme ⇒ / Item ⇓	Reportcoin 2019 [17]	BB2AR 2019 [37]	ARS-PS 2019 [18]	QS−RP
Base of calculation	EC	EC	EC	MPKC
Anonymization	$n(2Cost_M + 1Cost_H)$	$2Cost_M + 2nCost_H + 4nCost_M$	Using anonymous channel (e.g., tor)	$1Cost_H$
Report confidentiality	Not supported	Not supported	$2Cost_{Pair} + 8Cost_P + 2Cost_M + 1Cost_H$	$2Cost_{MPKC}$
Report integrity	$n(2Cost_M + 1Cost_H)$	The verifier checks	$2Cost_{Pair} + 12Cost_P + 5Cost_M + 3Cost_H$	$1Cost_H$

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

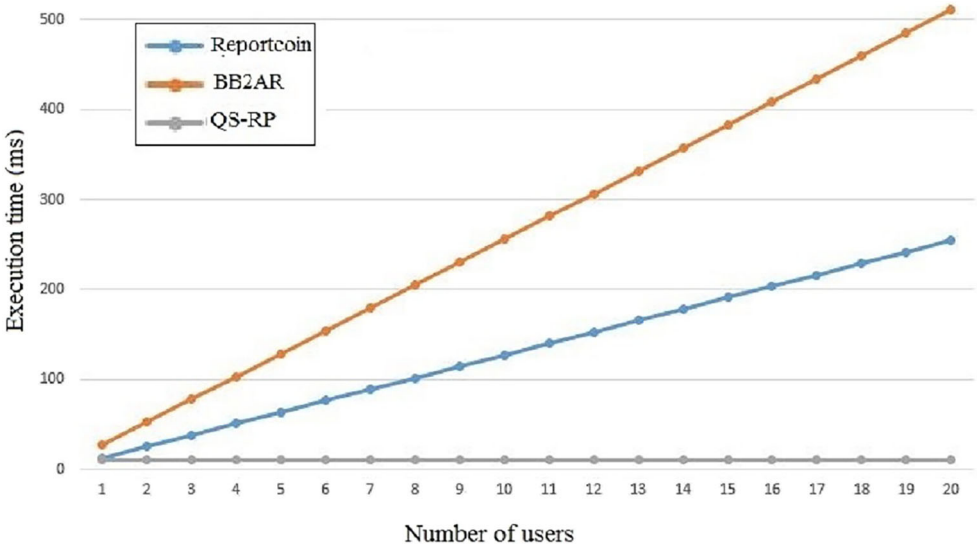

Fig. 4 The Comparison of Execution Time for User Anonymization

protocol sends the ZKP proofs to a retailer (7), and it then sends the verified tuples to the IDM (10). In the QS−RP, 5 parameters are sent by the whistleblower. To have a numerical comparison, the number of present users is assumed 10. According to this assumption, 14, 15, and 17 parameters are sent off-chain in the three mentioned protocols, whereas only 5 parameters are sent in the QS−RP. Therefore, we can say that the QS−RP is about 66% efficient than those mentioned protocols in communication overhead.

– *On-chain overhead:* According to *Reportcoin*'s concrete protocol, the whistleblower has to submit the created ring signature on the blockchain as a transaction. However, in other compared protocols (and QS−RP), no parameter is submitted by the whistleblower (the report is submitted on the blockchain by the system's central authority).

5.2.3 Timing

In this section, the execution time for users' anonymization methods (applying ring signatures or using ZKP schemes) and checking report integrity are compared. According to [45], the cryptographic primitives execution time for a user who has a smartphone with a Hisilicon Kirin 925 2.45-GHz processor, Android 4.4.2, and 3-GB memory is listed in the followings:

$$aligned T_{Pair} = 361.282 ms T_P = 200.67 ms T_M = 0.731 ms T_H = 11.26 msaligned$$

As creating new pseudonyms (or anonymization process) and checking report integrity should be executed for each report, the execution time for these two mentioned processes will be discussed below:

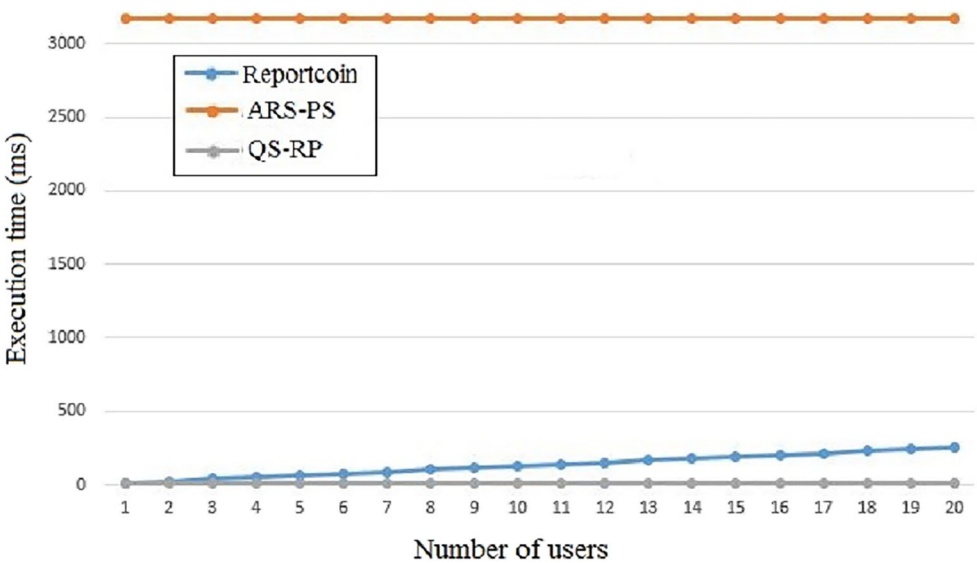

Fig. 5 The Comparison of Execution Time for Checking Report Integrity

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

- *Anonymization:* The comparison of execution time for user anonymization between *Reportcoin* [17] and BB2AR [37] with the QS−RP is shown in Fig 4. As previously mentioned in Section 2, to provide user anonymity, a ring signature is used in *Reportcoin* [17] and BB2AR [37]. Therefore, each user in *Reportcoin*/BB-2AR has to generate a ring signature to keep its identity private. Using a ring signature is the main item in execution time in that the execution time is directly related to the type of ring signature and the number of present users n in the ring. As shown in Fig 4 the execution time in *Reportcoin* and BB-2AR is linearly increased by increasing the number of users ($n(2T_M + 1T_H)$ for *Reportcoin* and $2T_M + 2nT_H + 4nT_M$ for BB2AR). However, the execution time in the QS−RP is a constant value for the user ($1T_H = 11.26\ ms$). To have a numerical example with the previous assumption ($n = 10$), we can say the QS−RP is 90% and 95% faster than *Reportcoin* and BB2AR, respectively.
- *Checking integrity:* The comparison of execution time for checking report integrity between *Reportcoin* [17] and ARS-PS [18] with the QS−RP is shown in Fig 5. Similar to the anonymization's execution time, the execution time for checking report integrity in *Reportcoin* is linearly increased by increasing the number of users ($n(2T_M + 1T_H)$) and the QS−RP is 90% faster than *Reportcoin*. But in ARS-PS, report integrity is taken as a constant time of 3168 *ms* ($2T_{Pair} + 12T_P + 5T_M + 3T_H$) so that it is 280 times bigger than QS−RP.

6 Conclusion

This paper presented a blockchain-based quantum-secure reporting protocol using the MPKC called QS−RP. To provide transparency and immutability, blockchain was selected as the system's infrastructure. The QS−RP provided several prominent features such as report confidentiality and integrity, user anonymity and untraceability, and the security against quantum computers. It was also proved that the QS−RP is secure against common attacks and provided C-SSMK and UF-SSMK. Finally, a comparison was provided to show that the QS−RP is more efficient than other recently-proposed reporting protocols.

Future scope: In the end, a number of paths that can be taken for future studies will be provided.

- Dynamic auditor selection is for creating decryption allowance for the private report of an idea that can prevent the presence of malicious auditors and increase the reporting protocol's reliability.
- Applying a byzantine-based consensus method for reporting protocols can be an attractive idea that can provide high reliability for reporting protocols among many malicious users.
- A combination of lattice-based cryptography [46, 47] and secret sharing can be applied to quantum-secure reporting systems so that some semi-trusted parties cooperate to reconstruct the report.
- Using attribute-based cryptography [48, 49] is another idea that can solve the mentioned problem since it can be used as a threshold system in such a way that authorities or privileged insiders, who have the determined attributed can obtain the sent report.
- Users (auditors) who have a private key can learn a unique function of the encrypted data. Therefore, functional encryption [50, 51] can be designed so that all things about the encrypted message can be fully recovered if all users (or some of them) are present.

Acknowledgements We as authors appreciate anonymous reviewers for their valuable comments on this work.

Declarations

Conflicts of Interest Saeed Banaeian Far declares that he has no conflict of interest. Maryam Rajabzadeh Asaar declares that she has no conflict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors

References

1. Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic system (2008). (2008).
2. Lin, Iuon-Chang, and Tzu-Chun Liao. "A survey of blockchain security issues and challenges." IJ Network Security 19, no. 5 (2017): 653–659.
3. Kaur, Avinash, Anand Nayyar, and Parminder Singh. "BLOCKCHAIN: A PATH TO THE FUTURE." Cryptocurrencies and Blockchain Technology Applications (2020): 25–42.
4. Na Shi, Liang Tan, Wenjuan Li, Xin Qi, Keping Yu, A blockchain-empowered AAA scheme in the large-scale HetNet, Digital Communications and Networks, 2020, ISSN 2352-8648, https://doi.org/10.1016/j.dcan.2020.10.002.
5. E. Ben Sasson et al., "Zerocash: Decentralized Anonymous Payments from Bitcoin," 2014 IEEE Symposium on Security and Privacy, San Jose, CA, 2014, pp. 459–474, doi: https://doi.org/10.1109/SP.2014.36.
6. Ruffing, T., Moreno-Sanchez, P., & Kate, A. (2014, September). Coinshuffle: Practical decentralized coin mixing for bitcoin. In European Symposium on Research in Computer Security (pp. 345-364). Springer, Cham.
7. Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J. A., & Felten, E. W. (2014, March). Mixcoin: Anonymity for bitcoin with accountable mixes. In International Conference on Financial

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Cryptography and Data Security (pp. 486-504). Springer, Berlin, Heidelberg.
8. C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao and K. Yu, "AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud," in IEEE Access, vol. 8, pp. 70604–70615, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2985762.
9. Feng, Chaosheng, Keping Yu, Moayad Aloqaily, Mamoun Alazab, Zhihan Lv, and Shahid Mumtaz. "Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV." IEEE Transactions on Vehicular Technology 69, no. 11 (2020): 13784–13795.
10. M. Li, L. Zhu and X. Lin, "Efficient and Privacy-Preserving Carpooling Using Blockchain-Assisted Vehicular Fog Computing," in IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4573–4584, June 2019, doi: https://doi.org/10.1109/JIOT.2018.2868076.
11. L. Li et al., "CreditCoin: A Privacy-Preserving Blockchain-Based Incentive Announcement Network for Communications of Smart Vehicles," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 7, pp. 2204–2220, July 2018, doi: https://doi.org/10.1109/TITS.2017.2777990.
12. Zhang, A., Lin, X. Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain. J Med Syst 42, 140 (2018). doi: 10.1007/s10916-018-0995-5.
13. K. Yu, L. Tan, X. Shang, J. Huang, G. Srivastava and P. Chatterjee, "Efficient and Privacy-Preserving Medical Research Support Platform Against COVID-19: A Blockchain-Based Approach," in IEEE Consumer Electronics Magazine, https://doi.org/10.1109/MCE.2020.3035520.
14. K. -P. Yu, L. Tan, M. Aloqaily, H. Yang and Y. Jararweh, "Blockchain-Enhanced Data Sharing with Traceable and Direct Revocation in IIoT," in IEEE Transactions on Industrial Informatics, https://doi.org/10.1109/TII.2021.3049141.
15. C. Feng, etc, "Efficient and Secure Data Sharing for 5G Flying Drones: A Blockchain-Enabled Approach", IEEE Network, https://doi.org/10.1109/MNET.011.2000223.
16. N. Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams," in IEEE Transactions on Dependable and Secure Computing, vol. 15, no. 5, pp. 840-852, 1 Sept.-Oct. 2018, https://doi.org/10.1109/TDSC.2016.2616861.
17. S. Zou, J. Xi, S. Wang, Y. Lu and G. Xu, "Reportcoin: A Novel Blockchain-Based Incentive Anonymous Reporting System," in IEEE Access, vol. 7, pp. 65544–65559, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2915956.
18. D. Liu, A. Alahmadi, J. Ni, X. Lin and X. Shen, "Anonymous Reputation System for IIoT-Enabled Retail Marketing Atop PoS Blockchain," in IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3527–3537, June 2019, doi: https://doi.org/10.1109/TII.2019.2898900.
19. H. Wang, Q. Wang, D. He, Q. Li and Z. Liu, "BBARS: Blockchain-Based Anonymous Rewarding Scheme for V2G Networks," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 3676–3687, April 2019, doi: https://doi.org/10.1109/JIOT.2018.2890213.
20. Shor, P. W. (1994, November). Algorithms for quantum computation: discrete logarithms and factoring. In Proceedings 35th annual symposium on foundations of computer science (pp. 124-134). Ieee.
21. Chen, L., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2016). NISTIR 8105 Report on Post-Quantum Cryptography. National Institute of Standards and Technology, 10.
22. Merkle, Ralph C. "A digital signature based on a conventional encryption function." Conference on the theory and application of cryptographic techniques. Springer, Berlin, Heidelberg, 1987.
23. Bahri, L., & Girdzijauskas, S. (2018, April). When trust saves energy: a reference framework for proof of trust (PoT) blockchains. In Companion Proceedings of the The Web Conference 2018 (pp. 1165-1169). https://dl.acm.org/doi/abs/10.1145/3184558.3191553
24. J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun and L. Li, "A Proof-of-Trust Consensus Protocol for Enhancing Accountability in Crowdsourcing Services," in IEEE Transactions on Services Computing, vol. 12, no. 3, pp. 429-445, 1 May-June 2019, https://doi.org/10.1109/TSC.2018.2823705.
25. Stumpf, Frederic, Omid Tafreschi, Patrick Rder, and Claudia Eckert. "A robust integrity reporting protocol for remote attestation." In Proceedings of the Workshop on Advances in Trusted Computing (WATC), p. 65. 2006.
26. Diffie, Whitfield, and Martin Hellman. New directions in cryptography. IEEE transactions on Information Theory 22, no. 6 (1976): 644–654.
27. Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 21, no. 2 (1978): 120–126.
28. Choi, H., Enck, W., Shin, J. et al. ASR: anonymous and secure reporting of traffic forwarding activity in mobile ad hoc networks. Wireless Netw 15, 525–539 (2009). doi: 10.1007/s11276-007-0067-0.
29. X. Liu, H. Zhao, X. Yang and X. Li, SinkTrail: A Proactive Data Reporting Protocol for Wireless Sensor Networks, in IEEE Transactions on Computers, vol. 62, no. 1, pp. 151–162, 2013, doi: 10.1109/TC.2011.207.
30. Carolina Tripp Barba, Luis Urquiza Aguiar, Monica Aguilar Igartua, Javier Parra-Arnau, David Rebollo-Monedero, Jordi Forn, Esteve Pallar's, A collaborative protocol for anonymous reporting in vehicular ad hoc networks, Computer Standards & Interfaces, Volume 36, Issue 1, 2013, Pages 188-197, ISSN 0920-5489, https://doi.org/10.1016/j.csi.2013.06.001. (http://www.sciencedirect.com/science/article/pii/S0920548913000615)
31. H. Li, G. Din and K. Nahrstedt, "Lynx: Authenticated anonymous real-time reporting of electric vehicle information," 2015 IEEE International Conference on Smart Grid Communications (SmartGridComm), Miami, FL, 2015, pp. 599-604, https://doi.org/10.1109/SmartGridComm.2015.7436366.
32. J. Kamel, I. Ben Jemaa, A. Kaiser and P. Urien, Misbehavior Reporting Protocol for C-ITS, 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 2018, pp. 1–4, doi: 10.1109/VNC.2018.8628407.
33. Li, Y., Zhao, Y., Ishak, S. et al. An anonymous data reporting strategy with ensuring incentives for mobile crowd-sensing. J Ambient Intell Human Comput 9, 2093–2107 (2018). doi: 10.1007/s12652-017-0529-x.
34. Buldas, A., Laanoja, R., & Truu, A. (2018, November). A blockchain-assisted hash-based signature scheme. In Nordic Conference on Secure IT Systems (pp. 138-153). Springer, Cham.
35. Kiktenko, E. O., Pozhar, N. O., Anufriev, M. N., Trushechkin, A. S., Yunusov, R. R., Kurochkin, Y. V., & Fedorov, A. K. (2018). Quantum-secured blockchain. Quantum Science and Technology, 3(3), 035004.
36. Torres, W. A. A., Steinfeld, R., Sakzad, A., Liu, J. K., Kuchta, V., Bhattacharjee, N., ... & Cheng, J. (2018, July). Post-quantum one-time linkable ring signature and application to ring confidential transactions in blockchain (lattice RingCT v1. 0). In Australasian Conference on Information Security and Privacy (pp. 558-576). Springer, Cham.
37. H. Wang, D. He, Z. Liu and R. Guo, Blockchain-Based Anonymous Reporting Scheme With Anonymous Rewarding, in IEEE Transactions on Engineering Management, vol. 67, no. 4, pp. 1514–1524, 2020, doi: 10.1109/TEM.2019.2909529.
38. Esgin, M. F., Zhao, R. K., Steinfeld, R., Liu, J. K., & Liu, D. (2019, November). MatRiCT: efficient, scalable and post-quantum

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

blockchain confidential transactions protocol. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (pp. 567-584).
39. Furqan Shahid, Abid Khan, Gwanggil Jeon, Post-quantum distributed ledger for internet of things, Computers & Electrical Engineering, Volume 83, 2020, 106581, ISSN 0045–7906, doi: https://doi.org/10.1016/j.compeleceng.2020.106581.(http://www.sciencedirect.com/science/article/pii/S004579061932659X).
40. Naor, Moni, and Moti Yung. "Universal one-way hash functions and their cryptographic applications." Proceedings of the twenty-first annual ACM symposium on Theory of computing. 1989.
41. Czypek, Peter. "Implementing Multivariate Quadratic Public Key Signature Schemes on Embedded Devices." Diss. Ph. D. thesis, Diploma Thesis, Chair for Embedded Security, RUB (2012).
42. Wolf, Christopher. "Multivariate Quadratic Polynomials in Public Key Cryptography." IACR Cryptology ePrint Archive 2005 (2005): 393.
43. X. Shen, L. Wang, H. Zhu and Y. Liu, A Multivariate Public Key Encryption Scheme With Equality Test, in IEEE Access, vol. 8, pp. 75463–75472, 2020, doi: 10.1109/ACCESS.2020.2988732.
44. Lu, Gang, et al. "Cryptanalysis of Novel Extended Multivariate Public Key Cryptosystem with Invertible Cycle." IJ Network Security 20.3 (2018): 509-514.
45. Kumar, Vinod, Musheer Ahmad, Adesh Kumari, Saru Kumari, and M. K. Khan. "SEBAP: A secure and efficient biometric's assisted authentication protocol using ECC for vehicular cloud computing." International Journal of Communication Systems (2019): e4103.
46. Babai, L. On Lovisz's lattice reduction and the nearest lattice point problem. Combinatorica 6, 1–13 (1986). doi: 10.1007/BF02579403.
47. Regev, Oded. "Lattice-based cryptography." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2006.
48. Waters, Brent. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2011.
49. Lewko, Allison, and Brent Waters. Decentralizing attribute-based encryption. Annual international conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 2011.
50. Lewko, Allison, et al. "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption." Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2010.
51. Boneh, Dan, Amit Sahai, and Brent Waters. "Functional encryption: Definitions and challenges." Theory of Cryptography Conference. Springer, Berlin, Heidelberg, 2011.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Content courtesy of Springer Nature, terms of use apply. Rights reserved.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com