Journal of Quantum Computing JQC, vol.1, no.2, pp.49-63, 2019

Quantum Blockchain: A Decentralized, Encrypted and Distributed Database Based on Quantum Mechanics

Chuntang Li[1], Yinsong Xu[1], Jiahao Tang[1] and Wenjie Liu[1, 2, *]

Abstract: Quantum blockchain can be understood as a decentralized, encrypted and distributed database based on quantum computation and quantum information theory. Once the data is recorded in the quantum blockchain, it will not be maliciously tampered with. In recent years, the development of quantum computation and quantum information theory makes more and more researchers focus on the research of quantum blockchain. In this paper, we review the developments in the field of quantum blockchain, and briefly analyze its advantages compared with the classical blockchain. The construction and the framework of the quantum blockchain are introduced. Then we introduce the method of applying quantum technology to a certain part of the general blockchain. In addition, the advantages of quantum blockchain compared with classical blockchain and its development prospects are summarized.

Keywords: Quantum computation, quantum blockchain, digital currency.

1 Introduction

Blockchain [Swan (2015)] is a new application mode of computer technology such as distributed data storage, point-to-point transmission, consensus mechanics, and encryption algorithm. As the underlying technology of Bitcoin, it is essentially a decentralized database. Unlike the general distributed storage, each node in the blockchain stores a copy of the blockchain database. Blockchain has the characteristics of decentralization, openness, and the information stored in it cannot be tampered with, so it has a wide range of applications in digital currency, information security industry, and smart contracts.

In the distributed network of blockchain, the communication and trust between nodes need to rely on digital signature technology, which mainly realizes the identification and the authenticity and integrity verification of information. The Elliptic Curves Cryptography (ECC) or the large integer factorization problem (RSA) is the encryption algorithm used in most blockchain digital signature technology. The security of these algorithms is based on the assumption of computational complexity for certain mathematical problems. When adding new data to the blockchain, the proof-of-work

[1] School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

[2] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, 210044, China.

[*] Corresponding Author: Wenjie Liu. Email: wenjiel@163.com.

JQC. doi:10.32604/jqc.2019.06715

algorithm is often used. The main task in this process is to guess a value (nonce) by calculation and solve the prescribed hash problem. However, the development of quantum computation [Nielsen and Chuang (2011)] will pose a potential risk to classical blockchain technology. Some quantum algorithms can effectively solve the problems corresponding to the encryption algorithm, making the digital signature in the blockchain become unsafe. For example, Shor's algorithm [Shor (1994)] can solve the integer factorization problem in polynomial time, which threatens the security of the RSA encryption algorithm. In addition, Grover's search algorithm can speed up an unstructured search problem quadratically, so in the process of mining, the hash value satisfying the conditions can be calculated by the quantum computer in a shorter time. This means that the parties that can acquire the quantum computer have an unfair advantage in obtaining the mining reward. In short, Quantum computers can solve some NP-hard problems faster than classical computers, so many researchers focus on the research of quantum blockchain, which is a combination of quantum technology and blockchain technology.

The rest of this paper is organized as follows: In Section 2, we briefly review the classical blockchain including the key technologies used in blockchain and the design of blockchain. In Section 3, the research status and progress of quantum blockchain in recent years is introduced. We briefly analyze the performance of the quantum blockchain and conclude this paper in Section 4.

2 Preliminaries

2.1 Key technologies used in blockchain

Hash Algorithm: The hash algorithm can transform an input value of arbitrary length into a binary value of fixed length. This binary value is called hash value, which can be used to verify the integrity of the data. The famous Proof-of-Work algorithm is the application of the hash algorithm. The hash value of the data is stored in the block of the blockchain. In addition, the signature commonly used in blockchains is also generated by hashing the private key and the data that needs to be signed.

Proof-of-Work: Proof-of-Work (POW) can be simply understood as a proof that you have done a certain amount of work. In a blockchain system, any node that wants to generate a new block and write it to the blockchain must resolve the POW puzzle in the blockchain network. POW puzzle is an NP-hard problem. Nodes that calculate and solve the POW puzzle can often get cryptocurrency as rewards. The difficulty value in the POW is an important reference for miners in mining, and it determines how many hash operations miners need to run to produce a valid block. During the mining process, the difficulty value can be dynamically adjusted according to the computing power in the whole blockchain network. In the Bitcoin system, the difficulty value is set to rule that the new block generation rate is maintained at 10 minutes regardless of the mining ability.

Timestamp: The blockchain system uses the timestamp to prove that the transaction did occur at this moment. Therefore, the ownership of the currency in the transaction has been transferred, and the previous owner cannot use the currency again. In addition, each block is also stamped with the correct timestamp to form a correct linked list in chronological order.

2.2 Design of blockchain

Structure of Bockchain: The blockchain system consists of a number of data blocks, which have neatly arranged records (transaction). Each block contains a timestamp, a hash value of its content and the hash value of the previous block. The blockchain is formed by linking each block with the hash value. Each block is generated after the previous block in chronological order. Once the block is confirmed to be valid, it can hardly be modified. The schematic of the classical blockchain is shown in Fig. 1.

Network of Blockchain: Nakamoto has depicted the steps to run the network of blockchain in Nakamoto [Nakamoto (2008)]:

Step 1: New transactions are broadcast to all nodes of the network.

Step 2: Each node collects new transactions into a block.

Step 3: Each node runs the POW algorithm for its block.

Step 4: When a node solves POW puzzle, it broadcasts the block to all nodes.

Step 5: Other nodes accepts the block only if all of the transactions are valid and unused.

Step 6: Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

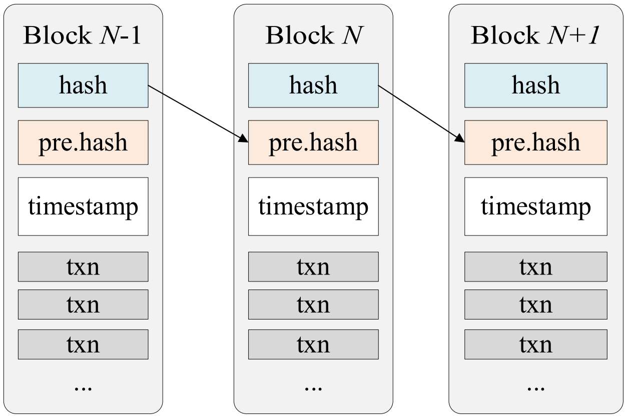


Figure 1: The schematic of the classical blockchain (“txn” stands for “transaction”)

3 Review of quantum blockchain

3.1 Construction of quantum blockchain

In 2018, Rajan et al. [Rajan and Visser (2018)] proposed a construction scheme of a quantum blockchain using entanglement in time [Rajan and Visser (2018)]. Entanglement in time means that microscopic particles such as photons that have never coexisted can also be entangled. Now we are going to introduce the construction of a quantum blockchain.

3.1.1 The construction of quantum chain

The notion of the chain is captured by the inseparability (entanglement) of quantum systems such as photons, and the blockchain is encoded as the GHZ (Greenberger-Horne-

Zeilinger) state [Carvacho, Graffitti, D'Ambrosio et al. (2017)] of the photons that have never coexisted.

In the conceptual design of this quantum blockchain, the data represented in the classical block is simplified into a string of two bits. The encoding procedure converts the record of each block, say r_1r_2, into a temporal Bell state generated at a specific time such as t=0:

$$|\beta_{r_1r_2}\rangle^{0,\ \tau} = \frac{1}{\sqrt{2}}(|0^0\rangle|r_2^{\ \tau}\rangle+(-1)^{r_1}|1^0\rangle|\overline{r_2}^{\tau}\rangle), \tag{1}$$

The superscript in kets indicates when the photons are absorbed, which provides a way to make timestamps in the blockchain. In particular, the first photon of a block is absorbed immediately.

When records are generated, the system encodes them into a temporal Bell state. These photons are then created and absorbed at their respective times. A specific example of such blocks would be:

$$|\beta_{00}\rangle^{0,\ \tau},\ |\beta_{10}\rangle^{\tau,2\tau},|\beta_{11}\rangle^{2\tau,3\tau}, \tag{2}$$

In order to realize design of the quantum blockchain, the bit strings of the bell state need to be linked together in chronological order using an entanglement in time. This link is implemented using a fusion process [Megidish, Shacham, Halevy et al. (2012)] in which temporal Bell states are recursively projected into a growing temporal GHZ state. The state of the quantum blockchain at $t\ =\ n\tau$ (starting at t = 0) is given below:

$$|GHZr_1r_2\ ...r_{2n}\rangle^{0,\tau,\tau,2\tau,2\tau...(n-1)\tau,(n-1)\tau,n\tau}=\frac{1}{\sqrt{2}}(|0^0r_2^{\ \tau}r_3^{\ \tau}...r_{2n}^{\ \ n\tau}\rangle+(-1)^{r_1}|1^0\overline{r_2}^{\tau}\overline{r_3}^{\tau}...\overline{r}_{2n}^{\ n\tau}\rangle) \tag{3}$$

Here the subscripts denote the concatenated string of all the blocks and superscripts refer to the time stamps. As an example of dynamic linking, let us now consider the first two blocks $|\beta_{00}\rangle^{0,\ \tau}$ and $|\beta_{10}\rangle^{\tau,2\tau}$. The system will create the blockchain $|GHZ_{0010}\rangle^{0,\tau,\tau,2\tau}$. Concatenating the third block then produces $|GHZ_{001011}\rangle^{0,\tau,\tau,2\tau,2\tau,3\tau}$. The classical information, $r_1r_2\ ...r_{2n}$ from the state (3) can be extracted during the decoding process. It was shown that the decoding process could be accomplished without measuring the full photon statistics or even detecting them in Megidish et al. [Megidish, Halevy, Pilnyak et al. (2017)].

3.1.2 Quantum blockchain network

In the quantum blockchain network, each node stores a copy of the blockchain. As in the classical blockchain, the aim of this stage is to add valid blocks in a decentralized manner. To construct the quantum blockchain network, the problem is that the network may consist of dishonest nodes and the generated blocks may come from a dishonest source.

The θ-protocol [McCutcheon, Pappa, Bell et al. (2016)] can be taken in the quantum network to verify the block generated from the untrusted source. The case studied by Rajan et al. [Rajan and Visser (2018)] of the verification protocol here is that the newly generated blocks are spatial bell states, and converting this to the related temporal case needs further study. More importantly, this is achieved in a decentralized manner by using other network nodes, which can also be dishonest.

First, we can use a quantum random number generator to pick a randomly chosen verifier node. Then the untrusted source shares a possible valid block, an n-qubit state. The untrusted source distributes each of the qubits to each node, j for verification. Now the verifying node generates random angles $\theta_j \in [0,\pi)$, where $\sum_j \theta_j$ is a multiple π. The angles are distributed to each node, including the verifier. Then they measure their qubit respectively in the basis,

$$|+\theta_j\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta_j}|1\rangle) \ , \tag{4}$$

$$|-\theta_j\rangle = \frac{1}{\sqrt{2}}(|0\rangle - e^{i\theta_j}|1\rangle) . \tag{5}$$

The results of each node $Y_j \in \{0,1\}$ is sent to the verifier. The necessary condition

$$\oplus_j Y_j = \frac{1}{\pi}\sum_j \theta_j (\mathrm{mod}\, 2) \tag{6}$$

is satisfied with probability 1 if the n-qubit state was a valid block, i.e., a spatial GHZ state. The verification protocol can effectively verify whether the newly generated block is valid.

3.2 Framework of quantum blockchain

In quantum blockchain, the process from the creation of transactions to the recording of information into the blockchain is important. Here, we mainly introduce the transaction process in the quantum blockchain by combining the framework of quantum-enhanced logic-based blockchain (QLB) proposed by Sun et al. [Sun, Wang, Kulicki et al. (2018)].

3.2.1 Overview of quantum-enhanced logic-based blockchain

In fact, QLB is an improvement and application of quantum-secured blockchain (QB) [Kiktenko, Pozhar, Anufriev et al. (2018)]. Improvements to the quantum-secured blockchain are achieved by replacing the classical Byzantine agreement protocol with the quantum honest-success Byzantine agreement protocol and incorporating quantum protection and quantum certificates into the syntax of transactions. The cryptocurrency created and transmitted in this blockchain is called qulogicoin. The quantum-secured blockchain developed by Kiktenko et al. [Kiktenko, Pozhar, Anufriev et al. (2018)] is the starting point of the research. Because of the use of quantum technology, QB is safer in a sense and can be immune to quantum computer attacks. A new quantum Byzantine agreement (QBA) protocol was developed to replace the classical Byzantine agreement in QB to improve the efficiency of the blockchain. In order to make the blockchain more powerful, quantum protection and quantum certificates will be embedded in the transaction syntax of QLB.

A new transaction is created by a node that wishes to transmit cryptocurrency to another node. A transaction T_x that says, "i sends qulogicoins obtained from another transaction to j" has the following form:

$$\mathrm{T_x} = (x, y_1, \cdots, y_n, j; \alpha, \phi; \beta_1, \psi_1 \cdots, \beta_n, \psi_n) . \quad (7)$$

Here x is the hash value of this transaction. Hash function can map a string of arbitrary length to a fixed length string. As in the QB, the Toeplitz hashing [12] is used, and the Toeplitz matrix is generated by key distribution through the quantum channel. $x = T_S(y_1, \ldots, y_n, j, \alpha, \phi; \beta_1, \ldots, \beta_n) \oplus r$, Here S and r are keys and T_S is the Toeplitz matrix generated by S. Here α and ϕ are Boolean functions for classical and quantum certificates, respectively. Here β is some classical data, and ψ is quantum data, which are regarded as certificates.

In a transaction, the quantum certificate is sent through the quantum channel, and other information is sent through the classic channel. Each miner checks whether the new transaction is consistent with the copy of his local ledger and forms an opinion on the admissibility of the transaction. Here consistency checking for $\mathrm{T_x} = (x, y_1, \cdots, y_n, j; \alpha, \phi; \beta_1, \psi_1 \cdots, \beta_n, \psi_n)$ means to check the following:

1. Message authentication: check if $x = T_S(y_1, \ldots, y_n, j, \alpha, \phi; \beta_1, \ldots, \beta_n) \oplus r$, where S and r is taken from the secret keys shared between the miner and the sender.
2. Check if the sender is the receiver of $T_{y_1}, \ldots, T_{y_n}$.
3. Check if T_{y_i} has been redeemed before this transaction, for all $i \in \{1, \ldots, n\}$.
4. Check if β_i satisfies α_{y_i}, where α_{y_i} is the classical protection of T_{y_i}.
5. Check if ψ_i satisfies ϕ_{y_i}, where ϕ_{y_i} is the quantum protection of T_{y_i}.

All miners then apply the honest-success quantum Byzantine agreement protocol (we will introduce in the next section) to the new transaction, arriving at a consensus regarding the correct version of the transaction and the admissibility of the transaction.

The QLB researchers also considered applying QLB to quantum bit commitment protocols. Bit commitment protocols are widely used in cryptographic protocols. They found that the mandatory punishment in the cheat-sensitive quantum bit commitment protocol (CSQBC) was completely ignored in the literature, so QLB was mainly used to execute the punishment. The basic idea is that in CSQBC, people who are detected to be cheating will not be able to obtain qulogicoin.

3.2.2 Quantum honest-success byzantine agreement

Definition (honest-success Byzantine agreement protocol (HBA)) A protocol among n agents in which a distinct agent (sender) holds an input value $x_s \in D$, and all other agents (the receivers) eventually decide on an output value in D is said to achieve honest-success Byzantine agreement if the protocol can guarantee the following:

1. If the sender is honest, then all honest agents decide on the same output value $y = x_s$.

2. If the sender is dishonest, then, either all honest agents terminate the agreement, or all honest agents decide on the same output value $y \in D$.

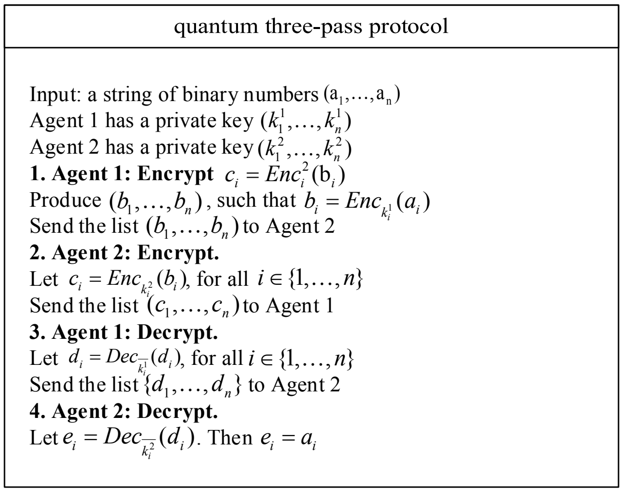
A HBA protocol is p-resilient ($0 < p < 1$) if the protocol still works when less than a fraction of p receivers is dishonest. The quantum honest-success Byzantine agreement (QHBA) protocol described here is $\frac{m-2}{m}$ - resilient, where m is the number of receivers.

The QHBA protocol has the following three phases:

Phase 1 (List distribution by quantum secure direct communication): Quantum secure direct communication (QSDC) [13] based on a quantum version of Shamir's three-pass protocol [14] is used to distribute those correlated lists. The quantum three-pass protocol is introduced as follows.

First, 0 and 1 are encoded as $|0\rangle$ and $|1\rangle$ respectively. The key space for encryption and decryption contains four X-gates $\{X(0), X(\frac{\pi}{2}), X(\pi), X(\frac{3\pi}{2})\}$, where $X(m) = \ |+\rangle\langle+| \ + e^{mi} \ |-\rangle\langle-|$ .The encryption of a qubit $|i\rangle$ with k can be defined as $Enc_k(i) = k\,|i\rangle$, and $Dec_k(i) = k\,|i\rangle$ is the decryption of a qubit $|i\rangle$, where $k \in \{X(0), X(\frac{\pi}{2}), X(\pi), X(\frac{3\pi}{2})\}$. Let $(X(m), X(m) = X(2\pi - m))$ be a pair of encryption/decryption keys.

The detailed steps of quantum three-pass protocol for a sender (agent 1) to send a sequence of bits to a receiver (agent 2) are shown in Fig. 2.



```
quantum three-pass protocol

Input: a string of binary numbers $(a_1,\ldots,a_n)$
Agent 1 has a private key $(k_1^1,\ldots,k_n^1)$
Agent 2 has a private key $(k_1^2,\ldots,k_n^2)$
**1. Agent 1: Encrypt** $c_i = Enc_i^2(b_i)$
Produce $(b_1,\ldots,b_n)$, such that $b_i = Enc_{k_i^1}(a_i)$
Send the list $(b_1,\ldots,b_n)$ to Agent 2
**2. Agent 2: Encrypt.**
Let $c_i = Enc_{k_i^2}(b_i)$, for all $i \in \{1,\ldots,n\}$
Send the list $(c_1,\ldots,c_n)$ to Agent 1
**3. Agent 1: Decrypt.**
Let $d_i = Dec_{\overline{k_i^1}}(d_i)$, for all $i \in \{1,\ldots,n\}$
Send the list $\{d_1,\ldots,d_n\}$ to Agent 2
**4. Agent 2: Decrypt.**
Let $e_i = Dec_{\overline{k_i^2}}(d_i)$. Then $e_i = a_i$
```

Figure 2: A quantum three-pass protocol for secure direct communication

Now the correlated lists can be distributed by using the quantum three-pass protocol. Let $\{P_1,\ldots,P_n,P_{n+1},\ldots,P_{n+d}\}$, Let P_1 be the sender of the QHBA protocol, $P_2,\ldots,P_n$ be receivers and $P_{n+1},\ldots,P_{n+d}$ be list distributors who are in charge of distributing lists of correlated numbers. For every $i \in \{n+1,\ldots,n+d\}$, P_i uses the quantum three-pass protocol to send a list of numbers L_k^i to agent $P_k \in \{P_1,\ldots,P_n\}$. In addition, the following specifications are satisfied:

1. For all $k \in \{1,\ldots,n\}$, $|L_k^i| = m$ for some integer m which is a multiple of 6 .
2. $L_k^i \,| \in \{0,1,2\}^m$, where $\frac{m}{3}$ numbers are 0 , $\frac{m}{3}$ numbers are 1 and $\frac{m}{3}$ numbers are 2 .
3. For all $k \in \{2,\ldots,n\}$, $L_k^i \in \{0,1\}^m$.
4. For all $j \in \{1,\ldots,m\}$, if $L_1^i[j]=0$, then $L_2^i[j]=\ldots=L_n^i[j]=0$.
5. For all $j \in \{1,\ldots,m\}$, if $L_1^i[j]=1$, then $L_2^i[j]=\ldots=L_n^i[j]=1$.

After those lists are distributed, $P_2,\ldots,P_n$ can communicate with P_1 to check whether those lists satisfy the above specification. If for $\theta \in [0,\frac{1}{2}]$, more than θn agents report that the lists distributed by P_i do not satisfy the specification, P_i is classified as a dishonest distributor. Similar to Bitcoin, those honest distributors will receive qulogicoins as rewards, while those who are dishonest will not.

Phase 2 (List formation by sequential composition): After Phase 1, $\{P_1,\ldots,P_n\}$ use a simply sequential composition procedure to form a unique list to be used in the next phase.

Assume there are h list distributors who are classified to be honest. Without loss of generality, we can use $\varepsilon^{n+1}=(L_1^{n+1},\ldots,L_n^{n+1}),\ldots,\varepsilon^{n+h}=(L_1^{n+h},\ldots,L_n^{n+h})$ to represent the lists distributed by those distributors $P_{n+1},\ldots,P_{n+h}$ .The main thing to do at this stage is to form a new sequence of lists $\varepsilon=(L_1,\ldots,L_2)$ which is to be used in the next phase. The ε is constructed by the sequential composition of $\varepsilon^{n+1},\ldots,\varepsilon^{n+h}$ .That is, let $L_1=L_1^{n+1},\ldots,L_1^{n+h},\ldots,L_n=L_n^{n+1},\ldots,L_n^{n+h}$. This means every honest distributor contributes $\frac{1}{n}$ to the final lists ε .

Phase 3 (Achieving consensus): In the previous phase, the correlated lists ε has been formed. Assuming that at least a half of the agents are honest. Then the agents $P_1,\ldots,P_n$ run the following steps to achieve consensus.

1. P_1 sends a binary number $b_{1,k}$ and a list of numbers $ID_{1,k}$ to all P_k, where $k \in \{2,\dots,\mathrm{n}\}$. The list $ID_{1,k}$ indicates all positions on L_1 where $b_{1,k}$ appears and its length is required to be $\frac{m}{3}$, where m is the length of L_1. If P_1 is honest, he will send the same message to all agents. That is, $(b_{1,k}, ID_{1,k}) = (b_{1,\mathrm{j}}, ID_{1,\mathrm{j}})$ for all $j,k \in \{2,\dots,n\}$. If P_1 is dishonest, then he will send different binary numbers and different lists of numbers to different agents, i.e., $(b_{1,k}, ID_{1,k}) \neq (b_{1,\mathrm{j}}, ID_{1,\mathrm{j}})$ for some j,k. If P_1 is honest, he will also use $b_{1,k}$ as the final value it outputs. If P_1 is dishonest, he will use $b_{1,k}$ or $1-b_{1,k}$ randomly as its final output value.

2. Agent P_k starts to analyze the obtained message $(b_{1,k}, ID_{1,k})$ with his own list L_k. If the analysis of P_k shows that $(b_{1,k}, ID_{1,k})$ is consistent with P_k and P_k is honest, he sends $(b_{1,k}, ID_{1,k})$ to all other agents $P_{j\neq 1}$. Here $(b_{1,k}, ID_{1,k})$ is consistent with P_k means that for all index $x \in ID_{1,k}$, $L_k[x] = b_{1,k}$. In addition, P_k will ascertain that P_1 is dishonest and sends $\perp$ to other agents if $(b_{k,j}, ID_{k,j})$ is not consistent with L_k, meaning that "I have received inconsistent message". A dishonest P_k sends $1-b_{1,k}$ with whatever indexes he chooses or simply $\perp$. The full information, which P_j receives from P_k will be denoted by $(b_{k,j}, ID_{k,j})$.

3. Every honest agent P_k considers the received data and acts according to the following after criterion after messages have been exchanged between $\{P_2,\dots,P_n\}$:

(a) If there is a set of agents H with $|H| \geq 2$ such that

i. for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k, and

ii. for some $i,j \in H$, $b_i^k \neq b_j^k$

Then P_k sets his output value to be $\perp$.

(b) If there is a set of agents H with $|H| \geq 2$ such that for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k and all $b_{j,k}$ are the same, and for all $i \notin H$, $(b_{i,k}, ID_{i,k})$ is not consistent with L_k, then H is the set of all honest agents and P_k sets his output value $v_k = b_{j,k}$.

(c) If there is a set of agents H with $|H| \geq 2$ such that for all $j \in H$, $(b_{j,k}, ID_{j,k})$ is consistent with L_k and all $b_{j,k}$ are the same, and for all $i \notin H$, the message sent by P_i is $\perp$, then P_k sets $v_k = b_{j,k}$.

(d) In all other cases, P_k sets his value to be $\perp$.

4. If at least $\frac{n}{2}$ agents output the same bit value $v \in \{0,1\}$, we can say the consensus is achieved. In this case, those agents whose output is the same as v are rewarded with some qulogicoins.

3.3 Quantum algorithm for a certain part of blockchain

Many other researchers have considered to applying the quantum algorithm to a certain part of the general blockchain. Here we mainly introduce the idea of using the analog Hamiltonian optimizers as a basis for the POW protocol and the method of applying Grover algorithm to the general blockchain for the process of mining.

3.3.1 Proof-of-work based on analog Hamiltonian optimizers

To shorten the transaction time and possibly increase the decentralization of the existing blockchain, Kalinin et al. [Kalinin and Berloff (2018)] proposed to use the analog Hamiltonian optimizers as a basis for the POW protocol [Kalinin and Berloff (2018)]. The quantum-computing platform used is called “analog Hamiltonian simulator/optimizer” (AHO) to distinguish it from quantum computer/quantum simulators that rely on entanglement and quantum superposition.

Although the development of blockchain technology has been relatively mature and has been applied in some digital currency systems such as bitcoin, it is still difficult for digital currency to replace some existing payment systems such as alipay and other payment systems. This is mainly because it takes too long to solve the POW puzzle when adding new blocks, so the transaction authentication time is too long. If the processing time is shortened, it will inevitably lead to the centralization of computing, which goes against the idea of decentralization at the beginning of the design of blockchain. Therefore, Kalinin et al. [Kalinin and Berloff (2018)] proposed moving the POW in the blockchain to the analog Hamiltonian optimizer, which would reduce transaction time and not result in centralized computing power. The optimal solution for finding a sufficiently large n-vector model is taken as the basic content of the POW protocol. They considered two of these problems: the quadratic unconstrained binary optimization (QUBO) problem and the quadratic continuous optimization (QCO) problem.

Quadratic unconstrained binary optimization (QUBO) problem:

$$\max\ z^H Qz,\ \ \text{subject to}\ z_i\ \in \left\{-1,1\right\}. \tag{8}$$

Quadratic continuous optimization (QCO) problem:

$$\max\ z^H Qz,\ \ \text{subject to}\ |\,z_i\,|\ =1. \tag{9}$$

Here z^H is the conjugate transpose of z. QUBO is the discrete version of QCO. The decision variables of QCO are constrained on the unit circle, which is a continuous domain. Look at the following two models:

The Ising model:

$$\min\ -\sum_{i<j} J_{ij} s_i\, s_j \text{ subject to } s_i \in \{-1,1\}. \tag{10}$$

XY model:

$$\min\ -\sum_{i<j} J_{ij} s_i\, s_j \text{ subject to } s_i\ =\ (\cos\theta_i, \sin\theta_i)\ . \tag{11}$$

The Ising model and XY model can be mapped to QUBO and QCO respectively (via $z_i\ = cos\theta_i\ +\ isin\theta_i$ for the XY model, and $z_i\ \in\ \{-1,1\}$ for the Ising model). Therefore, Simulators that minimize Ising Hamiltonian such as D-Wave, Coherent Ising Machine can solve the QUBO problem very well and Simulators that minimize XY Hamiltonian such as Gain-Dissipative Simulator on polariton condensates, Gain-Dissipative Simulator on photon condensates and Gain-Dissipative Simulator on quantum electro-dynamics are suitable for handling QCO problems. The schematic of the POW protocols that can be based on solving QUBO or QCO problems using the currently available analog Hamiltonian simulators is shown in Fig. 3.

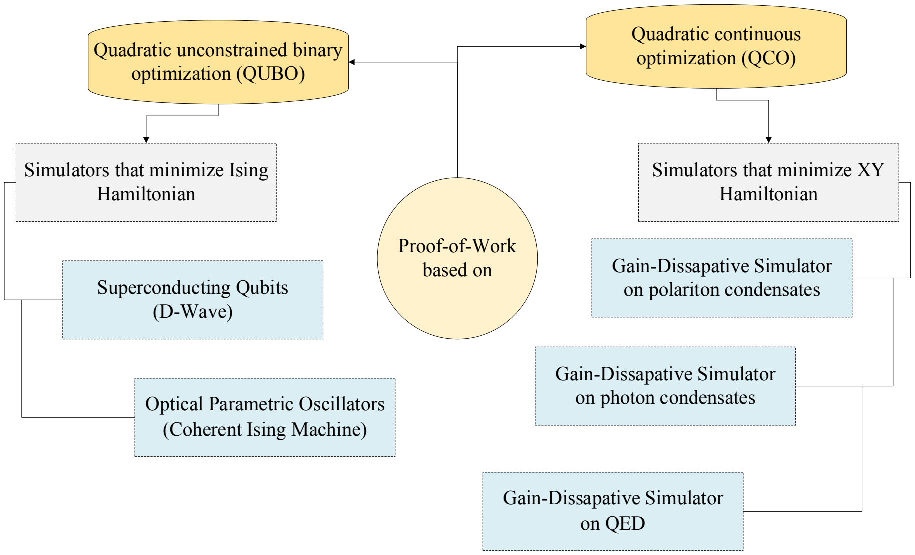

Quadratic unconstrained binary optimization (QUBO)
Quadratic continuous optimization (QCO)
Simulators that minimize Ising Hamiltonian
Simulators that minimize XY Hamiltonian
Proof-of-Work based on
Superconducting Qubits (D-Wave)
Gain-Dissapative Simulator on polariton condensates
Gain-Dissapative Simulator on photon condensates
Optical Parametric Oscillators (Coherent Ising Machine)
Gain-Dissapative Simulator on QED

Figure 3: The schematics of the POW protocols using analog Hamiltonian simulators

Although this research only considered using the analog Hamiltonian optimizers as a basis for the POW protocol, it can shorten the transaction time and potentially increase the decentralization of the existing blockchain.

3.3.2 Modified Grover algorithm for the mining process of blockchain

The Grover algorithm [Grover (1996)] is used to search a target item in an unstructured data set of size N. Compared with the classical search algorithm, Grover algorithm provides a quadratic speedup. The detailed steps of the Grover quantum search algorithm are as follows:

Step 1: Use Hadamard transform to initialize the system to the uniform superposition state$|s\rangle = \frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$.

Step 2: Apply the operator U_w where $U_w|x\rangle = (-1)^{f(x)}|x\rangle$. Here $f(x) \in \{0,1\}$. Moreover, $f(x)=1$ means that the target state is searched while 0 means not. This transformation changes the amplitude in front of the target state to negative, which means that the average amplitude is reduced.

Step 3: Apply the Grover diffusion operator$U_s = 2|x\rangle\langle x| - 1$, this operator makes the amplitude of the state to be searched higher than that of other states.

Step 4: Repeat Steps 2-3 about$\sqrt{N}$ times.

Since it is necessary to find a conditional nonce value in the mining process of the general blockchain, some researchers have suggested that the Grover quantum search algorithm can be applied to the mining process in the blockchain [Sapaev, Bulychkov, Ablayev et al. (2018)]. The mining process in Bitcoin is to substitute different nonce values, and then mix it with the transaction information in the block, the hash value of the previous block, the timestamp and other data, and then run the hash algorithm to get the hash value of the current block until the hash value of the current block meets a certain condition. As mentioned above, the Grover algorithm realizes the quadratic acceleration of unsorted database search, and applying it to the blockchain mining process can greatly improve the mining speed. First, the quantum register is divided into serval parts: nonce, hash, service qubits for implementing basic operations and a functional qubit for the Grover's algorithm. The basic steps of the mining process using the modified Grover Algorithm are as follows:

Step 1: Convert the state of the nonce register to a uniform superposition state by applying a Hardman conversion.

Step 2: The hash values of all nonce values are calculated by quantum parallelism and the result is stored in the hash register.

Step 3: Apply the oracle function to the hash register to find out which nonce values give a hash value below a certain threshold.

Step 4: Apply the reverse hash calculation process to "unwind" all qubits except the nonce qubits and functional qubits to the initial basis state.

Step 5: The Grover diffusion operator is applied to the nonce register.

Step 6: Repeat Steps 2-5 as needed.

The length of the nonce register in bitcoin is 32 bits, and the hash register is 256 bits long. Since the power of a set of nonce values is many times smaller than a set of hash values, it may happen that a hash value that satisfies the condition cannot be found after traversing all the nonce values, so the length of the nonce register is extended to 48 bits. It takes about 40 million seconds for a classic computer to handle the 48-bit nonce, which is 11,000 hours or 465 days. For quantum computers, this only takes 2 seconds. Therefore, the advantage of Quantum-Assisted Blockchain is obvious when compared with the classical Blockchain.

4 Conclusion and disscussion

In this paper, we review the research status of quantum blockchain in recent years. The construction and framework of the quantum blockchain are introduced. Then we introduce the method of applying the quantum algorithm to a certain part of the general blockchain. In addition, we will briefly discuss the performance advantages of quantum blockchain in this part.

Like the classical blockchain, the quantum blockchain also has some features such as decentralization and decentralization. The main characteristics of quantum blockchain are safety and efficiency. The security in quantum blockchain needs to be ensured. The way to guarantee the communication security between nodes is to use quantum secure direct communication (QSDC) [Kiktenko, Pozhar, Anufriev et al. (2018)] or quantum key distribution (QKD) [Gerhardt, Liu, Lamas-Linares et al. (2011); Bennett and Brassard (2014)]. Therefore, the authentication in the network is guaranteed by the properties of quantum physics. In addition, the digital signature can be used to verify that the owner possesses the bitcoin in classical blockchain. However, as mentioned above, the classical encryption algorithms used in digital signature such as RSA may become unsafe in the face of attacks from quantum computers. To solve this problem, the quantum digital signature scheme [Gottesman and Chuang (2001)] can be used in quantum blockchain. Therefore, the quantum blockchain has the characteristics of quantum security. Even the quantum blockchain can be immune to the attacks from quantum computers. The blockchain with quantum technology also has the characteristics of fast transaction processing speed. As mentioned above, the POW based on Analog Hamiltonian Optimizers can shorten the transaction time. This work will greatly promote the development of cryptocurrency. In addition, applying the Grover algorithm to the general blockchain can also improve the efficiency of mining process. However, in fact, anything is a double-edged sword. Before universal quantum computers can be widely available, the parties that have access to them have an unfair advantage in winning the mining awards. In all, compared with classical blockchain, the performance advantage of quantum blockchain mainly lies in its security and efficiency.

Finally, because quantum blockchain has the characteristics of faster processing speed and safer transaction based on quantum mechanics, it will have a very wide range of applications and many research directions in the future.

(1) Application of quantum blockchain. In the field of quantum digital currency, quantum blockchain has a good development prospect. Although many researchers have come up with some ideas for the quantum currency such as Quantum Bitcoin [Jogenfors (2016)] and qBitcoin [Ikeda (2017)], it is still very possible to improve these quantum currency systems with quantum blockchain. For instance, Quantum Bitcoin still uses a classical blockchain, just like the Bitcoin protocol. Therefore, we can consider using quantum blockchain to replace the classical blockchain used in Quantum Bitcoin. In addition, the method of creating a new, safer and more efficient currency system using quantum blockchain also deserves further study. Nevertheless, the use of blockchain is not only limited to quantum digital currency, those technologies based on distributed storage and consensus mechanisms all may benefit from quantum blockchain. Therefore, research can be carried out to apply quantum blockchain in other tasks such as electronic voting,

online auction and multiparty lotteries.

(2) An easy construction scheme of quantum blockchain. Rajan et al.'s [Rajan and Visser (2018)] construction scheme of a quantum blockchain (see Section 3.1) is a bit hard to implement in the sense that the entanglement in time is not easy to realize on a large scale. Thus, a new and more easily implemented construction scheme is now in demand to promote the development of quantum blockchain.

(3) Experimental test of quantum blockchain. Kiktenko et al. [Kiktenko, Pozhar, Anufriev et al. (2018)] have experimentally tested their quantum blockchain protocol by means of a three-party urban fibre network QKD in Moscow [Kiktenko, Pozhar, Anufriev et al. (2018)]. There are four nodes in their quantum blockchain network, and they use an urban fiber QKD network to procure authentication keys for two of the links connecting three nodes. Therefore, it is also a meaningful thing to expand the scale of the experiment. In addition, if other quantum blockchain protocols such as quantum-enhanced logic-based blockchain can also be tested experimentally, the feasibility of these protocols can be compared based on these test results. This work may provide a reference for constructing a real quantum blockchain platform in the future.

Acknowledgement: The authors would like to thank the anonymous reviewers and editor for their comments that improved the quality of this paper. This work is supported by National Natural Science Foundation of China (Grant Nos. 61502101, 61501247, 61672290 and 71461005), Natural Science Foundation of Jiangsu Province (Grant No. BK20171458), the Six Talent Peaks Project of Jiangsu Province (Grant No. 2015-XXRJ-013), the Practice Innovation Training Program Projects for Jiangsu College Students (Grant No. 201810300016Z), and the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

References

Bennett, C. H.; Brassard, G. (2014): Quantum cryptography: public key distribution and coin tossing. *Theoretical Computer Science*, vol. 560, no. 12, pp. 7-11.

Carvacho, G.; Graffitti, F.; D'Ambrosio, V.; Hiesmayr, B. C.; Sciarrino, F. (2017): Experimental investigation on the geometry of GHZ states. *Scientific Reports*, vol. 7, no. 1, pp. 13265.

Chamoli, A.; Bhandari, C. M. (2009**):** Secure direct communication based on ping-pong protocol. *Quantum Information Processing*, vol. 8, no. 4, pp. 347-356.

Gerhardt, I.; Liu, Q.; Lamas-Linares, A.; Skaar, J.; Kurtsiefer, C. et al. (2011): Full-field implementation of a perfect eavesdropper on a quantum cryptography system. *Nature Communications*, vol. 2, pp. 349.

Gottesman, D.; Chuang, I. (2001): Quantum digital signatures. arXiv:quant-ph/0105032v2.

Grover, L. K. (1996): A fast quantum mechanical algorithm for database search. *28th Annual ACM Symposium on the Theory of Computing*, pp. 212-219.mm

Ikeda, K. (2017): qBitcoin: a peer-to-peer quantum cash system. arXiv:1708.04955v2.

Jogenfors, J. (2016): Quantum bitcoin: an anonymous and distributed currency secured by the no-cloning theorem of quantum mechanics. arXiv:1604.01383v1.

Kalinin, K. P.; Berloff, N. G. (2018): Blockchain platform with proof-of-work based on analog hamiltonian optimisers. arXiv:1802.10091v1.

Kiktenko, E. O.; Pozhar, N. O.; Anufriev, M. N.; Trushechkin, A. S.; Yunusov, R. R. et al. (2018**):** Quantum-secured blockchain. *Quantum Science and Technology*, vol. 3, no. 3, 35004.

Krawczyk, H. (1994): Lfsr-based hashing and authentication. *Annual International Cryptology Conference*, pp. 129-139.

McCutcheon, W.; Pappa, A.; Bell, B. A.; McMillan, A.; Chailloux, A. et al. (2016): Experimental verification of multipartite entanglement in quantum networks. *Nature Communications*, vol. 7, pp. 13251.

Megidish, E.; Halevy, A.; Pilnyak, Y.; Slapa, A.; Eisenberg, H. S. (2017): Quantum tomography of inductively-created large multiphoton states. arXiv:1712.03633v1.

Megidish, E.; Shacham, T.; Halevy, A.; Dovrat, L.; Eisenberg, H. S. (2012**):** Resource efficient source of multiphoton polarization entanglement. *Physical Review Letters*, vol. 109, no. 8, 080504.

Nakamoto, S. (2008): Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf.

Nielsen, M. A.; Chuang, I. (2011): *Quantum Computation and Quantum Information*, 10th Anniversary Edition. Cambridge University Press.

Rajan, D.; Visser, M. (2018): Quantum Blockchain using entanglement in time. arXiv:1804.05979.

Sapaev, D.; Bulychkov, D.; Ablayev, F.; Vasiliev, A.; Ziatdinov, M. (2018): Quantum-assisted blockchain. arXiv:1802.06763v2.

Shor, P. W. (1994): Algorithms for quantum computation: discrete logarithms and factoring. *35th Annual Symposium on Foundations of Computer Science*, pp. 124-134.

Sun, X.; Wang, Q.; Kulicki, P.; Zhao, X. (2018): Quantum-enhanced logic-based blochchain I: quantum honest-success byzantine agreement and qulogicoin. arXiv:1805.06768.

Swan, M. (2015): Blockchain: Blueprint for a New Economy. O'Reilly Media.

Zhou, L.; Wang, Q.; Sun, X.; Kulicki, P.; Castiglione, A. (2018): Quantum technique for access control in cloud computing II: encryption and key distribution. *Journal of Network and Computer Applications*, vol. 103, pp. 178-184.