ECBC: A High Performance Educational Certificate Blockchain with Efficient Query

Yuqin Xu[1], Shangli Zhao[1], Lanju Kong[1], Yongqing Zheng[1,2],
Shidong Zhang[1], and Qingzhong Li[1](✉)

[1] School of Computer Science and Technology, Shandong University,
Jinan, China
xuyuqin_sdu@163.com, jnzsl163@126.com,
{klj, zyq, zsd, Lqz}@sdu.edu.cn
[2] Dareway Software Co., Ltd., Jinan, China

Abstract. Currently, most digital infrastructures for educational certificate management cannot guarantee data security and system trust. Using blockchain can solve this problem. However, there are still some defects with the existing blockchains that cannot be applied. Most of them are dependent on tokens, and limited by throughput and latency, moreover, no one can support certificate query with precise and high efficiency. In order to solve these problems, this paper presents educational certificate blockchain (ECBC) which can support low latency and high throughput, and provide a method to speed up queries. To reduce latency and increase throughput, consensus mechanism of ECBC uses the cooperation of peers to create blocks in place of the competition. ECBC builds a tree structure (MPT-Chain) which can not only provide an efficient query for a transaction, but also support historical transactions query of an account. MPT-Chain only needs short time to update and can speed up block verification. In addition, ECBC is designed with transaction format to protect user's privacy. The experiment shows that ECBC has better performance of throughput and latency, supporting quick query.

Keywords: Consensus mechanism · Blockchain scalability · Quick query

1 Introduction

The certificate is a manifestation of student's learning ability, it helps students to find a satisfactory job, of course, it can prevent us from getting a job if we provide a forged certificate. However, the validation process for certificates is lengthy and complex, which makes it possible to forge [1]. Therefore, it is imperative to establish a reliable digital infrastructure for certificates. In China, the website XueXinWang [2], as a certificate digital infrastructure, provides a lot of convenience. For example, it verifies the authenticity of the certificate quickly. But it also has many shortcomings, using central storage to save data cannot guarantee data security, under attack; the data may be lost, altered or leaked. In addition, the centralized system cannot guarantee system trust.

D.V. Hung and D. Kapur (Eds.): ICTAC 2017, LNCS 10580, pp. 288–304, 2017.
DOI: 10.1007/978-3-319-67729-3_17

Blockchain can guarantee data security and solve the problem of system trust [3] which is first raised in bitcoin [3, 4]. So using blockchain to achieve certificate system for the management of certificates will be reliable, safe and trustworthy. However, the existing blockchain for the management of certificates still show many shortcomings.

First of all, the existing blockchains are mostly dependent on tokens, but the certificates management does not require tokens. Secondly, consensus mechanisms (e.g. POW) waste a lot of computing resources, in addition, its throughput and latency cannot meet the requirements of certificate system [5, 6]. Thirdly, storing data transparently will lead to the disclosure of personal privacy [7]. Finally, only the ethereum [8] provides an index structure called MPT can achieve a quick query for the latest status of account, but cannot support the efficient query for history transactions of an account [9]. Querying the history records of a certificate holder is very important, because people always want to be able to query a person's education experience.

Based on the discussion above, this paper proposes an educational certificate blockchain (ECBC) to manage educational certificates. In order to improve performance, consensus mechanism does not need peer to compete to calculate the block's link value. The link value of the block needs to be generated by the cooperation of peers, and no one can know the link value of the block in advance. ECBC has the following advantages that it avoids waste of computing resources, has no fork, does not depend on tokens, and can meet the requirements of throughput and latency.

ECBC treats the issuance or revocation of a certificate as a transaction, which will be written into the blockchain, and designs transaction format to prevent privacy leaks. The privacy data of users is encrypted, which ensures that even if someone maliciously obtains the blockchain data, it is not possible to obtain users' information. Using Patricia tree [10] can quickly locate query results. Merkle tree [11, 12] is used to ensure that data accepted from others is not corrupt and not replaced, and even can check that others do not spoof or publish false data. Based on these, ECBC combines the features of Patricia tree and merkle tree, constructing a tree structure (MPT-Chain) to speed up query and ensure the correctness of query results in a distributed network.

In order to support querying history records efficiently, MPT-Chain extends the leaf nodes so that the leaf nodes can store the logical relationship of the account transaction chain. In addition, the node of MPT-Chain stores intermediate value for merkle root calculation, which can be used to speed up the MPT-Chain update and block verification.

The main contributions of this paper are as follows:

1. A consensus mechanism is proposed for blockchain, without bifurcation, and achieves high throughput and low latency.
2. This paper proposes a tree structure (MPT-CHAIN), which takes little time to update, supports query account transaction chain and speeds up block verification.

The remainder of the paper is organized as follows: Sect. 2 introduces related work which had done lots of work for certificate management and the performance of blockchain. In Sect. 3, educational certificate blockchain architecture is introduced. Section 4 describes consensus mechanism for creating blocks in detail. Section 5 introduces the MPT-Chain of ECBC. And Sect. 6 shows efficiency analysis and experiments.

2 Related Works

Prior to us, the Massachusetts Institute of Technology Media Laboratory (MIT Media Lab) has been noted problems of the existing digital infrastructures for educational certificate. They tried to use blockchain to solve the problem of system trust and data security, and designed a set of tools to display and validate educational certificates [1]. The overall design of the certificate architecture is simple. When the issuer signs an educational certificate, its hash value would be stored in the bitcoin's blockchain.

It is undeniable that the MIT Media Lab's solution can solve the problem of data security and system trust, but the educational certificate stored in the bitcoin's blockchain, which makes certificates administration more complex and relies on tokens. Bitcoin development has been severely constrained by its throughput and latency, the low performance is fatal for certificates management [6, 13].

In Bitcoin, the way that a transaction actually works "under the hood" is that it consumes a collection of objects called unspent transaction outputs ("UTXOs") created by one or more previous transactions, and then produces one or more new UTXOs, which can then be consumed by future transactions. A user's balance is thus not stored as a number; rather, it can be computed as the total sum of the denominations of UTXOs that they own [14].

UTXOs are stateless, and so are not well-suited to applications more complex than asset issuance and transfer that are generally stateful, such as various kinds of smart contracts. It would be very difficult to understand the logical relationship between transactions, because the relationship between the user and transactions is confused in UTXO model if we treat a transaction as the issuance or cancellation of a certificate.

The use of certificates is frequent in the certificate management process. But there is no strategy can support efficient query in bitcoin. Traversing data is not feasible, which will be more and more slowly while data increasing. The recent rise of the Ethereum to do some of this work, but for the management of certificates is not enough. They put forward an account model which can manage transactions better than UTXO model. Besides, Ethereum sets up three index trees (MPT) to speed up query, one for getting transaction, and one for getting account's balance, the remaining one is for receipt. All of them are for latest state of account, cannot support to find transaction chain of an account. But querying the history records of a certificate holder is very important.

3 ECBC Architecture

In this section, Educational Certificate Blockchain Architecture is introduced. ECBC uses blockchain to organize schools, regulators, students, and employers. It facilitates the review of the certificate data by the regulatory authority and also protects the security of certificate data and improves system trust.

The p2p-based educational certificate network consists of peers and entities:

Definition 1. Peer. Peer represents schools and regulators. It is versatile can be involved in creating blocks in the network. Each peer has an identity authentication, can use public key for encrypting messages and private key for signing blocks. It can ensure the security of message and blocks cannot be forged.

Definition 2. Quorum. Quorum refers to a peer who has the right to participate in the consensus process. The consensus mechanism of ECBC will dynamically will dynamically select some peers to become quorums which can generate blocks through cooperative consensus algorithms.

Definition 3. Entity. Entity represents students, and employers. ECBC provides light client for entities to verify and query the certificate, some entities are the holder of educational certificates, can get public and private key to protect user privacy. Entity can submit query request, and verify the correctness of the query results through block header.

Peers use consensus mechanism presented in this paper to generate a block, its basic structure is shown in Fig. 1. The block structure consists of a block header and a number of transactions. The block header includes link value of previous block, the link value of this block, the creator, the block height, the timestamp and the merkle root of MPT-Chain. The link value of the block is created in the peers' consensus process by all the peers' cooperation. Each block (except the genesis block) contains the link value of the previous block, thus forming a blockchain. In addition, the merkle root of MPT-Chain can guarantee the correctness of query results and the consistency of MPT-Chain in the network.

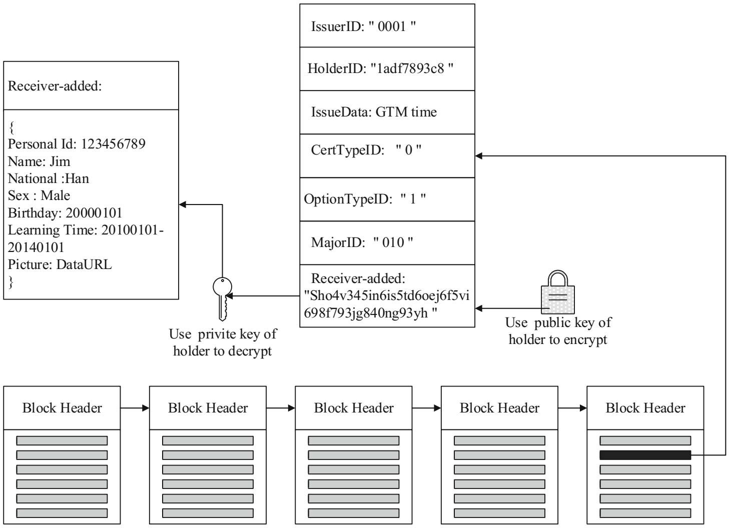

Fig. 1. ECBC architecture

The issuance or revocation of a certificate as a transaction will be written into blockchain, which is initiated by an issuer (such as school). Transaction format is designed in accordance with open badges specification [15], which includes IssuerID that can represent the issuer, and holderID can anonymously identify the certificate holder, the major, the type of certificate, the type of operation, the encrypted information concerning the privacy data, and the timestamp. The Operation Type is used to indicate whether the certificate was issued or revoked, because the data for the blockchain can only append. Blockchain as a distributed ledger, its data is open and transparent, which requires us to protect personal privacy [16]. Asymmetric encryption algorithm can guarantee data privacy, but also can cause the transaction cannot be verified. Therefore, we will use the asymmetric encryption algorithm encrypts the part of the transaction data which is related to user privacy, but does not affect the transaction verification, such as personal ID, name, date of birth, learning time, personal photos and so on.

4 Consensus Mechanism of ECBC

In this section, we will describe consensus mechanism of ECBC in detail. ECBC is a permission chain; the peer who wants to into the network must have permission. Each peer represents a trustworthy organizations or units, of course, we cannot rule out these peers which may become Byzantium peer. In bitcoin, the peers can join unlimited, which may lead to high possibility of byzantine peers [5, 6]. Therefore, the fault tolerance of POW is 51% of all the peers. The joining of peers must be permitted and peers' credibility is high, this characteristic for designing a consensus mechanism has brought different view. We reduce the fault tolerance of the network to one-third, and design a consensus mechanism which is scalable, high efficient, while ensures security and credibility. Creating block is a process of rotary, so we only describe a round of creation.

4.1 Dynamic Quorums

In order to ensure block creation secure and reliable, each block wants to take effect should reach a consensus by all the peers in the network. However, the number of peers in the network is increasing. All the peers are involved in consensus process, which is a waste of computing resources. So we select some peers called quorums to reach a consensus, but fixed quorums may cause security problems. For avoiding the occurrence of this phenomenon, we dynamically pick out quorums. Use computing power (P), and the number of times who had been as quorum (T) to calculate peer's comprehensive value (C_V). Calculation formula of C_V is as follows:

$$\mathrm{C\_V} = \mathrm{P}/\sqrt{\mathrm{T}} \tag{1}$$

To ensure that quorum is changed dynamically, the number of times who had been as quorum is used as a limiting factor. Suppose the number of peers in the network is N(N = 3f + 1), where f represents the number of Byzantine nodes that may exist in the network. Sort the peers from big to small according to C_V, select the first 2f + 1 of this sequence to be quorum. The time complexity of this algorithm is $\mathrm{O}(N * \log(N))$, time overhead is negligible for ECBC.

Quorums selected will reach a consensus on a block. A round of consensus process may create a block or may not. After a round of consensus process, the number of consensus round increases affirmatively and the block height may not. Each consensus round will correspond to a set of quorums that called view, when a round of consensus process ends, due to changes in T, resulting in view change.

4.2 Cooperation Consensus Algorithm

Cooperation consensus algorithm does not depend on computing power of the peer, but requires quorums to work together. It can be divided into three steps: The first step is quorums reach a consensus for the block's link value; the second step is to select primary peer which can create block in this round; the third step is quorums vote for block created by the primary peer.

In first step, each quorum generates a random number, and uses its private key to sign the random number, then sends it to others. When a quorum receives all the random numbers from other, using random numbers, the merkle root of transactions in the block, the link value of previous block and time stamp calculate hash value as the link value of this block. Link value is used to guarantee that block is not easily falsified. If anyone wants to modify any transaction value in the block, all previous blocks' link value need to be changed accordingly.

If the random number sent to others by the quorum is inconsistent, the link value will be different for different quorum. This phenomenon will lead to multiple primary peers in a consensus process, will also be multiple blocks. So this consensus process is no doubt a failure. Byzantine peers pay a small price can lead to a significant increase in the failure rate of consensus. To prevent such attacks, it is necessary to check the random number when the quorum receives a random number. The quorum packaged all random numbers into a set, and then broadcast it to others. All the quorum check random numbers by the set, if the one is not the same, the random number generator will be removed from quorums.

And then, select a quorum as primary peer to create block through random numbers. The primary peer is selected by the average of all the random numbers, whose random number is closest to the average and it broadcasted random number in the earliest time will be selected as primary peer. The primary peer can construct block and broadcast it to all the peers in the network. Constructing a complete block consists of packing the transaction into the block and calculating the MPT-Chain's merkle root in the block header. The calculation of merkle root will be highlighted in the next section.

When a quorum receives the block by the primary peer, and then verifies the correctness of the block, including the index root, transactions, block height, merkle root of MPT-Chain and then vote for the block. If a block can get votes more than half of the total number of quorum, which is f + 1. Then this block can be written in the blockchain, the height of blockchain and consensus round increases. Otherwise, consensus round increases.

Algorithm 1 Cooperation Consensus Algorithm

```
Input:    quorums
Output:   block
1: Each quorum generates a random number
2: Send random number with signature to others
3: If (the number of random numbers had been received =
the number of quorums)
4:Check random numbers
5:Calculate block’s link value
6:   Go to line 13
7: End if
8: Else if ( time out && don’t receive the random number
)
9:    The quorum removed
10:  the number of quorum = the number of quorum -1
11:   Go to line 3
12: End if
13: Select primary peer to create block and then broadcast
14: If( time out && no block by primary peer)
15:  Go to line 26
16:End if
17:Else if (verify block == true)
18:   Vote
19:End if
20: If (the number of vote = f +1)
21:   write block into blockchain
22:   block height = block height +1
23:   round = round +1
24:End if
25: Else
26:   round = round +1
27:End else
```

Considering that information may be lost in the process of transmission or quorum failure (such as earthquake and other natural disasters), which will cause messages cannot be transferred. We set time threshold to prevent such situations. If the waiting time is more than time threshold and others still does not receive the random number, block or vote by the quorum, we can come to the conclusion that the quorum cannot communicate. In order to prevent the infinite wait for random number in the consensus process, the quorum cannot communicate should be removing from quorums. When the block and enough votes cannot receive until waiting time is more than time threshold, the consensus process will be failure.

4.3 Basic Properties of Cooperation Consensus

Given that cooperation consensus is a new blockchain creation rule, it is imperative to first show that all peers eventually adopt or accept a certain block uniformly. For any block B, we define C_B is the creator of block B. The definition ψ_B is the time of B when it was first accepted or abandoned. In addition, $H(\psi_B)$ is defined as the height of blockchain when the time is ψ_B.

Proposition 1.1. (The Convergence of History). $P_{\mathrm{r}}(\psi_B < \infty) = 1$. In other words, every block is eventually either fully abandoned or fully adopted.

To prove the proposition, we make use of the following claim.

Claim 1.2. $\forall(\psi_B) < \infty$. For any block B, ψ_B is always less than infinity.

Proof. It can be seen from the algorithm of Cooperative consensus that the growth speed of educational certificate block chain is mainly influences by network delay. Let D be the delay diameter of the network. Assume that block B is either adopted or abandoned by all peers when time is Ψ_{B}. That is to say, at time Ψ_{B}, block B has votes which are not more than half of the quorums and a round of cooperation consensus has not ended. All peers in consensus network will continue to wait until the block receives votes more than half of the quorums, or the waiting time exceeds the time threshold. The time threshold is set to eliminate that a round of consensus cannot reach the termination of the state caused by the network delay.

Proof of Proposition 1.1. If $\psi_B = \infty$ then there are no new blocks added to blockchain. The probability of this case must be zero. As by Claim 1.2 we know that $\forall(\psi_B)$ is finite.

Proposition 1.3. (Resilience from 50% attacks). The 50% attack here is for the consensus network. If C_B is a byzantine peer and block B contains illegal transactions, When this round of consensus for block B is completed, $H(\psi_B)$ is not incremented. That is, all peers will abandon block B, the 50% attack initiated by the byzantine peer is a failure.

Proof. In consensus network, the upper bound of the number of byzantine peers is f, and the total number of peers is 2f + 1. If a byzantine peer becomes the primary peer construct block B and join all the byzantine peers in the network together to cheat. The maximum number of votes that block B can get is f, but the block that wants to join into blockchain must obtain votes at least f + 1. However, for a block that cannot pass validation, it is not possible to get votes that exceed f + 1. Therefore, we can conclude that in consensus network, 50% attacks which initiated by the byzantine peer always fail.

5 MPT-Chain of ECBC

ECBC not only solves the problem of data security and system trust, but also provides users with efficient query services. With the block height increasing, the number of transactions is also growing rapidly. The speed of linear query cannot meet user's requirement. This paper proposes a tree structure called MPT-Chain to speed up query, which combines the features of Patricia tree and merkle tree. MPT-Chain can ensure the consistency of distributed index and the correctness of query results.

Using Patricia tree as index can quickly locate the user's query results. And the update of Patricia tree does not need to spend too much computing resources, just need to modify the leaf node. Still, if primary peer broadcast index along with the block, huge data transmission may block the network. If the peers using transactions build index locally, this may cause the byzantine peer to return the wrong query result. Traversing data set is an only way to verifying the result.

Merkle tree is used to ensure that the data accepted from others is not corrupt and replaced, and can check that others do not spoof or publish false data. The merkle proof it provides is the basis of SPV (simple pay verification) [18], which can support the validation of the data in the light client. Based on these, this paper combines merkle tree and Patricia tree, constructs MPT-Chain which makes ECBC can guarantee the consistency of indexes in the distributed network and the correctness of query results.

5.1 Node Structure of MPT-Chain

MPT-Chain is a tree structure which contains four different types of nodes. The node structure is shown in Fig. 2. The structure of the root node consists of two parts: branch pointer and value. Branch pointer stores the pointer point to the branch node. Value field stores merkle root of the MPT-Chain, which stored in the block header will be changed when creates a block. The branch node is a list, its length is 17. HolderID in hexadecimal encoding format is used as search key, so all the branch node has 16 keys for storing the child pointer. The key in the branch node lists all possibilities of character, which reduces the trouble of dynamic updates. When search path reaches this branch node, the index number of the key represents the value of the search code. The value field of branch node stores merkle root of the merkle tree when taking the branch node as the root. Starting from the leaf node, calculated layer-by-layer until it reaches the branch node, stored merkle hash calculated in the value field.

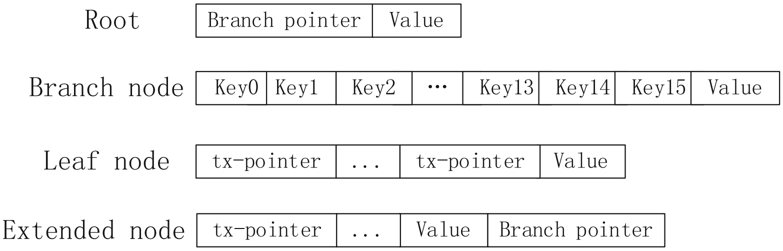

Fig. 2. Node structure of MPT-Chain

The leaf node is a list that can be dynamically changed in length. It consists of two parts: tx-pointer and value. The tx-pointer stores the pointer point to the transaction that can dynamically append when the holderID related transactions are increasing. The leaf node will add a new tx-pointer when a new transaction of the holderID is written into blockchain. The value field stored in leaf node is a hash that can be calculated by constructing the merkle tree of the transactions, which is pointed by tx-pointer in the leaf node. An extended node is an extension of a leaf node. It is used to solve the problems when a holderID encoding is a prefix for others, extend the leaf node of the holderID to an extended node in order to extend the encoding. When a leaf node extends to an extended node, the leaf node needs to be added a field to store the branch pointer. The value field stored in leaf node is a hash that can be calculated by the hash of the transaction pointed by tx-pointer and the value of the branch node pointed by branch pointer.

The value field in the branch node, the leaf node and the extended node is intermediate value of the merkle root. When primary node creates a new block, it needs to update the MPT-Chain; the update process will require the recalculation of the value field of the MPT-Chain's nodes. So the node of MPT-Chain stores intermediate value of the merkle root, which will be reused when the branch is not updated and can prevent the waste of computing resources. In addition, it can also shorten the block creation time.

5.2 The Example of MPT-Chain Structure

Users can query the certificates by holderID, and check the personal information by decrypting the Receiver-added information to confirm whether the owner of the certificate is consistent with the person who provides holderID. It is inspired by a zero-knowledge proof [17]. The requirement of users for the query is different, such as some users may only need to query a certificate, and some users may query all the holder's certificates.

The MPT implemented by the Ethereum is only the latest state query. It cannot support the history transactions of account. And with the application of blockchain in many fields, the index structure that can only retrieve the latest state will not meet the requirement of query and verification. Therefore, this paper proposes a MPT-Chain based on node structure which is extended from MPT. The leaf node of MPT-Chain contains multiple tx-points. These pointers will point to all transactions related to holderID, which constitute the holderID's transaction chain. Based on above, the tree structure proposed by this paper calls MPT-Chain.

Table 1. Example of transactions

Transaction	0	1	2	3	4
IssuerID	0001	0231	0032	0671	0001
HolderID	0517	89ca7f	05173	4a22f	0517
IssueData	20130601	20160601	20160601	20160601	20160607
CertTypeID	1	2	1	1	2
OptionTypeID	0	0	0	0	0
MajorID	001	022	402	013	001
Receiver-added	Ciphertext	Ciphertext	Ciphertext	Ciphertext	Ciphertext

In ECBC, holderID is an anonymous id that does not map the real world, and uniquely identifies a certificate holder. This paper builds the MPT-Chain to speed up query by using holderID as the search key. For ease of understanding, we have listed five educational certificate transactions that are shown in Table 1 and given the structure of MPT-Chain in Fig. 3. MPT-Chain stores the pointer of transaction in the tx-pointer of the leaf nodes. When using holderID to query, MT-Chain reads the holderID bit by bit and matches holderID starting from the root to the leaf node or the extended node.

Figure 3 use gray to represent the search path by using holderID in Table 1, and shows a tx-pointer that points to a transaction in the blockchain. It is assumed that the transactions in Table 1 exist in the blockchain in a graphical way. According to the data in Table 1, transaction 0 is a pre-transaction for transaction 4, the logical relationship can be learned by the structure of the leaf nodes or extended nodes, which is the difference between the MPT of ethereum. In addition, the latest Merkle root of MPT-Chain is also stored in the block header of the latest block; this relationship is expressed in Fig. 3 using dashed lines.

The update of MPT-Chain and the calculation of its Merkle root will affect the verification time of the block, thus affecting the transaction throughput. So, this paper chooses Patricia tree to speed up query, because as the data increases, the update of Patricia tree takes less time. Moreover, the value field of the branch node, the leaf node, and the extended node in MPT-Chain, which can be re-used in the calculation process, in order to shorten the block validation time.

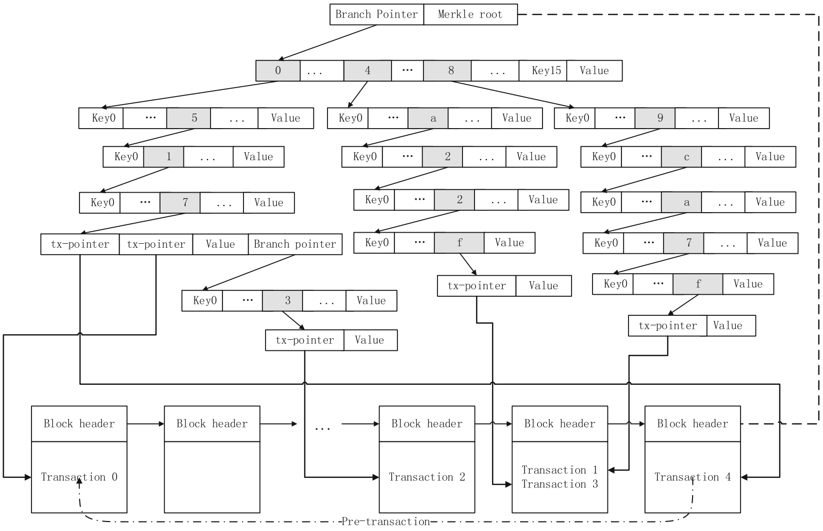

Fig. 3. MPT-Chain structure

When the primary peer starts creating a block, it first determines which transactions are stored in this block, and then updates the MPT-Chain based on these transactions to compute merkle root. After that, the primary peer stores merkle root of MPT-Chain in the block header. The primary peer broadcasts the block created to the network, other peers receive the block and check it. The other peers need to check the correctness of the transactions contained in the block and the data in the block header, including the calculation of merkle root for MPT-Chain. Using the transactions contained in the block to update MPT-Chain stored locally, calculate merkle root, compared it to the merkle root stored in block that is created by primary peer.

6 Efficiency Analysis and Experimental

This section discusses efficiency from the point of throughput, network delay and information transmission, and the theoretical analysis is proved by the experimental results.

The primary measure of ECBC's scalability is the number of transactions per second (TPS). The TPS is the rate of growth of the blockchain, multiplied by the size of blocks, and divided by the average size of a transaction. Thus,

$$\mathrm{TPS}(\lambda, \mathrm{b}) = \lambda \cdot \mathrm{b} \cdot \mathrm{K} \tag{2}$$

ECBC is a chain without fork, so block creation rate is the rate of growth of the blockchain defined as λ, b is the size of blocks and K is the average number of transactions per KB.

6.1 Delay and the Size of Blocks

As we have already seen, the delay in the network is a highly significant factor that impacts the rate of creation block. A measurement study which was recently presented by Decker and Wattenhofer [19] addresses the issue. They have set up a node on the Bitcoin network that connected to as many accessible nodes as possible. Since each such node announces new blocks to its neighbours, it is possible to record these events and estimate the time it takes blocks to propagate.

The experiment of Decker and Wattenhofer depicts this linear effect quite clearly. The interesting point is that the linear dependence on the block size, which is characteristic of a single link, also holds in aggregate for the entire network [19]. This paper adopts a linear model of the delay:

$$\mathrm{D}_{50\,\%}(\mathrm{b}) = \mathrm{D}_{\mathrm{prop}} + \mathrm{D}_{\mathrm{bw}} \cdot \mathrm{b} \tag{3}$$

The time it takes to get to 50% of the network's peers is quite accurately described by the best fit of such a linear relation to the data. Notice that D_{prop} is a measure of aggregate propagation delay, and D_{bw} is an aggregate measure in units of seconds per KB. The fit parameters are: D_{prop} is 1.8 s, and D_{bw} is 0.066 s per KB.

Through the above analysis, we can conclude that the growth rate of the chain is mainly affected by the network delay and the block size. We get $\lambda = \lambda(D, b)$, D is used to represent the network delay. From the experimental results of Decker and

Wattenhofer, we find that, in the ideal case, the network delay is proportional to the size of the transmitted data, that is $D = D(b)$, so we conclude that $\lambda = \lambda(b)$.

In the ECBC, a round of consensus, the three types of information needs to be transmitted between peers, which include random number, block and vote. The size of random number and vote is small, the network delay caused by random number and vote is smaller than block, so we can draw a conclusion that TPS most depends on the block size.

6.2 Estimate of the Achievable TPS

In ECBC, JSON format is used to build transactions. By testing, the value of K is about 16. The next important thing is that we need to do a measure with network delay and block size. In ECBC, after a round of consensus, it does not always create block which can be added to the blockchain. Maybe, this round of the consensus is futile. If a round of consensus time is too long, but the result is futile. There is no gain for increasing the throughput, on the other hand, this phenomenon reduces throughput. Therefore, we make a trade-off between transaction latency and block size. The block size we select is 200 KB, and set the time threshold for the consensus process. Time threshold for the random number and the vote is 2 s, the block's time threshold is 20 s.

In theory, the growth rate of the block is about 1/16 block per second when the network is in good condition, so TPS $\approx$ 200 if K = 16 and b = 200. This value is the theoretical estimate by using Decker and Wattenhofer's experimental data. Moreover, the experiment in this paper verifies the theoretical data by constructing an actual network environment. The peers in ECBC need permission to enter and the number of peers relative to the bitcoin is less than bitcoin which peers from the world and can be arbitrarily joined.

6.3 Experimental Results

At present, relatively mature blockchain technology has been open source which is convenience for our work, and we would like to appreciate these generous researchers and developers. Refer to some of the mature open source code, such as ethereum and hyperledger [20], some of the blockchain common technology which has been implemented is also applicable in ECBC. Such as network communication, signature, encryption and so on. Their contributions help us reduce our workload and speed up validation of theory this paper proposed.

ECBC learns membership management service module from hyperledger and achieves a peer who want to join needs to be allowed by network. Block data and MPT-Chain use levelDB to store, which is an efficient key-value database. This paper tested with 5, 50, 100, and 200 peers when we examined transaction latency and throughput of ECBC. In the case of good network conditions, we confirmed our theoretical analysis through experimental data. Hardware configuration of the peer is the same. The server is E5620 @ 2.40 GHz, 24 GB of memory, CentOS operating system, and Gigabit Ethernet directly between the peers.

The experimental data used in this paper is more than 1.6 million transactions from more than 500 blocks in ECBC. We compare creation time of block and transactions throughput with Bitcoin, which is significantly better than Bitcoin. Figure 4 shows transactions throughput, it can reach three hundred per second. Statistics show that the number of graduates is about 7 million in 2016 China, the throughput of ECBC is able to meet the requirements of certificates management. Figure 5 indicates creation time of block. According to the experimental data, we can see that transaction latency is about 10 s that entities can bear.

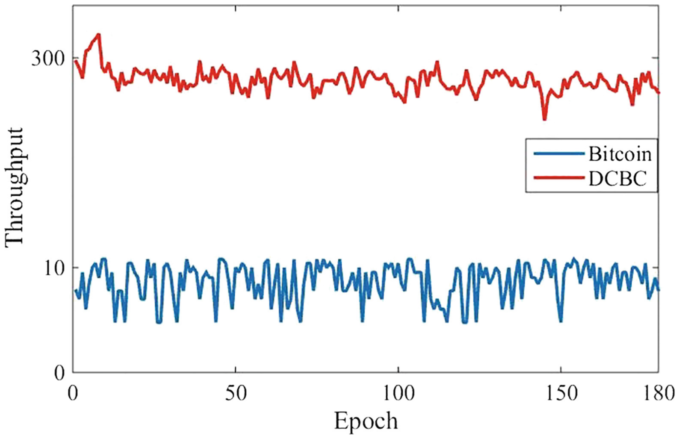

Fig. 4. Transaction throughput

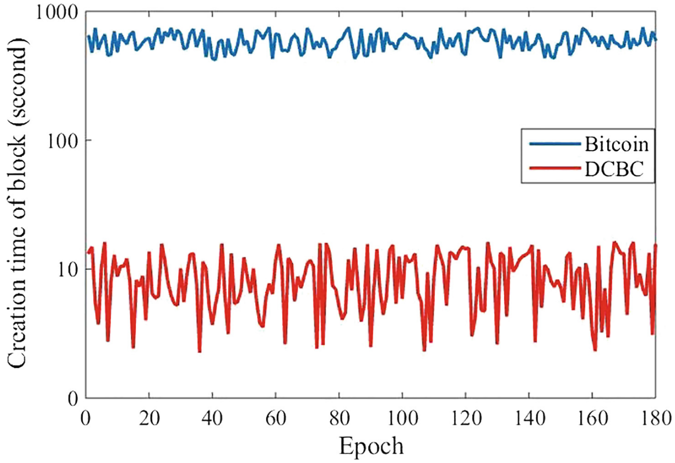

Fig. 5. Creation time of block

It is no doubt that MPT-Chain can speed up the query. But like other indexes, MPT-Chain also needs extra storage space. But it makes the query more efficient, which allows us to ignore the storage space occupied. Moreover, the storage resource is much cheaper than the computing resources. In Sect. 5.2, the MPT-Chain structure can support the holderID-based transaction query. However, its structure is not only available for accurate query, but also for range query. For example, select bachelor's degree certificates issued by a school. The above example can construct a composite search code (<IssuerID, CertTypeID>) and establish MPT-Chain based on the composite search code.

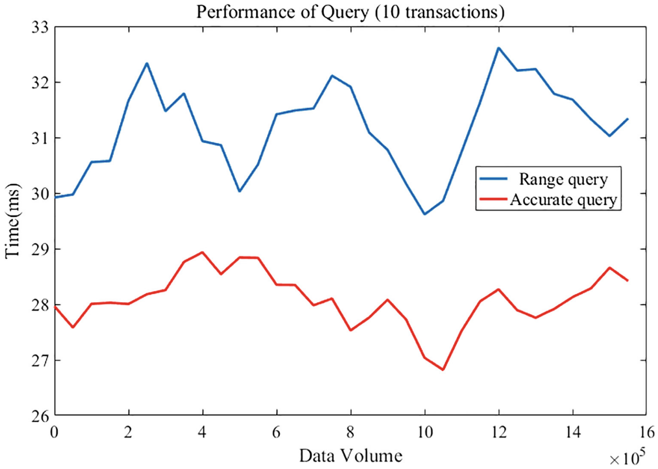

Fig. 6. Query time of MPT-Chain (10 transactions)

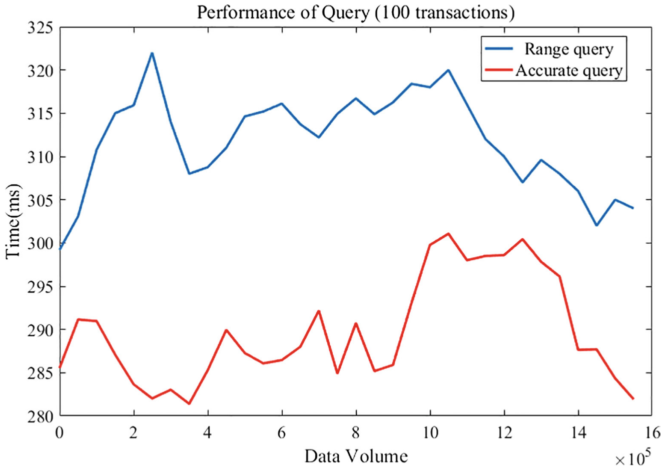

Fig. 7. Query time of MPT-Chain (100 transactions)

The experiment uses more than 1.6 million transactions to prove the efficiency of MPT-Chain, not only tests the accurate query efficiency based on holderID, but also builds multiple composite search keys (e.g. <IssuerID, CertTypeID>) to test the range query efficiency. Range query can be used for regulatory and statistical data. This paper uses multiple query statements to count the query time and calculate the average of query time. Figures 6 and 7 show the accurate query (e.g. select the historical transactions by holderID) and range query time respectively, when the query returns 10 transactions and 100 transactions. According to the experimental data, when using MPT-Chain as the query index, the query time is a millisecond-level user can tolerate.

MPT-Chain can speed up block validation because it stores intermediate values for calculating merkle root. The validation process of block needs to verify the correctness of the merkle root; the intermediate values can be reused. This paper also proves that MPT-Chain can speed up block validation by experiment; the update speed of MPT-Chain is faster than MPT. Moreover, when using account model, verification of the transaction may need to rely on the historical transaction of the account. Meanwhile, the contribution of MPT-Chain to the transaction throughput will become even greater, because of the high efficiency query of history record.

Through the above experimental results, we believe that the ECBC proposed in this paper can be applied to educational certificate management as a digital infrastructure. It not only can meet the requirements of delay and throughput, and supports millisecond query time for providing more convenient and efficient service. Therefore, we believe that, ECBC is a quiet useful educational certificate infrastructure, its application for real life can bring convenience to people's lives.

7 Conclusions

This paper had proposed an educational certificate blockchain, called ECBC, which can be used as an educational certificate infrastructure. It is permission chain that realized data security, system trust and provides management and query service for educational certificate. ECBC has a high throughput and low latency that can meet the needs of educational certificate management in real-world and has designed transaction format to protect personal privacy. The query index is called MPT-CHAIN, which can support high efficiency query, speed up block verification, and takes short time to update. We had proved our theoretical analysis and the feasibility of ECBC by using experimental data. In conclusion, it is believed ECBC proposed in this paper is a practical blockchain application which can be used as digital infrastructure to manage educational certificates and provide better service for user. Of course, our theory can not only be applied to the educational certificate. It can be applied to more fields, for example, proof of identity, proof of professional qualifications. The more areas to provide services are also what we are expanding.

Acknowledgment. This work is partially supported by National Key Research and Development Plan No. 2016YFB1000602, the Science and Technology Development Plan Project of Shandong Province No. 2016GGX101034, TaiShan Industrial Experts Programme of Shandong Province No. tscy20160404.

References

1. MIT Media Lab, educational certificates. http://certificates.media.mit.edu/
2. China Higher Educational Student Information Network (XueXinwang). http://www.chsi.com.cn/
3. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Consulted (2009)
4. Bitcoin wiki. Scalability (2015). https://en.bitcoin.it/wiki/Scalability
5. Eyal, I., Gencer, A.E., Sirer, E.G, Renesse, R.V.: Bitcoin-NG: a scalable blockchain protocol (2015). http://arxiv.org/abs/1510.02037
6. Luu, L., Narayanan, V., Baweja, K., Zheng, C., Gilbert, S., Saxena, P.: SCP: a computationally-scalable Byzantine consensus protocol for blockchains. Cryptology ePrint Archive, Report 2015/1168
7. Zyskind, G., Nathan, O., Pentland, A.: Decentralizing privacy: using blockchain to protect personal data. In: Security and Privacy Workshops, pp. 180–184. IEEE (2015)
8. Ethereum Project. https://www.ethereum.org/
9. Ethereum MPT. https://github.com/ethereum/wiki/wiki/Patricia-Tree
10. Jiang, J.: Implementing the PATRICIA data structure for compression algorithms with finite size dictionaries. In: International Conference on Data Transmission - Advances in Modem and Isdn Technology and Applications, pp. 123–127. IEEE Xplore (1992)
11. Dan, W., Sirer, E.G.: Optimal parameter selection for efficient memory integrity verification using Merkle hash trees. In: IEEE International Symposium on Network Computing and Applications, pp. 383–388 (2004)
12. Jakobsson, M., Leighton, T., Micali, S., Szydlo, M.: Fractal Merkle tree representation and traversal. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 314–326. Springer, Heidelberg (2003). doi:10.1007/3-540-36563-X_21
13. Sompolinsky, Y., Zohar, A.: Accelerating Bitcoin's transaction processing. Fast money grows on trees, not chains. In: Financial Cryptography, Puerto Rico (2015)
14. Thoughts on UTXOs by Vitalik Buterin, Co-Founder of Ethereum. https://medium.com/@ConsenSys/thoughts-on-utxo-by-vitalik-buterin-2bb782c67e53
15. Open Badges Specification. https://openbadges.org/
16. Yves-Alexandre, D.M., Erez, S., Samuel, S.W., Alex, S.P.: openPDS: protecting the privacy of metadata through safeanswers. PLoS ONE **9**(7), e98790 (2014)
17. Rackoff, C., Simon, D.R.: Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 433–444. Springer, Heidelberg (1992). doi:10.1007/3-540-46766-1_35
18. Gervais, A., Capkun, S., Karame, G.O., et al.: On the privacy provisions of Bloom filters in lightweight bitcoin clients. In: ACM Computer Security Applications Conference, pp. 326–335. ACM (2014)
19. Decker, C., Wattenhofer, R.: Information propagation in the Bitcoin network. In: 13th IEEE International Conference on Peer-to-Peer Computing (P2P), Trento, Italy, September 2013
20. IBM Hyperledger Project. https://www.hyperledger.org/