IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 5, 2019, accepted July 26, 2019, date of publication August 9, 2019, date of current version August 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2934092

Storage Mechanism Optimization in Blockchain System Based on Residual Number System

HAOJUAN MEI[1], ZHEN GAO[2], ZHAOHUI GUO[1], MING ZHAO[3], AND JINSHENG YANG[1]

[1]School of Microelectronics, Tianjin University, Tianjin 300072, China
[2]School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China
[3]Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China

Corresponding author: Zhen Gao (zgao@tju.edu.cn)

This work was supported by the Tianjin Natural Science Foundation (19JCYBJC15700) entitled research on the design and optimization for storage mechanism in blockchain system based on redundant residual number system.

ABSTRACT Huge storage volume is one of the main bottlenecks for the development of blockchain, so how to release the burden by optimizing the storage mechanism has become an important problem. Most of the current solutions would modify the architecture of blockchain, which weakens the characteristics of the decentralization, such as cloud storage. In this paper, a storage optimization mechanism based on residual number system is proposed to reduce the storage volume on each node. In addition, the recovery procedure of CRT-II (The new Chinese Remainder Theorem) is used to detect garbled data from devil nodes, which enable the proposed storage mechanism with strong fault tolerance capability. Both theoretical analysis and simulation results prove the effectiveness and reliability of the proposed scheme.

INDEX TERMS Blockchain, storage optimization, residue number system, CRT II.

I. INTRODUCTION

Since the emergence of Bitcoin in 2008, blockchain technology has come into people's vision and attracted great attention [1]. In a narrow sense, blockchain is a kind of time-series data blocks, which are connected to form a chain structure, and cryptography algorithm is applied to ensure that the distributed ledger cannot be falsified and forged [2]. Broadly speaking, blockchain is a new distributed infrastructure and computing paradigm that uses blockchain structure to verify and store data, and consensus and cryptography algorithm is applied for data updating and secure access, respectively. The blockchain technology has the characteristics of decentralization, anonymity, and suitability [3]. It has been widely concerned and applied in the fields of finance [4], [5], medical [6]–[8], education [9], [10] and food traceability [11]–[13].

Currently, one of the biggest problems in the blockchain is the huge storage volume on each node. By the end of 2018, the data stored in Ethereum was over 110 GB, and that stored in Bitcoin was over 190 GB [14]. Such a storage burden has become a key bottleneck that restricts the development of blockchain. Researchers have proposed some solutions to this problem.

The associate editor coordinating the review of this article and approving it for publication was Xiaochun Cheng.

In [15], DaYu Jia from Northeast University proposed a scalable storage mechanism of blockchain, in which a data replica allocation strategy [16] was proposed by using a distributed storage method [17]. In the storage mechanism, a complete blockchain was divided into several parts and distributed in the system. As shown in Figure 1, there are three roles for the nodes (user node, storage node, and verification node) and two kinds of chains (Position Chain and Proof of Reliability Chain, or P Chain and POR chain for short). Each node can have two or three identities at the same time. The P chain is stored on the user node and is used to record the address of the data on the storage node. The POR chain is stored on the verification node and is used to record the reliability evaluation of each storage node. When storing data, the new block is first encrypted and the number of copies is calculated according to the timeliness of the block. Then appropriate storage nodes are chosen to store the copies of the block according to the reliability information provided by the verification node. Finally, the location information of the storage node is returned to the user node and saved in the P chain. When reading data, the storage node location to the target block is first obtained from the P chain. Then the storage node returns the saved data to the user node and finally, the original data is obtained after decryption. Although this storage mechanism reduces the blockchain storage volume to a certain extent, it causes most of the data to

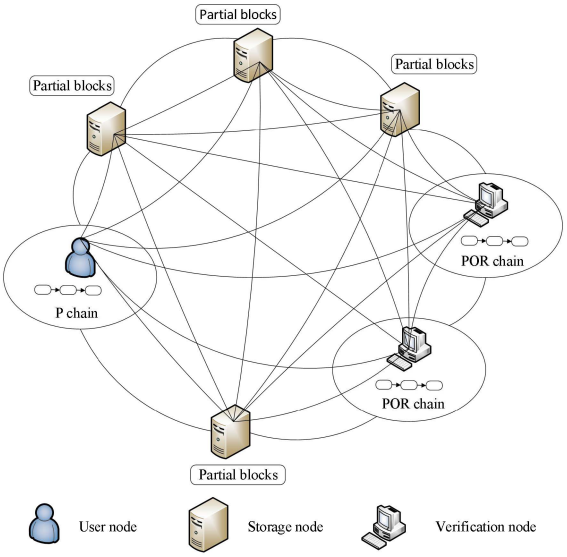


FIGURE 1. The storage mechanism of blockchain.

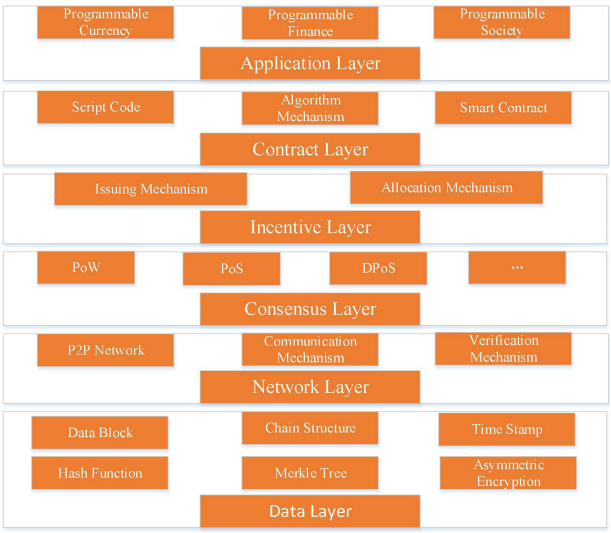


FIGURE 2. Framework of blockchain.

be stored in some nodes, which weakens the decentralization characteristic of blockchain.

The mini-blockchain (MBC) [18] was designed by J.D. Bruce as an improvement to the original blockchain. The MBC consists of 3 components: account tree, transaction tree and proof chain. The main innovation was the ''account tree'', which is basically a balance sheet storing the balance of every account. In this design, only the most recent transactions and the current account tree are stored. The MBC is thus much more scalable than the original blockchain since the MBC only grows when new accounts are created. Obviously, the traceability of earlier transactions is lost.

In this paper, a storage optimization mechanism for public blockchain is proposed based on the residual number system. In the proposed mechanism, the storage volume on each node is greatly compressed by only storing the remainder of each account to a modulus, and the updating of the account information on each node is performed independently. In addition, by combining the recovery property of CRT-II and two reliable remainders, a local partial redundant residual number system is constructed to detect garbled data from devil nodes.

The structure of this paper is as follows. In Section II, relevant theories are introduced briefly, including the structure of blockchain, consensus algorithms, storage methods and workflows, as well as the residue number system and redundant residue number system. In Section III, the proposed storage optimization mechanism is introduced, including the storage mechanism, workflow of the blockchain system based on the new storage mechanism, and the hybrid fault tolerance scheme based on CRT II and Raft. In Section IV, a case study is provided to further clarify the basic idea of the proposed storage scheme, and the complexity and overhead of the fault tolerance operation are also analyzed. Finally, the paper is concluded in Section V.

II. RELEVANT THEORETICAL KNOWLEDGE

In this section, we will introduce the background knowledge of blockchain, and the basic concepts of RNS and RRNS.

A. BLOCKCHAIN

The framework of blockchain is shown in Figure 2. Generally, blockchain can be divided into six layers: data layer, network layer, consensus layer, smart contract layer, incentive layer and application layer. Data layer encapsulates the chain structure of the underlying data blocks, the related asymmetric public-private key data encryption and timestamp technologies; Network layer includes distributed networking mechanism, data dissemination mechanism, data validation mechanism and P2P (Peer to Peer) networking technologies; Consensus layer mainly encapsulates all kinds of consensus algorithms that can be used to choose the node for generating the new block; Incentive layer mainly is responsible for the distribution mechanism of economic incentives, and usually appears in the public chain; Contract layer is composed of all kinds of scripts, algorithms and intelligent contracts and is the basis of blockchain programmability; Application layer defines various application scenarios and provides programmable interfaces to users to customize, initiate and execute contracts. The proposed scheme is mainly related to the data layer, network layer and consensus layer.

1) CONSENSUS ALGORITHM

In a distributed system, some nodes may become unreliable unintentionally (damaged nodes) or intentionally (devil nodes). The consensus algorithm is used to ensure the consistency of data in the distributed system even there are unreliable nodes in the network. Two commonly used consensus algorithms are introduced in this subsection. Proof of Work is the main consensus algorithm applied in public blockchain. And Raft consensus will be used in the proposed scheme.

a: PROOF OF WORK

Proof of Work (PoW) is the consensus algorithm applied in Bitcoin and Ethereum. In the Bitcoin system, the block header contains the harsh of the previous block, the information about the current block and a nonce. The nodes that want to get the right to generate this block (usually called miners) need to find the nonce that makes the harsh value of the block header smaller than a pre-defined threshold. Once a node finds such a nonce, the generated block would be broadcasted in the network, and other nodes would check whether the announced nonce meets the requirement. Once verified, this block would be appended to the blockchain on each node. Since solving such a 'hard problem' requires a huge consumption of calculation resources and electrical power, the correct answer (the nonce) can be proof of hard work.

b: RAFT ALGORITHM

In the distributed system, Paxos is a famous consensus algorithm [19], but its implementation complexity is very high. Raft algorithm is an optimization of Paxos [20] for practical application with low complexity.

In Raft, nodes are divided into three roles: leader, candidate and follower. Leader nodes are responsible for processing requests from clients and synchronizing logs to the follower. Candidate nodes are those may be chosen to be a header. Follower nodes are responsible to respond to requests from leader nodes.

The raft is divided into two stages, leader selection and transaction processing by the leader. Time is divided into terms in Raft, and a new leader is selected for each term. When a new term starts, all nodes change the state from follower to candidate and initiate voting requests to other nodes. Each node decides whether to vote based on whether the log content provided by the candidate is consistent with its own log content. If a candidate receives votes from more than half of the nodes, its state will be changed from candidate to leader, and start sending a heartbeat to other nodes to maintain the state. Followers would maintain a timer to monitor the heartbeat from the leader. If the heartbeats are not received within the time period, new leader selection would be initiated. After the leader is elected, all transaction operations must be processed by the leader. When a transaction from a client is received, the leader appends it to local log and forwards it to followers. All followers append the new transaction to their logs and return a confirmation to the leader. The leader will resend the new transaction to the followers until they acknowledge.

By applying the Raft algorithm, a small number of devil nodes will not affect the availability of the system. Since the proportion of devil nodes in the network is small and Raft needs more than half of the votes to elect the leader, the data stored on the leader node is reliable. However, the leader selection and transaction broadcasting process consume much network resources, so Raft algorithm is more suitable for small scale distributed network.

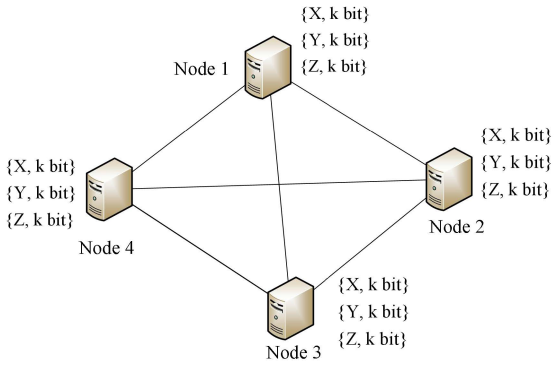


FIGURE 3. Blockchain storage mechanism.

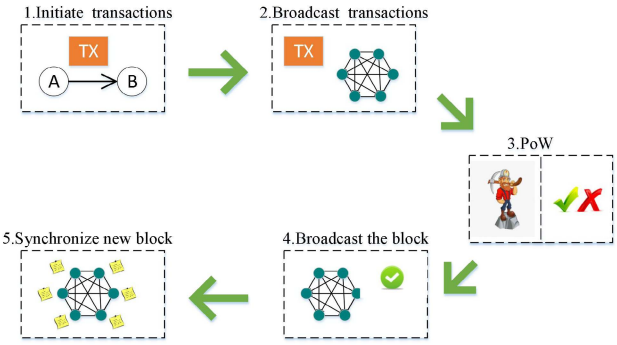


FIGURE 4. Workflow diagrams of a typical blockchain system.

2) STORAGE MECHANISM AND DEVIL NODES

In the traditional blockchain system, each node stores a complete ledger. As shown in Figure 3, for an account information X, a certain number of bits (e.g. k bits) are used to store it on each node. If the number of all accounts is N_a, the total storage volume on each node can be approximated as N_a*k.

As a decentralized system, devil nodes would exist in the blockchain system, especially in the public chain. The devil nodes may try to tamper the account information and destroy the data consistency. In this paper, two assumptions for the devil nodes are adopted [21]–[23]. One is that the devil nodes only account for a small proportion of all nodes in the network. The other is that the tamping behavior of devil nodes are independent.

3) BLOCKCHAIN WORKFLOW

As shown in Figure 4, the workflow of a blockchain system is introduced as follows. The tolerance to 'fault block' from devil nodes is achieved during the synchronization operation in the step.

a) Node B receives a transaction from node A;
b) Node B broadcasts the transaction to the whole network. After the transaction is received and verified on each node, it will be packed into a local block with a certain number of other valid transactions;
c) Each node tries to get the right to generate the new block according to the consensus algorithm, such as mining based on the PoW requirement;

d) Once a node generates a valid block, the block would be broadcasted to the whole network;
e) Each node verifies the PoW of the new block and then synchronizes the new block to the local chain.

B. RESIDUE NUMBER SYSTEM

1) BASIC THEORY

Residue number system [24] (RNS) is defined by a modulo set $\psi_n = \{m_1, m_2, \cdots, m_n\}$ with coprime numbers. The dynamic range of the RNS is [0, M], where $M = \Pi_{i=1}^{n} m_i$. In the dynamic range, a number X can be uniquely represented by a remainder vector $\Phi = \{x_1, x_2, \cdots, x_n\}$, where $x_i = X \ mod \ m_i = |X|_{m_i}, i = 1, 2, \cdots, n$.

Chinese remainder theorem (CRT) is usually used to recover X from the corresponding remainder vector Φ, and the recovery process is expressed as equation (1):

$$X = |\sum_{i=1}^{n} M_i |M_i^{-1}|_{m_i} x_i|_{mi} \tag{1}$$

where $M_i = M/m_i$ and $|M_i^{-1} M_i|_{m_i} = 1$.

The new CRT II is an improvement of CRT [25] for lower implementation complexity. If the remainder vector Φ is divided into two subsets $\{x_1, \ldots, x_q\}$ and $\{x_{q+1}, \ldots, x_n\}$, where $q = [n/2]$, two numbers X_1 and X_2 could be obtained based on the two subsets, respectively, according to the normal CRT. Then the final value X could be calculated as

$$X = X_2 + |k_0(X_1 - X_2)|_{m_1 \ldots m_q} m_{q+1} \ldots m_n \tag{2}$$

in which the parameter k_0 is chosen such that $k_0 * M_2 = 1 \ mod \ M_1$, where $M_1 = \prod_{i=1}^{q} m_i$ and $M_2 = \prod_{i=q+1}^{n} m_i$. And the values of X_1 and X_2 could also be calculated by applying the remainder vector partition and equation (2). Based on such recursive partition, the RNS system with n remainders is finally divided into n/2 minimum RNS systems with 2 remainders. Then the recovery of X starts with CRT over the n/2 pairs of remainders, and the n/2 results make another n/4 pair of remainders. This process is repeated for $\log_2 n$ times to generate the original value of X. Figure 5 shows a simple case for n = 8.

2) PARALLELIZATION PROPERTY OF RNS

Based on the property of modulus operation, the remainder vector of the linear operation result between two integers X and Y could be directly obtained by the same linear operation on the two corresponding remainder vectors ($\Phi x = \{x_1, x_2, \cdots, x_n\}$ and $\Phi y = \{y_1, y_2, \cdots, y_n\}$) over the modulo set $\psi_n = \{m_1, m_2, \cdots, m_n\}$. This property can be

$$\begin{cases} x_i = X \ mod \ m_i \\ y_i = Y \ mod \ m_i \\ (ax_i \pm by_i) \ mod \ m_i = (aX \pm bY) \ mod \ m_i \end{cases} \tag{3}$$

in which a and b are both integers. Based on this property, RNS is usually applied for parallelization of linear operations.

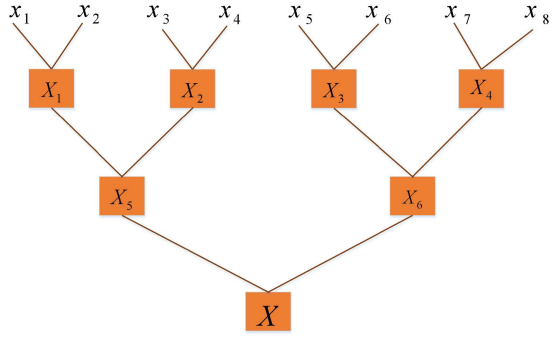


FIGURE 5. Recovery based on CRT II for an 8-element modulo set.

C. REDUNDANT RESIDUE NUMBER SYSTEM

1) BASIC THEORY

Redundant residue number system [26], [27] (RRNS) is formed by adding two or more redundant residue bases in the RNS and is used to detect and correct errors in the redundant remainder vector. The modulo set in RRNS can be expressed as $\psi_n = \{m_1, m_2, \cdots, m_h, m_{h+1}, \cdots, m_{h+r}\}$, which is composed of h information bases and r redundant bases ($n = h + r$). The RRNS enlarges the dynamic range to $[0, M_T - 1]$ ($M_T = \prod_{i=1}^{n} m_i$), which is further divided into the effective dynamic range [0, M-1] ($M = \prod_{i=1}^{h} m_i$) is and the invalid dynamic range $[M, M_T - 1]$ is. In RRNS, the effectiveness and validation of the recovered result are determined by the overflow judgment theorem.

2) OVERFLOW JUDGMENT THEOREM [28]

Assuming $\Phi_n^{'} = \{y_1, y_2, \cdots, y_n\}$ is the remainder vector of an integer Y (smaller than M) over the modulo set ψ_n, but some of the remainders (less than r) are changed to be wrong, we can define a subset $\Phi_l^{'}$ ($l > h$) in which the number of correct remainders is larger than or equal to h. Then if the value recovered based on $\Phi_l^{'}$ is within the invalid range (> M), it can be confirmed that the value is not correct and wrong remainders are included in $\Phi_l^{'}$. If the value is within the effective dynamic range (< M), it can be confirmed to be correct and all remainders in $\Phi_l^{'}$ are correct. Such verification of the recovered value is called the overflow judgment theorem and is applied in RRNS for fault tolerance.

III. NEW STORAGE OPTIMIZATION MECHANISM BASED ON RNS

This section, we will introduce the proposed storage mechanism, workflow of the blockchain system based on the new storage mechanism, and the hybrid fault tolerance scheme based on CRT II and Raft.

A. PRINCIPLE OF NEW STORAGE MECHANISM

In the proposed design, a modulo set $\psi_n = \{m_1, m_2, \cdots, m_n\}$ is pre-defined, and each node picks one modulo from it when joining the network. Then each node only stores the remainder of the account information over the picked modulo. As shown in Figure 6, the remainders of the same account information over different modulus are stored on different nodes, so an RNS is actually constructed in

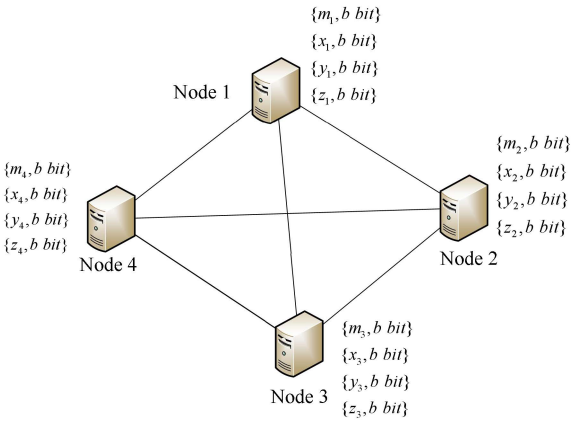


FIGURE 6. Storage optimization mechanism based on RNS.

a distributed way. According to the parallelization property of RNS for linear operation, the updating of account information could be performed independently on each node as shown equation (4), in which ΔX is the change of an account X. When a transaction needs to be verified on a node, the node needs to collect all the remainders of the related account from other nodes and applies CRT II to recovery the complete account information. The fault tolerance during the recovery will be introduced in detail in subsection C.

$$\begin{cases} x_i = X \ mod \ m_i \\ \Delta x_i = \Delta X \ mod \ m_i \\ (x_i \pm \Delta x_i) \ mod \ m_i = (X \pm \Delta X) \ mod \ m_i \end{cases} \tag{4}$$

Since the modulo could be much smaller than the value of the account, a small number of bits (e.g. b) are stored instead of k bits. Based on this storage mechanism, the compression rate of storage volume for the account information can be calculated as

$$\beta = \frac{b}{k} \tag{5}$$

A key point of the proposed storage mechanism is the design of the modulo set $\psi_n = \{m_1, m_2, \cdots, m_n\}$. First, equation (5) shows that the storage efficiency of the mechanism is determined by the bit-width b. The smaller is b, the higher is the storage efficiency of the mechanism. Second, since the original account information is a k-bit value, there should be enough large prime numbers among $[0, 2^b - 1]$ so that their product (the upper limit of the dynamic range of ψ_n) is greater than $2^k - 1$. An example of the design of the modulo set ψ_n will be given in the case study in Section IV.

B. BLOCKCHAIN SYSTEM BASED ON NEW STORAGE

Based on the storage mechanism proposed in subsection A, a new blockchain system can be constructed, and its workflow is shown in Figure 7. Compared with that of the general blockchain system described in Section II subsection A, the workflow of the new blockchain system has two differences. One is that each node only stores and updates the remainder of the account information over the local modulo.

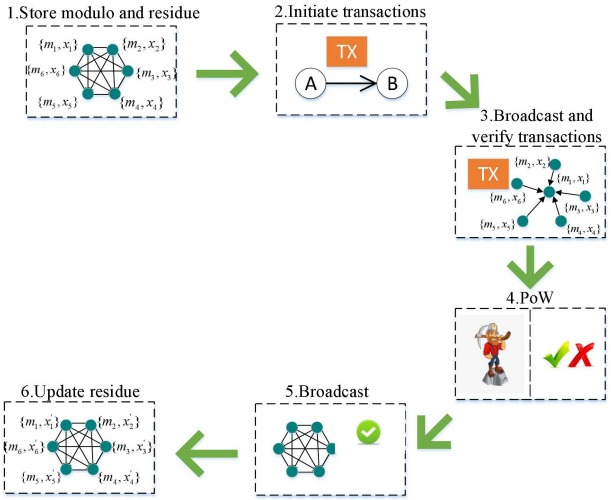


FIGURE 7. The workflow of a new blockchain system.

The other is that the transaction verification should be performed after collecting the complete remainder vector.

In addition, when a new node joins the network, synchronization will be performed on that node, and the procedure is described below:

1) Select a modulo from the modulo set ψ_n randomly;
2) Collect modulo and remainders of all accounts from other nodes;
3) Recover the complete information of all accounts;
4) Store the remainder of each account information to the local modulo.

C. A HYBRID FAULT TOLERANCE SCHEME BASED ON RAFT AND CRT II

In this paper, a hybrid fault tolerance scheme is proposed based on Raft and CRT II. The raft is used to maintain the data consistency for the two remainders (x_1 and x_2) of the account information over the first two modulus in the modulo set ψ_n (m_1 and m_2). Since the scale of the sub-network of nodes that select m_1 or m_2 is only 1/n of the whole network, the communication cost and time overhead brought by Raft would be acceptable.

With two reliable remainders of an account, the detection of the fault remainders of the account information over other modulus could be achieved during the account recovery based on CRT II. Based on the introduction in Section II subsection B, CRT II starts with CRT over $n/2$ pairs of remainders. In our design, the pair of $\{x_1, x_2\}$ is used to form a RRNS with another pair of remainders $\{x_i, x_j\}$, where $x_i = X \bmod m_i$ and $x_j = X \bmod m_j$. In this RRNS system, m_i and m_j are the information bases, and m_1 and m_2 are the redundant bases, so the upper limit of the effective dynamic range can be calculated as $M = m_i m_j$. Since x_1 and x_2 are known to be correct, the overflow judgment theorem can be used to check whether x_i and x_j are valid. If the result of CRT over $\{x_1, x_2, x_i, x_j\}$ is smaller than M, we can conclude that both x_i and x_j are valid. Otherwise, there must exist a fault value

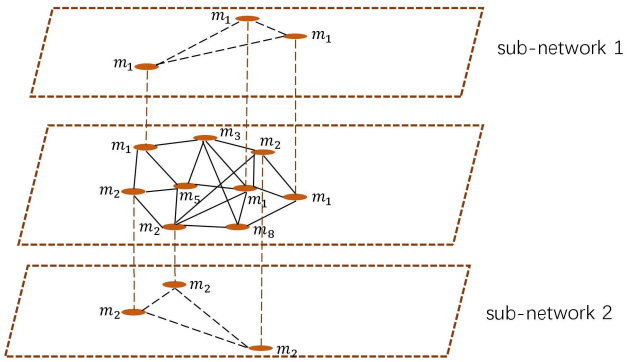


FIGURE 8. Logical partition of the whole network in the blockchain system.

among x_i and x_j. In this case, we can further form another two RRNSs as $\{x_1, x_2, x_i\}$ and $\{x_1, x_2, x_j\}$. The upper limit of the effective dynamic range of these two RRNSs are $M = m_i$ and $M = m_j$, respectively, then the overflow judgment theorem can be applied again to identify the fault remainder. The fault remainder needs to be replaced with the one from another node with the same module (m_i or m_j). By repeating this process to all the $(n/2 - 1)$ pairs of remainders (except the pair of $\{x_1, x_2\}$), we can finally get a complete remainder vector with n correct remainders, based on which the correct account information X could be recovered by normal CRT II.

Based on the above description, the whole network could be logically divided into three sub-networks as shown in Figure 8. Sub-network 1 and 2 are formed by nodes with modulo of m_1 or m_2 and are synchronized based on the Raft algorithm. Sub-network 3 is formed by all other nodes and is synchronized based on CRT II under the support of the sub-network 1 and 2. The synchronization within sub-network 1 and 2 is mainly achieved by network communication (change information and vote), and the computational complexity on each node is low (only comparison). So in our design, the scale of sub-network 1 and 2 is as small as possible, so that the network resource consumption and the synchronization delay are as small as possible. While for sub-network 3, the network scale is much larger, but the exchange of remainders could be finished between adjacent nodes, and the fault detection is mainly performed on the node. In this way, the hybrid fault tolerance architecture achieves an elegant trade-off between synchronization performance, response delay, network communication burden and the computational complexity on each node.

IV. CASE STUDY AND PERFORMANCE EVALUATION

In this section, we will show a case study that clarifies the basic idea of the proposed storage scheme and analyzes the complexity and overhead of the fault tolerance operation.

A. DEMO SYSTEM APPLYING THE PROPOSED STORAGE MECHANISM

Based on the analysis in Section III, the main parameters of the simplified demo system (k, b, and n) are set as follows. First, the account information in both Bitcoin and Ethereum systems is a 256-bit value, so we also choose $k = 256$ in the demo system. Second, the values of b and n are chosen to be 16 and 17, respectively, for a tradeoff between storage compression rate and the network overhead for exchange of remainders for transaction verification. When $b = 16$, the storage volume on each node is compressed to be $b/k = 1/16$ of that in the normal blockchain system. Among [0, 2^{16}-1], we can find the largest 17 prime numbers as listed in Table 1, and their product is larger than 2^{256}-1, which is the maximum possible value of the 256-bit account information. In this case, the node that needs to recover the complete account information needs 17 remainders (x_i, $i = 1, 2, \ldots, 17$), and 16 remainders are exacted from other nodes, so the network resource consumption and delay should be small.

TABLE 1. 17 elements of the modulo set.

Nation	m_1	m_2	m_3	m_4	m_5	m_6	m_7
Value	65327	65353	65357	65371	65381	65393	65407
Nation	m_8	m_9	m_{10}	m_{11}	m_{12}	m_{13}	m_{14}
Value	65413	65419	65423	65437	65447	65449	65479
Nation	m_{15}	m_{16}	m_{17}				
Value	65497	65519	65521				

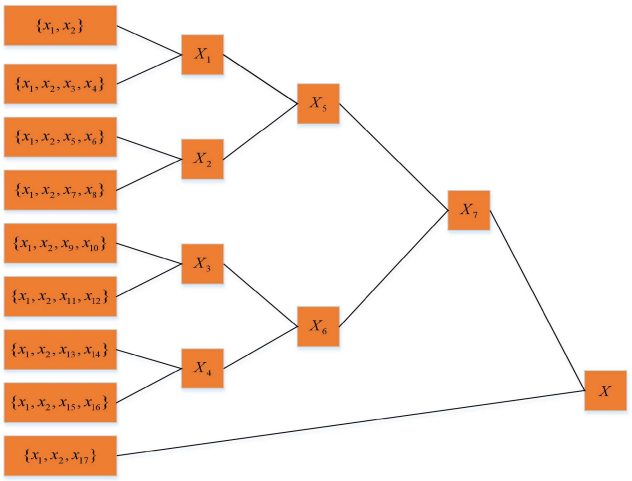


FIGURE 9. Recovery of X for the RNS with 17 remainders.

For fault tolerance, the 17 remainders are organized into 8 RRNSs and an RNS, including $\{x_1, x_2\}$, $\{x_1, x_2, x_3, x_4\}$, $\{x_1, x_2, x_5, x_6\}$, $\ldots$, $\{x_1, x_2, x_{15}, x_{16}\}$ and $\{x_1, x_2, x_{17}\}$. If all remainders are correct, the recovery of X would involve 8 CRTs over the 8 RRNSs and 9 CRTs over RNSs with two remainders as shown in Figure 9. If fault remainders exist, more data exchange and CRT operations would be involved, and the extra computational complexity and network overhead are related to the proportion of devil nodes. This will be analyzed with a probabilistic approach in the next section.

B. COMPLEXITY AND OVERHEAD ANALYSIS FOR FAULT TOLERANCE

Since the data consistency between nodes that select m_1 and m_2 is assumed to be achieved by Raft, we only need to

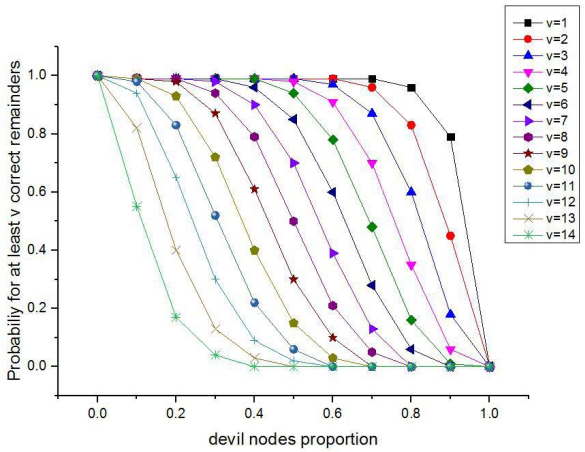


FIGURE 10. Probability for at least v correct remainders.

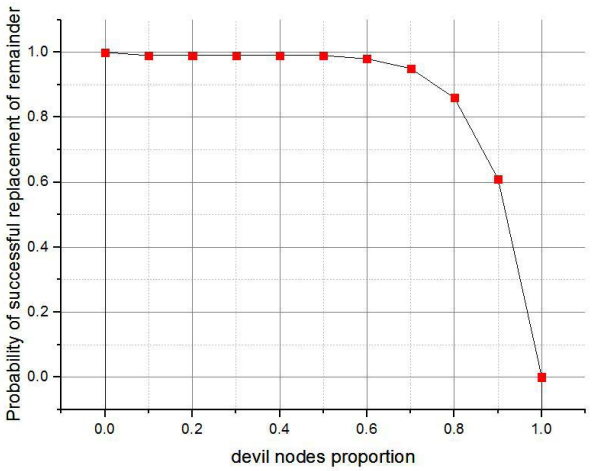


FIGURE 11. Probability of successful replacement of fault remainder.

consider the fault-tolerance problem in the sub-network 3 in Figure 8, which is composed of the nodes that select the modulo of m_3 to m_{17}. When fault remainders are detected, they would be replaced by new ones, which would produce extra network communication and CRT operations. The key factors for such overhead are the number of correct remainders that could be collected and the times of replacement for each fault remainder, and they are analyzed as follows.

1) PROBABILISTIC ANALYSIS OF THE NUMBER OF CORRECT REMAINDERS

We assume that the number of nodes in the sub-network 3 is N, among which the number of devil nodes is N_e, so the proportion of devil is denoted as $\alpha_e = N_e/N$. Since each node selects the modulo independently from the modulo set, and N is usually much larger than 15, the nodes that select the same modulo account for approximately 1/15 of all the nodes in sub-network 3, and the nodes that select different modulo are evenly distributed over the network. In this scenario, when 15 remainders for modulo of m_3 to m_{17} are collected, the probability that at least v of them are correct could be calculated as

$$p_v = \sum_{i=v}^{15} C_{15}^{i} (1-\alpha_e)^i \alpha_e^{(15-i)} \tag{6}$$

The results of equation (6) for different value of v and α_e are plotted in Figure 10. As can be seen from Figure 10, when $\alpha_e \leq 20\%$, the probability that at least 11 remainders are correct is over 80%, which means at most 4 remainders need to be replaced.

2) PROBABILISTIC ANALYSIS FOR SUCCESSFUL REPLACEMENTS

When a remainder is confirmed to be faulty, a new one would be exacted from another node with the same modulo. The probability that a correct remainder is obtained for the first time can be calculated as

$$p_1 = 1 - \alpha_e \tag{7}$$

Then the probability that a correct remainder is eventually obtained after u replacements can be calculated as

$$p_u = (1-p_1)^{u-1} p_1 \tag{8}$$

Finally, the probability of successful replacement within T times can be calculated as

$$\mathrm{P}_T = \sum_{u=1}^{T} p_u \tag{9}$$

When T=10, the probability of P_T for a different proportion of devil nodes is shown in Figure 11. From the figure we can see that, when $\alpha_e \leq 50\%$, the probability to get a correct one within 10 replacements is as high as 99.99%.

3) OVERHEAD EVALUATION FOR FAULT TOLERANCE

According to the above analysis, when the devil node's ratio $\alpha_e \leq 20$, the probability for 4 fault ones within the collected remainder vector is less than 20%, and a correct one could definitely be obtained for each of them within 10 replacements. So the total number of extra data transmissions should be less than 40 with high probability, which also implies 40 additional CRT operations. Since each remainder is only a 16-bit value and the CRT is performed over an RRNS with 4 or 3 remainders, such network and computation overhead are quite small relative to other normal operations in the blockchain system.

V. CONCLUSION

This paper proposes a new storage mechanism based on RNS to reduce the storage volume on nodes in the blockchain system. By just storing the remainder of the account information over a much smaller modulo, the volume on each node could be reduced by over 90%. The parallelization property of RNS over linear operations enables independent updates of the remainders on different nodes. In addition, by applying the Raft algorithm and constructing mini RRNSs during the CRT II operation, a hybrid fault tolerance scheme is proposed for fault remainders from devil nodes. A case study is provided to demonstrate the basic principle of the blockchain system

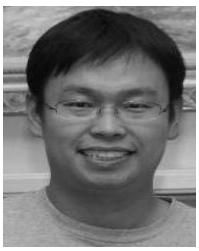
applying the new storage scheme, and the network and computation overhead of fault tolerance are evaluated to show the feasibility of the proposed scheme in practical applications.

Since the new storage scheme is also deployed in a decentralized form, it can be easily adopted in the typical blockchain system. In addition, the fault detection property of the scheme can be used to identify devil nodes in the network, and the sharing of this information with other nodes could improve the efficiency of account recovery. This would be an interesting topic to explore in future research.

REFERENCES

[1] J. A. D. Donet, C. Pérez-Sola, and J. Herrera-Joancomartí, "The bitcoin P2P network," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, 2014, pp. 87–102.
[2] S. Nakamoto. (2009). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf
[3] M. Pilkington, "Blockchain technology: Principles and applications," in *Research Handbook on Digital Transformations*, vol. 225. Paris, France: Social Science Electronic Publishing, 2016.
[4] M. Swan, *Blockchain: Blueprint for a New Economy*. Sebastopol, CA, USA: O'Reilly, 2015.
[5] K. Fanning and D. P. Centers, "Blockchain and its coming impact on financial services," *J. Corporate Accounting Finance*, vol. 27, no. 5, pp. 53–57, 2016.
[6] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *Proc. IEEE 18th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Sep. 2016, pp. 1–3.
[7] C. Pirtle and J. Ehrenfeld, "Blockchain for healthcare: The next generation of medical records," *J. Med. Syst.*, vol. 42, no. 9, p. 172, 2018.
[8] J. Yaxian, J. Zhang, J. Ma, C. Yang, and X. Yao, "BMPLS: Blockchain-based multi-level privacy-preserving location sharing scheme for telecare medical information systems," *J. Med. Syst.*, vol. 42, no. 8, p. 147, 2018.
[9] M. Sharples and J. Domingue, "The blockchain and kudos: A distributed system for educational record, reputation and reward," in *Proc. Eur. Conf. Technol. Enhanced Learn.*, 2016, pp. 490–496.
[10] J. Gresch, B. Rodrigues, E. Scheid, S. S. Kanhere, and B. Stiller, "The proposal of a blockchain-based architecture for transparent certificate handling," in *Proc. Int. Conf. Bus. Inf. Syst.*, 2018, pp. 185–196.
[11] D. Wang, "The food safety trace-ability technology based on blockchain," *Big Data Time*, 2018.
[12] G. Perboli, S. Musso, and M. Rosano, "Blockchain in logistics and supply chain: A lean approach for designing real-world use cases," *IEEE Access*, vol. 6, pp. 62018–62028, 2018. doi: 10.1109/ACCESS.2018.2875782.
[13] J. Li and X. Wang, "Research on the application of blockchain in the traceability system of agricultural products," in *Proc. 2nd IEEE Adv. Inf. Manage., Communicates, Electron. Automat. Control Conf. (IMCEC)*, May 2018, pp. 2637–2640.
[14] *Blockchain Monitoring*. Accessed: Dec. 8, 2018. [Online]. Available: https://blockchain.info/
[15] D. Jia, J. Xin, Z. Wang, W. Guo, and G. Wang, "Storage capacity scalable model for blockchain," *J. Frontiers Comput. Sci. Technol.*, vol. 12, no. 4, pp. 525–535, 2018. doi: 10.3778/j.issn.1673-9418.1709032.
[16] Y.-C. Tung, K. C.-J. Lin, and C.-F. Chou, "Bandwidth-aware replica placement for peer-to-peer storage systems," in *Proc. IEEE Global Telecommun. Conf.*, Dec. 2011, pp. 1–5.
[17] W. S. Ng, B. C. Ooi, K.-L. Tan, and A. Zhou, "PeerDB: A P2P-based system for distributed data sharing," in *Proc. Int. Conf. Data Eng.*, Mar. 2003, pp. 633–644.
[18] B. F. Franca. (2014). *Privacy and Pruning in the Mini-Blockchain*. [Online]. Available: http://cryptonite.info/files/Anonymity_account_tree.pdf
[19] N. Santos and A. Schiper, "Optimizing Paxos with batching and pipelining," *Theor. Comput. Sci.*, vol. 496, pp. 170–183, Jul. 2013.
[20] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Annu. Tech. Conf.*, 2014, pp. 305–319.
[21] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens "An on-demand secure routing protocol resilient to byzantine failures," in *Proc. 1st ACM Workshop Wireless Secur.*, 2002, pp. 21–30.
[22] D. Malkhi and M. Reiter, "Byzantine quorum systems," *Distrib. Comput.*, vol. 11, no. 4, pp. 203–213, 1998.
[23] A. Doudou and A. Schiper, "Muteness detectors for consensus with Byzantine processes," in *Proc. 17th ACM Symp. Principle Distrib. Comput.*, Puerto Vallarta, Mexico, 1997, p. 315.
[24] H. L. Garner, "The residue number system," *IRE Trans. Electron. Comput.*, vol. EC-8, no. 2, pp. 140–147, 1959.
[25] Y. Wang, "New Chinese remainder theorems," in *Proc. Conf. Rec. 32nd Asilomar Conf. Signals, Syst. Comput.*, 1998, pp. 165–171.
[26] F. Barsi and P. Maestrini, "Error correcting properties of redundant residue number systems," *IEEE Trans. Comput.*, vol. C-22, no. 3, pp. 307–315, Mar. 1973.
[27] S. Timarchi and K. Navi, "Efficient class of redundant residue number system," in *Proc. IEEE Int. Symp. Intell. Signal Process.*, Oct. 2008, pp. 1–6.
[28] L.-L. Yang and L. Hanzo, "Redundant residue number system based error correction codes," in *Proc. IEEE 54th Veh. Technol. Conf.*, Oct. 2001, pp. 1472–1476.

HAOJUAN MEI was born in Henan, China, 1995. She is currently pursuing the master's degree with the School of microelectronics, Tianjin University. Her research interests include integrated circuits and block chain technology.



ZHEN GAO received the B.S., M.S., and Ph.D. degrees in electrical and information engineering from Tianjin University, China, in 2005, 2007, and 2011, respectively, where he has been an Associate Professor, since 2014. From 2008 to 2010, he was a Visiting Scholar with GeorgiaTech. From 2011 to 2014, he was a Postdoctoral Researcher with the Wireless and Mobile Communication Research Center, Tsinghua University, China. His research interests include fault-tolerant signal processing, software defined radio, and block chain.

ZHAOHUI GUO was born in Shanxi, China, 1996. He is currently pursuing the master's degree with the School of microelectronics, Tianjin University. His research interests include microwave and blockchain Technology.

MING ZHAO is currently a Professor with the Beijing National Research Center for Information Science and Technology, Tsinghua University. His current research interests include key technologies of block chain systems, wireless and personal communications, and software radios.

JINSHENG YANG was born in Shanxi, China, in 1965. He is an Associate Professor with the School of Microelectronics, Tianjin University. His primary research interests include radio wave transmission and block chain technology.