IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Design and Optimization for Storage Mechanism of the Public Blockchain Based on Redundant Residual Number System

ZHAOHUI GUO[1], ZHEN GAO[1*], HAOJUAN MEI[1], MING ZHAO[2], and JINSHENG YANG[1]

1. Tianjin University, Tianjin 300072, China
2. Tsinghua University, Beijing 100084, China

ABSTRACT Blockchain has broad development potential and application prospects. At present, huge storage volume one each node becomes one of the primary bottlenecks that restrict the expansibility of blockchain. Optimization for storage mechanism has become a necessary way to accelerate the improvement of blockchain. There are a variety of solutions to relieve the storage burden by modifying the architecture of blockchain, but most of them weaken the decentralization property of the system, which is the core advantages of the blockchain. In this paper, an optimization scheme based on the redundant residual number system is proposed to dramatically reduce the storage volume of nodes in the blockchain system, and fault tolerance mechanism is designed based on the new storage scheme. Simulation experiments are performed to prove the effectiveness of the proposed mechanism.

INDEX TERMS Blockchain, fault tolerance, redundant residual number system, optimization for storage mechanism;

I. INTRODUCTION

Blockchain is the underlying technology of cryptocurrency such as Bitcoin[1]. In essence, blockchain is a decentralized, trustless, collectively maintained distributed database, which is traceable and tamper-resistant[2]. So it provides a new way to solve the problems of poor reliability, low security and high trust cost in the current centralized mode[3]. The blockchain has been widely researched and applied in fields of finance[4], medical[5], education[6], and food safety[7].

Compared with current centralized trading systems, the two primary bottlenecks in the public blockchain are the number of transactions per second and the huge storage volume. This paper focuses on the latter issue. By the end of 2018, the amount of data stored in Ethereum exceeded 110GB meanwhile exceeded 190GB in Bitcoin[8]. Problems in storage mechanism have become a primary bottleneck restricting the improvement of blockchain[9].

There are a variety of solutions for this problem contributed by scholars, blockchain communities and enterprises. In the Bitcoin White Paper by Nakamoto called as "The father of Bitcoin", all nodes are divided into three categories: “full node”, “light node” and “miner node”[10]. The light node initiates transfer transactions and verifies the payment after transactions are completed. The light node only needs to keep a copy of the block header of the longest proof-of-work (POW) chain. The full node verifies transactions and broadcasts them to all miner nodes. Each full node must keep a copy of the complete block of the longest POW chain. The miner node collects new transactions into its block and works on looking for a difficult nonce for this block. When a miner node finds the answer, it broadcasts the block to all nodes. Such a division based on the hardware capability of nodes can lower the entry threshold of blockchain so that the terminals with poor hardware performance such as mobile phones will be able to access the blockchain more easily. In Ethereum, sharding is used to change the mode of transaction verification [11][12]. Traditionally, each transaction must be verified by all nodes in the network. Sharding makes it possible to verify each transaction by parts of nodes so that the stored data on each node is only part of the longest POW chain but not a copy of the complete chain. In [13], InterPlanetary File System (IPFS) is proposed for data storage in the blockchain system. In this model, the miner nodes deposit the transactions data into the IPFS network and pack the returned IPFS hash of transactions into the block. Due to the characteristics of the IPFS network and the features of the IPFS hash, the data volume is greatly reduced [13], so this storage scheme is also adopted in EOSIO system [14]. In [15], ‘Section Blockchain’ is proposed to reduce the storage volume on each node. In this storage system, all nodes are equal and only need to keep block headers of POW

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

chain, the certain number of blockchain fragments, the certain number of database snapshots, and a table linking fragments and database snapshots[14]. The fragment is a section of POW chain and the snapshot contains the status of transaction records. Because there are only block headers stored on each node, the storage volume of each node is reduced greatly.

Although the current solutions can alleviate the storage problem of blockchain partly, they generally have two main disadvantages as following:

1) The decentralization property is weakened. For example, in Bitcoin, based on "role differentiation", it's possible to verify a payment without running a "full node", but the verification and execution of transactions initiated by "light node" cannot be done without "full node" or "miner node". So the "light nodes" must be supported by "full node", which increases the dependence on "full nodes" and weakens the decentralization of the system.
2) Almost all of the solutions are complicated. For example, in Ethereum, based on "sharding", there will be fragment merging for transaction verification. Meanwhile, there also are complicated rules concerned about inter-fragment communication to ensure sufficient security and consistency[16]. Section blockchain has the same problem.

In order to reduce the storage volume of account information on each node without destroying the decentralization property, this paper proposes a new storage mechanism based on redundant residual number system and a fault tolerance design within the framework. This new storage scheme is aimed at the public blockchain such as Bitcoin and Ethereum, in which all nodes are equal and some devil nodes may exist in the distributed network.

In the proposed scheme, the information for each account is partly stored on each node as its residual number relative to a module assigned to the node, so residual number system (RNS) is actually constructed in a distributed form in the network of the blockchain system. Since the modules could be an order of magnitude smaller than the value of normal account information, the storage volume of account information on each node could be dramatically reduced. In addition, to make the system capable to detect evil nodes, the redundant modules are distributed in the network, so that a redundant residual number system (RRNS) is formed and its error detection and correction capability are utilized to bring fault tolerance to the blockchain system. Furthermore, since the updates of account information on different nodes are completely parallel and distributed, the implementation complexity of the proposed storage scheme is very low. The remaining part of this paper is organized as follows. In Section II, the background knowledge of blockchain is introduced, including the basic concepts, workflow, storage mechanism, and fault tolerance. In Section III, the principle of RNS and RRNS are introduced, including the basic structure, the linear operation in RNS and the theorem on error detection and correction in RRNS. The new storage mechanism based on RRNS are introduced in Section IV, and a simplified storage system is established for idea demonstration and performance evaluation in Section V. A more general analysis of fault tolerance is given in Section VI. Finally, the paper is concluded in Section VII.

II. BACKGROUND KNOWLEDGE OF BLOCKCHAIN

In this section, we will introduce the background knowledge of blockchain related to this paper, including the basic concepts, workflow, storage mechanism, and fault tolerance.

A. BASIC CONCEPTS

Blockchain is a distributed database, and the unit of the data is a block, which is consisted of the header and the body**.** The header contains the attribute information of the block, including the previous hash, the timestamp and a nonce representing the POW. The body contains transaction records [17]. The block is generated every 10 minutes in Bitcoin and 15 seconds in Ethereum[18]. In the blockchain, the volume of header generally is fixed, but the volume of the body is growing fast as the number of nodes and transactions increases. The account information is essential for blockchain and occupies huge storage volume. The account information is stored in the body of the block as unspent transaction output in Bitcoin. Meanwhile, it's stored in the local database called stateDB in Ethereum. All blocks are linked to a traceable chain called as blockchain by previous hash and timestamp. The POW is actually that CPU works on looking for a nonce meeting the conditions through repeated complex operations[10]. The POW is used to ensure data consistency and solve the problem of fault tolerance in the blockchain.

B. WORKFLOW OF BLOCKCHAIN

The nodes in the public blockchain are equal and can be divided into transfer nodes denoted by TN, consensus nodes denoted by CN and storage nodes denoted by SN according to different functions. TNs initiate transactions, CNs verify, transmit and execute transactions from TNs, and SNs store data. There can be an intersection between different nodes. The steps to run the network are as following:

1) TNs initiate transactions;
2) CNs verify, accept, transmit and execute the transactions and update the blockchain by working on POW;
3) When CN finds the nonce, it broadcasts the block to SNs;
4) SNs verify the POW and append the new block to their local chain if it is verified.

In addition, when a new storage node joins the network, synchronization would be performed on that node so that it obtains a complete copy of the blockchain information.

C. STORAGE MECHANISM

SN is used to store data. The stored data on each node is a copy of the complete blockchain. As the main part of node storage

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

that occupies huge storage volume, the account information has the following characteristics:

- The account information has fixed bit-width denoted by k.
- The operation to account information must be linear operation including addition and subtraction.

The storage scheme of blockchain can be described as figure1.

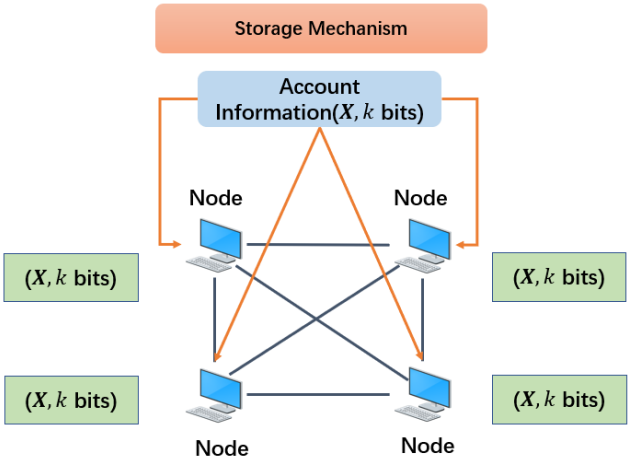


FIGURE 1. The storage mechanism of blockchain

D. FAULT TOLERANCE

Blockchain is a decentralized and open system based on the peer-to-peer network, and the data of transactions are stored on all the nodes. If some devil nodes try to tamper the local data, the data consistency will be damaged. In the peer-to-peer network, the devil nodes are also called as byzantine nodes, and their characteristics in blockchain can be described as following[19]-[21]:

- When the scale of blockchain is large enough, the devil nodes only account for a very small proportion.
- The appearance of devil nodes is considered as an independent behavior.
- The appearance of devil nodes has a constant probability.

The POW is designed in blockchain to solve the problem of fault tolerance from the economic perspective. If a node wants to find the nonce representing the POW, it must spend huge CPU effort. Once the CPU effort has been expended to make it satisfy the POW, the block cannot be changed without recoding the work. POW is essentially one-CPU-one-vote. In the blockchain, the majority decision is represented by the longest chain, which has the greatest POW effort invested in it. If a majority of CPU power is controlled by honest nodes, the authority chain will grow the fastest and outpace any competing chains. Generally, the devil nodes cannot damage the data consistency unless the CPU power controlled by devil nodes is greater than which controlled by honest nodes. This situation is called the 51% attack.

Actually, according to the characteristics of devil nodes, the CPU power controlled by devil nodes is always smaller than that controlled by the honest nodes, so it is still safe though devil nodes exist. But the risk of the 51% attack is getting higher as the CPU power becomes more and more centralized[22].

III. RNS AND RRNS

This section, we will introduce the basic concepts of RNS and RRNS.

A. BASIC PRINCIPLE OF RNS

(1) Basic Structure

The RNS consists of a set of modules $\psi_n = \{m_1, m_2, m_3 \dots m_n\}$ (called as "residue bases"), any two of which are relatively prime. The multiplication of all modules $M = \prod_{i=1}^{n} m_i$ is called "dynamic range". If X is a positive integer less than $M - 1$, it can be uniquely represented as a remainder vector $\Phi_n = \{x_1, x_2, x_3 \dots x_n\}$ by performing modular arithmetic on the residue bases[23]. This process can be expressed as a congruence formula as

$$X \equiv x_i (mod\ m_i), i = 1,2,3, \dots, n \quad (1)$$

In turn, if ψ_n and Φ_n are known, the value of X could be restored by equation (2),

$$X = \left| \sum_{i=1}^{n} M_i \left| M_i^{-1} \right|_{m_i} x_i \right|_M \quad (2)$$

in which $M_i = \frac{M}{m_i}$. $\left| M_i^{-1} \right|_{m_i}$ denotes the inverse element of M_i on m_i, $\left| M_i M_i^{-1} \right|_{m_i} = 1$. The recovery process based on the formula (2) is called Chinese Remainder Theorem (CRT) [24] and can also be written as:

$$X = \left| \sum_{i=1}^{n} w_i x_i \right|_M \quad (3)$$

in which $w_i = M_i \left| M_i^{-1} \right|_{m_i}$.

The basic structure of RNS can be described as figure2:

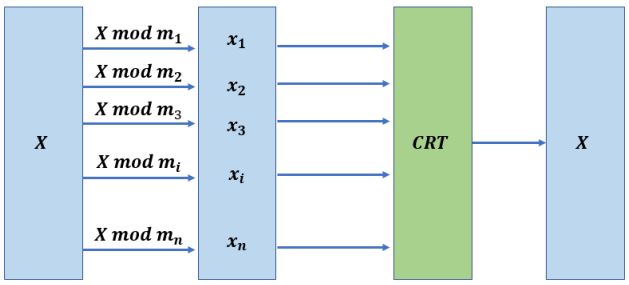


FIGURE 2. The basic structure of RNS

(2) Linear Operation in RNS

The linear operation in RNS is based on the modular arithmetic. Assuming X and Y are two integers, and their remainders to modular m_i are x_i and y_i, respectively, the remainder of the linear operation of X and Y equals the remainder of the same operation of x_i and y_i[25], as shown in equation (4).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

$$\begin{cases} x_i = X \bmod m_i \\ y_i = Y \bmod m_i \\ (aX \pm bY)_{m_i} = (ax_i \pm by_i)_{m_i} \end{cases} \quad (4)$$

The property in equation (4) is usually used to parallelize linear processing as shown in Figure 3:

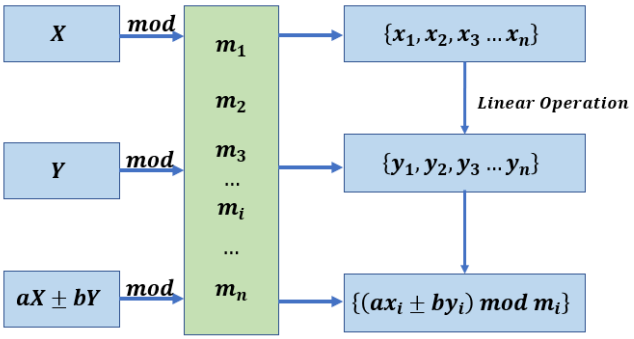


FIGURE 3. The linear operation in RNS

B. BASIC PRINCIPLE OF RRNS

(1) Basic Structure

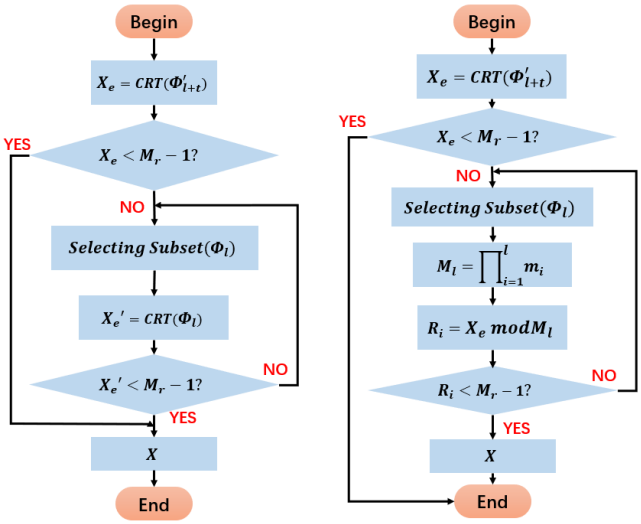
On the basis of RNS, we can construct an RRNS by adding redundant residue bases. The new module set could be expressed as $\psi_n = \{m_1, m_2, \dots, m_r, m_{r+1}, \dots, m_{r+s}\}$, in which r and s are the length of information bases and the length of redundant bases, respectively, and the total length of the residue bases is $r + s = n$. The first r bases are called information bases, and the other s bases are called redundant bases[26]. Usually, the bases in ψ_n are arranged from small to large in order. In this case, the valid dynamic range and the error range could be defined as $M_r = \prod_{i=1}^{r} m_i$ and $M_e = \prod_{i=r+1}^{n} m_i$, respectively.

(2) Error detection and correction in RRNS

The RRNS could restore the original data X when a limited number of wrong remainders exist within the remainder vector. Assuming $\Phi' = \{x'_1, x'_2, x'_3 \dots x'_n\}$ is the remainder vector of X to the residue bases, and some of them are tampered maliciously, the overflow decision theorem in RRNS for error detection could be described as following [27][28]:

Theorem1. Φ'_{l+t} denotes any $(l+t)$-dimensional subset of Φ', where $l(l \geq r)$ denotes the length of correct components inside Φ'_{l+t} and $t(t \geq 1)$ denotes the length of error components inside Φ'_{l+t}. If the recovered value based on Φ'_{l+t} according to CRT is X_e, we must have $X_e \notin [0, M_r - 1]$.

After error detection, if the number of error remainders is limited so that the product of the modules for the correct remainders is larger than M_r, there are two primary ways to correct the wrong result as shown in Figure 4. The first way is to select a subset of Φ'_{l+t} ($\Phi'_l, l \geq r$) and perform the CRT again on Φ'_l. As shown in Figure 4(a), this process could be repeated until the wrong remainders are out of Φ'_l, so that the recovered value is within the dynamic range [0, M_r-1], then this value is the correct one. Since multiple CRT operations are involved, the complexity of this method is high. In the second method as shown in Figure 4(b), CRT is only performed once to recover the value of X_e based on Φ'_{l+t}, then a subset of Φ'_{l+t} ($\Phi'_l, l \geq r$) is selected and the corresponding dynamic range [0, M_l-1] is determined. If the remainder of X_e to M_l is smaller than M_r, that remainder is the correct result.



(a) CRT on a new subset (b) Remainder on a new subset

FIGURE 4. The error detection and correction in RRNS

IV. NEW STORAGE MECHANISM BASED ON RRNS

Since the account information is the main part of the stored data on storage nodes, a basic storage mechanism based on RNS is proposed for account information in this section from the perspective of data compression, and the corresponding information updating and fault tolerance methods within the basic framework are developed according to the characteristics of the distributed network and the error detection and correction property of RRNS.

This section introduces the principle of the new storage scheme first in subsection A. Then the workflow of blockchain based on the new storage scheme is described in subsection B. The meanings of notations in this section are shown as the TABLE I for clarification.

TABLE I. THE MEANINGS OF NOTATIONS

Notation	Meaning
m_i	the module selected by each node
ψ_n	the module set consisted of m_i
X	account information
x_i	the remainder of X to m_i
ΔX	variation of X
Δx_i	the remainder of ΔX to m_i
x_{u_i}	the remainder of liner operation of x_i and Δx_i to m_i
b	the bit-width of module
k	the bit-width of account information
M_r	the threshold for overflow decision
M_c	the new and fixed threshold for overflow decision
n	the number of modules
L	the length of the subset of ψ_n
R_O	the storage volume in the traditional blockchain system
R_N	the storage volume in the proposed storage scheme
β	the compression ratio achieved by the new storage scheme

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

A. PRINCIPLE OF NEW STORAGE SCHEME

(1) Basic Storage Structure based on RNS

In the new storage structure, a module set $\psi_n = \{m_1, m_2, m_3 \dots m_n\}$ is constructed, and each storage node picks one module randomly when the node joins the blockchain system. For a system with a large number of storage nodes, the numbers of nodes that pick the same module are statistically the same. For an account information X, its remainder to m_i (x_i in equation (1)) would be stored on the node with module m_i.

When the account information is changed by ΔX due to transactions or reward, the stored data on the node with module m_i could be updated independently according to equation (4) as

$$x_{u_i} = (x_i \pm \Delta x_i) \; mod \; m_i \qquad (5)$$

in which Δx_i is the remainder of ΔX to modulo m_i.

When the complete account information needs to be recovered at the consensus nodes, a remainder vector of that account information would be collected from the storage nodes, and the recovery based on CRT could be performed according to equation (2). Such processing is shown in Figure 5.

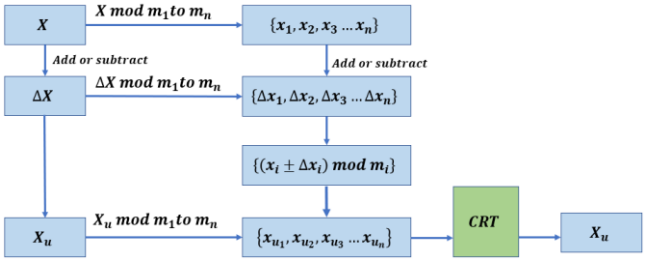


FIGURE 5. The constructed RNS in blockchain

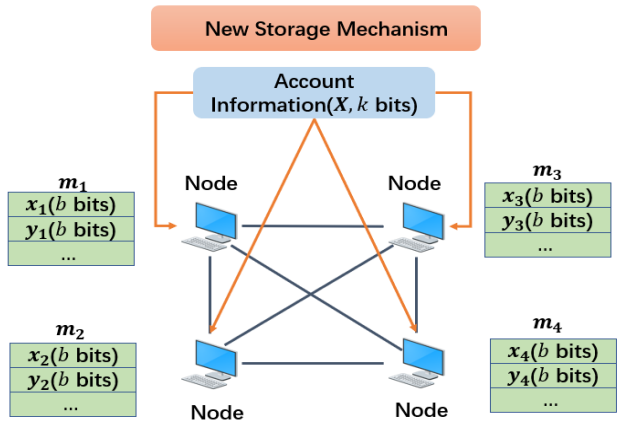


FIGURE 6. The new storage mechanism based on RRNS

Based on the above description, the structure of the RNS based storage system for account information is shown in Figure 6. Each storage node just stores a remainder of each account information to the module that assigned to the node. Since the bits width of the module (b) would be much smaller than that of the original account information (k), the storage volume on each node is reduced dramatically.

(2) Extension to RRNS for fault tolerance

In the proposed RNS based storage mechanism, some remainders may be wrong due to malicious tamper. In this case, redundant residual bases are introduced into the system for constructing an RRNS so that its error detection and correction capability could be used to recover the account information. In the presence of wrong remainders, the recovery process of account information is the same as that shown in Figure 4(b). The only difference is that, since the account information in the blockchain system is usually represented by a fixed number of bits, e.g. k = 256 in Bitcoin and Ethereum, the threshold for overflow decision can be changed from M_r to the maximum representable value of k bits $M_c = 2^k - 1$ as long as $M_r > M_c$.

To ensure $M_r > M_c$, some necessary limitations are set between the main parameters, including the bit-width of the module (b), the length of the subset of residue bases (L) and the size of residue bases (n). First, there should exist at least L prime numbers in the range of $[0, \; 2^b - 1]$, so that the product of these numbers is greater than 2^k-1. Second, n should be larger than L so that the redundant module could be used for error detection and correction.

Assuming that the storage volume on each SN in the traditional blockchain system and that based on the proposed storage scheme are R_O and R_N, respectively, the compression ratio achieved by the new storage scheme could be calculated as:

$$\beta = 1 - \frac{R_N}{R_O} = 1\text{- } b/k. \qquad (6)$$

Obviously, a smaller b would bring a higher compression ratio. However, as analyzed above, b should be large enough to ensure $M_r > M_c$.

B. BLOCKCHAIN BASED ON NEW STORAGE SCHEME

Based on the proposed storage scheme, a new blockchain system could be constructed. Relative to that in the normal blockchain system as described in section II, the workflow of the new blockchain system is shown in Figure 7. The working procedures of the system are described as following:

(1) Each SN stores the selected module and the remainders of all account information to that module;

(2) A transaction related to an account is sent from a TN to CN;

(3) To verify the transaction based on that account information, CN collects the pairs of module and remainder of that account from SNs, and recovers the complete account information;

(4) After a transaction completed, CN broadcasts the change of the account information (ΔX) to SNs (instead of the updated account information (X) in normal blockchain system);

(5) Each SN updates the stored remainder of the updated account information independently.

In addition, if a new storage node joins the network, synchronization would be performed on that node, and the standard procedures are described as following:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

(1) Select a module from the set randomly;
(2) Collect enough pairs of module and remainder of all accounts from other SNs;
(3) Recover the complete information for all accounts;
(4) Store the remainders of each account information to the local module.

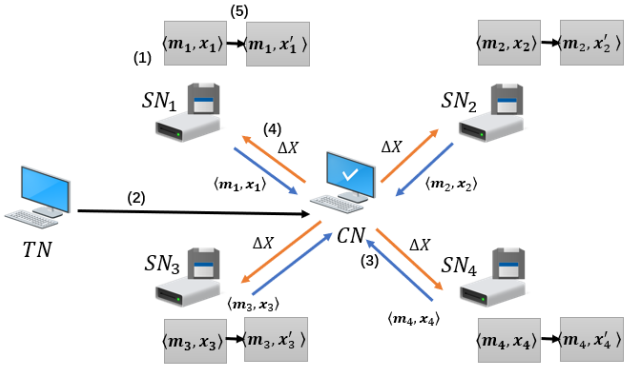


FIGURE 7. The workflow of blockchain based on the new storage scheme

V. CASE STUDY AND PERFORMANCE EVALUATION

In this section, a simplified blockchain system applying the proposed storage scheme is constructed to demo the main idea of the new storage scheme, and then its fault tolerance is analyzed in theory and evaluated by simulation.

A. DEMO SYSTEM WITH THE NEW STORAGE SCHEME

Based on the analysis in Section IV.A, the main parameters of the ideal demo system are set as the following:

1) The bit-width of complete account information is 256 (k = 256), which is consistent with that in Bitcoin and Ethereum systems;
2) The module set consists of 32 residual bases (n = 32), and the bit-width of each module is 16 (b = 16). The values of the 32 modules are listed in TABLE II. These numbers are the 32 largest primes smaller than 2^b = 65536.
3) Since the multiplication of any 17 out of the 32 modules is larger than 2^{256}-1, the minimum number of remainders that could be used to recover the account information is 17 (L = 17).
4) The number of storage nodes is 32, and the i-th storage node picks m_i as its residual base.

In this demo system, the storage compression ratio can be calculated as β = 1-16/256 = 93.75%, which means the storage volume is reduced to 1/16 of that in the normal blockchain system.

TABLE II. MODULE SET WITH 32 RESIDUAL BASES

m_1-m_4	m_5-m_8	m_9-m_{12}	m_{13}-m_{16}	m_{17}-m_{20}	m_{21}-m_{24}	m_{25}-m_{28}	m_{29}-m_{32}
65167	65183	65257	65293	65353	65393	65423	65479
65171	65203	65267	65309	65357	65407	65437	65497
65173	65213	65269	65323	65371	65413	65447	65519
65179	65239	65287	65327	65381	65419	65449	65521

B. ANALYSIS OF FAULT TOLERANCE

If N_e out of the 32 nodes tampers the remainder of account information, the number of correct remainders that the consensus node get is 32- N_e. According to the error detection and correction capability of RRNS system, if N_e >15, the consensus node would never recover the account information; if $N_e \leqslant 15$, the consensus node would recover the correct account information with a probability of 100% by trying enough times according to the procedures shown in Figure 4(b). The probability of success for one trial (one cycle in Figure 4(b)) could be calculated as

$$p = \frac{C_{32-N_e'}^{17}}{C_{32}^{17}} \times 100\% = \frac{\prod_{v=16}^{32}(v - N_e')}{\prod_{j=16}^{32} j} \times 100\% \quad (7)$$

Based on (7), the success probability by trying i times can be calculated as

$$q_i = (1-p)^{i-1}p \quad (8)$$

If the maximum number of trials is T, the success probability within T trials could be calculated as:

$$P_s = \sum_{i=1}^{T} q_i \quad (9)$$

For $T = C_{32}^{17}$, we would have P_s = 1. However, C_{32}^{17} = 565722720 is a huge number. So many trials would be a waste of computation resources of a consensus node, especially when the number of N_e is large. Considering that the proportion of devil nodes among all storage nodes should be much smaller than 15/32 in practice, T should be a reasonable small number. Figure 8 shows the success probability for T = 10^5. We can see that the success probability for $N_e \leqslant 10$ is close to 1 and that for N_e >10 decreases dramatically.

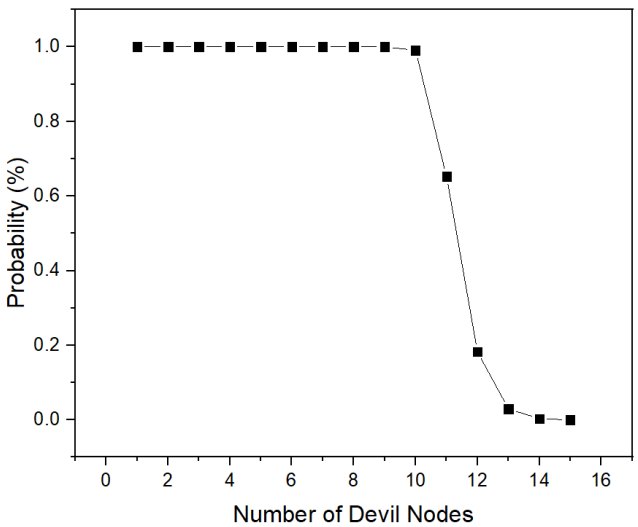


FIGURE 8. Success Probability for T = 10^5

C. SIMULATION EVALUATION OF FAULT TOLERANCE

To evaluate the theoretical analysis of fault tolerance, a simulation platform is established using Python 2.7. The program runs on a PC with Windows 10 operation system, Inter(R) Core(TM) i7-7700HQ CPU @ 2.8 GHz, and 8GB memory. The pseudocode of the program is shown in TABLE III. In this program, ψ and Φ represent the module set and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

the corresponding remainder vector, respectively. The account information (AC_i) is a random number generated from the range of [0,2^{256}-1]. Φ' represents the remainder vector in which N_e ($N_e \leqslant 15$) remainders are modified randomly to another value in the range of [0, m_i-1], respectively.
The program will be executed 20000 times to simulate the process for information of 20000 accounts in the demo system.

TABLE III. PSEUDOCODE OF PROGRAM

Output: number of trials *j* and consumed time when data is restored correctly

Begin:
$\psi = \{m_1, m_2, m_3 \dots m_{32}\}$
for ($i = 1{:}20000$)
  generate a random number AC_i from range [0,2^{256}-1]
  $\Phi_i = \{AC_i \ mod \ \psi\} = \{x_{i1}, x_{i2}, x_{i3} \dots x_{i32}\}$
  modify $N_e(N_e \leq 15)$ remainders in Φ_i randomly
  $\Phi_i' = \{x'_{i1}, x'_{i2}, x'_{i3} \dots x'_{i32}\}$
  X_{ie}=**CRT** (Φ_i')
  starttime
  if (X_{ie}<2^{256}) **then**
    X_{ie} is correct
    endtime
    return (0, endtime-starttime**)**
  else
    for (*j* =1:10^5)
      select $\{m_1', m_2', m_3' \dots m_{17}'\}$ randomly from ψ
      $M_j = \sum_{l=1}^{17} m_l'$
      $R_j = X_{ie} \ mod \ M_j$
      if (R_j <2^{256}) **then**
        break
      end if
    end for
    if (R_j <2^{256}) **then**
      R_j is correct
      endtime
      return (*j*, endtime-starttime**)**
    else
      can't restore data correctly within 10^5 times
    end if
  end if
end for
End

Figure 9 shows the success probability, represented by spots, and consumed time, represented by squares. We can see that the simulation result about the success probability is consistent with the theoretical results in Figure 8, and the consumed time increases for the larger number of devil nodes. When $N_e \leqslant 10$, the success probability is close to 1 and only 50 ms is consumed for the error correction, which indicates that the proposed new storage scheme is applicable in a practical system.

VI. DISCUSSION

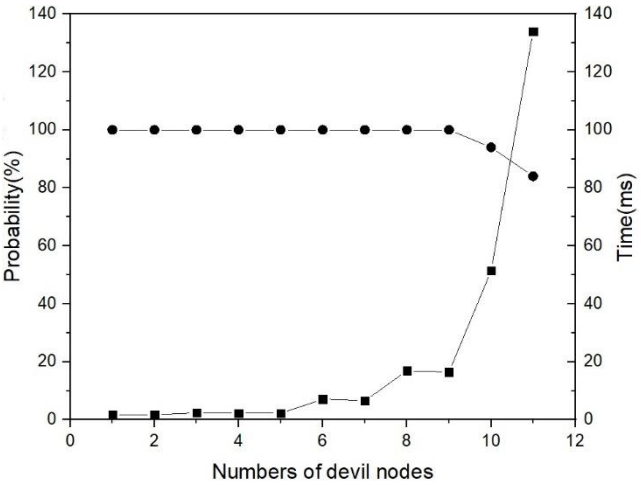


FIGURE 9. Success Probability and consumed time for T = 10^5

The demo system constructed in section V has only 32 nodes and the module for each node is different. In a practical blockchain system, there are usually thousands of nodes. If each node picks a module from the module set consisted of 32 residual bases randomly, the module set will be uniformly distributed in the network, and the 1/32 of all nodes will pick the same module on average. In this case, the proportion of devil nodes (α_e) will determine the success probability of the account recovery at the consensus nodes. In this section, we will discuss two different distribution of the module on devil nodes and analyze their influence on the success probability.

A. CASE ONE

It is assumed that all the nodes that pick m_i as the residual base are devil nodes. This means that every remainder corresponding to m_i cannot be used to restore the account information. If we divide all nodes to 32 groups according to the selected module, when over 15 groups are devil groups, the account information cannot be restored correctly. So the maximum tolerable proportion of devil nodes, in this case, is 15/32 = 46.875%
From Figure 8, we can see that success probability for $Ne \leqslant 10$ is close to 1 and that for $Ne > 10$ decreases dramatically. This means that if the storage scheme works efficiently, the devil groups should be smaller than 10 and the proportion of devil nodes is $\alpha_e \leq 31.25\%$.

B. CASE TWO

If devil nodes pick a module from the set uniformly, the proportion of devil nodes among all nodes that pick m_i is also α_e. This means that the probability of CN gets a correct remainder from the nodes that pick m_i as the residual base is $1 - \alpha_e$. If we divide all nodes into 32 groups according to the selected module, the probability of getting at least i correct remainders from 32 groups could be calculated as:

$$p' = \sum_{i}^{32} C_{32}^{i} (1 - \alpha_e)^i \alpha_e^{(32-i)} \qquad (10)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access

From Figure 9, we can see that when $N_e \leqslant 10$, the success probability is close to 1 and it takes a short time to complete error correction. This means that if α_e is small enough so that CN can receive at least 22 correct remainders from the 32 groups, the new storage scheme will be very efficient. Figure 10 shows the probability of getting at least 22 correct remainders in CN, from which we see that when $\alpha_e \leqslant 10\%$, CN has a high probability to receive at least 22 correct remainders. In this case, the success probability is close to 1.

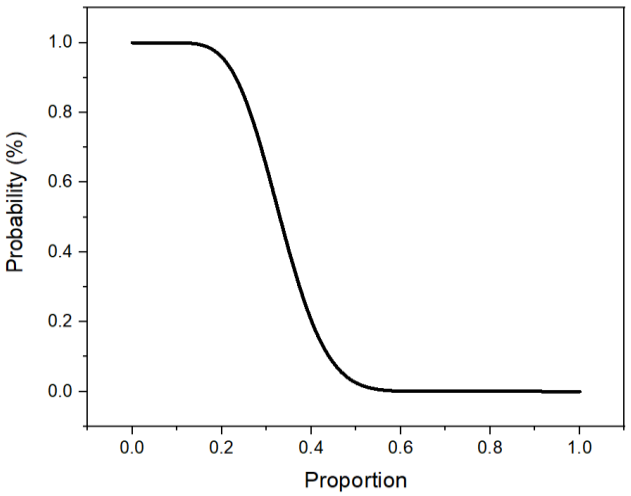


FIGURE10. The probability determined by α_e

VII. CONCLUSION

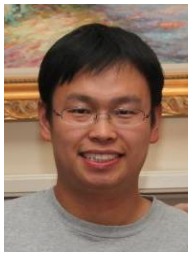
From the perspective of data compression, this paper proposes a new storage mechanism for blockchain based on RRNS. The new storage mechanism reduces the storage volume of each node dramatically and ensures that the system with certain fault tolerance without breaking the decentralization property of the blockchain system. A simplified demo system is given to show the basic idea of the new storage scheme, and the fault tolerance for general cases are also discussed.

REFERENCES

[1]. Crosby, Michael, et al. "Blockchain technology: Beyond bitcoin." Applied Innovation 2.6-10 (2016): 71.
[2]. Zyskind, Guy, and Oz Nathan, "Decentralizing Privacy: Using blockchain to protect personal data." 2015 IEEE Security and Privacy Workshops. IEEE, 2015.
[3]. Iansiti, Marco, and Karim R. Lakhani, "The truth about blockchain." Harvard Business Review 95.1 (2017): 118-127.
[4]. Swan, Melanie. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.
[5]. Tang, Huanrong, Ning Tong, and Jianquan Ouyang. "Medical Images Sharing System Based on Blockchain and Smart Contract of Credit Scores." 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN). IEEE, 2018.
[6]. Duan, Bin, Ying Zhong, and Dayu Liu. "Education application of blockchain technology: Learning outcome and meta-diploma." 2017 IEEE 23rd International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2017.
[7]. Yu, Wanjun, and Shiyuan Huang. "Traceability of Food Safety Based on BlockChain and RFID Technology." 2018 11th International Symposium on Computational Intelligence and Design (ISCID). Vol. 1. IEEE, 2019.
[8]. Blockchain Monitoring Website [Online], available: https://blockchain.info/, December 8, 2018.
[9]. Dorri, Ali, Salil S. Kanhere, and Raja Jurdak. "Towards an optimized blockchain for IoT." Proceedings of the second international conference on Internet-of-Things design and implementation. ACM, 2017.
[10]. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [Online], available: https://bitcoin.org/bitcoin.pdf, 2009.
[11]. Luu, Loi, et al. "A secure sharding protocol for open blockchains." Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016.
[12]. Zamani, Mahdi, Mahnush Movahedi, and Mariana Raykova. "RapidChain: Scaling blockchain via full sharding." Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2018.
[13]. Zheng, Qiuhong, et al. "An Innovative IPFS-Based Storage Model for Blockchain." 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI). IEEE, 2018.
[14]. Xu, Brent, et al. "EOS: An Architectural, Performance, and Economic Analysis."
[15]. Xu, Yibin. "Section-Blockchain: A Storage Reduced Blockchain Protocol, the Foundation of an Autotrophic Decentralized Storage Architecture." 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE, 2018.
[16]. Al-Bassam, Mustafa, et al. "Chainspace: A sharded smart contracts platform." arXiv preprint arXiv:1708.03778 (2017).
[17]. Iansiti, Marco, and Karim R. Lakhani. "The truth about blockchain." Harvard Business Review 95.1 (2017): 118-127.
[18]. Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151 (2014): 1-32.
[19]. Lei, Kai, et al. "Reputation-Based Byzantine Fault-Tolerance for Consortium Blockchain." 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2018.
[20]. Awerbuch, Baruch, et al. "An on-demand secure routing protocol resilient to byzantine failures." Proceedings of the 1st ACM workshop on Wireless security. ACM, 2002.
[21]. Doudou, Assia, and André Schiper. "Muteness detectors for consensus with Byzantine processes." in Proceedings of the 17th ACM Symposium on Principle of Distributed Computing,(Puerto. 1997.
[22]. Bastiaan, Martijn. "Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin." Available at http://referaat. cs. utwente.nl/conference/22/paper/7473/preventingthe-51-attack-astochastic-analysis-of-two-phase-proof-of-work-in-bitcoin. pdf. 2015.
[23]. Omondi, Amos R., and Benjamin Premkumar. Residue number systems: theory and implementation. Vol. 2. World Scientific, 2007.
[24]. Garner, Harvey L. "The residue number system." Papers presented at the March 3-5, 1959, western joint computer conference. ACM, 1959.
[25]. Barsi, Ferruccio, and Piero Maestrini. "Error correcting properties of redundant residue number systems." IEEE Transactions on Computers 100.3 (1973): 307-315.
[26]. Yau, SS-S., and Yu-Cheng Liu. "Error correction in redundant residue number systems." IEEE Transactions on Computers 100.1 (1973): 5-11.
[27]. Xiao Han-shen, Hu Jian-hao, and Ma ShangTor, "Low-complexity Error Correction Algorithms for Redundant Residue Number Systems." Vol.37. Journal of Electronics & Information Technology,2015.
[28]. Goh, Vik Tor, and Mohammad Umar Siddiqi. "Multiple error detection and correction based on redundant residue number systems." IEEE Transactions on Communications 56.3 (2008): 325-330.

ZHAOHUI GUO was born in Shanxi Province, China, 1996. He is currently pursuing the master's degree in School of microelectronics, Tianjin University. His research interest includes microwave and Blockchain Technology.
Email: 2017232048@tju.edu.cn

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2930125, IEEE Access



ZHEN GAO (Corresponding Author), received the BS, MS and Ph.D. degree in Electrical and Information Engineering from Tianjin University, China, in 2005, 2007 and 2011, respectively. During 2008.10 ~ 2010.11, he was a visiting scholar in GeorgiaTech. During 2011.7 ~ 2014.12, he was a Postdoc researcher in the Wireless and Mobile Communication Research Center in Tsinghua University, China. Since 2014.12, he is an Associate Professor in Tianjin University. His focus now is fault-tolerant signal processing, software defined radio, and blockchain.
Email: zgao@tju.edu.cn

HAOJUAN MEI was born in Henan Province, China, 1995. She is currently pursuing the master's degree in School of microelectronics, Tianjin University. Her research interest includes the integrated circuit and Blockchain Technology.
Email: may_1316708321@tju.edu.cn

MING ZHAO, Professor with the Research Institute of Information Technology of Tsinghua University. His current research interests include key technologies of the blockchain system, wireless and personal communications and software radios.
Email: zhaoming@tsinghua.edu.cn

JINSHENG YANG was born in Hebei Province, China in 1965. He is an Associate Professor with the School of microelectronics in Tianjin University. His primary research interests include Radio Wave Transmission and Blockchain Technology.
Email: jsyang@tju.edu.cn