Hindawi
Security and Communication Networks
Volume 2020, Article ID 8876317, 12 pages
https://doi.org/10.1155/2020/8876317

*Research Article*

# A Study on the Optimization of Blockchain Hashing Algorithm Based on PRCA

**Jinhua Fu**,[1,2] **Sihai Qiao**,[2] **Yongzhong Huang**,[1,3] **Xueming Si**,[1,4] **Bin Li**,[1,5] **and Chao Yuan**[1]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*
[2]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450001, China*
[3]*School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China*
[4]*School of Computer Science, Fudan University, Shanghai 201203, China*
[5]*Zhengzhou University, Zhengzhou 450001, China*

Correspondence should be addressed to Jinhua Fu; jinhua@zzuli.edu.cn

Received 12 March 2020; Revised 14 April 2020; Accepted 23 May 2020; Published 14 September 2020

Academic Editor: Yuan Yuan



Blockchain is widely used in encrypted currency, Internet of Things (IoT), supply chain finance, data sharing, and other fields. However, there are security problems in blockchains to varying degrees. As an important component of blockchain, hash function has relatively low computational efficiency. Therefore, this paper proposes a new scheme to optimize the blockchain hashing algorithm based on PRCA (Proactive Reconfigurable Computing Architecture). In order to improve the calculation performance of hashing function, the paper realizes the pipeline hashing algorithm and optimizes the efficiency of communication facilities and network data transmission by combining blockchains with mimic computers. Meanwhile, to ensure the security of data information, this paper chooses lightweight hashing algorithm to do multiple hashing and transforms the hash algorithm structure as well. The experimental results show that the scheme given in the paper not only improves the security of blockchains but also improves the efficiency of data processing.

## 1. Introduction

Blockchain is a kind of distributed general ledger technology, originated from the literature [1]. Initially, it was mainly used in the field of cryptocurrency, the most representative of which were Bitcoin and Litecoin [2], Monroe [3], and Zcash [4]. Amid its rapid development, blockchain technology can effectively guarantee the authenticity, security, and reliability of data. It also has been widely used in medical data [5], personal data protection [6], and data allocation scheme [7]. As the basic unit of blockchain, block consists of partition header including original data and block body including transaction data. Among them, block data are used to connect the previous block and index the data from the hash value of range block. Each blockchain transaction is conducted by using hash function interaction. It guarantees the security of blockchain.

However, with the continuous development of blockchain, its security issues become increasingly prominent. The lightweight hash function SHA1 in the blockchain is no longer regarded as an attacker that can withstand sufficient funds and computing resources. SHA256 can replace SHA1 for information exchange with good anticollision ability, while it cannot be changed at will. To avoid chain breakage, it is necessary to modify the hash values of all blocks behind the block at the same time. As a result, a large computational complexity is needed and the security of the blockchain is not guaranteed.

In the process of executing operations, the PRCA (Proactive Reconfigurable Computing Architecture) generates the optimal computation structure set by self-perception and dynamic selection. All the software and hardware variants are dynamically variable. Therefore, in the process of application processing, they can select optimal solutions according to the independent variables in the program to get the variable optimal solution sets with equivalent function and different computing efficiency [8]. Combining with blockchain, it can improve the performance

of the algorithm, improve the transmission efficiency, and enhance the security of hash algorithm.

This paper proposes an optimization scheme of blockchain hashing algorithm based on PRCA. Aiming at the blockchain hash algorithm structure, a reconfigurable hash algorithm with high performance is implemented in a full pipeline way. At the same time, 10,000 Mbp communication is realized by mimic computer to reduce data transmission delay, and data is read from memory by DMA, which improves transmission efficiency. In each transaction, the hash algorithm is negotiated and the mimic computer is reconstructed, which aims to transform the hash algorithm structure through using lightweight hash algorithm for many times. This scheme not only improves the efficiency of processing data for blockchain but also increases its security.

## 2. Proactive Reconfigurable Computing Architecture

*2.1. Definition of Proactive Reconfigurable Computation.* PRCA is an operation mechanism based on multidimensional reconstructed functional structure and dynamic multibody. When proactive reconfigurable computation is processing data, execution structures, such as computing, storage, and interconnection, are changing dynamically with the efficiency of transaction processing, instead of improving the algorithm to improve the operation performance without changing the basic hardware. There are many functional equivalents in PRCA, but they are accomplished by combining different hardware structures with this algorithm. The purpose is to achieve the high performance of computing, that is, how to automatically perceive variables to generate the optimal computing set and autonomously reconstruct the computing in the processing algorithm [9].

PRCA has variable infrastructure and algorithm, which makes it possible to obtain optimal solutions to different problems. It pursues different services and comprehensive high performance under different loads or other conditions, builds the most appropriate processing components, and forms the most appropriate architecture. Proactive reconfigurable computation combines the advantages of general computing and special computing to achieve the goal of solving problems efficiently. In terms of the general computing structure, it is characterized by its determined structures and variable algorithm and may calculate any computable problems with high efficiency. Its principle is shown in Figure 1.

*2.2. Proactive Reconfigurable Computer.* Proactive reconfigurable computer is a new type of computer developed according to the principle of mimetic computing to achieve the high performance of computing. The computational structure can be regarded as a high-order function. In the analysis of the calculation, the computational structure will generate the most efficient set of settlement structures by selecting the perceptual independent variables. The essence of proactive reconfigurable computer is the functionalization of computational structure. Its high performance and efficiency are very suitable for the processing and analysis of big data nowadays. Compared with the traditional computer, the energy efficiency of proactive reconfigurable computer has been improved more than 10 times. The structure of the principle prototype of the proactive reconfigurable computer is shown in Figure 2.

The purpose of proactive reconfigurable computer is to deal with intensive computing. It consists of an ATOM general microprocessor, four high-order reconfigurable large-scale reconfigurable FPGAs, and DDR3 memory, which connects LVDS bus FULL-MESH through floor GTX, and is controlled by the control unit BMC and synchronized by clock synchronization unit. The prototype supports multiple interfaces and storage media and reconstructs FPGA processing core, I/O interface, and on-chip interconnection network according to the application requirements, so as to achieve the purpose of high-efficiency computing [10].

Proactive reconfigurable computers use dynamic randomicity to build an asymmetric defense system, which expands the attack surface to weaken intrinsic attacks of feature sniffing and state transition [11]. Based on such a characteristic, 10,000 Mbp communication is realized by using FPGA to reduce data transmission delay, build a simulated hash structure, and improve the speed of hash value calculation of blockchain data. A Merkle tree is formed to match the algorithm, which makes it difficult for attackers to distinguish the complexity of the target and improves the security performance of the system [12]. The protection function of computer hardware is used to expand the area of attack, increase the difficulty of blockchain attack, and improve the antiattack ability.

## 3. Optimization of Blockchain Hash Algorithms Based on PRCA

*3.1. System Framework and Block Structure.* The proactive reconfigurable computer is configured as a node in the blockchain network. Users and proactive reconfigurable computers establish a connection. The proactive reconfigurable computer catches the data in the DDR memory and realizes the direct connection high-speed transmission from network to the memory data by the asynchronous FIFO, reducing the intermediate transmission level. In blockchain, a high-performance hash algorithm is implemented by means of pipelines and the key segment calculation data hash is extracted from memory [13]. After calculating the hash value, the result is encapsulated and transferred to the storage server to complete the storage of the blockchain. The specific system framework is shown in Figure 3.

The block stores all the information about transactions, including the generation time of transaction, the record index number of transaction, the hash value of transactions, bitcoin's expenditure address and its amount of expenditures, and other types of transaction. A Merkle value will be generated in the transaction. The hash node value in the transaction determines that each address cannot be repeatedly traded and forged. To further improve the security of transactions, a proactive reconfigurable hash is added to the blockchain, which is composed of various types and

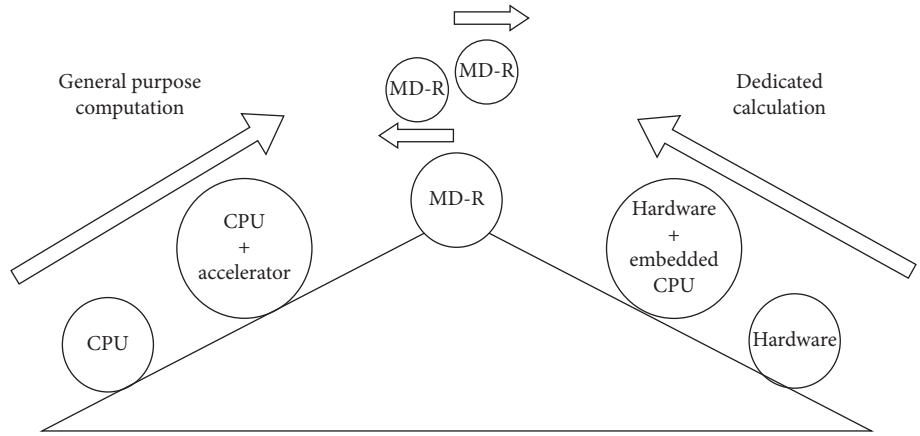


Figure 1: The basic concept of PRCA.

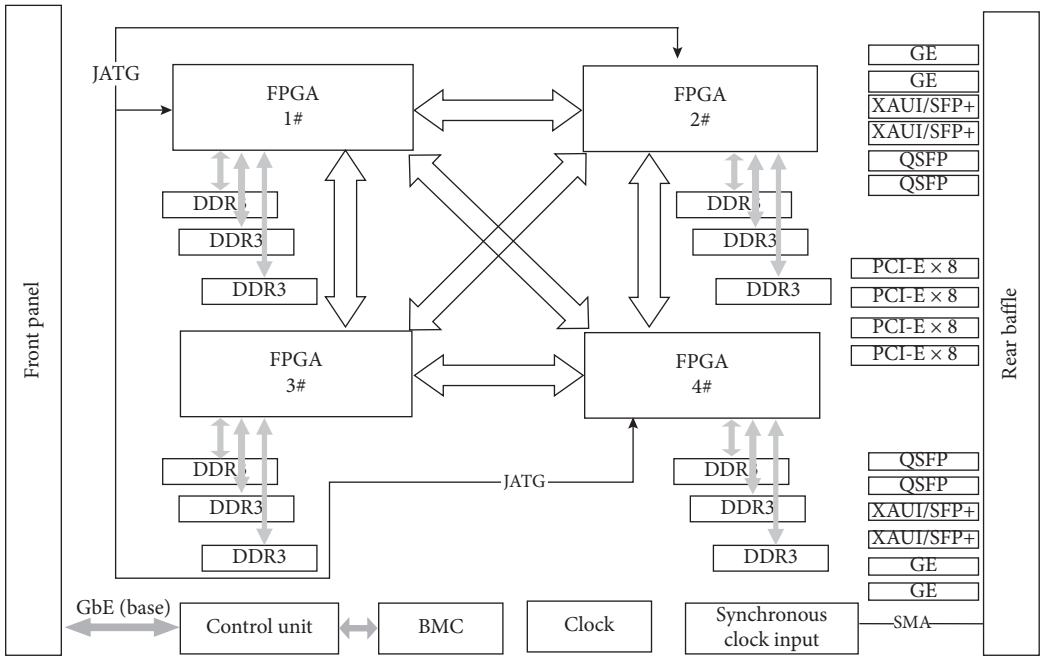


Figure 2: Principle prototype structure of the proactive reconfigurable computer.

structures of hash algorithms and can be used separately or in series. The concrete structure model is shown in Figure 4.

Unit nodes in blockchains monitor network traffic to calculate transaction volume [14]. Before the transaction is generated, the hash algorithm selection step will be added, and then the appropriate hash function will be selected from the hash list. The unit node uses the selected hash function to compete to find the hash value. Once the hash value is found, the block will be propagated to another node in the blockchain for verification.

In the interaction, the sensor layer on the spot collects data. The sensor transmits data to unit nodes and requests the transaction to store the data. If unit nodes successfully complete the transaction mining, the blockchain network will update the block. After that, the blockchain network returns the field layer data to the control layer. Then block mining will be started. After the block mining is finished, the blockchain network receives the node of transaction and broadcasts the block and validation request to other nodes. Other nodes using hash algorithm confirmed from the block header for verification. After the successful verification, they will update the block and store nodes and blocks. If the contents of transactions are transferring data or commands, the requested node will transfer the data or command to the other layers. The specific block mining and updating are shown in Figure 5.

At the same time, the random number generator randomly chooses the new hash algorithm at intervals, and the two sides negotiate again and update for new hash algorithm to improve security.

*3.2. Hash Algorithm Optimization.* Hash function is an important part of many cryptographic algorithms. An important component of blockchain technology is to apply

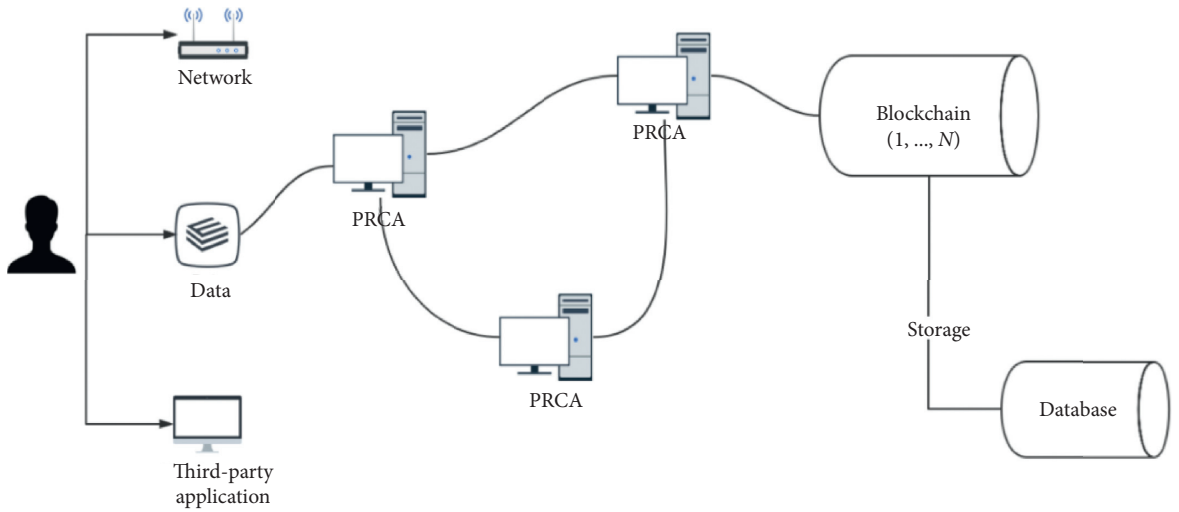


Figure 3: Blockchain system architecture based on PRCA.

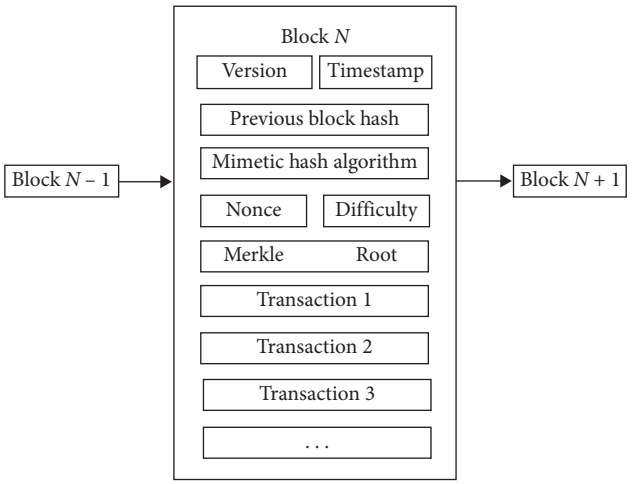


Figure 4: Proactive reconfigurable hash structure in blockchain.

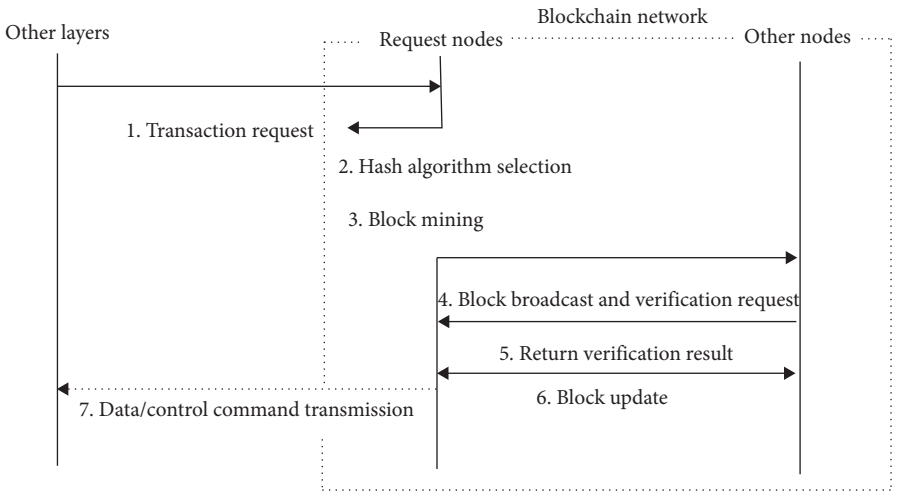


Figure 5: Mining and updating of block.

hash function for many operations. Hashing is a method of applying hash function to data that computes a relatively unique output for almost any size of input. It allows individuals to independently obtain input data and hash data and produce the same results, proving that the data has not changed. Take SHA256 as an example to illustrate the optimization and implementation of hash algorithm on proactive reconfigurable computers.

The throughput of the algorithm solves the computational performance of the algorithm. The specific implementation formula is as follows:

$$T = \frac{B \times f_{\max} \times N}{d}. \tag{1}$$

In equation (1), $T$ is the throughput, $B$ denotes the data block size, $f$ is the maximum clock frequency, $N$ is the pipeline series, and $d$ denotes the calculation delay. The number of pipeline series is proportional to frequency and throughput. In order to improve the throughput of the algorithm, we can use prediction and CSA strategies to reduce the delay of critical paths and use full-pipeline SHA1 and SHA256 algorithms.

The following is an introduction to the optimization of SHA256, which can be extended to SHA1.

*3.2.1. SHA256.* For messages with a length no more than $2^{64}$ bits, the hash algorithm SHA256 will produce a hash value with a length of 256 bits, which is called a message digest. The digest is a 32-byte array that can be represented by a hexadecimal string of length 64. The processing of the SHA256 algorithm is divided into five steps:

(i) Add great many 0 bits to the input data until 448 bits. Then add 64-bit length to the input data until 512 bits.

(ii) Divide the spliced 512-bit data into 16 groups: $M_0$–$M_{15}$.

(iii) Initialize the vectors $K_0$–$K_{63}$ and $h_0$–$h_7$, and let the initial values of $A$, $B$, $C$, $D$, $E$, $F$, $G$, and $H$ be $h_0$–$h_7$.

(iv) Set the variable $t$ to loop from 0 to 63 and then update as follows: $B_{t+1} = A_t, C_{t+1} = B_t, D_{t+1} = C_t, F_{t+1} = E_t, G_{t+1} = F_t, H_{t+1} = G_t$,

$$\begin{aligned} A_{t+1} &= H_t + \sum_1 (E_t) + \mathrm{Ch}(E_t, F_t, G_t) + K_t + W_t \\ &\quad + \sum_0 (A_t) + \mathrm{Maj}(A_t, B_t, C_t), \\ E_{t+1} &= H_t + \sum_1 (E_t) + \mathrm{Ch}(E_t, F_t, G_t) + K_t + W_t + D_t. \end{aligned} \tag{2}$$

(v) Let $h_0 = h_0 + A_{63}, h_1 = h_1 + B_{63}, h_2 = h_2 + C_{63}, h_3 = h_3 + D_{63}, h_4 = h_4 + E_{63}$, $h_5 = h_5 + F_{63}, h_6 = h_6 + G_{63}, h_7 = h_7 + H_{63}$. Output $h_0 - h_7$.

In the above algorithm, $\sum_1(E_t)$, $\sum_0(A_t)$, $\mathrm{Maj}(A_t, B_t, C_t)$, and $\mathrm{Ch}(E_t, F_t, G_t)$ are logical functions, and $W_t$ is updated according to

$$W_t = \begin{cases} M_t, & 0 \le t \le 15, \\ \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}, & 16 \le t \le 63. \end{cases} \tag{3}$$

From the processing of the SHA256 algorithm, it can be seen that the key is to update the values of A and $E$, which requires multiple addition operations and 64 cycles of iteration. Therefore, the optimization of these two operands will play an important role in reducing the time consumption of the algorithm.

*3.2.2. Critical Path Segmentation Optimization.* The time consumption of the SHA256 operation is mainly in the iteration part of Step 4, and the most time-consuming part is the calculation of $A$ and $E$ values. Therefore, adopting the method of critical path segmentation and combining with the parallel characteristics of FPGA computing resources can effectively shorten the time consumption.

$H_t$, $K_t$, and $W_t$ in the critical path do not need additional logical operations or do not depend on other operands of the current round. Therefore, the critical path of the algorithm is divided into the following formulas:

$$S_t = H_t + K_t + W_t, \tag{4}$$

$$\begin{aligned} A_{t+1} &= \sum_1 (E) + \mathrm{Ch}(E_t, F_t, G_t) + S_t \\ &\quad + \sum_0 (A) + \mathrm{Maj}(A_t, B_t, C_t), \end{aligned} \tag{5}$$

$$E_{t+1} = \sum_1 (E) + \mathrm{Ch}(E_t, F_t, G_t) + S_t + D_t. \tag{6}$$

In this way, $A$ and $E$ values will be updated and shortened from the original $6t_{\mathrm{ADD}}$ and $5t_{\mathrm{ADD}}$ to $4t_{\mathrm{ADD}}$ and $3t_{\mathrm{ADD}}$, where $t_{\mathrm{ADD}}$ denotes the time consumption of addition operations.

*3.2.3. Minimum Addition Optimization.* FPGA is suitable for bit operation. Carry-Save Adders (CSA) strategy can reduce addition operation, minimize critical path length, and ensure pipeline throughput. For $n$-bit binary numbers a, $b$, and $c$, the CAS operations are as follows:

$$\begin{aligned} S(a, b, c) &= a \wedge b \wedge c, \\ \mathrm{Ca}(a, b, c) &= [(ab) \mid (bc) \mid (ac)] \ll, \\ \mathrm{CSA}(a, b, c) &= S(a, b, c) + \mathrm{Ca}(a, b, c) = a + b + c. \end{aligned} \tag{7}$$

By dividing the critical paths, it takes $2t_{\mathrm{ADD}}$, $4t_{\mathrm{ADD}}$, and $3t_{\mathrm{ADD}}$ to calculate $S_t$, $A_{t+1}$, and $E_{t+1}$, respectively. Since the addition operation consumes a lot of time on the FPGA, the CSA method should be used to increase bit operation and reduce the addition operation, in order that the total time consumption can be reduced. By using the critical path

partitioning method and CSA strategy, formulas (4)~(6) are replaced by CSA operation in the following formulas:

$$S_t = \mathrm{CSA}\left(H_t, K_t, W_t\right), \tag{8}$$

$$A_{t+1} = \mathrm{CSA}\left(\mathrm{CSA}\left(\sum_1(E), \mathrm{Ch}\left(E_t, F_t, G_t\right), S_t\sum_0(A), \mathrm{Maj}\left(A_t, B_t, C_t\right)\right)\right), \tag{9}$$

$$E_{t+1} = \mathrm{CSA}\left(\sum_1(E), \mathrm{Ch}\left(E_t, F_t, G_t\right), S_t\right) + D_t. \tag{10}$$

The critical path segmentation method and the CSA strategy reduce the operation of $A_{t+1}$ and $E_{t+1}$ to only $2t_{\mathrm{ADD}}$, thus improving the efficiency of the algorithm.

*3.2.4. Pipeline Optimization.* After the optimization of critical path partition, the time consumption of the longest path is reduced. For serial computing, the total time consumption does not decrease. Therefore, it is necessary to use the parallel characteristics of FPGA and pipeline method for optimization, so as to truly reduce the total time consumption of computing.

According to the characteristics of the SHA256 algorithm and the optimization of critical path, the core processing of the algorithm is divided into three modules: $W$ module, split $S$ module, and update module $A-H$. The pipelining technology reduces time consumption by increasing resource utilization. Therefore, each module needs 64 computing units and a total of 192 computing units.

While data are being calculated, in the first clock cycle, the first data are input to the $W_0$ computing unit for processing in the first clock cycle. In the second clock cycle, the output of $W_0$ is taken as the input of $S0$, and $W_1$ is calculated. At the same time, the second data are input to $W_0$. In the third clock cycle, three computing units are processed in parallel, and so on. Until the 66th clock cycle, when all 192 units are running, the output of the first data is completed. When there is a large amount of data to be computed, one type of data is computed in a clock cycle, which reduces the time consumed by 64 iterations in the algorithm. Therefore, the throughput and resource utilization of the algorithm are greatly improved. The pipeline structure of the SHA256 algorithm is shown in Figure 6.

### 3.3. Communication and Network Optimization

*3.3.1. Communication Optimization.* For adapting to the calculation of blockchain hash, the concrete structure of proactive reconfigurable computer is shown in Figure 7, which mainly includes Hash_Core, I_10G, CTL_DDR3_0/1, State_U, Ctl_Core, and I_1G modules.

The functions of each module are as follows:

(i) *Hash_Core module*. The core processing module of hash computing is mainly responsible for hash calculation of blockchain data, which is implemented in full-pipeline mode and supports hash calculation of SHA1, SHA256, and so forth.

(ii) *I_10G module*. The data communication interface circuit based on 10,000 Mega mainly includes 10,000 Mega MAC interface, data buffer, and interface of module on the same chip. The module is mainly responsible for the input of data to be processed and the recovery of calculation results.

(iii) *CTL_DDR3_0 module*. The data communication interface circuit based on DDR3 mainly includes DDR3 interface, data buffer, and interface of on-chip module. This module is mainly responsible for data memory reading.

(iv) *CTL_DDR3_1 module*. The data communication interface circuit based on DDR3 mainly includes DDR3 interface, data buffer, and interface of on-chip module. This module is mainly responsible for data memory writing.

(v) *State_U module*. Acquire the on-chip state of each module, and then output it to Ctl_Core.

(vi) *Ctl_Core module*. The processor-based on-chip processing control core is mainly responsible for reporting the running state of the mimic computer and processing the control information.

(vii) *I_1G module*. Data communication interface based on Gigabit Ethernet interface is mainly used for communication of control information.

Block data are cached to CTL_DDR3_0 via I_10G network interface, hash values are read and calculated by Hash_Core, and results are cached into CTL_DDR3_1 and finally sent to the network by I_10G. The host computer controls the proactive reconfigurable computer in real time through I_1G Gigabit interface and Ctl_Core according to the information reported by State_U.

*3.3.2. 10G Network.* 10G network is implemented based on IP protocol, and the content of data transmission is controlled by external users. It uses FIFO interface to communicate with external devices [15]. In the process of transmitting control messages, if the receiver does not have an ARP response, the system will issue a timeout error because ARP does not respond; if there is a timeout transmission, the system will show the number of times of timeout transmission. If the transmission succeeds, the successful message will be returned; if the transmission fails, the error message which is retransmitted overtime will be returned. If there is a timeout and no information is received, the system will send out the wrong signal of communication channel, according to which the user will take appropriate action accordingly. The whole structure is shown in Figure 8.

In Figure 8, the sending port includes two FIFOs: the sending data FIFO (ip_snd_fifo) and the sending status FIFO (ip_snd_status_fifo). The sending data FIFO's depth is 65 bits and low 64 bits are data interface. The highest bit

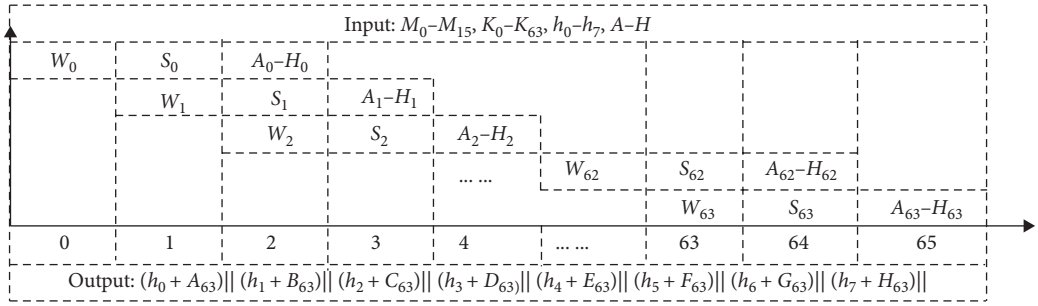


Figure 6: Pipeline structure of the SHA256 algorithm.

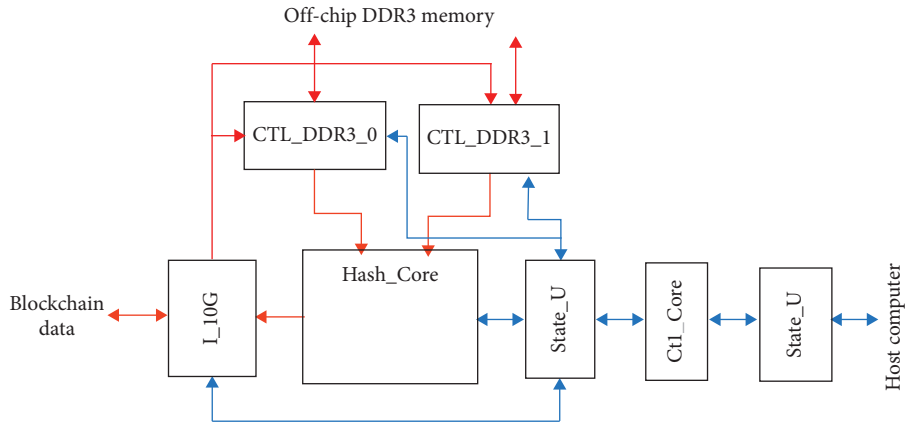


Figure 7: On-chip architecture of proactive reconfigurable computer.

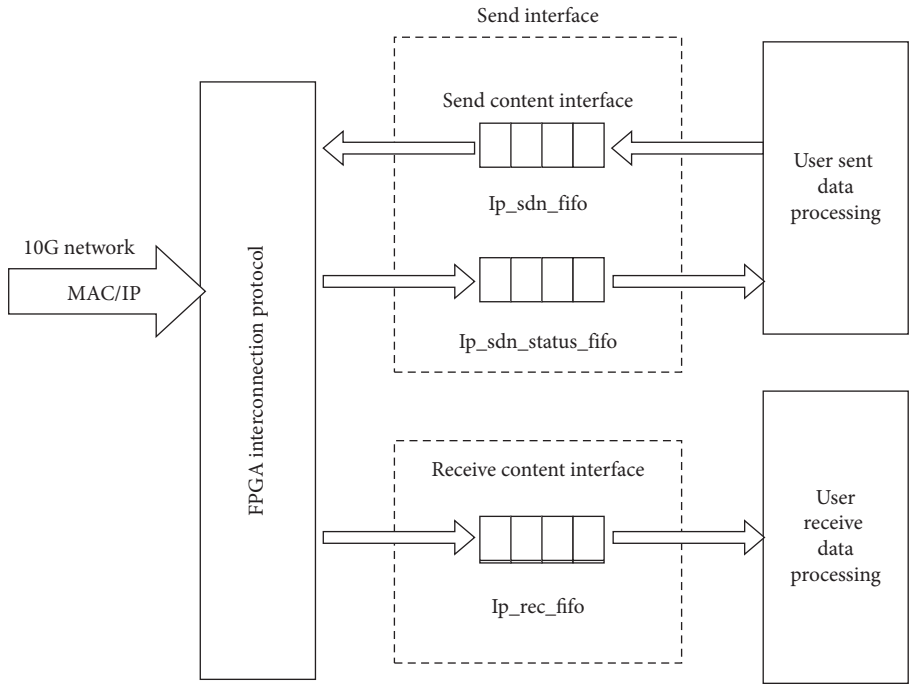


Figure 8: The whole structure of 10G network.

indicates whether the data transmission is the last one. If more than 1440 bytes of data are to be transmitted, multiple transfers are required. The sending status FIFO is used to identify whether there is an error in the data transmission. If there is an error like the timeout in the process of data transmission, all subsequent contents will be read out until the last one. Each data transmission corresponds to a state FIFO write. The receiving port has only one FIFO, that is, the

receiving data FIFO (ip_rec_fifo), which has a depth of 65 bits and low 64 bits as the data interface. The highest bit indicates whether the data transmission is the last frame of data, and the data received is identified by index number.

*3.3.3. Memory Management.* Read-write memory is implemented by four groups of FIFOs in burst mode. Every time before it reads and writes memory, it will calculate the memory address range according to the length of the data and store it in wrrdinfo_fifo. At the same time, the data will be cached in wfifo_fifo, and according to the information of wrrdinfo_fifo, the read-write arbitration module determines whether it is a reading operation or a writing one. If it is a writing operation, the data will be written to memory through the DDR write module. The process of reading memory data is similar to that of writing. The read information and data will be cached in out_rdinfo_fifo and rififo_fifo, respectively. The read-write structure of memory is shown in Figure 9, where the size of request information wrrdinfo_fifo and out_rdinfo_fifo is $16 * 64$ bits, and the size of reading and writing wfifo_fifo and rififo_fifo is $4096 * 64$ bits.

When the initialization of memory is completed, that is, phy_init_done is set to 1, the CTL_DDR3_0 and CTL_DDR3_1 modules are in the read-write state, and the read-write state jump will be completed according to the wrrdinfo_$q$[0] identifier bit, as shown in Figure 10. When it begins reading and writing memory, the address of memory will be counted according to the length of writing, and the reading and writing of the whole data will be completed. After the reading and writing operation is completed, it will jump to the idle state and wait for the next operation.

*3.4. Application of PRCA Blockchain.* Public and private keys in blockchains are a pair of keys obtained by a kind of algorithm. It will be encrypted with public key and decrypted with corresponding private key. After three times of SHA256 computation and one time of RIPEMD160 computation for the public key, a public key hash can be obtained, and the address can finally be obtained through base58 encoding [16]. Merkle tree is a kind of tree structure. In trading with blockchains, every transaction is hashed, and the final root is Merkle root [17]. Proof-of-work (PoW) is called mining in blockchains. CPU calculation uses the complexity of hash operation to determine PoW, and it will produce a value smaller than the specified target [18]. Block filter proposed in the blockchain is a fast search based on hash function, which can quickly determine whether a retrieved value exists in the searched set [19]. The application of hash algorithm in blockchain is shown in Figure 11.

In this paper, the communication equipment and network are optimized. In a relatively safe environment, a relatively simple and lightweight hash algorithm is chosen to replace the complex hash algorithm, so as to improve the running speed of the system and reduce the energy consumption of the system. Meanwhile, multiple hash algorithm is used to reduce the attack of length expansion and ensure the integrity and tamper-proofing of information, which reflects the security performance of blockchain.

## 4. Experimental Analysis

In this paper, proactive reconfigurable computer is used for experiments. The software platform is ISE software integrating design, simulation, integration, wiring, and generation. First, the comparison of CPU running speed and resource utilization is given by optimizing the hash algorithm deeply. Second, the collision resistance of proactive reconfigurable hashes is analyzed. Finally, the security of this scheme is analyzed from many aspects.

The configuration information of each computing unit used in the experiment is shown in Table 1.

*4.1. Performance Analysis.* On the proactive reconfigurable computer, the SHA256 and SHA1 algorithms are implemented, respectively. Their resource occupation, frequency, and throughput are shown in Table 2.

As seen from Table 2 and Figure 12, SHA256 and SHA1 implemented in a pipelined manner occupy less than 10% of the resources but with high throughput.

Next is the performance comparison of SHA256 and SHA1 between the proactive reconfigurable computer and CPU, as is shown in Table 3.

From Table 3, it can be seen that the proactive reconfigurable computer can realize the parallelism of multiple modules and can fully meet the application requirements of hash computing in blockchain. Taking Bitcoin three hash as an example, three SHA256 combinations are connected in series to form a cascade pipeline. The data can be directly input into the pipeline without waiting, and the results are output sequentially by the end, which is very efficient. Contrastively, CPU can only rely on multithreaded concurrency to improve computing performance, and its essence is still serial execution, which will not be competent for blockchain applications requiring large amounts of computing.

Meanwhile, the proactive reconfigurable computer is equipped with a 10-gigabit network, whose data transmission peak is about 10 Gbps, which can meet the communication requirements of blockchain high-frequency transactions. As each clock cycle can transmit 8 bytes of data, the clock frequency is 156.25 MHz; while the FIFO interface and frequency of DDR are 8 bytes and 156.25 MHz, the data transmitted by 10G network can be synchronized through FIFO cache and written into memory with 64 bytes and 300 MHz. Two memory modules are configured: one is responsible for writing operation of 10G network and reading operation of hash module, and the other is responsible for writing operation of hash module and reading operation of 10G network. The two memory modules work independently, which improves the efficiency of data transmission.

*4.2. Antiattack Analysis.* Hash operation is irreversible and gets different values for different contents. Any change of input information will lead to significant changes in hash

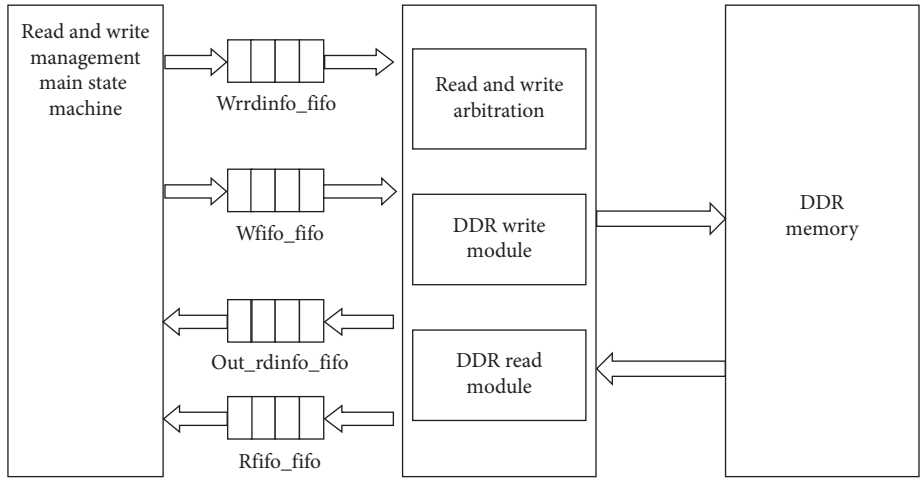


Figure 9: The read-write structure of memory.

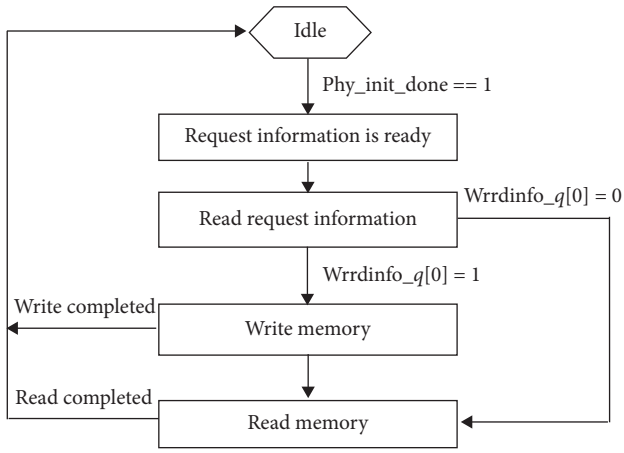


Figure 10: Memory state management mechanism.

results. Moreover, hash operation is also anticollision; that is, two pieces of information with the same hash result cannot be found, which can effectively prevent differential attack [20].

Assuming that the output value of hash function is uniformly distributed and the message digest has $m$ bits, the hash value has $n = 2^m$ possible outputs. For any $k$ ($k \le n$) random input, the probability of at least one collision is

$$\begin{aligned} p(n,k) &= 1 - \frac{n!}{(n-k)! \times n^k} \\ &= 1 - \left[\left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{k-1}{n}\right)\right] \\ &= 1 - \prod_{i=1}^{k-1}\left(1 - \frac{i}{n}\right) \\ &\approx 1 - \coprod \\ &= 1 - e^{-(k(k-1)/2n)}. \end{aligned} \tag{11}$$

If $p(n,k) > 0.5$, that is, $1 - e^{-(k(k-1)/2n)} = 1/2$, then $\ln 2 \approx (k^2/2n)$; this means $k \approx \sqrt{n}$.

According to the above calculation, if the hash function has an output digest of $m$ bits, then only $k = 2^{m/2}$ attempts will result in a collision with a probability of at least 50%. SHA1 and SHA256 are operations of $2^{160}$ and $2^{256}$ orders of magnitude, respectively. Table 4 gives the threshold of hash function conflict.

Bitcoin obtains hash data through the SHA256 algorithm and runs two iterations in block trading to mitigate the length expansion attack. PRCA blockchain system can be described by a triple tuple as $\Omega = \{\text{Block}, \text{Hash}, \text{Num}\}$, where "Block" represents block data, "Hash" represents hash algorithm, and "Num" represents iteration times. The multiple phases of $\Omega$ have many different hash combination schemes and can be represented by $\Omega = \{\text{Block}(t), \text{Hash}(t), \text{Num}(t)\}$ at time $t$, which is dynamic, diverse, and random.

The hash algorithm of PRCA blockchain system $\Omega$ is dynamically reconfigurable. After negotiation, the hash algorithm can be reconstructed dynamically and partially to complete the switching of different algorithms. In addition, "Block" is changing constantly, and the content of each transaction is unpredictable and completely different. Finally, "Num" can be negotiated by both sides to improve its security by increasing the number of iterations without significantly increasing the amount of computation. Obviously, the blockchain based on PRCA not only improves the complexity of internal hash operation but also combines the hash to increase the length of output, which greatly hinders the attackers from extending the blockchain and reduces the probability of collision.

*4.3. Security Performance Analysis.* Encryption of information is the key link of blockchain, which mainly includes hash function and asymmetric encryption algorithms [21]. Asymmetric encryption uses private key to prove the ownership of the node and is implemented by digital signature. Hash algorithm is used to transform the input of any

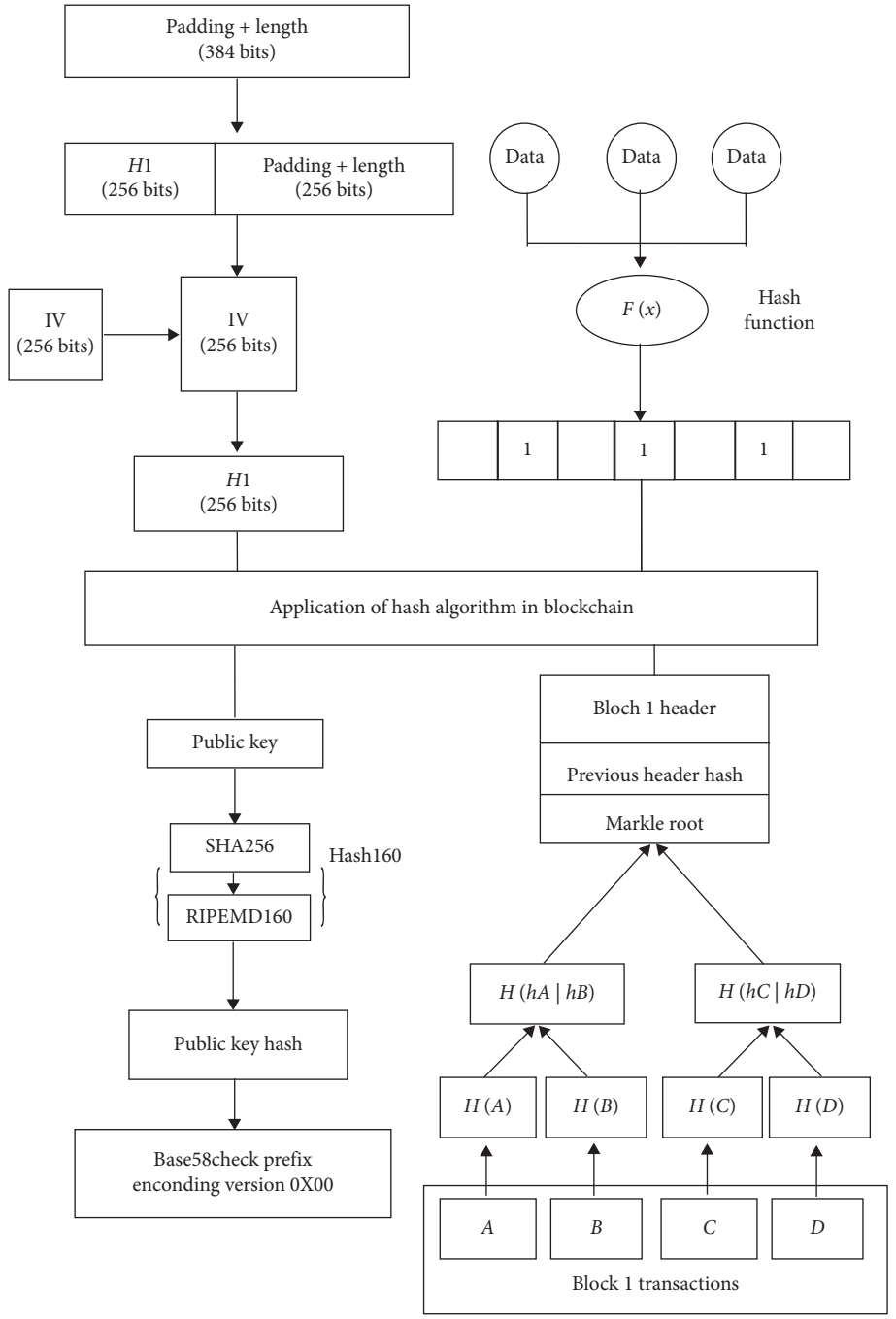


Figure 11: The application of hash algorithm in blockchain.

Table 1: The configuration information of each computing unit.

| Calculation component | Configuration information |
|---|---|
| CPU server | 4-core CPU; model: i5-7500; main frequency: 3.40 GHz; memory: 24 GB |
| PRCA | 4 FPGA cards; on-chip resources slices: 85920; memory: 24 GB |
| 10G switch | 24 1/10G SFP + ports; 4 10/100/1000 m electrical interface |

Table 2: The actual operation of SHA256 and SHA1.

| | Regs (687, 360) | LUTs (343, 680) | Slices (85, 920) | Frequency (MHz) | Throughput (Mbps) |
|---|---|---|---|---|---|
| SHA1 | 24,703 | 18,899 | 6106 | 243.8 | 124825.6 |
| SHA256 | 27669 | 25648 | 7745 | 172.0 | 88064 |

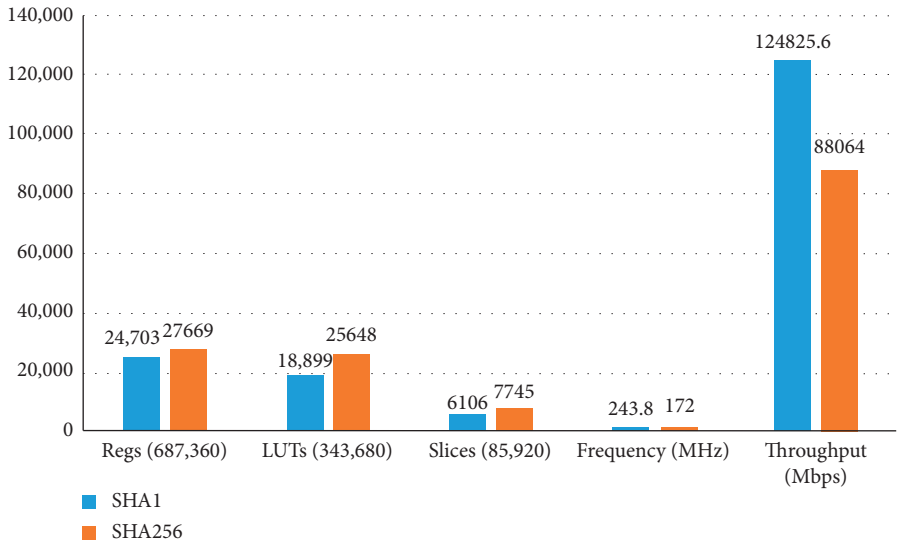


Figure 12: The actual operation of SHA256 and SHA1.

Table 3: The performance comparison of SHA256 and SHA1 between the proactive reconfigurable computer and CPU.

| Calculation component | Number of parallel modules | | Frequency (MHz) | Running speed (m) |
|---|---|---|---|---|
| PRCA | SHA1 | 40 | 200 | 8000 |
| | SHA256 | 24 | 150 | 4800 |
| CPU | SHA1 | — | — | 270.6 |
| | SHA256 | — | — | 119.3 |

Table 4: The threshold of hash function conflict.

| Hash function | Function collision threshold |
|---|---|
| SHA1 | $2^{80} \approx 1.2 \times 10^{24}$ |
| SHA256 | $2^{128} \approx 3.4 \times 10^{38}$ |

length into an output of fixed length consisting of letters and numbers, which is irreversible and tamper-proofing.

From the perspective of information security, the main advantages of this scheme are as follows:

(i) Multiple hash algorithms are jointly used to ensure the integrity and nontampering of information

(ii) There is a pseudorandom dynamic selection and the hash algorithm is updated to increase the difficulty of attack in time dimension

(iii) By using the hardware implementation of proactive reconfigurable computer, the attack surface is expanded and the attack threshold is raised

Obviously, the blockchain based on PRCA enhances the confidentiality, authenticity, and integrity of data and enhances the overall security of blockchain transactions with its reliability, security, and tamper-resistance.

## 5. Conclusions

In order to improve the efficiency and security of blockchain hash algorithm, a scheme of blockchain hash algorithm optimization based on PRCA is proposed in this paper. This scheme combines blockchain with proactive reconfigurable computer to improve the performance of blockchain hash function. In terms of security performance, several lightweight hash algorithms are used to exchange information to ensure the integrity and tamper-proofing of information. The proactive reconfigurable computer hardware is used to expand the attack surface, improve the attack threshold, and ensure the security of blockchain.

Blockchain security is the most important part of the system, which includes data, intelligent contract, privacy protection, and application risk. Meanwhile, the data of blockchain is unique. Under the condition of its own security, data writing cannot be changed. Based on the security problem of data immutability, the data structure,

cryptography technology, and communication network at the bottom of blockchain are improved to promote the healthy development of blockchain application.

## Data Availability

The data used support the findings of the study are available from the corresponding authors upon request.

## Additional Points

*Highlights*. In this paper, proactive reconfigurable computer is used for experiments. The software platform is ISE software integrating design, simulation, integration, wiring, and generation. First, the comparison of CPU running speed and resource utilization is given by optimizing the hash algorithm deeply. Second, the collision resistance of proactive reconfigurable hashes is analyzed. Finally, the security of this scheme is analyzed from many aspects.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This research work was supported by the Innovative Research Groups of the National Natural Science Foundation of China (61521003), Intergovernmental Special Programme of National Key Research and Development Programme (2016YFE0100300 and 2016YFE0100600), National Scientific Fund Programme for Young Scholar (61672470), and Science and Technology Project of Henan Province (182102210617).

## References

[1] Q. Lu and X. Xu, "Adaptable blockchain-based systems: a case study for product traceability," *IEEE Software*, vol. 34, no. 6, pp. 21–27, 2017.

[2] M. Padmavathi and R. M. Suresh, "Secure P2P intelligent network transaction using Litecoin," *Mobile Networks and Applications*, vol. 24, no. 2, pp. 318–326, 2018.

[3] I. Bentov and R. Kumaresan, "How to use Bitcoin to design fair protocols," *Lecture Notes in Computer Science*, vol. 8617, pp. 421–439, 2017.

[4] P. Katsiampa, "Volatility estimation for Bitcoin: a comparison of GARCH models," *Economics Letters*, vol. 158, pp. 3–6, 2017.

[5] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, no. 99, pp. 14757–14767, 2017.

[6] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 162–173, 2019.

[7] W. Pennington and J. Evans, "Blockchain-enabled, subscriber-based capital markets index data distribution," *The Journal of Index Investing*, vol. 7, no. 4, pp. 83–87, 2017.

[8] S. X. Xi, W. N. Zhang, Q. L. Zhou, S. XueMing, and B. Li, "High-throughput implementation of SHA512 algorithm based on mimetic computer," *Computer Engineering and Science*, vol. 40, no. 8, pp. 1344–1350, 2018.

[9] S. X. Chen, X. Y. Jiang, J. J. Cai, J. Y. Liu, and W. ChunMing, "Research on mimic security gateway technology based on attack transfer," *Journal of Communications*, vol. 39, no. S2, pp. 76–82, 2018.

[10] J. Steckert and A. Skoczen, "Design of FPGA-based radiation tolerant quench detectors for LHC," *Journal of Instrumentation*, vol. 12, no. 4, p. T04005, 2017.

[11] H. Xu, X. Chen, J. Zhou, Z. Wang, and H. Xu, "Research on basic problems of cognitive network intrusion prevention," in *Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security*, pp. 514–517, Leshan, China, December 2013.

[12] H. Li, R. Lu, L. Zhou, B. Yang, and X. Chen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655–663, 2014.

[13] Q. Wen, D. Wang, S. Feng, Y. Zhang, and G. Yu, "A novel cross-modal hashing algorithm based on multimodal deep learning," *Science China (Information Sciences)*, vol. 60, no. 9, pp. 50–63, 2017.

[14] Y. Kano and T. Nakajima, "A novel approach to solve a mining work centralization problem in blockchain technologies," *International Journal of Pervasive Computing and Communications*, vol. 14, no. 1, pp. 15–32, 2018.

[15] L. Xue, L. Yi, H. Ji, P. Li, and W. Hu, "Symmetric 100-Gb/s TWDM-PON based on 10g-class optical devices enabled by dispersion-supported equalization," *Journal of Lightwave Technology*, vol. 36, no. 2, pp. 580–586, 2018.

[16] A. S. Konoplev, A. G. Busygin, and D. P. Zegzhda, "A blockchain decentralized public key infrastructure model," *Automatic Control and Computer Sciences*, vol. 52, no. 8, pp. 1017–1021, 2018.

[17] H. Liu, A. Kadir, X. Sun, and Y. Li, "Chaos based adaptive double-image encryption scheme using hash function and *S*-boxes," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 1391–1407, 2018.

[18] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, "The blockchain as a decentralized security framework (future directions)," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 18–21, 2018.

[19] W. Liu, W. Qu, J. Gong, and K. Li, "Detection of superpoints using a vector bloom filter," *IEEE Transactions on Information Forensics & Security*, vol. 11, no. 3, pp. 514–527, 2017.

[20] W. H. Zhou, N. D. Jiang, and C. C. Yan, "Research on anti-collision algorithm of RFID tags in logistics system," *Procedia Computer Science*, vol. 154, pp. 460–467, 2019.

[21] D. Yaseen Khudhur, S. Saad Hameed, and S. M. Al-Barzinji, "Enhancing e-banking security: using whirlpool hash function for card number encryption," *International Journal of Engineering and Technology*, vol. 7, no. 2, pp. 281–286, 2018.