

Data Processing Scheme Based on Blockchain

Jinhua Fu^{1,2*}, Mixue Xu¹, Yongzhong Huang^{1,4}, Xueming Si^{1,3}, Chao Yuan¹

Correspondence:

jinhua@zzuli.edu.cn

²School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

Full list of author information is available at the end of the article.

Abstract: In the whitepaper of Bitcoin, a chain of block was proposed by Satoshi Nakamoto. And then blockchain has been rapidly developed. Blockchain is not only limited to the field of cryptocurrency, but also has been well applied in the Internet of Things(IoT), supply chain finance, electronic evidence storage, data sharing, e-government and other fields. Both the public chain and the alliance chain have been well developed. Especially in the field of data processing, blockchain has a good application prospect. Square Kilometre Array (SKA) is a plan of the joint venture of more than ten countries around the world, which is the largest synthetic aperture radio telescope in the world. In SKA, the processing scale of data is large, and there are many data processing nodes in it. The data will be processed in the cloud computing mode. In view of the SKA, this paper proposes a data processing scheme based on blockchain for anti-counterfeiting, anti-tamper and traceability of data. And authenticity and integrity of data are assured. It mainly includes data distribution, data operation and data sharing, which correspond to data reception, data algorithm processing and result sharing of data operation in SKA. With this scheme, the integrity, reliability and authenticity of data are guaranteed. And smart contract, homomorphic hashing, secure container, aggregate signature and one-way encrypted channel are adopted to ensure the intelligence, security and high performance of the scheme.

Keywords:Blockchain, Data Sharing, SKA, Cloud Computing, Privacy Protection

1. Introduction

Blockchain is a distributed ledger technology, which originated from the paper [1]. The blockchain was mainly used in the field of cryptocurrency at the beginning, and bitcoin was the most representative. In addition, Litecoin[2], Monroe [3] and Zcash[4] etc. are accepted. With the advent of Ethereum in 2013, the application field of blockchain has been expanded, in which the combination of smart contract and blockchain plays an important role. However, the target of Ethereum is mainly the public blockchain. Currently, the successful application cases mainly include issuing TOKEN and simple games such as Cryptokitties, which are mainly caused by the low transaction rate of Ethereum and insufficient privacy protection. In Bitcoin only seven transactions per second can be handled. And although the performance of Ethereum is better than Bitcoin's, it can handle only 15-20 transactions per second, and it still cannot meet the practical demand. In addition, some practical applications have higher requirements for privacy protection, which Ethereum cannot meet at present. In 2015, the Hyperledger project was launched, among which the Fabric which was dominated by IBM was the most recognized. Fabric is aimed at the

alliance blockchain, which basically meets the needs of practical applications in terms of performance, privacy protection and usability.

With the development of public chain and alliance chain, the application field of blockchain has been rapidly developed. In the Internet of things[5], supply chain finance, digital data storage certificate, data processing, e-government[6] and other fields, blockchain has been well applied. In the field of data processing, blockchain guarantees the authenticity, security and reliability of data[7]. Among them, the paper[8] introduces medical data sharing based on blockchain. Paper[9] introduces the use of blockchain to protect personal data. And paper[10] introduces the data distribution scheme based on blockchain.

Astronomical data have many characteristics such as a large amount of data, real-time requirements[11], complicated calculation process[12], heterogeneous calculation node[13], diverse storage model, various data access pattern[14], high expansibility, etc. High performance computing, distributed computing, parallel computing, uniform resource management, container technology and Telescope observation control system technology are needed[15]. Current related technologies, such as Apache Hadoop, OpenMP, MPI, etc., all face various problems in processing astronomical data[16]. In the process of SKA data, it is necessary to adopt the mode of cloud computing[17]. In the process of distributed data processing, we should pay more attention to the protection of data[18]. Therefore, the security of data distributed storage[19] and the integrity of data[20] are particularly important. In the process of data processing, there are very high requirements for the synchronization of time[21] and the optimization of algorithm in data merging[22]. Blockchain will play a positive role in ensuring the integrity, security and availability of data.

In the rest of this paper, section 2 mainly introduces the data distribution scheme based on blockchain, which mainly reflects the generation and collection of data in SKA. Section 3 introduces the scheme of data operation, which reflects the combination of collected data and related algorithm. Section 4 introduces the scheme of sharing data, which reflects the sharing problem of the result after the original data is processed by related algorithm. Section 5 concludes this paper.

2. Preliminaries

In this section, we first give some notations used in the rest of this paper. If S is a set, then $|S|$ means the number of elements in this set. If b is a real number, then $a \leftarrow b$ means $a = b$. If C is a node and c is an element, then $C \leftarrow c$ means sending c to C . If a and b are two real numbers, then $a || b$ means the cascading of a and b .

2.1 Bilinear mapping

There, G_1 and G_2 are two multiplicative cyclic groups of prime order p , g_1 is a generator of G_1 and g_2 is a generator of G_2 . ψ is a computable isomorphism from G_2 to G_1 , with $\psi(g_2) = g_1$. A bilinear pairing is defined as to be $e = (n, G_1, G_2, e, g_1, g_2)$ where $G_1 = \langle g_1 \rangle$, $G_2 = \langle g_2 \rangle$ and G_1, G_2 are multiplicative groups of order n . Let $e: G_1 \times G_2 \rightarrow G_T$ be a map with the following properties:

- **Bilinear:** $\forall u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}_n : e(u^a, v^b) = e(u, v)^{ab}$
 - **Non-degenerate:** There exists $u \in G_1, v \in G_2$ such that $e(u, v) \neq O$, where O means the identity of G_T
 - **Computability:** There is an efficient algorithm to compute $e(u, v)$ for all $u \in G_1, v \in G_2$
- Then e is a bilinear mapping.

2.2 Aggregate signature

Aggregate signature is a variant signature scheme used to aggregate any number of signatures into one signature. Supposed there are n users in the system $\{u_1, u_2, \dots, u_n\}$ and n public keys $\{pk_1, pk_2, \dots, pk_n\}$, n messages $\{m_1, m_2, \dots, m_n\}$ and n signatures $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ of these messages. The generator of the aggregate signature (here the generator can be arbitrary and does not need to be in $\{u_1, u_2, \dots, u_n\}$) can aggregate $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ to a short signature σ . Importantly, the aggregate signature is verifiable, that is, given a set of public keys $\{pk_1, pk_2, \dots, pk_n\}$ and its signatures of the original message set $\{m_1, m_2, \dots, m_n\}$, it can be verified that the user u_i has made a signature of message m_i respectively. The execution of the aggregate signature is described in detail below.

AS = (Gen, Sign, Verify, AggS, AggV) is a quintuple of polynomial time algorithm, details are as follows:

DS = (Gen, Sign, Verify) is a common signature scheme, also known as the benchmark for aggregate signature.

Aggregation Signatures Generation (AggS). On the basis of Gen and Sign, realizes the common signature function and the aggregation of $\{m_1, m_2, \dots, m_n\}, \{u_1, u_2, \dots, u_n\}$ and $\{\sigma_1, \sigma_2, \dots, \sigma_n\}$ and aggregating a new signature σ_n .

Aggregation signature verification (AggV). Supposed that each u_i corresponds to a public-private key pair $\{pk_i, sk_i\}$. If $\text{AggV}(pk_1, \dots, pk_n, m_1, \dots, m_n, \text{AggS}(pk_1, \dots, pk_n, m_1, \dots, m_n, \text{Sign}(sk_1, m_1), \dots, \text{Sign}(sk_n, m_n))) = 1$, then output 1, otherwise output 0.

Aggregate signature can also support incremental aggregation, so σ_1 and σ_2 can be aggregated to σ_{12} , then σ_{12} and σ_3 can be aggregated to σ_{123} .

2.3 Homomorphic Hash

Homomorphism is a mapping of two algebraic structures in abstract algebra that remain structurally constant. There, G_1 and G_2 are two groups, f is a mapping from G_2 to G_1 . If $\forall a, b \in G_2$, $f(ab) = f(a)f(b)$, then f is called a homomorphism from G_2 to G_1 .

The homomorphic hash has long been used in peer-to-peer networks[23], which is used with correction code, network code together against attack events. In a peer-to-peer network, each peer will get the original data block directly from the other peers, so the hash functions like SHA1 can be used to directly verify the correctness of the received data block by comparing the hash value of the received data block with the original hash value.

Using the homomorphic hash function mentioned in the dedication [24] $h_G(\cdot)$, There is a set of hash parameters in $h_G(\cdot)$, $G = (p, q, g)$. The parameter description is shown in table 1. Each of these elements in g can be represented as $x^{(p-1)/q} \bmod p$ where $x \in Z_p$ and $x \neq 1$.

$$h_G(\cdot) : \{0, 1\}^k \times \{0, 1\}^\beta \rightarrow \{0, 1\}^{\lambda_p} \quad (1)$$

$$rand(\cdot) : \{0, 1\}^k \times \{0, 1\}^t \rightarrow \{0, 1\}^t \quad (2)$$

Where $rand(\cdot)$ is a pseudo-random function, used as a pseudo random number generator to initialize homomorphism hash function parameters in the process of generating, random number generation in the process of tag generates, and determine the random data block in the process of challenge, thus make challenges can even cover all of the data.

For a block b_i , the hash value can be calculated according to:

$$h(b_i) = \prod_{k=1}^m g_k^{b_{k,i}} \bmod p \quad (3)$$

The hash values of original block (b_1, b_2, \dots, b_n) is $h(b_1), h(b_2), \dots, h(b_n)$.

Given a coding block e_j and a coefficient vector $(c_{j,1}, c_{j,2}, \dots, c_{j,n})$, then the homomorphic hash function $h_G(\cdot)$ satisfies the following equation:

$$h(e_j) = \prod_{i=1}^n h^{c_{j,i}}(b_i) \quad (4)$$

This feature can be used to verify the integrity of a code block. First, the publisher needs to calculate the homomorphic hash values of each data block in advance. The download downloads these homomorphic hash values, and once the verification block is received, it calculates its hash value using equation (3), and then uses equation (4) to verify the correctness of the verification block[25].

Table 1 Parameter description

| parameter | description |
|-------------|---|
| λ_p | Discrete logarithmic secure parameter (1024 bit) |
| λ_q | Discrete logarithmic secure parameter (257bit) |
| p | Random prime number $ p = \lambda_p$ |
| q | Random prime number $q p - 1, q = \lambda_q$ |

| | |
|----------------|--|
| β | Data block size (16 kB) |
| m | The number of subblocks contained in each bloc $m = \left\lceil \frac{\beta}{\lambda_q - 1} \right\rceil (512bit)$ |
| g | Generate the $1 \times m$ row vector, randomly selected from the Z_p |
| G | Hash parameters, including $G = (p, q, g)$ |
| n | Block number |
| <i>seed</i> | Seed of keystream generator |
| <i>MAXRAND</i> | Maximum possible output of $rand(\cdot)$ |
| <i>MINRAND</i> | Minimum possible output of $rand(\cdot)$ |

3. Results and Discussion

3.1 Blockchain-based Data Distribution Scheme

Here, we simplify the process of receiving astronomical data in SKA. The *SOURCE* represents the original astronomical data, and the *Data Receiving Station (DRS)* represents the real astronomical data receiving device. The setting of DRS itself is distributed, different DRS are responsible for receiving data within their own responsible areas. Considering the limitation of hardware functions, the DRS is only responsible for data reception, temporary storage and data forwarding, does not participate in data calculation. All data calculation is completed by the *Data Processing Node (DPN)*, which is connected to the blockchain. The concrete architecture is shown in Figure 1.

The processing process of data from *SOURCE* to *DRS* is relatively simple, which involves the processing of data format and setting of storage mode, which is not the focus of this paper. Here, the execution process of *DRS* to *DPN* is mainly introduced.

Here, we adopt the idea of distributed storage in IPFS, as shown in the Figure 2.

Here, each block contains a list of trading objects, a link to the previous block, and a hash value for the state tree/database.

Here we introduce the method to import data into blockchain. Let q be a large prime number, select $P \in G_1, Q \in G_2$ to define an additive group G_1 and a multiplicative group G_2 with order q . A bilinear mapping $e: G_1 \times G_2 \rightarrow G_T$, and hash functions $H: \{0,1\}^* \rightarrow \{0,1\}^*$, $H_0: \{0,1\}^* \rightarrow Z_q^*$, $H_1: \{0,1\}^* \times G_1 \rightarrow G_2$, $H_2: \{0,1\}^* \rightarrow G_1$, $H_{DV}: \{0,1\}^* \rightarrow G_1$. The number of data receiving stations is m , and the number of data processing nodes responsible for the i -th data

receiving station is m_i , current view is v .

(1) With the current view, calculate $P_v = v \cdot P$, Combined with the existing parameters, the system parameters are obtained $Params = \{G_1, G_2, e, q, P, Q, P_v, H_0, H_1, H_2, H_{DV}\}$

(2) The user u_i selects the random value $x_i \in Z_q^*$ as the secret value of it and calculates $P_i = x_i \cdot P, Q_i = H_1(ID_i \parallel P_i), D_i = v \cdot Q_i$ to generate the user's private key $S_i = (D_i, x_i)$.

Supposed the public key of the j -th ($j = 1, 2, \dots, m_i$) Data Processing Node (DPN_i^j) of i -th ($i = 1, 2, \dots, m$) Data Receiving Station (DRS_i) in the r -th round is $\{pk_i^1, pk_i^2, \dots, pk_i^{m_i}\}$. The data produced by the SOURCE is D_i^r , Each DRS consensus for the resulting data with static aggregate Practical Byzantine Fault Tolerance (PBFT) [26,27]. The specific process is shown in algorithm 1

Algorithm 1 Method to import data into blockchain

Input: the number of DRS m , round r , public keys set $\{v_i^1, v_i^2, \dots, v_i^{m_i}\}$.

Output: blocks B_r

1: $S_r \leftarrow DRS_{r \bmod m}$

2: **for** $i = 1$ **to** m

3: $M_i^r \leftarrow DPN_i^{(r+i) \bmod m_i}$

4: $M_i^r \leftarrow (D_i^r, H(D_i^r))$

5: M_i^r selects $r_i^r \in Z_q^*$ and computes $R_i^r = r_i^r \cdot P, h_i^r = H_0(ID_i^r \parallel D_i^r \parallel P_i^r \parallel R_i^r), T = H_2(P_v)$

6: M_i^r computes $V_i^r = D_i^r + h_i^r \cdot r_i^r \cdot T + x_i^r \cdot Q$ and outputs $\sigma_i^r = (V_i^r, R_i^r)$

7: $S_r \leftarrow (D_i^r, \sigma_i^r)$

8: S_r computes $V = \sum_{i=1}^m V_i, R = \sum_{i=1}^m h_i R_i, \sigma = (V, R), D \leftarrow D_1^r \parallel D_2^r \parallel \dots \parallel D_m^r$

9: **end for**

10: **for** $i = 1$ **to** m **do**

11: $M_i^r \leftarrow (D, \sigma)$

12: **for** $j = 1$ **to** m_i **do**

13: $DPN_i^j \leftarrow (D, \sigma)$

14: **end for**

15: **end for**

To verify the validity of aggregate signature σ , you can implement this algorithm. with the system parameter $Params$, the user's corresponding identity list $ID = \{ID_1, \dots, ID_n\}$, public keys list $P = \{P_1, \dots, P_n\}$, messages list $M = \{m_1, \dots, m_n\}$ and signature list $\sigma = \{\sigma_1, \dots, \sigma_n\}$ to computer $Q_i = H_1(ID_i \parallel P_i)$ and $T = H_2(P_v)$, and then verify the following equation:

$$e(V, P) = e\left(\sum_{i=1}^n Q_i, P_v\right) e(T, R) e\left(Q, \sum_{i=1}^n P_i\right) \quad (5)$$

If true, the validation passes; otherwise, the validation fails.

The correctness of this basic framework is given below. Theorems 1 and Theorems 2 respectively give the correctness of the verification process of a single signature and the correctness of the verification process of an aggregate signature.

Theorem 1 The verification process of a single signature is correct.

Proof: The verification process of the signature $\sigma_i = (V_i, R_i)$ that DRS_i does for D_i is as follows:

$$\begin{aligned}
 e(V_i, P) &= e(D_i + h_i r_i T + x_i Q, P) \\
 &= e(D_i, P) e(T, h_i r_i P) e(x_i Q, P) \\
 &= e(Q_i, P_v) e(T, h_i R_i) e(Q, P_i)
 \end{aligned} \tag{6}$$

Theorem 2 The verification process of aggregation signature is correct.

Proof:

$$\begin{aligned}
 e(V, P) &= e\left(\sum_{i=1}^n V_i, P\right) = e\left(\sum_{i=1}^n D_i + x_i Q + h_i r_i T, P\right) \\
 &= e\left(\sum_{i=1}^n D_i, P\right) e\left(\sum_{i=1}^n h_i r_i T, P\right) e\left(\sum_{i=1}^n x_i Q, P\right) \\
 &= e\left(\sum_{i=1}^n Q_i, P_v\right) \prod_{i=1}^n e(T, h_i R_i) \prod_{i=1}^n e(Q, P_i) \\
 &= e\left(\sum_{i=1}^n Q_i, P_v\right) e(T, R) e\left(Q, \sum_{i=1}^n P_i\right)
 \end{aligned} \tag{7}$$

3.2 Blockchain-based Data Operation Scheme

The Science Data Processor (SDP)[28] is the SKA Data processing module. The main Data is from the Central Signal Processor (CSP) module[29], the metadata is from the Telescope Manager (TM) module, and the Signal and Data Transport (SaDT) module is responsible for data transmission. Multiple regional data processing centers will be built. The main functions of SDP include:

- Extract data from CSP and TM modules
- Treat source data as data products that can be used for scientific research
- Archive and store data products
- Provide access to data products
- Control and feedback information to TM module for timely challenge observation

In SKA SDP, the two most important computational tasks are FFT[30] and Gridding[31]. These two algorithms account for an important part of the total computation, and their efficient implementation is of great help to the design of SKA SDP.

As shown in Figure 3, in the data calculation scheme based on blockchain, the Data Supply Node(DSN) and the Algorithm Supply Node(ASN) are separated, and all data and algorithm enter the Secure Container[32] through a one-way encrypted channel under the control of the Smart Contract(SM) to perform calculations. Providers and the provided time of data and algorithms are recorded on the blockchain through SM. Supposed there are w Data Supply Nodes and one Algorithm Supply Node. Before entering Secure Container, All the data D_i ($i = 1, 2, \dots, w$) and algorithms A are encrypted by the private key(sk_i) of DSN_i and the private key(sk_a) of ASN. And

the data and algorithm are signed by the private key SC_{sk} of the Secure Container. And then decrypt and verify it after entering Secure Container by the public key(pk_i) of DSN_i , the public key(pk_a) of ASN and the public key of Secure Container. The specific process is shown in algorithm 2 and algorithm 3.

Algorithm 2 Data Operation Based on Blockchain(1)

Input: hash function $H : \{0,1\}^* \rightarrow \{0,1\}^*$, private key of security container SC_{sk} , private keys set $\{sk_1, sk_2, \dots, sk_w\}$ of DSNs, private key sk_a of ASN

Output: block b

1: for $i = 1$ to w

2: $D_i' \leftarrow Enc_{SC_{sk}}(Sig_{sk_i}(D_i))$

3: $SC \leftarrow D_i'$

4: $b_i \leftarrow H(D_i) || time_i$

5: end for

6: $A' \leftarrow Enc_{SC_{sk}}(Sig_{sk_a}(A))$

7: $SC \leftarrow A'$

8: $b_a \leftarrow H(A) || time_a$

9: $b \leftarrow b_1 || b_2 || \dots || b_w || b_a$

Algorithm 3 Data Operation Based on Blockchain(2)

Input: hash function $H : \{0,1\}^* \rightarrow \{0,1\}^*$, public key of security container SC_{pk} , public keys set $\{pk_1, pk_2, \dots, pk_w\}$ of DSNs, public key pk_a of ASN pk_a

Output: block b

1: for $i = 1$ to w

2: $D_i \leftarrow Ver_{pk_i}(Dec_{SC_{pk}}(D_i'))$

3: $SC \leftarrow D_i$

4: $b_i' \leftarrow H(D_i) || time_i$

5: end for

6: $A \leftarrow Ver_{pk_a}(Dec_{SC_{pk}}(A'))$

7: $SC \leftarrow A$

8: $b_a' \leftarrow H(A) || time_a$

9: $b' \leftarrow b_1' || b_2' || \dots || b_w' || b_a'$

3.3 Blockchain-based Data Sharing Scheme

The Data Requirement Nodes which is Represented by these public keys $\{pk_1, pk_2, \dots, pk_r\}$ of the calculation result is determined in advance through the smart contract. Under intrusive surveillance, the calculated result Re is shared to the nodes represented by these public keys. Shared results, targets and Shared time are recorded on the blockchain through smart contracts.

The concrete architecture is shown in Figure 4.

Before recording these on blockchain, it is necessary to verify the target, and the target verifies the calculated results. If the verification passes, then signs it. If more than 2/3 of the target's signature is obtained, the block formed will be recorded on blockchain. The simple architecture is shown in figure 5.

Supposed that there are r Data Requirement Nodes (DRNs). The calculated results Re are encrypted by the public key pk_i of DRN_i ($i = 1, 2, \dots, r$) and signed by the private key SC_{sk} of SC to obtain Re_i . The cascading of hash value of Re_i and the time forms the block b_i . Homomorphic hash h is used to pk_i . Then, all the b_i ($i = 1, 2, \dots, r$) form the final block b . Finally, verify the homomorphic hash, if it passes, send the calculation results Re_i to the DRN_i , and decrypt it by the private key sk_i of DRN_i and the public key SC_{pk} of secure container. The specific process is shown in algorithm 4.

Algorithm 4 Data Sharing Based on Blockchain

Input: hash function $H : \{0,1\}^* \rightarrow \{0,1\}^*$, public keys set of the target $\{pk_1, pk_2, \dots, pk_r\}$, public key and private key of security container SC_{pk}, SC_{sk}

Output: block b

1: for $i = 1$ to r

2: $Re_i \leftarrow Enc_{SC_{sk}}(Enc_{pk_i}(Re))$

3: $b_i \leftarrow H(Re_i) \parallel time$

4: $h_i \leftarrow h(pk_i)$

5: end for

6: $b \leftarrow b_1 \parallel b_2 \parallel \dots \parallel b_r$

7: $h \leftarrow \prod_{i=1}^r h(pk_i)$

8: if $h = h(\sum_{i=1}^r pk_i)$

9: for $i = 1$ to r

10: $Re \leftarrow Dec_{sk_i}(Dec_{SC_{pk}}(Re_i))$

11: $pk_i \leftarrow Re$

12: end for

13: end if

4. Conclusion

This paper discusses data storage, data operation and data sharing in big data processing. Using the data structure of blockchain, combined with intelligent contract, homomorphic hashing, secure container, aggregate signature and one-way encrypted channel, ensures the authenticity,

integrity and reliability of data for the collection, calculation and results sharing of astronomical data. Combined with SKA project, this scheme is applied in astronomical data processing. It provides innovative ideas for the application of blockchain in the fields of large data volume, fast data generation, high complexity of data processing and high value of data processing results.

Abbreviations

IoT: Internet of Things; SKA: Square Kilometre Array; DRS: Data Receiving Station; DPN: Data Processing Node; PBFT: Practical Byzantine Fault Tolerance; SDP: Science Data Processor; CSP: Central Signal Processor; TM :Telescope Manager; SaDT :Signal and Data Transport; DSN :Data Supply Node; ASN:Algorithm Supply Node; SM:Smart Contract; DRNs:Data Requirement Nodes.

Acknowledgements

We gratefully acknowledge the anonymous reviewers for taking time reviewing our manuscript.

Authors' contributions

All authors contributed to the development of the ideas, design of the study, the theory, result analysis, and article writing.

Funding

This research work is supported by the Innovative Research Groups of the National Natural Science Foundation of China (Grant No.61521003), Intergovernmental Special Programme of National Key Research and Development Programme (2016YFE0100300, 2016YFE0100600) , National Scientific Fund Programme for Young Scholar (61672470) and Science and Technology Project of Henan Province (182102210617, 202102210351).

Availability of data and materials

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

¹State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China.

²School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China

³School of Computer science, Fudan University, Shanghai 201203, China

⁴School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China

Author E-mail

Jinhua Fu:jinhua@zzuli.edu.cn

Mixue Xu: mixue_xu@163.com

Yongzhong Huang:2389483289@qq.com

Xueming Si: sxm@fudan.edu.cn

Chao Yuan: yc_xxgcdx@163.com

Corresponding author: Jinhua Fu;E-mail:jinhua@zzuli.edu.cn

References

1. Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008
2. Padmavathi M, Suresh R M. Secure P2P Intelligent Network Transaction using Litecoin[J]. *Mobile Networks & Applications*, 2018:1-9.
3. Nicolas van Saberhagen. Cryptonote v 2. 0.HYPERLINK <https://cryptonote.org/whitepaper.pdf>, 2013.
4. Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza. Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP)*, 2014 IEEE Symposium on, pages 459-474. IEEE, 2014.
5. Xiaolong Xu, Xuyun Zhang, Honghao Gao, Yuan Xue, Lianyong Qi, Wanchun Dou. BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing, *IEEE Transactions on Industrial Informatics*, 2019. DOI: 10.1109/TII.2019.2936869.
6. Yadong Huang, Yueting Chai, Yi Liu, and Jianping Shen. Architecture of Next-Generation E-Commerce Platform. *TSINGHUA SCIENCE AND TECHNOLOGY*, 24(1): 18-29, 2019.
7. Guorui Li, Sancheng Peng, Cong Wang, Jianwei Niu, and Ying Yuan. An Energy-Efficient Data Collection Scheme Using Denoising Autoencoder in Wireless Sensor Networks. *TSINGHUA SCIENCE AND TECHNOLOGY*, 24(1):86-96, 2019.
8. Xia Q, Sifah E B, Asamoah K O, et al. MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain[J]. *IEEE Access*, 2017, 5(99):14757-14767.
9. Zyskind G, Nathan O, Alex. Decentralizing Privacy: Using Blockchain to Protect Personal Data[C]// *IEEE Security and Privacy Workshops*. IEEE Computer Society, 2015:180-184.
10. Kishigami J, Fujimura S, Watanabe H, et al. The Blockchain-Based Digital Content Distribution System[C]// *IEEE Fifth International Conference on Big Data and Cloud Computing*. IEEE Computer Society, 2015:187-190.
11. Luning Liu, Xin Chen, Zhaoming Lu, Luhan Wang, and Xiangming Wen. Mobile-Edge Computing Framework with Data Compression for Wireless Network in Energy Internet. *TSINGHUA SCIENCE AND TECHNOLOGY*, 24(3): 271-280, 2019.

12. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, Y. Li, "A survey of matrix completion methods for recommendation systems", *Big Data Mining and Analytics*, vol.1, no. 4, pp. 308-323, 2018.
13. Xiaolong Xu, Chengxun He, Zhanyang Xu, Lianyong Qi, Shaohua Wan, MZA Bhuiyan. Joint Optimization of Offloading Utility and Privacy for Edge Computing Enabled IoT. *IEEE Internet of Things Journal*, 2019. DOI: 10.1109/JIOT.2019.2944007.
14. Hanwen Liu, Huaizhen Kou, Chao Yan, Lianyong Qi. Link prediction in Paper Citation Network to Construct Paper Correlated Graph. *EURASIP Journal on Wireless Communications and Networking*, 2019. DOI: 10.1186/s13638-019-1561-7.
15. Hanisch R J, Jacoby G H. *Astronomical Data Analysis Software and Systems X*[J]. *Publications of the Astronomical Society of the Pacific*, 2001, 113(784):772-773.
16. Xiaoxiao Chi, Chao Yan, Hao Wang, Wajid Rafique, Lianyong Qi. Amplified LSH-based Recommender Systems with Privacy Protection. *Concurrency and Computation: Practice and Experience*, 2020. DOI: 10.1002/CPE.5681
17. Lianyong Qi, Wanchun Dou, Chunhua Hu, Yuming Zhou and Jiguo Yu. A Context-aware Service Evaluation Approach over Big Data for Cloud Applications, *IEEE Transactions on Cloud Computing*, 2015. DOI: 10.1109/TCC.2015.2511764.
18. Wenwen Gong, Lianyong Qi, Yanwei Xu. Privacy-aware Multidimensional Mobile Service Quality Prediction and Recommendation in Distributed Fog Environment. *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.
19. Lianyong Qi, Wanchun Dou, Wenping Wang, Guangshun Li, Hairong Yu, Shaohua Wan. Dynamic Mobile Crowdsourcing Selection for Electricity Load Forecasting. *IEEE ACCESS*, 6: 46926-46937, 2018.
20. iaolong Xu, Shucun Fu, Lianyong Qi, Xuyun Zhang, Qingxiang Liu, Qiang He, Shancang Li. An IoT-Oriented data placement method with privacy preservation in cloud environment. *Journal of Network and Computer Applications*, vol. 124, pp. 148-157, 2018.
21. C. Zhang, M. Yang, J. Lv, W. Yang, "An improved hybrid collaborative filtering algorithm based on tags and time factor", *Big Data Mining and Analytics*, vol.1, no.2, pp. 128-136, 2018.
22. Y. Liu, S. Wang, M. S. Khan, J. He, "A novel deep hybrid recommender system based on auto-encoder with neural collaborative filtering", *Big Data Mining and Analytics*, vol.1, no.3, pp. 211-221, 2018.
23. Lianyong Qi, Xuyun Zhang, Wanchun Dou and Qiang Ni. A Distributed Locality-Sensitive Hashing based Approach for Cloud Service Recommendation from Multi-Source Data. *IEEE Journal on Selected Areas in Communications*, 35(11): 2616-2624, 2017.
24. KROHN M, FREEDMAN M J, MAZIERES D. On-the-fly verification of rateless erasure codes for efficient content distribution[C]. *Proc of IEEE Symposium on Security and Privacy*. Lee Badger: IEEE, 2004: 226-240.

25. Shi H, Liu W, Cao T. An improved method of data integrity verification based on homomorphic hashing in cloud storage[J]. Journal of Hohai University, 2015, 43(3):278-282.
26. Chao Y , Mi-Xue X U , Xue-Ming SI . Optimization Scheme of Consensus Algorithm Based on Aggregation Signature[J]. Computer Science, 2018.
27. Miguel O T D C. Practical Byzantine Fault Tolerance[J]. Acm Transactions on Computer Systems, 2002, 20(4):398-461.
28. Broekema P C, Van Nieuwpoort R V, Bal H E. The Square Kilometre Array Science Data Processor. Preliminary compute platform design[J]. Journal of Instrumentation, 2016, 10(7):C07004-C07004.
29. Fiorin L, Vermij E, Van Lunteren J, et al. An energy-efficient custom architecture for the SKA1-low central signal processor[J]. 2015:1-8.
30. Moreland K, Angel E. The FFT on a GPU[C]. ACM Siggraph/eurographics Conference on Graphics Hardware. Eurographics Association, 2003:112-119.
31. Suter E, Friis H A, Vefring E H, et al. A Novel Method for Multi-Resolution Earth Model Gridding[C]. SPE Reservoir Simulation Conference, 20-22 February, Montgomery, Texas, USA. 2017.
32. Cahill C P, Martin J, Pagano M W, et al. Client-based authentication technology: user-centric authentication using secure containers[C]. Proceedings of the 7th ACM workshop on Digital identity management. ACM, 2011:83-92.

Figures

Figure 1. Data Distribution Based on Blockchain.The concrete architecture of the process of receiving astronomical data in SKA.The SOURCE represents the original astronomical data, and the Data Receiving Station (DRS) represents the real astronomical data receiving device. All data calculation is completed by the Data Processing Node (DPN), which is connected to the blockchain.

Figure 2.Distributed Storage in IPFS.Here is the distributed storage in IPFS.Each block contains a list of trading objects, a link to the previous block, and a hash value for the state tree/database.

Figure 3. Data Operation Based on Blockchain.In the data calculation scheme based on blockchain, the Data Supply Node (DSN) and the Algorithm Supply Node (ASN) are separated, and all data and algorithm enter the Secure Container through a one-way encrypted channel under the control of the Smart Contract (SM) to perform calculations.

Figure 4. Data Sharing Based on Blockchain.This is a data sharing architecture based on blockchain.

Figure 5. Validation of Smart Contract.This is the architecture of smart contract signature.If more than 2/3 of the target's signature is obtained, the block formed will be recorded on blockchain.

