A Blockchain-Based Trusted Data Management Scheme in Edge Computing

Ma Zhaofeng, *Member, IEEE*, Wang Xiaochang, Deepak Kumar Jain, Haneef Khan, Gao Hongmin, and Wang Zhen

***Abstract*—With rapid development of computing technologies, large amount of data are gathered from edge terminals or Internet of Things (IoT) devices, however data trust and security in edge computing environment are very important issues to be considered, especially when the gathered data are fraud or dishonest, or the data are misused or spread without any authorization, which may lead to serious problems. In this article, a blockchain-based trusted data management scheme (called BlockTDM) in edge computing is proposed to solve the above problems, in which we proposed a flexible and configurable blockchain architecture that includes mutual authentication protocol, flexible consensus, smart contract, block and transaction data management, blockchain nodes management, and deployment. The BlockTDM scheme can support matrix-based multichannel data segment and isolation for sensitive or privacy data protection, and moreover, we have designed user-defined sensitive data encryption before the transaction payload stores in blockchain system, and have implemented conditional access and decryption query of the protected blockchain data and transactions through smart contract. Finally, we have evaluated the proposed BlockTDM scheme security, availability, and efficiency with large amount of experiments. Analysis and evaluations manifest that the proposed BlockTDM scheme provides a general, flexible, and configurable blockchain-based paradigm for trusted data management with tamper-resistance, which is suitable for edge computing with high-level security and creditability.**

***Index Terms*—Blockchain, consensus, edge computing, smart contract, trusted data management.**

I. Introduction

DUE TO the evolution of computing technologies such as artificial intelligence, big data, and cloud-based computing, more and more data are created and gathered from user-side via traditional or mobile devices, and some analyzing and computing work can commit to cloud platform for processing. However, the cloud-based computing mode relies on the committing-waiting-return procedure, which is not so efficient for light-weight applications, in fact, with the processing and computing capacity enhancement, edge computing [1]–[3] provides a reasonable way for efficient dealing and processing between cloud server and local device, in which light-weight work is done in the edge environment(also called fog computing) without committing to cloud center [3]. However, data reliability and honesty are an important issues in edge computing, especially once fraud or dishonest data are integrated in the system, moreover, misusing of the captured data may lead to data leakage or privacy explorer [1]–[3]. In fact, traditional centralized system usually relies on the centered creditability; however, it is not often so reliable and trusted, especially there exists the risk data tampering from inner administrator or hacked by external intruder. Fortunately, blockchain [4]–[9] provides a decentralized, secure, tamper-resistance, and self-organization computing technology, which can support token-based economic application such as digital concurrency, international settlement and trade, shared economic, as for trusted data management, blockchain can ensure and enhance distributed, tamper-resistance secure data management [4]–[7].

Upon the edge computing technologies, Abbas *et al.* [1] surveyed mobile edge computing, which pointed out that edge computing is now a promising technology in 5G mobile environment. Esposito and Castiglione [2] studied challenges of connecting edge and cloud computing, in which edge computing architecture and challenges such as security, privacy and forensics are the most important issues to be considered. Shi and Dustdar [3] gave a full survey of the promise of edge computing and its application in future next mobile and Internet of Things (IoT) fields. Henry [8] studied the blockchain access privacy challenges, such as personal data protection, tradeoff between privacy and criminals prevention. Aste *et al.* [9] studied the foreseeable impact on society and industry of blockchain technologies, and the pointed out that blockchain is a new technology for future social and government management. Dinh *et al.* [10] studied blockchain for data processing, which include data storage, data protection, and data tamper-resistance. Gai *et al*. [11] proposed blockchain-enabled reengineering in cloud datacenters environment. Zhe [12] proposed blockchain-based decentralized trust management in vehicular networks, Xu *et al*.

Manuscript received June 23, 2019; accepted July 26, 2019. Date of publication August 6, 2019; date of current version January 16, 2020. The work was supported in part by the National Natural Science Foundation of China under Grant 60803157 and Grant 61272519, and in part by the Funds of Blockchain Joint Lab between BUPT and BCT, CAPTONE. Paper no. TII-19-2657. *(Corresponding author: Ma Zhaofeng.)*

M. Zhaofeng, W. Xiaochang, G. Hongmin, and W. Zhen are with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: mzf@bupt.edu.cn; wxc_vw@163.com; gaohm1024@sina.com; wangzhen_bupt@foxmail.com).

D. K. Jain is with the Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: deepak@cqupt.edu.cn).

H. Khan is with Jazan University, Jazan 22822, Saudi Arabia (e-mail: haneeskhan@jazan.edu.sa).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2019.2933482

[13] proposed blockchain-based decentralized content trust for docker images, which provided a new mechanism for image content protection in a decentralized way. Anjum *et al.* [14] proposed the blockchain standards for compliance and trust, however, it is hard to standardze the developing new technologies. Turkanović *et al*. [15] built up EduCTX blockchain projects and proposed blockchain-based higher education credit platform. Zhaofeng *et al.* [16] proposed a blockchain-based scheme for digital rights management (named DRMChain), the DRMCchain built up a trusted model of DRM. Hinsaku Kiyomoto *et al*. [17] proposed a blockchain-based dataset distribution scheme for anonymous data management.

II. BlockTDM: A Blockchain-Based Trusted Data Management Scheme

A. Conception Model of the Proposed BlockTDM scheme

Upon the problems in edge computing, in this article, we proposed a blockchain-based trusted data management scheme (called BlockTDM) with a flexible and configurable architecture, which includes the edge device layer, the blockchain network layer, the edge nodes layer, and the cloud center layer. In the proposed BlockTDM scheme, mutual authentication protocol, security-enhanced consensus and smart contract, and block and transaction data management are studied in details. The blockchain-based trusted data management architecture of the BlockTDM scheme is described as Fig. 1.

1) Edge Device Layer: The edge device layer includes traditional PC or mobile computing devices and is responsible for data gathering and collecting.

2) Blockchain Network Layer: After gathering the data from the edge devices, the data are committed to the DB or IPFS network for storage, and simultaneously stores the critical data and its hash value into the blockchain, thus all the operations such as data invoking, transferring, or usage are executed in the trusted environment with smart contract and output blocks and transactions with tamper-resistance, then it can provide high-level trust and security of data management.

3) Edge Nodes Layer: The edge nodes provide resource perception, service perception, task scheduling, and data collaboration and multiview that are related to blockchain layer and cloud center layer for data interaction.

4) Cloud Center Layer: The cloud processing layer is responsible for heavy-weight and complex problem solving and data storage of the edge computing.

B. BlockTDM Scheme Protocol and Algorithm

1) BlockTDM Mutual Authentication Protocol: The mutual authentication is necessary for the permissioned blockchain for KYC (know your customer), which is important for membership management and responsibility tracing. In the BlockTDM scheme, we proposed a mutual authentication protocol for the permissioned blockchain membership management. The mutual authentication protocol based on certificates is described in the following section.

a) Authentication initialization:

Step 1: The user and the peer have their certificate Cert_A and Cert_P. The certificate has the following data structure:

$$\text{Cert}_A = \{ID_A, K_{A\text{pub}}, \text{Date}_A, \text{Issuer}, \text{Algorithm}, \text{Sig}_{CA}(.)\}$$
$$\text{Cert}_P = \{ID_P, P_{P\text{pub}}, \text{Date}_P, \text{Issuer}, \text{Algorithm}, \text{Sig}_{CA}(.)\}. \quad (1)$$

Step 2: A selects a random integer r_A, $1 \le r_A \le n-1$, and computes

$$Q_A = r_A G. \quad (2)$$

Step 3: A then passes Q_A and Cert_A to peer *P*.

b) A authenticates P:

Step 1: *P* selects a random integer r_P, $1 \le r_P \le n\text{-}1$, and computes

$$Q_P = r_P G. \quad (3)$$

Step 2: *A* then passes Q_A and Cert_A to peer *P*.

Step 3: Once *A* receives the data Q_P and Cert_P, then *A* verifies the validity of Cert_P by the CA public key; if the verification is right, then it manifests Cert_P is issued by CA.

Step 4: *A* then computes

$$K_{AP} = r_A Q_P = r_A r_P G. \quad (4)$$

Step 5: The user *A* then encrypts the K_{AP} by $K_{P\text{pub}}$, and encrypts Cert_A by K_{AP} as follows:

$$C_1 = E_{K_{P\text{pub}}}(K_{AP})$$
$$C_2 = E_{K_{AP}}(\text{Cert}_A) \quad (5)$$

and passes the cipher C_1 and C_2 to peer *P*.

c) P authenticates A:

Step 1: The peer *P* then computes

$$K_{PA} = r_P Q_A = r_P r_A G. \quad (6)$$

Step 2: The peer *P* then encrypts the K_{PA} by the Peer's public key $K_{A\text{pub}}$, and encrypts Cert_P by K_{PA} as follows:

$$C_3 = E_{K_{A\text{pub}}}(K_{PA})$$
$$C_4 = E_{K_{PA}}(\text{Cert}_P) \quad (7)$$

and then passes the cipher *C*3 and *C*4 to A.

In fact, when *A* or *P* receives the cipher message, then *A* and *P* can verify the possible session key K_{AP} or K_{PA} are true or false as follows:

$$K_{PA} = D_{K_{P priv}}(E_{K_{P\text{pub}}}(K_{PA}))$$
$$\text{Cert}_A{}' = D_{K_{PA}}(E_{K_{PA}}(\text{Cert}_A)). \quad (8)$$

If $\text{Cert}'_A = \text{Cert}_A$, then it manifests that the session key is correct, otherwise the *A* or *P* cannot computes the correct Cert_A or Cert_P. Then the peer *P* authenticated the identity of *A*.

And as the above operation, *A* can authenticates *P* as that *P* authenticates *A*

$$K_{AP} = D_{K_{A priv}}(E_{K_{A\text{pub}}}(K_{AP}))$$
$$\text{Cert}_P{}' = D_{K_{AP}}(E_{K_{AP}}(\text{Cert}_P)). \quad (9)$$

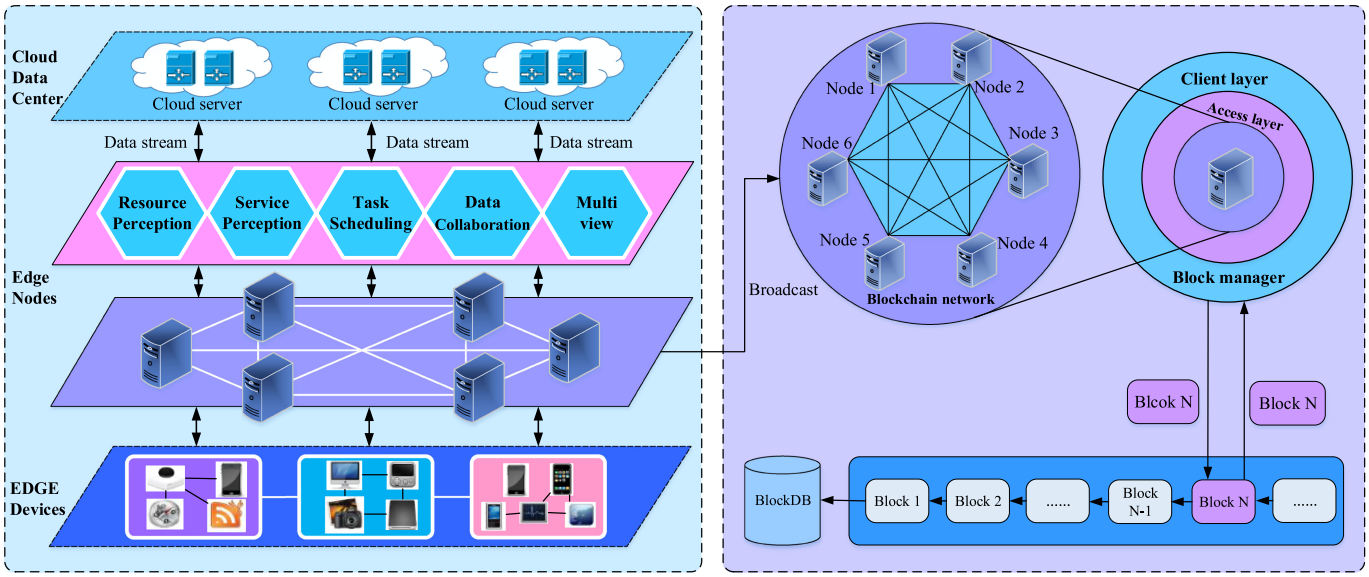

Fig. 1. BlockTDM trusted data management architecture.

And moreover, then $K_{AP} = K_{PA}$ is the session key, in fact, it is easily to see that there exists

$$K_{AP} = r_A r_P G = r_P r_A G = K_{PA}. \tag{10}$$

As for the application of the content protection, we can deduce the session key as follows:

$$K = H(x_{K_{AP}}, y_{K_{AP}}). \tag{11}$$

2) BlockTDM Security Enhanced Consensus: Usually, the blockchain works in a mechanism called consensus that collaborates the distributed participants as peers and decides how the transactions were confirmed and produced by the appropriate participants. In general, the permissioned blockchain uses RAFT or PBFT consensus to construct the blockchain system with authentication, authorization, and access control mechanism, such as Quorum using RAFT consensus, while Hyperledger Fabric uses PBFT for permissioned blockchain data management. The BlockTDM scheme is built based on the Hyperledger fabric for trusted transactional data management.

In fact, as we know the famous PBFT consensus is proposed by Miguel Castro and Barbara Liskovin 1999, which is based on the Byzantine Generals Problem proposed by Leslie Lamport *et al.* In 1982, which is a very important technology in distributed system.

The PBFT consensus solved a state machine replication problem with liveness & safety, the main contribution is the system can still work when there exists (n-1)/3 malefactor, where n is the maximum participants in the distributed system.

As for the fault tolerance, we can define a relationship of fault nodes numbers (f) with the total nodes (n) as follows:

$$n \geq \begin{cases} 2f+1, & \text{RAFT}, \ \text{mem}_f \in F \\ 3f+1, & \text{PBFT}, \ \text{mem}_f \in F \cup M \end{cases}. \tag{12}$$

where

$$F = \{f_1, f_2, \ldots, f_t\} \text{ and } M = \{m_1, m_2, \ldots, m_k\}.$$

In which F and M stand for fault participants set and malefactor set, respectively. Generally, in the RAFT consensus, there only exits fault participant, while in PBFT algorithm there may exist fault or malefactor participants.

In some special scene such as forensics or high-level security data management, it needs multiple part witness of the data security guarantee. To solve the problem, in BlockTDM scheme, we proposed multisignature [18]–[21] security-enhanced PBFT consensus for the high-level security of trusted data management, which is like the multisignature in Bitcoin system. When the client user commits a high-level security proposal to the endorsement peers, the system starts up the multisignature.

As is known to all, there are two kinds of multisignature that includes sequential multisignature and broadcast multisignature [18]–[21]. Considering that the blockchain is a P2P network supporting distributed self-organized system, thus in our proposed BlockTDM consensus, we proposed broadcast multisignature for security-enhanced consensus supporting.

a) Multisignature-based enhanced consensus: The client user U commits the proposal message m to the endorsing peer $P_i (1 \leq i \leq n)$ and the management peer P_M, where Q_i=d_i mod n, as for the ECDSA algorithm see [22]–[25].

Step 1: P_i randomly selects $k_i \in [1, n-1]$, and computes

$$R_i = k_i G = (x_i, y_i) \tag{13}$$

$$r_i = x_i \bmod n. \tag{14}$$

And then sends r_i to the management peer P_M.

Step 2: When the management peer P_M receives $r_i (i = 1, 2, \ldots, n)$, then computes

$$R_i = r_i G \tag{15}$$

$$R = \sum_{i=1}^{n} R_i. \tag{16}$$

And send *R*' to each endorsing peer $P_i (1 \le i \le n)$.

Step 3: As for the message m, $P_i (1 \le i \le n)$ computes

$$e = h(m) \tag{17}$$

$$s_i = \mathrm{ed}_i + r_i \bmod n. \tag{18}$$

then s_i is partial multisignature of the endorsing peer P_i of message *m*, that is: $\mathrm{sig}(m) = (r_i, s_i)$ and *Pi* sends the sig(*m*) to management peer P_M.

Step 4: After receives $\mathrm{sig}(m) = (r_i, s_i)$, then P_M computes

$$s = \sum_{i=1}^{n} s_i \tag{19}$$

$$Q = \sum_{i=1}^{n} Q_i. \tag{20}$$

b) Multisignature verification: The management peer P_M verifies whether following condition is satisfied or not:

$$R = sG - eQ. \tag{21}$$

In fact, according to the above signature procedure, we can easily verify the (21) as follows.

For a clear proof, let us define the right part of the above equation as *X*, then we have

$$\begin{aligned} X &= sG - eQ \\ &= \sum_{i=1}^{n} (\mathrm{ed}_i + r_i \bmod n)G - eQ \\ &= \sum_{i=1}^{n} \mathrm{ed}_i G + \sum_{i=1}^{n} r_i G \bmod n - e\sum_{i=1}^{n} Q_i \\ &= e\sum_{i=1}^{n} d_i G + \sum_{i=1}^{n} r_i G \bmod n - e\sum_{i=1}^{n} Q_i \\ &= e\sum_{i=1}^{n} Q_i + \sum_{i=1}^{n} r_i G \bmod n - e\sum_{i=1}^{n} Q_i \\ &= \sum_{i=1}^{n} r_i G \bmod n \\ &= \sum_{i=1}^{n} R_i \bmod n \\ &= R. \end{aligned} \tag{22}$$

That $X = R$, that is

$$R = sG - eQ. \tag{23}$$

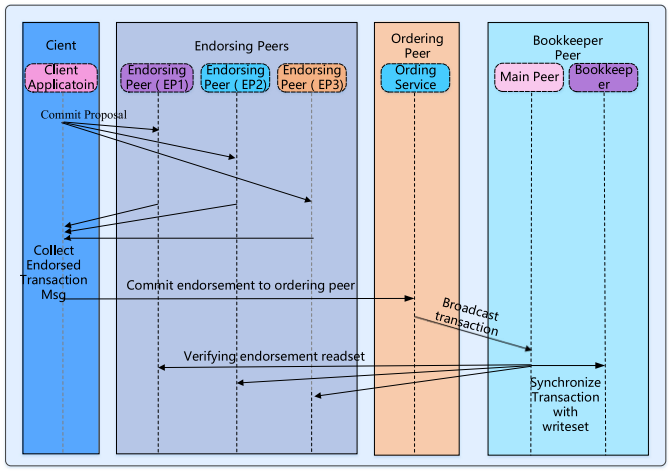

Fig. 2. BlockTDM block and transactions processing.

Hence, the multisignature is verified as true. The multisignature-based security-enhanced consensus is an optional selection (not a must) for the BlockTDM scheme.

3) BlockTDM Smart Contract: As for the BlockTDM scheme, we used Hyperledger chaincode as a smart contract, which is a kind of program code written in go or node.js language. The smart contract usually runs in a secured container that connecting to the endorsing peers, which initializes the blockchain status to process transactions submitted from client applications. A chaincode can be invoked to update or query the ledger in a proposed transaction, and chaincode can interoperates each other.

4) BlockTDM Block and Transaction Data Management: In our proposed BlockTDM scheme, we used Hyperledger fabric as the basic permissioned blockchain platform, where the consensus is collaborated among endorsement peers, ordering peers, and bookkeeper peer. The BlockTDM execution procedure is based on the BPFT consensus and follows the specific procedure that is defined in the Hyperledger fabric; the BlockTDM block, and transactions processing is described in Fig. 2.

1) The client submits a proposal message and sends it to endorsement peers.
2) The endorsing peers simulate a transaction.
3) The submitting client collects and broadcasts endorsing result to ordering service peer.
4) The ordering peer delivers transactions to the bookkeeper.

In which the client application commits PROPOSE message propMsg:

$$\mathrm{propMsg} = < \mathrm{PROPOSE}, \mathrm{tx}, [\mathrm{anchor}] >$$

to the endorsing peers for possible endorsing, where

$$\mathrm{tx} = < \mathrm{clientID}, \mathrm{chaincodeID}, txPayload, \mathrm{timestamp}, \mathrm{clientSig} >$$

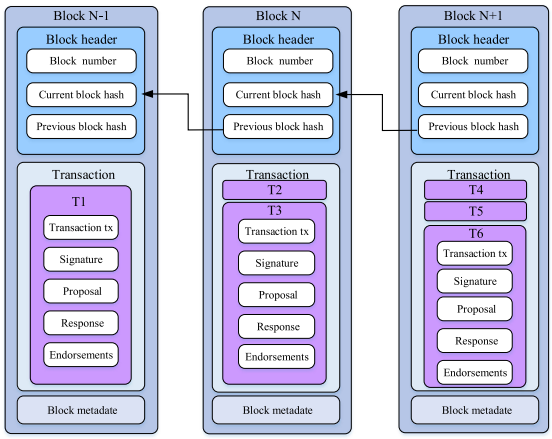

Fig. 3. BlockTDM block data structure.

and txPayload maybe the invoke transaction or deploy transaction

$$\text{txPayload} = < \text{operation}, \text{metadata} > \tag{24}$$

$$\text{txPayload} = < \text{source}, \text{metadata}, \text{policies} > \tag{25}$$

anchor contains read version dependencies, key-version pairs.

When an event deliver $(\text{seqno}, \text{prevhash}, \text{blob})$ occurs and a peer has applied all state updates for blobs with a sequence number lower than seqno.

The BlockTDM block data structure is described in Fig. 3, in which the block header includes number, current blockhash, previous blockhash, and usually a block includes one or more transactions in the block body, and the transactions are usually signed by digital signature algorithm and organized by Merkle trees.

5) BlockTDM Critical Data Protection: In the BlockTDM scheme, we proposed efficient approaches to ensure the data security and content protection: 1) Matrix-based multichannel data management based on block data segment and isolation. 2) User-defined payload data encryption and conditional access of high-level security data protection.

a) BlockTDM multichannel secure data management: The BlockTDM scheme is a permissioned blockchain and built up based on Hyperledger fabric, the matrix-based multichannel architecture is suitable for protecting the data that already stored in the blockchain system, the block data in each channel is visible to the user in the same channel, but invisible for other user outside the channel. The matrix-based multichannel model is essentially a data segment and isolation method to protect the block data being accessed in unauthorized channel. The matrix-based multichannel model is described as the channel matrix $C_{m\times n}$

$$C_{m\times n} = \begin{pmatrix} c_{11}, c_{12}, c_{13}, \ldots, c_{1i} \\ c_{21}, c_{22}, c_{23}, \ldots, c_{2j} \\ c_{31}, c_{32}, c_{33}, \ldots, c_{3k} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ c_{n1}, c_{n2}, c_{n3}, \ldots, c_{nm} \end{pmatrix} \tag{26}$$

where each element has different privilege for each user, that is

$$\text{priv}(c_{ij}) \neq \text{priv}(c_{\text{km}}), i \neq k \tag{27}$$

where c_{ij} and c_{km} are in different channels; thus different channel has different privilege of transaction access, in which the special sensitive or critical important data, the data content is isolated and is invisible in different channel.

b) BlockTDM user-defined data encryption: Although a multichannel method can isolate or segregate block data in different channels, blocks in the same channel are visible for all the users belong to the channel, which may explore user's data security or privacy. As for this problem, we proposed user-defined data encryption to protect sensitive and critical data or privacy.

Before the sensitive data are stored in a blockchain platform, we can first selectively encrypt user-side data or data segment into cipher, and then store the sensitive cipher data in a blockchain system; thus, it can prevent plaintext data from being explored to the user in the same channel. In the BlockTDM scheme, the three roles include the following:

1) the blockchain client user;
2) blockchain administration node;
3) blockchain audit node.

C. Sensitive Data Encryption Key Building Up

Step 1: The blockchain administration node, audit node, and client user compute its subkey according to its identity and control word as follows:

$$K_A = \text{HMAC}\left(\text{AdminID}, \text{AdminCW}\right) \tag{28}$$

$$K_D = \text{HMAC}(\text{AuditID}, \text{AuditCW}) \tag{29}$$

$$K_U = \text{HMAC}(\text{UserID}, \text{UserCW}). \tag{30}$$

Especially the control word is not the same as the login password, which can be modified frequently for secure usage.

Step 2: The BlockTDM management node then computes sensitive data encryption key K by secure hash function $H(.)$ as follows:

$$K = H\left(K_A \oplus K_D \oplus K_U\right). \tag{31}$$

Step 3: For the security, the BlockTDM management node computes the hash value of K:

$$H_K = H\left(K\right). \tag{32}$$

Then the BlockTDM management node signs K, the signature procedure is described as follows.

The BlockTDM management node randomly selects k and computes

$$kG = (x, y) \tag{33}$$

$$r = x \bmod n \tag{34}$$

$$e = h(K) \tag{35}$$

and computes

$$s = rk + \mathrm{ed} \bmod n \tag{36}$$

$$\mathrm{sig} = (r, s). \tag{37}$$

Then (r, s) is the signature of K.

Step 4: The BlockTDM management node stores the K-related hash value and its signature as follows:

$$K_{DB} = \mathrm{Store}(H(K), \mathrm{sig}(K)). \tag{38}$$

D. Sensitive Data Encryption and Decryption

Step 1: The BlockTDM commits its client user identity and control word to the management node, the management node then computes the user's key as follows:

$$K'_U = \mathrm{HMAC}(\mathrm{UserID}, \mathrm{UserCW}). \tag{39}$$

And then computes the K' as the above step to deduce the encryption key as follows:

$$K' = H\left(K_A \oplus K_D \oplus K_U\right). \tag{40}$$

Step 2: The management node computes

$$H_{K'} = H\left(K'\right). \tag{41}$$

Step 3: To verify, the management node verifies the signature of the stored $H(K)$ to check whether the stored key data are tampered or not

$$b\,\mathrm{Result} = \mathrm{Veri}(\mathrm{Sig}(L)). \tag{42}$$

Step 4: If bResult = True, then it manifests that the H(K) is not tampered and then compares $H_{K'} = H_K$ is true or not.

Step 5: If then it shows that the current user is the correct user registered in the blockchain system, otherwise the user is not valid and denies the later operation.

Step 6: Then the management node encrypts the payload data segment $M_i (1 \le i \le n_0)$ into cipher mode and store in the blockchain system

$$C_{\mathrm{payload}} = E_K(M_1, M_2, \ldots, M_{n_0}). \tag{43}$$

When necessary, the client user can request from the client side and decrypt the cipher payload data segment dynamically in the blockchain side. The decryption procedure is the same verification as the encryption procedure similarity as follows:

$$M = D_K(E_K(M_1|M_2|\ldots M_{n_0})). \tag{44}$$

1) BlockTDM Nodes and Peers Management

The BlockTDM blockchain computing paradigm is based on physical computing resource to provide blockchain management and block data service, which can be formally defined as a matrix of nodes, organizations, and peers. The logic relationship is described as follows:

1) The peer P provides blockchain-related operation and service, while the organizations O provide transaction and peers management, while nodes provide basic network infrastructure service.
2) According to the blockchain logical relationship, P is the peers matrix, O is the organizations matrix, and N is the nodes matrix, the resources have the following logic relationship:

$$P \subset O \subset N. \tag{45}$$

3) The P, O, N matrix is defined as follows:

$$N_{ij} = R_{ij} = \begin{pmatrix} r_{11}, r_{12}, \ldots, r_{1j} \\ r_{21}, r_{22}, \ldots, r_{2j} \\ \ldots\ldots\ldots\ldots\ldots \\ r_{i1}, r_{i2}, \ldots, r_{ij} \end{pmatrix}$$

$$R_i = \begin{cases} r_{i1} = IP, r_{i2} = \mathrm{port} \\ r_{i3} = \mathrm{MAC}, r_{i4} = \mathrm{RAM} \\ r_{i5} = \mathrm{Storage} \\ r_{i6} = \mathrm{Network\ Service} \end{cases} \tag{46}$$

$$O_{mn} = T_{mn} = \begin{pmatrix} \mathrm{t}_{11}, t_{12}, \ldots, t_{1n} \\ t_{21}, t_{22}, \ldots, t_{2n} \\ \ldots\ldots\ldots\ldots\ldots \\ t_{m1}, t_{m2}, \ldots, t_{mn} \end{pmatrix}$$

$$T_j = \begin{cases} \mathrm{t}_{j1} = ep_{j1}, t_{j2} = ep_{j2} \\ t_{i3} = ep_{j3}, t_{j4} = ep_{j4} \\ t_{j5} = op_{\mathrm{j1}}, t_{j6} = op_{\mathrm{j2}} \\ t_{j7} = \mathrm{sp}_{j1}, t_{j8} = \mathrm{sp}_{j2} \end{cases} \tag{47}$$

$$P_{kl} = S_{kl} = \begin{pmatrix} s_{11}, s_{12}, \ldots, s_{1l} \\ s_{21}, s_{22}, \ldots, s_{2l} \\ \ldots\ldots\ldots\ldots\ldots \\ s_{k1}, s_{k2}, \ldots, s_{kl} \end{pmatrix}$$

$$S_k = \begin{cases} s_{k1} = \mathrm{Sign/Verify} \\ s_{k2} = \mathrm{CreatBlock} \\ s_{k3} = \mathrm{CreatTrans} \\ s_{k4} = \mathrm{QueryBlock} \\ s_{k5} = \mathrm{QueryTrans} \end{cases}. \tag{48}$$

III. Implementation of BlockTDM Scheme

As for the above designing of the BlockTDM scheme, based on Hyperledger Fabric 1.3, we implemented the BlockTDM as a whole blockchain platform for trusted data management of critical and sensitive service-related data distribution and storage in edge computing. The BlockTDM implementation and development platform parameters are listed as the table, in which we developed different smart contracts for different kinds of data, such as documents, multimedia data are developed with different chaincodes to support the specific block data management. In the BlockTDM, we deployed two nodes, each node installed docker as runtime environment, in each node we deployed two orgs and two peers, thus each block data can be

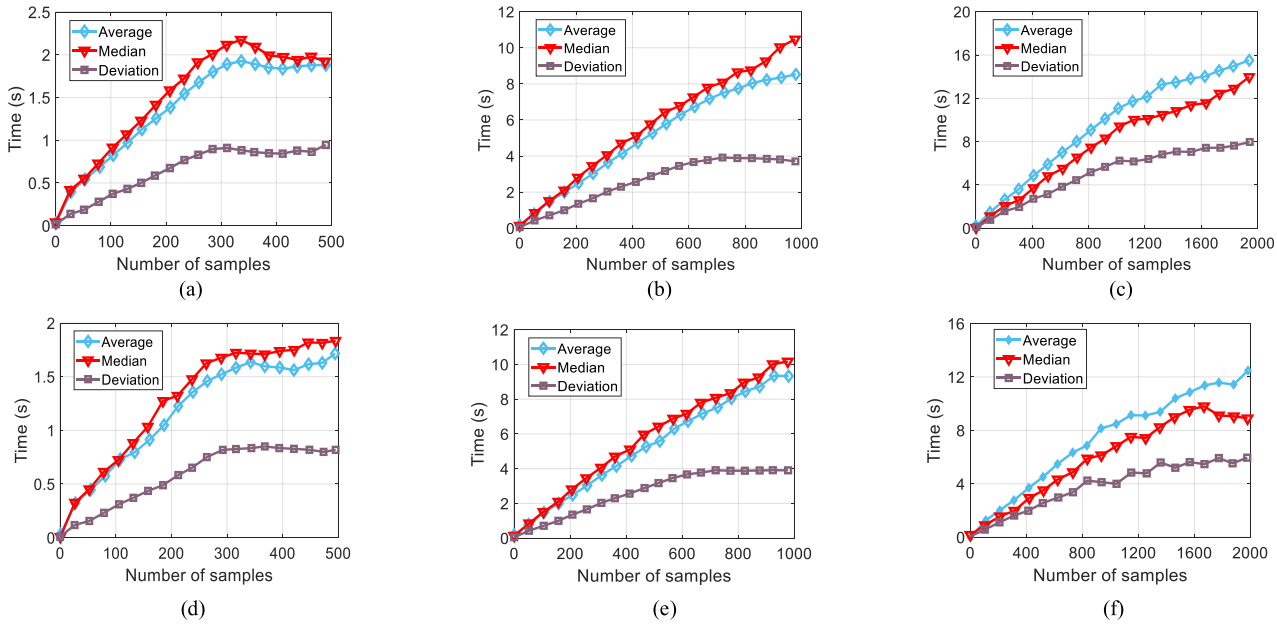

Fig. 4. Performance evaluation results of BlockTDM scheme.

stored in different peers, one of the block data instance is listed in Table I.

IV. Performance Evaluation of BlockTDM Scheme

The BlockTDM scheme evaluation is based on the Apache JMeter 4.0 in an automatic real-time online testing upon our implemented BlockTDM blockchain system, and for each block data operation evaluation, we evaluated the query operation by block_current_hashID, transaction txID in the BlockTDM system, we grouped the querying operation by block_current_hashID and txID into two turns; for each turn, we set the number of thread as 50, 100, 200, and for each thread we set the loop counts as 10, thus equivalently, for each turn we evaluated 500, 1500, 2000 virtual users query operations. The two groups query results are listed in Fig. 4 from (a)–(f).

Upon each query operation either by block_current_hashID or by txID, the response time is nearly at the same level, such as when the virtual users reach to 600 (total virtual user 1000), the query operation responsible time by block_current_hashID and by txID is, respectively, about 6.23 s and 6.18 s per ten turns, then the average response time is 0.623 s and 0.618 s per query. While in the 2000 total users group at the user 800, the response time reached 0.825 s and 0.791 s, respectively.

Comparing with typical blockchain technologies, Bitcoin, Litecoin, Dogecoin, Ethereum, Quorum, Bitshare, Hyperledger, EOS, Byteballet, Polkadot, Cosmos, etc., which are familiar to users, we gave a detailed comparison of blokchains with various factors such as consensus, blockchain type, complexity, maximum fault tolerance nodes, network assumption, data security, and privacy protection. The comparison is detailed in Table II, from which we can see that the proposed BlockTDM scheme is flexible, secure, and scalable for the trusted data management,

TABLE I
Block and Transaction Instance Created in the BlockTDM

Name	Data
Image Name	BUPT Campus
Image Location	Xitucheng Rrd 10#,Haidian,Beijing,China
Image Device	Huawei Mate 20 Pro; IMEI-861198049301112; WLAN-MAC-DC:16:B2:0A:C5:BA
Image Creation Time	2019-04-07 15:36:08
Image Source	BUPT-Campus
Image Description	BUPT-Campus-Gate, Beijing, China
BlockNumber	94
CurrentHash	a26d7ed9b5b7a5a47863bf2bc2279bf68bce978b9c1e8e5fcef2279b50e699fd
PreviousHash	f5d303520d42c06b7a8ab254a7ea352b902662276044ebd2c5dc24213fa436e8
Datahash	5d810ff51eac7563f9d167ec99d99d68082bb052b1c749ba79fb50c1f38c73ea
Number of tx	1
tx_id	1f5e69485b4f8061a934fdf776562b5cf5f45e92e7c29dec3daad395e517b0af
ProposalHash	eb12ddab9038e6d31381b0924be036af41dc5b20d5762d8f9832012bf22bcfab
Payload	{"value1":"BUPT Campus","value2":"Xitucheng Rrd 10#,Haidian,Beijing,China","value3":"Huawei Mate 20 Pro; IMEI-861198049301112; WLAN-MAC-DC:16:B2:0A:C5:BA","value4":"BUPT-Campus","value5":"BUPT-Campus-Gate, Beijing, China","value6":"2019-04-10 17:47:37","value7":"2899b39d241bf7ee314b5977e68983f24ba6949676df5920b31dfc457bf48583"}
Timestamp	Wed Apr 10 2019 17:47:40 GMT+0800 (CST)
Blockchain Peer,Org and Node	peer0.org0.CPSecChain.com peer1.org0.CPSecChain.com peer0.org1.CPSecChain.com peer1.org1.CPSecChain.com

TABLE II
BLOCKTDM SCHEME COMPARISON WITH RELATED WORK

Scheme &Consensus	Communication Complexity	Max faulty nodes	Network Assumption	Finality	Liveness (Availability)	Data security	Authentication	Authorization
BitCoin PoW	O(n)	2f+1≤n	Synchronous (10 minutes)	Probabilistic	Always available	weak (base58)	No	No
Ethereum PoW	O(n)	2f+1≤n	Synchronous (15 seconds)	Probabilistic	Always available	weak (Hex)	yes	no
Tendermint	$O(n^2)$	3f+1≤n	Partially synchronous	Deterministic	Not always	low	yes	no
Casper FFG	During PoW O(n), During BFT,$O(n^2)$	PoW,2f+1≤n , BFT,3f+1≤n	PoW, Partially Synchronous,BFT, Partially Synchronous	PoW, Probabilistic BFT, Deterministic	PoW, Always available; BFT,Not always	weak for BFT	yes	no
SBFT	0(1)	f+1=O(n)	Partially synchronous	Deterministic	Not always	low	yes	no
RAFT	O(n)	2f+1≤n	Partially synchronous	Deterministic	Not always	weak	yes	yes
PBFT hyperledger	$O(n^2)$	3f+1≤n	Partially synchronous	Deterministic	Not always	normal	yes	Yes channel
PBFT BlockTDM	$O(n^2)$	3f+1≤n	Partially synchronous	Deterministic	Not always	High User defined data encryption	yes	Yes Channel Access control

the BlockTDM can provide trust and credibility of data security with tamper-resistance and integrity assurance.

V. CONCLUSION

A blockchain-based trusted data management scheme (called BlockTDM) in edge computing was proposed in this article to solve the above problems, in which we proposed a flexible and configurable blockchain architecture that includes mutual authentication protocol, flexible consensus, smart contract, block and transaction data management, blockchain nodes management, and deployment. The BlockTDM scheme can support matrix-based multichannel data segment and isolation for sensitive or privacy data protection, and moreover, we designed user-defined sensitive data encryption before the transaction payload stores in blockchain system, and we implemented conditional access and decryption query of the protected blockchain data and transactions through smart contract. Analysis and evaluations manifest that the proposed BlockTDM scheme provides a general, flexible, and configurable blockchain-based paradigm for trusted data management with high credibility.

ACKNOWLEDGMENT

The authors are grateful to those anonymous reviewers for their careful reviewing of this article with detailed remarks and advice.

REFERENCES

[1] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.
[2] C. Esposito, A. Castiglione, F. Pop, and K.-K. Raymond Choo, "Challenges of connecting edge and cloud computing: A security and forensic perspective," *IEEE Cloud Comput.*, vol. 4, no. 2, pp. 13–17, Mar./Apr. 2017.
[3] W. Shi and S. Dustdar, "The promise of edge computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
[4] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.
[5] The Bitcoin Project, 2008. [Online]. Available: https://bitcoin.org
[6] The Ethereum Project, 2013. [Online]. Available: https://www.ethereum.org
[7] The Hyperledger Project, 2016. [Online]. Available: http://www.Hyperledger.org
[8] R. Henry, A. Herzberg, and A. Kate, "Blockchain access privacy: challenges and directions," *IEEE Secur. Privacy*, vol. 16, no. 4, pp. 38–45, Jul./Aug. 2018.
[9] T. Aste, P. Tasca, and T. Di Matteo, "Blockchain technologies: The foreseeable impact on society and industry," *Computer*, vol. 50, no. 9, pp. 18–28, 2017.
[10] T. T. Anh Dinh, R. Liu, and M. Zhang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018.
[11] K. Gai, K.-K. R. Choo, and L. Zhu. "Blockchain-enabled reengineering of cloud datacenters," *IEEE Cloud Comput.*, vol. 5, no. 6, pp. 21–25, Nov./Dec. 2018.
[12] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung, "Blockchain-based decentralized trust management in vehicular networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1495–1505, Apr. 2019.
[13] Q. Xu *et al.*, "Blockchain-based decentralized content trust for docker images," *Multimedia Tools Appl.*, vol. 77, no. 14, pp. 18223–18248, 2018.
[14] A. Anjum, M. Sporny, and A. Sill, "Blockchain standards for compliance and trust," *IEEE Cloud Comput.*, vol. 4, no. 4, pp. 84–90, Jul./Aug. 2017.
[15] M. Turkanović and M. Hölbl, "EduCTX: A blockchain-based higher education credit platform," *IEEE Access*, vol. 6, pp. 5112–5127, 2018.
[16] M. Zhaofeng, J. Ming, G. Hongmin, and W. Zhen, "Blockchain for digital rights management," *Future Gener. Comput. Syst.*, vol. 89, no. 12, pp. 746–764, 2018.
[17] S. Kiyomoto, M. S. Rahman, and A. Basu, "Blockchain-based anonymized dataset distribution platform," in *Proc. IEEE Int. Conf. Softw. Eng. Res.*, 2017, pp. 85–92.
[18] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary Version*, 1998.
[19] A. Boldyreva, "Threshold signatures, multisignaturesand blind signatures based on the gap-diffiehellman-group signature scheme," in *Public key Cryptography*. New York, NY, USA: Springer-Verlag, pp. 31–46, 2002.
[20] S. S. M. Chow, L. C. K. Hui, S. M. Yiu, and K. P. Chow, "Forward-secure multisignature and blind signature schemes," *Appl. Math. Comput.*, vol. 168, no. 2, pp. 895–908, 2005.
[21] C. Claude, J. Stanisław, K. Jihye, and T. Gene, "Secure acknowledgment aggregation and multisignatures with limited robustness," *Comput. Netw.*, vol. 50, no. 10, pp. 1639–1652, 2006.
[22] V. S. Miller, "Use of elliptic curve in cryptography," in *Proc. Conf. Theory Appl. Cryptograph. Tech.*, 1986, pp. 417–426.
[23] Standard Specifications for Public-Key Cryptography, IEEE. Standard P1363, 2000.
[24] Public Key Cryptography for the Finacial Service Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), ANSI X9.62, 1999.
[25] V. Dhillon, D. Metcalf, and M. Hooper, "The Hyperledger Project," 2019. [Online]. Available: http://www.Hyperledger.org
[26] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, Paper 30.

Ma Zhaofeng (M'17) received the Ph.D. degree from Xi'an Jiaotong University, Shaanxi Sheng, China, in 2014.

He engages in science research and education work in School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. His research interests include blockchain, mobile Internet security, and digital rights management.

Dr. Zhaofeng is an Association for Computer Machinery Member and CCF Member.

Haneef Khan received the B.Tech. degree in electronics and instrumentation from Uttar Pradesh Technical Univeristy, Lucknow, UP, India and the M.Tech. degree in VLSI Design from Maharishi Dayanand University, Rohtak, Haryana, India.

He is currently a Lecturer with the Department of Computer and Network Engineering, Jazan University, Kingdom of Saudi Arabic. He has more than eight years of teaching experience and his main research interests include blockchain and wireless networking communication.

Wang Xiaochang is currently working toward the master's degree in blockchain and security research work with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China, since 2018.

She had finished two blockchain projects and Dapps. Her research interests include blockchain, network security, and cryptography.

Gao Hongmin is currently working toward the Ph.D. degree in blockchain and security research work with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China, since 2017.

He had finished three blockchain projects and applications. His research interests include blockchain, network security, cryptography, and secure multi-part computing technology.

Deepak Kumar Jain received the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is currently an Assistant Professor with the Department of Automation, Institute of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include block chain and deep-learning-based network security.

Wang Zhen is currently working toward the Ph.D. degree in blockchain and security research work with the School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China, since 2014.

His research interests include blockchain, mobile network security, cryptography protocol, and algorithm designing.