Journal of Network Intelligence ©2023 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **8**, Number **2**, May **2023**

Iot Data Security Authentication and Key Negotiation Scheme Based on Edge Computing and Blockchain

Sheng Wang*

Information Technology Institute
Anhui Vocational College of Defense Technology
Lu an, 237011, China
ws@acdt.edu.cn

Xiangzhen Zhou

Faculty Information Science and Technology
National University of Malaysia
Selangor, 43600, Malaysia
100618@shengda.edu.cn

*Corresponding author: Sheng Wang

Received December 22, 2022, revised February 27, 2023, accepted April 11, 2023.

Abstract. *With the development of Internet of Things (IoT) technology, the number of mobile smart devices has proliferated and brought a lot of convenience to people's daily life. However, conventional mobile cloud computing architectures are not well suited for low latency and mobility requirements due to the long distance between the central cloud and portable end devices. To solve these problems, mobile edge computing has emerged. However, large-scale edge computing also has many security risks, among which trusted authentication of mobile smart devices is the basis for solving related security problems. The emergence of blockchain technology provides a new way of thinking for solving the above problems. In response to the above problems, this paper proposes an IoT data security authentication and key negotiation scheme based on edge computing and blockchain. First, the constituent elements of the proposed scheme and the deployment of blockchain network in mobile edge environment are described in detail. Second, a short signature-based edge data key negotiation protocol is proposed, which is mostly employed for bi-directional authentication between devices as well as between people and devices. To prevent crucial information from being lost or modified, the public identity of devices and users is recorded in the blockchain. The experimental results show that the proposed scheme can achieve secure authentication between IoT nodes and nodes without relying on the central node, and solves the problems of over-centralized authentication and public key replacement. The proposed scheme has lower service latency compared to various other schemes.*

Keywords: Internet of things; Edge computing; Blockchain; Secure authentication; Key negotiation

1. **Introduction.** With the continuous upgrade of IoT end devices and the development of new network technologies, IoT is becoming more and more widely used in various fields. According to Juniper Research, the number of connected IoT devices worldwide will grow from 21 billion in 2018 to 50 billion in 2023 [1,2,3]. The interconnection of large-scale end devices has brought great convenience to people, such as smart home, smart

transportation, smart payment and smart grid. However, with this comes various security issues. IoT devices are capable of generating, processing and exchanging large amounts of critical data and privacy-sensitive information that are relevant to people's lives. However, a large number of IoT devices are in an unsupervised state and have limited resources and computing power to use some traditional protection means and security techniques. Malicious attackers can steal users' private information through IoT devices or listen to transmission channels, thus posing a great threat to people's life, property, and privacy security [4,5]. Therefore, trusted authentication of IoT devices and users is particularly important in order to promote the security development of IoT.

Most of the current IoT systems adopt a centralized architecture with a central server to store and process the data of end devices. However, with the increasing number of IoT terminal devices, the scale of data and information is getting bigger and bigger, which leads to the burden of the central server and makes its processing speed of data slower and slower. On the other hand, the access of massive devices also brings the pressure of network expansion and increases the high cost of infrastructure construction and maintenance. If the central server fails or is compromised by malicious attackers, it will cause incalculable damage to the whole system [6,7]. To solve these problems, mobile edge computing has emerged. By deploying edge nodes near mobile terminals, mobile edge computing can transfer some tasks according to the main cloud server to the platform's outer edge. However, this pre-downloaded data may be corrupted beforehand. The vulnerability of highly virtualized edge computing systems is greater than that of centralized cloud computing infrastructures, which can seriously threaten the security of edge data. In order to guarantee information security in an edge cloud computing, it is necessary to ensure the integrity of data on both the remote cloud and edge nodes. Therefore, data integrity verification is important for the security of mobile edge computing.

Device authentication schemes for IoT in edge computing environments mainly use public-key cryptosystems instead of symmetric keys, and thus need to rely on trusted third parties. It is vulnerable to internal manipulation attacks and single point of failure. On the other hand, However, the current user authentication methods used by Wireless Sensor Network (WSN), the IoT's perception layer, are insufficient for sizable IoT systems. All authentication methods used in multi-gateway situations must rely on reliable gateway nodes or third parties [8,9,10]. The interaction process involved in the authentication process necessitates the transfer of information between numerous gateways.

Therefore, in order to solve the above problems, blockchain technology, as an emerging technology, has gradually attracted the attention of relevant researchers [11,12]. Blockchain is a distributed ledger with self-maintenance function and automatic synchronization function, which consists of a sequence of data block groups. These informational units are linked together in a chain structure using dates. Each block contains valid transactions that have been verified by nodes over a period of time. The blockchain ledger is jointly maintained by all nodes in the blockchain network. Each node achieves the consistency of the data stored in the ledger through a consensus algorithm. At the same time, the combination of consensus mechanisms and cryptographic techniques makes the data unmodifiable. Blockchain technology allows all members to back up the ledger containing all transaction data and update it when new transactions appear to maintain integrity. Blockchain technology, as a distributed ledger technology, completely eliminates the dependence of the system on a central authority [13,14,15]. Applying blockchain technology to IoT makes the centralized network structure marginalized and can effectively solve the single point of failure problem and the performance bottleneck of the central node. Applying blockchain technology to the IoT can enhance the availability of the system and greatly improve the security of the IoT system. At the same time, the multi-party maintenance, irreversible, open, transparent and traceable features of blockchain ledger also provide a feasible solution for the identity authentication of IoT devices.

Therefore, in this paper, a decentralized IoT scheme is constructed based on edge computing technology and blockchain technology. Then, based on this, a short signature-based edge data key negotiation protocol is proposed respectively, which can accomplish the corresponding secure authentication without the participation of third-party organizations. This scheme has lower latency and can effectively avoid the problem of missing queried databases in the edge cache. Finally, the proposed distributed IoT security authentication scheme is tested by embedded devices and existing blockchain platforms.

1.1. **Related Work.** A unique use of mobile cloud computing, mobile edge computing can offer services with lower latency than mobile cloud computing. Edge cloud platforms (Cloudlet) [16] can move computational and storage resources close to the mobile device, but usually there is little guarantee of quality of service (QoS) and user experience (QoE) for mobile devices.

As yet another widely used idea in edge computing, fog computing aims to handle applications on billions of smart devices at the edge of the network. However, mobile networks do not incorporate the cloudlet and fog computing processing capability. QoS and Qo E are difficult to be satisfied especially if mobile users often switch among several locations. Due to the ultra-low service latency, mobile edge computing has penetrated many aspects of people's daily life [17,18], like as medical treatment, education through tutoring, and government services. Recently, security concerns in mobile edge computing have attracted a lot of attention. Mao et al. [19] proposed a mobile edge computing scheme suitable for data security verification.

Blockchain technology, as a tamper-evident distributed ledger technology, enables privacy protection and sharing while achieving data traceability and verifiability. Currently, more and more researchers at home and abroad are applying blockchain technology to IoT security authentication. Rajakumari and Parwekar [20] proposed a blockchain-based security model and protocol that can ensure the validity and integrity of WSN cryptographically authenticated data. Goyat et al. [21] proposed a blockchain-based WSN security authentication mechanism that enables sensor nodes to securely move from one cluster to another. Revanesh and Sridhar [22] advocated using distributed systems and blockchain and smart contract technology for IoT device authentication. The devices inside the area can be validated for mutual identification by the distributed system, which can also create a secure virtual region. A blockchain-based adaptive authentication and authorisation strategy for IoT gateways was presented by Casado-Vara and Corchado [23], which primarily authenticates and approves devices and may add IoT devices without any physical interaction. In order to increase the anonymity of blockchain, Guerrero- Sanchez et al. [24] integrated blockchain technology into intelligence IoT system and developed a privacy-preserving strategy. Feng et al. [25] designed a blockchain-based distributed trust model for IoT with a built-in reputation mechanism. This model is able to achieve end-to-end trust among IoT devices without experiencing traditional blockchain latency and without relying on any common root of trust.

However, none of these security authentication schemes mentioned above can be used with IoT gadgets in a mobile edge computing setting. This is because the untrustworthy behavior of third parties cannot be avoided in the traditional data integrity verification methods, which implies that there is no guarantee of data security. While this is going on, the present solutions for solving the data integrity issue in a mobile edge computing environment fall short of the requirements for extremely low latency.

1.2. **Motivation and contribution.** To solve this problem, this paper proposes a secure authentication and key negotiation scheme for IoT data based on edge computing and blockchain based on the theoretical foundations related to blockchain, short signatures, and edge cache. A short signature-based edge data key negotiation protocol is proposed, which has lower latency and can effectively avoid the problem of missing queried databases in the edge cache. First, the constituent elements of the proposed scheme and the deployment of blockchain networks in mobile edge environments are described in detail. Second, a brief edge data key negotiation protocol based on signatures is put forth. This protocol is primarily utilized for bi-directional identity authentication between devices, devices, and users. To prevent important data from being lost or altered, the blockchain stores the public identifying information of devices and users.

The main innovations and contributions of this study are shown below.

(1) To address the problem of data integrity assessment forthe mobile edge computing environment, combine blockchain technology and propose an IoT data security authentication and key negotiation scheme based on edge computing and blockchain, which effectively avoids the problem of untrustworthiness of third parties in the traditional authentication process.

(2) A short signature-based edge data key negotiation protocol is proposed which considers three cases, i.e., unilateral, multilateral, and multilateral combined with cloud. This scheme has lower latency and can effectively avoid the problem of missing queried databases in the edge cache.

2. The fundamentals of mobile edge computing and blockchain.

2.1. Architecture of Mobile Edge Computing.

Mobile edge computing. as a latest computing paradigm can solve the problem of high service latency in mobile cloud computing environment. In the mobile edge computing scenario, cloud services are deployed on edge servers close to mobile terminals. Mobile edge computing is a "hardware + software" system that accelerates the download speed of application data in the network with ultra-low latency, ultra-high bandwidth, and high real-time characteristics to provide a better experience for mobile device users in the coverage area.

The classical system architecture of mobile edge computing is shown in Figure 1, which contains a total of three layers, from top to bottom, the cloud server layer, the edge server layer and the mobile terminal layer, respectively. According to the European Telecommunications Standards Institute (ETSI), the mobile edge computing system can be divided into three layers: system-level, host-level, and network-level. System-level management is the core component of the mobile edge computing system and is responsible for the management of available resources, lifecycle and other tasks. Host-level management is mainly responsible for mobile edge platform management and virtualized infrastructure management. In cloud computing technology, other hardware and software resources such as CPU, memory, etc. are provided to the users of the network as shared services for a fee. The size and capacity of these resources can be increased or decreased according to the needs of the client. With Network Functions Virtualization (NFV) technology, the nodes are no longer limited by the hardware architecture. The entire system is virtualized into blocks according to functional categories. Each virtualized network module may consist of one or more virtual machines running different software and processes. Common examples of network function virtualization applications include: intrusion detection devices, firewalls, load balancers, and WAN gas pedals.

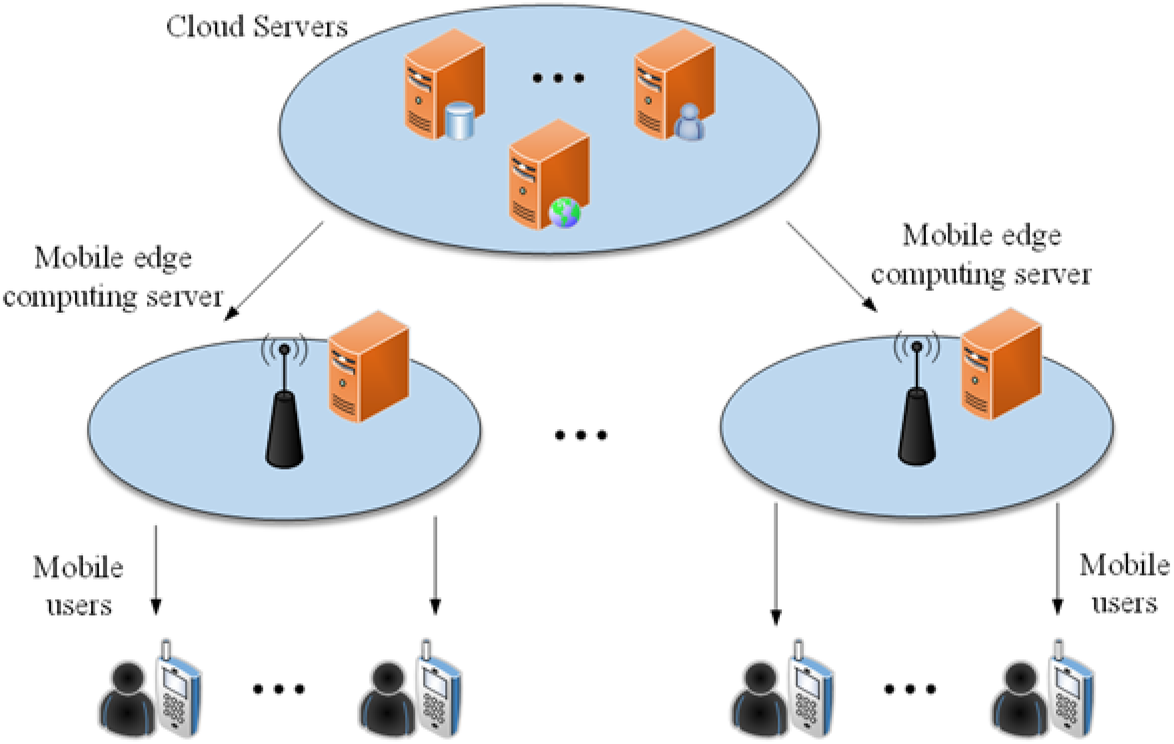

Figure 1. Architecture for Mobile Edge Computing.

2.2. **Blockchain Technology.** Blockchain technology as a tamper-evident distributed ledger technology enables privacy protection and sharing while achieving data traceability and verifiability.

Blockchain is essentially a distributed database, also called a distributed ledger, that is maintained and continuously grown by multiple parties. Blockchain simplifies the account reconciliation process through cryptography and distributed messaging protocols, and maintains large amounts of data through decentralization. The data in a distributed ledger is maintained by all nodes together. It is only possible to add records in the distributed ledger, but records that have already occurred cannot be changed. Unlike traditional centralized bookkeeping, a blockchain system can reach consensus without centralized control, while cryptography ensures that every transaction is non-repudiation and protects user information and transaction records from leakage as much as possible. A blockchain is a chain-like data structure consisting of multiple blocks. A block header and a block body make up single block. The prior block's hash value is contained in the block header, that is also utilized to link to it. The block body records the transactions that have been confirmed.

All transactions in the blockchain are summarized using Merkle trees, where the leaf nodes represent the hash values of the transactions. The introduction of Merkle trees effectively ensures the tamper-evidence of the blockchain, since any change in transaction data will cause the value of the root node to change, resulting in a change in the hash value of the entire block.

3. The proposed IoT data security authentication and key negotiation scheme.

3.1. **Problems with traditional edge data integrity.** Caching techniques can back up the more frequently used data and facilitate the speed of accessing them later. Remote calls in distributed systems also consume a lot of performance because of the network overhead, which can lead to longer overall response times, using caching techniques can greatly improve device performance and reduce many unnecessary overheads. When there is an increase in user requests, the pressure on the database will increase significantly and caching techniques can effectively reduce the storage pressure on the database. The traditional cloud storage model has better physical security and lower cost for long-term storage, but it is difficult for users to get the requested data with ultra-low latency, especially for bulky data such as videos. Although edge caching has great advantages in terms of cost and latency, it also has certain security drawbacks. Therefore, effective data integrity verification of edge caches is particularly important and is an important guarantee for users to use cached data safely in mobile edge computing environments.
Through the investigation of existing work, we found the following two main issues.
(1) The third-party audits introduced in existing data integrity verification schemes for mobile edge computing are not necessarily fully trusted. Traditional edge data integrity verification frameworks cannot guarantee that the verification results returned are accurate, so the security of user data is still under serious threat.
(2) The existing blockchain-based mobile edge computing architecture is not comprehensive enough to achieve trusted data integrity verification services well.

In order to solve the above problems, this paper proposes an IoT data security authentication and key negotiation scheme. In the mobile edge computing environment, smart contracts are used instead of third-party auditing for integrity verification services, thus ensuring the trustworthiness of the verification results.

3.2. **Security authentication scheme design.** To address the untrustworthiness of third-party services in mobile edge computing and combine the decentralized features of blockchain technology, this paper proposes a data security authentication scheme, as shown in Figure 2. (1) User: The one who first demanded data integrity checking. Geographical locations can vary between distinct customers. The user communicates with the central cloud server. The user first uploads the data files to the central cloud server. The edge nodes pre-download some of the data in advance to provide lower latency services to the users. The user typically makes an edge node data security validation request and gets the confirmation feedback signal from the smart contract to make certain that the saved data is not altered.

Edge nodes: Servers placed at the cloud's perimeter. In order to deliver ultra-low delay services to consumers inside the coverage region, all of these nodes are managed by an endpoint at the base station. Each edge node only offers a small amount of storage space for data because of cost and capacity limitations. The whole user data file cannot be downloaded in advance by edge nodes. The edge node locates the matching data block inside this previously downloaded data after receiving a security authentication request from a user, then gathers the created evidence to transmit to the smart contract during validation.
(2) Cloud Services Provider (CSP): Each CSP has robust computational power and ample storage capacity in order to deliver complete services quickly. When using data storage services, local storage capacity might be significantly reduced, especially for lightweight edge devices with little room for storage. In the system suggested in this work, CSPs generate matching proofs for lacking data blocks on edge devices as well as large-scale files of information for users.
(3) Smart Contract: Executor of the data integrity check. The primary functions of smart

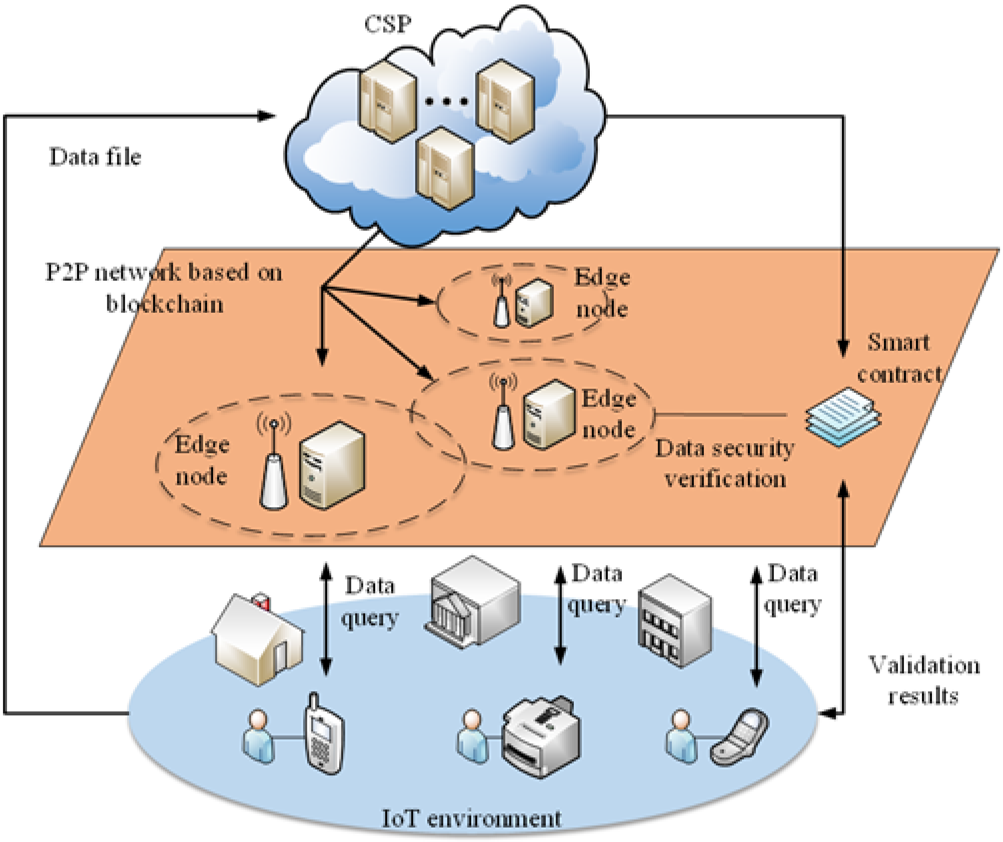

Figure 2. IoT Data Security Certification Solution.

contracts are to verify the proofs transmitted by edge devices or CSPs and to provide the user with the verification findings. Smart contracts have some computational and storage capabilities and can execute the functions written at development time. Thus, the smart contract in the framework can be considered as a database that stores the user's primary data files and can also perform validation operations efficiently.

In this study, it is expected that users would prefer sending security authentication queries to the edge devices rather than the far-off central cloud so as to obtain reduced latency. The pre-downloaded data blocks will be used by the requested edge node to compute the proof. If the user requests a data block that is not present on the current end device, the edge node will try to get the missing data from a nearby edge node.

When the blockchain is first initialized to create key pairs, mobile end users must be connected to the network. The utilization of smart contracts and storage services requires payment from mobile end users. In the blockchain network, edge nodes have the ability to take on the role of miner nodes and are therefore qualified to offer services and collect the associated benefits from mining. Mobile end-users require the use of data stored in the edge servers and pay accordingly for this.

3.3. **Secure communication framework.** To ensure that the nodes in the above model can communicate securely in the blockchain as well as in the P2P network, this paper proposes a secure communication framework for the issue with mobile edge computing's data security validation. In this framework, mobile terminals need to generate private keys and broadcast their public keys.

Firstly, the registration phase is divided into two main parts, namely the registration of the sensor nodes and the registration of the users. The specific registration process is

as follows.
The gateway node first selects a unique identity SIDj for the sensor node and calculates a shared key for both. The gateway node then sends (SID_j, K_j) to the sensor node over a secure channel.

$$K_j = h\left(SID_j \parallel \alpha\right) \tag{1}$$

where α is the value of the key chosen for the gateway node.
The sensor node generates a random number r_j , and calculates the secure transaction information.

$$S_1 = h\left(SID_j \parallel r_j\right), S_2 = h\left(SID_j \parallel CLK_j \parallel FSS_j\right) \tag{2}$$

$$S = h\left(S_1 \parallel S_2\right) \tag{3}$$

where CLK_j denotes the CPU clock frequency sum of sensor node j, FSS_j denotes the free storage space of sensor node j, and $h()$ denotes the single hash function. The gateway node initiates a registration transaction to the blockchain network via the client and stores the sensor node's authentication information (SID_j, S) in the blockchain ledger.
The elliptic curve algorithm is used to generate the user's private/public key pair y, Y.

$$X_i = h\left(ID_i \parallel PW_i\right), Y_i = h\left(ID_i \parallel BPW_i\right) \tag{4}$$

Where ID_i represents the identity of user i and PW_i represents the static password of user i.
Queries are made on the blockchain network via the client to get public information about the gateway node.

$$M_i = mP_i \tag{5}$$

The user's information update process does not require the involvement of a gateway node. As the biometric information of an individual is unique and cannot be changed, the solution is queried on the client side using the user's biometric key $BPW_i' = h\left(R_i'\right)$. R_i'This is a random key.
Subsequently, verify that $A_i' = A_i$ holds. A_i for the user's publicly identifiable information.

$$X_i = h\left(ID_i \parallel PW_i\right), Y_i = h\left(ID_i \parallel BPW_i'\right) \tag{6}$$

$$A_i' = X_i \oplus Y_i \oplus P_i \tag{7}$$

If this does not hold, the identity information of this user is incorrect and the update process is terminated, otherwise a new static password needs to be entered on the client.
The session key for the negotiation between user i and sensor node j in this scheme is SK.

$$SK = h\left(M_1 \parallel M_3 \parallel nM_1\right) = h\left(M_1 \parallel M_3 \parallel mM_3\right) \tag{8}$$

where m *is the random number* chosen by the user and n is the random number chosen by sensor node j.
IoT device A then generates a random number N_1 and obtains the current timestamp T_1 . Then, it uses its own private key to calculate the digital signature a.

$$a = Sign_{SKA}\left(SID_A \parallel N_1 \parallel T_1\right) \tag{9}$$

Where the SID_A represents the identity of IoT device A. IoT device A gets the public key PK_B of IoT device B by querying the blockchain from the client (SID_A, N_1, T_1, a). After encrypting it with PK_B , it gets the authentication request message A_{uh} and sends it to IoT device B.
After receiving the request message from IoT device A, IoT device B first decrypts it

with its own private key. Then, IoT device B obtains the current time intercept T_2 of the system and verifies whether the following condition holds.

$$|T_2 - T_1| \leq \Delta T \tag{10}$$

where ΔT is the maximum transmission delay allowed. If the above conditions do not hold then connected device B terminates the session, otherwise it initiates a verification transaction to the blockchain network via the client.
After receiving the authentication response message, IoT device A first decrypts it with its own private key and then verifies whether the following conditions hold.

$$|T_3 - T_2| \leq \Delta T \tag{11}$$

where T_3 is the current timestamp of IoT device A.

If the above condition does not hold then the session is terminated, otherwise IoT device A initiates a verification transaction to the blockchain network via the client.
Finally, IoT device A calculates the session key with connected device B.

$$K_{AB}{}' = h\left((SK_A * PK_B)\,\|N_1\|\,N_2\right) \tag{12}$$

Where N_2 is a random number generated by connected device B and SK_A the private key of IoT device A.
Once a smart contract is deployed, it is incredibly challenging to change. Therefore, the whole blockchain system may be susceptible to attackers when there are security flaws in the implemented smart contracts. Using fundamental security analysis tools like Chaincode Scanner, Securify, etc., it is essential in this situation to properly verify the smart contract code and fix any security flaws. A code security examiner that targets on smart contracts that have not yet been implemented in Hyperledger Fabric is called Chaincode Scanner. ChainSecurity, a blockchain security business, created the Securify analyzer to find out whether ethereum smart contracts are secure. Although less reliable than Ether, Hyperledger Fabric can complete transactions with consensus in less than a second. Hyperledger Fabric, as opposed to other blockchain technologies, is generally better suited for large-scale business blockchain applications. Chaincode Scanner is employed in this paper.

3.4. **Edge Data Key Negotiation Protocol Based on Short Signatures.** The short signature based edge data key negotiation protocol is proposed for the missing problem of queried data blocks in edge nodes and the need of ultra-low latency. The scheme considers three cases, with single edge, multilateral and combination of multilateral and cloud. Suppose G_1 is a group, P is any generating element in G_1 , and G_2 is another multiplicative cyclic group. The mapping $e : G_1 \times G_1 \to G_2$ is usually referred to as a dual-linear coupling.
For any $a, b \in G_1$, there is a useful method to calculate $e(a, b)$.

$$e\left(a^x, b^y\right) = e(a, b)^{xy} \tag{13}$$

$$e(a, bc) = e(ba, c) = e(a, b)e(a, c) \tag{14}$$

If $e\left(P,\ P\right)\ \neq\ 1$, then P is non-degenerate.
Three primary functional elements make up the brief signature: KeyGen, Sign and Verify.
1) KeyGen: The data owner chooses a random number α as the private key S_K and αP as the public key PK.
2) Sign: The message m is signed as Sig.

$$Sig = \frac{1}{H(m) + \alpha}P \tag{15}$$

3) Verify: Verify the reliability of the message m.

$$\begin{aligned} e(H(m)P+\alpha P, Sig) &= e\left((H(m)+\alpha)P, \frac{1}{H(m)+\alpha}P\right) \\ &= e(P,P)^{(H(m)+\alpha)\frac{1}{H(m)+\alpha}} \\ &= e\left(P,P\right) \end{aligned} \tag{16}$$

If both sides of the above equation are equal, the signature is generated by the person who has the private key α. For easy understanding, the symbols used in the protocol part and their explanations are listed in Table 1. Three primary functional elements make up the brief signature: $\{m_1, m_2, m_3, ..., m_n\}$. As shown in Figure 3, the key negotiation protocol is divided into five steps and three cases (single-sided, multilateral, and a combination of multilateral and cloud) are considered for the missing query data block problem

Table 1. Main Symbol Description.

Symbol	**Description**
F	Data files
n	Number of data blocks
P	Any of the generating elements of the input group
Y	Public Key
H	Hash functions
c	Number of data blocks for random queries
$I = \{s_1, s_2, s_3, \ldots, s_c\}$	The set of indexes of the queried data block
i	Elements in I
v_i	Pseudo-random numbers
O	Collection of data block indexes in edge nodes
chal	Challenge request, chal=$\{(i,\ v_i\ )\}$
$\hat{I}$	Homomorphic verifiable tag (HVT)
proof	Evidence
t	Number of edge nodes
a	Private Secret Key

For the one-sided case, the edge node needs to compute $\{R, \mu, \eta\}$ and transact with the smart contract as proof upon receipt of the challenge demand and the block's signature.

$$R = \sum_{i=s_1}^{s_c} v_i Y \tag{17}$$

$$\mu = \sum_{i=s_1}^{s_c} v_i H\left(m_i\right) P \tag{18}$$

$$\eta = P - P^2 \sum_{i=s_1}^{s_c} \frac{v_i}{\delta_i} \tag{19}$$

For the multilateral example, there are not enough data blocks available on a single edge node, so close-by edge nodes will be queried as an assistance. The subsequent edge endpoint calculates $\{R, \mu, \eta\}$ and sends it to the smart contract as a *proof*. The edge node E_j can provide a valid data block index set O_j for the missing data blocks as shown

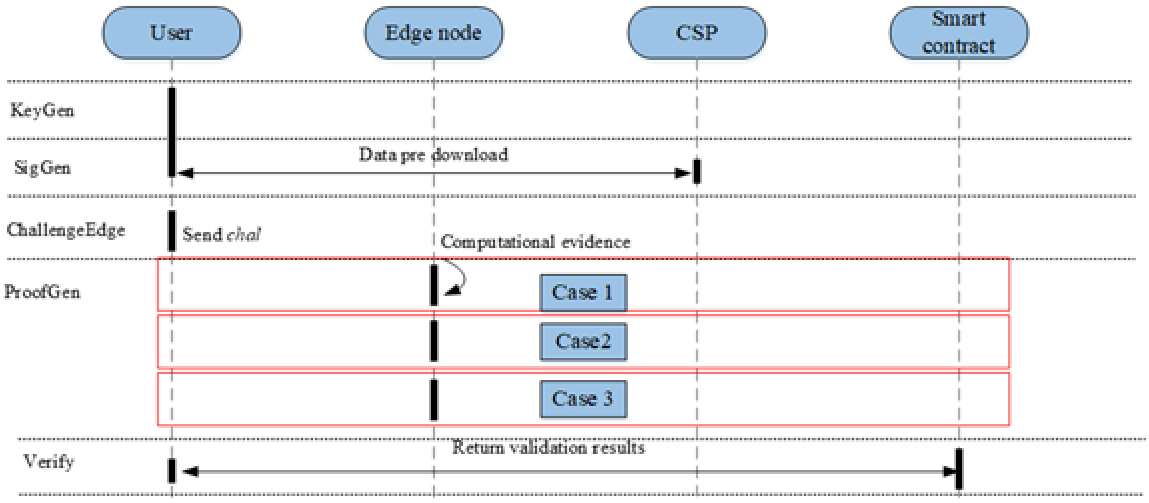

Figure 3. Short Signature Based Edge Data Key Negotiation Protocol.

below.

$$\begin{aligned} I_j &= (I - (I \cap O_1) \cup (I \cap O_2) \cup \ldots \cup (I \cap O_{j-1})) \cap O_j \\ &= (I - I \cap (O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{j-1})) \cap O_j \\ &= I \cap O_j - I \cap O_j \cap (O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{j-1}) \, , j \in \{1, 2, 3, \ldots, t\} \end{aligned} \tag{20}$$

If $O_1 \cup O_2 \cup O_3 \cup \ldots \cup O_{t-1} \cup O_t = I$, E_t is the last edge node. In a similar one-sided case, the last edge node E_t sends $\{R, \mu, \eta\}$ as proof to the smart contract.

$$R = \sum_{j=1}^{t} \sum_{i=s_1}^{s_{e_j}} v_i Y \tag{21}$$

$$\mu = \sum_{j=1}^{t} \sum_{i=s_1}^{s_c} v_i H\left(m_i\right) P \tag{22}$$

$$\eta = P - P^2 \sum_{j=1}^{t} \sum_{i=s_1}^{s_c} \frac{v_i}{\delta_i} \tag{23}$$

For the case of combined multilateral and cloud, individual edge nodes and their adjacent edge nodes are unable to supply enough data blocks, they finally turn to the central cloud for assistance. The computation process is similar to the multilateral case. The last edge node receives the feedback from the central cloud and eventually returns the proof set to the smart contract.

Finally, when getting the proof , the final smart contract computation is performed.

$$e(\eta, P) \cdot e(\mu + R, P) = e(P, P) \tag{24}$$

If the above equation holds, it is obtained that the query data is complete. Then, the smart contract shows TURE for the confirmation outcome to the customer. If not, the smart contract sends FALSE result as the output.

The algorithm for generating *proof* on the edge nodes is shown in Table 2.

4. Testing and analysis of security authentication system.

Table 2. Short signature-based *proof* generation algorithm.

```
Edge node: proof generate
Input:{$P, n, F, H, \alpha$}
Output:{TRUE or FALSE}
$R \leftarrow 0, \mu \leftarrow 0, \eta \leftarrow P$;
        for $i = s_1$ to $s_c$ do
            $proof1 \leftarrow v_i Y,\ R \leftarrow R + proof1$
            $proof2 \leftarrow v_i H(m_i) P, \mu \leftarrow \mu + proof2$
            $proof3 \leftarrow v_i/\delta_i, \eta \leftarrow (\eta - P^2 \cdot proof3)$
        end for
    return $R, \mu, \eta$
end
send $R, \mu, \eta$
```

4.1. **System operating environment.** In this paper, a Hyperledger Fabric Kafka cluster environment is built to serve as the underlying blockchain network, which is deployed in a virtual machine with the network topology shown in Figure 4, mainly containing three Zookeeper nodes, four Kafka nodes, three Orderer sorting service nodes, and four Peer nodes.

Two organizations are added to the blockchain network, Orgl and Org2, each of which contains two Peer nodes and one CA node. The CA nodes in both organizations Orgl and Org2 are deployed on Peer0 node, which is mainly used to issue digital certificates and keys for all members in their organizations and realize the identity management function of MSP. Meanwhile user devices and self-made IoT devices act as users in the blockchain network. The key size used in the experiment is 125 bits and the pseudo-random number size is 65 bits.

The three ports (7051, 7052, and 7053) of the Peer node are the service port of the node, the chain code event listening port, and the Event_Hub block event listening port, respectively. The specific configurations of the server, user devices and homebrew IoT devices used in this test environment are shown in Table 3.

Table 3. Device configuration parameters.

Equipment name	Configuration	System
Server	Xeon E5-2620 v4, 64G	Ubuntu Desktop 14.04 LTS
Mobile user devices	Snapdragon 8 Gen 2, 16 G	Android 13 Beta 2
IoT devices	S5P6818, 1G	linux3.4.39

4.2. **Security Comparison.** The security comparison results between this scheme and the schemes in the literature [26,27,28,29,30,31] are shown in Table 4, where Y represents that the attack can be resisted and N represents that the attack cannot be resisted.
A1: Resistance to Replay Attacks, A2: Resistance to Internal Privilege Attacks, A3: Resistance to Node Capture Attacks, A4: Resistance to Impersonation Attacks, A5: Resistance to User Anonymity, A6: Resistance to Untraceability, A7: Resistance to Guessing Attacks, A8: Bidirectional Authentication, A9: Session Key Negotiation, A10: Resistance to Forward Security Attacks, A11: Resistance to Known Key Attacks, A12: Resistance to Smart Card loss attacks, and A13: time synchronization problems. We can see that

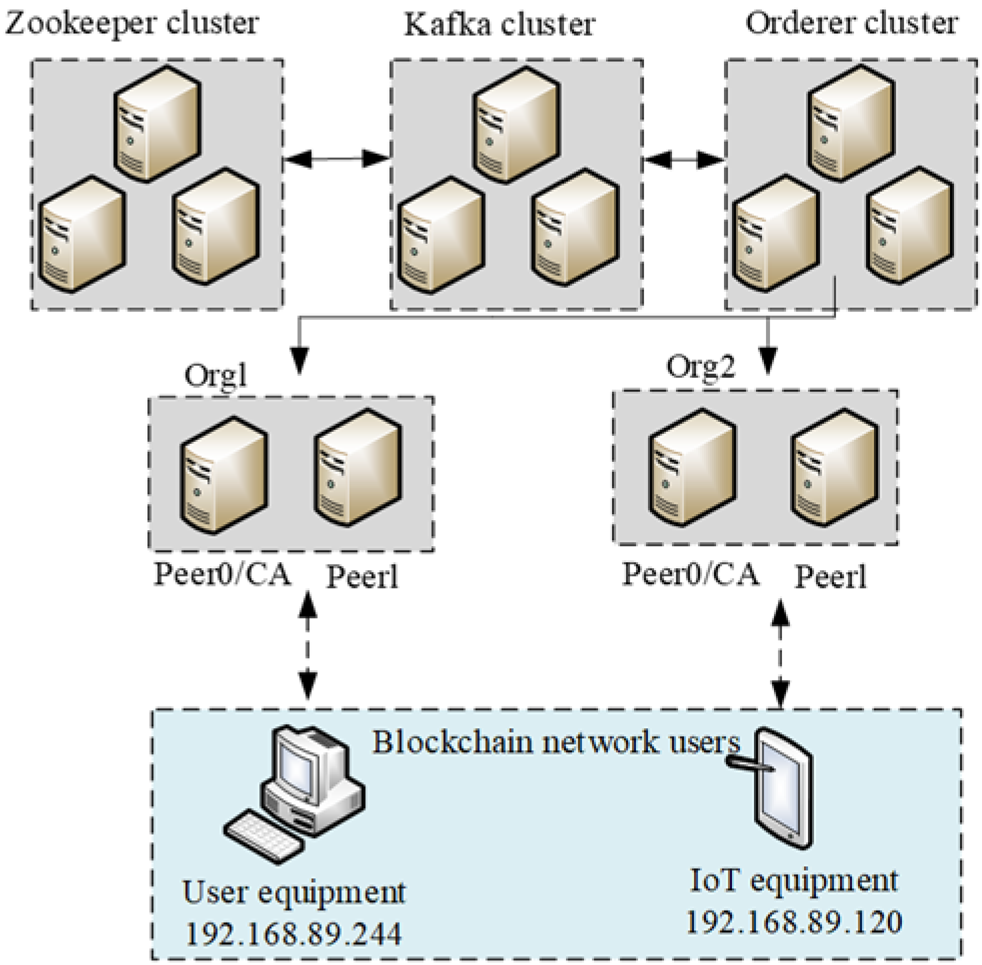

Figure 4. Test environment network topology.

Table 4. Security Comparison.

Function	**[26]**	**[27]**	**[28]**	**[29]**	**[30]**	**[31]**	**Ours**
A1	Y	Y	Y	Y	Y	Y	Y
A2	Y	Y	Y	Y	Y	Y	Y
A3	Y	Y	Y	Y	Y	N	Y
A4	N	Y	Y	Y	N	Y	Y
A5	N	Y	N	N	N	N	Y
A6	N	Y	N	N	N	N	Y
A7	Y	Y	Y	Y	N	Y	Y
A8	Y	Y	Y	Y	Y	Y	Y
A9	Y	Y	Y	Y	Y	Y	Y
A10	Y	Y	N	N	N	N	Y
A11	Y	Y	Y	Y	Y	Y	Y
A12	Y	Y	Y	Y	N	N	Y
A13	N	N	N	N	N	N	Y

the schemes of literature [26] and literature [30] are not resistant to impersonation attacks. The schemes in [26,28,29,30,31] do not have user anonymity and untraceability, and the schemes in [28,29,30,31] are not resistant to forward security attacks. The literature [28,29] cannot resist guessing attacks, while the literature [30] cannot resist node capture attacks. In addition, the schemes of literature [26,27,28,29,30,31] all suffer from the time synchronization problem. It can be seen that the security of this scheme is the highest.

4.3. **Communication Overhead Comparison.** To verify the effectiveness of the key negotiation scheme, the proposed scheme is compared with Rivest-Shamir-Adleman (RSA) based scheme, and Boneh-Lynn-Shacham (RLS) based scheme.
Figure 5 shows the results of the comparison experiments in the one-sided case. The edge node in the experiment has pre-downloaded every block of data that the user requests to certification. It can be seen that the response time of the three schemes increases with the number of query data blocks, and the proposed scheme outperforms the other two schemes in terms of response time, this shows that brief signatures are more effective at verifying data security. Similarly, the results of the other two cases are shown in Figure 6 andFigure 7, respectively. Overall, the proposed solution in this paper has lower service latency.

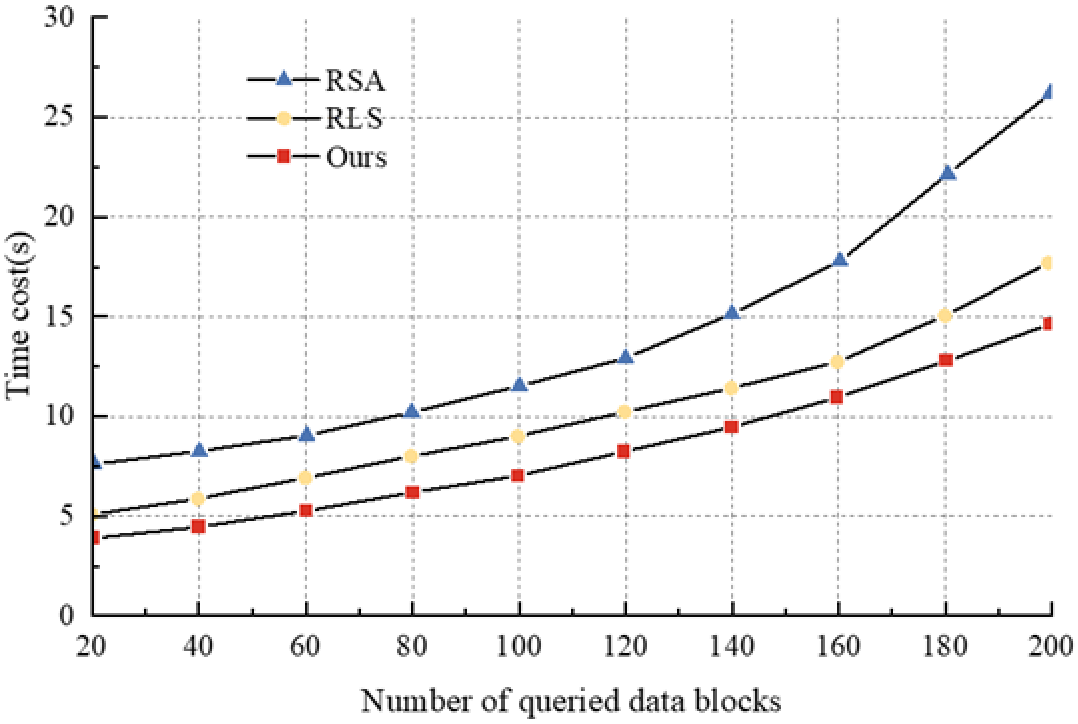

Figure 5. Overall communication overhead in the single-sided case

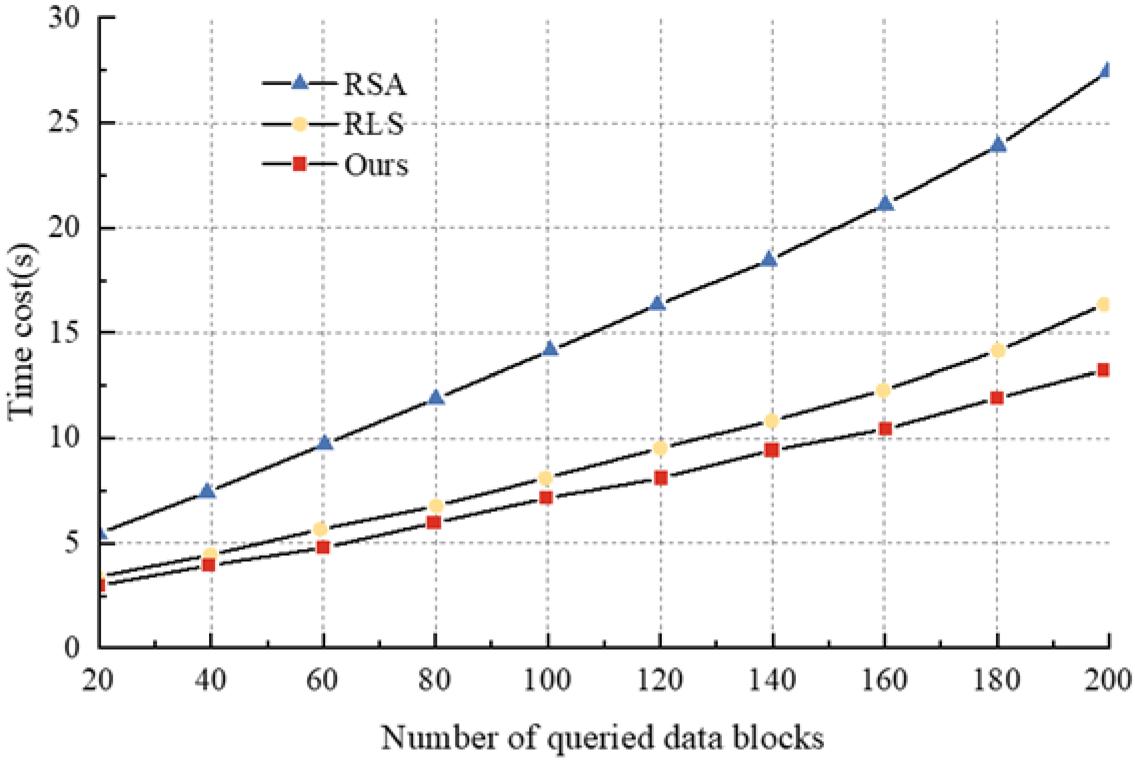

Figure 6. Overall Communication Overhead in the Multilateral Case

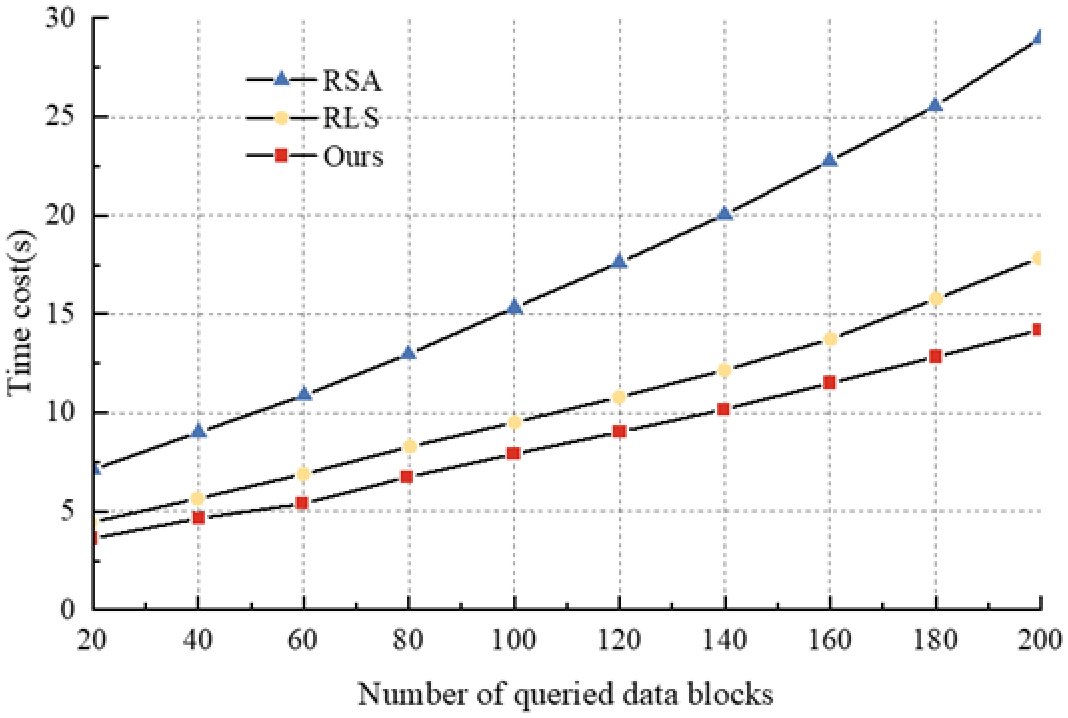

Figure 7. Overall communication overhead in the case of multiple edges combined with cloud

5. **Conclusion.** In this paper, a decentralized IoT scheme is constructed based on edge computing technology and blockchain technology. Then, a short signature-based edge data key negotiation protocol is proposed on this basis respectively, which can accomplish the corresponding secure authentication without the participation of third-party organizations. The scheme has lower latency and can effectively avoid the problem of missing queried databases in the edge cache. Finally, the proposed distributed IoT security authentication scheme is tested with embedded devices and the existing blockchain platform Hyperledger Fabric. The proposed scheme is compared with other related schemes in terms of both security and communication overhead. The results show that the proposed scheme is more secure and has less communication overhead, which means that the proposed scheme has lower service latency. The scheme designed in this paper is still in the research stage and has many shortcomings. The issue of poor agreement efficiency with current blockchain technology will directly impact the response time for authentication. Transactions move more quickly in the Hyperledger Fabric cluster environment created in this study because there are fewer nodes there. The consensus method must be upgraded in order to increase the authentication efficiency since, if it is to be used on a broad scale, the current consensus mechanism is plainly no longer enough.

REFERENCES

[1] T.-Y. Wu, L. Wang, X. Guo, Y.-C. Chen, and S.-C. Chu, "SAKAP: SGX-Based Authentication Key Agreement Protocol in IoT-Enabled Cloud Computing," *Sustainability*, vol. 14, no. 17, 11054, 2022.

[2] T.-Y. Wu, Q. Meng, S. Kumari, and P. Zhang, "Rotating behind Security: A Lightweight Authentication Protocol Based on IoT-Enabled Cloud Computing Environments," *Sensors*, vol. 22, no. 10, 3858, 2022.

[3] T.-Y. Wu, T. Wang, Y.-Q. Lee, W. Zheng, S. Kumari, and S. Kumar, "Improved Authenticated Key Agreement Scheme for Fog-Driven IoT Healthcare System," *Security and Communication Networks*, vol. 2021, 6658041, 2021.

[4] N. Ansari and X. Sun, "Mobile Edge Computing Empowers Internet of Things," *IEICE Transactions on Communications*, vol. E101.B, no. 3, pp. 604-619, 2018.

[5] P. M. Kumar and U. D. Gandhi, "Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application," *The Journal of Supercomputing*, vol. 11, no. 2, pp. 59-78, 2017.

[6] R. N. Li, T. Y. Song, B. Mei, H. Li, X. Z. Cheng, and L. M. Sun, "Blockchain for Large-Scale Internet of Things Data Storage and Protection," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 762-771, 2019.

[7] T.-Y. Wu, L. Yang, Z. Lee, S.-C. Chu, S. Kumari, and S. Kumar, "A Provably Secure Three-Factor Authentication Protocol for Wireless Sensor Networks," *Wireless Communications and Mobile Computing*, vol. 2021, 5537018, 2021.

[8] T.-Y. Wu, Q. Meng, L. Yang, X. Guo, and S. Kumari, "A provably secure lightweight authentication protocol in mobile edge computing environments," *The Journal of Supercomputing*, vol. 78, pp. 13893-13914, 2022.

[9] R. Kumar and R. Tripathi, "DBTP2SF: A deep blockchain-based trustworthy privacy-preserving secured framework in industrial internet of things systems," *Transactions on Emerging Telecommunications Technologies*, vol. 5, no. 8, pp. 109-121, 2021.

[10] A. Ferdowsi and W. Saad, "Deep Learning for Signal Authentication and Security in Massive Internet-of-Things Systems," *IEEE Transactions on Communications*, vol. 67, no. 2, pp. 1371-1387, 2019.

[11] T.-Y. Wu, X. Guo, Y.-C. Chen, S. Kumari, and C.-M. Chen, "SGXAP: SGX-Based Authentication Protocol in IoV-Enabled Fog Computing," *Symmetry*, vol. 14, no. 7, 1393, 2022.

[12] T.-Y. Wu, X. Guo, L. Yang, Q. Meng, and C.-M. Chen, "A Lightweight Authenticated Key Agreement Protocol Using Fog Nodes in Social Internet of Vehicles," *Mobile Information Systems*, vol. 2021, 3277113, 2021.

[13] T.-Y. Wu, L. Yang, Q. Meng, X. Guo, and C.-M. Chen, "Fog-Driven Secure Authentication and Key Exchange Scheme for Wearable Health Monitoring System," *Security and Communication Networks*, vol. 2021, 8368646, 2021.

[14] S. C. Li, N. Zhang, and S. Y. Lin, "Joint Admission Control and Resource Allocation in Edge Computing for Internet of Things," *IEEE Network*, vol. 32, no. 1, pp. 72-79, 2018.

[15] A. Das, M. Degeling, D. Smullen, and N. Sadeh, "Personalized Privacy Assistants for the Internet of Things: Providing Users with Notice and Choice," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 35-46, 2018.

[16] C.-M. Chen, X. Deng, W. Gan, J. Chen, and S. K. Islam, "A secure blockchain-based group key agreement protocol for IoT," *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.

[17] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C.-M. Chen, "An Improved Blockchain-Based Authentication Protocol for IoT Network Management," *Security and Communication Networks*, vol. 2020, 8836214, 2020.

[18] Q. Mei, H. Xiong, Y.-C. Chen, and C.-M. Chen, "Blockchain-Enabled Privacy-Preserving Authentication Mechanism for Transportation CPS with Cloud-Edge Computing," *IEEE Transactions on Engineering Management*, 2022. [Online]. Available: https://doi.org/10.1109/TEM.2022.3159311

[19] Y. Y. Mao, C. S. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322-2358, 2017.

[20] P. A. Rajakumari and P. Parwekar, "Boosting Blockchain Mechanism Using Cryptographic Algorithm in WSN," *Rising Threats in Expert Applications and Solutions*, vol. 3, pp. 509-517, 2022.

[21] R. Goyat, G. Kumar, M. Alazab, R. Saha, R. Thomas, and M. K. Rai, "A secure localization scheme based on trust assessment for WSNs using blockchain technology," *Future Generation Computer Systems*, no. 9, pp. 210-223, 2021.

[22] M. Revanesh and V. Sridhar, "A trusted distributed routing scheme for wireless sensor networks using blockchain and meta-heuristics-based deep learning technique," *Transactions on Emerging Telecommunications Technologies*, vol. 3, pp. 137-145, 2021.

[23] R. Casado-Vara and J. Corchado, "Distributed e-health wide-world accounting ledger via blockchain," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2381-2386, 2019.

[24] A. E. Guerrero-Sanchez, E. A. Rivas-Araiza, J. L. Gonzalez-Cordoba, M. Toledano-Ayala, and A. Takacs, "Blockchain Mechanism and Symmetric Encryption in A Wireless Sensor Network," *Sensors*, vol. 20, no. 10, pp. 2798-2801, 2020.

[25] H. H. Feng, W. S. Wang, B. Q. Chen, and X. S. Zhang, "Evaluation on Frozen Shellfish Quality by Blockchain Based Multi-Sensors Monitoring and SVM Algorithm During Cold Storage," *IEEE Access*, vol. 8, pp. 54361-54370, 2020.

[26] K. Patil, N. Sonawane, E. Patil, K. Kulkarni, and P. Padiya, "Blockchain-Based Security for Super-peer Wireless Sensor Networks," *IC-BCT 2019*, vol. 4, pp. 241-256, 2020.

[27] F. Wang, J. Xu, X. Wang, and S. Cui, "Joint Offloading and Computing Optimization in Wireless Powered Mobile-Edge Computing Systems," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1784-1797, 2018.

[28] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint Optimization of Radio and Computational Resources for Multicell Mobile-Edge Computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89-103, Jun. 2015.

[29] D. Satria, D. Park, and M. Jo, "Recovery for overloaded mobile edge computing," *Future Generation Computer Systems*, vol. 70, pp. 138-147, 2017.

[30] S. Taherizadeh, A. C. Jones, I. Taylor, Z. M. Zhao, and V. Stankovski, "Monitoring self-adaptive applications within edge computing frameworks: a state-of-the-art review," *Journal of Systems and Software*, vol. 136, pp. 19-38, 2018.

[31] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Mobile-Edge Computing Come Home: Connecting things in future smart homes using LTE device-to-device communications," *IEEE Consumer Electronics Magazine*, vol. 5, no. 4, pp. 77-83, 2016.