A Blockchain Based User Subscription Data Management and Access Control Scheme in Mobile Communication Networks

Kaiping Xue, *Senior Member, IEEE*, Xinyi Luo, *Graduate Student Member, IEEE*, Hangyu Tian, Jianan Hong, David S.L. Wei, *Senior Member, IEEE*, and Jian Li, *Member, IEEE*

***Abstract*—In mobile communication networks, when a user roams to and accesses a foreign network, the foreign operator needs to request the user's subscription data from a centralized authentication server, which is managed by the user's home operator. However, centralized authentication introduces single point of failure. Meanwhile, the real-time participation of the home operator and the trust relationship between the foreign operator and the home operator are difficult to guarantee. In this paper, by adopting blockchain and smart contracts, we propose a secure and efficient access control scheme of user subscription data in roaming scenarios. We further design a flexible user authentication scheme, which utilizes derivable tokens based on the proposed access control scheme. By using blockchain to store and manage user subscription data, access control can be decentralized without any trusted third party. Besides, by implementing automatic verification of access privilege through smart contracts, the limitation of the home operators' real-time participation is eliminated. In addition, to further improve security and reduce the on-chain storage overhead, we optimize the data encryption and storage scheme utilizing threshold secret sharing. Our security and performance analysis show that the proposed user subscription data access control scheme for roaming service provides high-level security while causing acceptable time and storage overhead.**

***Index Terms*—Access control, blockchain, derivable token, smart contract, subscription data management.**

I. Introduction

ADVANCES in radio technology and the miniaturization of mobile devices, that is, electronic devices with the capability of computation and wireless communications, have led to the rapid development of mobile communications, enabling users to access useful and timely information anywhere and anytime [1]. However, the radio transmission coverage of mobile devices is limited, making roaming services extremely important. Wireless roaming enables mobile devices to move from the coverage area of one wireless server to that of another wireless server and maintain the wireless access service during this process. In mobile communications, the operator who contracts with a user is called the home operator and stores the user's subscription data, such as consumption habits, terminal information, and service contracts. On the contrary, operators that belong to roaming domains outside the user's home network are referred to as foreign operators. They are supposed to contact a roaming user's home operator to obtain his/her subscription information and perform the authentication procedure to provide corresponding roaming services.

When a foreign operator requests information from the home operator, access control of user subscription data is necessary for the sake of security. To prevent malicious attackers from illegally obtaining user data, requests for subscription data must be authenticated. When receiving a request from a foreign operator, the home operator first needs to verify whether it has signed a roaming agreement with the foreign operator. Then, it is necessary to confirm whether the specific user is roaming to the foreign operator's domain. Besides, when a user roams to a foreign network, his/her home operator needs to authenticate the user in the foreign network. However, existing roaming authentication protocols, both three-party protocols, e.g., [2]–[4] and two-party protocols, e.g., [5]–[7], rely on trusted and centralized authentication servers. Moreover, when a foreign operator requests a user's subscription data, it should establish a trust relationship with the home operator through mutual authentication. Once the home authentication server fails, the authentication procedure will subsequently get the wrong results, leading to the risk of user privacy leakage.

Since Bitcoin [8] was proposed and received widespread attention, its underlying technology, blockchain, has been regarded as an excellent tool to solve the single point of failure problem in centralized systems. Due to its decentralized and non-tamperable features, and has been introduced to various application domains [9]–[11]. Besides, the consensus mechanism used in the blockchain technology helps establish trust

Manuscript received May 14, 2021; revised August 27, 2021 and October 26, 2021; accepted December 16, 2021. Date of publication December 24, 2021; date of current version March 15, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61972371, and in part by theYouth Innovation Promotion Association of the Chinese Academy of Sciences (CAS) under Grant Y202093. The review of this article was coordinated by Prof. Liehuang Zhu. *(Corresponding author: Jian Li.)*

Kaiping Xue, Xinyi Luo, and Jian Li are with the School of Cyber Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: kpxue@ustc.edu.cn; lxy0213@mail.ustc.edu.cn; lijian9@ustc.edu.cn).

Hangyu Tian is with Ant Financial Services Group, Hangzhou, Zhejiang 310013, China (e-mail: thy2014@mail.ustc.edu.cn).

Jianan Hong is with the School of Cyber Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, China (e-mail: hongjn@sjtu.edu.cn).

David S.L. Wei is with the Department of Computer and Information Science, Fordham University, Bronx, NY 10458 USA (e-mail: wei@cis.fordham.edu).

Digital Object Identifier 10.1109/TVT.2021.3138203

relationships among nodes in a decentralized network and maintain consistency [12]. Therefore, to solve security problems caused by centralized authentication and eliminate the requirement of trust relationships among operators, blockchain has been introduced to conduct management and access control of user subscription data [13]–[15]. It should be noted that storing data on blockchain leads to the risk of privacy leakage, making privacy protection particularly important [16]. From another perspective, in existing mobile communication networks, user authentication usually requires the user's home authentication server's real-time participation, which brings a considerable delay of the authentication process. To solve these problems, we adopt smart contracts [17]. Smart contracts are programs, which are deployed on the blockchain that can automatically and correctly execute under the drive of consensus mechanisms. By embedding access control policies in blockchain, smart contracts can then authenticate requests and make decisions without home operator's participation.

In view of the above considerations, in this paper, we adopt blockchain and smart contracts for secure and efficient access control of user subscription data in mobile roaming scenarios. Firstly, blockchain can record access policies with high reliability by virtue of its immutable storage and can automatically and correctly execute programs by utilizing smart contracts, thus handling requests without the home operator's participation, enhancing authentication efficiency. Secondly, with the help of consensus mechanisms, blockchain can provide solid trust relationships in decentralized environments, thereby eliminating the need for any trusted authentication server and providing superior security. Moreover, to reduce the on-chain storage overhead and enhance the security of authentication based on smart contracts, we further propose an improved data storage scheme for user subscription data utilizing threshold secret sharing. In conclusion, this paper makes the following key contributions:

1) We propose a secure and efficient access control scheme of user subscription data in mobile roaming scenarios based on blockchain and smart contracts. The proposed scheme doesn't rely on any trusted third party, eliminates the need for the home operator's real-time participation during authentication processes, and enables dynamic management of access privilege of subscription data.
2) With the help of the proposed users' subscription data access control scheme, we design a flexible user authentication scheme based on derivable tokens, where a user's security token can be derived from the foreign operator's identity. The proposed authentication scheme guarantees that the security tokens delivered to each foreign operator are different and cannot be abused by other operators.
3) Moreover, considering blockchain's limitation on storage, we further optimize the data encryption and storage scheme by virtualizing threshold secret sharing. The improved algorithm significantly reduces the storage overhead and enhances the efficiency of authentication. Meanwhile, the features of threshold encryption ensure that operators can only acquire the user subscription data through appropriate cooperation, which further enhances security.

The rest of our paper is organized as follows: Section II introduces existing access control schemes based on blockchain. In Section III, we provide some basic preliminaries. Section V describes the detailed design of our proposed access control of user subscription data and the roaming authentication scheme Section VI introduces an improved subscription data sharing scheme based on threshold secret sharing, and Section VII provides a security analysis of our scheme. Finally, Section VIII illuminates the detailed deployment of our scheme and gives the related performance analysis. We finally conclude our work in Section IX.

II. Related Work

With security, auditability, and non-tamperability being its salient features, blockchain has been used in access control management in a large number of scenarios, such as the Internet of Things [11], [18], [19], cloud storage [10], [20], [21], and medical data storage [22]–[24], acting as a trusted third party. In this section, we mainly introduce the existing blockchain-based access control work from two aspects: implementation-based and access control model-based. For clarity, we also summarize these related work in TABLE I.

A. Classification by Implementation Technology

1) Schemes Based on Blockchain's Reliable Storage: The most basic function of blockchain is to provide reliable and public storage. By recoding security policies on-chain, entities can conduct authentication and decisions themselves without any trusted authentication center. For example, Xu *et al.* [15] proposed a secure and distributed authentication scheme for wireless mobile networks based on redactable blockchain, allowing users to manage their self-sovereign identities (SSI) through blockchain, thus enabling users to control their identity information. Liu *et al.* [25] also regarded blockchain as reliable storage and proposed a group-authentication scheme for vehicles based on secret sharing and dynamic proxy mechanism. In conclusion, the above schemes regard blockchain as a public and immutable database and record the security policies and other information on-chain, thus realizing decentralized access control. However, since the proposal of Ethereum, blockchain can provide not only reliable storage but also reliable execution of programs of smart contracts.

2) Schemes Based on Smart Contract: Many schemes have been proposed to realize reliable and automatic data access control based on smart contracts. As executable programs deployed on the blockchain, smart contracts can record access control policies securely and publicly, authenticate users automatically, and determine whether to allow requests. When there are frequent access control queries and updates, a centralized authentication server will become a security and performance bottleneck. Therefore, in recent years, blockchain has been used to conduct access control to solve security and performance issues that are challenging for traditional centralized architectures.

Novo *et al.* [26] proposed a new blockchain-based decentralized architecture for IoT device management that uses smart contracts to enforce access control policies. And Sharma *et al.* [27]

proposed a blockchain-based distributed framework for automotive industry. Patil *et al.* [28] utilized the secret computational model of physically unclonable function (PUF) to design a privacy-preserving authentication protocol based on smart contracts that provide data provenance and transparency in IoT networks. Nguyen *et al.* [29] aimed at improving the consensus mechanism to make it more appropriate for mobile roaming scenarios and thus proposed BlockRoam, a blockchain-based roaming service management system with better performance. To reduce storage overhead, Kumar *et al.* [30] introduced Bell-LaPadula model for access control that categorizes peers and transactions in different clearance and security levels, so that some peers do not need to maintain too much blockchain data. The essential idea of the above schemes is to execute the identity authentication of requesters through smart contracts. However, the important user subscription data is ignored and cannot be directly managed through blockchain; therefore, user roaming access privilege cannot be effectively verified by smart contracts.

3) Schemes Based on Non-Public Chain: In addition to public chain scenarios, there are also data access control schemes based on private or consortium blockchains. HDG (Healthcare Data Gateway) [31] is a smart application on mobile phones that uses a private blockchain to store data and allows patients to share medical data easily. To provide privacy protection for the attributes, Pal *et al.* [32] proposed a dual blockchain architecture that adopts a secure private blockchain to store attributes and access control strategies. However, the private chain is essentially a centralized database because its operation rights belong to a single entity, which cannot satisfy our need for a decentralized system. For striking a balance between security and performance, consortium blockchain is usually a better choice. For Internet of Drones (IoD), Bera *et al.* [33] proposed an access control scheme for unauthorized unmanned aerial vehicle (UAV) detection that records authenticate transactions on the blockchain formed by ground station servers for Big Data analytics. Xu *et al.* [34] designed an RSU-assisted (i.e., road-side unit) authentication protocol for the Internet of Vehicles (IoV) where multiple TAs (i.e., Trusted Authorities) form a consortium to manage relative parameters through blockchain.

B. Classification by Access Control Model

According to the adopted access control model, existing access control schemes can be divided into three categories: role-based access control (RBAC) [35], attribute-based access control (ABAC) [36], and capability-based access control (CBAC) [37], [38]. Although these schemes differ in detail and have different characteristics, they still have the problems of trust establishment and single points of failure caused by the centralized authority. Therefore, combining the existing access control model with blockchain technology provides a feasible way to solve the problems caused by the centralized architecture.

In RBAC-based schemes [39], [40], each user is associated with a role that represents a set of permissions assigned by the system. Role assignments and verification should be conducted in a secure way when adopting the RBAC model. For these goals, Nyame *et al.* [40] proposed to combine the ECDSA algorithm and RBAC model to provide a secure mechanism for accessing knowledge in knowledge management systems (KMS) and implemented the scheme through smart contracts to provide security and decentralization.

ABAC-based schemes, e.g., [41], [42], set a set of attributes for entities as decision elements and flexibly decide a user's access rights through the attributes he/she owns, solving the problem of accessing dynamicity. Alansari *et al.* [36] proposed a system that allows federated organizations to implement attribute-based access control while providing privacy protection. When a user's attribute set matches a specific policy, he/she then gets the access right to the corresponding data. Gao *et al.* [43] proposed TrustAccess, which is a ciphertext-policy and attribute-hiding access control scheme based on blockchain. TrustAccess achieves trustworthy and secure access control while providing privacy preservation of policies and attributes. Chen *et al.* [44] introduced K-anonymity technique and attribute-based access control mechanism to propose a medical data sharing scheme based on the consortium blockchain. Iftekhar *et al.* [45] implemented ABAC mechanism using Hyperledger Fabric to gain access to the IoT device.

In CBAC-based schemes, a user should prove that he/she has the corresponding capability when requesting a specific resource. In general, CBAC has better scalability than RBAC and ABAC, and can implement fine-grained access control and capability delegation [38]. Maesa *et al.* [46] proposed to leverage the Bitcoin framework to publish access control policies that indicate the access and delegation rules. BlendCAC [47] is a comprehensive authorization architecture that includes capacity management, delegation propagation, and access right validation, utilizing a specially designed identity-based capability token.

Although these works have addressed some security issues in existing access control schemes from different perspectives through blockchain and smart contracts, there are still no effective solutions for access control of subscription data in roaming scenarios. In essence, the access control is still implemented via only identity authentication (rather than more comprehensive information included in subscription data). Therefore, these solutions cannot satisfy our goals of directly verifying operators' authority through blockchain without the assistance of any other entities. Besides, our scheme also aims to implement data distribution through the blockchain and enables the access control process to be triggered directly by users rather than operators.

III. Preliminaries

In this section, we introduce the concepts of Ethereum, smart contract, and threshold secret sharing scheme.

A. Ethereum and Smart Contract

Ethereum [48] is the second-largest cryptocurrency after Bitcoin, and it is the first cryptocurrency to support smart contracts. A smart contract is defined to be automatically executed on a computer system when certain conditions are met. Smart contracts can perform many functions beyond cryptocurrency trading, such as secure trading, financial data recording, crowdfunding, and so on. Ethereum supports the deployment and execution of smart contract codes and permanently stores the results in the blockchain of Ethereum. Smart contracts are deployed and invoked through transactions, which contain contract codes and/or invocation instructions. A transaction will be collected by a miner who will also execute relevant code to get corresponding results and then be returned to the associated user. Gas is used to measure the cost of each step in the smart contract code. Indeed, using gas can limit the length of smart contracts deployed on Ethereum, i.e., Gas limits the ability of a smart contract to execute without restriction. To be noted, in our work, we deploy our contracts on the Ethereums test network, measure the gas needed for each transaction, and accordingly compute the actual deployment and operational costs based on the current exchange rate.

B. Threshold Secret Sharing Scheme

In a (k, n)-threshold scheme [49], the dealer generates n secret shares in such a way that each of the n shares possesses a share and only when the number of shares is equal to or greater than the threshold k, these shares can cooperate to recover the secret. If the number of shares is less than the threshold k, no secret related information can be obtained. The secret sharing scheme can effectively reduce the security risk brought by an untrusted secret key custodian and avoid the key escrow function being centralized to a single node. Moreover, it can also prevent the key loss, and even if more than $n-k$ shares are lost, the secret key can still be recovered. Finally, an attacker must steal at least k shares to recover the secret, and therefore the occurrence of a single point bottleneck on security can be effectively avoided.

The Shamir threshold sharing scheme is used in our scheme, and its process is given below.

- Select a Galois field F_q and set the participant set as $P = \{P_1, P_2, \ldots, P_n\} (n \leq q)$. Then, set k as the threshold value and choose the secret information $S \in F_q$. Select n different non-zero elements $x_1, x_2, \ldots, x_n$ on F_q and expose these elements.
- Randomly select $k-1$ degree polynomials $g(x) = a_0 + a_1 x + \cdots + a_{k-1} x^{k-1}$ on F_q, where $a_0 = S$. For each $i \in [1, n]$, compute $S_i = g(x_i)$, and distribute (x_i, S_i) as a sub-secret to member P_i separately.
- Any k members can share the sub-secrets they hold, and thereby cooperate to recover the secret S through Lagrange interpolation formula. Let the sub-secret of k member be $\{(x_{i_1}, s_{i_1}), \ldots, (x_{i_k}, s_{i_k})\}$, and given the Lagrange interpolation formula as follows:

$$g(x) = \sum_{r=1}^{k} S_{i_r} \prod_{\substack{t=1 \\ t \neq r}}^{k} \frac{x - x_{i_t}}{x_{i_r} - x_{i_t}}, \quad (1)$$

  we can further compute the secret parameter $S = a_0 = g(0)$ from Eq. (1).

IV. System Model and Security Assumptions

A. System Model

In existing mobile communication networks, taking 5 G's service-oriented interface as an example, the Unified Data Management (UDM) stores the subscription data of User Equipment's (UE) in a local database. There are different types of network services of UE such as network packages or legal monitoring requirements, and different network elements, e.g., Access and Mobility Management Function (AMF), Session Management Function (SMF), Policy Control Function (PCF), etc. Different types of network elements can access different types of subscription data through service-based interfaces. When such a request is sent through the Security Edge Protection Proxy (SEPP) in the control panel signaling, the UDM first determines whether the operator corresponding to the visiting SEPP (vSEPP) has signed a roaming agreement with it. Then, it queries the current attachment location information of UE to determine whether the UE is accessing the network through this operator. Subsequently, the UDM returns the relevant signing data when it determines that the operator meets the requirements. In the proposed scheme, we introduce blockchain into the existing network architecture to achieve decentralized data access control and service authentication.

The proposed scheme consists of the following entities:

- *Home operator*: A user's home operator accepts the user's registration and stores user's subscription data.
- *Foreign operator*: A foreign operator is the operator who has signed a roaming agreement with the user's home operator. A user may roam to several different foreign networks at different times. These foreign operators have to obtain the subscription data when the user is attaching and requesting service.
- *User*: A user signs up with the home operator and then possibly roams to another operator's corresponding network. A user's subscription data is managed by the home operator and can be provided to foreign operators through access control.
- *Database*: The database is used to store the user's subscription data and is decentralized and managed off-chain. The storage addresses are recorded on-chain and can be provided to the foreign operator after the successful verification of its access privilege.
- *Blockchain*: Our proposed system adopts a consortium blockchain that is maintained by multiple operators. The blockchain records storage pointers of users' subscription data in the off-chain database. Besides, smart contracts deployed on the blockchain automatically manage the access control privilege of user subscription data.

B. Security Assumptions

We assume that a user can communicate with his/her home operator in a secure manner. Legal users and operators are assumed to be semi-trusted and will strictly execute the protocol process, but may try to obtain private data from other users/operators. The private keys of users and operators and the symmetric keys involved in the scheme are securely stored. In this paper, we adopt an off-chain and distributed file system such as IPFS [50] to store user subscription data, and the off-chain database is secure and reliable by decentralized storage. The database is accessible by anyone, but only the specific operators can modify some data. Meanwhile, the data is encrypted and can only be decrypted by users/operators with access permission. We assume that there are active attackers and passive attackers in the system. Passive attackers can eavesdrop on communication channels to obtain some transmitted data, while active attackers try to tamper or delete the contract data information stored by users.

C. Design Goals

Our goal is to achieve efficient and effective access control using blockchain technology in mobile roaming scenarios, and the following requirements should be satisfied.

- *Data storage security*: User subscription data should be stored securely both on-chain and off-chain, and it is necessary to ensure that operators who do not meet the predefined access policy cannot acquire user subscription data. In addition, since the subscription data is the private property of the user's home operator, it should also be guaranteed that users cannot obtain the subscription data directly.
- *Tamper-resistant*: All smart contract operations that have been performed should be ensured to be irreversible and unchangeable. Meanwhile, all transactions should be permanently recorded on the blockchain.
- *Authentication security*: Access control scheme should guarantee secure authentication of operators. Specifically, an operator has the right to access a user's subscription data only when it has a roaming partnership with the user's home operator, and the user is now attaching to the operator.
- *Data distribution security*: Our proposed scheme uses blockchain to distribute and manage a user's subscription data and to realize access control in foreign operators through smart contracts deployed on the blockchain. By virtue of blockchain, user subscription data distribution no longer requires the real-time participation of the user's home operator to be online. Moreover, user subscription data is guaranteed to be distributed only to operators who meet the access control requirements.

V. Our Proposed Scheme

In this section, we will explain in detail how our scheme implements secure access control and the secure distribution of user subscription data.

A. Overview

Our scheme mainly leverages smart contracts to realize secure storage, distribution of user subscription data, and access control authentication. There are two types of contracts in our scheme: authority contracts and storage contracts. Authority contract is used to deploy access control policies, while storage contracts are responsible for data storage and distribution. Table II shows the main functions in the two contracts. For clarity, we use AC to represent authority contract and SC to represent storage contract throughout the rest of this paper.

TABLE I
Related Work of Roaming Access Control

Implementation Technology			Access Control Model		
Reliable Storage	*Smart Contract*	*Non-Public Chain*	*RBAC*	*ABAC*	*CBAC*
[15, 25]	[26–29]	[31–34]	[39, 40]	[36, 43, 44]	[38, 46, 47]

TABLE II
Smart Contract Function

Contract steps	Function
AC.Init_provider	Record roaming partner
AC.Init_user	Record user access control information
AC.Judge	Access control judgment
SC.Record	Record the index information of the user's data
SC.Search	Return the index information of the user's data

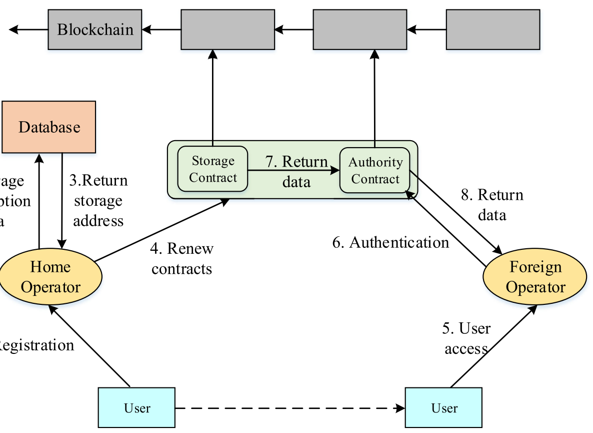


Fig. 1. Overall architecture.

The overall architecture of our scheme is shown in Fig. 1. Firstly, user signs contracts with his/her home operator and subscription data will be stored in SC. To reduce on-chain storage overhead, the original user subscription data is encrypted and stored in an off-chain database maintained by the home operator. The blockchain will securely record storage addresses and decryption keys. In order to prevent malicious nodes from illegally obtaining user subscription data, we design AC to authenticate data requests from foreign operators. Only when a user roams to an operator's responsible network and provides the credential, the operator can then use the credentials to request the data from AC. Then, AC determines whether the operator has a roaming partnership with the user's home operator and decides whether to provide the data. Besides, in order to prevent user data leakage, the on-chain data is also encrypted. Moreover, our proposed scheme also supports users to pay for the network service through smart contracts.

The overall process of the proposed scheme consists of three parts: Firstly, operators sign roaming agreements when deciding to establish roaming partnerships. Each operator records all the roaming partnership in AC for subsequent access control

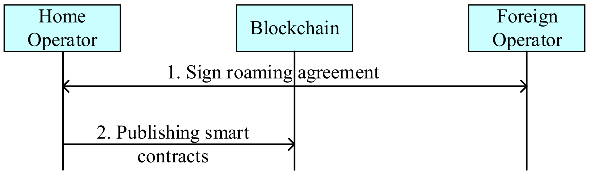


Fig. 2. Signing Roaming Agreement.

authentication. Then, each operator collects the subscription data of each user locally, encrypts and stores it in an off-chain database, and records the storage addresses and encryption key on the blockchain through SC. Finally, when a user roams to a foreign network, the associated foreign operator can send a subscription data access request to AC. AC then automatically judges whether to provide the user's subscription data to the operator. Through these three stages, the proposed scheme realizes not only secure and automatic access control of users' subscription data but also flexible and dynamic allocations of access control rights, and meanwhile realizes the access control without the real-time participation of the home operator.

Although the above scheme is able to provide secure and automatic access control of user subscription data, for enhancing security and reducing storage overhead, we further propose a data management scheme based on threshold secret sharing as an enhancement of the original scheme. The improved scheme is described in Section VI.

B. Signing Roaming Agreement

As shown in Fig. 2, in the roaming agreement signing phase, user's home operator stores the verification information of the operator with which the roaming agreement has been signed in the smart contract AC.

1) Before interacting with the blockchain, the user's home operator obtains the public key and address from all other operators who have signed roaming agreements with it.
2) The home operator records the relevant information of all operators who have established a roaming relationship with it by sending transactions to smart contract AC. Taking foreign operator v as an example, AC will record its public key PK_v and blockchain account acc_v.

C. User Registration

In this stage, users register a network access agreement with their home operator. Then, the home operator saves and publishes the index information of users' subscription data to the blockchain. The premise of this stage is that there is a secure communication channel between the operator and the user.

The protocol is shown in **Algorithm 1** and Fig. 3 with the following steps. We use the input of one user and one foreign operator as an example to illustrate our scheme, though in fact our algorithm could accept multiple inputs at the same time.

1) After the user signs a service agreement with the home operator, the operator obtains the user's subscription data such as service level, identity validity information, real-name information, etc. Taking user UE as an example, both UE and the home operator choose a random number non_{UE} and non_H, respectively, and exchange the random number with each other through a secure channel. In this way, both can get a security token $token = \{non_{UE}||non_H\}$.
2) UE submits a deposit to the contract for subsequent roaming billing.
3) The home operator stores UE's subscription data into an off-chain database after symmetric encryption, and privately stores the corresponding decryption key K_{UE}.
4) The home operator gets the storage path add_{UE} of UE's data in the database. Subscription data for different businesses can be stored in the same or different paths. If stored in different paths, this step returns a list of those paths.
5) The home operator uses $h_v = hash^2\{non_{UE}||non_H||ID_v\}$ for each operator v that has signed a roaming agreement as input to call AC and updates the contract status. Among h_v, ID_v represents the identity of the foreign operator, which can be replaced by acc_v. Then, home operator stores UE's data information $M_v = E_K\{E_{PK_V}\{add_{UE}||K_{UE}\}\}$ in SC. Here the home operator double-encrypts the storage path add_{UE} and decryption key K_{UE} of UE's subscription data in M_v. The inner encryption is asymmetric encryption, the encryption key is the public key PK_V of the foreign operator. The outer encryption is symmetric, and the encryption key is

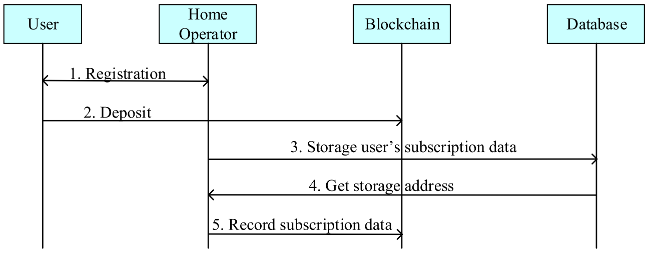


Fig. 3. Users Registration.

Algorithm 1: Initialization Phase.

```
    Input: Foreign operator address $acc_v$ and id $ID_v$;
           Users' id $ID_{UE}$; Data storage path $add_{UE}$
           and decryption key $K_{UE}$; Random number:
           $non_{UE}\&non_H$
    Output: Save results: $true$ or $false$
Calculation:
  $h_v = hash^2\{non_{UE}||non_H||ID_v\}$;
  $K = hash\{non_{UE}||non_H\}$;
  $M_v = E_K\{E_{PK_V}\{add_{UE}||K_{UE}\}\}$.
Record the above data in the contract:
invoke $AC.Init\_provider(ACC_v)$:
    record $map[acc_v] = ID_v$;
invoke $AC.Init\_user(h_v)$:
    record $map[ID_{UE}] = h_v$;
invoke $SC.Record(M_v)$:
    record $map[ID_v] = M_v$.
```

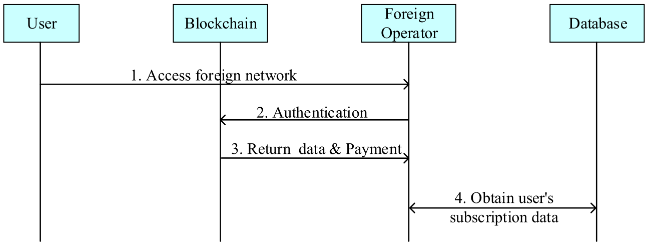


Fig. 4. Access Control of Subscription Data.

the symmetric key $K = hash\{non_{UE}||non_H\}$ generated by the home operator.

After the above stages, different users have now signed a service contract with the home operator. Due to the limited storage capacity of the blockchain, the operator stores users' subscription data in an off-chain database, and the blockchain only stores the storage address and encryption key of different users' data. Apart from this, to prevent unauthorized operators and users from obtaining relevant data content from the blockchain, we perform two layers of encryption processing on the data stored on-chain. If any user's subscription data changes, the home operator should restore the data into the database. Then, the home operator renegotiates the security token (denoted as $token' = \{non'_{UE}||non'_H\}$) with the user, re-computes the relative information (i.e., h_v, K, M_v *et al.* in Algorithm 1) using the new security token, and updates them in AC.

D. Access Control of Subscription Data

This stage occurs when any user accesses a roaming network, the core network element in the roaming domain needs to obtain the subscription data of the user. The foreign operator submits an access request to the blockchain. Then, the smart contract automatically judges the authority of the foreign operator and responds. The protocol is shown in **Algorithm 2** and Fig. 4 with the following steps:

1) After a user UE accesses the roaming network, UE first derives and sends the security token $token_v = \{token||ID_v\}$ to the foreign operator. This step is done off-chain and the token is encrypted by the operator's public key. Then, the foreign operator generates security key $K = hash\{token\}$ for the encrypted information based on the received security token.
2) In this step, the foreign operator sends an access request to AC, including the authentication information $T_v = hash\{token_v\}$ and the identity of UE. AC determines whether $hash\{T_v\}$ is the same with h_v that stored during user registration stage. If not, the program terminates.
3) If the foreign operator passes the authentication of AC, SC will be called and returns M_v to the operator. After the foreign operator obtains M_v, it can perform the decryption process $add_{UE}||K_{UE} = D_K\{D_{SK_V}\{M_v\}\}$ through its private key SK_V and symmetric key K to obtain add_{UE} and K_{UE}. The payment comes from the deposit previously deposited by the user, and is automatically executed through the contract to prevent malicious tenant fraud.

Algorithm 2: Access control judgment.

```
Input: Address of data applicant $acc_v$ and id $ID_v$;
       User's id $ID_{UE}$; Security token: $token_v$
Output: Encrypted data about the user: $M_v$
Calculation:
  $T_v = hash\{token_v\}$;
Make access control judgments:
$AC.Judge(T_v, ID_v)$ :
if $hash(T_v)$ equals $h_v$ and $acc_v$ is legal then
    invoke $SC.Search$;
end
$SC.Search(ID_v)$ :
if $map[acc_v]$ equals $ID_v$ then
    return $M_v = map[ID_v]$;
end
```

4) Finally, the foreign operator can use add_{UE} to query the database for the encrypted subscription data of the user. The foreign operator can then decrypt the context using the symmetric key K_{UE}.

At this stage, our scheme uses smart contracts to achieve access control of user subscription data and uses a two-layer encryption scheme to ensure that unauthorized entities cannot obtain information about the user's data. Even if the online database is attacked or its permission control system is otherwise invalid, the messages obtained from the online database will still be in ciphertext format. The token ensures that the roaming network can obtain relevant data only when roaming actually occurs.

VI. Storage Scheme Based on Threshold Secret Sharing

In the scheme described in Section V, the index information of a user's subscription data stored in the database is directly recorded on-chain. Thus, there is a risk that, even without smart contract access control judgments, the full node can still obtain the encrypted information on the blockchain (though the node cannot decrypt it). On the other hand, for each foreign operator of the user's home operator, it is necessary to store a copy of user information encrypted with its public key on the blockchain, bringing significant storage overhead. Therefore, in this section, we introduce the subscription data sharing scheme based on threshold secret sharing as an enhancement and supplement to our scheme. The entire protocol consists of three stages.

A. Signing Roaming Agreement

In the process of signing roaming agreements between operators, similar to the previous scheme, each operator publishes relevant information of the operators with who it has signed a roaming agreement to AC. For convenience, each operator is assigned a unique number (denoted by i).

B. User Registration

Fig. 3 shows the user registration process, and the difference from the previous scheme lies in step 4 in the figure. For each foreign operator who has signed a roaming agreement, the home operator calls AC to update the contract status and adds the input value: $h_v = hash^2\{non_{UE}||non_H||ID_v\}$ where ID_v represents the identity of the foreign operator. For each user, a threshold t_{UE} is specified, which makes it necessary for the foreign operator to request subscription data with at least t_{UE} operator's confirmation. The user's home operator will generate a symmetric key $K = hash\{non_{UE}||non_H\}$ for the user. The storage path and encryption key of the subscription data are encrypted as follows: $M_v = E_K\{add_{UE}||K_{UE}\}$. Then, the home operator will record $hash\{M_v\}$ in SC for integrity verification.

We adopt the Shamir threshold secret sharing scheme [49] in our scheme for data management with optimal storage overhead. After obtaining the user's subscription data, the home operator chooses a $t_{UE} - 1$ degree polynomial $f(x)$, which corresponds to the secret M_v. Then, the home operator calculates security sharing $f(i)$ and securely passes $f(i)$ and $hash\{M_v\}$ to each blockchain node controlled by operator numbered i. Since $f(i)$ saved by each blockchain node is different, to avoid breaking the consensus, only $hash\{M_v\}$ is stored in the blockchain to verify data integrity. And $f(i)$ is stored locally and secretly by different blockchain nodes.

C. Access Control of Subscription Data

After storing a user's data, when the user attaches to a roaming network, the foreign operator needs to request the user's subscription data, as shown in Fig. 4. The difference from the previous protocol lies in Steps 2 & 3. After the foreign operator receives the token about the user and sends a query request to AC, AC first determines whether the operator is a legitimate roaming partner. If true, AC then checks whether the security token provided by the operator is valid by judging whether $hash^2\{token_v\}$ and h_v are the same. If valid, AC then extracts the public key PK_v and returns the audited message.

For data distribution, we adopt an improved consensus based on PBFT [51]. In each round of consensus, a master node is selected as the leader. The master node first collects all transactions, including different data requests. It then packages and distributes the judgment results to each blockchain node where the results are obtained by executing AC. Each blockchain node i also executes AC to verify whether the results sent by the master node are reasonable, and then determines whether to encrypt the secret slice $f(i)$ with PK_v, the foreign operator's public key, finally sends the encryption result to the foreign operator.

The foreign operator will obtain several encrypted M_v slices in the process. If the number of shares is less than t_{UE}, the process terminates. Otherwise, the operator uses its local private key SK_v to decrypt these slices and performs the secret recovery process. The recovery process is: $M_v = \sum^{t_{UE}} l_i f(i)$, where l_i is the weight coefficient corresponding to the secret recovery. When the secret is successfully recovered, the foreign operator can verify its correctness through $hash\{M_v\}$ in SC. After that, the decryption process can be performed as: $add_{UE}||K_{UE} = D_K\{M_v\}$. Then, the foreign operator can obtain the required user's subscription data from the off-chain database through the decrypted information.

The above storage scheme that based on the threshold secret sharing enhances our original scheme, improving security and reducing the on-chain storage overhead. This method relies more on smart contracts and blockchain consensus rather than encryption to protect data security. Therefore, foreign operators are prevented from maliciously acquiring (encrypted) data because the smart contract and other operators will not allow illegal data requests. Compared to storing copies encrypted by the public keys of different operators in the blockchain, the threshold-based storage scheme allows a requester to obtain corresponding data only when the blockchain reaches a consensus. When an operator signs roaming agreements with multiple operators at the same time, only one copy of the ciphertext needs to be stored in the storage contract, effectively reducing the on-chain storage overhead.

VII. Security Analysis

In this section, we provide a security analysis of our scheme and discuss how our scheme enables secure storage and distribution of subscription data in mobile roaming scenarios. For the convenience of description, in the remaining sections, we will refer to the scheme described in Section V as the original scheme and the scheme described in Section VI as the improved scheme.

A. Data Storage Security

The user subscription data is stored in off-chain databases with symmetric encryption. Unless the symmetric key is leaked, no illegal entity, except for the home operator, can obtain any data. Besides, through AC, operators who request the subscription data will be authenticated. Only when a user attaches to a legal roaming network and provides the operator with relevant access credentials can the operator obtain the index information about the user's subscription data from AC, including the storage address and encryption key. Since all the on-chain data is public, a full node can obtain on-chain data without access control, causing a risk of information leakage. Against this problem, the home operator applies a two-layer encryption to the index information before recording it on-chain. Thus, for a user, it is impossible to decrypt without the private key of the foreign operator. For other operators, because they cannot obtain a user's access credentials before the user accesses their network, they cannot generate the decryption key. Moreover, to further enhance the security, in our improved scheme, user subscription data is not directly encrypted and stored on the blockchain, but is stored by different blockchain nodes after being fragmented, and only a hash is recorded on the blockchain for integrity judgment. In this way, even the encrypted data can only be obtained after an on-chain consensus verification, further preventing the possibility of information leakage.

B. Tamper-Resistant

There may be attackers who want to tamper with user subscription data in our scheme. However, data in the off-chain database is encrypted by the home operator, and the data integrity verification mechanism adopted by the database guarantees that it will not be tampered with, which is not on our radar. The index information of a user's subscription data stored on the blockchain is guaranteed by the blockchain's consensus mechanism to prevent tampering. This security guarantee is based on the security assumptions of the blockchain consensus. For example, in the PoW-based public chain, it is assumed that the computing power of the attacker does not exceed 50% of the total computing power of the entire network. The consensus algorithm of consortium blockchain requires that the number of malicious nodes does not exceed $1/3$ of the total number of participating consensus nodes.

C. Authentication Security

Authentication security means that the operator can obtain the user's subscription data only when a user roams to an operator's responsible network and the operator has a roaming partnership with the user's home operator. To achieve this, we set a value $T_v = hash\{token_v\}$ for each operator v in AC; an operator can obtain a user's data information only if the operator provides a valid T_v in the request transaction. On the one hand, the operator can only acquire $token_v$ when the user roams to its responsible network and actively provides the token. On the other hand, the value t_v corresponds to each operator is independent of each other. Therefore, other operators cannot obtain the user's subscription data by replaying the request information. Furthermore, once the user leaves the roaming network, if the user's subscription data is not updated, we believe there is no need to update the security token provided by the user to the foreign operator. Suppose the home operator updates any user's subscription data, as mentioned in Section V-C. In such a case, operators will not only update the user's subscription data in the off-chain database but also renegotiate a new security token with the user and update the relevant smart contract. In this way, any roaming network that the user has not yet accessed after the update cannot obtain the user's updated security token, and therefore cannot generate relevant credentials to request the user's data from AC.

D. Data Distribution Security

In the original scheme, access control of users' subscription data is automatically performed through a smart contract. Only operators that have passed the authentication are distributed the index information of the requested user's subscription data. On the contrary, the improved scheme proposed in Section VI based on Shamir threshold secret sharing [49] satisfies the security and recoverability based on the original scheme. This is because the restoration method uses k coordinate values to determine the $k-1$ order equation through Lagrange interpolation, and the constant term is secret information. Therefore, less than k coordinates cannot restore the equation correctly and thus cannot obtain any information of user's subscription data. However, any k coordinates can quickly restore the original encrypted data.

TABLE III
THE GAS VALUE OF EACH FUNCTION

Transaction	Gas	Gwei	Ether
AC.Init_provider	$104,491$	$5,224,550$	0.052
AC.Init_user	$44,815$	$2,240,750$	0.022
SC.Record	$127,959$	$6,397,950$	0.064
AC.Judge	$43,904$	$2,195,200$	0.022
SC.Search	$14,143$	$707,150$	0.071

TABLE IV
TIME COST OF CONTRACT FUNCTION

Transaction	Time cost (ms)
AC.Init_provider	172.951
AC.Init_user	154.405
SC.Record	214.697
AC.Judge	167.305
SC.Search	177.187

VIII. PERFORMANCE ANALYSIS

To evaluate the performance of the proposed scheme, we make a prototype and test the performance from the aspects of computational complexity, runtime cost, and storage overhead, and finally, we discuss the scalability based on these analyses. In our experiments, we adopt the Ethereum private chain built based on Ganache [52] as the underlying blockchain. The operations of mobile users and operators are based on Javascript (node.js). Users and operators communicate with Ethereum through $web3.js$. All measurements are performed on a desktop running Windows 10 equipped with 6 cores, 3.0 GHz Intel Core i5, and 8 GB DDR4 RAM.

We implement the main functions described in Table II listed in Section V-A and the corresponding two smart contracts, i.e., AC and SC. In our proposed scheme, user subscription data is encrypted and stored in the operators' local database, and the storage address and key are encrypted and uploaded to the blockchain. Since we are mainly concerned about the performance of the on-chain part, there is no need to conduct experiments on real user subscription data. On the contrary, we directly randomly generate a series of $\{ID_{UE}, add_{UE}||K_{UE}\}$ to simulate the on-chain data.

A. Computational Complexity

In Ethereum, the cost of running smart contracts is directly evaluated by gas, that is, each invocation and execution of the smart contract will consume a certain amount of gas. Thus, to evaluate performance of the proposed scheme, we deploy our contracts on Ropsten, the Ethereum's official test network [53], to test the gas spent amount of each step, as shown in Table III. To show the cost more clearly, we further convert gas to Ether based on the real world exchange rate in August 2021, i.e., 1 gas $=$ 50 GWei (1 GWei $= 10^{-9}$ Ether).

B. Runtime Cost

The local runtime cost of our scheme is shown in Table IV, which shows that our contract can achieve the expected functionality within an acceptable time overhead.

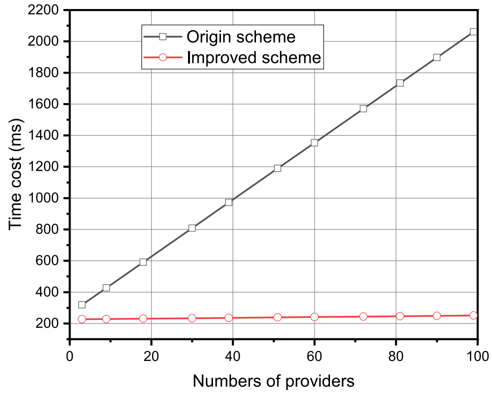


Fig. 5. Comparison of time spent in the process of user registration.

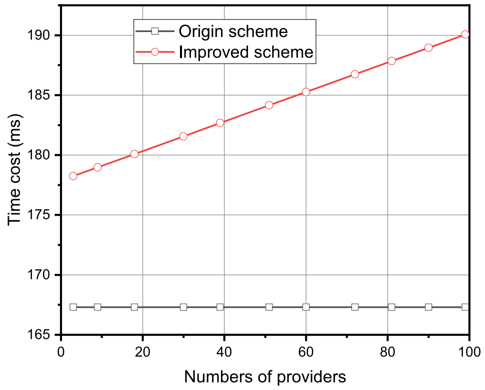


Fig. 6. Comparison of time spent in the process of access control of subscription data.

We implemented the Shamir secret sharing scheme used in the improved scheme based on Javascript (node.js). Compared with the original scheme, the improved scheme requires additional local operations.

Firstly, in the process of recording a user's subscription data, in addition to recording the hash of the user's relevant information into the blockchain, the operator should also send the user's data to other operators after sharing operations. Secondly, when a foreign operator requesting data from the contract, in addition to waiting for the judgment result through the contract, the operator also needs to combine different slices locally to restore the original data. So it will bring extra time cost. Without loss of generality, we assume that the communication delay between different entities is 10 ms. During the user registration phase of the scheme, we can see from Fig. 5 that the time cost of the original scheme will increase as the number of participating operators increases (this number is also the threshold value in the secret sharing scheme). This is because the blockchain needs to record the encrypted data of each operator separately. Thus, the time cost is more than the overhead of data sharing in the improved scheme.

It can be seen from Fig. 6 that as the number of participating operators increases, the user data storage and acquisition steps in the improved scheme will bring higher time costs due to the use of the Shamir secret sharing scheme. Because the access control judgment of each operator is done in parallel, this part will not bring extra time cost. Therefore, in the access control of the user subscription data phase of our scheme, more time cost comes from the threshold secret sharing data scheme. Finally, we superimpose the two parts to get the total time cost of our scheme. Here we think that the operator registration process is in parallel, so only the average time cost is considered. It can be seen from Fig. 7 that although the improved scheme will bring additional time cost, because the original scheme needs to record data for each additional operator, the overall overhead of the improved scheme will be much lower. But relative to the cost of blockchain consensus, the increase in time cost is acceptable whether it is compared with the consensus in Ethereum or PBFT [54]. And the improved scheme brings a more decentralized data storage solution and lower blockchain storage overhead.

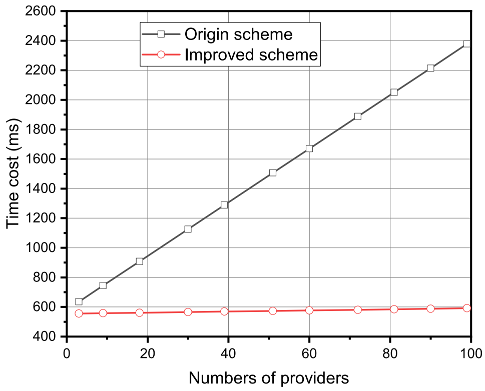


Fig. 7. Comparison of time spent in the full process.

C. Storage Overhead

In the original scheme, we store a user's subscription data in an off-chain database, and store its index address and encryption key information on blockchain after encryption. Here we use symmetric encryption algorithm and asymmetric encryption algorithm for encryption in turn. We assume that encryption is performed using the symmetric encryption algorithm AES with a key length of $128 - bit$ and the asymmetric encryption algorithm RSA with a key length of $256 - bit$. And we assume that the index of the user's subscription data in the database is represented by a string of less than $256 - bit$. In this way, the length of the ciphertext obtained after encryption is $256 - bit$. In the improved scheme, we only store the hash value of the user's subscription data on blockchain for integrity verification. Also, we choose SHA-256 algorithm, and the hash value obtained after processing is also $256 - bit$. Under our design, the storage data of a user's subscription data in the two schemes on the blockchain are the same. However, in the original scheme, the subscription data of each user needs to be encrypted separately for different foreign operators. In the improved scheme, the user's subscription data only needs to be hashed once. This has led to a difference in the storage requirements of the two schemes on the blockchain. The changes in the size of the data that needs to be stored on the blockchain with respect to the number of users and the number of operators under the two schemes are

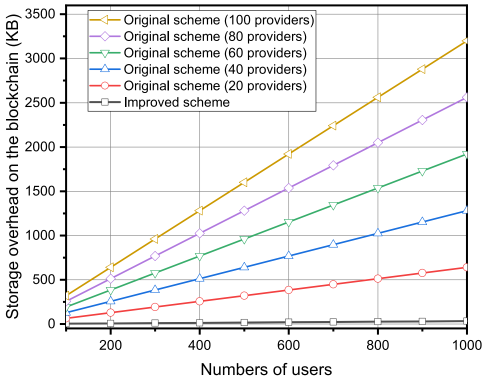


Fig. 8. Storage Requirements on Blockchain.

shown in Fig. 8. It can be seen from the figure that compared to the original scheme, the improved scheme greatly reduces the storage pressure on the blockchain.

D. Scalability

The scalability mainly includes two aspects, i.e., processing capacity and storage overhead, and we therefore analyze scalability from these two aspects, respectively.

Processing capacity. Since the processes (e.g., user subscription, access control, or data update) run in smart contracts, each process can be regarded as a blockchain transaction. Thus, the processing capacity of the proposed scheme is equal to the *transactions per second (TPS)* value of the underlying blockchain. In our experiments, we adopt the Ethereum private chain built based on Ganache as the underlying blockchain, thus it is reasonable to take Ethereum 2.0 (which can also be regarded as a Ethereum private chain [55]) as a reference. According to CoinDesk which is one of the most famous cryptocurrency communities, Ethereum 2.0 will boost network speeds to 100,000 transactions per second [56]. Therefore, we consider that in the most ideal situation, the proposed scheme can process 100,000 user requests per second. It should be noted that the runtime shown in Table IV is the processing time of a single function (or transaction), however, by adopting multiple smart contracts, the requests from different users can be executed in parallel.

Storage overhead. Since the actual users' subscription data is stored in the operators' local database, and only the addresses and keys are uploaded to the blockchain, the storage overhead of the proposed scheme is negligible. As Fig. 8 shows, the storage overhead is linear with the number of users. When adopting the improved scheme with threshold secret sharing (see Section VI), 1,000 users only need about 50 KB storage space. Considering that the average size of an Ethereum block is about 2 MB, in the most ideal situation, one block can therefore contain 40,000 users roughly.

IX. Conclusion

In this paper, we proposed a blockchain-based user data access control scheme to determine whether a foreign operator has the authority to obtain a roaming user's subscription data. In our scheme, the dynamic management of the access right of a user's subscription data can be effectively achieved through the blockchain without the participation of the user's home operator. The basic scheme requires a large on-chain storage overhead; Thus, we further introduced threshold secret sharing and proposed a storage-friendly scheme, significantly reducing the on-chain storage overhead. Our proposed scheme provided high-level security and solved the issue of requiring the real-time response of home operators. However, the consensus of blockchain may cause a negative impact on performance. To evaluate performance, we implemented the contract and evaluated its execution cost. Results showed that both time and storage overhead is within the practically acceptable range.

References

[1] G. Yang, Q. Huang, D. S. Wong, and X. Deng, "Universal authentication protocols for anonymous wireless communications," *IEEE Trans. Wireless Commun.*, vol. 9, no. 1, pp. 168–174, Jan. 2010.

[2] J. Zhu and J. Ma, "A new authentication scheme with anonymity for wireless environments," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 231–235, Feb. 2004.

[3] C.-C. Lee, M.-S. Hwang, and I.-E. Liao, "Security enhancement on a new authentication scheme with anonymity for wireless environments," *IEEE Trans. Ind. Electron.*, vol. 53, no. 5, pp. 1683–1687, Oct. 2006.

[4] C.-C. Chang and H.-C. Tsai, "An anonymous and self-verified mobile authentication with authenticated key agreement for large-scale wireless networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3346–3353, Nov. 2010.

[5] D. He, J. Bu, S. Chan, C. Chen, and M. Yin, "Privacy-preserving universal authentication protocol for wireless communications," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 431–436, Feb. 2011.

[6] J. K. Liu, C.-K. Chu, S. S. Chow, X. Huang, M. H. Au, and J. Zhou, "Time-bound anonymous authentication for roaming networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 1, pp. 178–189, Jan. 2015.

[7] Q. Yang, K. Xue, J. Xu, J. Wang, F. Li, and N. Yu, "AnFRA: Anonymous and fast roaming authentication for space information network," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 2, pp. 486–497, Feb. 2019.

[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008, Accessed: Jan. 2022. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[9] X. Luo, K. Xue, J. Xu, Q. Sun, and Y. Zhang, "Blockchain based secure data aggregation and distributed power dispatching for microgrids," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5268–5279, Nov. 2021.

[10] L. Zhu, Y. Wu, K. Gai, and K.-K. R. Choo, "Controllable and trustworthy blockchain-based cloud data management," *Future Gener. Comput. Syst.*, vol. 91, pp. 527–535, 2019.

[11] K. Gai, Y. Wu, L. Zhu, Z. Zhang, and M. Qiu, "Differential privacy-based blockchain for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 6, pp. 4156–4165, Jun. 2020.

[12] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3548–3558, Jun. 2019.

[13] S. Rouhani and R. Deters, "Blockchain based access control systems: State of the art and challenges," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2019, pp. 423–428.

[14] D. D. F. Maesa, P. Mori, and L. Ricci, "A blockchain based approach for the definition of auditable access control systems," *Comput. Secur.*, vol. 84, pp. 93–119, 2019.

[15] J. Xu, K. Xue, H. Tian, J. Hong, D. S. L. Wei, and P. Hong, "An identity management and authentication scheme based on redactable blockchain for mobile networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5760–5772, Jun. 2020.

[16] A. Dua, N. Kumar, A. K. Das, and W. Susilo, "Secure message communication protocol among vehicles in smart city," *IEEE Trans. Veh. Technol.*, vol. 67, no. 5, pp. 4359–4373, May 2018.

[17] V. Buterin, "A next-generation smart contract and decentralized application platform," 2014, Accessed: Jan. 2022. [Online]. Available: https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf

[18] S. Shafeeq, M. Alam, and A. Khan, "Privacy aware decentralized access control system," *Future Gener. Comput. Syst.*, vol. 101, pp. 420–433, 2019.

[19] T. Chen, L. Zhang, K.-K. R. Choo, R. Zhang, and X. Meng, "Blockchain based key management scheme in fog-enabled IoT systems," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 766–10 778, Jul. 2021.

[20] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?," *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan./Feb. 2018.

[21] Y. Zhang, B. Li, B. Liu, Y. Hu, and H. Zheng, "A privacy-aware PUFs-based multi-server authentication protocol in cloud-edge IoT systems using blockchain," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13958–13974, Sep. 2021, doi: 10.1109/JIOT.2021.3068410.

[22] J. Xu *et al.*, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8770–8781, Oct. 2019.

[23] T. Benil and J. Jasper, "Cloud based security on outsourcing using blockchain in E-health systems," *Comput. Netw.*, vol. 178, 2020, Art. no. 107344.

[24] L. Soltanisehat, R. Alizadeh, H. Hao, and K.-K. R. Choo, "Technical, temporal, and spatial research challenges and opportunities in blockchain-based healthcare: A systematic literature review," *IEEE Trans. Eng. Manag.*, 2020, to be published, doi: 10.1109/TEM.2020.3013507.

[25] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan, and Y. Zhang, "Blockchain empowered cooperative authentication with data traceability in vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4221–4232, Apr. 2020.

[26] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.

[27] P. K. Sharma, N. Kumar, and J. H. Park, "Blockchain-based distributed framework for automotive industry in a smart city," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4197–4205, Jul. 2019.

[28] A. S. Patil, R. Hamza, A. Hassan, N. Jiang, and H. Yan, "Efficient privacy-preserving authentication protocol using PUFs with blockchain smart contracts," *Comput. Secur.*, vol. 91, 2020, Art. no. 101958.

[29] C. T. Nguyen *et al.*, "BlockRoam: Blockchain-based roaming management system for future mobile networks," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2021.3065672.

[30] R. Kumar and R. Tripathi, "Scalable and secure access control policy for healthcare system using blockchain and enhanced bell–lapadula model," *J. Ambient Intell. Humanized Comput.*, vol. 2, no. 12, pp. 2321–2338, 2021.

[31] X. Yue, H. Wang, D. Jin, M. Li, and W. Jiang, "Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control," *J. Med. Syst.*, vol. 40, no. 10, pp. 1–8, 2016.

[32] S. Pal, T. Rabehaja, A. Hill, M. Hitchens, and V. Varadharajan, "On the integration of blockchain to the Internet of Things for enabling access right delegation," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2630–2639, Apr. 2019.

[33] B. Bera, A. K. Das, and A. K. Sutrala, "Private blockchain-based access control mechanism for unauthorized UAV detection and mitigation in internet of drones environment," *Comput. Commun.*, vol. 166, pp. 91–109, 2021.

[34] Z. Xu, W. Liang, K.-C. Li, J. Xu, and H. Jin, "A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles," *J. Parallel Distrib. Comput.*, vol. 149, pp. 29–39, 2021.

[35] J. Shahen, J. Niu, and M. Tripunitara, "Cree: A performant tool for safety analysis of administrative temporal role-based access control (ATRBAC) policies," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 5, pp. 2349–2364, Sep./Oct. 2021.

[36] S. Alansari, F. Paci, and V. Sassone, "A distributed access control system for cloud federations," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 2131–2136.

[37] P. N. Mahalle, B. Anggorojati, N. R. Prasad, and R. Prasad, "Identity authentication and capability based access control (IACAC) for the Internet of Things," *J. Cyber Secur. Mobility*, vol. 1, no. 4, pp. 309–348, 2013.

[38] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the Internet of Things," *Math. Comput. Modelling*, vol. 58, no. 5-6, pp. 1189–1205, 2013.

[39] N. Mundbrod and M. Reichert, "Object-specific role-based access control," *Int. J. Cooperative Inf. Syst.*, vol. 28, no. 01, 2019, Art. no. 1950003.

[40] G. Nyame, Z. Qin, K. O.-B. O. Agyekum, and E. B. Sifah, "An ECDSA approach to access control in knowledge management systems using blockchain," *Information*, vol. 11, no. 2, pp. 1–12, 2020.

[41] Y. Zhang, D. Zheng, and R. H. Deng, "Security and privacy in smart health: Efficient policy-hiding attribute-based access control," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2130–2145, Jun. 2018.

[42] H. Guo, E. Meamari, and C.-C. Shen, "Multi-authority attribute-based access control with smart contract," in *Proc. Int. Conf. Blockchain Technol.*, 2019, pp. 6–11.

[43] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "TrustAccess: A trustworthy secure ciphertext-policy and attribute hiding access control scheme based on blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5784–5798, Jun. 2020.

[44] Y. Chen, L. Meng, H. Zhou, and G. Xue, "A blockchain-based medical data sharing mechanism with attribute-based access control and privacy protection," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–12, 2021.

[45] A. Iftekhar, X. Cui, Q. Tao, and C. Zheng, "Hyperledger fabric access control system for Internet of Things layer in blockchain-based applications," *Entropy*, vol. 8, no. 23, 2021, Art. no. 1054.

[46] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Proc. Int. Federation Inf. Process. Int. Conf. Distrib. Appl. Interoperable Syst.*, Springer, 2017, pp. 206–220.

[47] R. Xu, Y. Chen, E. Blasch, and G. Chen, "BlendCAC: A blockchain-enabled decentralized capability-based access control for IoTs," in *Proc. IEEE Int. Conf. Internet Things IEEE Green Comput. Commun. IEEE Cyber Phys. Soc. Comput IEEE Smart Data*, 2018, pp. 1027–1034.

[48] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," 2017, Accessed: Jan. 2022. [Online]. Available: https://files.gitter.im/ethereum/yellowpaper/VIyt/Paper.pdf

[49] A. Shamir, "How to share a secret," *Commun. Assoc. Comput. Machinery*, vol. 22, no. 11, pp. 612–613, 1979.

[50] N. Nizamuddin, H. R. Hasan, and K. Salah, "IPFS-blockchain-based authenticity of online publications," in *Proc. Int. Conf. Blockchain*. Springer, 2018, pp. 199–212.

[51] V. Gramoli, "From blockchain consensus back to byzantine consensus," *Future Gener. Comput. Syst.*, vol. 107, pp. 760–769, 2020.

[52] "Ganache: A personal blockchain for ethereum development," Accessed: Jan. 2022. [Online]. Available: https://www.trufflesuite.com/ganache

[53] "The ethereum block explorer: ROPSTEN (revival) TESTNET," Accessed: Jan. 2022. [Online]. Available: https://ropsten.etherscan.io

[54] H. Sukhwani, J. M. Martínez, X. Chang, K. S. Trivedi, and A. Rindos, "Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric)," in *Proc. 36th IEEE Symp. Reliable Distrib. Syst.*, 2017, pp. 253–255.

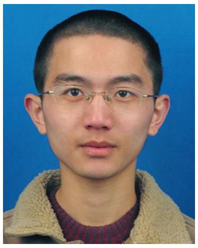
[55] "Ethereum 2.0," Accessed: Jan. 2022. [Online]. Available: https://ethereum.org/en/eth2/

[56] C. Kim, "Confessions of a sharding skeptic," 2020, Accessed: Jan. 2022. [Online]. Available: https://www.coindesk.com/sharding-eth-2-podcast

Kaiping Xue (Senior Member, IEEE) received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, China, in 2003, and the Ph.D. degree with the Department of Electronic Engineering and Information Science, USTC, in 2007. From May 2012 to May 2013, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA. He is currently a Professor with the School of Cyber Science and Technology, USTC. He has authored and coauthored more than 100 technical papers in various archival journals and conference proceedings. His research interests include next generation Internet architecture design, transmission optimization, and network security. He is on the Editorial Board of several journals, including the IEEE TRANSACTIONS OF DEPENDABLE AND SECURE COMPUTING (TDSC), IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (TWC), and IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT (TNSM). He was also a (Lead) Guest Editor for many reputed journals/magazines, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), *IEEE Communications Magazine*, and *IEEE Network*. He is an IET Fellow.

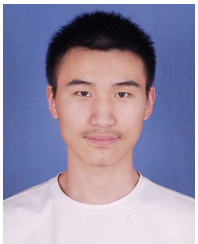
Xinyi Luo (Graduate Student Member, IEEE) received the bachelor's degree in information security from the School of the Gifted Young, University of Science and Technology of China (USTC), Hefei, China, in 2020. She is currently a Graduated Student with the School of Cyber Science and Technology, USTC. Her research interests include network security and cryptography.

Hangyu Tian received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, China, in 2018, and the master's degree in communication and information system with the Department of Electronic Engineering and Information Science, USTC, in 2021. He is currently an Engineer with Ant Financial Services Group, Hangzhou, China. His research interests include network security and cryptography.



Jianan Hong received the bachelor's degree from the Department of Information Security, University of Science and Technology of China (USTC), Hefei, China, in 2012, and the Ph.D. degree with the Department of Electronic Engineering and Information Science, USTC, in 2018. During 2018–2021, he was a Research Engineer with Huawei Shanghai Research Institute, Shanghai, China. He is currently an Assistant Researcher with the School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests include secure cloud computing and mobile network security.

David S.L. Wei (Senior Member, IEEE) received the Ph.D. degree in computer and information science from the University of Pennsylvania, Philadelphia, PA, USA, in 1991. From May 1993 to August 1997, he was on the Faculty of Computer Science and Engineering, University of Aizu, Aizuwakamatsu, Japan, as an Associate Professor and then a Professor. He is currently a Professor with Computer and Information Science Department, Fordham University, Bronx, NY, USA. He has authored and coauthored more than 100 technical papers in various archival journals and conference proceedings. His research interests include cloud computing, Big Data, IoT, and cognitive radio networks. He was the Guest Editor or Lead Guest Editor for several special issues in the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON CLOUD COMPUTING, and IEEE TRANSACTIONS ON BIG DATA. He was also an Associate Editor for the IEEE TRANSACTIONS ON CLOUD COMPUTING during 2014–2018 and *Journal of Circuits, Systems and Computers* during 2013–2018.



Jian Li (Member, IEEE) received the B.S. degree from the Department of Electronics and Information Engineering, Anhui University, Hefei, China, in 2015, and the Ph.D degree with the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2020. From November 2019 to November 2020, he was a Visiting Scholar with the Department of Electronic and Computer Engineering, University of Florida, Gainesville, FL, USA. He is currently a Postdoctoral Researcher with the School of Cyber Science and Technology, USTC. His research interests include wireless communications, satellite networks, and next-generation Internet.