ICPDS: Design and Analysis of PSSOA and Blockchain Fusion Scheme for Cross-Hospital Medical Systems

Yu Wang[1], Xiaolin Qin[1], Ling Xiong[2], Mingxing He[2]*

[1] Chengdu Institute of Computer Applications, University of Chinese Academy of Sciences, China
[2] School of Computer and Software Engineering, Xihua university, China
wy_20212021@163.com, qinxl2001@126.com, xiongling.swjtu@aliyun.com, he_mingxing64@aliyun.com

Abstract

With the rapid development of information technology, medical treatments are gradually transforming into digitalization and informatization. The diverse sources and scopes of the medical data are sensitive and significant. As important data assets, medical data are constantly being analyzed and mined. Because of the huge commercial value of medical data, the problems of privacy leakage exist in the centralized information system. They cannot adapt to complex medical environment. Most of the current authentication mechanisms use passwords for verification. It is not easy for users to remember all usernames and passwords. Based on these problems, we design single-sign-on (called SSO) authentication (Password Single-Sign-On Authentication scheme, PSSOA) and blockchain (Improved Cross-Blockchain based Privacy-Preserving Date Sharing scheme, ICPDS) fusion schemes. Users are verified by multiple identity servers in this scheme. Then identity servers issue authentication tokens with threshold manners. The protected identity servers regularly update their master secret shares to defend against malicious adversaries.

Keywords: Blockchain, PSSOA, ICPDS, Medical data, Threshold

1 Introduction

Medical data are significant data assets. They can show the users' personal health status. Current medical data are difficult to share among different organizations, which are hindrances for medical diagnoses. However, the exchange and sharing of information between different organizations can bring more value to medical data. In the future, there will be increasing needs for cross-regional and cross-institutional medical information sharing.

Electronic health records [1] are collections of information about medical activities. However, with the explosive growth of data, the risks of medical data leakage are constantly increasing [2-4]. Many hospital information systems have problems such as low security capability and substandard technical measures. When the hospital systems are attacked, private information will be leaked. Cloud computing platforms can provide powerful computing capabilities for medical data users. However, the services model of directly sharing data have brought a series of privacy security issues and data ownership issues.

As public decentralized distributed ledgers, blockchain [5] has the characteristics of multi-party maintenance and non-tampering [6]. It is beneficial to solve the privacy problems in the process of medical data sharing. Blockchain can effectively prevent data from being maliciously tampered with or abused by third parties and reselling. However, in the blockchain systems, the transparency of transaction records will significantly increase the risks of privacy leakage, such as analyzing transaction records to obtain user transaction rules [7]. There are no unified managers in the blockchain, and the adopted information transfer mechanisms and consensus mechanisms also bring new opportunities and challenges to privacy protection.

Based on blockchain technologies, we have proposed electronic health records based on cross-chain privacy-preserving data sharing in my preliminary works [8]. The schemes are made up of two blockchains. In this scheme, we came up with the concept of cross-chain technology. Wanchain is a branch of cross-chain technologies. Wanchain was established as a fork of Ethereum. Compared with Ethereum, Wanchain supports private transactions and cross-chain transactions, which is suitable for solving the current medical situation. However, in this scheme, we didn't consider the problems of user identity registration, identity authentication and dynamic key management.

Password is a common user authentication method in mobile system [9]. This authentication methods provide a unique identity identification for users [10-11]. However, it is very difficult for users to enroll an ID in different service providers and know the username and password from memory. To improve this situation, service providers can authenticate users through identity servers. Users provide unique identity information to identity servers. After being authenticated, the users obtain an authentication token and use the token [12] to request services from the service providers. This method, known as password-based SSO authentication. With the basis of those technologies, we design a method of fusion of blockchain and password-based SSO authentication schemes.

While password-based SSO authentication has been well-received in the application world, critical security issues have also been raised against this scheme. Existing

*Corresponding Author: Xiaolin Qin; E-mail: qinxl2001@126.com
DOI: 10.53106/160792642023032402021

researches mainly have the single point of failure problem. When the identity servers are destroyed, the adversaries can forge tokens without verification [13]. The users' passwords could be leaked from compromised identity servers. The compromised passwords have significant commercial value to the adversaries. Because most users have a habit, for the convenience of self-remembering, the passwords set by users in different systems will have high similarity or the same situation. The adversaries can analyze the password habits of the users through the leaked passwords, so as to obtain the passwords of the users in different systems. Through the obtained passwords, malicious operations such as stealing, tampering and deleting the users' private information are performed to seek greater benefits.

Recently, a password-based threshold SSO authentication which proposed in [14] to prevent permanent leakage of mobile users. Users are verified by multiple identity servers in this scheme. Identity servers update their master secret shares at regular intervals. The master secret shares whose number is the threshold are collected from different epoch by adversaries. It's impossible to forge valid tokens. These solutions can avoid privacy leakage in user identity registration and authentication. The contributions of thesis are as follows:

Application scenarios are designed. The application scenarios are shown in Figure 1. Based on the double-chain structure of scheme CPDS [8], we propose an improved CPDS called ICPDS. The schemes consist of three blockchains. The first blockchain is made up of hospitals (with cross-chain capability) and medical institutions or clinics affiliated with the hospital (without cross-chain capability). The second blockchain is made up of the hospitals' internal nodes. The aim of this blockchain is to select honest and trustworthy cross-chain nodes. The third chain is Wanchain formed by the cross-chain nodes of each hospital. This blockchain is used to cross-chain operation.

To realize the security protection of user information registration, we need to design a protection scheme during the user registration stage. Inspired by [14], We incorporate password-based threshold SSO authentication schemes into the ICPDS. Users are verified by multiple identity servers in this scheme. Then identity servers issue authentication tokens with threshold manners. Protected identity servers regularly update their master secret shares to deter adversaries who can permanently destroy the master secret share.

Safety certificates are designed in the scheme. We prove that the schemes are safe by method of mathematical proof. Those schemes provide strong security for user registration. We also design an experiment. Experiments show that the schemes can resist various kinds of malicious attacks. We also conduct comprehensive performance analysis. The experimental results show that this scheme has great advantages in storage cost and so on.

The overall frameworks of the thesis are shown below. In Section 2, we list some of the current research results. We analyze those results. In view of the shortcomings of current research results, we raise questions and put forward solutions. In Section 3, we describe the design and implementation in detail. In Section 4, we design the experimental analysis of our proposed scheme, which shows the superiority of our scheme. In Section 5, we analyze the security of the scheme. Through mathematical proof and analysis of attack types, we prove that our schemes are safe. In Section 6, we summarize the scheme.

2 Related Works

2.1 Healthcare Blockchain

In recent years, blockchain technologies and medical fields have developed vigorously. Based on the characteristics of blockchain, it can deal with the problems of data storage and privacy protection in the medical field. Through blockchain technologies, medical data can be stored and shared among different entities [15].

In terms of data storage and access control. [16] first focused on thinking about blockchain technology. It introduces blockchain technology to address the feasibility of Electronic Medical Records. [17] proposed a new blockchain-based trust model. [18] analyzed the feasibility of applying blockchain technology to a storage scheme for medical data to protect patient privacy data. [19] designed a blockchain-based medical data privacy protection system DPS. [20] proposed the idea of using blockchain as a shared data access control management system. The schemes combine the multi-chain platform and the encrypted CP-ABE access control strategy [21] based on the ciphertext policy attribute to realize the proof of concept. But using blockchain technology offers more advantages.

In terms of addressing issues such as single points of failure and privacy breaches. A data sharing and access control framework proposed in [22]. [23-24] proposed two classes of attribute-based access control systems. However, the schemes are generally inefficient. [25] proposed a novel deep learning architecture named. The entropy theory is used to measure the uncertainty of direct trust values in [26]. [27] proposed a novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection.

In terms of data security sharing and privacy protection. Cryptographic algorithms or security models are often used to solve this problem [29-39]. After the emergence of blockchain, many scholars believe that this technology is a good solution to solve the problem [40-47]. [48] proposed a cross-blockchain based EHR Privacy-preserving scheme, which uses relay-chain to achieve secure access to EHR data when patients visit different hospitals. [49] proposed a new term for block cloud. Blockchain technologies are used to share medical data in a cloud environment. This approach addresses the challenges of how healthcare providers and organizers, public health agencies, healthcare providers, and governments need to collaborate and develop measures for policy implementation [50-55]. This solutions can effectively store and manage electronic medical data in the cloud environment, but the implementation cost of medical system is unknown.

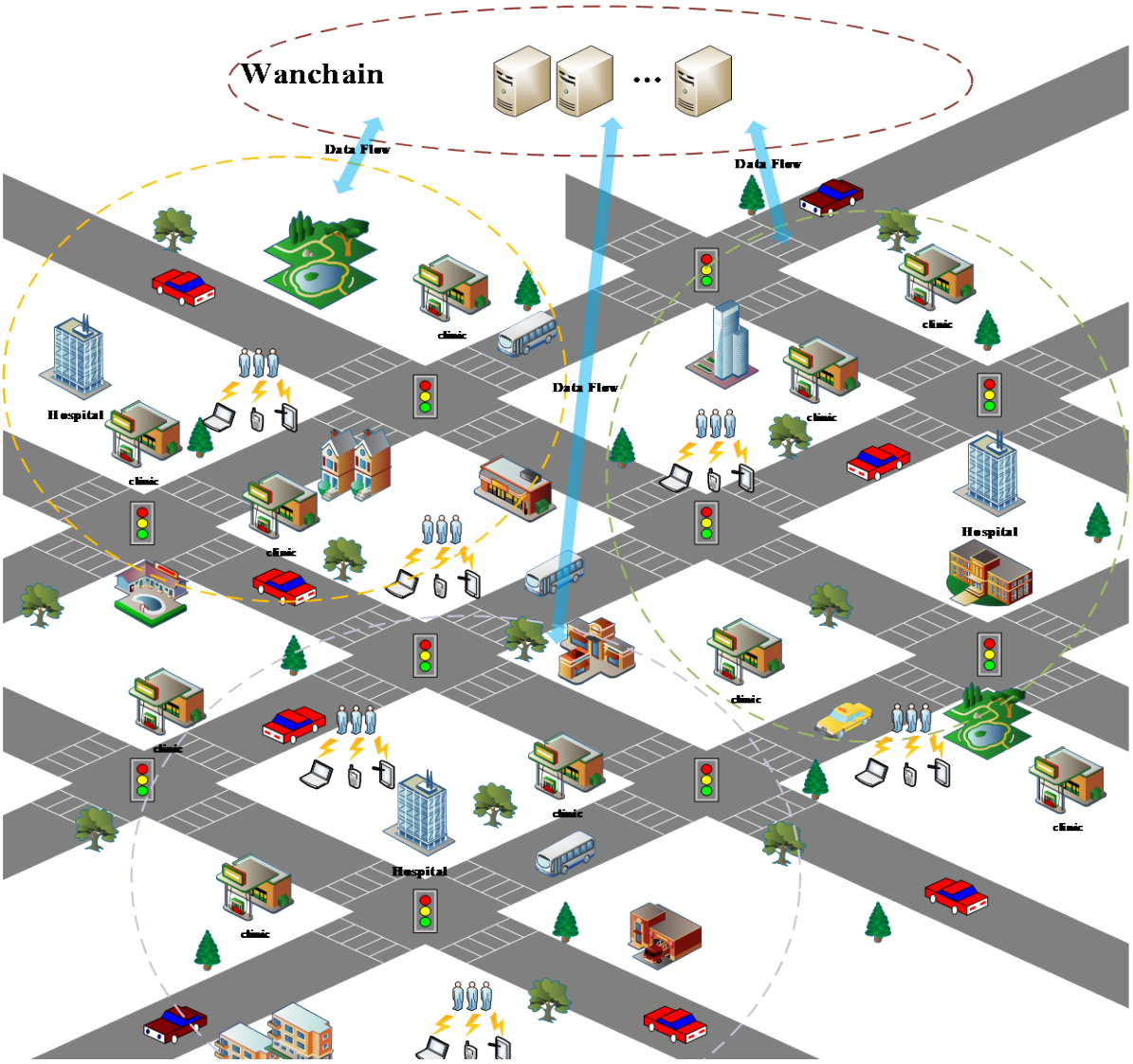

Figure 1. Application scenario

2.2 Password-based SSO Authentication

In the blockchain systems, the users need to register an account on the blockchain. During this process, we found that "how to ensure the privacy and security of blockchain accounts" is an urgent problem to be solved. Passwords are an important method we use in our daily life to keep our accounts safe. Recent years, the researches of domestic and foreign literature show that password-based SSO authentication is one of the most significant application methods in the mobile environment. As far as identity authentication technologies are concerned, there are many authentication methods currently, which mainly utilize other authentication factors [56-63].

In the current research schemes, in order to provide the security of identity authentication, [12] and [58] proposed security enhancement research from the aspects of identity authentication tokens and passwords. But these schemes still have problems. They can directly extract the user's password information directly from the identity servers' database. Most of these adversaries come from malicious workers of service providers. To solve this situation, one of the existing common methods is to generate and send tokens to multiple identity servers in threshold manners [64-67]. In this way, the security of the tokens are guaranteed as long as adversaries compromise no more than identity servers whose number is the threshold. Integrating this schemes into password-based SSO authentication schemes also face the problem of password leakage [68]. In order to prevent the users' passwords from being leaked, the traditional methods are that the identity server only stores the hash value of the password [69-70], rather than the full passwords. Use the hash value as a credential for user authentication. This methods guarantee that even if the hash values are leaked, the adversaries cannot recover the passwords. However, unique passwords are inherently low entropy. Adversaries can retrieve user passwords via dictionary guessing attacks (DGA) [71-72]. Adversaries can calculate the hashes of all candidate passwords and identify the hashes that match the target password.

To ensure the security of passwords and authentication tokens, [73] introduced a concept based on password-threshold (PASTA) [74]. The generated tokens are distributed to multiple identity servers. The authentication credentials stored in each identity server are calculated from the server-

derived password. During the authentication stage, the users first interact with the identity servers and obtain the passwords by retrieving the server-derived passwords. The interactions don't reveal any information about passwords to the identity servers. Only if the users have the correct passwords, can they calculate a valid server-derived password. Then, the users calculate the corresponding credentials to authenticate. However, this methods are vulnerable to online password testing attacks and permanent compromise of the key share on the identity server. [14] proposed a secure and efficient password-based threshold single-symbol authentication scheme. The schemes can resist online password testing attacks, and can also resist long-term and permanent intrusion of certain identity servers. Based on those schemes, we decide to design fusion scheme to realize the secure management and storage of user identity and privacy information.

3 Design and Implementation

3.1 The Construction of Scheme ICPDS

This scheme ICPDS is a three-chain structure. The solutions consist of three blockchains. The first blockchain is made up of hospitals (with cross-chain capability) and medical institutions or clinics affiliated with the hospital (without cross-chain capability). Hospitals, medical institutions, and clinics store all user information. The main functions are to collect and store user information. Users register their personal information through the blockchain. The second blockchain is made up of the hospital's internal nodes. The aim of this blockchain is to select honest and trustworthy cross-chain nodes. Nodes with cross-chain capabilities are spread across the hospitals' blockchain network. The hospitals' internal blockchain use POS consensus algorithm to select cross-chain nodes. The third blockchain is a multi-dimensional chain formed by the cross-chain nodes of each hospital. The blockchain is mainly used for cross-chain transactions of users' information. When the users have the need of cross-hospital treatment, the users' medical records can be transferred to hospitals through the blockchain. When the transfers are successful between two hospitals, the original blockchain no longer keeps users' information. The network models are shown in Figure 2.

3.2 Scheme of PSSOA

3.2.1 Entity Objects Involved

Users (blockchain account): The scheme consists of a set of users. There are no connections between different users. Each user has a mobile device, a unique identity identifier ID_u and a password psw_u. The users need to request authentication tokens from the identity servers. In this scheme, the users' passwords are unique and the users' authentication tokens are stored in the mobile device that the users use to access.

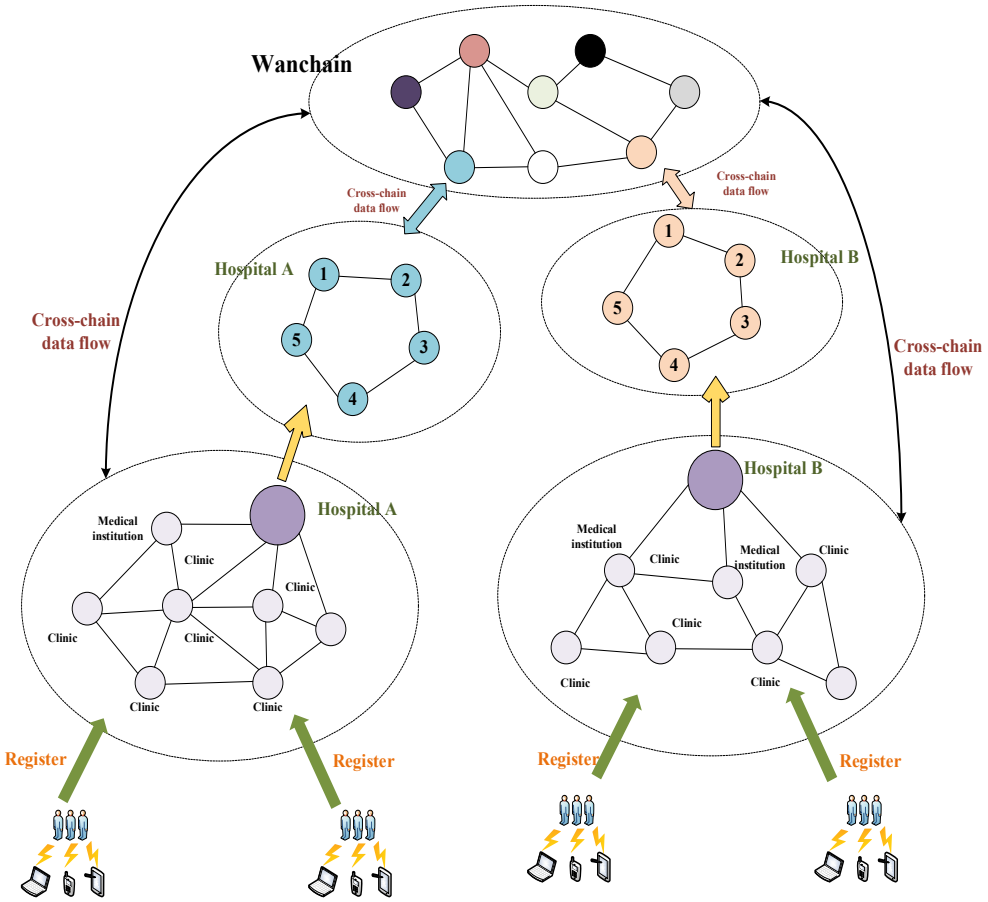

Figure 2. Network model of ICPDS

Identity servers (consensus node): The schemes contain a set of identity servers, which denote as $\{IS_1, IS_2,..., IS_n,\}$. The identity servers verify the users by checking whether the users have the correct password.

Application servers (local database): Application servers are under the jurisdiction of the mobile service provider. They authenticate users with identity servers. If the users are authenticated, the mobile services can be requested from the application servers.

3.2.2 Users Authentication Process

Identity authentication is mainly performed when users go to the hospital for medical treatment. A blockchain account ID_u needs to be created on the corresponding blockchain. The specific process of identity authentication is shown in Figure 3. Identity authentication includes registration and identity authentication processes:

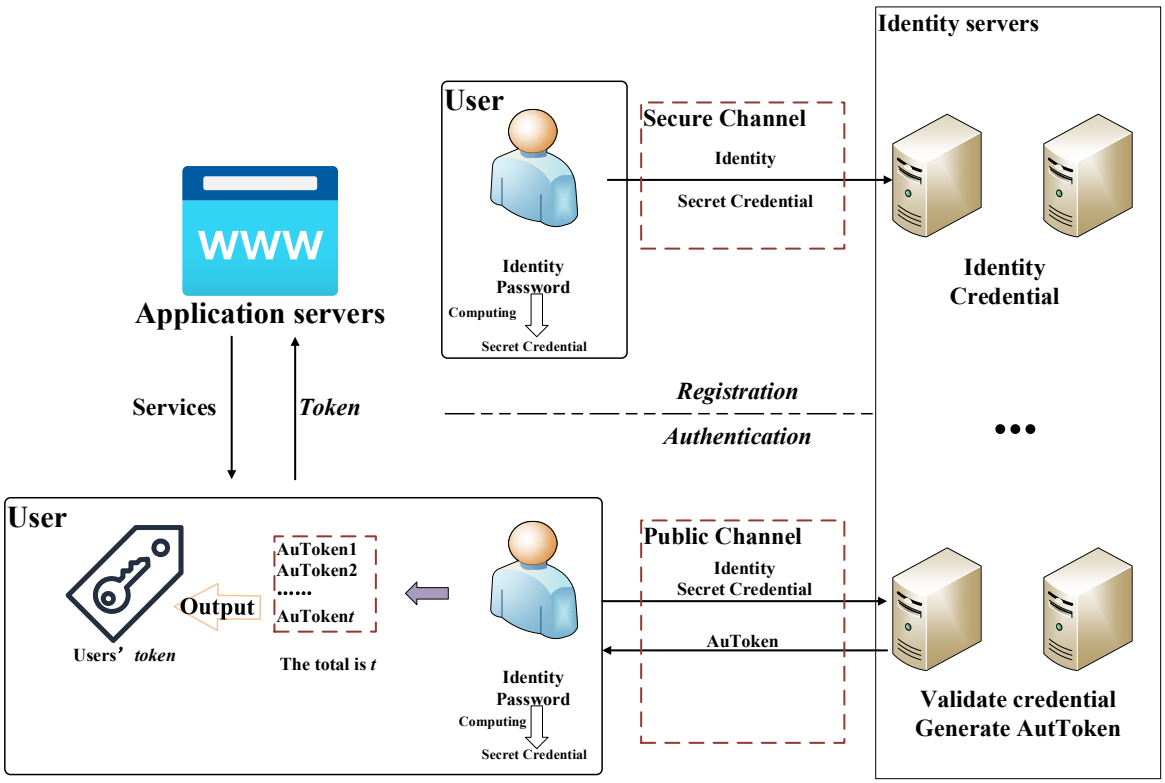

Figure 3. The process of PSSOA

Registration. During this stage, master keys are shared by identity servers share in a (t, n)-threshold method, and every identity server has a master key share. Users generate an identity ID_u and a unique password psw_u to register. Based on psw_u , users calculate secret credentials for subsequent authentication. After this stage, every identity server maintains ID_u and corresponding credentials, and users need to remember their passwords for subsequent authentication. In this stage, secure channels are built between the users' mobile device and every identity server with protocols.

Authentication. The credential is first calculated by users, then the identity and credential are sent to the identity server by mobile device. Credentials are checked for validity by each identity server. If the check is successful, the authentication token shares generated by identity servers with their master secret shares and send them to users. After receiving authentication token shares whose number is the threshold, valid authentication tokens can be reconstructed. The users can request mobile services with these tokens from the mobile application servers that store the public parameters.

Program process:

Stage 1: Registration. System parameters are generated and the users register with the identity servers.

Setup. We define a set of system parameter SP. The description of SP is shown in Table 1.

MasterKeyGen. The identity servers IS_i use the public parameters to generate master secret share msk_i by executing protocols. As shown in Algorithm 1, where the master key denotes as msk, the corresponding public master key denotes as $PMSK$, IS_i's the master key share denotes as msk_i, and the corresponding public master key share denotes as $PMSK_i$.

Table 1. Notation for scheme

Notation	Description
$\mathcal{L}$	Security parameter
p	Order
P	Generator
G	Additive group
Gr	Multiplicative group
e	Bilinear mapping pair
$h, \hbar, H$	Hash function
PRF	Pseudorandom function
Enc	Symmetric-key encryption algorithm

θ	The upper limit of authentication token requests
σ	The upper limit of the user's failure to authenticate to the identity servers
t	Threshold
n	The number of identity servers

Algorithm 1. Distributed secret sharing protocol

Require: Security parameter $\mathcal{L}$, identity servers' indexes {1, ..., n}, threshold number t, the index i of indentity server who executes this algorithm.

Ensure: Identity servers share a secret *msk* and corresponding public key *PMSK*;

For i = 1, 2, ..., n, IS_i calculates a secret share msk_i and the corresponding public share $PMSK_i$.

1. IS_i chooses $c_{i,0} \in Z_p^*$ randomly and a polynomial $f_i(x) = c_{i,0} + c_{i,1}x + ... + c_{i,t-1}x^{t-1}$ over Z_p with degree at most $t-1$ such that $f_i(0) = c_{i,0}$;

2. For $\in$ = 1, 2, ..., $t-1$, IS_i sends $c_{i,0}P$ and $c_{\epsilon,0}P$ to all other identity servers. IS_i sends $f_i(j)$ secretly to IS_j for j = 1, 2, ..., n ; $j \neq i$

3. After receiving $f_i(i)$ from IS_j ($j \in [1, n], j \neq i$), IS_i verfies $f_j(i)P = \sum_{\gamma=0}^{t-1} i^{\gamma} \cdot c_{j\gamma}P$. If the verification fails, IS_i rejects;

4. IS_i calculates the secret share $msk_i = \sum_{\gamma=1}^{n} f_{\gamma}(i)$, the public share $PMSK_i = msk_iP$;

5. IS_i calculates public key $PMSK = \sum_{i=1}^{n} c_{i,0}P$, securely stores msk_i, maintains {$PMSK_1$, $PMSK_2$, ..., $PMSK_n$}, and deletes other values generated above.

6. Here, the secret key $msk = \sum_{i=1}^{n} c_{i,0}$ is distributed to all identity servers but does not appear explicitly in the scheme.

Registration: The process of the user register in the identity server is as follows:

1) U generates ID_u and unique password psw_u. U chooses $r \in Z_p^*$ randomly, calculates $psw_u^{'} = rH(psw_u)$, and sends $(ID_u, psw_u^{'})$ to all IS_i.

2) The IS_i first checks if the ID_u already exists in local database. If the ID_u already exists, the IS_i informs the U and states that the ID_u already exists; otherwise, the IS_i stores the ID_u and determines the U's group number. We assume that the U is the first member in the group, and the IS_i needs to generate a new server-side key.

3) With distributed key share protocol, IS_i generates the server-side key share. The protocol is shown in Algorithm 1, where $c_{i,0}$, $c_{i,1}$, ... , $c_{i,t-1}$ is selected newly. It's different from generating the master key share, server-side key denotes as *sdk* , corresponding public server-side key denotes as *SDPK*, IS_i's server-side key share denotes as sdk_i, and corresponding public server-side share denotes as $SDPK_i$.

4) IS_i calculates $\delta_i^{'} = sdk_i \cdot psw_u^{'}$, and send it to U.

5) U check the validity of $\delta_i^{'}$ as follows:

$$e(\delta_i^{'}, P) = e(psw_u^{'}, SDPK_i). \quad (1)$$

After receiving t valid signature (denoted by $\{\delta_1^{'}, \delta_2^{'}, \cdots, \delta_t^{'}\}$), U calculates as follows:

$$\omega_{\eta} = \prod_{1 \le l \le t, l \neq \eta} \frac{l}{l-\eta}. \quad (2)$$

$$\delta_u = r^{-1} \sum_{\eta=1}^{t} \omega_{\eta} \delta_{\eta}^{'}. \quad (3)$$

U verifies the correctness of δ_u as follows:

$$e(\delta_u, P) = e(H(psw_u), SDPK). \quad (4)$$

Here, based on Lanrange interpolation, $\delta_u = sdkH(psw_u)$ is a signature of psw_u under the server-side key *sdk* , which is generated with a BLS signature.

6) U calculates a server-derived password sdp_u as follows:

$$sdp_u = PRF(h(\delta_u), psw_u). \quad (5)$$

7) U calculates $sdp_i = \hbar(sdp_u \| i)$, send sdp_i to IS_i .

8) IS_i stores sdp_i , and start two counters $\theta_u = 0$ and $\sigma_u = 0$.

9) U deletes δ_u , r , sdp_u , sdp_i ; IS_i deletes $\delta_i^{'}$, securely stores ID_u , sdp_i , sdk_i , θ_u , δ_u , for subsequent authentication of the U.

10) For other users in the same group, the IS_i generates signatures with sdk_i . For the first user in a different group, identity servers repeat the above steps to generate a new server-side key share for it.

After this stage, users need to remember their password to authenticate.

Stage 2: Login. The users request authentication tokens from the identity servers. Without identity servers' assistance, U will not be able to calculate the server-derived password sdp_u .Without sdp_u , U cannot calculate the authentication credential sdp_1 , sdp_2, ..., sdp_n through the authentication of the identity server. Therefore, during the login stage, U interacts with the identity servers to retrieve sdp_u . Then, U calculates the credentials to identify himself. When authenticated, the identity servers will send authentication token shares to U.

Step 1: Retrieve the password sdp_u derived by the server. Retrieve.

1) U inputs password psw_u , chooses $\alpha \in Z_p^*$ randomly, calculate $psw_u^{'} = \alpha \cdot H(psw_u)$, U sends $\{ID_u, psw_u^{'}\}$ to IS_i .

2) After receiving ID_u , IS_i checks $\theta_u \le \theta$ and $\sigma_u \le \sigma$, it aborts if the check fails; if the check is successful, it retrieves the corresponding server-side key share sdk_j , calculates

$\delta_i^{'} = sdk_i \cdot psw_u^{'}$, θ_u increases by 1, chooses $r_i \in Z_p^*$ randomly, sends $\{\delta_i^{'}, r_i\}$ to U.

3) After receiving $\left\{\delta_i^{'}, r_i\right\}$, U verifies its validity as follows:

$$e\left(\delta_i^{'}, P\right) = e\left(psw_u^{'}, SDPK_i\right). \tag{6}$$

If the check fails, U will be rejected.

4) After receiving t valid signature (denoted by $\left\{\delta_1^{'}, \delta_2^{'}, \cdots, \delta_t^{'}\right\}$), U conputes $\xi_k = \prod_{1 \le l \le t, l \ne k} \frac{l}{l-k}$ and $\delta_u = \alpha^{-1} \sum_{k=1}^{t} \xi_k \delta_k^{'}$. U verifies the correctness of δ_u by checking $e(\delta_u, P) = e(H(psw_u), SDPK)$.

5) U deletes $psw_u^{'}$, α, $\delta_i^{'}$; IS_i deletes $psw_u^{'}$.

Step 2: Authentication and generation of authentication tokens.

Authentication: U is authenticated by the authentication server and requests an authentication token.

1) U calculates $sdp_u = PRF(h(\delta_u, psw_u)$ and $sdp_i = \hbar\left(sdp_u \| i\right)$, send $EncSDP_i = Enc(sdp_i, r_i)$ to IS_i.

2) After receiving $EncSDP_i$, IS_i decrypt $EncSDP_i$ and obtain r_i. If decryption fails or r_i is different from r_i that is sent before, the IS_i aborts and increases s σ_u by 1; if decryption is successful, the IS_i calculates $Au_i = msk_i \cdot H($ *Token*), where *Token* represents a message that may contain information and attributes of U, an expiration time, a property with controlled access strategy, and other supporting information. IS_i calculates $EncAuToken_i = Enc$ (sdp_i, $Au_i \|$ *Token*), send $EncAuToken_i$ to U.

3) After receiving $EAuToken_i$, U decrypt it and obtain Au_i and *Token*. U verifies the validity of Au_i by checking e (Au_i, P) = e (H (*Token*), $PMSK_i$) if the check fails, it rejects.

4) After receiving t number of valid signatures (denoted by $\{Au_1, Au_2, ..., Au_t\}$), U calculates $X_v = \prod_{1 \le \varepsilon \le t, \varepsilon \ne v} \frac{\varepsilon}{\varepsilon - v}$ and $Au_u = \sum_{v=1}^{t} X_v Au_v$. U verifies the correctness of Au_u by checking e (Au_u, P) = e (H (*Token*), $PMSK$) .

5) U deletes sdp_u, sdp_i, $EncSDP_i$, r_i, Au_i, $EncAuToken_i$; IS_i deletes $EncSDP_i$, r_i, Au_i, $EAuToken_i$.

6) $AuToken_u$=$\{Token, Au_u\}$ as U's authentication token.

Stage 3: Key Update. Every identity server updates its master secret shares to resist the leakage of the shares. The update of the master secret shares is performed only once in an epoch.

UpdateShare. For every IS_i, at the end of an epoch, it performs the update operation and updates master secret share, the process is as follows.

1) IS_i chooses a polynomial $l_i(x) = d_{i,1}x + d_{i,2}x^2 + \ldots + d_{i,t-1}x^{t-1}$ over Z_p with degree at most $t-1$ randomly.

2) For $\epsilon = 1, 2, ..., t-1$, IS_i send $b_{i,\epsilon}P$ to all identity servers. IS_i send $l_i(j) modp$ to IS_j ($j = 1, 2, ..., n; j \ne i$) secretly.

3) After receiving $l_j(i)$, IS_i check as follows:

$$l_j(i)P = \sum_{\gamma=1}^{t-1} i^{\gamma} b_{j\gamma} P. \tag{7}$$

If the check fails, it aborts.

4) IS_i calculates a new master key share $msk_i^{'}$ as follows:

$$msk_i^{'} = msk_i + \sum_{j=1}^{n} l_j(i). \tag{8}$$

The corresponding public master secret share is $PMSK_i^{'} = msk_i^{'} P$.

5) IS_i deletes $l_i(x)$, $d_{i,\epsilon}$, $l_i(j)$, msk_i .

Finally, IS_i reset θ_u and σ_u . A new epoch begins.

3.1.3 Correctness Proof

Proof. Master secret key is constant.

$$msk = \sum_{i=1}^{n} f_i(0). \tag{9}$$

Assume the updated master secret is as follow:

$$msk^{'} = \sum_{i=1}^{n} f_i^{'}(0). \tag{10}$$

Because:

$$f_i^{'}(x) = f_i(x) + l_i(x). \tag{11}$$

So:

$$msk^{'} = \sum_{i=1}^{n} f_i^{'}(0) = \sum_{i=1}^{n} f_i(0) + l_i(0). \tag{12}$$

As described above:

$$l_i(0) = 0. \tag{13}$$

So, we can get:

$$\begin{aligned} msk^{'} &= \sum_{i=1}^{n} f_i^{'}(0) = \sum_{i=1}^{n} f_i(0) + l_i(0) \\ &= \sum_{i=1}^{n} f_i(0) = msk. \end{aligned} \tag{14}$$

In conclusion, identity servers only update its master scret share. The master secret key is constant.

4 Performance Evaluation

We implement of our scheme by use C++ language and version 5.6.1 of MIRACL library. Users execute the algorithm, all the experiments are conducted on a smartphone (HUAWEI MATE 20) with Android 9.0 system, a Kirin 980 CPU with memory 6 GB. Identity servers execute the algorithms, all the experiments are conducted on a desktop with Intel Core i7-12700 CPU and the clock rate is 4.90GHz and the memory is 32.00 GB.

The schematic diagram of our proposed scheme is shown in Figure 4. We use functions to illustrate the implementation process of the scheme. To make the specification clearer, we define a set of notations. This is shown in Table 2.

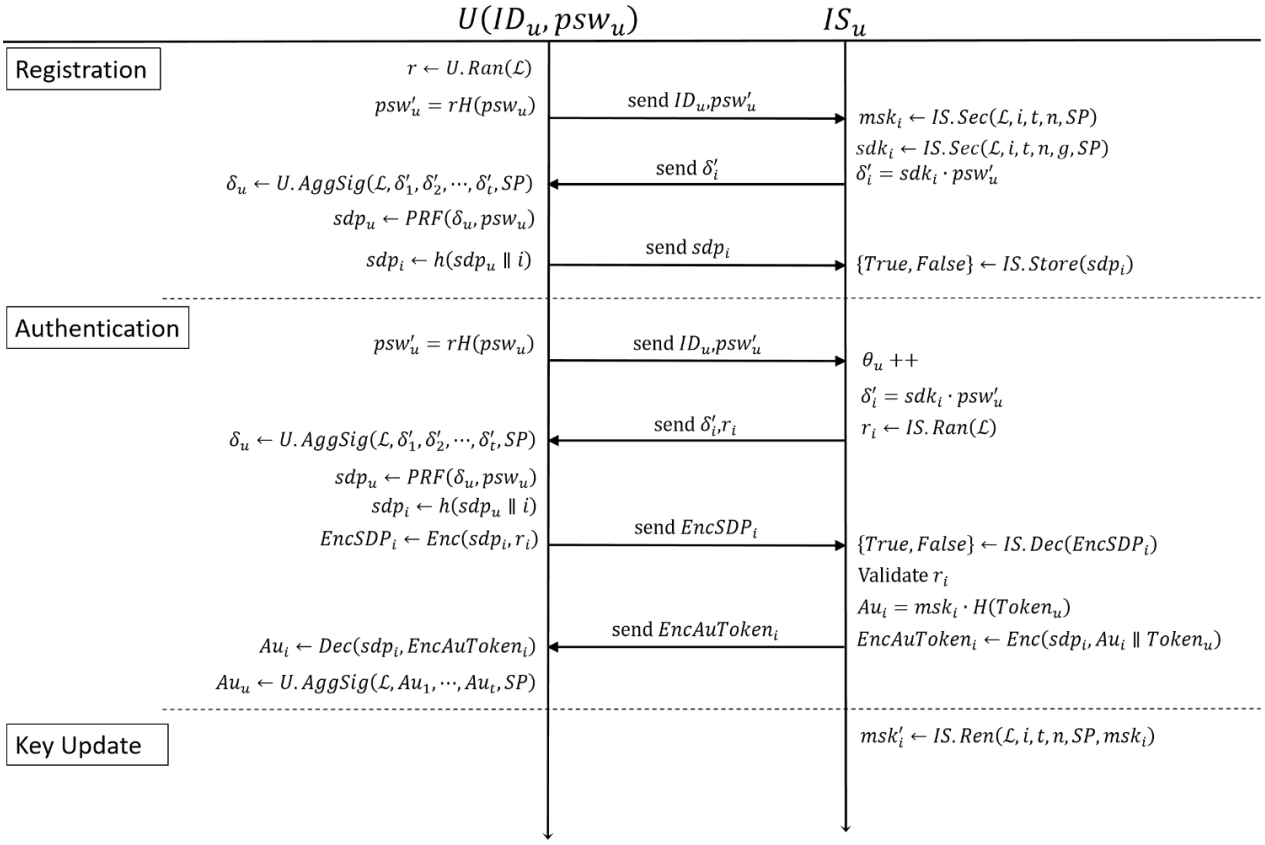

Figure 4. The process of scheme implementation

Table 2. Notation for operations

Notation	Description
U.	Function with *U*
IS.	Function with *IS*
Sec()	Function in Algorithm 1
Ran()	Random function
AggSig()	Aggregate signature
Ren()	Key update function

4.1 Storage Overhead

In our scheme, users need to remember their identity and password. Users can authenticate with unique informations. The users request authentication tokens from the authentication servers. As a result, users can deploy system functions easily.

We pay more attention to storage overhead in the identity server. We represent the storage overhead of the identity server in our scheme in Figure 5 and compare it with PASTA in ICPDS. Through comparison, we find that our proposed scheme can reduce storage overhead by about 45%. Because in PASTA with ICPDS, different users' server-side keys are different, which brings heavy storage overhead to the system. In our scheme, a hybrid mechanism is used in our scheme to solve this problem. The identity servers only need to maintain server-side keys for every group of users, which reduce the storage overhead of the system greatly and save lots of storage costs.

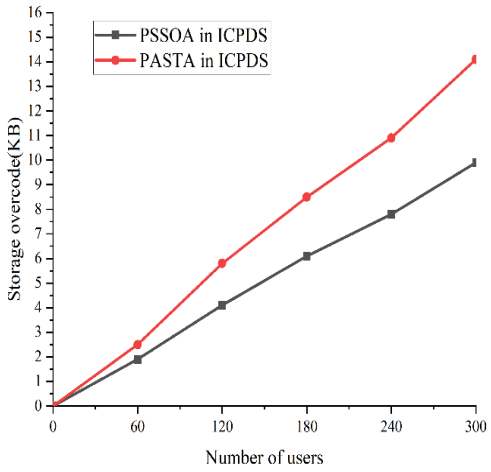

Figure 5. Comparison of storage overhead on the identity server

4.2 Communication Overhead

Our proposed scheme calls for four rounds of communication between user and identity server. In contrast, the PASTA scheme calls for only two rounds of communication. But the extra two rounds of communication in our scheme ensures that the identity server can get the authentication result. During the registration process, the communication cost on the user side of our scheme is the same as PASTA.

In Figure 6, we show the communication cost of our proposed scheme and the PASTA scheme in the authentication stage. Compared with PASTA, our scheme brings more communication costs to users. The users need to communicate with identity servers in the first two rounds of communication. It can ensure the synchronization ofidentity servers. Only in this way, the total number of authentication requests made by users on identity server is the same. In PASTA, this synchronization is not unnecessary. Users only need to communicate with t identity servers. In this way, identity servers in PASTA should assume heavy storage costs to guarantee security. With the proliferation of smartphones, the additional communication cost is acceptable. During the authentication stage of our proposed scheme, there is no need for identity servers to communication with each other. The communication cost on every identity server is constant.

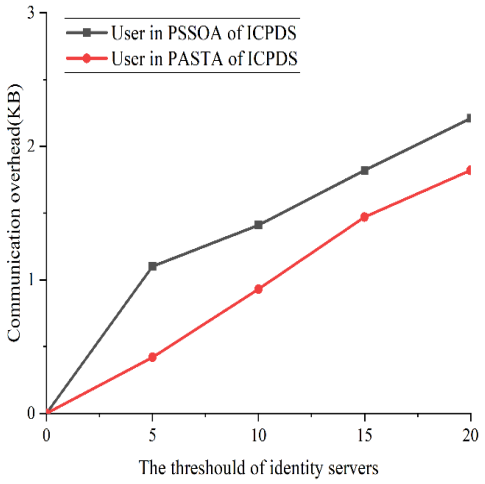

Figure 6. Communication Overhead on the user

4.3 Computation Overhead

We analyze the effificiency of our scheme by counting the number of the point multiplication operation in both G, the exponentiation in G_r, bilinear pairing operation required in each stage, point addition, integer multiplication, hashing, etc. The description of notations are as shown in Table 3. To show the computational effificiency of our proposed scheme, we compare the new scheme with PASTA schemes, where the scheme of CPDS is used to the platform environment.

Table 3. Notation for operations

Notation	Description
TP_{GT}	Time of calculating pairing e
TM_G	Time of multiplication in G
TE_{nc}	Time of symmetric-key encryption
TD_{ec}	Time of symmetric-key decryption
TA_G	Time of group operation in G
TH_G	Time to hash a value into G
TM_{Z_p}	Time of multiplication in Z_p
TH_{Z_p}	Time to hash a value into Z_p
TA_{Z_p}	Time of addition in Z_p
TC_{PRF}	Time of calculating a PRF $PRF(\cdot)$

During the authentication stage of the mobile client, the process of retrieving the server derived password, calculating the credentials used for authentication, and calculating the authentication token incur a corresponding computational cost. Expression for the costs are as follows:

$$\begin{aligned} TA_u &= (4t+4)\cdot TP_{G_T} + (2t+3)\cdot TM_G \\ &+2t\cdot TA_G + (t+3)\cdot TH_G \\ &+(2t-2)\cdot TM_{Z_p} + (4t-4)\cdot TA_{Z_p} \\ &+(t+1)\cdot TH_{Z_p} + 2t\cdot(TEnc+TDec) + TC_{PRF}. \end{aligned} \tag{15}$$

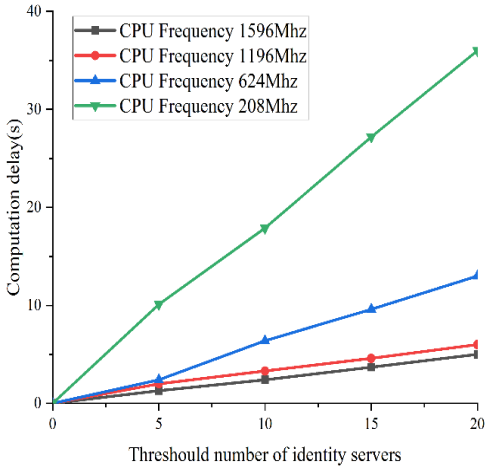

Figure 7. Computation delay on the user

Figure 7 represents the computational latency of the authentication stage of the user on different CPU frequencies of the smartphone. Computational latency on this insensitive smartphone is very low. Specifically, when the identity servers whose number is the threshold t = 5, only one authentication needs to be completed within 3 seconds and a token is obtained for users equipped with smartphones with a frequency of 624MHz. Therefore, our proposed scheme has a good application scenario in the aspect of mobile users.

During the authentication stage, the identity server side's computational costs are constant, and the corresponding computational costs are as follows:

$$TA_{IS_i} = TM_G + TEnc + TDec. \tag{16}$$

During key update stage, identity servers need to calculate a new master key share to update the security protection. The corresponding computational costs are as follows:

$$\begin{aligned} TR_{IS_i} &= t(n-1)\cdot TM_G \\ &+(n-1)(t-1)\cdot TA_G \\ &+(3n-2)(t-1)\cdot TM_{Z_p} \\ &+(n(t-1)+1)\cdot TA_{Z_p}. \end{aligned} \tag{17}$$

Figure 8 shows the computational delay when identity servers update its master key share. We can conclude that when there are 10 identity servers and the threshold t = 10, it takes less than 50ms to update the master key share.

According to the efficiency analysis and performance evaluation, we can conclude that our proposed schemes are more efficient than the PASTA [15] scheme in terms of storage overhead. Furthermore, compared with PASTA, the authentication between the user and the identity server in our proposed scheme requires two additional rounds of communication, which brings a slight communication cost to the users. However, these additional communication costs are mainly to protect our scheme from cryptographic testing. These costs also don't burden current mobile devices with a heavy burden. Therefore, in a mobile environment, these communication costs are tolerable among mobile users. Security analysis and performance evaluation show that our proposed scheme can improve security. Our scheme has greatly advantages over other schemes in terms of performance.

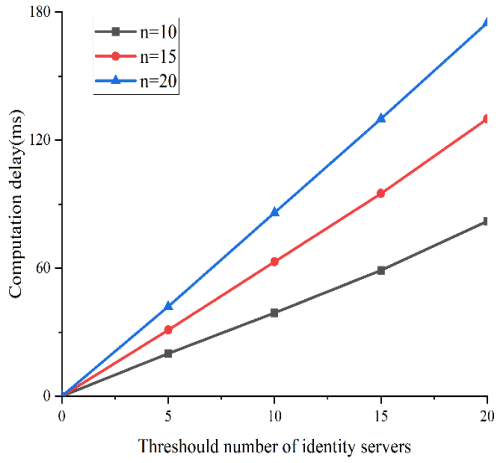

Figure 8. Computation delay on the identity server

5 Security Analysis

5.1 Adversary Malicious Attack

Suppose that the adversaries retrieve the master key msk by collecting any t master key shares. However, the master key msk share is updated periodically. We prove that adversaries that collect master key shares generated in t epochs from two different epochs still cannot successfully retrieve master key msk, and these collected master key shares are denoted as $\{msk_1, msk_2, ..., msk_{t'}, msk'_{t'+1}, mask'_{t'+2}, ..., msk'_t\}$, $(1 < t' < t < n)$. The master key expression is as follows:

$$\begin{aligned} msk &= \sum_{i=1}^{t} \omega_i msk_i = \sum_{i=1}^{t} \omega_i msk'_i \\ &= \sum_{i=1}^{t} \left(\prod_{\substack{1\le\eta\le t \\ \eta\ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\ &= \sum_{i=1}^{t} \left(\prod_{\substack{1\le\eta\le t \\ \eta\ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} \left(f_j(i) + l_j(i) \right) \right). \end{aligned} \tag{18}$$

Adversaries retrieve master key *msk*, The retrieved master key expression is as follows:

$$
\begin{aligned}
msk_A &= \sum_{i=1}^{t_A} \omega_i msk_i + \sum_{i=t_A+1}^{t} \omega_i msk_t' \\
&= \sum_{i=1}^{t_A} \left(\prod_{\substack{1 \le \eta \le t_A \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\
&+ \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} \left(f_j(i) + l_j(i) \right) \right) \\
&= \sum_{i=1}^{t_A} \left(\prod_{\substack{1 \le \eta \le t' \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\
&+ \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) + \sum_{j=1}^{n} l_j(i) \right) \\
&= \sum_{i=1}^{t_A} \left(\prod_{\substack{1 \le \eta \le t_A \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\
&+ \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\
&+ \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} l_j(i) \right) \\
&= \sum_{i=1}^{t} \left(\prod_{\substack{1 \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} f_j(i) \right) \\
&+ \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} l_j(i) \right) \\
&= msk + \sum_{i=t_A+1}^{t} \left(\prod_{\substack{t_A \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} l_j(i) \right).
\end{aligned}
\tag{19}
$$

Because the adversaries cannot collect $l_j(i)$, where j = 1, 2, ..., n, i= t_A + 1, t_A + 2, ..., t, the adversaries cannot calculate *msk* . The expression is as follows:

$$
\begin{aligned}
msk &= \sum_{i=1}^{t_A} \omega_i msk_i + \sum_{i=t_A+1}^{t} \omega_i msk_t' \\
&- \sum_{i=t+1}^{t} \left(\prod_{\substack{t \le \eta \le t \\ \eta \ne t}} \frac{\eta}{\eta - i} \right) \left(\sum_{j=1}^{n} l_j(i) \right).
\end{aligned}
\tag{20}
$$

Therefore, our schemes are capable of resisting adversary attacks, and our proposed schemes are secure.

5.2 Malicious User

There are two types of attacks that malicious users may perform to compromise the security of our schemes. The first attack is to retrieve the victim's password through an online DGA, and the second attack is to authenticate to the identity servers by performing an impersonation attack, and then obtains the authentication token. In our scheme, we use a rate limiting mechanism to limit the number of authentication token requests from users in a certain epoch. This makes it impossible for malicious users to retrieve the victim's password by performing an online dictionary guessing attack. At the same time, the identity servers also need to verify the users' credentials sdp_i . Without the password, it is impossible to calculate the corresponding credentials correctly. Therefore, our schemes are able to defend against malicious users' attacks.

6 Conclusion

In this thesis, we propose a fusion scheme of PSSOA and ICPDS. Through the integration of password-based SSO authentication schemes, our proposed scheme can solve the problem that users cannot easily remember a large number of user passwords. We embed the concept of threshold in the scheme. Each identity server stores a credential built on the users' server-derived password. This credential is used to authenticate the user. In this way, the adversaries cannot retrieve the password from the credentials by performing an offline DGA. Identity servers update their master secret shares termly. The adversaries collect master secret shares whose number is the threshold from different epochs. The adversaries cannot forge valid authentication tokens with master secret shares collected from different epochs. Thus, the security of users' privacy information is ensured. We mathematically prove the security of the scheme. We also conduct comparative experiments. We have done a comprehensive performance evaluation of the program. We prove that this schemes provide stronger security guarantees than existing schemes.

Acknowledgement

This work is supported by Regional Key Project of Science and Technology Service Network Program (STS Program), Chinese Academy of Sciences, No.KFJ-STS-QYZD-2021-21-001. This work is also supported by Science and Technology Planning Project of Sichuan Province, No. 2020YFQ0056.

References

[1] D. Blumenthal, M. Tavenner, The "meaningful use" regulation for electronic health records, *New England Journal of Medicine*, Vol. 363, No. 6, pp. 501-504, August, 2010.

[2] T. Wang, A. Liu, Research on privacy protection of medical information in big data, *Network and Information Security*, Vol. 38, No. 8, pp. 28-32, August, 2019.

[3] Y. Zhou, Brief analysis of network security and privacy protection of medical big data, *Information security and communications privacy*, Vol. 15, No. 9, pp. 28-32, July, 2017. DOI: 10.3969/j.issn.1009-8054.2017.09.007

[4] R. Hu, Y. He, X. Fan, Research on safety technology of medical privacy protection, *Journal of Beijing electronic science and technology institute*, Vol. 26, No. 3, pp. 46-54, September, 2018.

[5] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system，https://nakamotoinstitute.org/bitcoin, October, 2008.

[6] Y. Yuan, F. Wang, Blockchain: the state of the art and future trends, *Acta automatica sinica*, Vol. 42, No. 4, pp. 481-494, April, 2016.

[7] L. Zhu, F. Gao, M. Shen, Y. Li, B. Zheng, H. Mao, Z. Wu, Survey on privacy preserving techniques for blockchain technology, *Journal of computer research and development*, Vol. 54, No. 10, pp. 2170-2186, October, 2017.

[8] Y. Wang, M. He, CPDS: A Cross-Blockchain Based Privacy-Preserving Data Sharing for Electronic Health Records, *2021 IEEE 6th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, Chengdu, China, 2021, pp. 90-99.

[9] L. Wu, J. Wang, K.-K. R. Choo, D. He, Secure key agreement and key protection for mobile device user authentication, *IEEE Transactions on Information Forensics and Security*, Vol. 14, No. 2, pp. 319-330, February, 2019.

[10] L. Lamport, Password authentication with insecure communication, *Communications of the ACM*, Vol. 24, No. 11, pp. 770-772, November, 1981.

[11] R. Chatterjee, A. Athayle, D. Akhawe, A. Juels, T. Ristenpart, pASSWORD tYPOS and how to correct them securely, *2016 IEEE Symposium on Security and Privacy (SP)*, San Jose, CA, USA, 2016, pp. 799-818.

[12] B.-C. Neuman, T. Ts'o, Kerberos: An authentication service for computer networks, *IEEE Communications magazine*, Vol. 32, No. 9, pp. 33-38, September, 1994.

[13] A. Barth, C. Jackson, J. Mitchell, Robust defenses for cross-site request forgery, *The 15th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2008, pp. 75-88.

[14] Y. Zhang, C. Xu, H. Li, K. Yang, N. Cheng, X. Shen, PROTECT: Efficient Password-Based Threshold Single-Sign-On Authentication for Mobile Users against Perpetual Leakage, *IEEE Transactions on Mobile Computing*, Vol. 20, No. 6, pp. 2297-2312, June, 2021.

[15] Y. Cao, F. Jia, G. Manogaran, Efficient Traceability Systems of Steel Products Using Blockchain-Based Industrial Internet of Things, *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 9, pp. 6004-6012, September, 2020.

[16] M. Swan, Blockchain Thinking: The Brain as a Decentralized Autonomous Corporation, *IEEE Technology and Society Magazine*, Vol. 34, No.4, pp. 41-52, December, 2015.

[17] K. Bendiab, N. Kolokotronis, S. Shiaeles, S. Boucherkha, WiP: A Novel Blockchain-Based Trust Model for Cloud Identity Management, *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress*, Athens, Greece, 2018, pp. 724-729.

[18] D. Ivan, Moving Toward a Blockchain-Based Method for the Secure Storage of Patient Records, pp. 1-11, August, 2016. https://www.healthit.gov/sites/default/files/9-16-drew_ivan_20160804_blockchain_for_healthcare_final.pdf.

[19] H. Li, L. Zhu, M. Shen, F. Gao, X. Tao, S. Liu, Blockchain-Based Data Preservation System for Medical Data, *Journal of Medical Systems*, Vol. 42, No. 8, Article No. 141, August, 2018.

[20] M. Jemel, A. Serhrouchni, Decentralized access control mechanism with temporal dimension based on blockchain, *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, Shanghai, China, 2017, pp. 177-182.

[21] J. Bethencourt, A. Sahai, B. Waters, Ciphertext-policy attribute-based encryption, *2007 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, 2007, pp. 321-334.

[22] S. Wang, Y. Zhang, Y. Zhang, A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems, *IEEE Access*, Vol. 6, pp. 38437-38450, June, 2018.

[23] S. Alansari, F. Paci, A. Margheri, V. Sassone, Privacy-preserving access control in cloud federations, *2017 IEEE 10th International Conference on Cloud Computing (CLOUD*), Hononolu, HI, USA, 2017, pp. 757-760.

[24] S. Alansari, F. Paci, V. Sassone, A distributed access control system for cloud federations, *2017 IEEE 37th International Conference on Distributed Computing Systems*, Atlanta, GA, USA, 2017, pp. 2131-2136.

[25] Y. Gao, X. Xiang, N. Xiong, B. Huang, H. J. Lee, R. Alrifai, X. Jiang, Z. Fang, Human action monitoring for healthcare based on deep learning, *IEEE Access*, Vol. 6, pp. 52277-52285, September, 2018.

[26] J. Zhao, J. Huang, N. Xiong, An effective exponential-based trust and reputation evaluation system in wireless sensor networks, *IEEE Access*, Vol. 7, pp. 33859-33869, March, 2019.

[27] W. Zhang, S. Zhu, J. Tang, N. Xiong, A novel trust management scheme based on Dempster–Shafer evidence theory for malicious nodes detection in wireless sensor networks, *The Journal of Supercomputing*, Vol. 74, No. 4, pp. 1779-1801, April, 2018.

[28] A. Fu, X. Zhang, N. Xiong, Y. Gao, H. Wang, J. Zhang, VFL: a verifiable federated learning with privacy-preserving for big data in industrial IoT, *IEEE Transactions on Industrial Informatics*, Vol. 18, No. 5, pp. 3316-3326, May, 2022.

[29] F. Xia, R. Hao, J. Li, N. Xiong, L. Yang, Y. Zhang, Adaptive GTS allocation in IEEE 802.15.4 for real-time wireless sensor networks, *Journal of Systems Architecture*, Vol. 59, No. 10, pp. 1231-1242, November, 2013.

[30] P. Kumar, R. Kumar, G. Srivastava, G. Gupta, R. Tripathi, T. Gadekallu, N. Xiong, PPSF: a privacy-preserving and secure framework using blockchain-based machine-learning for IoT-driven smart cities, *IEEE Transactions on Network Science and Engineering*, Vol. 8, No. 3, pp. 2326-2341, July-September, 2021.

[31] Y. Yao, N. Xiong, J. Park, L. Ma, J. Liu, Privacy-preserving max/min query in two-tiered wireless sensor networks, *Computers & Mathematics with Applications*, Vol. 65, No. 9, pp. 1318-1325, May, 2013.

[32] C. Wu, C. Luo, N, Xiong, W. Zhang, T.-H. Kim, A greedy deep learning method for medical disease analysis, *IEEE Access*, Vol. 6, pp. 20021-20030, April, 2018.

[33] H. Cheng, Z. Xie, Y. Shi, N. Xiong, Multi-step data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM, *IEEE Access*, Vol. 7, pp. 117883-117896, August, 2019.

[34] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, X. Liang, UAV autonomous target search based on deep reinforcement learning in complex disaster scene, *IEEE Access*, Vol. 7, pp. 117227-117245, August, 2019.

[35] S. Huang, Z. Zeng, K. Ota, M. Dong, T. Wang, N. Xiong, An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5G networks, *IEEE transactions on network science and engineering*, Vol. 8, No. 1, pp. 347-365, January-March, 2021.

[36] C. Dai, X. Liu, H. Cheng, L. Yang, M. Deen, Compressing Deep Model with Pruning and Tucker Decomposition for Smart Embedded Systems, *IEEE Internet of Things Journal*, Vol. 9, No. 16, pp. 14490-14500, August, 2022.

[37] S. Cao, S. Niu, G. Xiong, X. Qin, P. Liu, Student Model and Clustering Research on Personalized E-learning, *Journal of Internet Technology*, Vol. 22, No. 4, pp. 935-947, July, 2021.

[38] Y. Zhou, Y. Zhang, H. Liu, N. Xiong, A. Vasilakos, A bare-metal and asymmetric partitioning approach to client virtualization, *IEEE Transactions on Services Computing*, Vol. 7, No. 1, pp. 40-53, January-March, 2014.

[39] W. Fang, Y. Li, H. Zhang, N. Xiong, J. Lai, A. Vasilakos, On the throughput-energy tradeoff for data transmission between cloud and mobile devices, *Information Sciences*, Vol. 283, pp. 79-93, November, 2014.

[40] H. Wang, Y. Song, Secure cloud-based EHR system using attribute-based cryptosystem and blockchain, *Journal of Medical Systems*, Vol. 42, No. 8, Article No. 152, August, 2018.

[41] X. Zheng, R. Mukkamala, R. Vatrapu, J. Ordieres-Mere, Blockchain-Based Personal Health Data Sharing System Using Cloud Storage, *2018 IEEE 20th International Conference on E-Health Networking, Applications and Services (Healthcom)*, Ostrava, Czech Republic, 2018, pp. 1-6.

[42] T. Xue, Q. Fu, C. Wang, X. Wang, A Medical Data Sharing Model via Blockchain, *ACTA Automatica Sinica*, Vol. 43, No. 9, pp. 1555-1562, September, 2017.

[43] B. Shen, J. Guo, Y. Yang, MedChain: Efficient Healthcare Data Sharing via Blockchain, *Applied Sciences*, Vol. 9, No. 6, Article No. 1207, March, 2019.

[44] A. Siyal, A. Junejo, M. Zawish, K. Ahmed, A. Khalil, G. Soursou, Applications of blockchain technology in medicine and healthcare: challenges and future perspectives, *Cryptography*, Vol. 3, No. 1, pp. 1-20, March, 2019.

[45] Z. Shae, J. Tsai, On the design of a blockchain platform for clinical trial and precision medicine, *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, Atlanta, GA, USA, 2017, pp. 1972-1980.

[46] G. Dagher, J. Mohler, M. Milokovic, P. Marella, Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology, *Sustainable Cities and Society*, Vol. 39, pp. 283-297, May, 2018.

[47] F. Li, K. Liu, L. Zhang, S. Huang, Q. Wu, EHRChain: A Blockchain-Based EHR System Using Attribute-Based and Homomorphic Cryptosystem, *IEEE Transactions on Services Computing*, Vol. 15, No. 5, pp. 2755-2765, September-October, 2022.

[48] S. Cao, J. Wang, X. Du, X. Zhang, X. Qin, CEPS: A Cross-Blockchain based Electronic Health Records Privacy-Preserving Scheme, *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1-6.

[49] H. Kaur, M. Alam, R. Jameel, A. Mourya, V. Chang, A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment, *Journal of Medical Systems*, Vol. 42, No. 8, Article No. 156, August, 2018.

[50] X. Li, X. Huang, C. Li, R. Yu, L. Shu, EdgeCare: leveraging edge computing for collaborative data management in mobile healthcare systems, *IEEE Access*, Vol. 7, pp. 22011-22025, February, 2019.

[51] S. Wang, B. Middleton, L. Prosser, C. Bardon, C. Spurr, P. Carchidi, A. Kittler, R. Goldszer, D. Fairchild,

A. Sussman, G. Kuperman, D. Bates, A cost-benefit analysis of electronic medical records in primary care, *The American Journal of Medicine*, Vol. 114, No. 5, pp. 397-403, April, 2003.

[52] H.-T. Wu, C.-W. Tsai, Toward blockchains for health-care systems: applying the bilinear pairing technology to ensure privacy protection and accuracy in data sharing, *IEEE Consumer Electronics Magazine*, Vol. 7, No. 4, pp. 65-71, July, 2018.

[53] A. Yazdinejad, G. Srivastava, R. Parizi, A. Dehghantanha, K. Choo, M. Aledhari, Decentralized authentication of distributed patients in hospital networks using blockchain, *IEEE Journal of Biomedical and Health Informatics*, Vol. 24, No. 8, pp. 2146-2156, August, 2020.

[54] Y. Chen, L. Zhou, S. Pei, Z. Yu, Y. Chen, X. Liu, J. Du, N. Xiong, KNN-BLOCK DBSCAN: Fast clustering for large-scale data, *IEEE transactions on Systems, Man, and Cybernetics: systems*, Vol. 51, No. 6, pp. 3939-3953, June, 2021.

[55] X. Qin, L. Zhang, L. Yang, S. Cao, Heuristics to sift extraneous factors in Dixon resultants, *Journal of symbolic computation*, Vol. 112, pp. 105-121, September-October, 2022.

[56] W. Tang, K. Zhang, J. Ren, Y. Zhang, X. Shen, Flexible and efficient authenticated key agreement scheme for bans based on physiological features, *IEEE Transactions on Mobile Computing*, Vol. 18, No. 4, pp. 845-856, April, 2019.

[57] H. Li, Y. Dai, L. Tian, H. Yang, Identity-based authentication for cloud computing, *First International Conference on Cloud Computing*, Beijing, China, 2009, pp. 157-166.

[58] Q. Jiang, J. Ni, J. Ma, L. Yang, X. Shen, Integrated authentication and key agreement framework for vehicular cloud computing, *IEEE Network*, Vol. 32, No. 3, pp. 28-35, May/June, 2018.

[59] A. Yang, E. Pagnin, A. Mitrokotsa, G. Hancke, D. Wong, Two-hop distance-bounding protocols: Keep your friends close, *IEEE Transactions on Mobile Computing*, Vol. 17, No. 7, pp. 1723-1736, July, 2018.

[60] C. Huang, R. Lu, X. Lin, X. Shen, Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles, *IEEE Transactions on Vehicular Technology*, Vol. 67, No. 11, pp. 11169-11180, November, 2018.

[61] G. Xu, H. Li, Y. Dai, K. Yang, X. Lin, Enabling efficient and geometric range query with access control over encrypted spatial data, *IEEE Transactions on Information Forensics and Security*, Vol. 14, No. 4, pp. 870-885, April, 2019.

[62] Y. Zeng, C. Sreenan, N. Xiong, L. Yang, J. Park, Connectivity and coverage maintenance in wireless sensor networks, *The Journal of Supercomputing*, Vol. 52, No. 1, pp. 23-46, April, 2010.

[63] L. Shu, Y. Zhang, Z. Yu, L. Yang, M. Hauswirth, N. Xiong, Context-aware cross-layer optimized video streaming in wireless multimedia sensor networks, *The Journal of Supercomputing*, Vol. 54, No. 1, pp. 94-121, October, 2010.

[64] M. Abdalla, S. Miner, C. Namprempre, Forward-secure threshold signature schemes, *Topics in Cryptology — CT-RSA 2001*, San Francisco, CA, USA, 2001, pp. 441-456.

[65] A. Boldyreva, Threshold signatures, multisignatures and blind signatures based on the gap-Diffifie-Hellman-group signature scheme, *International Workshop on Public Key Cryptography*, Miami, FL, USA, 2003, pp. 31-46.

[66] I. Damgård, M. Koprowski, Practical threshold RSA signatures without a trusted dealer, *Advances in Cryptology — EUROCRYPT 2001*, Innsbruck, Austria, 2001, pp. 152-165.

[67] R. Gennaro, S. Goldfeder, A. Narayanan, Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security, *Applied Cryptography and Network Security*, Guildford, UK, 2016, pp. 156-174.

[68] C. Diomedous, E. Athanasopoulo, Practical password hardening based on tls, *Detection of Intrusions and Malware, and Vulnerability Assessment*, Gothenburg, Sweden, 2019, pp. 441-460.

[69] R. Canetti, Universally composable security: A new paradigm for cryptographic protocols, *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, Newport Beach, CA, USA, 2001, pp. 136-145.

[70] D. Wang, P. Wang, Offline dictionary attack on password authentication schemes using smart cards, In: Y. Desmedt (Eds.), *Information Security*, Cham, CH, 2015, pp. 221-237.

[71] D. Wang, Z. Zhang, P. Wang, J. Yan, X. Huang, Targeted online password guessing: An underestimated threat, *2016 ACM SIGSAC Conference on Computer and Communications Security*, Vienna, Austria, 2016, pp. 1242-1254.

[72] A. Narayanan, V. Shmatikov, Fast dictionary attacks on passwords using time-space tradeoff, *Proceedings of the 12th ACM conference on Computer and communications security*, Alexandria, VA, USA, 2005, pp. 364-372.

[73] S. Agrawal, P. Miao, P. Mohassel, P. Mukherjee, PASTA: PASsword-based Threshold Authentication, *2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto, Canada, 2018, pp. 2042-2059.

[74] A. Shami, How to share a secret, *Communications of the ACM*, Vol. 22, No. 11, pp. 612-613, November, 1979.

Biographies

Yu Wang received the M.E. degree in School of Computer and Software Engineering from Xihua University, Sichuan, China. He is currently working in Chengdu Institute of Computer Applications, University of Chinese Academy of Sciences, China.

Xiaolin Qin received Ph.D. degree in computer software and theory from Graduate University of CAS, in 2011. From May 2014 to June 2015, he was a postdoctoral fellow at Department of Computer and Information Science, Linköping University, Sweden. He is currently a professor at Chinese Academy of Sciences (CAS) as well as University of CAS, China. He is a CCF senior and ACM/IEEE member.

Ling Xiong received the M.S. and Ph.D. degrees in the School of Information Science and Technology of Southwest Jiaotong University (SWJTU), Chengdu, PR China. She is an associate professor in school of computer and software engineering, Xihua university. She is also currently pursuing the postdoctoral research in the the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), Chengdu, PR China.

Mingxing He received the M.S. degree from Chongqing University and the Ph.D. degree from Southwest Jiaotong University, in 1990 and 2003, respectively. He is currently a Full Professor with the School of Computer and Software Engineering, Xihua University, Chengdu, China. He is a member of the ACM and a Senior Member of CACR.