Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 2, pp. 775–787, DOI 10.12694/scpe.v26i2.4076

RESEARCH ON HETEROGENEOUS CROSS-DOMAIN IDENTITY AUTHENTICATION AND CONTROL IN CLOUD ENVIRONMENT

KAI XU[*], FEIFEI YU[†], ZHI YANG[‡], JIANJUN ZHANG[§], ZHIGUANG SONG[¶], AND SHITAN LIANG[‖]

Abstract. To fulfill the need for cross-domain authentication in a hybrid cloud setting, the study focuses on identity authentication schemes that bridge various password systems, the author proposes a study on heterogeneous cross domain identity authentication and control in cloud environments. Introduce a multi center authentication management mechanism based on PKI to control and track the anonymous identities of users in different password system security domains. In the process of bidirectional authentication between users and cloud service providers, the scheme successfully negotiates session keys and converts anonymous identities across different password systems. Results indicate that the cloud-based cross-domain identity authentication scheme, without certificate signatures, involves three exponential operations during user registration, four exponential operations and three bilinear operations during the initial cross-domain authentication, and three bilinear operations during subsequent cross-domain stages. Meanwhile, the identity authentication scheme based on PTPM and certificateless public key requires three exponential operations during user registration, five exponential operations and three bilinear operations during the initial cross-domain authentication, and three bilinear operations during repeated cross-domain phases. This scheme achieves cross domain authentication in heterogeneous systems and uses lower computation time for dot multiplication and hash operations. Compared to other schemes, it achieves better computational efficiency while completing cross domain authentication in heterogeneous systems, while compared to the EIMAKP scheme, it has better computational efficiency. This approach effectively safeguards against replay, substitution, and man-in-the-middle attacks, ensuring secure cross-domain identity authentication across diverse password systems. It balances robust security measures with computational efficiency, thereby enhancing overall system reliability and integrity.

Key words: Hybrid cloud, Heterogeneous systems, Cross domain authentication, Anonymity, Bidirectional authentication

1. Introduction.

With the rapid development of information technology in the cloud environment, cloud computing can scale, virtualize, automate and centralize services based on its own deployment complexity, network resource sharing, flexibility and portability, and security. Reliable network cloud services can be built through distributed features, and a communication structure between various servers can be established to complete massive cloud computing resource portability applications and access. Users can low-cost intercommunication network services [1]. For network security issues in cloud environments, cloud computing can provide reliable transmission and communication for server users to interact with information based on its distributed characteristics, and try to prevent external malicious attacks and illegal access to resources, playing a certain protective role. Due to the flexible sharing of cloud computing, it has led to the massive use of multi cloud environments and platforms. Through multi cloud environments, cross domain cloud services can be provided, as well as the emergence of private clouds [2]. The server side can control and lock the data centers of its cloud providers, making online cloud resource services have the characteristic of connectivity. This has brought great convenience to people's lives and the development of the online economy [3]. However, while generating benefits, it also makes many criminals eager to attack and break down network services to obtain greater illegal benefits, making network interactions more complex and increasing the risk of security breaches. The annual increase in network security breaches is 12% year-on-year, making our personal, community, and corporate information more transparent. Nowadays, criminals use illegal means to organize, purposefully, and systematically modify and obtain information data, bringing new huge security challenges to cloud services [4].

[*] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (Corresponding author, KaiXu89@126.com)
[†] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (FeifeiYu96@163.com)
[‡] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (ZhiYang55@126.com)
[§] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (JianjunZhang7@163.com)
[¶] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (ZhiguangSong5@126.com)
[‖] Aostar Information Technologies Co., Ltd., Chengdu, Sichuan, 610000, China (ShitanLiang2@163.com)

Such incidents continue to grow every year. In January 2020, a massive amount of data from a giant cosmetics company was leaked online. The reason for this was that the company publicly disclosed an unsecured database online, which was discovered by security researchers and contained a total of 440336852 records, this includes port numbers, network IP addresses, references, etc. used within the company. Once leaked database information is obtained by malicious individuals for illegal operations, it will cause huge losses to the entire company [5]. In August 2020, shortly after the opening of the New Zealand Securities Exchange, there were several crashes, which not only disrupted the exchange's stock prices and index quotes, but also disrupted its debt market [6]. The reason is that the exchange was attacked by distributed denial of service (DDoS) attacks on its website.

2. Literature Review. In today's cloud computing environment, the mainstream authentication systems mainly include those based on PKI, IBC, and CL-PKC. Among them, the PKI mechanism is widely used due to its mature authentication mechanism, complete structure, and high security. Pradhan, R. et al. proposed an energy aware cloud task scheduling algorithm. It extracts concepts from traditional minimum, maximum, and minimum heuristics and integrates them with energy models. These heuristic algorithms are implemented in heterogeneous cloud environments. The EACTS energy model is designed to assess energy usage in cloud data centers. This algorithm predicts construction time, cloud utilization, and energy consumption based on benchmark data. Through experiments, the EACTS algorithm offers valuable insights into balancing energy efficiency and completion time. It provides a comparative analysis of different scheduling parameters to inform decision-making regarding optimization strategies [7]. Krishnadoss, P. et al. introduced an enhanced seagull optimization algorithm, amalgamating features from both cuckoo search (CS) and seagull optimization algorithm (SOA). This hybrid approach aims to optimize task scheduling in heterogeneous cloud environments by minimizing cost and time parameters. Through comparison with multi-objective ant colony optimization (MO-ACO), ACO, and Min Min algorithms using the Cloudsim 3.0 toolkit, the proposed algorithm's performance was evaluated. Simulation results indicate that the novel seagull optimization algorithm outperforms its counterparts, demonstrating its effectiveness in cloud computing task scheduling [8]. Pradhan, R. et al. introduced a novel approach to optimize task scheduling in cloud data centers, aiming to reduce both duration and energy consumption. This method employs genetic algorithms, where each chromosome represents a scheduling arrangement of independent tasks across available clouds or machines. Fitness functions are utilized to minimize overall execution time, with energy consumption assessed based on the achieved minimum completion time. The effectiveness of this approach was validated through testing on synthetic and benchmark datasets, demonstrating superior performance compared to conventional cloud task scheduling algorithms like Min Min, Max Min, and election heuristic algorithms in heterogeneous multi-cloud systems [9].

The author introduces a novel cross-domain identity authentication scheme tailored for mixed cloud environments, addressing the limitation of existing cloud-based authentication systems in supporting cross-domain authentication between disparate cryptographic systems. This proposed scheme, leveraging Public Key Infrastructure (PKI) and Certificateless Cryptography (CLC), enables secure identity authentication and access between users utilizing CLC and PKI public key cryptographic systems. Introduce a multi center authentication management mechanism based on PKI, with cloud authentication centers as the interaction center for the authentication process, in order to achieve bidirectional cross domain identity authentication between users and cloud service providers, and complete the control and tracking of user identity information throughout the entire process. Utilize temporary identities to achieve anonymity of user identities, and maintain traceability and controllability of user temporary identities and anonymous malicious behavior.

3. Research Methods.

3.1. Preparatory knowledge.

3.1.1. Hierarchical ID Tree Structure. The ID tree structure comprises a hierarchy where the root node represents the identity authentication center's identifier within the security domain, and the leaf nodes denote the identifiers of users and cloud service providers within the same security domain[10]. If the identity of the trusted third-party key generation center (KGC) in the security domain is IDKGC, the user identity in the security domain is $ID_{U_{ser}}$, and the identity form of the user in the security domain is defined as $ID_{KGC}||ID_{U_{ser}}$.

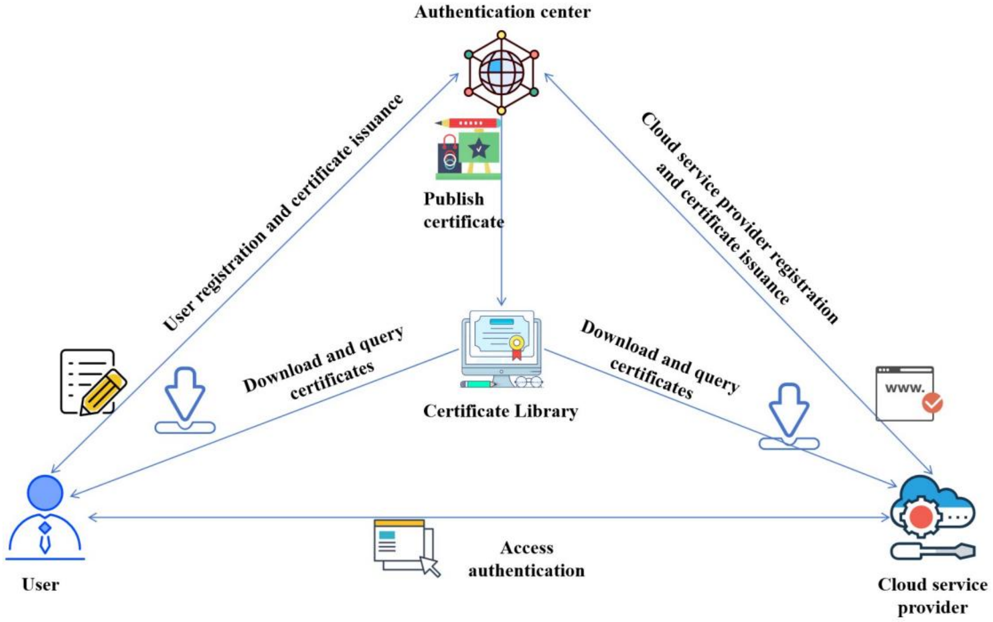
Authentication center
Publish certificate
User registration and certificate issuance
Cloud service provider registration and certificate issuance
Download and query certificates
Certificate Library
Download and query certificates
www.
User
Access authentication
Cloud service provider

Fig. 3.1: Basic structure of PKI system

3.1.2. Related Difficulties and Assumptions.

CDH problem: G is an additive cyclic group of order q, and P is one of its generators. For any unknown $a, b \in Z_q^*$, given $aP, bP \in G$ calculate abP.

CDH assumption: For any algorithm A, there is no probability polynomial that the CDH problem can be successfully solved in time.

DLP problem: G is an additive cyclic group of order q, and P is one of its generators. For any unknown $a \in Z_q^*$, given $aP \in G$, calculate a.

DLP assumption: For any algorithm A, there is no probability polynomial that the DLP problem can be successfully solved in time [11].

3.1.3. Basic composition and structure of PKI system. In the PKI system, the authentication center is mainly responsible for verifying the authenticity of user identity information in the region, managing user digital certificates, and accepting services such as certificate revocation and updates. Users can access the certificate repository through the Lightweight Directory Access Protocol (LDAP) to query or download certificates. The basic structure of the PKI system is shown in Figure 3.1.

3.2. Cross domain identity authentication scheme based on heterogeneous systems.

3.2.1. Cross domain identity authentication model based on heterogeneous systems. The cross-domain identity authentication model based on heterogeneous systems involves four key entities:

1. Authentication Center: This entity, represented by CA (1), handles tasks such as application, issuance, revocation, and querying of certificates within its managed security domain. Additionally, the cloud authentication center CA manages these tasks within various security domains employing different password systems, as well as bidirectional authentication of user identities in different domains and temporary user identity conversion in different password systems;
2. Users U_p, authenticate their identity via the management center within their security domain and then use the cloud authentication center CA to validate their identity's legitimacy. The confirmation result is then relayed to the cloud service provider, enabling authentication of the visiting user through a trusted third party;

3. A cloud service provider provides users with various cloud service resources, which are authenticated by the management center within its own security domain[12]. The legitimacy of the cloud service provider within its security domain is confirmed through the cloud authentication center CA, and the verification results are returned to the visiting user, achieving authentication of the cloud service provider by a trusted third party.
4. The cloud key distribution center is mainly responsible for user authentication and the generation and distribution of partial keys in this security domain, and is responsible for tracing the true identity of malicious users.

For ease of description, let any two clouds be divided into Cloud 1 and Cloud 2. Cloud 1 is based on the PKI system, while Cloud 2 is based on the CLC system. In the initial stage, the cloud authentication center CA authenticates and issues certificates for its managed domains (Cloud 1 and Cloud 2). Users from various security domains initiate access requests to the remote cloud service providers they need to access. Upon receiving the request, the cloud service provider forwards the user's identity details to the cloud authentication center CA for authentication. If the authentication is successful, the cloud authentication center CA creates a compatible identity format for the user within the password system of the cloud service provider and returns it. In case of authentication failure, the result is directly communicated. Concurrently, the user submits the identity information of the cloud service provider to the cloud authentication center CA for authentication. If the authentication results are all passed, a trust connection is established. Although it is necessary to establish trust relationships between cloud service providers and users in various security domains through a cloud authentication center CA, as long as the trust relationship is established, users and cloud service providers no longer need a cloud authentication center CA to provide authentication for this trust relationship. Faced with the increasing number of security domains in cloud environments, adopting multi authentication centers between clouds to solve the security and performance bottlenecks of a single authentication center [13,14].

3.3. Specific Implementation Process of Certification Scheme.

3.3.1. System initialization. The Cloud Identity Management Center CA is responsible for managing the security and other related matters of various system authentication servers, while providing public parameters for the PKI and CLC password systems. For ease of description, $CA^{(1)}$ belongs to the PKI system and $KGC^{(2)}$ belongs to the CLC system; Input security parameters λ. the system selects the qth order additive cyclic group G1 and multiplicative cyclic group G2, defines a bilinear mapping $e : G_1 \times G_2 \to G_2$, selects the generator of group G1 as $P \in G_1$, and selects three secure hash functions H_1, H_2, and H_3; CA randomly selects a system master key $s \in \mathbb{Z}_q^*$ and calculates the system public key $P_{pub} = sP$; Publicly disclose system parameters $= \{q, G_1, G_2, e, P, H_1, H_2, H_3, P_{pub}\}$; $CA^{(1)}$ Randomly select a system master key $s^{(1)} \in \mathbb{Z}_q^*$ and calculate the system public key $P_{pub-1} = s^{(1)}P$; And publicly disclose the system parameters$s_1 = \{q, G_1, G_2, e, P, H_1, H_2, H_3, P_{pub-1}\}$; $KGC^{(2)}$ randomly selects a system master key $s^{(2)} \in \mathbb{Z}_q^*$ and calculates the system public key $P_{pub-2} = s^{(2)}P$; Public parameter $s_2 = \{q, G_1, G_2, e, P, H_1, H_2, H_3, P_{pub-2}\}$ [15].

3.3.2. User Registration. User registration specifically includes $CA^{(1)}$ - PKI User Registration and $PKG^{(2)}$ -CLC user registration.

1) $CA^{(1)}$ - PKI User Registration. If user $u_p^{(1)}$ randomly selects parameter $x_p^{(1)}, r_p^{(1)} \in \mathbb{Z}_q^*$, then the user's private key is $sk_p^{(1)} = x_p^{(1)}$ and the public key is $PK_P^{(1)} = x_p^{(1)}P$, and the user calculates the temporary identity $TID_p^{(1)} = H_1(ID_i||r_p^{(1)}P)$, users download their own root certificate through the certificate repository, extract the public key P_{pub-1}, and read the local timestamp $T_p^{(1)}$.

Send a certificate application $Encrypt(ID_P^{(1)}, ID_{CA}^{(1)}, TID_P^{(1)}, T_P^{(1)}, P_{pub-}, r_p^{(1)})_{P_{pub-}}$ encrypted by P_{pub-1}; $CA^{(1)}$ receives a message and decrypts the application message using its own private key to verify the legitimacy of $ID_P^{(1)}$ identity. It also checks whether $ID_P^{(1)}$ already exists in the registered user list, verifies whether the temporary identity is correct, and verifies the validity of $TID_P^{(1)} \overset{?}{=} H_1(ID_P^{(1)}||r_p^{(1)}P)$ and timestamp $T_P^{(1)}$, if the verification fails, the application failure information will be returned. If the verification passes, $CA^{(1)}$ randomly selects $z_p^{(1)} \in \mathbb{Z}_q^*$ and calculates $Z_P^{(1)} = z_p^{(1)}P$.

Issue certificate $Cert_P^{(1)} = \{m_p^{(1)}, T_{begin}^{(1)}, T_{end}^{(1)}, \delta_P^{(1)}, PK_P^{(1)}, ID^{(1)}, Z_P^{(1)}\}$ for user's temporary identity $TID_P^{(1)}$, among them, $m_P^{(1)}$ is the user's certificate information, $T_{begin}^{(1)}$ and $T_{end}^{(1)}$ are the valid start and end dates of the certificate, $\delta_P^{(1)}$ is the signature information of $CA^{(1)}$ on the user's identity, and $\delta_P^{(1)} = s^{(1)}H_1(m_P^{(1)}||ID^{(1)}) + z_P^{(1)}$. When the authentication center $CA^{(1)}$ first verifies the user's identity as legitimate, it saves the registration list $\{ID_P^{(1)}, TID_P^{(1)}, T_{P1}^{(1)}, r_p^{(1)}P, PK_P^{(1)}\}$ for the user. $\{Cert_p^{(1)}, ID_P^{(1)}, TID_P^{(1)}, T_{P1}^{(1)}\}_{PK_P^{(1)}}$ is the timestamp of the user's certificate issuance time, places the certificate in the certificate repository for storage, and sends the certificate $u_P^{(1)}$ issued to the user $u_P^{(1)}$. The user EE downloads the certificate and verifies its validity [16]. Read the timestamp $T_P^{(1)}$, verify the validity of the timestamp, and determine whether equation 3.1 is valid. If it is not, the certificate will be rejected.

$$\delta_P^{(1)}P \overset{?}{=} P_{pub-1}H_1(m_p^{(1)}) + Z_P^{(1)} \tag{3.1}$$

2) $PKG^{(2)}$ -CLC user registration. Based on the ID tree structure, the identity of cloud service provider $CS_C^{(2)}$ within the $PKG^{(2)}$ system is $ID_{CS} = ID^{(2)}||ID_{CS}^{(2)}$, among them, $ID^{(2)}$ is the identity information of $KGC^{(2)}$, $ID_{CS}^{(2)}$ is the true identity information of $CS_C^{(2)}$, the user randomly selects the secret value $x_{CS}^{(2)}, r_{CS}^{(2)} \in \mathbb{Z}_q^*$, calculates the user's public key $OK_{CS}^{(2)} = x_{CS}^{(2)}$ and temporary identity $TID_{CS}^{(2)} = H_1(ID_{CS}||r_{CS}^{(2)}P)$; User $CS_C^{(2)}$ sends a registration application $Encrypt(ID_{CS}, TID_{CS}^{(2)}, OK_{CS}^{(2)}, r_{CS}^{(2)}P)_{P_{pub-2}}$ encrypted by P_{pub-2} to $KGC^{(2)}$; After receiving user messages, $KGC^{(2)}$ decrypts them using its own system master key $s^{(2)}$ to verify the legality of ID_{CS} identity and the correctness of temporary identity $TID_{CS}^{(2)} \overset{?}{=} H_1(ID_{CS}||r_{CS}^{(2)}P)$, if the verification fails, the application failure message will be returned; If the verification is successful, calculate $Q_{CS} = H_1(TID_{CS}^{(2)})$ and the user's partial private key $d_{CS}^{(2)} = s^{(2)}Q_{CS}$; Read the local timestamp $T_{CS}^{(2)}$ and save the user registration list $\{ID_{CS}, TID_{CS}^{(2)}, PK_{CS}^{(2)}, r_{CS}^{(2)}P, T_{CS}^{(2)}\}$; Return $Q_{CS}, d_{CS}^{(2)}$ to $u_{CS}^{(2)}$ through a secure channel; $ID_C^{(2)}$ calculates the private key $sk_{CS}^{(2)} = x_{CS}^{(2)}d_{CS}^{(2)}$.

3.3.3. PKI $\longleftrightarrow$ CLC cross domain authentication . This scheme uses a cloud based user identity management center (CA) in the cross domain authentication part to establish access identities for verified user identities. While verifying user identities, it also completes session key negotiation and determines the method of establishing access identities through password system identification sent through secure domains. Among them, CL is the identifier of the CLC cryptographic system, and PI is the identifier of the PKI cryptographic system. If the identity information of users or cloud service providers in the PKI domain is verified, it will be completed by the cloud authentication center CA to issue temporary access identities based on the CLC system for users. Similarly, if the identity information of users or cloud service providers in the CLC system is verified, it will be completed by the cloud authentication center CA to issue temporary access identities based on the PKI system for users [17]. Once a temporary access identity is established, users and cloud service providers can no longer rely on trust from cloud authentication centers and establish trust links for cross domain access between different password systems. The specific implementation process of cross domain identity authentication scheme is shown in Figure 3.2.

User $u_P^{(1)}$ randomly selects $n_P^{(1)} \in \mathbb{Z}_q^*$ and calculates the session key negotiation parameter $N_P^{(1)} = n_P^{(1)}sk_P^{(1)}$ $PK_{CS}^{(1)}N_P^{(1)} = n_P^{(1)}P^2$; User $u_P^{(1)}$ enters the temporary identity $TID_{CS}^{(1)}$ and password pw, and randomly selects $Cu_P \in \mathbb{Z}_q^*$; Calculate $w = H_3(TID_P^{(1)}||pw), h_p = H_2(mes_P||TID_P^{(1)}||w||T_P^{(1)}||PK_P^{(1)}||N_P^{(1)}||Cu_P)$, where Cu_P is a random parameter of the session message to maintain the freshness of the message, and T_p is the local timestamp; Send certificate information $\{cert_P, mes_P TID_P^{(1)}, w, N_P^{(1)}, T_P^{(1)}, PK_P^{(1)}, Cu_P, P_{pub-1}\}_{P_{pub2}}$ to $CS_C^{(2)}$.

After receiving the message, $cs_s^{(2)}$ obtains mes_p to determine whether it is an access request. If it is not an access request, it is rejected; Otherwise, according to $TID_P^{(1)}$, read user information from the access user list and verify the validity of the information; If the user information does not exist, upload $\{cert_P, ID_{CS}, L\}_{P_{pub}}$ to CA. CA obtains user $ID_P^{(1)}$ from $cert_P$ and verifies the legitimacy and $\delta_P^{(1)}P \overset{?}{=} P_{pub-1}H_1(m_P^{(1)}||ID^{(1)}) + Z_P^{(1)}$ of $ID_P^{(1)}$, if the verification is successful, a temporary access identity $ID_P = ID^{(2)}||TID_P^{(1)}$ will be established for the user, and $\alpha \in \mathbb{Z}_q^*$ will be randomly selected (Equation 3.2):

$$\delta_P = sH_1(ID_P) + \alpha \tag{3.2}$$

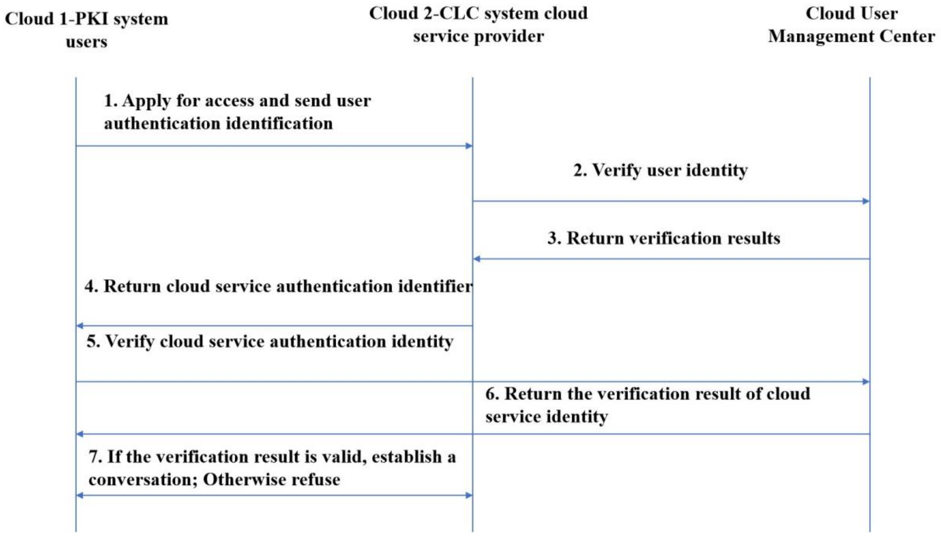
Cloud 1-PKI system users
Cloud 2-CLC system cloud service provider
Cloud User Management Center
1. Apply for access and send user authentication identification
2. Verify user identity
3. Return verification results
4. Return cloud service authentication identifier
5. Verify cloud service authentication identity
6. Return the verification result of cloud service identity
7. If the verification result is valid, establish a conversation; Otherwise refuse

Fig. 3.2: Implementation process of cross domain identity authentication scheme based on heterogeneous systems

Send $Enc\{ID_P, result, \delta_P, \alpha P\}_{P_{pub-2}}$ to $KGC^{(2)}$; If the verification result does not pass, return ”⊥”.

If the $KGC^{(2)}$ judgment result is passed, verify the correctness of the signature $\delta_P \overset{?}{=} P_{pub}H_1(ID_P)+\alpha P$, determine the source of the message, and read the local timestamp T_{local}. In order to ensure security and establish a shorter effective duration of temporary identity access T_{Temp}, a temporary access list $\{ID_P.TID_P, T_{local}, T_{Temp}, T_P^{(1),PK_P^{(1)}}\}$ is established for users

$$K = \frac{M_P^{(1)}p^2}{x_{CS}^{(2)}PK_P^{(1)}} = N_P^{(1)} \tag{3.3}$$

$$h_P = H_2(mes_P||TID_P^{(1)}||w||T_P^{(1)}||PK_P^{(1)}||K||Cu_P) \tag{3.4}$$

Randomly select $f_C \in \mathbb{Z}_q^*$, calculate $F_C = x_{CS}^{(2)}f_CP$ and session key $CK = (h_P + f_C)PK_P^{(1)}x_{CS}^{(2)}$; After receiving the authentication result of the user's identity, $CS_C^{(2)}$ sends a $Encrypt(TID_{CS}^{(2)}, \{ID_{CS}, r_{CS}^{(2)}P\}_{P_{pub}}, d_{CS}^{(2)}, PK_{CS}^{(2)}, F_C, Cu_P)_{PK_P^{(1)}}$ to the user.

When user $u_p^{(1)}$ receives a message, verify whether $(d_{CS}^{(2)} + x_{CS}^{(2)})P \overset{?}{=} P_{pub-2}H_1(TI\ D_{CS}^{(2)}) + PK_{CS}^{(2)}$ is established, if the verification is successful, send $\{\{ID_{CS}, r_{CS}^{(2)}P\}_{P_{pub}}, TID_{CS}^{(2)}, d_{CS}^{(2)} + x_{CS}^{(2)}, PK_{CS}^{(2)}\}_{P_{pub}}$ to CA. After obtaining the message, CA verifies the legitimacy of the $ID^{(2)}$ security domain and checks its certificate validity and $TID_{CS}^{(2)} \overset{?}{=} H_1(ID_{CS}||r_{CS}^{(2)}\ P, d_{CS}^{(2)} + x_{CS}^{(2)})P \overset{?}{=} P_{pub2}H_1(TID_{CS}^{(2)}) + PK_{CS}^{(2)}$. After verification is completed, $\beta \in \mathbb{Z}_q^*$ is randomly selected and signature $\delta_C = sH_1(ID_P) + \beta$ is calculated.

Send $\{ID_P, result, \delta_C\beta P\}_{P_{pub-2}}$ to $CA^{(2)}$; If the verification result does not pass, return ”⊥”; After receiving the result, if the result is valid, $CS_C^{(2)}$ calculates the user session key according to equation 3.5 and establishes the service [18].

$$CK = (h_PPK_{CS}^{(2)} + F_C)sk_P^{(1)} \tag{3.5}$$

3.4. CLC → PKI cross domain authentication. User $u_C^{(2)}$ under the certificateless public key cryptography system CLC wishes to access cloud resource $CS_P^{(1)}$ under the PKI system. Some steps are consistent with

PKI ↔ CLC cross domain authentication, so they will not be elaborated. Only elaborate on the core content of the differences. In the CA authentication section of the cloud authentication center, since at this time, user $u_C^{(2)}$ accessing cloud resource $CS_P^{(1)}$ under the user's uncertified public key cryptography CLC, therefore, at this point, the cloud authentication center CA will verify $u_C^{(2)}$'s identity information and issue a temporary certificate signed by CA for $u_C^{(2)}$, thereby establishing a connection between heterogeneous system user $u_C^{(2)}$ and cloud service providers. The other authentication modes are consistent with PKI ↔ CLC cross domain authentication.

3.5. Repeated cross domain authentication. Due to the establishment of a user identity list during the initial session, there is no need to interact with the cloud identity management center, which reduces the load on the cloud identity management center. The repeated cross domain authentication process is as follows.

(1). User $u_P^{(1)}$ randomly selects $n_{P_i}^{(1)} \in \mathbb{Z}_q^*$ and calculates the session key negotiation parameters $N_{P_i}^{(1)} = n_{P_i}^{(1)} sk_P^{(1)} PK_{CS}^{(1)}$ and $N_{P_i}^{(1)} = n_{P_i}^{(1)} P^2$; User $u_P^{(1)}$ enters temporary identity $TID_P^{(1)}$ and password pw, and randomly selects $C_{u_{p_i}} \in \mathbb{Z}_q^*$.

(2). Calculate $w = H_3(TID_P^{(1)}||pw)$ and $h_P = H_2(mes_P||TID_P^{(1)}||W||T_{P_i}^{(1)}||PK_{P_i}^{(1)}|| N_{P_i}^{(1)}||Cu_{P_i})$, where Cu_{P_i} is a random parameter of the session message, maintaining the freshness of the message, and T_{P_i} is the local timestamp.

(3). Send encrypted authentication information $Enc\{cert_P, mes_{P_i}, TID_{P_i}^{(1)}, w, N_P^{(1)}, T_{P_i}^{(1)}, Cu_{P_i}\}_{PK_{CS}^{(2)}}$ to $CS_C^{(2)}$. After receiving the authentication information, $CS_C^{(2)}$ obtains mes_{P_i} to determine whether it is an access request. Based on the user's temporary identity, $TID_P^{(1)}$ obtains the corresponding user information from the access user list and verifies whether w is the same as the list w saved in the list; Verify if $T_{P_i}^{(1)}$ has exceeded the specified validity period. If the verification is not successful, terminate the verification; Otherwise, calculate $K = \frac{M_P^{(1)} P^2}{x_{CS}^{(2)} PK_P^{(1)}} = N_{P_i}^{(1)}$ and $h_{P_i} = H_2(mes_{P_i}||TID_P^{(1)}||K||Cu_{P_i})$; Randomly select $f_{C_i} \in \mathbb{Z}_q^*$, calculate $F_{C_i} = x_{CS}^{(2)} f_{C_i} P$ and session key $CK = (h_{P_i} + f_{C_i}) PK_P^{(1)} x_{CS}^{(2)}$; After receiving the authentication result of the user's identity, $CS_C^{(2)}$ reads the local timestamp T_{C_i} and sends $Enc(TID_{CS}^{(2)}, \{ID_{CS}, r_{CS}^{(2)} P\}_{P_{pub}}, d_{CS}^{(2)} + x_{CS}^{(2)}, PK_{CS}^{(2)}, F_{C_i}, T_{C_i}, Cu_{P_i})_{PK_{CS}^{(1)}}$ to user $u_P^{(1)}$.

(4). After receiving a duplicate authentication return message, user $u_P^{(1)}$ checks the freshness of T_{C_i} and compares it with the message freshness parameter Cu_{P_i} to see if it is consistent. If the above judgment only fails once, the service will be terminated and the session will be stopped; Otherwise, calculate the session key $CK = (h_P PK_{CS}^{(2)} + F_C) sk_P^{(1)}$ and establish a service.

3.6. Safety and Performance Analysis.

3.6.1. Security analysis.

1) Bidirectional entity authentication. In each security domain, users within the domain achieve mutual authentication between users and resources within the security domain through the original authentication method. In the cross-domain authentication model for heterogeneous systems, users engage in bidirectional authentication with diverse cloud service providers through a request to the cloud identity management center CA. Initially, the cloud service provider authenticates the user's identity, ensuring the security and legitimacy of the user's domain via the cloud identity management center CA. Subsequently, the user authenticates the identity of the cloud service provider, verifying the security and legitimacy of the provider's domain through the same cloud identity management center CA. The authentication outcomes for each domain member are communicated solely by the trusted cloud identity management center CA, thereby completing bidirectional authentication between users and cloud service providers. Additionally, session negotiation keys are established during this bidirectional authentication process. Once the bidirectional authentication is completed, the cloud authentication center no longer provides trust support, reducing the burden on the cloud authentication center [19].

2) Anti replay attack. This scheme adds timestamps and random parameters to maintain session freshness in the cross domain authentication part for the message transmission process of authentication information exchange. Only when the timestamp and freshness parameters of the read message are valid and the random parameters that maintain session freshness during interaction are correct, will the authentication process participants consider the message to be valid. If a malicious attacker wants to replay the intercepted message to a new authentication interaction process to deceive the authentication system, but because the random parameters for maintaining session freshness obtained by the replay attacker from the intercepted authentication interaction message and the new authentication interaction message are different, the replay attacker cannot complete the authentication process through the intercepted authentication interaction message. Therefore, this scheme can effectively resist replay attacks.

3) Anti replacement attack. In this scheme, both w and h_P are bound to the temporary identity of user $u_P^{(1)}$, where the temporary identity $TID_P^{(1)} = H_1(ID_i||r_P^{(1)}P)$ is established based on the user's real identity IDi. In the initial conversation, $\delta_P^{(1)}$ is the signature information of $CA^{(1)}$ for the user's identity, $\delta_P^{(1)} = s^{(1)}H_1(m_P^{(1)}||ID^{(1)} + z_P^{(1)})$ is bound to the identity of $CA^{(1)}$, and $d_{CS}^{(2)} + x_{CS}^{(2)} = s^{(2)}H_1(TID_{CS}^{(2)}) + x_{CS}^{(2)}$ is bound to the identity information of $CS_C^{(2)}$; In repeated cross domain authentication, $w = H_3(TID_P^{(1)}||pw), h_P = H_2(mes_P||TID_P^{(1)}||W||T_{P_i}^{(1)}||PK_{P_i}^{(1)}||Cu_{P_i})$ has bound $u_P^{(1)}$ identity information, if the attacker replaces the user identity in the mutual authentication message, it cannot be verified by the receiving party. Therefore, this scheme can resist substitution attacks.

4) Anti Man in the Middle Attack and Key Security Analysis. Due to the fact that all parameters during interactive authentication are encrypted and sent through the destination public key, not only does it avoid the use of secure channels, but it also prevents intermediaries from obtaining parameters and avoiding intermediary attacks. Moreover, the calculations involved in authentication are based on difficulties such as discrete logarithms to ensure the security of parameter information, enabling effective protection of key negotiation parameters.

5) Traceability of bidirectional anonymity. In the authentication section, both $u_P^{(1)}$ and $CS_C^{(2)}$ use temporary identities $TID_P^{(1)}$ and $TID_C^{(2)}$ to replace the original identity form, achieving bidirectional anonymity between users and cloud service providers. If $u_P^{(1)}$ sends an illegal message, $CS_C^{(2)}$ will submit $TID_P^{(1)}$ to $CA^{(1)}$ in Cloud 1; $CA^{(1)}$ will query the registration list that saves the user's real information when issuing the certificate for the user, and confirm whether $ID_P^{(1)}$'s temporary identity is $TID_P^{(1)}$ again through $TID_P^{(1)} \overset{?}{=} H_1(ID_P^{(1)}||r_P^{(1)}\ P)$, if the verification is successful, it indicates that $ID_P^{(1)}$ is the owner of the temporary identity $TID_C^{(2)}$. When $CS_C^{(2)}$ provides malicious services, send its temporary identity $TID_C^{(2)}$ to the key distribution center $KGC^{(2)}$ of the CLC system; $KGC^{(2)}$ will query the registration list of $CS_C^{(2)}$ using the same calculation method. If verified, it will be determined that IDCS provides malicious services. Therefore, this scheme has bidirectional anonymity and traceability [20].

4. Result analysis.

4.1. Security comparison of cross domain identity authentication schemes. Members of a heterogeneous cryptographic system based on PKI and CLC can achieve bidirectional authentication in a cloud environment and have the feature of anonymous tracking. Throughout the authentication process, this scheme demonstrates robust resilience against replay attacks, man-in-the-middle attacks, and substitution attacks. Table 1 showcases a security comparison between this scheme and several other authentication models, including the identity-based multi-trust domain grid authentication model, wireless body area network anonymous authentication with provable security, and highly secure identity-based authentication key negotiation protocol. The results highlight that this scheme excels in security, as indicated by the checkmark (√) denoting compliance with security conditions, while the "x" symbol represents non-compliance.

4.2. Performance Analysis. For performance analysis, this scheme will be evaluated against EIMAKP, a cloud-based cross-domain identity authentication scheme utilizing certificateless signature, as well as identity authentication schemes based on PTPM and certificateless public key. Additionally, comparisons will be made

Table 4.1: Security comparison results of various cross domain identity authentication schemes

Programme	Anti replay attack	Anti Man in the Middle attack	Anti replacement attack	Mutual authentication	Anonymous tracking
Identity based multi trust domain grid authentication model	√	√	×	√	×
Anonymous authentication in wireless body area networks with provable security	√	√	×	√	×
A highly secure identity based authentication key agreement protocol	√	√	√	√	×
Author's proposal	√	√	√	√	√

with identity authentication schemes designed for multi-server environments. Due to the relatively low computational cost of point multiplication, only the computational costs of bilinear operations and exponential calculations with high computational costs are considered. Table 2 provides insights into computational efficiency, with Te representing bilinear operation time and TE representing exponential calculation time. Checkmarks (√) indicate conditions met, while "x" symbols denote unmet conditions. Regarding computational efficiency, the cloud-based cross-domain identity authentication scheme without certificate signatures involves three exponential operations during user registration, four exponential operations and three bilinear operations in initial cross-domain authentication, and three bilinear operations in repeated cross-domain phases. In comparison, the identity authentication scheme based on PTPM and certificateless public key entails three exponential operations during user registration, five exponential operations and three bilinear operations in initial cross-domain authentication, and three bilinear operations in repeated cross-domain phases. However, the cloud-based cross-domain identity authentication scheme based on certificateless signature, as well as the identity authentication schemes based on PTPM and certificateless public key, and those designed for multi-server environments, have not achieved cross-domain authentication for heterogeneous systems. The EIMAKP scheme achieves cross domain authentication for heterogeneous systems, but requires three bilinear operations when establishing cross domain authentication for the first time. This scheme achieves cross domain authentication in heterogeneous systems and uses lower computation time for dot multiplication and hash operations. When considering computational efficiency and achieving cross-domain authentication in heterogeneous systems, identity authentication schemes in multi-server environments outperform cloud-based cross-domain authentication schemes based on certificateless signatures and those relying on PTPM and certificateless public keys. Additionally, these schemes exhibit superior computational efficiency compared to the EIMAKP scheme.

4.3. Cross domain authentication execution efficiency. In order to verify the execution efficiency of the scheme, the author conducted simulation experiments in a Windows 10 − 64 bit, 16GB memory, 3.2GHz Intel Xeon i7 CPU, and vmware software environment. Multiple 64 bit Ubuntu 18.04 operating system virtual machines were installed in vmware for experimentation. Install the dependent GMP function library (GNU Multiple Precision Arithmetic Library) and glib function library in the Ubuntu virtual machine. Repeat the same experiment 20 times, and take the average value as the final experimental record. This experiment simulates 9 types of users as shown in Table 4.3, representing the authentication efficiency between different security domains.

To comprehensively assess the performance of the cross-domain authentication scheme presented in this chapter, we will analyze its efficiency in both initial and repeated cross-domain authentication scenarios. Furthermore, we will compare the efficiency of repeated authentication with the time consumption of messages of varying lengths. The experimental results will be compared with those of cutting-edge authentication protocol designs, particularly the identity-based cross-domain direct anonymous authentication mechanism and the proxy re-signature-based cross-domain authentication scheme. These protocols have demonstrated superior

Table 4.2: Comparison of computational efficiency of various cross domain identity authentication schemes

Programme	User registration stage	First cross domain authenti- phase	Repeated cross domain authentication -cation stage	cross- -domain authentication
EIMAKP	0	3Te	0	√
Cloud based cross domain identity authentication scheme based on certificate free signature	3TE	3Te+4TE	3TE	×
Identity authentication scheme based On PTPM and certificateless public key	3TE	3Te+5TE	3TE	×
Identity authentication scheme in a multi server environment	0	0	×	×
Author's proposal	0	0	0	√

Table 4.3: Scheme Execution Efficiency (s)

Customer type	User registration stage	Cross domain authentication stage	Repeated cross domain authentication stage
CLC->PKI	0. 018105	0. 058526	0. 00560
CLC->IBC	0. 018266	0. 027001	0. 00451
CLC->PKI,IBC	0. 017885	0. 074546	0. 00820
PKI->IBC	0. 022813	0. 033368	0. 00321
PKI->CLC	0. 023551	0. 037707	0. 00236
PKI->IBC,CLC	0. 023060	0. 423415	0. 00541
IBC->CLC,PKI	0. 017526	0. 083130	0. 00723
IBC->CLC	0. 018003	0. 035640	0. 00432
IBC->PKI	0. 017775	0. 065014	0. 00351

performance in terms of current user computational cost and message interaction rounds, making them highly relevant for cross-domain authentication. The results of these comparisons are depicted in Figures 4.1, 4.2, and 4.3.

Figure 4.1 illustrates that during the initial cross-domain authentication process, the identity-based cross-domain direct anonymous authentication mechanism and the proxy re-signature-based cross-domain identity authentication scheme operate seamlessly within the same authentication system. This eliminates the need for complex identity conversion calculations, streamlining the authentication process. Only the signature algorithm, verification algorithm, and bilinear mapping with high computational cost are required. On the contrary, the proxy re-signature-based cross-domain identity authentication scheme lacks a dedicated public key encryption algorithm, resulting in higher computational costs compared to the identity-based cross-domain direct anonymous authentication mechanism, this scheme does not adopt dual line down mapping, and a temporary identity issuance mode is designed in the identity conversion section, making it close to the proxy signature cost of certificate conversion. In the repeated cross domain authentication stage of this scheme, after the user establishes a session for the first time, the security domain of both parties in the established user access list does not require trust support and computation from the cloud authentication center, effectively reducing computational costs. As depicted in Figure 4.2, the repeated cross-domain authentication costs of the proxy re-signature-based cross-domain identity authentication scheme are consistently lower compared to the identity-based cross-domain direct anonymous authentication mechanism. Additionally, the cost remains

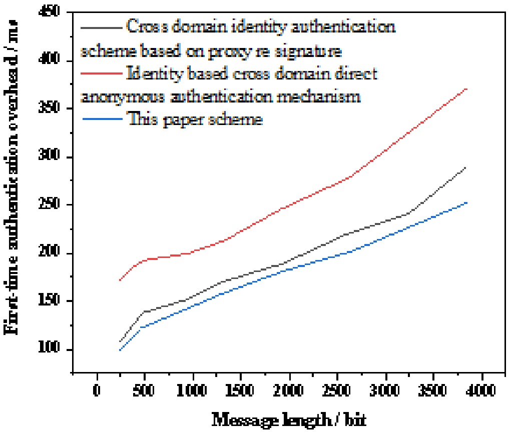

Fig. 4.1: Relationship between first cross domain authentication and message length

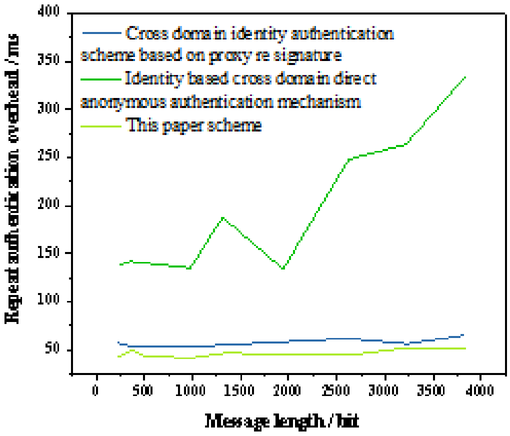

Fig. 4.2: Relationship between repeated cross domain authentication and message length

stable even when dealing with messages of varying orders of magnitude. Furthermore, Figure 4.3 demonstrates that this scheme exhibits higher efficiency in repeated authentication compared to the proxy re-signature-based cross-domain identity authentication scheme, despite the latter's similar efficiency in repeated cross-domain authentication [21].

5. Conclusion. The author proposes a cross domain identity authentication scheme based on heterogeneous systems in a hybrid cloud environment. Taking into account the distribution complexity of current cloud deployment patterns, a heterogeneous password system composed of PKI and CLC is used to manage various security domains and verify and convert temporary user identities across domains through an inter cloud identity authentication center. This scheme surpasses existing cross-domain identity authentication schemes by enabling cross-domain authentication across two types of password systems while maintaining security and achieving higher efficiency. Moving forward, the next phase of research will focus on developing cross-domain au-

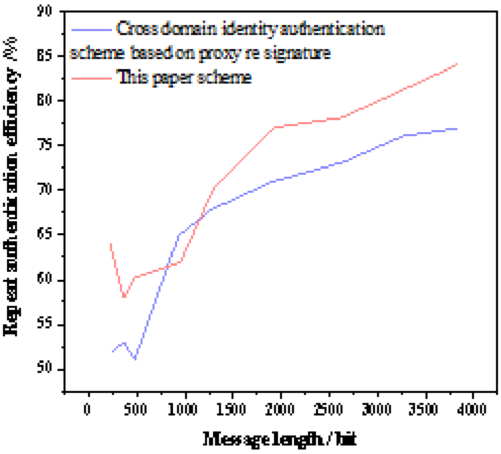

Fig. 4.3: Relationship between the efficiency of repeated cross domain authentication and message length

thentication schemes tailored for heterogeneous systems within cloud environments, without relying on trusted centers.

REFERENCES

[1] Lv, Y., Liu, W., & Wang, Z. (2020). Heterogeneous cross-domain identity authentication scheme based on proxy resignature in cloud environment. Mathematical Problems in Engineering, 2020, 1-12.

[2] Xuan, S., **ao, H., Man, D., Wang, W., & Yang, W. (2021). A cross-domain authentication optimization scheme between heterogeneous IoT applications. Wireless Communications and Mobile Computing, 2021, 1-14.

[3] Chen, J., Zhan, Z., He, K., Du, R., Wang, D., & Liu, F. (2021). XAuth: Efficient privacy-preserving cross-domain authentication. IEEE Transactions on Dependable and Secure Computing, 19(5), 3301-3311.

[4] Chen, R., Shu, F., Huang, S., Huang, L., Liu, H., Liu, J., & Lei, K. (2021). BIdM: A blockchain-enabled cross-domain identity management system. Journal of Communications and Information Networks, 6(1), 44-58.

[5] Wang, M., Rui, L., Yang, Y., Gao, Z., & Chen, X. (2022). A blockchain-based multi-CA cross-domain authentication scheme in decentralized autonomous network. IEEE Transactions on Network and Service Management, 19(3), 2664-2676.

[6] Feng, C., Liu, B., Guo, Z., Yu, K., Qin, Z., & Choo, K. K. R. (2021). Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones. IEEE Internet of Things Journal, 9(8), 6224-6238.

[7] Pradhan, R. , & Satapathy, S. C. . (2022). Energy-aware cloud task scheduling algorithm in heterogeneous multi-cloud environment. Intelligent decision technologies: An international journal, 17, 3789-3800.

[8] Krishnadoss, P. , Poornachary, V. K. , Krishnamoorthy, P. , & Shanmugam, L. . (2023). Improvised seagull optimization algorithm for scheduling tasks in heterogeneous cloud environment. Computer, material, and continuum (in English)(2), 18.

[9] Pradhan, R. , & Satapathy, S. . (2022). Energy aware genetic algorithm for independent task scheduling in heterogeneous multi-cloud environment. Journal of Scientific & Industrial Research, 15(3), 3992-4002.

[10] Feng, C., Liu, B., Guo, Z., Yu, K., Qin, Z., & Choo, K. K. R. (2021). Blockchain-based cross-domain authentication for intelligent 5G-enabled internet of drones. IEEE Internet of Things Journal, 9(8), 6224-6238.

[11] Chen, J., Zhan, Z., He, K., Du, R., Wang, D., & Liu, F. (2021). XAuth: Efficient privacy-preserving cross-domain authentication. IEEE Transactions on Dependable and Secure Computing, 19(5), 3301-3311.

[12] Shen, M., Liu, H., Zhu, L., Xu, K., Yu, H., Du, X., & Guizani, M. (2020). Blockchain-assisted secure device authentication for cross-domain industrial IoT. IEEE Journal on Selected Areas in Communications, 38(5), 942-954.

[13] Tong, F., Chen, X., Wang, K., & Zhang, Y. (2022). CCAP: A complete cross-domain authentication based on blockchain for Internet of things. IEEE Transactions on Information Forensics and Security, 17, 3789-3800.

[14] Xuan, S., **ao, H., Man, D., Wang, W., & Yang, W. (2021). A cross-domain authentication optimization scheme between heterogeneous IoT applications. Wireless Communications and Mobile Computing, 2021, 1-14.

[15] Jia, X., Hu, N., Su, S., Yin, S., Zhao, Y., Cheng, X., & Zhang, C. (2020). IRBA: An identity-based cross-domain authentication scheme for the internet of things. Electronics, 9(4), 634.

[16] Wang, M., Rui, L., Yang, Y., Gao, Z., & Chen, X. (2022). A blockchain-based multi-CA cross-domain authentication scheme

in decentralized autonomous network. IEEE Transactions on Network and Service Management, 19(3), 2664-2676.

[17] Huang, C., Xue, L., Liu, D., Shen, X., Zhuang, W., Sun, R., & Ying, B. (2022). Blockchain-assisted transparent cross-domain authorization and authentication for smart city. IEEE Internet of Things Journal, 9(18), 17194-17209.

[18] Liu, Q., Gong, B., & Ning, Z. (2020). Research on CLPKC-IDPKC cross-domain identity authentication for IoT environment. Computer Communications, 157, 410-416.

[19] Zhang, Y., Luo, Y., Chen, X., Tong, F., Xu, Y., Tao, J., & Cheng, G. (2022). A lightweight authentication scheme based on consortium blockchain for cross-domain IoT. Security and Communication Networks, 2022, 1-15.

[20] Long, W., Wu, C. H., Tsang, Y. P., & Chen, Q. (2021). An end-to-end bidirectional authentication system for pallet pooling management through blockchain internet of things (BIoT). Journal of Organizational and End User Computing (JOEUC), 33(6), 1-25.

[21] Yan, X., Li, L., Chen, J., & Sun, L. (2023). Public key based bidirectional shadow image authentication without pixel expansion in image secret sharing. Frontiers of Information Technology & Electronic Engineering, 24(1), 88-103.

Edited by: Bradha Madhavan
Special issue on: High-performance Computing Algorithms for Material Sciences
Received: May 31, 2024
Accepted: Jul 17, 2024