Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 2078032, 12 pages
https://doi.org/10.1155/2020/2078032

Research Article

Heterogeneous Cross-Domain Identity Authentication Scheme Based on Proxy Resignature in Cloud Environment

Yongyang Lv, Wenju Liu, and Ze Wang

School of Computer Science and Technology, Tiangong University, Tianjin 300387, China

Correspondence should be addressed to Yongyang Lv; zp928634209@qq.com, Wenju Liu; 928634209lwj@sina.com, and Ze Wang; wangze@tjpu.edu.cn

Received 26 June 2020; Revised 10 September 2020; Accepted 4 October 2020; Published 17 November 2020

Academic Editor: Nazrul Islam

Based on proxy resignature, the signature transformation between trust domains of different cryptographic systems is realized with the help of the cloud authentication center, so as to achieve cross-domain access between users and cloud service providers in heterogeneous environment. Hierarchical ID tree realizes the uniqueness of identity in communication, and the security of heterogeneous cross-domain identity authentication is enhanced by the two-factor authentication of "password + key" and temporary identity replacing real identity. The security of the scheme is proved under the CK model, which can anonymously trace entity identity, resist replay attacks, replacement attacks, and man-in-the-middle attacks, and the simulation experiment is carried out. By searching it in related fields, no paper on heterogeneous cross-domain identity authentication based on proxy resignature has been published yet. The results of this paper show that the proposed scheme has better computing performance and higher security.

1. Introduction

A trend of integration has begun among various cloud services in the current cloud environment [1], and more cloud services need to be connected with other cloud services of different domains. Different security domains may adopt different security management mechanisms and password systems [2], and each is only responsible for identity authentication and management within its domain. When users access other domains of different cryptographic systems, there is the problem of heterogeneous cross-domain authentication. Current identity authentication schemes based on mainstream cryptography systems are (a) PKI (public key infrastructure) system based on digital certificates [3], (b) identity-based cryptography (IBC) [4, 5], and (c) certificateless cryptography (CLC) [6]. Among them, PKI system is the best system to guarantee network security, which can provide identity authentication in the open cloud environment. The CLC can effectively solve the key escrow problem in the IBC system.

Proxy resignature was proposed by Blaze et al. [7] at the European Cryptography Conference in 1998, and the specific definition of formal security was given by Ateniese and Hohenberger [8]. In proxy resigning, the semitrusted proxy uses the resigning key to convert the trustee's signature into the entrusting side's signature for the same message, but it cannot generate the message's legal signature on behalf of either side of them. The meaning of "semi-trusted" is believing that the proxy will make the signature transformation according to the scheme. The proxy resignature is used to guarantee the confidentiality, bidirectional authentication, unforgeability, and anonymity of identity information. Malicious attackers cannot obtain the identity information of the sender or receiver from the ciphertext, which plays an effective role in protecting the privacy of user identity on both sides and allows the intercloud identity authentication center to verify the user's identity information and return the authentication results, reducing the computation load carried by users. Yang et al. [9] proposed a threshold proxy resignature scheme to prevent agents from abusing the power of signature conversion. Tian [10] proposed a lattice-based identity proxy resignature scheme in the random prediction model, but the signature length was large and the practicability was poor. Tian et al. [10]

constructed a lattice-based proxy resignature scheme to resist the attack of quantum computing. Yang et al. [11] proposed a separable online/offline proxy resignature scheme, which effectively improved the real-time performance of the proxy resignature. Wang and Lv [12] constructed two server-assisted proxy resignature schemes, both of which are provable and secure in the random prediction model, but the second scheme cannot resist the collusion attack from the server and the malicious agent. In order to reduce the computational cost of the verifier, the papers [12, 13] constructed a secure server-assisted verification agent resignature scheme under the random prediction model and the standard model, respectively. However, the existing proxy resignature schemes [9–13] almost all realize the existence unforgery, which can only ensure that the attacker cannot forge the signature of new messages. In order to meet the security requirements of cross-domain authentication in cloud computing environment, Yang et al. [9], based on CDH and CRF assumptions, proposed a strongly nonfalsifiable server-assisted authentication proxy resignature algorithm under the standard model and delegated most of the computing tasks of signature verification to the server.

Literature [14] uses certificates and PKI to realize cross-domain certification scheme, but both schemes involve complex certificate management and need to afford relatively high computational cost. Literature [15] proposes a grid-based PKI multidomain authentication model, but the model cannot resist forgery attack. Literature [16, 17] takes IBC domain authentication server as an entity in PKI domain and adopt the method of exchanging certificates for authentication, which is inefficient and the trusted domains are not of the same level. Literature [18] proposed an identity authentication scheme based on PTPM and certificatelessness, which realizes the credibility of authentication results between users and cloud service providers but does not consider cross-domain authentication and other issues. Literature [19] proposes a key exchange protocol for cross-domain authentication in the wireless grid, but the use of symmetric encryption causes high computing cost. Literature [20] proposes a cross-domain authentication scheme based on blockchain, inheriting such security defects as blockchain algorithm vulnerability. Literature [21] proposes cross-domain authentication based on different cryptographic systems, but a heavy load is carried by intercloud authentication centers, which is likely to lead to single point authentication failure. Literature [22] proposes key negotiation between different cryptographic systems to achieve cross-domain authentication between trusted domains of different levels, but users carry a large amount of computation and communication. At present, signature encryption algorithm has been widely used in cross-domain authentication schemes, but most of the authentication algorithms are based on the same cryptographic system or use the same system parameters in different cryptographic systems. This security mechanism does not apply well to the actual Internet of Things authentication scenario. Wang et al. [23] proposed a signature scheme based on PKI and IBC, which not only satisfies anonymity but also supports bidirectional verification. However, it has problems such as large traffic and large computation. The cross-domain authentication mechanism proposed by Ferrag et al. [24] can meet the requirements of internal security but does not verify the security of temporary keys. Wang et al. [25] propose a scheme to ensure the security of temporary keys, but it does not support the use of different system parameters in each domain environment. In addition, in the existing cross-domain authentication technologies [26, 27], certificate authentication requires detection from the book to the root certificate. The authentication path is too long and the efficiency of path authentication is low, which greatly affects the practical application scenarios of cross-domain authentication technology.

Most of the existing cross-domain authentication models cannot implement the authentication of different cryptographic systems well. In the authentication schemes which can realize different cryptographic systems, there are either big security problems or high computational cost. This paper proposes a heterogeneous cross-domain identity authentication scheme under the cloud environment. Based on the highly antiforgery proxy resignature algorithm by server-aided verification in literature [28], the scheme realizes the identity authentication and secure access between users of CLC and PKI and cloud service providers. Cloud authentication (CA) center is introduced to issue certificates for the security domains of different cryptographic systems and provides signature transformation for cross-domain users, so that users can access the security domains of different cryptographic systems. The scheme uses the hierarchical ID tree to realize the uniqueness of identity in communication and enhances the security of heterogeneous cross-domain identity authentication through two-factor authentication of "password + key." According to the analysis, the security of the scheme is verified under the CK model, which can resist replay attack, replacement attack, and man-in-the-middle attack. Meanwhile, temporary identity is introduced for the anonymous tracing in the authentication, realizing bidirectional authentication between users and cloud service providers. Finally, the simulation experiment is carried out to further strengthen the security proof of the scheme. Compared with the existing literature, it has higher security and computational efficiency. Searching it in related fields, no paper on heterogeneous cross-domain identity authentication based on proxy resignature has been published yet.

Section 2 of this paper introduces the basic knowledge used in the scheme. Section 3 describes in detail the heterogeneous authentication scheme. Section 4 provides proof of the scheme's security and makes a comparison of the existing work and the scheme in this paper. Section 5 gives the conclusion.

2. Preliminary Knowledge

2.1. Bilinear Mapping. Let G_1 and G_2 be cyclic groups of order p, p is a prime number, and g is a generator of G_1. Define bilinear mapping e: $G_1 \times G_1 \longrightarrow G_2$.

2.2. Assumptions of the Security Theory. Computational CDH (Computational Diffie-Hellman) problem: for a given triplet $(g, g^a, g^b) \in G_1^3$, for any $a, b \in Z_p^*$, compute $g^{ab} \in$.

Definition 1 (CDH hypothesis). For any probabilistic polynomial time algorithm B, the probability of successfully solving CDH problem is $\mathrm{Adv}_{\mathrm{CDH}}(B) = \Pr[B(g, g^a, g^b) = g^{ab} : g \in G_1, a, b \in Z_p^*]$. If $\mathrm{Adv}_{\mathrm{CDH}}(B)$ is negligible, CDH problem on G1 is difficult [17].

Definition 2 (CRH hypothesis). Suppose a family of anticollision hash functions $H_K : \{0,1\}^* \longrightarrow \{0,1\}^{n_k}$, where k is an indicator and n_k is the length of the output message. The probability of any probabilistic polynomial time algorithm B successfully finding a pair of collision of H_K is $\mathrm{Adv}_{\mathrm{CRH}}(B) = \Pr[B(k) = (m_0, m_1) : m_0 \neq m_1, H_K(m_0) = H_K(m_1)]$. If $\mathrm{Adv}_{\mathrm{CRH}}(B)$ is negligible, then H_K is anticollision [29].

2.3. CK Security Model. The CK (Canetti-Krawczyk) security model [30, 31] defines two attack models as the AM model for authenticated links and the UM model for unauthenticated links. In the ideal model AM, any attacker cannot forge, tamper, and replay messages and can only pass the same message once and has the ability to query session key, call operation, compromise protocol participants, expose the session key, and test the session key.

Definition 3. Given that A is any attacker in the AM, if the session key of the authentication protocol is safe in the AM, the properties below are satisfied.

Property 1. Both parties can obtain the same session key after they are not compromised and execute the agreement successfully.

Property 2. The attacker A makes the test of attacking the session key query, and according to the result, A can correctly determine whether the output value of the session key is a random value or A real value with the probability not exceeding $(1/2) + \varepsilon$ (ε represents any value that can be ignored within the security parameter range).

3. Heterogeneous Cross-Domain Identity Authentication Scheme Based on Proxy Resignature

3.1. Heterogeneous Cross-Domain Authentication Model Based on Proxy Resignature. The cross-domain authentication model under heterogeneous environment is shown in Figure 1. The model includes five participating entities: (1) cloud service provider (CSP), which provides users with a variety of cloud services and uses secure devices of Trusted Platform Module (TPM) to store, encrypt, and sign sensitive data such as keys and random numbers; (2) user (U), who uses any terminal device that supports Portable TPM (Portable TPM, PTPM) security module to access the cloud service and complete the cross-domain identity authentication process with the cloud service provider. TPM and

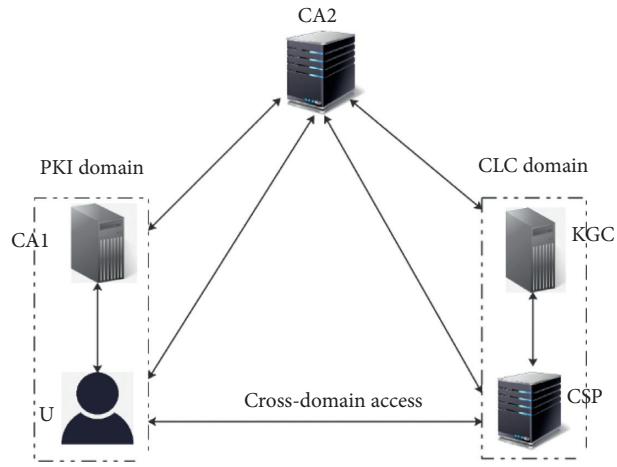


Figure 1: Heterogeneous cross-domain authentication model based on proxy resignature.

PTPM can ensure credible identity authentication and correct authentication results; (3) PKI domain certification center (CA1), which is responsible for the application, issuance, revocation, and inquiry of certificates of users in the domain and signing their temporary identity in the domain; (4) CLC key generation center (KGC), which mainly generates and distributes part of the keys for users in the domain and cloud service providers and is responsible for tracing the true identity of users with malicious anonymous behaviors; (5) intercloud authentication center (CA2) for identity authentication between different trust domains and signature conversion.

3.2. Scheme Description. In this scheme, any two trusted domains are set as PKI domain and CLC domain, respectively. CA1 is the authentication center of PKI domain, KGC is the key generation center of CLC domain, and intercloud authentication center (CA2) generates resignature keys for domains of different cryptosystems and provides trust support and signature conversion. At the same time, it verifies the legitimacy of the subdomain of different cryptographic systems, and if it is legitimate, it issues a certificate for the security domain. The subdomains manage users and cloud service providers in their own domains, respectively, and provide authentication for users in their own security domains to access cloud service providers and authentication of public cross-domain identities from other domains. In this scheme, if a user of a security domain sends access requests to the CSP of another security domain with a different password system, the CSP will, after receiving it, verify the message and send the user's message to the CA2, which uses the resignature keys to transform the signature on the user's certificate given by CA1 into one by KGC or one by KGC into one by CA1, followed by the conversion of the certificate. Then, the converted certificate and related identity information are sent to CA1 or KGC, where the converted signature is verified. If the verification is passed, the identity information of the user is sent to the CSP, which then sends out a response. The user, receiving the response,

verifies the CSP's identity. If the whole process works out, the cloud service provider establishes a trust connection with the user. The process of cross-domain authentication scheme based on proxy resignature under the heterogeneous environment is shown in Figure 2.

Because they share the same proxy resignature key and work independently, each intercloud authentication center (CA2) is equal on the signature transformation, so this paper only discusses the heterogeneous cross-domain authentication scheme based on a single cloud certification center, which can be easily extended to multiple intercloud authentication center with the security ensured.

3.2.1. System Establishment. Let G_1 and G_2 be cyclic groups of order p, p is a prime number, and g is a generator of G_1. Define bilinear mapping e: $G_1 \times G_1 \longrightarrow G_2$. Select two anticollision hash functions $H_1: \{0,1\}^* \longrightarrow \{0,1\}^{n_c}$ and $H_2: \{0,1\}^* \longrightarrow \{0,1\}^{n_m}$, $n_m < p$, $n_c < p$, and the output of the hash function is a member of Z_p^*. Randomly select three elements g_1, g_2, and $u \in G_1$, and randomly select nm elements $(u_1, \ldots, u_{n_m})$; the symbol "||" represents the string connection operator, exposing the system parameter $p_a = \{G_1, G_2\ p, g, e, g_1, g_2, u, (u_i)_{i=1}^{n_m}, H_1, H_2\}$.

PKI authentication center (CA1) randomly selects $\alpha \in Z_p^*$ as the master key of the system and calculates the public key $\mathrm{PK}_{\mathrm{CA1}} = g^{\alpha}$. Key generation center KGC randomly selects $\beta \in Z_p^*$ as the master key of the system and calculates the public key $\mathrm{PK}_{\mathrm{KGC}} = g^{\beta}$. The intercloud authentication center (CA2) randomly selects $\theta \in Z_p^*$ as the master key of the system and computes the public key $\mathrm{PK}_{\mathrm{CA2}} = g^{\theta}$. Finally, the public keys $\mathrm{PK}_{\mathrm{CA1}}$, $\mathrm{PK}_{\mathrm{KGC}}$, and $\mathrm{PK}_{\mathrm{CA2}}$ are published. The public-private key pair for U is $\{\mathrm{PK}_U, \mathrm{sk}_U\}$, and the public-private key pair for CSP is $\{\mathrm{PK}_{\mathrm{CSP}}, \mathrm{sk}_{\mathrm{CSP}}\}$.

Intercloud authentication center (CA2) generates resignature keys for domains of different cryptography systems and verifies the legitimacy of the subdomain of the security domain of different cryptography systems, and if it passes the verification, a certificate is issued to it. According to the proxy resignature generation algorithm proposed in literature [16], the resignature key generation process in this paper is as follows: CA2 randomly selects $r_p \in Z_p^*$, calculates $R_p = g^{r_p}$, and sends to CA1; CA1 calculates $R_{p1} = R_p g_2^{\alpha}$ through its own private key and sends it to KGC; KGC calculates $R_{p2} = g_2^{\beta}/R_{p1}$ through its own private key and returns the result to CA2; CA2 calculates the resignature key $rk = R_p R_{p2} = g_2^{\beta-\alpha}$. Because this paper has more symbols, Table 1 explains the meaning of these symbols.

3.2.2. Identity Generation. In this scheme, the hierarchical ID tree in literature [32] is adopted to define the ID value in order to realize the uniqueness of identity. As shown in Figure 3, in the 2-tier ID tree, the root node is the identity mark of the CLC key generation center or the authentication center CA of the PKI domain, and the leaf node is the identity mark of the users and cloud service providers in the trusted domain. If the identity of CA1 in the PKI domain is DN_{α} and that of user U is DN_U, then the real identity of U is defined as $\mathrm{ID}_U = \mathrm{DN}_{\alpha} || \mathrm{DN}_U$. Similarly, the identity of KGC of the CLC domain is DN_{β}, and the identity of the CSP is $\mathrm{DN}_{\mathrm{CSP}}$; then, the real identity of the CSP is defined as $\mathrm{ID}_{\mathrm{CSP}} = \mathrm{DN}_{\beta} || \mathrm{DN}_{\mathrm{CSP}}$.

3.2.3. Key Generation

(1) User registration of the PKI domain

(1) User U selects a random secret value of r_U. Calculate temporary identity $\mathrm{TID}_U = H_1(\mathrm{ID}_U \| g^{r_U})$ according to U's real identity ID_U. Encrypt the registration request $\mathrm{En}\{\mathrm{ID}_U, \mathrm{ID}_{\mathrm{CA1}}, \mathrm{TID}_U, g^{r_U}, \mathrm{PK}_U\}_{\mathrm{PK}_{\mathrm{CA1}}}$ with CA1's public key PK_{β}, and send it to CA1.

(2) CA1 uses its master key to decrypt the received registration message and verify whether U is a legitimate user of local security domain, by verifying user temporary identity $\mathrm{TID}_U = H_1(\mathrm{ID}_U \| g^{r_U})$. If it fails, give a response of failure or else check in the registered user list whether ID_U already exists. If so, the certificate is issued directly to U, and if not, CA1 randomly selects $S_U \in Z_p^*$, generates the certificate message $m_{\mathrm{CA1} \longrightarrow U}$ composed of CA1, TID_U, and PK_U, computes $M_{\mathrm{CA1} \longrightarrow U} = H_2(m_{\mathrm{CA1} \longrightarrow U}) = (M_{\mathrm{CA1} \longrightarrow U,1}, \ldots M_{\mathrm{CA1} \longrightarrow U,n_m}) \in \{0,1\}^{n_m}$, $E_1 = u \prod_{i=1}^{n_m} (u_i)^{M_{\mathrm{CA1} \longrightarrow U,i}}$, and $h = H_1(m_{\mathrm{CA1} \longrightarrow U} \| g^{S_U})$ and uses CA1's private key α to generate the signature $\delta_{\mathrm{CA1} \longrightarrow U} = (\delta_{\mathrm{CA1} \longrightarrow U,1}, \delta_{\mathrm{CA1} \longrightarrow U,2}) = ((g_2)^{\alpha} (Eg_3^h)^{S_U}, g^{S_U})$ of the certificate message $m_{CA1 \longrightarrow U}$. CA1 issues to U a certificate $\mathrm{Cert}_U = \{\mathrm{TID}_U, \mathrm{PK}_U, \mathrm{ID}_{\mathrm{CA1}}, T_{\mathrm{begin}}, T_{\mathrm{end}}, m_{CA1 \longrightarrow U}, \delta_{CA1 \longrightarrow U}\}$, where $T_{\mathrm{begin}}, T_{\mathrm{end}}$ is the valid start and end time of the certificate. CA1 saves $\{\mathrm{ID}_U, \mathrm{TID}_U, g^{r_U}, \mathrm{PK}_U\}$ in the list of registered users, stores the certificate to the certificate library, reads the local timestamp T_U, and sends the response $\mathrm{En}\{\mathrm{ID}_U, \mathrm{ID}_{\mathrm{CA1}}, T_U, \mathrm{Cert}_U\}_{\mathrm{PK}_U}$ to U.

(3) User U decrypts the response through its private key and checks freshness of the timestamp T_U, verifies the validity of certificate Cert_U with public key PK_{CA1} of the root certificate CA1, and stores $\{\mathrm{PK}_U, \mathrm{sk}_U, \mathrm{Cert}_U\}$ in PTMP if it is valid; otherwise, the registration fails and the certificate is refused.

(2) User registration of the CLC domain

(1) Cloud service provider CSP selects the random secret value r_{CSP}, $x_{\mathrm{CSP}} \in Z_p^*$ and computes the public key $\mathrm{PK}_{\mathrm{CSP}} = g^{x_{\mathrm{CSP}}}$. According to the real identity of the cloud service provider (CSP), calculate the temporary identity $\mathrm{TID}_{\mathrm{CSP}} = H_1(\mathrm{ID}_{\mathrm{CSP}} \| g^{r_{\mathrm{CSP}}})$. The message applying for registration is encrypted through the public key of the KGC and $\mathrm{En}\{\mathrm{ID}_{\mathrm{CSP}}, \mathrm{TID}_{\mathrm{CSP}}, g^{r_{\mathrm{CSP}}}, \mathrm{PK}_{\mathrm{CSP}}\}_{\mathrm{PK}_{\mathrm{KGC}}}$ is sent to KGC.

(2) After decrypting the encrypted message with its master key, KGC obtains the real identity $\mathrm{ID}_{\mathrm{CSP}}$

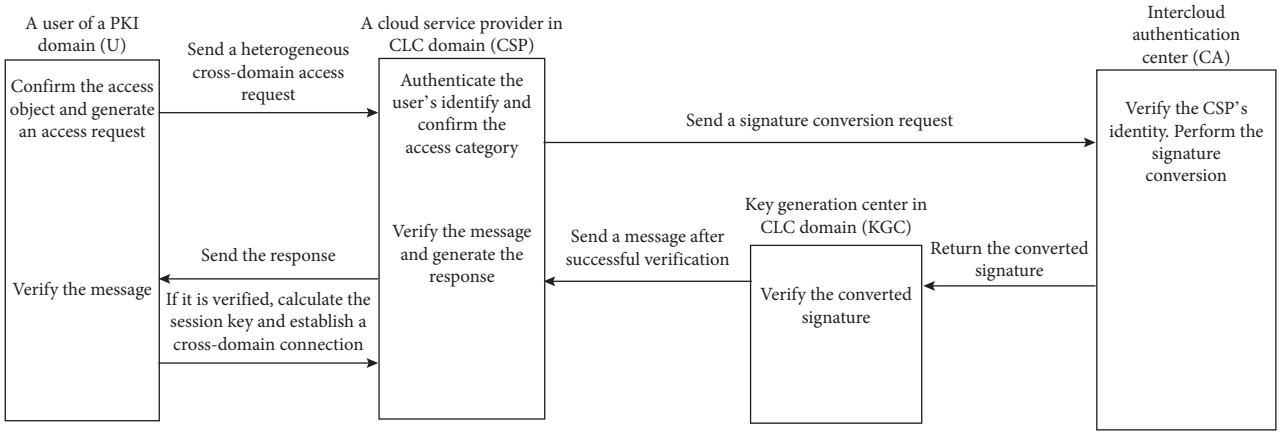


Figure 2: Flowchart of heterogeneous cross-domain authentication scheme based on proxy resignature.

Table 1: Explanation of symbols.

Symbol	Meaning of the symbol
CLC	Certificateless cryptography
PKI	Public key infrastructure
	User
	Cloud service provider
KGC	Key generation center in the CLC domain
	Certificate authority in the PKI domain
	Cloud certification center
ID_U	The true identity of user U
ID_{CSP}	The true identity of cloud service provider (CSP)
TID_U	Temporary identity of user U
TID_{CSP}	Temporary identity of cloud service provider (CSP)
PK_α	The public key of the certificate authority (CA1)
PK_β	The public key of the key generation center (KGC)
PK_{CA}	The public key of the cloud certification center CA2
Rk	The resignature key of the CA2
DN_α	The identity of the certificate authority (CA1)
DN_β	The identity of the key generation center (KGC)
DN_U	The identity of user U
DN_{CSP}	The identity of cloud service provider (CSP)
α	The master key of the certificate authority (CA1)
β	The master key of the key generation center (KGC)
θ	The master key of the cloud certification center (CA2)
pk_U	The public key of user U
sk_U	The private key of user U
pk_{CSP}	The public key of the cloud service provider (CSP)
psk_{CSP}	The partial private key of the cloud service provider (CSP)
sk_{CSP}	The private key of the cloud service provider (CSP)
$\delta_{CA1 \longrightarrow U}$	The CA1's signature of the user's certificate
$Cert_U$	A certificate issued by a CA1 to user U

according to DN_{CSP} and verifies whether the temporary identity $TID_{CSP} = H_1(ID_{CSP} \| g^{r_{CSP}})$ is correct. If not, give the response of failure, or else compute $Q_{CSP} = H_1(TID_{CSP})$ and the partial private key $psk_{CSP} = (Q_{CSP})^{\beta}$. Read the local timestamp T_{CSP}, return the message $En\{psk_{CSP}, T_{CSP}, Q_{CSP}\}_{PK_{CSP}}$ to the CSP, and save $\{ID_{CSP}, TID_{CSP}, g^{r_{CSP}}, PK_{CSP}, T_{CSP}\}$ in the user registration list.

(3) After receiving the message, the CSP uses its own private key to decrypt the message and verify the freshness of the timestamp T_{CSP}, calculates the complete private key $sk_{CSP} = (x_{CSP}, psk_{CSP})$, and keeps it in PTMP secretly. Finally, the public key pk_{CSP} is shared.

3.2.4. Cross-Domain Authentication

(1) PKI domain ⟶CLC domain cross-domain authentication

(1) User U randomly selects $y \in Z_p^*$ and uses the private key $sk_U = (x_U, psk_U)$ to calculate the key negotiation parameter $Y' = g^y$, randomly select the password value pw, and calculate $w = H_1(TID_U \| pw)$. Let $m_1 = (request_1, ID_{CSP,} TID_U, w, T_U, N_U, Y')$, where $request_1$ is the identity of access request, T_U is the timestamp, and N_U is the random parameter to keep the freshness of the message. Using the signature algorithm in literature [13], user U randomly selects $r_m \in Z_p^*$, calculates $M_1 = H_2(m_1) = \{M_{1,i}\}_{i=1}^{n_m} \in \{0,1\}^{n_m}$, $h_1 = H_1(m_1 \| g^{r_m})$, and $E_1 = u \prod_{i=} 1^{n_m} (u_i)^{M_{1,i}}$, and uses the user's private key sk_u to generate the signature of a message m_1, $(\delta'_U = \delta'_{U,1}, \delta'_{U,2}) = ((g_2)^{sk_u}(E_1 g_3^{h_1})^{r_m}, g^{r_m})$. Read certificate $Cert_U$ and send the authentication request $En\{request_1, ID_{CSP}, TID_U, w, T_U, N_U, Y', \delta'_U, Cert_U\}_{PK_{CSP}}$ to the cloud service provider CSP.

(2) After receiving the message, the CSP uses its own private key to decrypt the message and perform the following operations:
First check whether $request_1$ is an access request; if not, then refuse to accept it; if so, then proceed to the next step.
Extract the public key PK_U of U in the certificate $Cert_U$ and calculate $m_1 = (request_1, ID_{CSP,} TID_U, w, T_U, N_U, Y')$ to verify whether $e = (\delta'_{U,1}, g) = e(g_2, PK_U) e(E_1 g_3^{h_1}, \delta'_{U,2})$ is true. If not, the authentication is terminated. If it is true, the authentication of U's identity is completed.

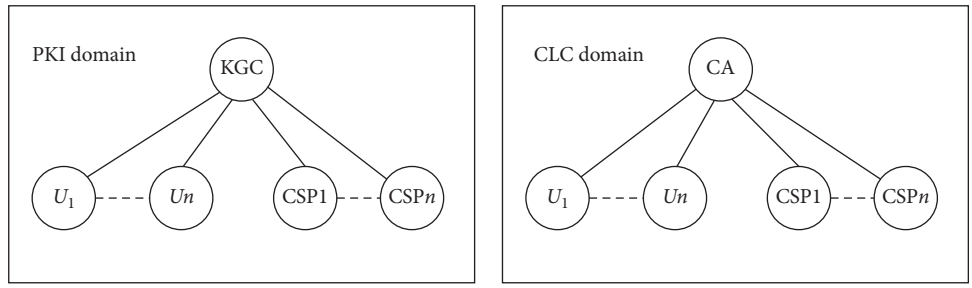


Figure 3: Hierarchical ID tree.

Then check whether user U's identity exists. If so, go straight to repeat the cross-domain authentication. If it does not exist, check the freshness of timestamp T_U and verify whether the start and end time of the certificate are valid. If any of the above steps fails, terminate the access; otherwise, proceed to the next step.

Randomly select $N^1_{CSP} \in Z_p^*$, read timestamp T^1_{CSP}, and send the message applying for the certificate conversion $\mathrm{En}\{\mathrm{ID_{CSP}}, \mathrm{ID_{CA2}}, T^1_{\mathrm{CSP}}, N^1_{\mathrm{CSP}}, \mathrm{Cert}_U\}_{\mathrm{PK_{CA2}}}$ to CA2.

(3) CA2 decrypts the message with its own master key and performs the following operations:

Check the freshness of timestamp T^1_{CSP} and use the public key of CA1 to verify the validity of Cert_U. If the verification fails, return the message of failure, or else, perform the next step.

Use the resignature key $rk = g_2^{\beta-\alpha}$ to convert Cert_U's signature by CA1 $\delta_{\mathrm{CA1}\longrightarrow U} = (\delta_{\mathrm{CA1}\longrightarrow U}, 1, \delta_{\mathrm{CA1}\longrightarrow U,2}) = ((g_2)^{\alpha}(Eg_3^h)^{S_U}, g^{S_U})$ into a signature by KGC $\delta_{\mathrm{KGC}\longrightarrow U} = (\delta_{\mathrm{KGC}\longrightarrow U,1}, \delta_{\mathrm{KGC}\longrightarrow U,2}) = (rk(\delta_{CA1\longrightarrow U,1}, \delta_{CA1\longrightarrow U,2})) = ((g_2)^{\beta}(Eg_3^h)^{S_U}, g^{S_U})$. Then, Cert_U issued by CA1 is converted into Cert^1_U issued by KGC. The validity period of Cert^1_U can be set to be very short, and CA2 cannot calculate the private keys of CA1 and KGC through the resignature key; that is, it cannot issue certificates in place of CA1 and KGC.

CA2 calculates $H_1(\mathrm{ID_{CA2}})$, reads the timestamp T_{CA}, and sends the message $\mathrm{En}\{\mathrm{ID_{CSP}}, \mathrm{ID_{CA2}}, T_{\mathrm{CA}}, N^1_{\mathrm{CSP}}, H_1(\mathrm{ID_{CA2}}), \mathrm{Cert_U}, \mathrm{Cert}^1_U\}_{\mathrm{PK_{KGC}}}$ to KGC.

(4) KGC take the following steps concerning the message:

KGC checks the freshness of the timestamp T_{CA} after receiving the message. If the verification fails, the authentication is terminated. If it passes the verification, the signature of Cert^1_U, $\delta_{\mathrm{KGC}\longrightarrow U} = (\delta_{\mathrm{KGC}\longrightarrow U,1}, \delta_{\mathrm{KGC}\longrightarrow U,2})$ is verified by KGC's public key, which means calculating $e(\delta_{\mathrm{KGC}\longrightarrow U,1}, g) = -e(g_2, \mathrm{PK_{KGC}})e(Eg_3^h, \delta_{\mathrm{KGC}\longrightarrow U,2})$ and seeing whether it is valid or not. If it is not, the authentication is terminated. If it is valid, Cert^1_U is a legal certificate, and KGC accepts the certificate Cert_U and sends the message $\mathrm{En}\{\mathrm{ID_{CSP}}, \mathrm{ID_{CA2}}, N^1_{\mathrm{CSP}}, H_1(\mathrm{ID_{CA2}}), \mathrm{Cert}_U, \mathrm{Cert}^1_U\}_{\mathrm{PK_{CSP}}}$ to the CSP.

(5) After receiving the message, the CSP will check whether N^1_{CSP} in the message is the same as the random parameter in the message applying for transformation. If not, the authentication will be terminated. Otherwise, save $\{\mathrm{TID}_U, w, N1, D1, \mathrm{Cert}_U\}$ in the authentication list, $N1$ and $D1$as the number of times and valid time of U repeating cross-domain authentication. Finally, the CSP randomly selects $N^2_{\mathrm{CSP}}, z \in Z_p^*$ and uses the private key to calculate the key negotiation parameter $Z' = g^z$, reads the timestamp T^2_{CSP}, calculates the signature of $\mathrm{ID_{CSP}}$ $\delta'_{\mathrm{CSP}} = (H_1(\mathrm{ID_{CA2}}))^{\beta}$, sends the response $\mathrm{En}\{\mathrm{request}_1, \mathrm{ID_{CSP}}, \mathrm{ID_{CA2}}, \mathrm{TID}_U, Z', Y', T\ \mathrm{CSP}^2, N^2_{\mathrm{CSP}}, N_U, \mathrm{PK_{CSP}}, H_1(\mathrm{ID_{CA2}})\delta'_{\mathrm{CSP}}\}_{\mathrm{PK}_U}$ to user U, and calculates the session key with U, $K = (\mathrm{PK}_U)^{\mathrm{sk_{CSP}}}(Y')^z$.

(6) User U checks whether N_U in the response is consistent with the authentication request message sent, checks the freshness of timestamp T^2_{CSP}, verifies whether $e(\delta'_{\mathrm{CSP}}, g) = e(H_1(\mathrm{TID_{CSP}}), \mathrm{PK}_{\beta})$ is true, calculates whether $H_1(\mathrm{ID_{CA2}})$ is the same as that in the response message, and terminates the authentication if any step fails. If all hold, save $\{\mathrm{ID}_{CA2}, \mathrm{ID_{CSP}}, \mathrm{PK_{CSP}}, H_1(\mathrm{ID}_{CA2})\}$ to the authentication list and calculate the session key $K = (\mathrm{PK_{CSP}})^{\mathrm{sk}_U}(Z')^y$. The PKI domain will establish a trusted heterogeneous cross-domain connection with the CLC domain.

(2) CLC domain $\longrightarrow$ PKI domain cross-domain authentication

When a user in the CLC domain sends an access request to a CSP in the PKI domain, the KGC in the CLC domain signs the certificate issued by CA2 and sends it to user U, who then sends it to the CSP as part of the access request. The rest steps are the same as that in the "PKI domain $\longrightarrow$ CLC domain cross-domain authentication," so it will not be repeated.

3.2.5. Repeated Cross-Domain Authentication. User U and cloud service provider CSP pass the first cross-domain authentication, and the cloud service provider records the user's identity information in the user registration list. Repeated cross-domain authentication is mainly used to determine whether the number of times of domain crossing and timestamps are within the valid range through the session keys provided by users and cloud service providers, so as to determine whether the repeated cross-domain authentication is successful. Repeated cross-domain authentication no longer requires interaction with the intercloud authentication center, and users and cloud service providers are not required to carry heavy loads of computation. This means the completion of security authentication of the bidirectional cross-domain identity. The repeated cross-domain authentication model is shown in Figure 4.

(1) User U reads the timestamp T_i, selects random parameters N_i, $y_i \in Z_p^*$, calculates key negotiation parameter $Y_i = g^{y_i}$, enters temporary id TID_U and password pw, calculates $w' = H_1(\mathrm{TID}_U \| pw)$, and sends the message $\mathrm{En}\{\mathrm{request}_i, \mathrm{ID}_{\mathrm{CSP}}, \mathrm{TID}_U, w, T_i, N_i, Y_i, \mathrm{Cert}_U\}_{\mathrm{PK}_{\mathrm{CSP}}}$ to CSP.

(2) After receiving the message, the CSP uses its own private key to decrypt the message and then performs the following operations:

Determine whether $\mathrm{request}_i$ is an access request, check the freshness of the timestamp T_i, query user information in the access user list according to TID_U, and verify whether it is the same as w in the user list. If they are different, terminate authentication and return the information of error to user U.

Verify whether $D1$ exceeds the time validity, and verify whether $N1$ exceeds the maximum number of visits. If either of them exceeds its range, then terminate the authentication.

If any part of the above verification fails, stop execution or update the access list to $N1 = N1 + 1$. The CSP reads timestamp T_o, selects the random parameters N_O, $z_i \in Z_p^*$, calculates the session key parameter $Z_i = g^{z_i}$, and calculates the session key $K_i = (\mathrm{PK}_U)^{\mathrm{sk}_{\mathrm{CSP}}} (Y_i)^{z_i}$. Finally, it sends the message $\mathrm{En}\{\mathrm{request}_i, T_o, N_i, N_o, Z_i, Y_i\}_{\mathrm{PK}_U}$ to U.

(3) User U checks whether the N_i in the response is consistent with the one in the authentication request sent, checks the freshness of the timestamp T_o, and terminates authentication if the verification fails. If the above verification is passed, the session key $K_i = (\mathrm{PK}_{\mathrm{csp}})^{\mathrm{sk}_U} (Z_i)^{y_i}$ is calculated and a trusted heterogeneous cross-domain connection is established between the PKI domain and the CLC domain.

4. Scheme Analysis

4.1. Security Analysis. The security of key generation and cross-domain authentication algorithm proposed in this scheme is based on the security of proxy resignature scheme proposed in literature [28], which has been proved. This scheme is based on the CK model proposed in literature [33, 34] to prove the security of cross-domain identity authentication scheme.

This scheme describes cross-domain identity authentication as a protocol ψ in the AM. The security of the protocol ψ is analyzed under the CK security model. Since the algorithm has been proved to be unforgeable, it is only necessary to prove that the protocol ψ satisfies the two properties of Definition 3, in order to prove that the session key of the protocol ψ is secure in the AM.

(1) Because neither of the message participants of the protocol is compromised by the attacker A in the AM, both user U and the cloud service provider CSP can obtain the key negotiation parameters Y_U and Z_{CSP} that are not tampered with and calculate and obtain the same session key K, which satisfies the first property of Definition 3 concerning session key security.

(2) Assuming that the attacker A initiates q rounds of guessing in the AM, there is an Algorithm B which uses the nonnegligible probability ε based on the guessing results of A to correctly distinguish whether the session key of the protocol ψ is a real value or a random value. Randomly select the number of rounds for testing sessions, $n \in \{1, 2, 3, \ldots, q\}$. In n rounds of session, the input value of B is Y_U, Z_{CSP}, and K, among which Y_U is the key negotiation parameter of user U, Z_{CSP} is the key negotiation parameter of the CSP, and K is the response of query. The following 2 situations are discussed:

(1) A is selected in the n-th round of sessions. If A can guess whether the response value is real or random with the probability of $(1/2) + \varepsilon$, B can also guess whether the input value is real or random with the probability of $(1/2) + \varepsilon$, because if the input of B is a real session, the response A is the real value of the session key Y, and if the input is a random value, the response K is a random value.

(2) A is not selected in the n-th round of sessions. Choosing another round except the n-th round, B can guess whether the input is a real value or random value with the probability of 1/2. The probability that the tested session is the n-th session is $1/q$, and the probability of A guessing correctly the test response is $(1/2) + \varepsilon$. The

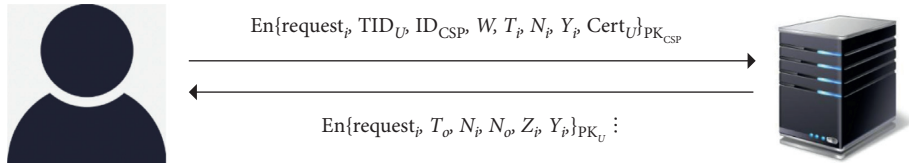


Figure 4: Repeated cross-domain authentication model.

probability that the test session is not the n-th session is $1 - (1/q)$, and the probability of A guessing right the test response is 1/2, so the probability of making a successful guess is $(1/q)((1/2) + \varepsilon) + (1 - (1/q))(1/2)$. From this, the probability that B guesses the right session key is $(1/2) + (\varepsilon/q)$; that is, the second property of Definition 2 is satisfied, so the session key of the protocol ψ is secure.

4.2. Antireplay Attacks. In this scheme, during cross-domain authentication, user U and the cloud service provider CSP randomly select the local timestamp and random parameters which keep the session fresh to ensure the validity of the message. If a malicious attacker intercepts the message and replays it in the cross-domain authentication, the verification conducted by the receiving party will fail, because the freshness of the timestamp of the replayed message is different from that of the original one. As a result, this scheme can effectively resist replay attacks.

4.3. Antireplacement Attacks. In this scheme, the real identities of user U and the cloud service provider CSP are replaced by randomly selected secret values as their temporary identities TID_U and TID_{CSP}, and, at the stage of key generation, the KGC signs the temporary identity of user U in CLC domain and the CA signs the user's certificate in PKI domain, so as to protect user identity. In the cross-domain authentication, $w = H_2(\mathrm{TID}_U || pw)$ binds the password and user's temporary identity and further strengthens the security by the two-factor authentication of "password + key." If the attacker replaces the user's identity in the message interaction of cross-domain authentication, the authentication will fail at the time when the other party receives the message for authentication. Therefore, this scheme can effectively resist the replacement attack.

4.4. Anonymous Tracking of Entity Identity. In order to ensure the identity security of user U and the CSP, the temporary identities TID_U and TID_U are established to replace the real identities ID_U and ID_{CSP}, so as to realize the anonymity of identity. If an illegal user sends an illegal request to the cloud service provider, the CSP submits TID_U and certificate Cert_U to the authentication CA2 for verification. After CA2 verifies the validity of Cert_U, it searches the user registration list $\{\mathrm{ID}_U, \mathrm{TID}_U, g^{r_U}, \mathrm{PK}_U\}$ according to the temporary identity TID_U and verifies whether the temporary identity is TID_U by $\mathrm{TID}_U = H_1(\mathrm{ID}_U \| g^{r_U})$. If the verification is passed, it means that the user who sends the illegal message is ID_U. CA2 will send the result to the CSP. If the user is a user in the CLC domain, TID_U will be sent to the KGC for authentication, and the remaining steps are the same as the above ones. In this way, the scheme can anonymously trace the entity's identity.

4.5. Anti-Man-in-the-Middle Attacks. When user U crosses the domain to access the cloud service provider, user U includes in the message the signature by CA1 on its temporary identity and encrypts the message through the public key of the CSP in the communication. The CSP can decrypt the message only by means of its own private key and then verifies the message, thus ensuring that the identity is real. Man-in-the-middle attacks are resisted.

4.6. Simulation Experiment. AVISPA, an automatic formal security verification tool, is used to analyze the security properties of the scheme. AVISPA is a formalized security verification tool widely recognized and used in the industry that analyzes the potential security risks of security protocols at a very fine level of granularity and defines security services in protocols, such as key confidentiality, authentication, and capability against man-in-the-middle attack and replay attack, with great precision. In addition, AVISPA integrates OFMC, Cl-ATSE, SATMC, TA4SP, and other four background model analysis tools. In this scheme, OFMC and Cl-ATSE are selected for mutual verification to ensure the reliability of analysis results. The source code is shown in Figure 5.

This scheme uses the HLPSL language built in AVISPA tool to describe the process of the identity authentication scheme in this paper. In the process of identity authentication, public key, multiplication, addition, and logarithm operations are essentially one-way functions, and their inverse operations are difficult to obtain, so we replace these operations with one-way hash functions with the same security properties. In this model, the attacker has complete control over the entire network and can forward, modify, replay, block, and forge any information at any location in the network. Meanwhile, the attacker can also pretend to be a protocol participant and have the same knowledge as the protocol participant but cannot crack the encryption function defined in AVISPA.

The experimental model was independently verified by Cl-ATSE and OFMC analysis engines for many times, proving that the scheme in this paper is safe against replay attack, substitution attack, and man-in-the-middle attack. The verification results are shown in Figures 6(a) and 6(b), respectively, and the results are all safe.

The above is the security analysis of this scheme. Compared with cross-domain schemes in recent years, it can be seen from

```
role role_user(U,CSP,CA1,KGC:agent,Sk:symmetric_key,H1,H2,Pred,Ebilinear:hash_func,SND,RCV:channel(dy))
played_by U
def=
        local
                State:nat,
                Yy,Y,W,Tu,Nu,M1,Sigma,G,K1,TIDu,Sku,PW,Request,IDcsp,G2,W1,G3,Rm,Certu,
                Request1,Z,Tcsp2,Ncsp2,H1IDca1,Ebi,IDcs,PKu,Skcs:text

        const
                sp1,sp2,u_csp_sig:protocol_id
        init
                State:=0
        transition
                0.
                State=0/\RCV(start)=|>
                State':=2/\Yy':=new()
                /\Y':=Pred(G,Yy')
                /\W':=H1(TIDu,PW)
                /\Tu':=new()
                /\Nu':=new()
                /\Rm':=new()
                /\M1':=H2(Request,TIDu,IDcsp,W',Tu',Nu',Y')
                /\Sigma':=Pred(Pred(Pred(G2,Sku),Pred(W1,G3)),Pred(G,Rm'))
                /\SND(Request.TIDu.IDcs.W'.Tu'.Nu'.Y'.Sigma'.Certu)
                /\secret({Sk},sp1,{U})
                5.
                State=5/\RCV(Request1,TIDu,IDcsp,Z',Y',Tcsp2,Ncsp2,Nu,H1IDca1,Sigma')=|>
                State':=6/\Sigma':=Pred(Pred(Pred(G2,Sku),Pred(W1,G3)),Pred(G,Rm))
                /\K1':=Pred(Pred(PKu,Skcs),Pred(Y,Z))
                /\secret({K1},sp2,{U})
end role
```

```
role role_csp(U,CSP,CA1,KGC:agent,H1,H2,Pred,Ebilinear:hash_func,SND,RCV:channel(dy))
played_by CSP
def=
        local
                State:nat,
                Yy,Y,W,Tu,Nu,M1,Sigma,G,K2,TIDu,Sku,PW,Request,IDcsp,G2,W1,G3,Rm,Certu,
        Request1,Z,Tcsp2,Ncsp2,H1IDca1,Ebi,PKu,Ncsp,IDca1,IDcs,Zz,Skcs,SKcsp:text

        const
                sp3,sp4,u_csp_sig:protocol_id
        init
                State:=1
        transition
                1.
                State=1/\RCV(Request.TIDu.IDcs.W'.Tu'.Nu'.Y'.Sigma'.Certu)=|>
                State':=4/\M1':=H2(Request,TIDu,IDcsp,W',Tu',Nu',Y')
                /\Ebi':=Ebilinear(Pred(G2,PKu),Pred(G3,W1))
                /\Ncsp':=new()
                /\SND(IDcsp,IDca1,Ncsp',Certu)
                4.
                State=4/\RCV(IDca1,IDcsp,Ncsp',H1IDca1,Certu)=|>
                State':=7/\Ncsp2':=new()/\Zz':=new()
                /\Z':=Pred(G,Zz')
                /\Sigma':=Pred(Pred(Pred(G2,Sku),Pred(W1,G3)),Pred(G,Rm))
                /\SND(Request1,TIDu,IDcsp,Z',Y,Tcsp2,Ncsp2,Nu,H1IDca1,Sigma')
                /\K2':=Pred(Pred(PKu,Skcs),Pred(Y,Z))
                /\secret({SKcsp},sp3,{CSP})
                /\secret({K2},sp4,{CSP})
 end role
```

```
role role_ca1(U,CSP,CA1,KGC:agent,H1,Pred,Ebilinear:hash_func,SND,RCV:channel(dy))
played_by CA1
def=
        local
                State:nat,
                Yy,Y,W,Tu,Nu,M1,Sigma,Sku,G,TIDu,PW,Request,IDcsp,G2,W1,G3,Rm,Certu,
        Request1,Z,Tcsp2,Ncsp2,H1IDca2,Ebi,PKu,IDca2,Ncsp:text

        const
                u_csp_sig:protocol_id
        init
                State:=2
        transition
                2.
                State=2/\RCV(IDcsp,IDca2,Ncsp',Certu)=|>
                State':=8/\H1IDca2':=H1(IDca2)
                /\SND(IDca2,IDcsp,Ncsp',H1IDca2,Certu)
end role
```

```
role role_kgc(U,CSP,CA1,KGC:agent,H1,Pred,Ebilinear:hash_func,SND,RCV:channel(dy))
played_by KGC
def=
        local
                State:nat,
                Yy,Y,W,Tu,Nu,M1,Sigma,Sku,G,TIDu,PW,Request,IDcsp,G2,W1,G3,Rm,Certu,
        Request1,Z,Tcsp2,Ncsp2,H1IDca2,Ebi,PKu,IDca2,Ncsp:text

        const
                u_csp_sig:protocol_id
        init
                State:=3
        transition
                3.
                State=3/\RCV(IDca2,IDcsp,Ncsp',H1IDca2,Certu)=|>
                State':=9/\Sigma':=Pred(Pred(Pred(G2,Sku),Pred(W1,G3)),Pred(G,Rm))
                /\SND(IDca2,IDcsp,Ncsp',H1IDca2,Certu)
end role
```

```
role session(U,CSP,CA1,KGC:agent,Sk:symmetric_key,H1,H2,Pred,Ebilinear:hash_func,SND,RCV:channel(dy))
def=
        local
                SN1,SN2,SN3,SN4,RV1,RV2,RV3,RV4:channel(dy)
        composition
                role_user(U,CSP,CA1,KGC,Sk,H1,H2,Pred,Ebilinear,SN1,RV1)
                /\role_csp(U,CSP,CA1,KGC,H1,H2,Pred,Ebilinear,SN2,RV2)
                /\role_ca1(U,CSP,CA1,KGC,H1,Pred,Ebilinear,SN3,RV3)
                /\role_kgc(U,CSP,CA1,KGC,H1,Pred,Ebilinear,SN4,RV4)
end role
```

```
role environment()
def=
        const
                u,csp,ca1,kgc:agent,
                sp1,sp2,sp3,sp4,u_csp_sig:protocol_id,
                h1,h2,pred,ebilinear:hash_func,
                sk:symmetric_key
                intruder_knowledge={u,csp,ca2,kgc}
        composition
                session(u,csp,ca1,kgc,sk,h1,h2,pred,ebilinear,h1,h1)
                /\session(i,csp,ca1,kgc,sk,h1,h2,pred,ebilinear,h1,h1)
                /\session(u,i,ca1,kgc,sk,h1,h2,pred,ebilinear,h1,h1)
                %/\session(u,csp,i,kgc,sk,h1,h2,pred,ebilinear,h1,h1)
                %/\session(u,csp,ca1,i,sk,h1,h2,pred,ebilinear,h1,h1)
end role
```

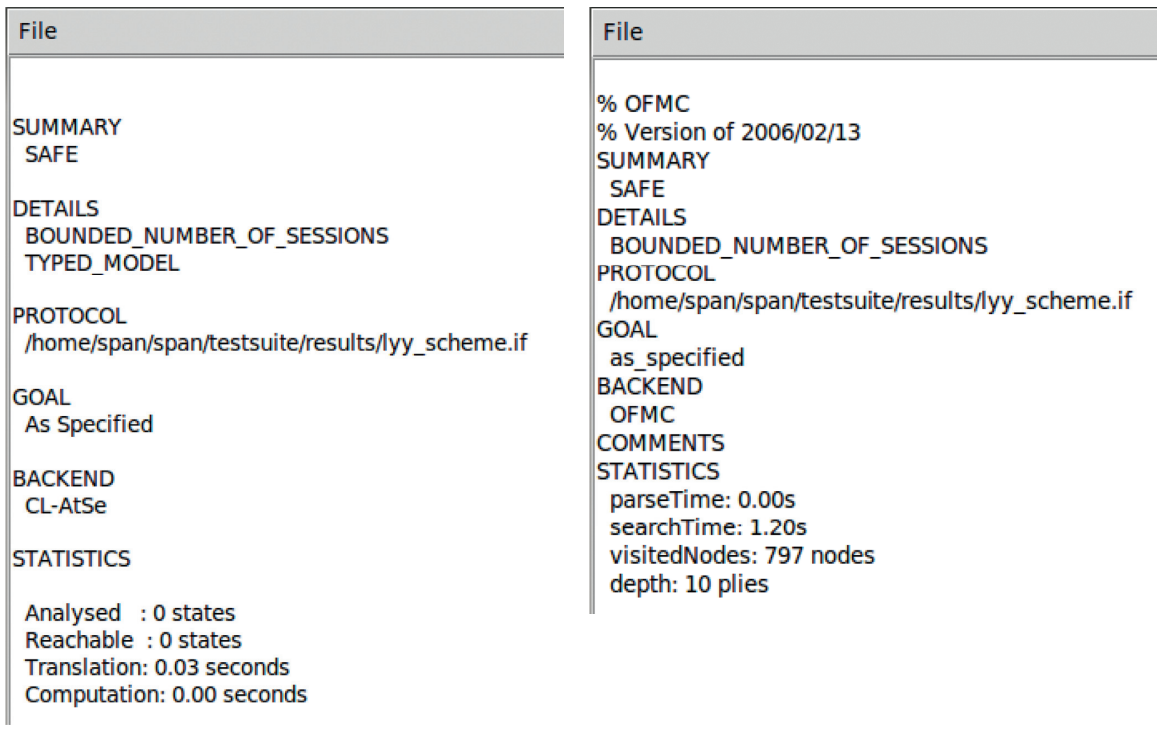
Figure 5: Simulation experiment source code.

Table 2 that this scheme is superior in ensuring security. "No" means that the literature does not meet the performance, and "Yes" means that the literature meets the performance.

This scheme uses hierarchical ID tree to define the ID values of users, cloud service providers, and other entities to realize the uniqueness of entity identity. Compared with literature [34–39], this scheme replaces the real identity with the temporary identity, and the KGC or CA1 signs user U's temporary identity, further enhancing security and meanwhile realizing anonymous tracking. Compared with literature [33–36], the KGC or CA1 in this scheme signs the temporary identity of user U and encrypts the message by the public key of the CSP, which results in better performance in resisting man-in-the-middle attacks. Compared with literature [34, 36], this scheme randomly selects the local timestamp and random parameters for keeping the session fresh to ensure the validity of the message in cross-domain authentication, which realizes the resistance of the replay attack. Compared with literature [37], this scheme can resist the replacement attack by using temporary identities and the two-factor authentication of "password + key," making itself more secure. Compared with literature [33, 34, 37–39], this scheme realizes cross-domain identity authentication under different cryptographic systems, which better satisfies the needs of contemporary society.

4.7. Performance Analysis. On account of the higher computational cost of double linear calculation and exponent operation, compared with multiplication, addition, and hash function, this scheme will be compared with others concerning the computational cost of double linear calculation and exponent operation in the three stages as key generation, the first-time cross-domain authentication, and repeated cross-domain authentication. Pa means the time required for bilinear calculation, and Dex means the time for exponent operation.

As shown in Table 3, the scheme performs two exponential operations in the process of key generation. The first-



(a) (b)

Figure 6: (a) CL-ATSE analysis engine validation results. (b) OFMC analysis engine validation results.

Table 2: Security comparison of heterogeneous cross-domain authentication schemes based on proxy resignature.

Scheme	Anonymous tracking	Anti-man-in-the-middle attack	Antireplay attacks	Antireplacement attacks	Two-factor authentication
[33]	Yes	No	Yes	Yes	Yes
[34]	No	No	No	Yes	No
[35]	No	No	Yes	Yes	No
[36]	No	No	No	Yes	No
[37]	No	Yes	Yes	No	No
[38]	No	Yes	Yes	Yes	No
[39]	No	Yes	Yes	Yes	No
Our scheme	Yes	Yes	Yes	Yes	Yes

Table 3: Comparison of computational cost of the heterogeneous cross-domain authentication schemes based on proxy resignature.

Scheme	Key generation	First cross-domain authentication	Repeated cross-domain authentication
[18]	2Dex + 2 Pa	5Dex + 3 Pa	3Dex
[28]	Dex + 2 Pa	6Dex + Pa	3Dex
[30]	2Dex	7Dex	7Dex
[31]	3Dex	6Dex	3Dex
[33]	2Dex + 2 Pa	3Dex + 4 Pa	3Dex
Our scheme	2Dex	3Dex + 3 Pa	3Dex

time cross-domain authentication needs three-time bilinear calculation and three-time exponent operation. The repeated cross-domain authentication does not require verification of certificate and complex bilinear operation. What is more, the authentication is clearer. Compared with the literature [18, 28, 30, 31, 33], the overall computational efficiency is higher. The computational cost of the first-time cross-domain authentication is close to this scheme and the scheme in literature [30], but the cost of the scheme in literature [30] is much higher in the repeated cross-domain authentication.

Compared with literature [33], after receiving the response, user U does not need to send another authentication request to the intercloud authentication center to guarantee the legitimacy of the identity of the CSP, which increases security and reduces the cost. At the same time, this paper does not use the secure channel when requesting access, which increases the reality of the scheme. According to research, this paper is the first one to propose a cross-domain identity authentication scheme based on proxy resignature under the heterogeneous environment.

5. Conclusion

The authentication based on the PKI password system is the most widely used authentication mechanism at present, and the authentication scheme with certificateless password system can effectively solve such problems as the key escrow problem existing in the IBC system, making it more popular. This paper proposes a heterogeneous cross-domain authentication scheme for the PKI cryptosystem and certificateless cryptosystem, which can anonymously track the entity's identity and effectively resist replay attack, replacement attack, and man-in-the-middle attack. The analysis shows that the heterogeneous cross-domain authentication scheme proposed in this paper has better computing performance and higher security and can effectively meet the current complex requirements for cross-domain access in cloud environment. The next step will be to investigate cross-domain authentication schemes based on lattice or other mathematical problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Key Project Foundation of Tianjin (Grant no. 15ZXHLGX003901), Tianjin Natural Science Foundation (Grant no. 19JCYBJC15800), and National Natural Science Foundation of China (Grant no. 61702366).

References

[1] Y. Zhang, X. Wang, and X. Liu, "Overview of cloud computing environment security," *Journal of Software*, vol. 27, no. 6, pp. 1328–1348, 2016.

[2] Z. Zheng, "Status quo and development on cross-domain authentication based on public key cryptosystems," in *Proceedings of the 2012 International Symposium on Information Technologies in Medicine and Education*, pp. 1106–1109, Hokkaido, Japan, December 2012.

[3] Y. Tang, Y. Zhang, Y. He et al., "A multi-channel unified identity authentication method for energy information system based on PKI," *Telecommunications Science*, vol. 35, no. 6, 2019.

[4] H. A. Elbaz, "Trusting identity based authentication on hybrid cloud computing," in *Proceedings of the 4th International Conference, CloudComp 2013*, Springer, Wuhan, China, October 2013.

[5] T. Junfeng and S. Kehui, "Trust-distributed-based authentication mechanism using hierarchical identity-based cryptography," *Journal of Computer Research and Development*, vol. 52, no. 7, pp. 1291–1312, 2015.

[6] Z. Dong, L. Zhang, and J. Li, "Security enhanced anonymous remote user authentication and key agreement for cloud computing," in *Proceedings of the 2014 IEEE 17th International Conference on Computational Science and Engineering (CSE)*, IEEE, Chengdu, China, December 2014.

[7] M. Blaze, G. Bleumer, and M. Struss, "Divertible protocols and atomic proxy cryptography," in *Proceedings of EUROCRYPT'98*, pp. 127–144, Helsinki, Finland, June 1998.

[8] G. Ateniese and S. Hohenberger, "Proxy re-signatures: new definitions, algorithms, and applications," in *Proceedings of the 12th ACM CCS*, pp. 310–319, Alexandria, VA, USA, November 2005.

[9] X. Yang, G. Gao, and C. Wang, "On-line/off-line threshold proxy re-signature scheme through the simulation approach," *Applied Mathematics & Information Sciences*, vol. 9, no. 6, pp. 3251–3261, 2015.

[10] M. Tian, "Identity-based proxy re-signatures from lattices," *Information Processing Letters*, vol. 115, no. 4, pp. 462–467, 2015.

[11] X. Yang, C. Li, Y. Li et al., "Divisible on-line/off-line proxy resignature," *Applied Mathematics & Information Sciences*, vol. 9, no. 2, pp. 759–767, 2015.

[12] Z. Wang and W. Lu, "Server-aided verification proxy resignature," in *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1704–1707, Melbourne, Australia, July 2013.

[13] X. D. Yang, Y. N. Li, G. J. Gao et al., "Sever-aided verification proxy re-signature scheme in the standard model," *Journal of Electronics & Information Technology*, vol. 38, no. 5, pp. 1151–1157, 2016, in Chinese.

[14] N. Aye, H. S. Khin, T. T. Win et al., "Multi-domain public key infra structure for information security with use of a multi-agent system," *Intelligent Information and Database Systems*, pp. 365–374, Springer, Berlin, Germany, 2013.

[15] X. Lu and D. Feng, "An identity-based authenti- cation model for multi-domain grids," *Acta Electronica Sinica*, vol. 34, no. 4, pp. 579–582, 2006.

[16] B. Yang, G. Q. Chen, and Y. H. Sun, "Research on a new identity-based authentication model for multi-domains," *Computer Security*, vol. 8, pp. 15–18, 2010.

[17] B. Yang, *Research on the Combination of Identity-Based Cryptographic Techniques and Public Key Infrastructure*, PLA Information Engineering University, Zhengzhou, China, 2009.

[18] Z. Wang, Z. Han, and J. Liu, "Identity authentication scheme based on PTPM and certifificateless public key in cloud environment," *Journal of Software*, vol. 27, no. 6, pp. 1523–1537, 2016.

[19] Y. Li, W. Chen, Z. Cai, and Y. Fang, "CAKA: a novel certificateless-based cross-domain authenticated key agreement protocol for wireless mesh networks," *Wireless Networks*, vol. 22, no. 8, pp. 2523–2535, 2016.

[20] X. Ma, W. Ma, and X. LIU, "A cross domain authentication scheme based on blockchain technology," *Acta Electronica Sinica*, vol. 046, no. 011, pp. 2571–2579, 2018.

[21] Z. Jiang and J. Xu, "Heterogeneous cross-domain identity authentication scheme based on signature in cloud environment," *Journal of Computer Applications*, vol. 40, no. 3, pp. 740–746, 2020.

[22] C. Yuan, W. Zhang, and X. Wang, "EIMAKP: heterogeneous cross-domain authenticated key agreement protocols in the EIM system," *Arabian Journal for Science and Engineering*, vol. 42, no. 8, pp. 3275–3287, 2017.

[23] C. Wang, C. Liu, Y. Li et al., "Two-way anonymous heterogeneous signcryption scheme based on PKI and IBC China," *Journal of Communications*, vol. 10, pp. 14–21, 2017.

[24] M. A. Ferrag, L. A. Maglaras, H. Janicke et al., "Authentication protocols for internet of things: a comprehensive survey," *Security and Communication Networks*, vol. 2017, Article ID 6562953, 2017.

[25] C. Wang, C. Liu, S. Niu et al., "An authenticated key agreement protocol for cross-domain based on heterogeneous signcryption scheme wireless communications and mobile computing conference (IWCMC)," in *Proceedings of the 2017 13th International*, IEEE, Seoul, Korea, pp. 723–728, August 2017.

[26] B. He, "Improvement and Research on Mechanism of Certificate Revocation Based on PKI," Master's thesis in Chinese, Shanghai Jiao Tong University, Shanghai, China, 2015.

[27] Y. Zhang, "Design of Cross-Domain Authentication System for Multiple Security Element Based on PKI," Master's thesis in Chinese, Taiyuan University of Technology, Taiyuan, China, 2015.

[28] X. Yang, F. An, YangPing et al., "Cross-domain authentication scheme based on proxy re-signature in cloud environment," *Chinese Journal of Computers*, vol. 42, no. 4, pp. 756–771, 2019.

[29] T.-T. Tsai, Y.-M. Tseng, and S.-S. Huang, "Efficient strongly unforgeable id-based signature without random oracles," *Informatica*, vol. 25, no. 3, pp. 505–521, 2014.

[30] L. Yang, J.-F. Ma, and Q. Jiang, "Direct anonymous attestation scheme in cross trusted domain for wireless mobile networks," *Journal of Software*, vol. 23, no. 5, pp. 1260–1271, 2012.

[31] Y.-W. Zhou, B. Yang, Z.-Q. Wu et al., "Direct anonymous authentication scheme in cross-domain based on identity," *Chinese Science: Information Science*, vol. 44, no. 9, pp. 1102–1120, 2014.

[32] H. Li, Y. Dai, L. Tian et al., "Identity-based authentication for cloud computing," 2009.

[33] X. Yang, F. An, P. Yang et al., "Cloud-based cross-domain identity authentication scheme without certificate signature," *Computer Engineering*, vol. 2017, no. 11, pp. 134–139, 2017.

[34] Z. Zhou, L. LI, and Z. Li, "Efficient cross-domain authentication scheme based on block chain technology," *Computer Engineering*, vol. 38, no. 02, pp. 316–320, 2018.

[35] Z. Qikun, G. Yong, Z. Quanxin et al., "A dynamic and cross-domain authentication asymmetric group key agreement in telemedicine application," *IEEE Access*, vol. 6, pp. 24064–24074, 2018.

[36] Z Li, Z. Chen, Y.-L Qin et al., "Cross-domain certification scheme based on zero-knowledge proof," *Computer & Digital Engineering*, vol. 42, no. 3, pp. 446–449, 2014.

[37] D. He, S. Zeadally, N. Kumar, and J.-H. Lee, "Anonymous authentication for wireless body area networks with provable security," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2590–2601, 2017.

[38] Y. Chen, G. Dong, J. Bai, Y. Hao, F. Li, and H. Peng, "Trust enhancement scheme for cross domain authentication of PKI system," in *Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Cyber C*, pp. 103–110, Guilin, China, October 2019.

[39] X. Jia, S. N. Hu, Y. ZhaoYin, X. Cheng, and C. Zhang, "IRBA: an identity-based cross-domain authentication scheme for the internet of things," *Electronics*, vol. 9, no. 4, p. 634, 2020.