In the field of cloud computing, most research on identity management has concentrated on protecting user data. However, users typically leave a trail when they access cloud services, and the resulting user traceability can potentially lead to the leakage of sensitive user information. Meanwhile, malicious users can do harm to cloud providers through the use of pseudonyms. To solve these problems, we introduce a reputation mechanism and design a reputation-based identity management model for cloud computing. In the model, pseudonyms are generated based on a reputation signature so as to guarantee the untraceability of pseudonyms, and a mechanism that calculates user reputation is proposed, which helps cloud service providers to identify malicious users. Analysis verifies that the model can ensure that users access cloud services anonymously and that cloud providers assess the credibility of users effectively without violating user privacy.

1. Introduction

Cloud computing is a computing model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and other services) that can be rapidly provisioned and released with minimal management effort or cloud service provider (CSP) interaction [1]. Large numbers of users may simultaneously engage in cloud computing services, making the “multitenant” feature an important property of cloud computing according to the Cloud Security Alliance (CSA) [2]. However, the multitenant property brings the following new problems:

(i) Privacy leaks because of external user data: in an open environment, users must be authenticated to access cloud services. If users employ their actual names (or fixed usernames) to log in, sensitive information such as login names or even long-term behavior may be revealed by data mining or other techniques and used illegally by the CSP.

(ii) Management problems caused by an excessive number of tenants: to preserve privacy, users must employ different pseudonyms to access each cloud computing session. In this case, it is difficult for users to recall a large number of pseudonyms and passwords. Meanwhile single-use pseudonyms make no contribution to the development of a user’s reputation.

(iii) Security threats to the CSP caused by multiple tenants: by accessing cloud services, malicious users may take the opportunity to attack the CSP through activities such as stealing data, performing vulnerability scanning, and launching denial of service (DoS) attacks. In particular, if allowed to log in by a pseudonym, malicious users can launch whitewash attacks, where the malicious user can continue to visit the CSP normally after an attack or initiate another attack by creating a new pseudonym.

Considering the above problems, traditional identity management mechanisms that store user identities in a database directly are no longer applicable [3]. In this paper, we design a new identity management model based on user reputation, which is herein denoted as reputation-based identity management (RIM). The main contributions of RIM are listed as follows:
(i) Anonymous user access: in RIM, we design a method in which each user takes a different pseudonym for each session when accessing cloud services. No link between a user identity and a corresponding pseudonym is provided, and no link is provided between the pseudonyms of a single user. Pseudonym usage does not affect user attestation, and it decreases the input of private user information, rendering it impossible for tenants to spy on each other.

(ii) User attestation: in our model, users firstly register with an identity provider (IdP), which provides the user with a formal identity certificate. The identity certificate is the basis for the user to prove their legitimate status to the CSP.

(iii) Reputation attestation: RIM records and determines user reputations as user activities accumulate with access to successive cloud computing sessions and provides proof of the reputation. The proof assures the credibility of the reputation, ensuring that the reputation indeed belongs to its owner. The proof also guarantees nonrepudiation; that is, a user cannot deny the reputation assigned to them. As such, the proof ensures unforgeability. Users cannot promote their reputation without the authorization of the IdP. The introduction of a reputation does not affect the anonymity of users.

The remainder of this paper is organized as follows. Section 2 describes related work. Section 3 introduces the background and technology of cloud computing along with identity management. Section 4 introduces the proposed RIM and describes its design and realization. Section 5 analyzes the correctness and security of our model. Finally, the last section concludes the paper and proposes future work.

2. Related Work

The focus of identity management for cloud computing is user privacy. In [4], the authors proposed an approach to preserve the privacy of users based on zero-knowledge proof protocols and semantic matching techniques. The approach also enhanced the interoperability across multiple domains. Although the study realized the goal of concealing user identities, the CSP could still obtain sensitive information through data mining techniques because user identities were consistent throughout the entire process. In [5], the authors proposed an entity-centric approach for identity management in cloud computing. The approach was based on personally identifiable information (PII) and on anonymous identification to mediate interactions between users and cloud services. Although the identification was anonymous, the PII released privacy related information. In [6], the authors took advantage of attribute-based encryption and signature technology to conceal user identities in cloud computing. However, because users must employ a single unchanging certificate to obtain authentication from the CSP, an attacker may ascertain a user’s identity easily through the static certificate. In [7], the authors improved the approach proposed in [4], where the registry center was not required to be online at all times. However, the improved approach had the same disadvantages with respect to user privacy protection as the original approach. In [8, 9], the authors designed and introduced a method that integrated blind signature and hash chain techniques to protect user privacy in cloud computing. However, in every session, users employ the same pseudonym to log in, which results in linkability between different sessions. Although the above studies achieve the goal of user identity concealment, the successive behaviors of individual users can be associated. Thus, through analyzing user behaviors, the CSP can potentially compromise user privacy. In [10], the authors proposed a method for anonymity using group digital signature technology. Because this method introduced no correlation between user signatures, this method is an improvement over the aforementioned methods. However, the use of group digital signature technology forbids members from joining and leaving a group dynamically, which conflicts with the openness of cloud computing.

In this paper, we propose an approach that achieves user identity anonymity. Moreover, by this approach, the CSP is unable to link user behavior to user identity. While the anonymity of identity protects user privacy, it tends to enable CSP attack by malicious users because the CSP is unable to trace the users involved in the attack. Therefore, our approach introduces trust management to address the shortcomings of identity anonymity. The approach determines user reputations and binds the reputation to user identity. The CSP utilizes user reputation to distinguish malicious users and to reduce the threat of attack by malicious users [11].

Previous reputation research has focused on peer to peer (P2P) systems, which has been developed into various systems, for example, the EigenTrust [12], the PeerTrust [13], and the PowerTrust [14] systems. Along with the development of electronic business, reputation research has had a service oriented focus, where the reputation of service providers has been evaluated to protect consumers. In [15, 16], reputation was employed as a means of choosing service providers. In cloud computing, most research has focused on protecting the CSP, such as what was done in [17, 18]. However, the use of a reputation model to manage users in cloud computing, such as that employed in this paper, has not been the subject of previous research. In [19], the authors designed an identity management model for a noncloud computing environment. The model supposed that users must be previously registered, which conflicts with the required openness of cloud computing. In [20, 21], the authors designed a reputation-based identity management model that used online systems for applications in, for example, electronic business and forums. The online system must recognize user identities or unique identifiers to accumulate reputation data. However, tracking user identities tends to leak user information, which threatens privacy. This is why these approaches cannot be applied to cloud computing. Our proposed RIM introduces reputation to manage users and simultaneously protects user privacy, which has not been investigated by other researchers.
User privacy protection requires a user identity to be anonymous such that it cannot be tracked. However, the determination of reputation must confirm user identity and track user transaction history. Therefore, reputation management and identity anonymity appear to be contradictory [22]. In [22], the authors alleviated this contradiction and designed an anonymous reputation system in a P2P environment using electronic cash technology to provide feedback to users according to behavior. However, according to this approach, the amount of feedback one user gives to another is restricted by the quantity of electronic coins in the former user's possession. A user with no electronic coins is therefore barred from involvement in the transaction. As such, this method contradicts with the openness of cloud computing. In [23], the authors used blind signature technology to achieve a reputation-based identity management approach in the C-S mode. This approach provides the service provider with user reputations while guaranteeing user anonymity. However, under this approach, a service provider can confirm that orders derive from the same user and accumulate the history of user actions, resulting in the potential violation of user privacy. In [24], authors achieved an anonymous reputation system based on zero-knowledge authentication and digital signature technology. However, [4] the sessions between the user and service provider under this approach are linkable, enabling service providers to violate user privacy. From the above analysis, we can see that these last two examples manage to provide user reputation only by divulging user identity. Our RIM overcomes these shortcomings by determining user reputation while ensuring that user identity and pseudonyms are unlinkable.

3. Technology Background

3.1. Cloud Computing Architecture. Cloud computing provides three main service delivery models to the public, including the following [2]:

(i) Software as a Service (SaaS): this is a software delivery model in which software and its associated data are hosted in the cloud and are typically accessed by users using a thin client.

(ii) Platform as a Service (PaaS): this is the delivery of a computing platform and solution stack as a service, which provides all the facilities required to support the complete life cycle of building and delivering web applications and services from the Internet.

(iii) Infrastructure as a Service (IaaS): this delivers computer infrastructure (typically a platform virtualization environment) as a service, along with raw storage and networks. Rather than purchasing servers, software, data-center space, or network equipment, clients purchase the resources as a fully outsourced service.

Based on the above approach, a number of similar abstract service models have been recently promoted, such as Hardware as a Service (HaaS), Data as a Service (DaaS), and Communication as a Service (CaaS). Cloud computing management services ensure the reliability, availability, and security of core services. Figure 1 [26, 27] illustrates the basic service architecture of cloud computing.

Cloud identity management takes charge of identity management throughout the entire cloud services stack. Identity management can be divided into three categories [28]: isolated user identity model, federated user identity model, and centralized user identity model, which are defined as follows:

(i) Isolated user identity model: in this model, the service provider acts as both credential provider and identifier provider to their clients. A user obtains separate unique identifiers from each service/identifier provider transacted with.

(ii) Federated user identity model: this can be defined as the set of agreements, standards, and technologies that enable a group of service providers to recognize user identifiers and entitlements from other service providers within a federated domain.
(iii) Centralized user identity model: this model employs a single identifier and credential provider that is used by all service providers to provide identifiers and credentials to users.

In the centralized user identity model, the job of identity management has been taken over by an identity provider from the service provider. This reduces the burden of the service provider and also reduces the number of certificates required by users to hold. In addition, the centralized user identity model facilitates the delivery of the identity management service from the CSP to the users. In an identity provider management domain, users first obtain a formal certificate and then access cloud services in the same domain using this certificate. This property makes the centralized user identity model suitable for cloud computing, and, therefore, we employ a centralized model in our design, as shown in Figure 2.

3.2. Basic Technology. We introduce identity-based signature, blind signature, and zero-knowledge authentication technology to implement RIM.

The identity-based signature was proposed by Shamir [29] in 1984. Here, a user’s identity is first disclosed as a public key and then used to generate a private key. In [29], the authors designed a conceptual model but did not provide for any real implementation. Since then, studies of the identity-based signature have been conducted, but no efficient and provable signature scheme was concluded until Boneh et al. [30, 31] promoted an identity-based encryption scheme. Identity-based signature comprises four stages: setup, private key extraction, signature, and verification. Using an identity as a public key has a natural legitimacy, which can simplify the process of key distribution.

The concept of blind signature was suggested by Chaum [32]. The approach ensures that the signer does not know the specific content of the message and that the message owner can obtain the signature of the message. Blind signature consists of three operations: blinding, signing, and blindness removal. Blind signature has many characteristics conducive to protecting user privacy such as (1) blindness, where the signer does not know the specific content of the message to be signed and (2) untraceability, where, if a signature is leaked, the signer has no idea of when and by whom the message was signed.

Zero-knowledge proof authentication technology refers to a prover’s assurance of ownership of some secret information, without revealing any useful knowledge about the secret information to the verifier. Well-known zero-knowledge authentication schemes mainly include the Feige-Fiat-Shamir [33] scheme, the Guillou-Quisquater [34] scheme, and the Schnorr [35] scheme.

The proposed RIM achieves an identity management model that is suitable for cloud computing by employing methods related to the above techniques that are most suitable for specific application scenarios. In Section 4.3, we give a detailed exposition of our design.

4. Model Design and Implementation

In this section we firstly give some assumptions for RIM and then explain the detailed implementation of the model.

4.1. Assumptions. We must make some reasonable assumptions to ensure proper RIM functionality. Firstly, we assume the existence of an independent arbitration agency whose function is similar to a governmental authority that can ensure user privacy while simultaneously providing a secure source of anonymous user information under very specific circumstances. As such, the agency proposed herein is different from a trusted third party in an ordinary sense. The agency need not have a continual online presence. Moreover, the agency does not collude with other users, and user information is protected from malicious users. Because cloud services are open to the public as a paid service, it is necessary to reveal anonymous user information when a CSP suffers an attack or has economic disputes with users. Therefore, RIM must be able to identify anonymous users to address these specific problems. If a CSP seeks anonymous information, it must provide a range of certificates and follow prescribed security protocols to ensure that the application submitted is legitimate.

Another important assumption is that the communication channel in RIM is secure. SSL and IPSec protocols can be used to ensure the security of the channel.

4.2. Model Design. Four roles are managed in RIM: User, CSP, IdP, and deanonymizing authority (DA), which are defined as follows:

(i) User: the consumer of cloud services who requests identity anonymity and benefits from the service.

(ii) CSP: the service provider who, after the transaction between the User and CSP, provides feedback regarding the transaction.

(iii) IdP: the service provider who not only provides registration services to the User but also determines the User reputation, which is indicative of the degree
of trust, based on the feedback obtained from the CSP. The IdP issues the User reputation certificate.

(iv) DA: an authority that can reveal User pseudonyms and provide User identity-related information to the CSP.

RIM has five stages, namely, Environment Initialization, User Registration, Identity Authentication, Reputation Computation, and Pseudonym Disclosure.

In the Environment Initialization stage, RIM first creates the public parameters of the IdP, DA, and CSP. It then generates the public and private keys used for signatures. The operations of key generation are as follows: \( \lambda \) is the security parameter, \( PK \) denotes the public key, and \( SK \) denotes the private key. The following describes the probabilistic polynomial time algorithm:

(i) \( (PK_{IdA}, SK_{IdA}) \leftarrow Setup_{IdP_{Id}}(1^{\lambda}) \): it is a key-generation algorithm that takes \( \lambda \) as the input and outputs a pair \( (PK_{IdA}, SK_{IdA}) \) of public and secret keys used by the IdP to sign the identity certificate.

(ii) \( (PK_{IdA}, SK_{IdA}) \leftarrow Setup_{IdP_{blind}}(1^{\lambda}) \): given the security parameter \( \lambda \), it creates a pair of keys \( (PK_{IdA}, SK_{IdA}) \) which is used by the IdP to generate a blind signature.

(iii) \( (PK_{CSP}, SK_{CSP}) \leftarrow Setup_{CSP}(1^{\lambda}) \): it outputs a pair of keys \( (PK_{CSP}, SK_{CSP}) \) which is used by the CSP to create a feedback certificate.

(iv) \( (PK_{DA}, SK_{DA}) \leftarrow Setup_{DA}(1^{\lambda}) \): it gives the DA a pair of keys \( (PK_{DA}, SK_{DA}) \) which is used to encrypt and decrypt User identities.

Keys \( (PK_{IdA}, SK_{IdA}) \) and \( (PK_{IdA}, SK_{IdA}) \) can be combined into a single pair. However, in such a situation, if the keys are disclosed, the IdP will collapse. When they are separated, if one pair of keys is compromised, for example, if the blind signature key is compromised, only the credibility of the reputation is affected while the identity certificate remains valid. Owing to security considerations, we maintain these two key pairs as separate.

The Registration operation and Verification operation contained in the User Registration stage are defined as follows:

(i) Registration: the User initiates the operation \( (C_{Id}, rep, C_{rep}) \leftarrow Register(SK_{Id}, Id) \) to register with the IdP. The User obtains an identity \( (Id) \) as the input and then receives the identity certificate \( C_{rep} \), reputation \( (rep) \), and reputation certificate \( C_{rep} \). Here, \( rep \) is the initial reputation value for new users given by the IdP.

(ii) Verification: the operation \( 1/\bot \leftarrow CheckReg(PK_{IdA}, C_{Id}, rep, C_{rep}) \) is employed by the User to authenticate \( C_{Id}, rep, C_{rep} \). This operation takes the IdP public key \( PK_{IdA} \) as one of the inputs, which will confirm the legitimate source of these certificates. If verified, \( 1 \) will be returned as the result; otherwise, \( \bot \) will be returned.

The first time a User enters the cloud service system, the User must register with the IdP using their own identity. The IdP will determine whether the identity is redundant or on the blacklist. If the answer is no, the verification is passed. The IdP gives the User an initial reputation value and issues an identity certificate and reputation certificate. The identity that registered to the IdP is known to the public. It can be a URL or e-mail address associated with the User. The public identity carries less privacy and is easily verified by the IdP. Because of the openness of cloud services, the rules for accessing cloud services cannot be overly strict. In RIM, all users that meet the basic safety requirements are allowed access. After Registration, the Verification operation is performed to verify the identity certificate and the reputation certificate. The above processes are illustrated in Step 1 of Figure 3(a).

After the User has a recognizable valid identity, access is granted to the CSP using the pseudonym generated according to this valid identity, and the CSP will authenticate the pseudonym. The Identity Authentication stage includes the Pseudonym Generation operation and Authentication operation shown in steps 2 and 3 of Figure 3(a), which are described as follows:
(i) Pseudonym Generation: taking Id, the User identity certificate $C_{\text{id}}$, and rep as the inputs, the operation $P_u \leftarrow \text{Gen}_\text{pmy}(Id, C_{\text{id}}, \text{rep})$ outputs the User pseudonym $P_u$, which is the only identification presented to the CSP.

(ii) Authentication: the operation $1/\bot \leftarrow \text{Authenticate}(P_u, C_{\text{rep}})$ is used to authenticate the User identity through $P_u$ by the CSP. This operation is also used to confirm the credibility of the User reputation.

The Pseudonym Generation operation encrypts Id using the public key provided by the DA. As such, if a dispute arises between the User and the CSP, RIM would decrypt $P_u$ with the private key provided by the DA to restore the User Id. The Authentication protocol applies the $\Sigma$-protocol to conduct the authentication. During the process, the CSP can extract $P_u$, $C_{\text{rep}}$, and rep but not the User Id.

After the transaction between the User and the CSP, RIM enters the Reputation Computation stage, where the User rep is updated on the basis of feedback provided by the CSP. This stage includes a series of operations including the Reputation Backfeeding operation, Blinding operation, Applying for Reputation Update operation, Blind signing operation, and Reputation Update Confirmation operation, which are defined as follows:

(i) Reputation Backfeeding: the CSP calls $(\text{rep}_\text{new}, C_{\text{rep}_\text{new}}) \leftarrow \text{Gran}(P_u)$ and provides reputation feedback $\text{rep}_\text{new}$ and its certificate $C_{\text{rep}_\text{new}}$ according to the performance of $P_u$. The $C_{\text{rep}_\text{new}}$ guarantees that $\text{rep}_\text{new}$ is from the CSP.

(ii) Blinding: $P_u$ randomly selects the blind factor $\text{Nonce}$ as the input of the operation $\text{Nonce}_\text{blind} \leftarrow \text{Blind}(\text{Nonce})$ to obtain the blinded value $\text{Nonce}_\text{blind}$.

(iii) Applying for Reputation Update: given the inputs $\text{rep}_\text{new}, C_{\text{rep}_\text{new}}, \text{Nonce}_\text{blind}$, $P_u$ calls $1/\bot \leftarrow \text{Update}(\text{rep}_\text{new}, C_{\text{rep}_\text{new}}, \text{Nonce}_\text{blind})$ to apply for reputation updating. The IdP verifies the reputation feedback based on $\text{rep}_\text{new}$ and $C_{\text{rep}_\text{new}}$. A successful operation returns 1; otherwise, it returns $\bot$.

(iv) Blind Signing: the IdP calls $C_{\text{blind}} \leftarrow \text{Blind}\_\text{sign}(\text{Nonce}_\text{blind})$ to generate the blind signature $C_{\text{blind}}$ of the value $\text{Nonce}_\text{blind}$.

(v) Reputation Update Confirmation: the User removes the blindness of $C_{\text{blind}}$ and obtains the certificate $C_{\text{Nonce}}$ of $\text{Nonce}$. Then, the User calls $1/\bot \leftarrow \text{Confirm}\_\text{Update}(C_{\text{Nonce}})$ to submit the request for confirming the update of reputation to the IdP. If the IdP successfully updates the User rep, it returns 1; otherwise, it returns $\bot$.

The CSP provides feedback to $P_u$ using the Reputation Backfeeding operation, as illustrated by Step 5 in Figure 3(b). The Applying for Reputation Update Confirmation operation submits the reputation feedback $\text{rep}_\text{new}$ and a blinded random value $\text{Nonce}_\text{blind}$ to the IdP and expects that the IdP will update $\text{rep}$. This operation is illustrated by Step 6 in Figure 3(b). The IdP verifies the feedback and determines that it is valid. However, the IdP cannot determine for whom the feedback is designated because the IdP cannot track the feedback to the User Id. The IdP uses the Blind Signing operation to sign $\text{Nonce}_\text{blind}$ and gives the result $C_{\text{blind}}$ back to $P_u$, as illustrated by Step 7 in Figure 3(b). Because the User generates $P_u$ directly, the User can locate their own $P_u$. The User obtains $C_{\text{blind}}$ through $P_u$ and then removes the blindness of $C_{\text{blind}}$ to obtain $C_{\text{Nonce}}$. The User subsequently calls the Reputation Update Confirmation operation and submits $C_{\text{Nonce}}$ to the IdP to confirm the updating of rep, which is illustrated by Step 8 in Figure 3(b). The IdP verifies $C_{\text{Nonce}}$, and, if passed, the value of $\text{rep}$ is updated. The IdP cannot link $\text{Nonce}$ to $\text{Nonce}_\text{blind}$, which is why the blind signature is introduced. This method ensures that, even in the case of collusion between the IdP and CSP, the IdP cannot obtain the User Id through $P_u$.

The aforementioned processes address the entire process of new user access to cloud services, including User Registration, accessing services, and reputation update. The User who is already registered must first call the Reputation Withdrawal operation to acquire a value for $\text{rep}$ from the IdP and then follow all subsequent operations, as the aforementioned. The Reputation Withdrawal operation is illustrated in Figure 4(a), which is described as follows:

![Figure 4: Schematic of the standard access process and Pseudonym Disclosure.](image-url)
(i) Reputatian Withdrawal: the User acquires \(rep\) and \(C_{rep}\) by the operation \((rep, C_{rep}) \leftarrow Withdraw(Id)\) using its own \(Id\).

The last stage is the Pseudonym Disclosure stage. If the User has done harm to the CSP, the CSP must gain access to the User \(Id\). This stage includes the Anonymity Removal operation and the Identity Mapping operation, which are described as follows:

(i) Anonymity Removal: when the operation \(g^{id} \leftarrow \text{De-Anonymity}(P_u, SK_{DA})\) is called, the DA opens \(P_u\) using its private key \(SK_{DA}\). The output of this operation is not the User \(Id\) itself but \(g^{id}\), which is generated by the IdP, where \(g\) is one of the public parameters.

(ii) Identity Mapping: \(Id\) calls \(Id \leftarrow \text{Map}(g^{id})\) to map \(g^{id}\) to the User \(Id\) and then resolves the disputes offline.

The CSP submits a malicious behavior report for the User and pseudonym \(P_u\) to the DA and applies for opening a pseudonym, as illustrated in Step 10 in Figure 4(b). The DA uses the Anonymity Removal operation to open \(P_u\) and obtains \(g^{id}\). The DA submits \(g^{id}\) to the IdP. Then the IdP calls the Identity Mapping operation and retrieves the User \(Id\), as illustrated by Step 11 in Figure 4(b). Therefore, for security considerations, the DA does not recover the \(Id\) directly. This is one of the methods that ensure user privacy and prevent the misuse of a user’s identity.

The above provides an overview of RIM. The following provides a concrete realization of our model.

4.3. RIM Realization. In this section we provide a description of the concrete realization of RIM based on Hidden Identity-Based Signatures proposed by Kiayias and Zhou [36]. Hidden Identity-Based Signatures meet user requirements for anonymity in a cloud computing environment, but they do not satisfy the need for a unique user pseudonym for each session. We therefore extended Hidden Identity-Based Signatures by introducing reputation and blind signature to fulfill the requirements mentioned above. In RIM, we achieve blind signature using the Schnorr scheme [25]. The public parameters discussed previously [25, 36] are identical, so the RIM is made more comprehensive by combining the two techniques.

In the Environment Initialization stage, we generate public parameters \((p, g, G, G_T, e)\), where \(G\) and \(G_T\) are cyclic groups of prime order \(p\), \(g\) is a generator for \(G\) and \(G_T\), \(e\) is a bilinear map, \(e: G \times G \rightarrow G_T\), and \([G] = [G_T]\). \(h\) is selected from \(G \setminus \{1\}\) randomly. \(H\) is a hash function, \(H : \{0, 1\}^* \rightarrow Z_p\). The operation \(Setup_{IDP}.Jsign(1^k)\) randomly selects \(x, y \leftarrow Z_p\) and computes \(X = g^n, Y = g^l\), where the public and private key pairs are given as \(PK_{id,j} = (X, Y), SK_{id,j} = (x, y)\). Similarly, the operation \(Setup_{IDP}.blind(1^k)\) generates the key pair \(PK_{id,b} = M, SK_{id,b} = m\), where \(m = r^s Z_p^*\) and \(M = g^m\). The operation \(Setup_{CSP}(1^k)\) generates the key pair \(PK_{CSP} = (I, J), SK_{CSP} = (i, j)\) for feedback signature using the same method described above. The key pair generation of \(Setup_{IDP}(1^r)\) is more complicated. It randomly selects \(u, v \leftarrow G, w \leftarrow G \setminus \{1\}\) and \(b, d \leftarrow Z_p\), resulting in \(w = u^b = v^d\), which provides the key pair \(PK_{DA} = (u, v, w), SK_{DA} = (b, d)\). After the keys are generated, RIM publishes the public keys and keeps the private keys secret.

In the User Registration stage, the IdP provides the User with the identity certificate, reputation, and reputation certificate. We introduce the technology of short signatures [37] to issue these certificates. The IdP randomly selects \(r, q \leftarrow Z_q\) and then issues identity certificate \(C_{id} \leftarrow (g^{1/(x+l+kr)}, r)\) and reputation certificate \(C_{rep} \leftarrow (g^{1/(x+re+q)}, q)\), noting that \(Id\) and \(rep\) denote the identifier and reputation belonging to the User. After that, the IdP gives the User a triple \((C_{id}, C_{rep}, rep)\).

The User verifies these certificates using the IdP public key. When the equation \(e(g^{1/(x+l+kr)}, r)Xg^{1/(x+re+q)} = e(g, g)\) holds, the User accepts the identity certificate. In the same way, the User verifies the reputation certificate. If the two certificates are both valid, the operation \(CheckReg(PK_{id,j}, C_{id}, C_{rep}, rep)\) returns 1. If either certificate is invalid, both certificates are discarded.

The Pseudonym Generation and Authentication operations in the Identity Authentication stage are usually carried out together in practice. In this paper, we extend the method introduced in [36] to authenticate the User identity, User identity certificate, and the link between the User identity and identity certificate. Our approach requires verification that the reputation generated from the pseudonym belongs to the User we have just authenticated. The specific process is described as follows:

(i) The User employs linear encryption [37] to commit \(Id\) in \((U, V, W)\). It satisfies the constraints \(U = w^t, V = v^t, W = w^{k+1} g^{id}\), where \(l\) and \(k\) are randomly selected: \(K, l \leftarrow Z_p\).

(ii) The User commits \(C_{id} \leftarrow (S, R)\); that is, \(S = g^l C_{id}\) and \(R = g^r h^l Y^t\), where \(r_1\) and \(r_2\) are randomly selected: \(r_1, r_2 \leftarrow Z_p\).

(iii) The User authenticates their identity using the zero-knowledge proof technique [35], as described in Figure 5. Finally, if the equation holds, the User is assured of possessing a legitimate identity, although the CSP would have no knowledge of that identity.

(iv) The process of authenticating the User identity certificate is described in Figure 6. If the equations hold, the User is assured of possessing a legitimate identity certificate, although the CSP would have no knowledge of that certificate.

(v) It must also be verified that the identity and the identity certificates belong to the same User. The verification process is described in Figure 7. If the equations are true, the identity is bound to the identity certificate.

(vi) Moreover, the legitimacy of the reputation and that the reputation belongs to the User must also be verified. The reputation is open to the public, so it is authenticated using the IdP public key. When verifying that the reputation belongs to the User, the User identity is hidden. To this end, a secret message is selected, and ownership of the reputation certificate can be verified only if the User has the secret
Thus far, the legitimacy of the \( \text{Id}_d, C_{\text{Id}}, \) and \( C_{\text{rep}} \) has been verified. Moreover, the linkability between the \( \text{Id} \) and the \( C_{\text{Id}} \) and \( C_{\text{rep}} \) has been verified. Though the \( \text{Id} \) has been encrypted, it can be verified that the \( C_{\text{Id}} \) derives from the \( \text{IdP} \).

Together with the verification of the legitimacy of the reputation, all of the above comprise the operation \( \text{Authenticate}(P_u, C_{\text{rep}}) \). Because the reputation need not be secret, the reputation can be authenticated simply by inputting the \( \text{IdP} \) public key.

Next, we address the \( e(C_{\text{rep}}, X(g^{e(g_Y d)}) = e(g, g) \) operation \( \text{Gen\_pny}(Id, C_{\text{Id}}, \text{rep}) \). Here, the pseudonym \( P_u \) is just the random challenge \( \sigma \) used by the CSP for authenticating the \( \text{User} \). \( P_u \) meets cloud computing security requirements. The security analysis will be conducted in the next section.

Our implementation of the Reputation Computation stage is described as follows:

(i) Reputaiton Withdrawal: the operation \( \text{Withdraw}(\text{Id}) \) returns the reputation certificate, which is signed by the \( \text{IdP} \). The specific implementation of the reputation certificate is equivalent to that of the identity certificate.

(ii) Reputation Backfeeding: we continue to use the short signature \( [38] \) to sign the feedback given by the CSP; that is, \( C_{\text{rep, new}} = (g^{(i+\text{rep, new})} m, n) \), where \( n \) is randomly selected: \( n \leftarrow Z_p \).

(iii) Blinding: this operation is performed by the \( \text{User} \) and the \( \text{IdP} \). The \( \text{IdP} \) possesses the key pair \((M, m)\). The \( \text{IdP} \) randomly selects \( r_b, r_h \in Z_p \) and computes \( x_b = g^{r_b} \). Then, \( P_u \) obtains \( x_b \) from the \( \text{IdP} \) and selects a random number \( \text{Nonce} \) that is to be signed. \( P_u \) computes \( x_b^* = g^{r_h} M^{-r_b} x_b, \) \( e_{\text{Id}} = H(x_b^*, \text{Nonce}) \) and \( e_{\text{User}} = e_{\text{Id}} + d_{\text{bh}} \), where \( d_{\text{bh}} \leftarrow Z_p, t_b \leftarrow Z_p \). The \( \text{User} \) assigns \( e_{\text{User}} \) as the blind signature of \( \text{Nonce} \); that is, \( e_{\text{User}}^{\text{blind}} = e_{\text{User}} \).

(iv) Applying for Reputation Update. The \( \text{IdP} \) verifies the reputation feedback, which is signed by the CSP. If the equation \( e(g^{(1/(i + \text{rep, new})} m, l_{\text{rep, new}}^m n)) = e(g, g) \) holds, we conclude that the verification has passed.

(v) Blind Signing: the \( \text{IdP} \) computes \( y_b = r_b + e_{\text{User}} \cdot m \) and \( C_{\text{blind}} = y_b \). This is also the process of signing the blind random number \( \text{Nonce} \). The \( \text{IdP} \) does not know the plaintext of \( \text{Nonce} \) throughout the process.

(vi) Reputation Update Confirmation: the \( \text{User} \) removes the blindness of \( C_{\text{blind}} \) (i.e., \( y_b \)) through computing \( y_b^* = y_b + t_b \). Then, the pair \((e_{\text{User}}^*, y_b^*)\) is obtained, which is the signature of \( \text{Nonce} \) denoted by \( C_{\text{blind}} \). Finally, the \( \text{User} \) calls \( \text{Confirm\_Update}(C_{\text{Nonce}}, \text{Nonce}) \) to confirm the update of the reputation. The \( \text{IdP} \) determines if the equation \( y_b^* = g^{y_b} M^{t_b} \) holds. If so, the \( \text{IdP} \) then examines the equation \( e_{\text{User}}^* = H(x_b^*, \text{Nonce}) \). When both equations hold, the \( \text{IdP} \) concludes that the signature is valid.

\[ K_1 = \cdot Z_p \]
\[ U = u^k \]
\[ V = v^k \]
\[ W = w^k g^k \]
\[ \theta_{\text{Id}}, \theta_k, \theta_l \leftarrow Z_p \]
\[ B_1 = u \cdot \theta_k \]
\[ B_2 = V \cdot \theta_k \]
\[ B_3 = W \cdot e(\theta_k, \theta_l) g^k \]
\[ \xi_{\text{Id}} = \theta_{\text{Id}} + \sigma \cdot \theta_k \]
\[ \xi_k = \theta_k + \sigma \cdot k \]
\[ \xi_l = \theta_k + \sigma \cdot l \]

Figure 5

message, as described in Figure 8(a). Linkability between the reputation certificate and the \( \text{User} \) identity is then verified, as described in Figure 8(b). If the equations hold, the reputation and reputation certificate are concluded to belong to the \( \text{User} \).

(vii) Finally, we verify that the encrypted identity certificate is from the \( \text{IdP} \) according to the encrypted identity.

We assume that

\[ B_{11} = e(g, XWR) \theta_1 \cdot e(S, W) \theta_k \cdot e(g, W) \theta_{\text{Id}} \cdot e(S, g) \theta_k \cdot e(g, g) \theta_k \theta_1 \cdot e(S, h) \theta_1 \cdot e(g, h) \theta_k \theta_1 \cdot e(S, w) \theta_k \cdot e(g, w) \theta_{\text{Id}} \cdot (e(g, g) \theta_k \theta_1) \cdot (e(S, W) \theta_1) \]

(1)
After verifying the blind signature, the IdP performs the actual reputation updating operation. We introduce the feedback-based reputation calculation method [39] to estimate the User reputation. Here, three factors determine the total value of User trustworthiness, that is, \( r_{\text{new}}, r_{\text{hist}}, r_{\text{user}} \), where \( r_{\text{new}} \) represents the new reputation feedback the CSP provides after the transaction, \( r_{\text{hist}} \) is the history feedback provided by the previously accessed CSP, and \( r_{\text{user}} \) is the overall reputation the User had prior to the current transaction. After accessing the service, the User reputation is calculated by

\[
\hat{r} = (\omega_{\text{new}} \times r_{\text{new}}) + (\omega_{\text{hist}} \times r_{\text{hist}}) + (\omega_{\text{user}} \times r_{\text{user}})
\]

where \( \omega_{\text{new}}, \omega_{\text{hist}}, \omega_{\text{user}} \) denote the new feedback weight, the updated history feedback weight, and the updated overall reputation weight prior to the current transaction, respectively. All the weights have values in \([0,1]\) and satisfy the constraint \( \omega_{\text{new}} + \omega_{\text{hist}} + \omega_{\text{user}} = 1 \). The details have been described in [39].

In the Pseudonym Disclosure stage, the DA and IdP decrypt the User pseudonym and then restore the User Id. The details of the two operations are as follows:

(i) Anonymity Removal: the DA decrypts the secret information \((U, V, W)\) using the secret key \((b, d)\) in the possession of the DA to restore \( g^{id} \); that is, \( g^{id} = W/(U^{b}V^{d}) \).

(ii) Identity Mapping: the IdP maps \( g^{id} \) to the User Id using the internal mapping table that is built at the beginning.

Through the abovementioned operations, we can deliver an identity management service. The User generates a pseudonym using this service and then employs the pseudonym as the identity for accessing cloud services without exposing any real identity information. The User reputation, within a certain range, indicates to what extent the CSP can trust the user, based on the possibility that the User may carry out malicious activities.

5. Model Analysis

In this section, we verify the correctness and provide a detailed security analysis of the proposed model.

**Definition 1.** User Registration correctness is described as follows. If the identity certificate and the reputation certificate can be verified with a probability of 1, then one concludes that the User Registration is correct. This is given by the following:

\[
\Pr\left(\left| \begin{array}{c}
(PK_{\text{id},s}, SK_{\text{id},s}) \leftarrow \text{Setup} \_ \text{IdP} \_ \text{Jsign} (1^k) \\
(C_{\text{id}, \text{rep}, C_{\text{rep}}}) \leftarrow \text{Register} (SK_{\text{id},s}, \text{Id}) \\
\text{CheckReg} \left( PK_{\text{id},s}, C_{\text{id}, \text{rep}, C_{\text{rep}}} \right) = 1 \\
\text{P}_{u} \leftarrow \text{Gen} \_ \text{Pny} (\text{Id}, C_{\text{id}, \text{rep}}) \\
\text{Authenticate} \left( P_{u}, C_{\text{rep}} \right) = 1
\end{array} \right| \right) = 1.
\]

**Theorem 2.** RIM User Registration is correct.

**Proof.** If the above probability is 1, then the output of each operation must be correct, and the identity certificate and the reputation certificate can pass the verification using the public keys in the possession of the IdP. The validity of the identity certificate was verified in [38], and we need only to verify the validity of the reputation certificate. The following formula (3) confirms that \( e(g^{1/(x+rep+q)}, Xg^{rep}Y^q) = e(g, g) \), verifying the reputation certificate is valid. Consider

\[
e(g^{1/(x+rep+q)}, Xg^{rep}Y^q) = e(g^{1/(x+rep+q)}, g^x g^{rep} (g^r)^q) = e(g^{1/(x+rep+q)}, g^{x+rep+q}) = e(g, g) \tag{3}.
\]

**Definition 3.** Identity Authentication correctness is described as follows. The CSP authenticates the User through the User pseudonym \( P_u \). If all the operations return correct outputs with a probability of 1, we conclude that the Identity Authentication is correct. This is given by the following:

\[
\Pr\left(\left| \begin{array}{c}
(PK_{\text{id},s},SK_{\text{id},s}) \leftarrow \text{Setup} \_ \text{IdP} \_ \text{Jsign} (1^k) \\
(PK_{\text{CSP}}, SK_{\text{CSP}}) \leftarrow \text{Setup} \_ \text{CSP} (1^k) \\
\text{P}_{u} \leftarrow \text{Gen} \_ \text{Pny} (\text{Id}, C_{\text{id}, \text{rep}}) \\
\text{Authenticate} \left( P_{u}, C_{\text{rep}} \right) = 1 \\
\text{Update} \left( \text{rep}_{\text{new}}, C_{\text{rep}_{\text{new}}} \right) \leftarrow \text{Grant} (P_{u}) \\
C_{\text{blind}} \leftarrow \text{Blind} (\text{None})
\end{array} \right| \right) = 1.
\]

**Theorem 4.** RIM Identity Authentication is correct.

**Proof.** The CSP must authenticate the User identity, the validity of the identity certificate, the linkability of the identity and identity certificate, the validity of the reputation certificate, and the linkability of the identity and reputation certificate. All of the above have been proven in [25] with the exception of the linkability of the Id and reputation certificate. Therefore, we only verify here the linkability of the Id and reputation certificate. The verification follows the same approach as that of formula (3) but is conducted by the CSP. Verifying the linkability of the Id and reputation certificate follows according to formula (4), which indicates that the User owns the reputation certificate, and formulas (5) and (6), which verify that the User identity is linked with reputation certificate. Consider

\[
T^\sigma g^{\delta Y - \delta q} = \left(g^{tY^q}\right)^{\sigma t} \left(g^{\theta Y - \delta q}\right)^{\alpha t} = g^{\theta Y - \delta q} \tag{4}
\]

\[
B = U^{-\delta t} \beta_1 \sigma t \tag{5}
\]

\[
V^{-\delta t} \beta_2 \sigma t = V^{-\delta t} \beta_2 \sigma t \tag{6}
\]

The Reputation Computation accuracy involves two aspects: (1) the correctness of the operations such as reputation acquisition, reputation feedback, and reputation update; and (2) the fact that the reputation evaluation algorithm can accurately calculate the user degree of trustworthiness. The accuracy of the evaluation algorithm has been verified in [39]. Therefore, in this paper, we only define the correctness of the operations.

**Definition 5.** Reputation Calculation correctness is described as follows. If the reputation-related operations return correct outputs with a probability of 1, one concludes that the Reputation Calculation is correct. This is given by the following:

\[
\Pr\left(\left| \begin{array}{c}
(PK_{\text{id},s}, SK_{\text{id},s}) \leftarrow \text{Setup} \_ \text{IdP} \_ \text{Jsign} (1^k) \\
(PK_{\text{id},s}, SK_{\text{id},s}) \leftarrow \text{Setup} \_ \text{IdP} \_ \text{blind} (1^k) \\
(PK_{\text{CSP}}, SK_{\text{CSP}}) \leftarrow \text{Setup} \_ \text{CSP} (1^k) \\
\left( \text{rep}, C_{\text{rep}} \right) \leftarrow \text{Withdraw} (\text{Id}) \\
\text{CheckReg} \left( PK_{\text{id},s}, C_{\text{id}, \text{rep}, C_{\text{rep}}} \right) = 1 \\
\text{P}_{u} \leftarrow \text{Gen} \_ \text{Pny} (\text{Id}, C_{\text{id}, \text{rep}}) \\
\text{Authenticate} \left( P_{u}, C_{\text{rep}} \right) = 1 \\
\text{Update} \left( \text{rep}_{\text{new}}, C_{\text{rep}_{\text{new}}} \right) \leftarrow \text{Grant} (P_{u}) \\
\text{C}_{\text{blind}} \leftarrow \text{Blind} (\text{None})
\end{array} \right| \right) = 1.
\]

10 Mathematical Problems in Engineering
Theorem 6. RIM Reputation Calculation operations are correct.

Proof. The operation Withdraw(Id) and the subsequent verification steps of the reputation are equivalent to the registration operations. The verification of the reputation feedback is similar to that given in formula (3). Because the correctness of the registration operations has been proven in Theorems 2 and 4, we here prove only the correctness of the blind signature. Formula (7) removes the blindness of the signature, obtaining the intermediate value \( x_b \). If the equation \( e_b^x = H(x_b^s, \text{Nonce}) \) holds, we conclude that the blind signature is correct. Along with Theorems 2 and 4, the proof verifies the correctness of the reputation operations. One has

\[
\begin{align*}
g^{x^s}M^i & = g^{x^s\cdot x_b^s}M^i \cdot g^{x^s\cdot m + u_b}M^{e - d_b} \\
& = x_b^g g^{x^s}g^{u_b}g^{-m(e - d_b)} \\
& = x_b^g g^{x^s}g^{-m(e - d_b)} = x_b^g f \cdot M^{-d_b} \\
& = x_b^s.
\end{align*}
\]

Definition 7. Pseudonym Disclosure correctness is described as follows. The DA decrypts the pseudonym to restore the User Id with a probability of 1. This is given by the following:

\[
\begin{array}{l}
(PK_{id,s}, SK_{id,s}) & \leftarrow \text{Setup IA} \cdot \text{set} (1^\lambda) \\
(PK_{DA}, SK_{DA}) & \leftarrow \text{Setup DA} (1^\lambda) \\
\text{CheckReg} (PK_{id,s}, C_{id}, rep, C_{rep}) & = 1 \\
P_u & \leftarrow \text{GenPry} (Id, C_{id}, rep) \\
\text{Authenticate} (P_u, C_{rep}) & = 1 \\
g^{id} & \leftarrow \text{DeAnonymity} (P_u, SK_{DA}) \\
Id & \leftarrow \text{Map} (g^{id})
\end{array}
\]

\[= 1.\]

Theorem 8. RIM Pseudonym Disclosure is correct.

Proof. Operations conducted prior to the Pseudonym Disclosure process have been proven in Theorems 2, 4, and 6. The correctness of decrypt (U, V, W) has been proven in [36]. Finally, the IdP queries the mapping table to obtain the User Id. Therefore, we conclude that the Pseudonym Disclosure is correct.

5.2. Security of the Model

5.2.1. Security of Reputation. Before discussing the security of reputation, we assume that the private keys of the IdP, CSP, and User have not been leaked. If the private keys were leaked, the RIM would be open to the public. The data of the IdP are stored in a cloud environment, and the cloud service is open to the public. Therefore, an attacker can obtain an Id and reputation pair, thereby conducting a plaintext attack by analyzing them. In this paper, we will not consider a situation where an attacker modifies the reputation through the cloud underlying infrastructure, and the attacks are all chosen as plaintext attacks.

Unforgeability of reputation refers to the fact that an unauthorized user has no way of modifying their own or another’s reputation value. If the attacker (or Adversary; Adv) wishes to modify a reputation value without permission, the Adv must forge a reputation certificate issued by the IdP or a feedback certificate signed by the CSP. Next, we provide a formal definition of reputation unforgeability.

Definition 9. If a probabilistic polynomial-time adversary Adv has the advantage \( \text{Advantage}_{unf}(\lambda) = \Pr[(PrivK_{unf}(\lambda) = 1) - 1/2] \), which is negligible in \( \lambda \), then one concludes that the reputation value is unforgeable. The experiment \( PrivK_{unf}(\lambda) \) is defined as follows:

(i) Adv is given the public parameters \( (p, g, G, G_T, e) \), as described in the Environment Initialization stage. The register random oracle \( O_{reg} \) provides Adv with a User identity, reputation value, and reputation certificate. The feedback random oracle \( O_{CSP} \) provides Adv with reputation feedback and the feedback certificate. The pseudonym random oracle \( O_{id} \) provides Adv with the User pseudonym \( P_u \).

(ii) The Adv generates \( (Id_0, rep_0), (Id_1, rep_1) \) and \( ((P_u)_0, rep_{new_0}), ((P_u)_1, rep_{new_1}) \).

(iii) The challenger randomly selects \( b \) from \( \{0, 1\} \); that is, \( b \leftarrow \{0, 1\} \). Then, the challenger calls \( \text{rep}_b, (C_{rep}) \leftarrow \text{Withdraw}(Id_b) \) and \( (rep_{new}, C_{rep_{new}}) \leftarrow \text{Grant}((P_u)_b) \) to acquire the pair \( (rep_b, C_{rep}) \) and \( (rep_{new}, C_{rep_{new}}) \).

(iv) The Adv gives a guess \( b' \), and if \( b = b' \), then output is 1; otherwise, output is 0.

Now, we reduce this Adv to an adversary \( (Adv)_b \) for the BB signature. Firstly, \( (Adv)_b \) initializes a hash table \( H \) for the simulation of the random oracles \( O_{reg} \) and \( O_{CSP} \) that are employed by Adv. If Adv queries (Id, rep) or \( (P_u, rep_{new}) \), \( (Adv)_b \) would resort to \( H \). If the query has already been on the table, \( (Adv)_b \) would return the corresponding value; if not, \( (Adv)_b \) would sample a \( (Id, rep) \) or \( (P_u, rep_{new}) \) and place it in \( H \) and return it in the end. When Adv queries a reputation signature included in \( (Id, rep) \) or \( (P_u, rep_{new}) \), \( (Adv)_b \) forwards the query to BB-signing oracle \( O_{BB} \) and returns the response. After sufficient queries, Adv challenges the challenger using \( (Id_0, rep_0), (Id_1, rep_1), \) or \( ((P_u)_0, rep_{new_0}), ((P_u)_1, rep_{new_1}) \). \( (Adv)_b \) extracts the pair \( (rep_0, rep_1) \) or \( (rep_{new_0}, rep_{new_1}) \) and challenges the BB-signing challenger. The BB-signing challenger picks a number \( b \) in \( \{0, 1\} \), \( (Adv)_b \) then returns \( (C_{rep}) \) or \( (C_{rep_{new}}) \) to Adv. Adv makes a guess of \( b' \) and gives it to the BB-signing challenger through \( (Adv)_b \). If \( b = b' \), then not only does Adv have the ability to forge a reputation certificate but also \( (Adv)_b \) can forge a
BB signature. According to the above mentioned method, we reduce \(Adv\) to \((Adv)_{H}\), and we conclude that if the BB signature is secure, then the reputation certificate cannot be forged.

Verifiable Reputation means that the reputation indeed belongs to the authenticated user. If the adversary can forge a pseudonym and the pseudonym can pass verification by the CSP while the DA cannot open the pseudonym or only open it as an unregistered identity, then we conclude that the adversary is successful. This property is guaranteed by Hidden Identity-Based (HIB) Signatures [25].

**Definition 10.** If a probabilistic polynomial-time adversary \(Adv\) has the advantage \(Advantage_{\text{ver}}(\lambda) = \Pr(PrivKe_{\text{ver}}(\lambda) = 1) - 1/2\), which is negligible in \(\lambda\), then one concludes that the reputation value is verifiable. The experiment \(PrivKe_{\text{ver}}(\lambda)\) is defined as follows:

(i) The register random oracle \(O_{\text{reg}}\) provides \(Adv\) with a User identity, reputation value, and reputation certificate, and the pseudonym random oracle \(O_{\text{id}}\) provides \(Adv\) with a User pseudonym \(P_{u}\). \(Adv\) obtains the public parameters \((p, g, G, G_{T}, e)\) as described in the Environment Initialization stage.

(ii) \(Adv\) generates the pair \((Id_{0}, (C_{id})_{0}, rep_{0})\) and \((Id_{1}, (C_{id})_{1}, rep_{1})\) and then gives it to the challenger.

(iii) The challenger randomly selects \(b\) from \(\{0, 1\}\); that is, \(b \leftarrow \{0, 1\}\). Then, he calls \((P_{u})_{b} \leftarrow GenPny(Id_{b}, (C_{id})_{b}, rep_{b})\) to acquire the pseudonym \((P_{u})_{b}\).

(iv) \(Adv\) gives a guess of \(b'\), and if \(b = b'\), \(1 \leftarrow \text{Authenticate}(P_{u})_{b}, (C_{rep})_{b}\) and \(1 \leftarrow \text{DeAnonymity}(P_{u})_{b}, SK_{DA}\) hold, and then output is \(1\); otherwise, output is \(0\).

Now, we assume that an adversary \((Adv)_{H}\) is against the HIB signature. Using three random oracles, namely, the registration random oracle \(O_{\text{reg}}\), the User identity random oracle \(O_{\text{id}}\), the signature oracle \(O_{\text{sign}}\), and the environment initialization stage, \(Adv\) acquires the User identity, identity certificate, and the signature of a message signed by the User. Moreover, these three random oracles can simulate \(O_{\text{reg}}\) and \(O_{\text{id}}\). In [36], the random oracle \(O_{\text{corrupt Oracle}}(\cdot)\) was used for disclosing a pseudonym to perform a cipher text attack. However, in this paper, we assume that the DA is trustworthy. Therefore, we simulate only a plaintext attack. If \(Adv\) can disclose a pseudonym, then \((Adv)_{H}\) must be able to break the HIB signature. Now, we begin the process of reducing \(Adv\) to \((Adv)_{H}\). \((Adv)_{H}\) removes the part relevant to reputation from \((Id_{0}, (C_{id})_{0}, rep_{0})\) and \((Id_{1}, (C_{id})_{1}, rep_{1})\) provided by \(Adv\) and then proceeds with the HIB signature process. During the authentication process, we maintain a constant challenge factor \(\sigma\). The reputation was included when \(\sigma\) was generated. By analyzing the \(\Sigma\)-protocol, we know that \(\sigma\) is randomly selected, so its value does not affect the final result of authentication. Finally, \((Adv)_{H}\) takes \(\sigma\) as a pseudonym \((P_{u})_{b}\) and returns it to \(Adv\). \((P_{u})_{b}\) contains the information regarding a User identity. \(Adv\) gives a guess of \(b'\), and if \(Adv\) can successfully forge a pseudonym, then \((Adv)_{H}\) can crack the HIB signature with the same probability.

Nonrepudiation means that, under any circumstances, the User must admit that the reputation that has been submitted belongs only to the User, and, therefore, regardless of how low the reputation value, the User can never deny ownership. If the adversary has the ability to forge a pseudonym and the DA opens it to find that it corresponds to a legitimate registered user identity, then we conclude that the adversary is successful.

**Definition 11.** If a probabilistic polynomial-time adversary \(Adv\) has the advantage \(Advantage_{\text{rep}}(\lambda) = \Pr(PrivKe_{\text{rep}}(\lambda) = 1) - 1/2\), which is negligible in \(\lambda\), then one concludes that the reputation value has the property of Nonrepudiation. The experiment \(PrivKe_{\text{rep}}(\lambda)\) is defined as follows:

(i) \(Adv\) is given the public parameters \((p, g, G, G_{T}, e)\) as described in the Environment Initialization stage. The register random oracle \(O_{\text{reg}}\) provides \(Adv\) with a User identity, reputation value, and reputation certificate. The pseudonym random oracle \(O_{\text{id}}\) provides \(Adv\) with a User pseudonym \(P_{u}\).

(ii) \(Adv\) generates \((Id_{0}, (C_{id})_{0}, rep_{0})\) and \((Id_{1}, (C_{id})_{1}, rep_{1})\) and then gives it to the challenger.

(iii) The challenger randomly selects \(b\) from \(\{0, 1\}\); that is, \(b \leftarrow \{0, 1\}\). Then, the challenger calls \((P_{u})_{b} \leftarrow GenPny(Id_{b}, (C_{id})_{b}, rep_{b})\) to acquire the pseudonym \((P_{u})_{b}\).

(iv) \(Adv\) gives a guess of \(b'\), and if \(b = b'\), \(1 \leftarrow \text{Authenticate}(P_{u})_{b}, (C_{rep})_{b}\) and \(1 \leftarrow \text{DeAnonymity}(P_{u})_{b}, SK_{DA}\) are true, and then output is \(1\); otherwise, output is \(0\).

The method that reduces \(Adv\) to an adversary for the HIB signature is equivalent to the method used in the analysis of Verifiable Reputation. We therefore omit the reducing process.

5.2.2. Secure Anonymity. In addition to the aforementioned reputation security, it is necessary to consider the security of anonymity. To this end, we assume that the IdP is not safe; namely, its private keys can be leaked. Also, we assume that the IdP and CSP can collude together to compromise anonymity. These assumptions comply with the actual situation for the IdP and CSP may belong to the same institution. However, in the Reputation Calculation process, we assume that private keys in the possession of the IdP cannot be leaked to the User but can be leaked to the CSP because reputation is used to constrain the User, and the IdP and User are antithetical.

Anonymous security includes the unlinkability not only between pseudonyms (denoted as \(P_{u}-\)unlinkability) but also between a pseudonym and a User \(Id\) (denoted as \(Id-P_{u}-\)unlinkability). \(P_{u}-\)unlinkability includes the unlinkability between the pseudonyms of the same user and the unlinkability between the pseudonyms of different users.
With regard to the probability $\text{Pr}[(P_u)_0 = (P_u)_1]$, if the two pseudonyms $(P_u)_0$ and $(P_u)_1$ given in Definition 12 are negligible, then we conclude that $(P_u)_0$ and $(P_u)_1$ are unlinkable.

A number of factors make one pseudonym different from another. Firstly, for different users, their identities are different, and the randomly selected parameter $r$ ensures that the identity certificates are different. Moreover, the parameters used for encrypting the identity and identity certificate are different, making $S, R, U, V,$ and $W$ different. The parameters used for authentication are also different. Finally, the reputation of each user is different in most cases. The above factors constitute the challenge factor $\sigma$ (i.e., $P_u$). If an association exists between pseudonyms, the one-way hash function used to generate pseudonyms will be compromised, although this is impossible. Secondly, for the pseudonyms of the same user, the parameters used to generate $\sigma$, excluding the identities, are different. Therefore, in this case, the pseudonyms are unlinkable.

$Id-P_u$-unlinkability ensures that a user identity cannot be inferred from the user pseudonym. In the process of user authentication, $U, V, W, S,$ and $R$, which are the cipher text of a user identity and identity certificate, generate the pseudonym $P_u$ along with other factors. The unlinkability between $P_u$ and the user identity is guaranteed by the strength of linear encryption.

**Definition 12.** If a probabilistic polynomial-time adversary $Adv$ has the advantage $\text{Advantage}_{Adv}^{\text{Id}}(\lambda) = \text{Pr}[(\text{PrivK}_{\text{IdP}}(\lambda) = 1) - 1/2]$, which is negligible in $\lambda$, then we conclude that a User identity and corresponding pseudonym are un unlinkable. The experiment $\text{PrivK}_{\text{IdP}}(\lambda)$ is defined as follows:

(i) $Adv$ is given the public parameters $(p, g, G, G_T, e)$, as described in the Environment Initialization stage. The register random oracle $O_{\text{reg}}$ provides $Adv$ with a User identity, reputation value, and reputation certificate. The pseudonym random oracle $O_{\text{Id}}$ provides $Adv$ with a User pseudonym $P_u$.

(ii) $Adv$ generates $(Id_0, (C_{id0}, rep_0))$ and $(Id_1, (C_{id1}, rep_1))$ and then gives them to the challenger.

(iii) The challenger randomly selects $b$ from $\{0, 1\}$; that is, $b \leftarrow \{0, 1\}$. Then, the challenger calls $(P_u)_b \leftarrow \text{Gen}_{\text{Pny}}(Id_b, (C_{idb}, rep_b))$ to acquire the pseudonym $(P_u)_b$.

(iv) $Adv$ gives a guess of $b'$, and if $b = b'$ then output is 1; otherwise, output is 0.

Nonrepudiation and Verifiable Reputation are based on the unlinkability between a User identity and pseudonym. Therefore, in the analysis of nonrepudiation and Verifiable Reputation, we have proved that a User identity and pseudonym are unlinkable.

5.2.3. Reputation and Anonymity. The introduction of reputation will influence User anonymity. The CSP can obtain User identity information through reputation in three ways:

1. reputation feedback;
2. reputation submitted by $P_u$; and
3. changes of reputation. Subsequent analysis indicates that all three ways will fail, provided the blind signature is not compromised, greater than 50 users are registered in the IdP, and the CSP is not synchronized with the IdP.

The introduction of the blind signature ensures that the process of updating a reputation will not disclose User identity information. $P_u$ submits the $\text{Nonce blind}$ to the IdP to apply for modification of the reputation value. The User provides the signature of $\text{Nonce}$ to the IdP to confirm the reputation update. The technology of blind signature ensures the unlinkability between both signatures of $\text{Nonce blind}$ and $\text{Nonce}$. Therefore, even collusion between the IdP and CSP cannot reveal User identity information.

When the number of Users registered in the IdP is greater than 50, the CSP cannot locate a User through their reputation value. We assume that the number of users registered in the IdP is $n$ and divide the interval of reputations, say $[0, 1]$, into $N$ independent values from which these $n$ users choose reputations independently (User reputation values are calculated independently of each other, and the reputation values can be considered uniformly distributed on the interval.). The probability distribution of two users with an equivalent reputation is then $P(n) \sim 1 - 1/\exp(n^2/2N)$ (http://en.wikipedia.org/wiki/birthday_problem). As we can see from $P(n)$, when $N = 100$, the probability that two users have equivalent reputation values increases rapidly with increasing $n$. When $n$ approaches 50, the probability resides very close to 1. When $N = 300$, as the number of users approaches 60, the probability continues to reside very close to 1. For simplicity, we take $N = 100$ in this paper. Therefore, when $n$ is greater than 50, a set of users will have equivalent reputations, making it difficult for the CSP to locate a User based upon their reputation. To make the model more general, we assume that $A$ is the collection of users that have the same reputation; that is, $A = \{a_1, a_2, \ldots, a_m\}$ for $1 \leq m \leq n$ (at least one user in the collection provides a reputation). To mine user privacy, the CSP must continuously track users. Assuming the CSP seeks to track user $a_j$ for $j \in \{1, \ldots, m\}$, after the next transaction between $a_j$ and the CSP, the CSP will acquire a set of users $B$ based on the reputation provided by the User through the corresponding pseudonym $(P_{a_j})$. The CSP will obtain the solution $A \cap B = \emptyset, a_j \notin (A \cap B)$, or $a_j \in (A \cap B)$ regardless of whether or not the reputation is identical in these two steps. Therefore, when $n > 50$, the CSP cannot obtain User identity information from the reputation value. The openness of cloud computing ensures that $n$ will be far greater than 50, so that, in practice, the reputation will not disclose User identities.

Through disrupting the synchronization of the IdP and the CSP, the CSP cannot locate a User through an update of the reputation value. For the aforementioned user set $A$, when the CSP gives feedback to a pseudonym, the CSP would monitor the reputation changes in the IdP and then link the pseudonym to a user identity. In this case, the CSP must be synchronized with the IdP. For example, suppose a pseudonym $(P_u)_j$ of user $a_j$ accesses the CSP with reputation...
repj. After that, the CSP grants feedback to (Puj)j to increase the reputation value. Then, the CSP monitors the users in set A whose reputation has been increased, links (Puj)j to aij, and then records the operations of (Puj)j. With a long-term monitor, the CSP will eventually record all the actions of the user and compromise the user's privacy. To solve this problem, the User can choose a random waiting time to confirm the reputation value update so that the synchronization will be disrupted; thus, the CSP cannot determine if the pseudonym that the CSP has just granted belongs to the user whose reputation value has changed.

6. Conclusions

In this paper, we designed and implemented RIM, an identity management model for cloud computing that enables users to access cloud services using pseudonyms so as to ensure the unlinkability not only between different pseudonyms but also between a user and their corresponding pseudonym. In this way, user privacy can be protected. In addition, by calculating the reputation of users, RIM can assist CSPs to identify malicious users. RIM compensates for the shortcomings of identity management introduced by the multitenant feature and openness of the cloud computing environment.

In this paper, we assumed that the CSP honestly provides credible reputation feedback to users. However, in fact, malicious CSPs that provide dishonest assessments of user behavior may exist. In addition, a CSP may violate service level agreements (SLA). The various security vulnerabilities of CSPs pose a threat to users. Therefore, our future goal is to assess the credibility of CSPs and improve the reputation evaluation mechanism so as to provide better protection of user privacy. The access control mechanism in the cloud environment by means of reputation is also a valuable research topic that can be explored in the future.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


Submit your manuscripts at http://www.hindawi.com