PRIAM: Privacy Preserving Identity and Access Management Scheme in Cloud

Jinbo Xiong [1,2], Zhiqiang Yao [1,2], Jianfeng Ma [1,2], Ximeng Liu [2], Qi Li [2], Jun Ma [2]
[1] Faculty of Software, Fujian Normal University
Fuzhou, 350108 - China
[e-mail: jbxiongfj@gmail.com, yzq@fjnu.edu.cn, jfma@mail.xidian.edu.cn]
[2] School of Computer Science and Technology, Xidian University
Xi'an, 710071 - China
[e-mail: snbnix@gmail.com, qilijs@gmail.com, sijunhan@163.com]
*Corresponding author: Jianfeng Ma

Received March 15, 2013; revised November 26, 2013; accepted December 28, 2013; published January 29, 2014

Abstract

Each cloud service has numerous owners and tenants, so it is necessary to construct a privacy preserving identity management and access control mechanism for cloud computing. On one hand, cloud service providers (CSP) depend on tenant's identity information to enforce appropriate access control so that cloud resources are only accessed by the authorized tenants who are willing to pay. On the other hand, tenants wish to protect their personalized service access patterns, identity privacy information and accessing newfangled cloud services by on-demand ways within the scope of their permissions. There are many identity authentication and access control schemes to address these challenges to some degree, however, there are still some limitations. In this paper, we propose a new comprehensive approach, called Privacy pReserving Identity and Access Management scheme, referred to as PRIAM, which is able to satisfy all the desirable security requirements in cloud computing. The main contributions of the proposed PRIAM scheme are threefold. First, it leverages blind signature and hash chain to protect tenant's identity privacy and implement secure mutual authentication. Second, it employs the service-level agreements to provide flexible and on-demand access control for both tenants and cloud services. Third, it makes use of the BAN logic to formally verify the correctness of the proposed protocols. As a result, our proposed PRIAM scheme is suitable to cloud computing thanks to its simplicity, correctness, low overhead, and efficiency.

Keywords: Privacy preserving, identity authentication, on-demand access control, cloud security, service-level agreement

A preliminary version of this paper appeared in ACM ASIACCS cloudcomputing '13, May 7-10, 2013, Hangzhou, China. This version includes an extension of the system model, IAM security requirements analysis, and formal verification the correctness of the PRIAM scheme. This work is supported by the Changjiang Scholars and Innovative Research Team (No.IRT1078), the Key Program of NSFC-Guangdong Union Foundation (No.U1135002), the National Natural Science Foundation of China (No.61370078), and the Major National S&T Program (No.2011ZX03005002). We thank all reviewers for helpful comments.

http://dx.doi.org/10.3837/tiis.2014.01.017

1. Introduction

Cloud computing is a new service delivery paradigm, which provides a huge virtualization service platform for large numbers of tenants by leveraging existing approaches and computing paradigms, such as distributed computing, grid computing, service-oriented architecture, and information system infrastructures consisting of comprehensive pools of computers, mobile terminals, networks, applications and storage resources [1].

Rather than purchasing physical infrastructures and devices, tenants can save a lot of money by leasing these shared resources from CSP(e.g. Microsoft, CloudSafe, IBM, Amazon, Google, Salesforce.com, GoGrid, Box.net, etc.) in the way of pay on-demand. On the other hand, CSP is able to implement economies of scale, workloads balance, and reduce of resource consumption per tenant by sharing the shared infrastructures and devices among those larger number of tenants. It also allows tenants to outsource their own data to cloud datacenter as casually as possible. Furthermore, cloud computing is able to provide flexible, efficient, and great reliability cloud services for remote tenants with on-demand access ways in anytime from anywhere [2].

There are a lot of above advantages for cloud computing, however, without appropriate privacy protection and security solutions, it is difficult to reflect those potential advantages, which may even become a disaster for both tenants and CSP [3]. Several privacy and security challenges in cloud computing demonstrate that the identity and access management (IAM) issues (e.g., issues related to tenants identity and credential management, issues related to tenants and cloud service authentication, issues related to data verification, integrity, and confidentiality etc.) are the primary concerns in cloud computing [1] [4].

IAM consists of identity authentication and access control, which are two prime security services for cloud computing, and have been regarded as two of the top seven cloud security threats [5]. Identity authentication is the basis of access control and provides assurance to system about the entity's (e.g., tenants, services) identity without leaking any privacy. Access control scheme can help system to decide whether to permit the entity to acess a service or not. The distinctive properties of diverse cloud services and multi-tenant generate some new security challenges for identity authentication and access control in cloud computing. Therefore, in the following subsection, we firstly outline the IAM seurity requirements in cloud computing.

1.1 IAM Security Requirements in Cloud Computing

Based on the above analysis, the proposed PRIAM scheme should satisfy the desirable IAM security requirements, which can be highlighted as follows.

(1) Multi-tenant identity privacy: It is a feature unique to cloud computing, where a well tenants identity privacy preserving mechanism is necessary when using a shared infrastructure or service. Any entity in system except the auditor server (AS) has no capability to trace the real identity of the authenticated tenant. Thus, the tenant's identity privacy information needs to be protected throughout the overall accessing processes of cloud service.

(2) Outsourcing data security and mutual authentication: Cloud service provides access to outsourcing data and data security, but the challenge is to essure that only authorized tenants or other entities can obtain access to it [1]. Therefore, messages interacted between the tenants and CSP to gain access permissions need to be authenticated and provided with the protection

of integrity and authenticity. The tenants require to authenticate CSP to acquire the desired cloud services rather than potentially malicious services. Likewise, CSP must authenticate the tenants to provide data security and prevent service abuse.

(3) Service-level agreements: Both the on-demand service and the on-demand access control necessitate the use of well-defined service-level agreements (SLA), which is a service contract defined the level of cloud service between tenants and CSP. SLA is necessary for CSP to control the classification and use of cloud resources in cloud computing. Likewise, tenant also requires SLA to obtain more ideal cloud services, responsibilities, priorities, and warranties.

(4) On-demand access control: Diversity of cloud services and diverse access requirements demand on-demand access control services, which should be flexible enough to capture context, credential- or attribute-based access requirements and compatible with SLA [6]. Tenants require enjoying more high-quality cloud services as they subscribed from CSP based on SLA, their attributes, security levels, and economic capability, etc. Moreover, CSP should satisfy the on-demanded access requirements for the tenants with the corresponding SLA, meanwhile, provide appropriate access control to the tenants.

(5) Unlinkability: None of entities in system except AS is able to link any different sessions or transactions to a particular tenant, or link two different sessions or transactions to the same tenant. It is especially important in cloud computing because of its powerful calculation and reasoning ability.

(6) Accounting: The billing model of cloud services demands appropriate accounting of tenants and service activities. Tenants are able to access desired cloud services by using some denomination tokens, and the system should prevent the double spending or no consumption deduction problem without leaking any privacy information of tenants.

(7) Scalability: The system should allow numerous tenants and CSP to join in and exit voluntarily, and also allow the services to join in or exit without lowering efficiency.

1.2 Our Contributions

Inspired from the recent development of cloud computing and the above security requirements, in this paper, we make a comprehensive consideration about the identity authentication and access control, which is beneficial to the consistency and compatibility for designing protocols and scheme, meanwhile, provides security assurance for identity privacy in cloud computing. And on this basis, we propose a privacy preserving identity and access management (PRIAM) scheme in cloud computing. Our proposed PRIAM scheme is able to satisfy all the seven desirable security requirements for IAM, and is a first comprehensive framework in cloud computing to the best of our knowledge.

In our proposed PRIAM scheme, the registration server (RS) issues only one credential chain to each tenant no matter how many cloud services the tenant attempts to subscribe. After registration, the tenant may purchase some tokens from CSP. Then, the tenant is able to obtain the pre-authorization by passing the mutual authentication phrase between the tenant and a policy decision point (PDP). Hence, a secure channel is established between the tenant and CSP. Finally, PDP grants appropriate permissions to the tenant with SLA according to the relevant attributes extracting from the tenant's access request messages and a token. Upon receiving the permissions from PDP, the tenant can access the cloud services in an on-demand way.

The novelties of our PRIAM scheme stems from the two cryptographic primitives: Blind signature [7] and Hash chain [8]. Blind signature can hide the tenant's identity and attribute

information to protect the tenant's privacy and provide the property of unlinkability. Hash chain can provide a series of credentials to one tenant only need RS to issue once, so the efficiency is greatly improved compared with existing schemes. Furthermore, another novelty of our proposed PRIAM scheme lies in employing SLA to provide on-demanded resource access for the authorized tenants. In addition, our proposed PRIAM scheme also satisfies all the other IAM security requirements aforementioned, such as accounting and scalability. The last highlight of our proposed PRIAM is using the BAN logic [9] to formally verify the correctness of our proposed protocols.

The idea of using blind signature, hash chain, SLA and the BAN logic to construct our proposed PRIAM scheme lies in its concept-simplicity, acceptable key management costs and computation overheads, high efficiency and correctness. Therefore, we believe our proposed PRIAM scheme is appropriate for cloud computing nowadays.

The rest of this paper is organized as follows: In Section 2, we review the previous works on identity authentication and access control in cloud computing. Section 3 presents the system models, assumptions and cryptography primitives. In Section 4, we elaborate the PRIAM scheme, and give the concrete steps of the five protocols in the PRIAM scheme. We use the BAN logic to formally verify the correctness of the PRIAM scheme in Section 5. And in section 6, we present the security-related requirement analysis and performance analysis of the PRIAM scheme. Finally, we conclude the paper in Section 7.

2. Related Works and Limitations

The study of identity privacy and access security is essential and has obtained the great interest from academic community recently. Quite a few literatures have been published to address those challenges in diverse application environments, some of them related to our work are classified into three types and briefly reviewed in this section.

Identity authentication and privacy. Lin et al. [10] leveraged an evidence-token approach to manage identity authentication without a trusted authority and proposed an efficient cooperative authentication scheme to protect vehicle privacy for VANETs. Lu et al. [11] proposed an efficient and privacy-preserving aggregation scheme, and in [12], a dynamic privacy-preserving key management scheme is proposed to protect vehicle user's privacy. Zhu et al. [13] proposed a secure and privacy-preserving authentication scheme to support secure communications and anonymous authentication. However, above schemes are mainly applies to VANETs. In cloud computing, Bertino et al. [3] proposed an identity management system to achieve user authentication to a CSP, but has a risk of disclosing the "master secret key" and the tenant's privacy information. Angin et al. [14] proposed an entity-centric identity management system which supports the management of entities multiple digital identities in cloud computing. Another work about protecting personally identifiable information (PII) was proposed by Chen et al. [15] [16]. A novel framework was presented by using data mining tool (decision tree) to forecast information asset from PII database and determine an appropriate security level for protecting the tenant's PII. Yang et al. [17] combined batch verification with identity-based signature to propose an efficient broadcast authentication scheme. Ranchal et al. [18] used the predicates over encrypted data and multi-party computing to propose an approach for identity management without trusted third party. Li et al. [19] proposed a user authentication and key agreement scheme, which can achieve usr anonymity for hiding login user's real identity and keep the efficiency and security. Recently, Chow et al. [5] proposed the SPICE scheme for digital identity management system that can satisfy unlinkability,

delegatable authentication, and other properties in cloud computing. However, the above methods mainly meet requirements identity authentication and privacy protection. It is far away to say that they are the desirable schemes as they always leave some unsatisfied properties.

Access control in cloud computing. Another research hotspot about access control in cloud computing is using a well known cryptographic primitive, such as multilevel security and attribute based encryption (ABE). Xiong et al. [20] combined the multilevel securiy with action-based access control to propose an action-based multilevel access control scheme in cloud computing. Wang et al. [21] and Liu et al. [22] proposed a hierarchical attribute-based encryption scheme for fine-grained access control in cloud computing. In [23], Liu et al. proposed a weighted attribute based encryption scheme with ciphertext policy to protect data confidential and access security. Another ABE-based access control schemes [24-27], tenants have a set of attributes, and the encrypted data in cloud has associated to access policies. Only tenants with attribute set matching the access policy can decrypt and access the requested data. However, the object which above schemes protected is mainly on data, not cloud services, e.g. resources, infrastructures, platforms, interfaces, etc. Almutairi et al. [28] proposed a novel distributed access control architecture, which combines the principles of software engineering and security management to address security issues in cloud computing. Lu et al. [29] combined attribute-based access control with privacy-preserving scalar product computation to propose an efficient user-centric privacy access control scheme, and in [30], Lin et al. proposed a strong privacy-preserving scheme, but the two schemes are used in eHealth systems. From the above descriptions, existing privacy-preserving access control schemes do not support authentication, and are not sufficient to satisfy all the security properties aforementioned in cloud computing.

Privacy-preserving authentication and access control. Ren et al. [31] proposed a novel privacy preserving authentication and access control scheme, which integrates blind signature and hash chain to establish a mutual authentication, key establishment protocol, and differentiated service access control. However, in [32], Li et al. demonstrated that Ren et al. scheme is vulnerable to service abuse attacks, and proposed a security improvement scheme to enhance the performance of user operational phase. Tan et al. [33] used partially blind signature [34] with a trusted third party to propose a lightweight conditional privacy-preserving authentication and access control scheme, which provides mutual authentication, accountability and differentiated access control. But above schemes mainly suitable for pervasive computing environments. In cloud computing, Ruj et al. [35] used attribute based signature [36] to implement authenticity and identity privacy, and proposed a new privacy preserving authenticated access control scheme, which is decentralized and robust, prevents replay attacks, and has acceptable overheads. However, Ruj et al. scheme is also vulnerable to service abuse attacks, and does not support on-demand access control with SLA in cloud computing.

Up to now, there is no scheme that can satisfy the desirable IAM security requirements in an efficient yet cryptographically secure manner. Motivated by the above works and based on our previous work [37], in PRIAM scheme, we leverage blind signature, hash chain, the SLA and the BAN logic to establish and formally verify the protocols of the registration, token withdrawal, mutual authentication, pre-authorization, and on-demand access control to provide all the security properties for identity privacy and access control in cloud computing. As a result, our proposed PRIAM scheme is suitable to cloud computing because of its simplicity, correctness, low overhead, and efficiency.

3. Models and Primitives

3.1 System Model

A sample system instance of a cloud computing environment is shown in **Fig. 1**. Generally, a cloud computing environment is composed of three types of entities, namely *tenants*, *cloud services*, and *IAM servers* (executing identity and access control policy) respectively, besides, there also have the underlying wired and wireless networks, communications, and system infrastructures. In cloud computing, a tenant may have many different approaches to access various types of cloud services.

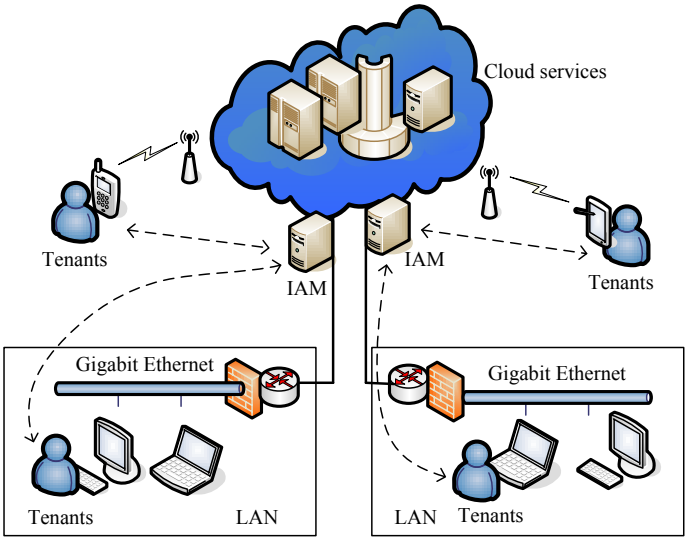


Fig. 1. A sample system of a cloud computing

In our system model, we consider six kinds of entities, namely tenants *U*, cloud service providers *CSP*, registration servers *RS*, auditor servers *AS*, authentication and policy decision point *PDP* with a back-end access control policy repository, and cloud services policy enforcement point *PEP* respectively. *RS*, *AS*, *PDP*, and *PEP* constitute the identity and access management services in cloud computing. **Fig. 2** shows the system model of our proposed PRIAM scheme.

(1) *U* is able to access different authorized cloud services anytime anywhere with some resource-limited terminal devices via wired or wireless networks. After a tenant *U* subscribes a cloud service with an identifier *SID*, *U* accesses the service with *SID* via any nearby *PEP*.

(2) *RS* takes charge of the registrations of all *U* and cloud services. That is to say, if a new type of cloud service in *CSP* requests to join the system, *RS* will add it, otherwise, *RS* will remove it from the system.

(3) *PEP* receives a cloud service request from *U* and forwards it to *PDP*. After receiving a access grant from *PDP*, *PEP* allows *U* to access the required service within access permissions. Otherwise, *PEP* rejects *U*'s access request. *PEP* also acts as a cloud service access point.

(4) *PDP* firstly authenticates *U* in terms of the cloud service request. *PDP* decides whether or not to authorize *U* based on the description of *U*'s attributes along with SLA and issues a final decision to *PEP*.

(5) *CSP* is responsible for providing the authorized *U* with cloud service data. If *CSP* develops a new type of cloud service, it needs to join the system and sends the join request to

RS. Different tenants can access multiple services in the same *CSP* through various distributed *PEP*.

(6) *AS* is an online trusted third party server, which is used for the collections of all session records, including registrations, access requests, authentications, access decisions, authorizations, etc.

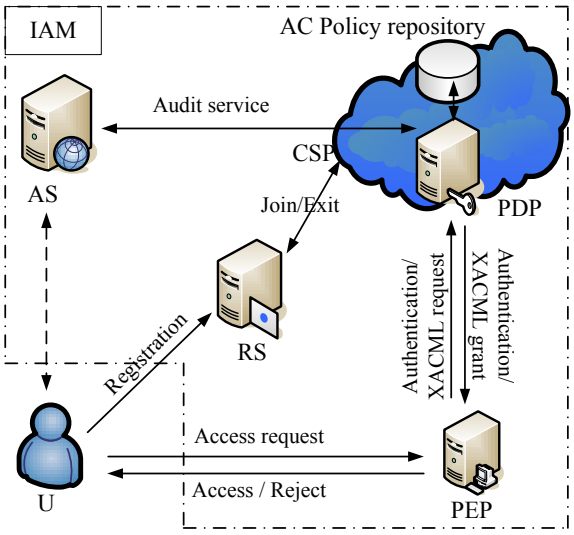


Fig.2. System model of the PRIAM scheme

The main purpose of our proposed PRIAM scheme is to provide tenant's privacy protection and anonymous mutual authentication between *U* and *PDP* or *CSP*. It also provides on-demanded access to cloud services for the tenants in cloud services.

3.2 Security Model

PRIAM scheme just considers honest but curious cloud servers (e.g. *RS*, *AS*, *CSP*, *PDP*, and *PEP*) as the same in [26] does. That is, all the servers will follow PRIAM scheme in general, but attempt to seek more privacy information in terms of their inputs. We assume that these servers are more concerned with tenant's identity and access permission information than others. Furthermore, tenants in cloud computing would try to access cloud services either within or outside the scope of their own permissions, and unauthorized tenants may work independently or cooperatively in order to obtain the access permissions. In our security model, we assume that all the communication channels between *PEP* and the tenants are open and insecure, and all the other communication channels in the system are secure through certain methods (e.g. SSL, or IPsec) [33], these discussions are out of our scope and not included in this paper.

3.3 Cryptographic Primitives

Our proposed PRIAM scheme is based on two cryptographic techniques, namely blind signature and hash chain. A brief review of these two techniques is provided as follows.

3.3.1 Blind Signature

The blind signature scheme is a special digital signature scheme to protect the privacy of actual message and tenant, in which the content of a message is disguised from its signer [7]. Actually, blind signature is an interact protocol between tenant and signer. When a tenant

wants a signer to sign a message m, she first "blinds" the message m to m' by introducing a random "blinding factor" k, then sends the blinded message m' to the signer. The signer signs the blinded message m' and gets the signature s', and then forwards the s' back to the corresponding tenant. Finally, the tenant "unblinds" the signature s' to the signature s, which is the blind signature of the original massage m.

Generally speaking, blind signature schemes can be implemented based on two well known digital signature schemes, the Rivest-Shamir-Adleman (RSA) scheme and discrete logarithm scheme [7]. In addition to all the properties of digital signature, the blind signature scheme also satisfies the other two properties. The one is blind or anonymity, that is the signer did not know the concrete information of the signed message. The other is unlinkability or untraceability, which prevents the signer from linking a blinded message he signed to the unblinded message, the signer also did not know when he signed the message and who he signed to. These can be very important in cloud service applications or environments where privacy and sensitive information need to be protected.

3.3.2 Hash Chain

One-way hash function $H(m)$ is a very useful and computationally efficient cryptographic function, which acts on a message of arbitrary size as its input and outputs a fixed size hash value $h = H(m)$. One-way hash function possesses some properties as follows [38]. ① Given a message m, it is very easy to compute h. ② Given the output h, it is computationally infeasible to derive the original message m. ③ Given a message m, it is very hard to find another message m' satisfying $H(m) = H(m')$.

By applying one-way hash function $H(m)$ repeatedly on an initial message m, we can get a hash chain of outputs $H^i(m)$. The reverse order of these outputs can be used for the purpose of authentication. The hash value $H^{i-1}(m)$ can be proven to be true if $H^i(m)$ has been proven to be true because of the one-wayness property of hash function [31]. Hash chain was first proposed by Lamport [8], which is used in an authentication scheme. Recently, Ren et al. adopted a hash chain technique for efficient privacy preserving authentication in pervasive computing environments [31].

4 Our proposed PRIAM Scheme

In this section, we present our proposed privacy preserving identity authentication and on-demand access control scheme (PRIAM) for cloud computing. For simplicity, we assume that tenants can control the source addresses of the outgoing medium access control frames. This is a prerequisite for general anonymous communication system. Otherwise, the tenant can be identified by the MAC address. Refer to He et al. [39] for detailed techniques which are out of scope of this paper [33]. The notations and their descriptions to be used in our scheme are shown in **Table 1**.

Our proposed PRIAM scheme is composed of five phases: registration phase, token withdrawal phase, tenant pre-authorization phase, on-demand access control phase, and token spending phase. The descriptions of each phase are discussed in the following subsections.

4.1 Registration Phase

A tenant U firstly registers at a nearby RS when U wants to login the system. After finishing

the registration, U will obtain an authorized credential from RS. The authorized credential, then, is able to be used in the following token withdrawal phase and mutual authentication phase between U and PDP when U attempts to access a desired cloud service.

Table 1. Notations and descriptions

Notations	Descriptions
$PriK_A$	Private key of an entity A
$PubK_A$	Public key of an entity A
K	Symmetric key
$Sign_{PriK_A}(m)$	Signing message m using private key $PriK_A$
$H(m)$	Secure one-way hash function on m
M_i	The ith receipt from CSP
t_i	Temporary identity
$Enc_{PubK_A}(m)$	Encrypt message m using public key $PubK_A$
$Dec_{PriK_A}(m)$	Decrypt message m using private key $PriK_A$
$En_K(m)$	Encrypt message m using symmetric key K
$Cert(U)$	The certificate of U's public key
⟶	Transmission through an insecure channel
⟹	Transmission by a secure channel

The registration phase is shown in **Fig. 3**.

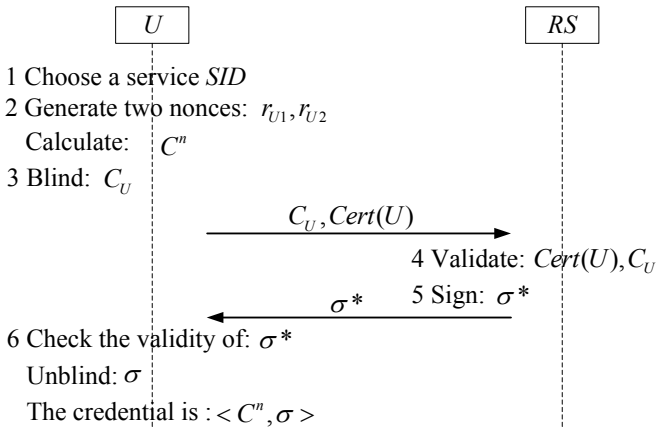


Fig. 3. Registration phase

This phase is described as follows:

Step R1. A tenant U firstly chooses a desired cloud service SID, and signs it with one's private key $PriK_U$ to get $S = Sign_{PriK_U}(SID)$.

Step R2. U randomly generates two fresh nonces r_{U1} and r_{U2}. Then, U calculates the credential chain $C^n = H^n(S, U, r_{U2})$.

Step R3. U blinds C^n by using r_{U1} to calculate $C_U = C^n * Enc_{PubK_{RS}}(r_{U1})$, then sends C_U and a certificate $Cert(U)$ to RS.

Step R4. *RS* validates $Cert(U)$ and confirms whether C_U is correct, which is able to prevent the message-substitution attack of malicious attackers.

Step R5. *RS* signs C_U as $\sigma^* = Sign(C_U)_{PriK_{RS}}$ and sends σ^* to *U*.

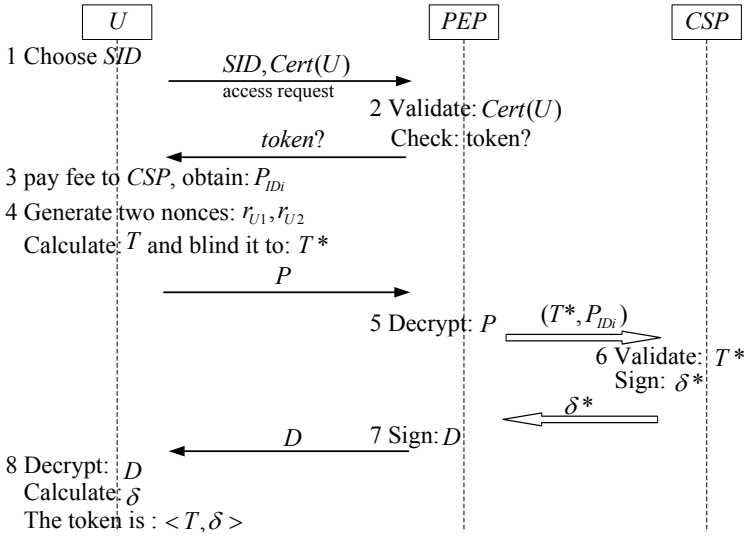
Step R6. *U* checks σ^*, unblinds σ^* to $\sigma = r_{U1}^{-1} * \sigma^* = Sign_{PriK_{RS}}(C^n)$, and confirms the authenticity of σ by verifying C^n is or not equate to $Dec_{PubK_{RS}}(\sigma)$. If equal, *U* obtains a credential $< C^n, \sigma >$.

Note that, the tenant identity *U* in C_U is not revealed to *RS* because it is protected by the fresh nonce r_{U1} in blind signature. Our proposed PRIAM scheme also allows role registration where a role represents a group of tenants with some similar properties, then, *RS* issues role credential to all corresponding tenants. It will reduce management cost and enhance efficient greatly.

After registration, the tenant requires pay fee to pre-purchase some tokens for many different kinds of cloud services from *CSP*, tenants are able to access cloud services using token after mutual authentication between the tenants and *PDP*. Next, we describe the token withdrawal phase.

4.2 Token Withdrawal Phase

A tenant needs to pre-purchase some tokens from *CSP* via *PEP* to access the desired cloud services. Each token in our scheme contains a payment identity P_{IDi} with λ-bit integer and the *CSP*'s signature on it, where λ is a system parameter. Tokens withdrawal phase is in the following way as shown in **Fig. 4**.



This phase is described as follows:

Step T1. A tenant *U* wants to access a service *SID*, and sends an access request containing *SID* and $Cert(U)$ to *PEP*.

Step T2. *PEP* validates $Cert(U)$ and checks *U*'s token. If *U* has not submitted a token, then, *PEP* will give a reply to *U* and ask *U* to submit a token.

Step T3. *U* pays fee to *CSP* via *PEP* to purchase the corresponding service. Then, *CSP* gives

P_{IDi} to U.

Step T4. U randomly generates two fresh nonces r_{U1} and r_{U2}, calculates hash value $T = H(P_{IDi}, U, r_{U1})$, and blinds it as $T^* = T * Enc_{PubK_{SID}}(r_{U2})$. Finally, U calculates $P = Enc_{PubK_{PEP}}(T^*, P_{IDi})$ and sends P to PEP.

Step T5. PEP obtains P and decrypts it with $PriK_{PEP}$ and sends (T^*, P_{IDi}) to CSP.

Step T6. CSP receives (T^*, P_{IDi}), validates T^*, and signs it as $\delta^* = Enc_{PriK_{SID}}(T^*)$ and forwards δ^* to PEP.

Step T7. PEP obtains δ^*, signs it as $D = Enc_{PriK_{PEP}}(\delta^*)$ and sends it to U.

Step T8. U receives D, decrypts it to δ^*, and calculates $\delta = r_{U2}^{-1} * \delta^* = Enc_{PriK_{SID}}(T)$, which is CSP's signature on T. Finally, U records $< T, \delta >$ as a token.

As a result of the blinding factor r_{U2}, PEP and CSP cannot deduce T and δ from δ^*. That is to say, given a token $< T, \delta >$, PEP and CSP doesn't link it to U. Each token corresponds to a monetary value and can be used to purchase a certain amount of service data. We can get multi-denomination tokens if CSP uses different public and private key pairs for each kind of denomination. For simplicity, we focus mainly on single-denomination tokens in this paper.

4.3 Tenant Pre-authorization Phase

The target of this phase is to establish both a mutual authentication and a secure channel between the tenant and PDP or CSP by negotiating a session key based on the credentials and tokens. A tenant uses this session key to access authorized service SID in cloud computing without revealing any information about identity, except for the type of the subscribed service. Conceptually, the tenant pre-authorization phase works as follows.

A tenant U first sends an access request to PEP, which contains a service type SID with accessing capability claim, an authorized credential from RS, and a token corresponding to SID. PEP transforms this message into XACML request format and then forwards this XACML request message to PDP. PDP decrypts and validates the authenticity of the credential and the token when it receives the forwarded XACML request message. If the results hold, PDP ascertains that U indeed has the capability to access the claimed service SID although it doesn't know who the tenant is. PDP then, replies to PEP with an XACML acknowledgement with accept message containing its signature used to mutual authentication. We know that there is a secure channel between PEP and PDP from the security models aforementioned, so that PEP is able to securely obtain the reply secret messages from PDP. Note that these processes are transparent to U. Upon receiving the secret message from PDP, PEP randomly generates a fresh nonce and computes two fresh session keys, one for encryption key, the other for integrity key. Then, PEP encrypts the secret message with the encryption key and finally replies to U with the access acknowledgement. And now, U computes two session keys in the same manner and decrypts the access acknowledgement message from PEP with the shared encryption key, and then, U authenticates PDP. If the result holds, U ascertains that PDP is valid and the service is legal. These processes constitute the mutual authentication phase between U and PDP, and now both parties share the same two fresh session keys, and the secure channel between U and CSP has been established.

The concrete steps of the tenant pre-authorization protocol are shown in **Fig. 5**.

Step A1. We firstly introduce a receipt M_i to prevent abuse attack [32]. For M_i, U needs to

be paid using the *token* for the required services from *CSP*, and *CSP* will release a unique receipt number M_i to the paid tenant *U*, and *U* must keep M_i secret and randomly generates a fresh nonce r_U.

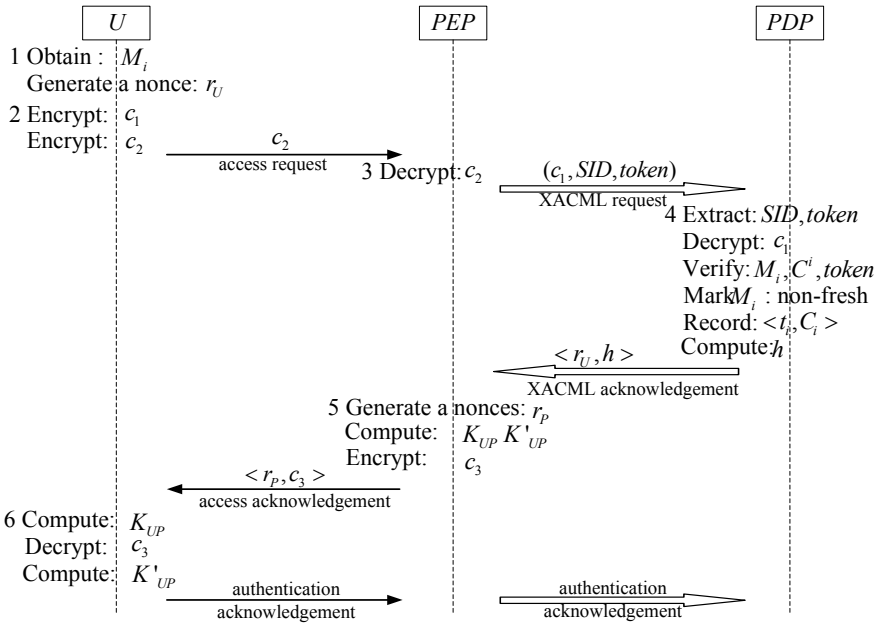


Fig. 5. Pre-authorization protocol

Step A2. *U* encrypts $.., t_i, r_U, C^i$ and C^n to $c_1 = Enc_{PubK_{PDP}}(M_i, t_i, r_U, C^i, C^n)$, that is used for the purpose of authentication. Meanwhile, *U* keeps the secret of C^i, r_U, and M_i. When the authorized credential chain is used for the first time, the process was described in [31], where t_i is a temporary identity used to enhance the efficiency of the tenant pre-authorization phase. Then, *U* encrypts c_1, *SID*, and *token* to $c_2 = Enc_{PubK_{PEP}}(c_1, SID, token)$, and forwards c_2 to a nearby *PEP*.

Step A3. *PEP* decrypts c_2 and transforms c_1, *SID*, and *token* to XACML request, and sends it to *PDP* through a secure channel, which we assumed in our system security models.

Step A4. *PDP* receives the XACML request, extracts *SID* and *token*, decrypts c_1, and verifies the validity of M_i, C^i, and *token*. If the verifying is passed, *PDP* marks M_i non-fresh, records the tuple $< t_i, C^i >$, which will be used in next round for reducing search time. Furthermore, *PDP* calculates the hash value $h = H(r_U, C^i, C^n)$. Finally, *PDP* forwards $< r_U, h >$ to *PEP* through the secure channel.

Step A5. *PEP* randomly generates a fresh nonce r_P after obtaining $< r_U, h >$, and calculates two session keys $K_{UP} = H(h, r_P, r_U, 0)$ and $K'_{UP} = H(h, r_P, r_U, 1)$, which are used for encryption and integrity verification respectively. And then, *PEP* encrypts h and r_P with K_{UP} to $c_3 = En_{K_{UP}}(h, r_P)$, and sends $< r_P, c_3 >$ to *U*.

Step A6. Upon obtaining $< r_P, c_3 >$, *U* gets r_P and calculates $K_{UP} = H(h, r_P, r_U, 0)$, then decrypts c_3 with K_{UP}. Next, *U* takes apart the plaintext of c_3 into h' and r_P', and checks whether $h' = H(r_U, C^i, C^n)$ and $r_P' = r_P$. If both results are correct, *U* confirms that *PDP* is legal,

and that *SID* is credible. Finally, *U* calculatetes the integrity verification session key $K'_{UP} = H(h, r_P, r_U, 1)$ and forwards a mutual authentication acknowledgement to *PEP*. Therefore, *U* has two shared session keys, which are the same as in *PEP*.

Step A7. *PEP* receives the acknowledgement message and simply forwards to *PDP*. *PDP* validates this message, if the result is accept, this is concluded the mutual authentication.

Our proposed PRIAM scheme is able to resist against abuse attack. As described in **Fig. 5**, an adversary does not have a valid M_i and correct c_1, thus, the adversary cannot pass *PDP*'s verification of M_i, and also cannot get the rightful c_3. Furthermore, each M_i is permitted to use only once and it is encrypted in c_1. Only *PDP* can decrypt c_1 and after that, M_i would be marked as non-fresh value and can no longer be used. Therefore, we can resist against abuse attack as the same as it was described in [32].

A secure channel between *U* and *CSP* is established by finishing the mutual authentication processes. Now *U* obtains pre-authorization, but still can not obtain the real access permissions. Next, we describe the phase of granting access permissions.

4.4 On-Demand Access Phase

The target of this phase is that a tenant can access the desirable services on-demand with the valid token. *CSP* provides different services about the same *SID* to the tenants, and grants the tenants appropriate access permissions in terms of their attributes, tokens, and authorized credentials. Furthermore, the tenants may subscribe the services in an on-demand way according to their financial resources and application within their scope of authority.

Our proposed PRIAM scheme relies on SLA to implement on-demand access control. SLA implements attribute mapping, isolation constraints for cloud services and resource sharing to resist against side-channel attacks [28]. Attribute mapping is a function that each attribute set is maped to some recommended cloud services for tenants to select. Meanwhile, SLA provides a virtualized view of cloud services at the levels for which SLA is negotiated. Besides the advantages mentioned above, SLA usually contains quality-of-service (QoS) parameters and auditing functions to enhance the effective and efficient of our proposed PRIAM scheme. The on-demand access authorization processes are shown in **Fig. 6**.

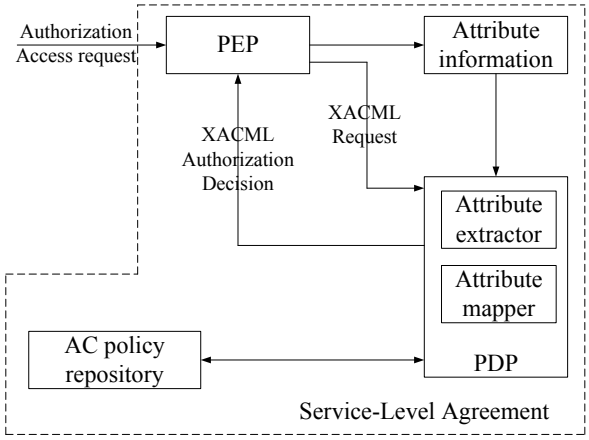


Fig. 6. SLA-oriented on-demand access control

Step O1. *PEP* receives an access request from a tenant, transforms it into XACML request, and sends the XACML request along with attribute information to *PDP*.

Step O2. Once the request is received, PDP uses the attribute extractor to extract attribute

set $Attribute_i$, and uses the attribute mapper to map $Attribute_i$ to cloud services with suitable service-level.

Step O3. *PDP* checks whether $Attribute_i$ and security level $SecLev_i$ [40] along with the token remaining balance $Token_i$ matching SLA and access control policies. Our proposed PRIAM scheme adopts attribute based access control [27] mechanism, and the access control policies are stored in the policy repository in *PDP*. If matching result is correct, *PDP* assigns corresponding access permissions with the "least privilege principle", tranforms it into XACML authorization decision and returns this decision back to *PEP*.

Step O4. *PEP* encrypts this decision with the shared encryption key and sends it to the tenant through a insecure channel. After decrypting the ciphertext and obtaining the access permissions, the tenant can safely enjoy the authorized cloud service *SID* in a pay-buy-the-hour or an on-demand way in cloud computing.

4.5 Tokens Spend Phase

The token spend phase is comparative simple. In step A4, *PDP* checks a *token* $<T,\delta>$ after receiving the access request message, the check must be successful for a authentic *token* $<T,\delta>$ due to $\delta = Enc_{PriK_{SID}}(T)$, *PDP* is able to validate δ using the public key of *SID* and confirm P_{IDi} because it was issued by *CSP*. Furthermore, *PDP* can also know the remaining balance of the token by checking the receipt *M* formed the "eligible paid tenants list". If both the validate results are correct, *PDP* will use the token to grant the access permissions to the tenant *U* in the on-demand access phase. *U* can access appropriate amount of requested cloud services in an on-demand way or pay-buy-the-hour model depending on the remaining balance of the token. *PDP* doesn't get any useful information about the tenant's identity form T because of the capability of blind signature. Meanwhile, *PDP* cannot link to *U* and doesn't know who requested the cloud service. Therefore, the tenant's identity privacy is well protected.

5 Correctness Verification of Our PRIAM Scheme

In this section, we formally verify the correctness of our pre-authorization protocal in our proposed PRIAM scheme based on the BAN logic [9]. BAN logic is a formal logic based on belief, which is widely used to formally reason about authentication and secueity protocols. The correctness of our protocol means that, after executing the tenant pre-authorization protocol, both *U* and *PEP* not only ascertain that they own the same fresh session key, but also assure that they achieve the same belief.

In our pre-authorization protocol, there is a secure communication channel between *PEP* and *PDP* that is assumed in our system security model, thus, both *PEP* and *PDP* trust the integrity and authenticity of the messages exchanged between them. Therefore, for simplicity, we transfer the pre-authorization protocal into the following generic representation as shown in **Fig. 7**, and further idealize in the same Figure. We omit all the unnecessary steps and pay close attention to the messages exchanged between *U* and *PEP*, then, verify whether both of them can assure that they share the same fresh session key K_{UP}.

The main notations we used are shown in **Table 1**, and other notations follow those presented in BAN logic [9]. In order to verify the pre-authorization protocal, we first make some assumptions shown in **Fig. 8**. Specially, in **Fig. 8**, notation "‘ ” means belief, “ ♯”

means generation.

Next, we would like to interpret the meaning of these assumptions in **Fig. 8**. (1) and (2) indicate that both U and PEP believe PEP owns a public key $PubK_{PEP}$. (3) and (4) state that U and PEP generate two fresh nunces r_U and r_P, so, assure their freshness. (5)-(8) are related to the authorized credentials shared between U and PEP.

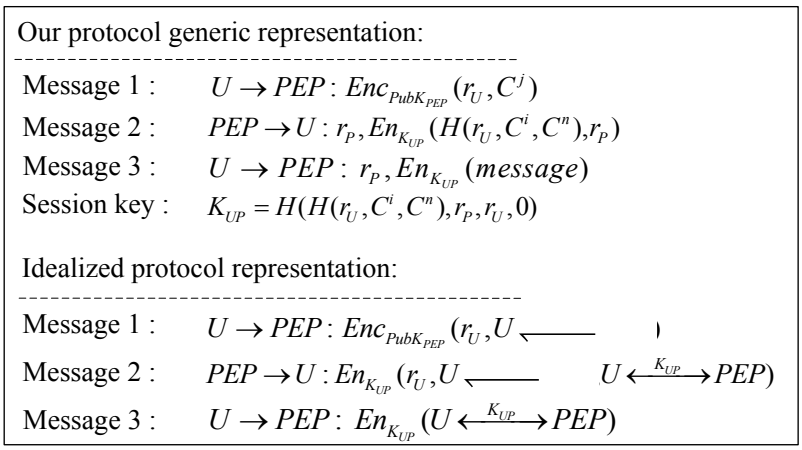


Fig. 7. Generic and idealized representation of the pre-authorization protocal

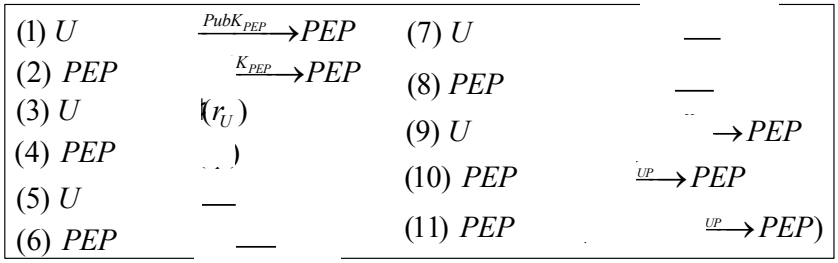


Fig. 8. Assumptions of formal verification

When j=n, the authentication server PDP can easily verify the integrity and authenticity of C^n though the attached signature, therefore, it can believe that C^n is the secret shared between U and itself although has no information about who U is. When $j < n$, PDP keeps the same belief due to the one-wayness property of the hash chain. In Step A4, the M_i C^i, and $token$ values, after verification, are stored in PDP. Then, M_i is marked non-fresh to prevent abuse attack, so, it can ensure that C^i is freshness. Furthermore, each C^j $(j < n)$ is freshness because of each of which used only once. The formal verification on above beliefs about the hash chain could be found in [41]. Assumption (9) means that U believes PEP can control right K_{UP} because K_{UP} depends on a nonce r_P generated by PEP, another nonce r_U and share secret C^i sent by itself. (10) and (11) hold because PEP constructs the fresh session key K_{UP} with a shared secret between U and PEP. The detailed verification process is shown in **Fig. 9**, and the interpretation of each verification step is omitted here because of space limitaions. Equations (16), (17), (19), and (24) together accomplish the verification process.

6. Analysis of our PRIAM Scheme

6.1 Security Requirement analysis of our PRIAM

Our proposed PRIAM scheme possesses many outstanding security-related properties as analyzed below.

6.1.1 Multi-tenant identity management

According to the property of the blind signature, the registration phase and the token withdrawal phase, our PRIAM is able to well protect *U*'s identity privacy information. *PDP* or *CSP* only obtains *U*'s service type *SID* to grant suitable access permissions to *U*, who is able to achieve anonymous authentication by blind signature without disclosing any other identity and context information. Moreover, all the accessing service data traffic transmitted between *U* and *CSP* is protected by the shared session key, so these data containing *U*'s privacy information is also well protected against any other unauthorized part.

In our proposed PRIAM scheme, *U* only needs to register once for obtaining the credentials without registration every time when access request is needed. Furthermore, our proposed PRIAM scheme is able to adopt Chen et al. scheme [16] to implement secure interoperation among different *RS* located in different circles of trust. This SSO and interoperation function can effectively improve the efficiency and flexibility of our identity management.

Verification process:

Message 1 : $U \rightarrow PEP : Enc_{\cdot\cdot\cdot}(r_U, U \longleftarrow \quad)$

(12) $PEP \triangleleft$ ⟶ *///(2), seeing rule*

Message 2 : $PEP \rightarrow U : En_{K_{\cdot\cdot\cdot}}(r_U, U \longleftarrow \quad U \xleftrightarrow{K_{UP}} PEP)$

(13) $U \triangleleft$ $r_U, U \longleftarrow \quad U \xleftrightarrow{K_{UP}} PEP)$

(14) U $\longleftarrow \quad U \xleftrightarrow{K_{UP}} PEP)$ *///(5),(13), Message-meaning rule*

(15) U $\longleftarrow \quad U \xleftrightarrow{K_{UP}} PEP)$ *///(3),(14), Nonce-verification rule and Freshness rule*

(16) U $\xrightarrow{UP} PEP$

(17) U $\xrightarrow{P} PEP$

Message 3 : $U \rightarrow PEP : En_{K_{UP}}(U \xleftrightarrow{K_{UP}} PEP)$

(18) $PEP \triangleleft$ $I \xleftrightarrow{K_{UP}} PEP)$

(19) PEP $\xrightarrow{P} PEP$ *///(10)*

(20) PEP $\xrightarrow{K_{UP}} PEP)$ *///(11)*

(21) PEP $\xrightarrow{UP} PEP)$ *///(18),(19), Message-meaning rule*

(22) PEP $\xrightarrow{K_{UP}} PEP$ *///(20),(21), Nonce-verification rule*

Fig. 9. Formal verification of the pre-authorization protocol

6.1.2 Outsoursing data security and mutual authentication

Our proposed PRIAM scheme is able to achieve mutual authentication between *U* and *PDP* to implement outsourcing data security. Firstly, *U*'s credential is acquired from a blind signature of *RS*, which is a trusted server, so *PDP* can verify the validity of *U*'s credential and know that *U* is indeed legal and authorized. However, *CSP* cannot obtain any useful information about *U*'s identity even if gets the access request from the authorized *U*.

Furthermore, as shown in Fig.5, in step A1, U encrypts one's certificate C^i and a fresh nonce r_U with the public key of *PDP* for authentication. Finally, in step A6, U obtains and decrypts c_3 which the corresponding plaintext should contain a hash value of C^i and the fresh nonce r_U. Therefore, U can verify the identity of *PDP* through confirm C^i and r_U. Our system security model assumes that $Enc(x)$ function is an IND-CPA secure and $H(x)$ is a collision-resistant hash function, so we can make sure that U can authenticate *PDP* successfully.

Our proposed PRIAM scheme is also able to achieve authenticated key agreement between U and *PEP*, with implicit key authentication.

6.1.3 Service-level Agreements

Our proposed PRIAM scheme supports SLA to implement effective and efficient service level management and on-demand access control. If the meaning and scope of service types are carefully defined, cloud services are able to be well classified into many kinds of levels in cloud services. Each service level requires a corresponding attribute set of U. The attribute extractor in *PDP* can extract attributes from the access request message, and then, the attribute mapper maps these attributes to appropriate cloud service level for U to select.

6.1.4 On-demand Access Control

Our proposed PRIAM scheme is able to implement on-demand access control in the on-demand access phase. The authorized credentials of different service types are signed by different private keys, which supports fine-grained access control. Meanwhile, the attribute mapper is able to well map attribute set to a scope of service levels. The combinational usage of several authorized credentials at the same time and SLA can further improve the ability to enable higher level on-demand service access control [6]. This will also improve the flexibility and scalability of our proposed PRIAM scheme.

6.1.5 Unlinkability

As can be seem from the literature [31] [42], unlinkability means that for both insiders and outsiders, neither of them could link any session to a specific tenant, and neither of them could link two different sessions to the same tenant. Our proposed PRIAM scheme is able to achieve unlinkability based on the following reasons.

On one hand, for the insiders, *PDP* or *CSP* is able to link up n sessions in the same credential chain to the same tenant because of using the hash chain, where n is the length of the hash chain. Furthermore, our scheme also adopts the blind signature to prevent *PDP* or *CSP* linking to a particular tenant because of it protected by the random fresh nonces. Moreover, there is no relationship among different hash chains and credential chains so that *PDP* or *CSP* could not aggregate any useful information from the inter-hash chains or inter-credential chains to link to a particular tenant.

On the other hand, for the outsiders, the tenant's authorized credential is always combined with a random fresh nonce and transmitted in ciphertext form. Thus any outsiders cannot link a session to a particular tenant, and no one could be associate with two or more sessions to the same tenant. Therefore, the unlinkability can achieve in our proposed PRIAM scheme.

6.1.6 Data Transmission Protection

The tenant pre-authorization phase in our proposed PRIAM, in step A5, two fresh session keys

are choosed for protecting the transmission data traffics of cloud services between *U* and *CSP*. Therefore, the integrity and confidentiality of service data can be guaranteed based on the encrpytion tachniques.

6.1.7 Accounting

In our proposed PRIAM scheme, the tenant's credential is authorized by *RS* only when *U* is correct register and *RS* validates the certificate of *U*. Both the unique receipt M_i and one-time usage property of the authorized credentials stop the double spending of *token*s and provide a well accounting capability. *PDP* saves an "eligible paid tenants list" and records the pair of paid tenants and their corresponding receipts after *U* correctly paid for the requested cloud services by using the *token*, it also help to prevent the double spending of the *token* and further provided accountability.

6.1.8 Scalability

In the PRIAM scheme, there are lots of *PDPs*, *PEPs*, and many kinds of *CSPs* distributed in our system. The tenants may choose a nearby *PEP* to send the access request messages. Our system allows many *PEPs* or *PDPs* to join in or exit. Compared with the existing schemes, one tenant in the PRIAM can acquire credentials of different cloud service types from different *CSPs* through one *PDP* once, which is according to the actual situation in cloud services. It is very useful and convenient for the tenants. Even *CSP* also can join in or exit the system freely in the PRIAM scheme.

6.1.9 Formally Verification

In section 5, we formally verified the correctness of our proposed PRIAM scheme by using the BAN logic. Therefore, PRIAM scheme is secure, and is able to resist against active and passive attacks in cryptosystem.

Compared with existing similar schemes in the literatures, our proposed PRIAM satisfies more properties and desirable requirements. **Table 2** demonstrates the comparison results in terms of the security-related requirements aforementioned. The advantages of our proposed scheme are shown clearly.

Table 2. Security requirement features comparisons

Security properties	PRIAM	[35]	[5]	[33]	[16]	[31]
Multi-tenant identity privacy	√	√	√	Partial Yes	√	√
Mutual authentication	√	√	Delegation	√	×	√
Service-level agreement	√	×	×	×	×	×
On-demand access control	√	Distributed	×	×	×	×
Tenant privacy preserving	√	√	√	√	√	√
Unlinkability	√	×	√	√	×	√
Outsoursing data security	√	√	√	√	√	√
Accounting	√	√	√	√	×	√
Scalability	√	√	√	√	√	√
Formally verification	√	×	×	×	×	√

Note: The basic scheme in [33] is considered in the above table, " √ " means satisfying the property.

6.2 Performance Analysis of the PRIAM

6.2.1 Key Management Overhead

The key management cost in our proposed PRIAM scheme is optimistic. *RS* needs to manage one certificate per tenant and the corresponding tenant's attributes. Our scheme is compatibility with role, and *RS* has the delegation property. We assume that one certificate is issued for a role represented a group of tenants, thus the number of certificates can be significantly reduced. Each time a tenant attempts to purchase a token and access a cloud service, the tenant should only know and manage the public key of *SID*, *PEP* and *PDP*. But in [33], it does not need any public key of *SID* and thus it does not provide any on-demand access service for every kind of services.

6.2.2 Storage Overhead

In our proposed PRIAM scheme, *AS* only needs to store very little key information, it is usually offline and works only when a dispute happens, and is similar to the *TTP* in [33]. *RS* is responsible for the tenant registration phase. Neither *AS* nor *RS* stores privacy information about the tenants. *PDP* stores two values $< C^i, C^n >$ for each active credential chain and one C^i for each used but unexpired chain, which is the same overhead as *SP* in [31]. *PEP* doesn't require to store long-term tenant information, only maintains two session keys in each session. *U* should maintain three fresh nonces for registration, purchase token, and pre-authorization phase, and two session keys in each ongoing session. Our scheme employs hash chain to obtain credential chain which can get a tradeoff of the computation and storage cost.

6.2.3 Communication Overhead

Communication overhead is mainly produced in the tenant pre-authorization phase, where there are two rounds of interactions to achieve the mutual authentication and establish the shared session key between *U* and *PDP*. The communication overhead of our scheme is equivalent to existing schemes [31] [33], which also requires two rounds to achieve our goal. Note that two rounds is the minimal number to establish mutual authentication. The requirements analysis discussion shows our scheme satisfies more security-related requirements, meanwhile, it does not bring more overheads. Therefore, our proposed PRIAM scheme is extremely lightweight and more efficient in the sense of running overhead.

6.2.4 Computation Overhead

Computation overhead is also mainly produced in the tenant pre-authorization phase, where the tenant operates two public key encryption and one symmetric encryption per session in the mutual authentication phase, all remaining operations are three nonces and two hash values. *PEP* needs to calculate one public key decryption, one nonce, one symmetric cryptographic, and two hash values. *PDP* requires to perform one public key decryption and one hash operation. The tenant pre-authorization phase requires no additional exponential computation. We compare the computation overheads of our proposed PRIAM scheme with existing schemes [31] [33] in **Table 3**.

Table 3. A comparison of the computation overheads

	Entity	Pub. Key	Sig. Verify.	Nonce Gen.	Hash Oper.	Symmet. Key	Expon.
PRIAM	U	2	0	3	2	1	0
	PEP	0	0	1	2	1	0

	PDP	1	1	0	1	0	0
[33]	U	1	0	1	3	1	0
	AP	0	0	0	2	1	0
	SP	1	1	1	1	0	0
[31]	U	1	0	1	2	3	2
	AP	0	0	1	2	3	0
	SP	1	1/n	0	0	0	1

Note: The basic scheme in [33] is considered in the above table.

7. Conclusion

Identity privacy and access control are regarded as two of the top seven security challenges in cloud computing. In this paper, we proposed a privacy preserving identity and access management (PRIAM) scheme for the cloud computing. Our proposed PRIAM scheme combines the blind signature with hash chain technology to establish the tenant's credentials, tokens, and pre-authorization, leverages the service level agreement to implement on-demand access control for both the tenants and CSPs, and employs the BAN logic to formally verify the correctness of the proposed protocols.

The outstanding advantages of our proposed PRIAM scheme are as follows: ① simplicity of concepts, operations, and implementations, ② satisfying all the desirable security-related requirements, and ③ acceptable overall overheads. Therefore, we believe that our proposed PRIAM scheme is well suitable for cloud computing. As part of future work, we would design more efficient pre-authorization protocol, and combine multi-level security to construct fine-grained on-demand access control model in cloud computing.

References

[1] H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments," *Security & Privacy, IEEE,* vol. 8, pp.24-31, 2010. Article (CrossRef Link)

[2] M. Van Dijk and A. Juels, "On the impossibility of cryptography alone for privacy-preserving cloud computing," in *Proc. of the 5th USENIX conference on Hot topics in security*, pp.1-8, 2010. http://portal.acm.org/citation.cfm?id=1924934

[3] E. Bertino, F. Paci, R. Ferrini, and N. Shang, "Privacy-preserving digital identity management for cloud computing," *Data Engineering,* vol. 32, pp.21-27, 2009. http://sites.computer.org/debull/A09mar/A09MAR-CD.pdf#page=23

[4] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications,* vol. 34, pp.1-11, 2011. Article (CrossRef Link)

[5] S. Chow, Y. J. He, L. Hui, and S. Yiu, "SPICE: Simple privacy-preserving identity-management for cloud environment," in *Proc. of Applied Cryptography and Network Security*, pp.526-543, 2012. Article (CrossRef Link)

[6] J. Chen, Y. Wang, and X. Wang, "On-demand security architecture for cloud computing," *Computer,* vol. 45, pp.73-78, 2012. Article (CrossRef Link)

[7] D. Chaum, "Blind signatures for untraceable payments," in *Proc. Of Crypto '82*, pp.199-203, 1982.

[8] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM,* vol. 24, pp.770-772, 1981. Article (CrossRef Link)

[9] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," in *Proc. of the RSLA '89,* pp.233-271, 1989. http://rspa.royalsocietypublishing.org/content/426/1871/233.short

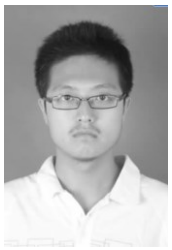
[10] X. Lin, and X. Li, "Achieving efficient cooperative message authentication in vehicular ad hoc networks," *IEEE T. Vehicular Technology*, vol.62, no.7, pp.3339-3348, 2013. Article (CrossRef Link)
[11] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems,* vol.23, no.9, pp.1621-1631, 2012. Article (CrossRef Link)
[12] R. Lu, X. Lin, X. Liang, and X. Shen, "A dynamic privacy-preserving key management scheme for location-based services in VANETs," *IEEE Transactions on Intelligent Transportation Systems,* vol.13, no.1, pp.127-139, 2012. Article (CrossRef Link)
[13] H. Zhu, T. Liu, and G. Wei, "PPAS: privacy protection authentication scheme for VANET," *Cluster Computing*, pp.1-14, 2013. Article (CrossRef Link)
[14] P. Angin, B. Bhargava, R. Ranchal, N. Singh, M. Linderman, and L. B. Othmane, "An entity-centric approach for privacy and identity management in cloud computing," in *Proc. of IEEE RDS '10,* pp.177-183, 2010. Article (CrossRef Link)
[15] J. Chen, G. Wu, L. Shen, and Z. Ji, "Differentiated security levels for personal identifiable information in identity management system," *Expert Systems with Applications,* vol.38, pp. 14156-14162, 2011. Article (CrossRef Link)
[16] J. Chen, G. Wu, and Z. Ji, "Secure interoperation of identity managements among different circles of trust," *Computer Standards & Interfaces,* vol.33, pp.533-540, 2011. Article (CrossRef Link)
[17] H. Yang, H. Kim, H. Li, E. Yoon, X. Wang, and X. Ding, "An efficient broadcast authentication scheme with batch verification for ADS-B messages," *KSII Transactions on Internet and Information Systems (TIIS)*, vol.7, no.10, pp.2544-2560, 2013. Article (CrossRef Link)
[18] R. Ranchal, B. Bhargava, L. B. Othmane, L. Lilien, A. Kim, and M. Kang, "Protection of identity information in cloud computing without trusted third party," in *Proc. IEEE RDS '10,* pp. 368-372, 2010. Article (CrossRef Link)
[19] C. T. Li, C. C. Lee, C. Y. Weng, and C. I. Fan, "An extended multi-server-based tenant authentication and key agreement scheme with tenant anonymity," *KSII Transactions on Internet and Information Systems (TIIS),* vol. 7, no.1, pp.119-131, 2013. Article (CrossRef Link)
[20] J. Xiong, Z. Yao, and J. Ma, "Action-based multilevel access control for structured document," *Journal of Computer Research and Development*, vol.50, no.7, pp.1399-1408, 2013. http://d.wanfangdata.com.cn/periodical_jsjyjyfz201307005.aspx
[21] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *Proc. of ACM CCS '10,* pp.735-737, 2010. Article (CrossRef Link)
[22] X. Liu, J. Ma, J. Xiong, and G. Liu, "Ciphertext-policy hierarchical attribute-based encryption for fine-grained access control of encryption data," *International Journal of Network Security*, vol.16, no.4, pp.351-357, 2014. http://ijns.femto.com.tw/contents/ijns-v16-n6/ijns-2014-v16-n6-p437-443.pdf
[23] X. Liu, J. Ma, and J. Xiong, "Ciphertext policy weighted attribute based encryption scheme," *Journal of Xi'an Jiaotong University*, vol.47, no.8, pp.44-48, 2013. http://d.wanfangdata.com.cn/periodical_xajtdxxb201308008.aspx
[24] F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," in *Proc. of ISPEC '11*, pp.83-97, 2011. Article (CrossRef Link)
[25] M. Nabeel, E. Bertino, and M. Kantarcioglu, "Towards privacy preserving access control in the cloud," in *Proc. of CollaborateCom '11*, pp.172-180, 2011. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6144802
[26] S. C. Yu, C. Wang, K. Ren, and W. J. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. of INFOCOM '10*, pp.1-9, 2010. Article (CrossRef Link)
[27] J. Kolter, R. Schillinger, and G. Pernul, "A privacy-enhanced attribute-based access control system," in *Proc. of DAS XXI '07,* pp.129-143, 2007. Article (CrossRef Link)

[28] A. A. Almutairi, M. I. Sarfraz, S. Basalamah, W. G. Aref, and A. Ghafoor, "A distributed access control architecture for cloud computing," *IEEE Software,* vol.29, pp.36-44, 2012. Article (CrossRef Link)
[29] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Transactions on Parallel and Distributed Systems,* vol.24, no.3, pp.614-624, 2013. Article (CrossRef Link)
[30] X. Lin, R. Lu, X. Shen, Y. Nemoto and N. Kato, "SAGE: A strong privacy-preserving scheme against global eavesdropping for eHealth systems," IEEE Journal on Selected Areas in Communications, vol.27, no.4, pp.365-378, 2009. Article (CrossRef Link)
[31] K. Ren, W. Lou, K. Kim, and R. Deng, "A novel privacy preserving authentication and access control scheme for pervasive computing environments," *IEEE Transactions on Vehicular Technology,* vol. 55, pp.1373-1384, 2006. Article (CrossRef Link)
[32] C. T. Li, M. S. Hwang, and Y. P. Chu, "Further improvement on a novel privacy preserving authentication and access control scheme for pervasive computing environments," *Computer Communications,* vol. 31, pp. 4255-4258, 2008. Article (CrossRef Link)
[33] Z. Tan, "A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments," *Journal of Network and Computer Applications,* vol. 35, pp. 1839–1846, 2012. Article (CrossRef Link)
[34] M. Ruckert, and D. Schroder, "Fair partially blind signatures," in *Proc. of Africacrypt '10,* pp. 34-51, 2010. Article (CrossRef Link)
[35] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy preserving access control with authentication for securing data in clouds," in *Proc. of IEEE/ACM CCGrid '12,* pp.556-563, 2012. Article (CrossRef Link)
[36] J. Li, M. H. Au, W. Susilo, D. Xie, and K. Ren, "Attribute-based signature and its applications," in *Proc. of ACM ASIACCS '10,* pp.60-69, 2010. Article (CrossRef Link)
[37] J. Xiong, Z. Yao, and J. Ma, "PRAM: privacy preserving access management scheme in cloud services," in *Proc. of ACM ASIACCS cloudcomputing '13,* pp.41-46, 2013. Article (CrossRef Link)
[38] R. C. Merkle, "One way hash functions and DES," in *Proc. of CRYPTO '8*9, pp.428-446, 1990. Article (CrossRef Link)
[39] Q. He, D. Wu, and P. Khosla, "The quest for personal control over mobile location privacy," *IEEE Communications Magazine,* vol.42, pp.130-136, 2004. Article (CrossRef Link)
[40] J. Xiong, Z. Yao, and J. Ma, "Multilevel access control model for video database," *Journal on Communications*, vol.33, no.8, pp.147-154, 2012. http://www.joconline.com.cn/ch/reader/view_abstract.aspx?file_no=20120818
[41] A. Weimerskirch and D. Westhoff, "Zero common-knowledge authentication for pervasive networks," in *Proc. of ACM SAC '0*3, pp.73-87, 2003. Article (CrossRef Link)
[42] S. Xu and M. Yung, "K-anonymous secret handshakes with reusable credentials," in *Proc. of ACM CCS '0*4, pp.158-167, 2004. Article (CrossRef Link)

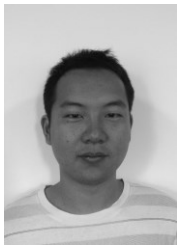
Jinbo Xiong, received the M.S. degree in Communication and Information Systems from Chongqing University of Posts and Telecommunications, China, in 2006, and received the Ph.D. degree in Computer Systems Architecture from Xidian University, China, in 2013. Currently, he is a lecturer in Fujian Normal University, Member of the ACM, Member of the China Computer Federation (CCF), and Student Member of the IEEE. His research interests mainly focus on identity privacy and access control in cloud computing, big data security and privacy.

Zhiqiang Yao, received the M.S. degree from East China Normal University, China, in 1992. Currently, he is a professor in Fujian Normal University, a Ph.D. candidate in Xidian University, ACM Professional Membership, Senior Member of China Computer Federation (CCF). His research interests mainly focus on security in cloud computing, multimedia security.

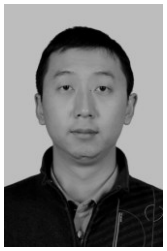
Jianfeng Ma, received the Ph.D. degrees in Computer Software and Communications Engineering from Xidian University, China, in 1995. Currently, he is a professor, Ph.D. supervisor, IEEE Member, Senior Member of China Computer Federation (CCF), Senior Member of Chinese Institute of Electronics (CIE). His research interests are in distributed systems, wireless and mobile computing systems, computer networks, and network security. He has published over 150 refereed articles in these areas and coauthored ten books.



Ximeng Liu, Ph.D. Candidate, IEEE Student Member, CCF Student Member. Since 2013, he has already publish exceed twenty research articles in both journals and conferences, such as IET information security, Journal on Communications. His research interests include public key cryptography, network security, big data security and privacy.



Qi Li, received the M.S. degree from Xidian University, China, in 2009. Currently, he is a Ph.D. Candidate at the School of Computer Science and Technology, Xidian University, China. IEEE Student Member. His research interests include attribute-based cryptography and information security.



Jun Ma, received the M.S. in Computer Software from Zhengzhou Institute of information technology, China, in 2006. Currently, he is a Ph.D. Candidate at the School of Computer Science and Technology, Xidian University, China. CCF Member and IEEE Student Member. His research interests mainly focus on identity privacy and access control in network security.