Shared Authority Based Privacy-Preserving Authentication Protocol in Cloud Computing

Hong Liu, *Student Member, IEEE*, Huansheng Ning, *Senior Member, IEEE*, Qingxu Xiong, *Member, IEEE*, and Laurence T. Yang, *Member, IEEE*

Abstract—Cloud computing is an emerging data interactive paradigm to realize users' data remotely stored in an online cloud server. Cloud services provide great conveniences for the users to enjoy the on-demand cloud applications without considering the local infrastructure limitations. During the data accessing, different users may be in a collaborative relationship, and thus data sharing becomes significant to achieve productive benefits. The existing security solutions mainly focus on the authentication to realize that a user's privative data cannot be illegally accessed, but neglect a subtle privacy issue during a user challenging the cloud server to request other users for data sharing. The challenged access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions. In this paper, we propose a shared authority based privacy-preserving authentication protocol (SAPA) to address above privacy issue for cloud storage. In the SAPA, 1) shared access authority is achieved by anonymous access request matching mechanism with security and privacy considerations (e.g., authentication, data anonymity, user privacy, and forward security); 2) attribute based access control is adopted to realize that the user can only access its own data fields; 3) proxy re-encryption is applied to provide data sharing among the multiple users. Meanwhile, universal composability (UC) model is established to prove that the SAPA theoretically has the design correctness. It indicates that the proposed protocol is attractive for multi-user collaborative cloud applications.

Index Terms—Cloud computing, authentication protocol, privacy preservation, shared authority, universal composability

1 Introduction

Cloud computing is a promising information technology architecture for both enterprises and individuals. It launches an attractive data storage and interactive paradigm with obvious advantages, including on-demand self-services, ubiquitous network access, and location independent resource pooling [1]. Towards the cloud computing, a typical service architecture is anything as a service (XaaS), in which infrastructures, platform, software, and others are applied for ubiquitous interconnections. Recent studies have been worked to promote the cloud computing evolve towards the internet of services [2], [3]. Subsequently, security and privacy issues are becoming key concerns with the increasing popularity of cloud services. Conventional security approaches mainly focus on the strong authentication to realize that a user can remotely access its own data in on-demand mode. Along with the diversity of the application requirements, users may want to access and share each other's authorized data fields to achieve productive benefits, which brings new security and privacy challenges for the cloud storage.

An example is introduced to identify the main motivation. In the cloud storage based supply chain management, there are various interest groups (e.g., supplier, carrier, and retailer) in the system. Each group owns its users which are permitted to access the authorized data fields, and different users own relatively independent access authorities. It means that any two users from diverse groups should access different data fields of the same file. Thereinto, a supplier may want to access a carrier's data fields, but it is not sure whether the carrier will allow its access request. If the carrier refuses its request, the supplier's access desire will be revealed along with nothing obtained towards the desired data fields. Actually, the supplier may not send the access request or withdraw the unaccepted request in advance if it firmly knows that its request will be refused by the carrier. It is unreasonable to thoroughly disclose the supplier's private information without any privacy considerations. Fig. 1 illustrates three revised cases to address above imperceptible privacy issue.

- *Case 1.* The carrier also wants to access the supplier's data fields, and the cloud server should inform each other and transmit the shared access authority to the both users;
- *Case 2.* The carrier has no interest on other users' data fields, therefore its authorized data fields should be properly protected, meanwhile the supplier's access request will also be concealed;
- *Case 3.* The carrier may want to access the retailer's data fields, but it is not certain whether the retailer will accept its request or not. The retailer's authorized data fields should not be public if the retailer

- *H. Liu and Q. Xiong are with the School of Electronic and Information Engineering, Beihang University, Beijing, China. E-mail: liuhongler@ee.buaa.edu.cn, qxxiong@buaa.edu.cn.*
- *H. Ning is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China, and the School of Electronic and Information Engineering, Beihang University, Beijing, China. E-mail: ninghuansheng@ustb.edu.cn.*
- *L.T. Yang is with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei, China, and the Department of Computer Science, St. Francis Xavier University, Antigonish, NS, Canada. E-mail: ltyang@stfx.ca.*

Manuscript received 3 Nov. 2013; revised 23 Dec. 2013; accepted 30 Dec. 2013. Date of publication 24 Feb. 2014; date of current version 5 Dec. 2014. Recommended for acceptance by J. Chen.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPDS.2014.2308218

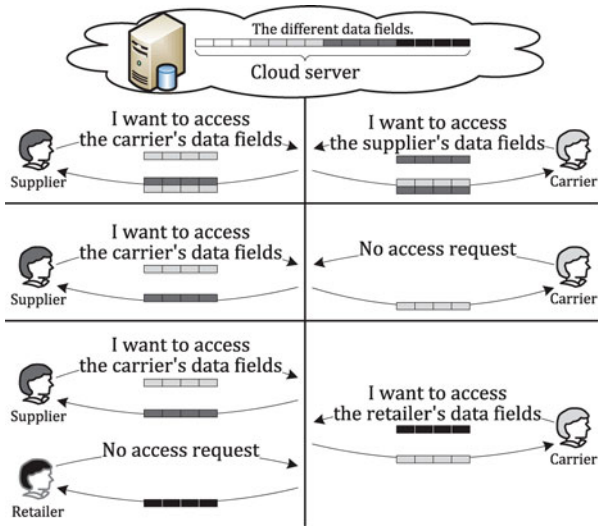

Fig. 1. Three possible cases during data accessing and data sharing in cloud applications.

has no interests in the carrier's data fields, and the carrier's request is also privately hidden.

Towards above three cases, security protection and privacy preservation are both considered without revealing sensitive access desire related information.

In the cloud environments, a reasonable security protocol should achieve the following requirements. *1) Authentication*: a legal user can access its own data fields, only the authorized partial or entire data fields can be identified by the legal user, and any forged or tampered data fields cannot deceive the legal user. *2) Data anonymity*: any irrelevant entity cannot recognize the exchanged data and communication state even it intercepts the exchanged messages via an open channel. *3) User privacy*: any irrelevant entity cannot know or guess a user's access desire, which represents a user's interest in another user's authorized data fields. If and only if the both users have mutual interests in each other's authorized data fields, the cloud server will inform the two users to realize the access permission sharing. *4) Forward security*: any adversary cannot correlate two communication sessions to derive the prior interrogations according to the currently captured messages.

Researches have been worked to strengthen security protection and privacy preservation in cloud applications, and there are various cryptographic algorithms to address potential security and privacy problems, including security architectures [4], [5], data possession protocols [6], [7], data public auditing protocols [8], [9], [10], secure data storage and data sharing protocols [11], [12], [13], [14], [15], [16], access control mechanisms [17], [18], [19], privacy preserving protocols [20], [21], [22], [23], and key management [24], [25], [26], [27]. However, most previous researches focus on the authentication to realize that only a legal user can access its authorized data, which ignores that different users may want to access and share each other's authorized data fields to achieve productive benefits. When a user challenges the cloud server to request other users for data sharing, the access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions. In this work, we aim to address a user's sensitive access desire related privacy during data sharing in the cloud environments, and it is significant to design a humanistic security scheme to simultaneously achieve data access control, access authority sharing, and privacy preservation.

In this paper, we address the aforementioned privacy issue to propose a shared authority based privacy-preserving authentication protocol (SAPA) for the cloud data storage, which realizes authentication and authorization without compromising a user's private information. The main contributions are as follows.

1) Identify a new privacy challenge in cloud storage, and address a subtle privacy issue during a user challenging the cloud server for data sharing, in which the challenged request itself cannot reveal the user's privacy no matter whether or not it can obtain the access authority.
2) Propose an authentication protocol to enhance a user's access request related privacy, and the shared access authority is achieved by anonymous access request matching mechanism.
3) Apply ciphertext-policy attribute based access control to realize that a user can reliably access its own data fields, and adopt the proxy re-encryption to provide temp authorized data sharing among multiple users.

The remainder of the paper is organized as follows. Section 2 introduces related works. Section 3 introduces the system model, and Section 4 presents the proposed authentication protocol. The universal composability (UC) model based formal security analysis is performed in Section 5 Finally, Section 6 draws a conclusion.

2 Related Work

Dunning and Kresman [11] proposed an anonymous ID assignment based data sharing algorithm (AIDA) for multi-party oriented cloud and distributed computing systems. In the AIDA, an integer data sharing algorithm is designed on top of secure sum data mining operation, and adopts a variable and unbounded number of iterations for anonymous assignment. Specifically, Newton's identities and Sturm's theorem are used for the data mining, a distributed solution of certain polynomials over finite fields enhances the algorithm scalability, and Markov chain representations are used to determine statistics on the required number of iterations.

Liu et al. [12] proposed a multi-owner data sharing secure scheme (Mona) for dynamic groups in the cloud applications. The Mona aims to realize that a user can securely share its data with other users via the untrusted cloud server, and can efficiently support dynamic group interactions. In the scheme, a new granted user can directly decrypt data files without pre-contacting with data owners, and user revocation is achieved by a revocation list without updating the secret keys of the remaining users. Access control is applied to ensure that any user in a group can anonymously utilize the cloud resources, and the data owners' real identities can only be revealed by the group manager for dispute arbitration. It indicates the storage overhead and encryption computation cost are independent with the amount of the users.

Grzonkowski and Corcoran [13] proposed a zero-knowledge proof (ZKP) based authentication scheme for cloud services. Based on the social home networks, a user centric approach is applied to enable the sharing of personalized content and sophisticated network-based services via TCP/IP infrastructures, in which a trusted third party is introduced for decentralized interactions.

Nabeel et al. [14] proposed a broadcast group key management (BGKM) to improve the weakness of symmetric key cryptosystem in public clouds, and the BGKM realizes that a user need not utilize public key cryptography, and can dynamically derive the symmetric keys during decryption. Accordingly, attribute based access control mechanism is designed to achieve that a user can decrypt the contents if and only if its identity attributes satisfy the content provider's policies. The fine-grained algorithm applies access control vector (ACV) for assigning secrets to users based on the identity attributes, and allowing the users to derive actual symmetric keys based on their secrets and other public information. The BGKM has an obvious advantage during adding/revoking users and updating access control policies.

Wang et al. [15] proposed a distributed storage integrity auditing mechanism, which introduces the homomorphic token and distributed erasure-coded data to enhance secure and dependable storage services in cloud computing. The scheme allows users to audit the cloud storage with lightweight communication overloads and computation cost, and the auditing result ensures strong cloud storage correctness and fast data error localization. Towards the dynamic cloud data, the scheme supports dynamic outsourced data operations. It indicates that the scheme is resilient against Byzantine failure, malicious data modification attack, and server colluding attacks.

Sundareswaran et al. [16] established a decentralized information accountability framework to track the users' actual data usage in the cloud, and proposed an object-centered approach to enable enclosing the logging mechanism with the users' data and policies. The Java ARchives (JAR) programmable capability is leveraged to create a dynamic and mobile object, and to ensure that the users' data access will launch authentication. Additionally, distributed auditing mechanisms are also provided to strengthen user's data control, and experiments demonstrate the approach efficiency and effectiveness.

In the aforementioned works, various security issues are addressed. However, a user's subtle access request related privacy problem caused by data accessing and data sharing has not been studied yet in the literature. Here, we identify a new privacy challenge, and propose a protocol not only focusing on authentication to realize the valid data accessing, but also considering authorization to provide the privacy-preserving access authority sharing. The attribute based access control and proxy re-encryption mechanisms are jointly applied for authentication and authorization.

3 System Model

Fig. 2 illustrates a system model for the cloud storage architecture, which includes three main network entities: users (U_x), a cloud server (S), and a trusted third party.

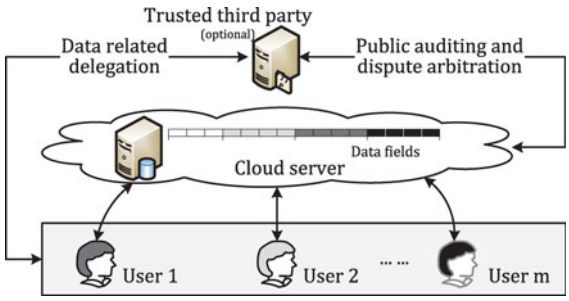

Fig. 2. The cloud storage system model.

- *User.* An individual or group entity, which owns its data stored in the cloud for online data storage and computing. Different users may be affiliated with a common organization, and are assigned with independent authorities on certain data fields.
- *Cloud server.* An entity, which is managed by a particular cloud service provider or cloud application operator to provide data storage and computing services. The cloud server is regarded as an entity with unrestricted storage and computational resources.
- *Trusted third party.* An optional and neutral entity, which has advanced capabilities on behalf of the users, to perform data public auditing and dispute arbitration.

In the cloud storage, a user remotely stores its data via online infrastructures, flatforms, or software for cloud services, which are operated in the distributed, parallel, and cooperative modes. During cloud data accessing, the user autonomously interacts with the cloud server without external interferences, and is assigned with the full and independent authority on its own data fields. It is necessary to guarantee that the users' outsourced data cannot be unauthorized accessed by other users, and is of critical importance to ensure the private information during the users' data access challenges. In some scenarios, there are multiple users in a system (e.g., supply chain management), and the users could have different affiliation attributes from different interest groups. One of the users may want to access other associated users' data fields to achieve bi-directional data sharing, but it cares about two aspects: whether the aimed user would like to share its data fields, and how to avoid exposing its access request if the aimed user declines or ignores its challenge. In the paper, we pay more attention on the process of data access control and access authority sharing other than the specific file oriented cloud data management.

In the system model, assume that point-to-point communication channels between users and a cloud server are reliable with the protection of secure shell protocol (SSH). The related authentication handshakes are not highlighted in the following protocol presentation.

Towards the trust model, there are no full trust relationships between a cloud server S and a user U_x.

- *S is semi-honest and curious.* Being semi-honest means that S can be regarded as an entity that appropriately follows the protocol procedure. Being curious

TABLE 1
Notations

Notation	Description
S, U_x	The cloud server, and a user (i.e., cloud data owner).
PID_{U_x}	U_x's pseudorandom identifier (pseudonym).
T_{U_x}	U_x's identity token that is assigned by S.
sid_{S_x}, sid_{U_x}	The pseudorandom session identifier of S, U_x.
α, σ, β, r_{U_x}	The randomly generated numbers.
$R_{U_x}^{U_y}$	The access request pointer that represents U_x's access desire on U_y's data fields.
D_{U_x}, $\dot{D}_{U_x}$	U_x's own authorized data fields, and U_x's temp authorized data fields.
A_{U_x}, L_{U_x}, P_{U_x}	The data attribute access list, re-structure data access list, and data access policy.
$\{mpk/msk\}$	The pairwise master public/privacy keys.
$\{pk/sk\}$	The pairwise public/privacy keys.
k_{Σ_x}, k_{U_x}	The aggregated keys, and the re-encryption keys.
V^{ℓ}	The locally computed value V according to the same algorithm.
C_{S_x}, C_{U_x}	The ciphertexts.
$\mathcal{F}_{S_x}(x, P_{U_x})$	The defined polynomial owned by S.
$\mathcal{F}_{U_x}(x, L_{U_x})$	The defined polynomial owned by U_x.

means that S may attempt to obtain U_x's private information (e.g., data content, and user preferences). It means that S is under the supervision of its cloud provider or operator, but may be interested in viewing users' privacy. In the passive or honest-but-curious model, S cannot tamper with the users' data to maintain the system normal operation with undetected monitoring.

- *U_x is rational and sensitive.* Being rational means that U_x's behavior would be never based on experience or emotion, and misbehavior may only occur for selfish interests. Being sensitive means that U_x is reluctant to disclosure its sensitive data, but has strong interests in other users' privacy.

Towards the threat model, it covers the possible security threats and system vulnerabilities during cloud data interactions. The communication channels are exposed in public, and both internal and external attacks exist in the cloud applications [15]. The internal attacks mainly refer to the interactive entities (i.e., S, and U_x). Thereinto, S may be self-centered and utilitarian, and aims to obtain more user data contents and the associated user behaviors/habits for the maximization of commercial interests; U_x may attempt to capture other users' sensitive data fields for certain purposes (e.g., curiosity, and malicious intent). The external attacks mainly consider the data CIA triad (i.e., confidentiality, integrity, and availability) threats from outside adversaries, which could compromise the cloud data storage servers, and subsequently modify (e.g., insert, or delete) the users' data fields.

4 The Shared Authority Based Privacy-Preserving Authentication Protocol

4.1 System Initialization

The cloud storage system includes a cloud server S, and users $\{U_x\}$ $(x = \{1, \ldots, m\},\ m \in \mathbb{N}^*)$. Thereinto, U_a and U_b are two users, which have independent access authorities on their own data fields. It means that a user has an access permission for particular data fields stored by S, and the user cannot exceed its authority access to obtain other users' data fields. Here, we consider S and $\{U_a, U_b\}$ to present the protocol for data access control and access authority sharing with enhanced privacy considerations. The main notations are introduced in Table 1.

Let $\mathbb{BG} = (q, g, h, \mathbb{G}, \mathbb{G}', e, H)$ be a pairing group, in which q is a large prime, $\{\mathbb{G}, \mathbb{G}'\}$ are of prime order q, $\mathbb{G} = \langle g \rangle = \langle h \rangle$, and H is a collision-resistant hash function. The bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}'$ satisfies the bilinear non-degenerate properties: i.e., for all $g, h \in \mathbb{G}$ and $a, b \in Z_q^*$, it turns out that $e(g^a, h^b) = e(g, h)^{ab}$, and $e(g, h) \neq 1$. Meanwhile, $e(g, h)$ can be efficiently obtained for all $g, h \in \mathbb{G}$, and it is a generator of $\mathbb{G}'$.

Let S and U_x respectively own the pairwise keys $\{pk_S, sk_S\}$ and $\{pk_{U_x}, sk_{U_x}\}$. Besides, S is assigned with all users' public keys $\{pk_{U_1}, \ldots, pk_{U_m}\}$, and U_x is assigned with pk_S. Here, the public key $pk_\tau = g^{sk_\tau} (\mathrm{mod}\ q)$ $(\tau \in \{S, U_x\})$ and the corresponding privacy key $sk_\tau \in \mathbb{Z}_q^*$ are defined according to the generator g.

Let $\mathcal{F}(R_{U_x}^{U_y}(R_{U_y}^{U_x})^T) = Cont \in \mathbb{Z}_q$ describe the algebraic relation of $\{R_{U_x}^{U_y}, R_{U_y}^{U_x}\}$, which are mutually inverse access requests challenged by $\{U_x, U_y\}$, and $Cont$ is a constant. Here, $\mathcal{F}(.)$ is a collision-resistant function, for any randomized polynomial time algorithm A, there is a negligible function $p(k)$ for a sufficiently large value k:

$$\mathrm{Prob}\Big[\{(x, x'); (y, y')\} \leftarrow A(1^k) : (x \neq x', y \neq y') \wedge \mathcal{F}\Big(R_{U_y}^{U_x}\Big(R_{U_x'}^{U_y'}\Big)^T\Big) = Cont\Big] \leq p(k).$$

Note that $R_{U_\dagger}^{U_*}$ is a m-dimensional Boolean vector, in which only the $*$-th pointed-element and the $\dagger$-th self-element are 1, and other elements are 0. It turns out that:

- $\mathcal{F}(R_{U_x}^{U_y}(R_{U_y}^{U_x})^T) = \mathcal{F}(2) = Cont$ means that both U_x and U_y are interested in each other's data fields, and the two access requests are matched;
- $\mathcal{F}(R_{U_x}^{U_y}(R_{U_y}^{U_{\tilde{x}}})^T) = \mathcal{F}(R_{U_{\tilde{y}}}^{U_y}(R_{U_y}^{U_x})^T) = \mathcal{F}(1)$ means that only one user (i.e., U_x or U_y) is interested in the other's data fields, and the access requests are not matched. Note that $U_{\tilde{x}}/U_{\tilde{y}}$ represents that the user is not U_x/U_y;
- $\mathcal{F}(R_{U_x}^{U_{\tilde{y}}}(R_{U_y}^{U_{\tilde{x}}})^T) = \mathcal{F}(0)$ means that neither U_x nor U_y is interested in each other's data fields, and the two access requests are not matched.

Let $\mathbb{A}$ be the attribute set, there are n attributes $\mathbb{A} = \{A_1, A_2, \ldots, A_n\}$ for all users, and U_x has its own attribute set $A_{U_x} \subset \mathbb{A}$ for data accessing. Let A_{U_x} and P_{U_x} be monotone Boolean matrixes to represent U_x's data attribute access list and data access policy.

- Assume that U_x has $A_{U_x} = [a_{ij}]_{n \times m}$, which satisfies that $a_{ij} = 1$ for $A_i \in \mathbb{A}$, and $a_{ij} = 0$ for $A_i \notin \mathbb{A}$.
- Assume that S owns $P_{U_x} = [p_{ij}]_{n \times m}$, which is applied to restrain U_x's access authority, and satisfies that $p_{ij} = 1$ for $A_i \in P_{U_x}$, and $p_{ij} = 0$ for $A_i \notin P_{U_x}$. If $a_{ij} \leq p_{ij} \forall i = \{1, \ldots, n\}, j = \{1, \ldots, m\}$ holds, it will be regarded that A_{U_x} is within P_{U_x}'s access authority limitation.

Note that full-fledged cryptographic algorithms (e.g., attribute based access control, and proxy re-encryption) can be exploited to support the SAPA.

4.2 The Proposed Protocol Descriptions

Fig. 3 shows the interactions among $\{U_a, U_b, S\}$, in which both U_a and U_b have interests on each other's authorized data fields for data sharing. Note that the presented interactions may not be synchronously launched, and a certain time interval is allowable.

4.2.1 $\{U_a, U_b\}$'s Access Challenges and S's Responses

$\{U_a, U_b\}$ respectively generate the session identifiers $\{sid_{U_a}, sid_{U_b}\}$, extract the identity tokens $\{T_{U_a}, T_{U_b}\}$, and transmits $\{sid_{U_a}\|T_{U_a}, sid_{U_b}\|T_{U_a}\}$ to S as an access query to initiate a new session. Accordingly, we take the interactions of U_a and S as an example to introduce the following authentication phase. Upon receiving U_a's challenge, S first generates a session identifier sid_{S_a}, and establishes the master public key $mpk = (g_i, h, h_i, \mathbb{BG}, e(g,h), H)$ and master privacy key $msk = (\alpha, g)$. Thereinto, S randomly chooses $\alpha \in \mathbb{Z}_q$, and computes $g_i = g^{\alpha^i}$ and $h_i = h^{\alpha^{i-1}}$ $(i = \{1, \ldots, n\} \in \mathbb{Z}^*)$.

S randomly chooses $\sigma \in \{0,1\}^*$, and extracts U_a's access authority policy $P_{U_a} = [p_{ij}]_{n\times m}$ $(p_{ij} \in \{0,1\})$, and U_a is assigned with the access authority on its own data fields D_{U_a} within P_{U_a}'s permission. S further defines a polynomial $\mathcal{F}_{S_a}(x, P_{U_a})$ according to P_{U_a} and T_{U_a}:

$$\mathcal{F}_{S_a}(x, P_{U_a}) = \prod_{i=1,j=1}^{n,m} (x + ijH(T_{U_a}))^{p_{ij}} \pmod q.$$

S computes a set of values $\{M_{S_a0}, M_{S_a1}, \{M_{S_a2i}\}, M_{S_a3}, M_{S_a4}\}$ to establish the ciphertext $C_{S_a} = \{M_{S_a1}, \{M_{S_a2i}\}, M_{S_a3}, M_{S_a4}\}$, and transmits $sid_{S_a}\|C_{S_a}$ to U_a.

$$\begin{aligned}
M_{S_a0} &= H(P_{U_a}\|D_{U_a}\|T_{U_a}\|\sigma),\\
M_{S_a1} &= h^{\mathcal{F}_{S_a}(\alpha,P_{U_a})M_{S_a0}},\\
M_{S_a2i} &= (g_i)^{M_{S_a0}}, (i = 1, \ldots, n),\\
M_{S_a3} &= H(e(g,h)^{M_{S_a0}}) \oplus \sigma,\\
M_{S_a4} &= H(sid_{U_a}\|\sigma) \oplus D_{U_a}.
\end{aligned}$$

Similarly, S performs the corresponding operations for U_b, including that S randomly chooses $\alpha' \in \mathbb{Z}_q$ and $\sigma' \in \{0,1\}^*$, establishes $\{g'_i, h'_i\}$, extracts $\{P_{U_b}, D_{U_b}\}$, defines $\mathcal{F}_{S_b}(x, P_{U_b})$, and computes $\{M_{S_b0}, M_{S_b1}, \{M_{S_b2i}\}, M_{S_b3}, M_{S_b4}\}$ to establish the ciphertext C_{S_b} for transmission.

4.2.2 $\{U_a, U_b\}$'s Data Access Control

U_a first extracts it data attribute access list $A_{U_a} = [a_{ij}]$ $(a_{ij} \in \{0,1\},\ a_{ij} \le p_{ij})$ to re-structure an access list $L_{U_a} = [l_{ij}]_{n\times m}$ for $l_{ij} = p_{ij} - a_{ij}$. U_a also defines a polynomial $\mathcal{F}_{U_a}(x, L_{U_a})$ according to L_{U_a} and T_{U_a}:

$$\mathcal{F}_{U_a}(x, L_{U_a}) = \prod_{i=1,j=1}^{n,m} (x + ijH(T_{U_a}))^{l_{ij}} \pmod q.$$

It turns out that $\mathcal{F}_{U_a}(x, L_{U_a})$ satisfies the equation

$$\begin{aligned}
\mathcal{F}_{U_a}(x, L_{U_a}) &= \prod_{i=1,j=1}^{n,m} (x + ijH(T_{U_a}))^{p_{ij}-a_{ij}}\\
&= \mathcal{F}_{S_a}(x, P_{U_a})/\mathcal{F}_{S_a}(x, A_{U_a}).
\end{aligned}$$

Afterwards, U_a randomly chooses $\beta \in \mathbb{Z}_q$, and the decryption key $k_{A_{U_a}}$ for A_{U_a} can be obtained as follows:

$$k_{A_{U_a}} = (g^{(\beta+1)/\mathcal{F}_{S_a}(\alpha,A_{U_a})}, h^{\beta-1}).$$

U_a further computes a set of values $\{N_{U_a1}, N_{U_a2}, N_{U_a3}\}$. Here, $f_{S_a i}$ is used to represent x^i's coefficient in $\mathcal{F}_{S_a}(x, P_{U_a})$, and $f_{U_a i}$ is used to represent x^i's coefficient in $\mathcal{F}_{U_a}(x, L_{U_a})$:

$$\begin{aligned}
N_{U_a1} &= e\left(M_{S_a21}, \prod_{i=1}^{n}(h_i)^{f_{U_a i}} h^{f_{U_a 0}}\right),\\
N_{U_a2} &= e\left(\prod_{i=1}^{n}(M_{S_a2i})^{f_{U_a i}}, h^{\beta-1}\right),\\
N_{U_a3} &= e(g^{(\beta+1)/\mathcal{F}_{S_a}(\alpha,A_{U_a})}, M_{S_a1}).
\end{aligned}$$

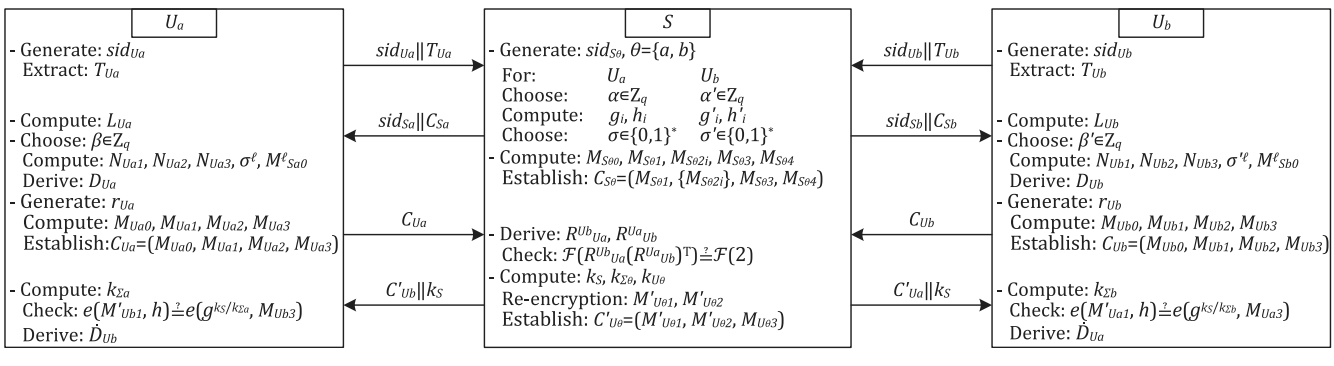

Fig. 3. The shared authority based privacy-preserving authentication protocol.

It turns out that $e(g,h)^{M_{S_a0}}$ satisfies the equation

$$e(g,h)^{M_{S_a0}} = \left(\frac{N_{U_a3}}{(N_{U_a1}N_{U_a2})}\right)^{1/f_{U_a0}}.$$

For the right side of (1), we have,

$$\begin{aligned} N_{U_a1} &= e\left(g^{\alpha^i M_{S_a0}}, \prod_{i=1}^{n}(h_i)^{f_{U_a i}} h^{f_{U_a0}}\right) \\ &= e(g,h)^{\alpha M_{S_a0}\sum_{i=1}^{n}(\alpha^{i-1}f_{U_a i}+f_{U_a0})} \\ &= e(g,h)^{M_{S_a0}\mathcal{F}_{U_a}(\alpha,L_{U_a})}, \end{aligned}$$

$$\begin{aligned} N_{U_a2} &= e\left(\prod_{i=1}^{n} g^{\alpha^i M_{S_a0} f_{U_a i}}, h^{\beta-1}\right) \\ &= e(g,h)^{M_{S_a0}\left(\sum_{i=1}^{n}\alpha^i f_{U_a i}+f_{U_a0}-f_{U_a0}\right)(\beta-1)} \\ &= e(g,h)^{M_{S_a0}\beta\mathcal{F}_{U_a}(\alpha,L_{U_a})-M_{S_a0}f_{U_a0}}, \end{aligned}$$

$$\begin{aligned} N_{U_a3} &= e\left(g^{(\beta+1)/\mathcal{F}_{S_a}(\alpha,A_{U_a})}, h^{f_{S_a0}M_{S_a0}}\prod_{i=1}^{n}(h_i)^{f_{S_a i}M_{S_a0}}\right) \\ &= e(g,h)^{(\beta+1)/\mathcal{F}_{S_a}(\alpha,A_{U_a})\mathcal{F}_{S_a}(\alpha,P_{U_a})M_{S_a0}} \\ &= e(g,h)^{M_{S_a0}\beta\mathcal{F}_{U_a}(\alpha,L_{U_a})+M_{S_a0}\mathcal{F}_{U_a}(\alpha,L_{U_a})}. \end{aligned}$$

U_a locally re-computes $\{\sigma^\ell, M^\ell_{S_a0}\}$, derives its own authorized data fields D_{U_a}, and checks whether the ciphertext C_{S_a} is encrypted by $M^\ell_{S_a0}$. If it holds, U_a will be a legal user that can properly decrypt the ciphertext C_{S_a}; otherwise, the protocol will terminate

$$\begin{aligned} \sigma^\ell &= M_{S_a3} \oplus H(e(g,h)^{M_{S_a0}}), \\ M^\ell_{S_a0} &= H\big(P_{U_a}\|D_{U_a}\|T_{U_a}\|\sigma^\ell\big), \\ D_{U_a} &= M_{S_a4} \oplus H\big(sid_{U_a}\|\sigma^\ell\big). \end{aligned}$$

U_a further extracts its pseudonym PID_{U_a}, a session-sensitive access request $R^{U_b}_{U_a}$, and the public key pk_{U_a}. Here, $R^{U_b}_{U_a}$ is introduced to let S know U_a's data access desire. It turns out that $R^{U_b}_{U_a}$ makes S know the facts: 1) U_a wants to access U_b's temp authorized data fields $\dot{D}_{U_b}$; 2) R_a will also agree to share its temp authorized data fields $\dot{D}_{U_a}$ with U_b in the case that U_b grants its request.

Afterwards, U_a randomly chooses $r_{U_a} \in \mathbb{Z}^*_q$, computes a set of values $\{M_{U_a0}, M_{U_a1}, M_{U_a2}, M_{U_a3}\}$ to establish a ciphertext C_{U_a}, and transmits C_{U_a} to S for further access request matching

$$\begin{aligned} M_{U_a0} &= H(sid_{S_a}\|PID_{U_a}) \oplus R^{U_b}_{U_a}, \\ M_{U_a1} &= g^{pk_{U_a}r_{U_a}}, \\ M_{U_a2} &= e(g,h)^{r_{U_a}}, \\ M_{U_a3} &= h^{r_{U_a}}. \end{aligned}$$

Similarly, U_b performs the corresponding operations, including that U_b extracts A_{U_b}, and determines $\{L_{U_b}, \mathcal{F}_{U_b}(x, L_{U_b}), f_{U_b i}\}$. U_b further randomly chooses $\beta' \in \mathbb{Z}_q$, and computes the values $\{N_{U_b1}, N_{U_b2}, N_{U_b3}, \sigma'^\ell, M^\ell_{U_b}\}$ to derive its own data fields D_{U_b}. U_b also extracts its pseudonym PID_{U_b} and an access request $R^{U_a}_{U_b}$ to establish a ciphertext C_{U_b} with the elements $\{M_{U_b0}, M_{U_b1}, M_{U_b2}, M_{U_b3}\}$.

4.2.3 $\{U_a, U_b\}$'s Access Request Matching and Data Access Authority Sharing

Upon receiving the ciphertexts $\{C_{U_a}, C_{U_b}\}$ within an allowable time interval, and S extracts $\{PID_{U_a}, PID_{U_b}\}$ to derive the access requests $\{R^{U_b}_{U_a}, R^{U_a}_{U_b}\}$:

$$\begin{aligned} R^{U_b}_{U_a} &= H(sid_{S_a}\|PID_{U_a}) \oplus M_{U_a0}, \\ R^{U_a}_{U_b} &= H(sid_{S_b}\|PID_{U_b}) \oplus M_{U_b0}. \end{aligned}$$

S checks whether $\{R^{U_b}_{U_a}, R^{U_a}_{U_b}\}$ satisfy $\mathcal{F}(R^{U_b}_{U_a}(R^{U_a}_{U_b})^T) = \mathcal{F}(2) = Cont$. If it holds, S will learn that both U_a and U_b have the access desires to access each other's authorized data, and to share its authorized data fields with each other. S extracts the keys $\{sk_S, pk_{U_a}, pk_{U_b}\}$ to establish the aggregated keys $\{k_S, k_{\Sigma_\theta}\}$ by the Diffie-Hellman key agreement, and computes the available re-encryption key k_{U_θ} for U_θ ($\theta \in \{a,b\}$):

$$\begin{aligned} k_S &= (pk_{U_a}pk_{U_b})^{sk_S} = g^{(sk_{U_a}+sk_{U_b})sk_S}, \\ k_{\Sigma_\theta} &= (pk_{U_\theta})^{sk_S} = g^{sk_{U_\theta}sk_S}, \\ k_{U_\theta} &= k_{\Sigma_\theta}/pk_{U_\theta}. \end{aligned}$$

S performs re-encryption to obtain $M'_{U_\theta 1}$. Towards U_a/U_b, S extracts U_b/U_a's temp authorized data fields $\dot{D}_{U_b}/\dot{D}_{U_a}$ to compute M'_{U_b2}/M'_{U_a2}:

$$\begin{aligned} M'_{U_\theta 1} &= (M_{U_\theta 1})^{k_{U_\theta}} = g^{k_{\Sigma_\theta}r_{U_\theta}}, \\ M'_{U_a2} &= M_{U_a2}E_{k_{\Sigma_b}}(\dot{D}_{U_a}), \\ M'_{U_b2} &= M_{U_b2}E_{k_{\Sigma_a}}(\dot{D}_{U_b}). \end{aligned}$$

Thereafter, S establishes the re-structured ciphertext $C'_{U_\theta} = (M'_{U_\theta 1}, M'_{U_\theta 2}, M_{U_\theta 3})$, and respectively transmits $\{C'_{U_b}\|k_S, C'_{U_a}\|k_S\}$ to $\{U_a, U_b\}$ for access authority sharing. Upon receiving the messages, U_a computes $k_{\Sigma_a} = (pk_S)^{sk_{U_a}}$, and performs verification by comparing the following equation:

$$e\big(M'_{U_b1}, h\big) \stackrel{?}{=} e(g^{k_S/k_{\Sigma_a}}, M_{U_b3}).$$

For the left side of (2), we have,

$$e\big(M'_{U_b1}, h\big) = e\big(g^{sk_{U_b}sk_S r_{U_b}}, h\big).$$

For the right side of (2), we have,

$$\begin{aligned} e\big(g^{k_S/k_{\Sigma_a}}, M_{U_b3}\big) &= e(g^{(pk_S)^{sk_{U_b}}}, h^{r_{U_b}}) \\ &= e(g,h)^{g^{sk_S sk_{U_b} r_{U_b}}}. \end{aligned}$$

U_a derives U_b's temp authorized data fields $\dot{D}_{U_b}$:

$$\dot{D}_{U_b} = E^{-1}_{k_{\Sigma_a}}\left(M'_{U_b2}e\left(M'_{U_b1}, h\right)^{-k_{\Sigma_a}/k_S}\right).$$

Similarly, U_b performs the corresponding operations, including that U_b obtains the keys $\{k_S, k_{\Sigma_b}\}$, checks U_b's validity, and derives the temp authorized data field $\dot{D}_{U_a}$.

In the SAPA, S acts as a semi-trusted proxy to realize $\{U_a, U_b\}$'s access authority sharing. During the proxy re-encryption, $\{U_a, U_b\}$ respectively establish ciphertexts $\{M_{U_a1}, M_{U_b1}\}$ by their public keys $\{pk_{U_a}, pk_{U_b}\}$, and S generates the corresponding re-encryption keys $\{k_{U_a}, k_{U_b}\}$ for $\{U_a, U_b\}$. Based on the re-encryption keys, the ciphertexts $\{M_{U_a1}, M_{U_b1}\}$ are re-encrypted into $\{M'_{U_a1}, M'_{U_b1}\}$, and $\{U_a, U_b\}$ can decrypt the re-structured ciphertexts $\{M'_{U_b1}, M'_{U_a1}\}$ by their own private key $\{sk_{U_a}, sk_{U_b}\}$ without revealing any sensitive information.

Till now, $\{U_a, U_b\}$ have realized the access authority sharing in the case that both U_a and U_b have the access desires on each other's data fields. Meanwhile, there may be other typical cases when U_a has an interest in U_b's data fields with a challenged access request $R^{U_b}_{U_a}$.

1. In the case that U_b has no interest in U_a's data fields, it turns out that U_b's access request $R^{U_b}_{U_b}$ and $R^{U_b}_{U_a}$ satisfy that $\mathcal{F}(R^{U_b}_{U_a}(R^{U_b}_{U_b})^T)=\mathcal{F}(1)$. For U_a, S will extract a dummy data fields D_{null} as a response. U_b will be informed that a certain user is interested in its data fields, but cannot determine U_a's detailed identity for privacy considerations.
2. In the case that U_b has an interest in U_c's data fields rather than U_a's data fields, but U_c has no interest in U_b's data fields. It turns out that the challenged access requests $R^{U_b}_{U_a}$, $R^{U_c}_{U_b}$, and $R^{U_{\tilde{b}}}_{U_c}$ satisfy that $\mathcal{F}(R^{U_b}_{U_a}(R^{U_c}_{U_b})^T)=\mathcal{F}(R^{U_c}_{U_b}(R^{U_{\tilde{b}}}_{U_c})^T)=\mathcal{F}(1)$, in which $U_{\tilde{b}}$ indicates that the user is not U_b. D_{null} will be transmitted to $\{U_a, U_b, U_c\}$ without data sharing.

In summary, the SAPA adopts integrative approaches to address secure authority sharing in cloud applications.

- *Authentication.* The ciphertext-policy attribute based access control and bilinear pairings are introduced for identification between U_θ and S, and only the legal user can derive the ciphertexts. Additionally, U_θ checks the re-computed ciphertexts according to the proxy re-encryption, which realizes flexible data sharing instead of publishing the interactive users' secret keys.
- *Data anonymity.* The pseudonym PID_{U_θ} are hidden by the hash function so that other entities cannot derives the real values by inverse operations. Meanwhile, $U_{\tilde{\theta}}$'s temp authorized fields $\dot{D}_{U_{\tilde{\theta}}}$ are encrypted by k_{Σ_θ} for anonymous data transmission. Hence, an adversary cannot recognize the data, even if the adversary intercepts the transmitted data, it will not decode the full-fledged cryptographic algorithms.
- *User privacy.* The access request pointer (e.g., $R^{U_x}_{U_\theta}$) is wrapped along with $H(sid_{S_\theta}\|PID_{U_\theta})$ for privately informing S about U_θ's access desires. Only if both users are interested in each other's data fields, S will establish the re-encryption key k_{U_θ} to realize authority sharing between U_a and U_b. Otherwise, S will temporarily reserve the desired access requests for a certain period of time, and cannot accurately determine which user is actively interested in the other user's data fields.
- *Forward security.* The dual session identifiers $\{sid_{S_\theta}, sid_{U_\theta}\}$ and pseudorandom numbers are introduced as session variational operators to ensure the communications dynamic. An adversary regards the prior session as random even if $\{S, U_\theta\}$ get corrupted, or the adversary obtains the PRNG algorithm. The current security compromises cannot correlate with the prior interrogations.

5 Formal Security Analysis with the Universal Composability Model

5.1 Preliminaries

The universal composability model specifies an approach for security proofs [28], and guarantees that the proofs will remain valid if the protocol is modularly composed with other protocols, and/or under arbitrary concurrent protocol executions. There is a real-world simulation, an ideal-world simulation, and a simulator Sim translating the protocol execution from the real-world to the ideal-world. Additionally, the Byzantine attack model is adopted for security analysis, and all the parties are modeled as probabilistic polynomial-time Turing machines (PPTs), and a PPT captures whatever is external to the protocol executions. The adversary controls message deliveries in all communication channels, and may perform malicious attacks (e.g., eavesdropping, forgery, and replay), and may also initiate new communications to interact with the legal parties.

In the real-world, let π be a real protocol, $\mathcal{P}_i$ ($i = \{1, \ldots, I\} \in \mathbb{N}^*$) be real parties, and $\mathcal{A}$ be a real-world adversary. In the ideal-world, let $\mathcal{F}$ be an ideal functionality, $\tilde{P}_i$ be dummy parties, and $\tilde{A}$ be an ideal-world adversary. $\mathcal{Z}$ is an interactive environment, and communicates with all entities except the ideal functionality $\mathcal{F}$. Ideal functionality acts as an uncorruptable trusted party to realize specific protocol functions.

Theorem 1. UC Security. *The probability, that $\mathcal{Z}$ distinguishes between an interaction of $\mathcal{A}$ with $\mathcal{P}_i$ and an interaction of $\tilde{A}$ with $\tilde{P}_i$, is at most negligible probability. We have that a real protocol π UC-realizes an ideal functionality $\mathcal{F}$, i.e.,* $\textsc{Ideal}_{F,\tilde{A},Z} \approx \textsc{Real}_{\pi,\mathcal{A},\mathcal{Z}}$.

The UC formalization of the SAPA includes the ideal-world model Ideal, and the real-world model Real.

- Ideal: Define two uncorrupted idea functionalities $\{\mathcal{F}_{\text{ACCESS}}, \mathcal{F}_{\text{SHARE}}\}$, a dummy party $\tilde{\mathcal{P}}$ (e.g., $\tilde{\mathcal{U}}_\theta$, $\tilde{\mathcal{S}}$, $\theta \in \{a, b\}$), and an ideal adversary $\tilde{\mathcal{A}}$. $\{\tilde{\mathcal{P}}, \tilde{\mathcal{A}}\}$ cannot establish direct communications. $\tilde{\mathcal{A}}$ can arbitrarily interact with $\mathcal{Z}$, and can corrupt any dummy party $\tilde{\mathcal{P}}$, but cannot modify the exchanged messages.
- Real: Define a real protocol π_{share} (run by a party $\mathcal{P}$ including $\mathcal{U}_\theta$ and $\mathcal{S}$) with a real adversary $\mathcal{A}$ and an environment $\mathcal{Z}$. Each real parties can

TABLE 2
Ideal Data Accessing Functionality: $\mathcal{F}_{\text{ACCESS}}$

Initialization:
- Upon input INITIALIZE at $\mathcal{P}$: If $\mathcal{P}$ is uncorrupted, and $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ is recorded, remove the existing $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$, generate a session identifier $sid_{\mathcal{P}_\theta}$, record and output $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ to $\mathcal{P}$. Else, output $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ to $\tilde{\mathcal{A}}$.

Message exchange:
- Upon input SEND from $\mathcal{P}$: If $\mathcal{P}$ is uncorrupted, and $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ is recorded, extract a message $m_{\mathcal{P}_\theta}$ to record $send(sid_{\mathcal{P}_\theta}, m_{\mathcal{P}_\theta}, \mathcal{P})$, and output $send(sid_{\mathcal{P}_\theta}, m_{\mathcal{P}_\theta}, \mathcal{P})$ to $\mathcal{P}$. Else, if $\mathcal{P}$ is corrupted and $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ is recorded, record and output $send(sid_{\mathcal{P}_\theta}, \mathcal{P})$ to $\mathcal{P}$. Else, ignore.
- Upon input RECEIVE from $\mathcal{U}_\theta$: If $\mathcal{U}_\theta$ is uncorrupted, and $init(sid_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ is recorded, record $rec(sid_{\mathcal{S}_\theta}, m_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$, and output $rec(sid_{\mathcal{U}_\theta}, m_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ to $\mathcal{U}_\theta$. Else, if $\mathcal{U}_\theta$ is corrupted and $init(sid_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ is recorded, record and output $rec(sid_{\mathcal{S}_\theta}, \mathcal{U}_\theta)$ to $\tilde{\mathcal{A}}$. Else, ignore.
- Upon input RECEIVE from $\mathcal{S}$: If $\mathcal{S}$ is uncorrupted, and $init(sid_{\mathcal{S}_\theta}, \mathcal{S})$ is recorded, record $rec(sid_{\mathcal{U}_\theta}, m_{\mathcal{S}_\theta}, \mathcal{S})$, and output $rec(sid_{\mathcal{S}_\theta}, m_{\mathcal{S}_\theta}, \mathcal{S})$ to $\mathcal{S}$. Else, if $\mathcal{S}$ is corrupted and $init(sid_{\mathcal{S}_\theta}, \mathcal{S})$ is recorded, record and output $rec(sid_{\mathcal{U}_\theta}, \mathcal{S})$ to $\tilde{\mathcal{A}}$. Else, ignore.

Nonce generation:
- Upon input GENERATE from $\mathcal{P}$: If $\mathcal{P}$ is uncorrupted, generate a random number $r_{\mathcal{P}_\theta}$, record $gen(r_{\mathcal{P}_\theta}, \mathcal{P})$, and output $gen(r_{\mathcal{P}_\theta})$ to $\mathcal{P}$. Else, output $gen(r_{\mathcal{P}_\theta}, \mathcal{P})$ to $\tilde{\mathcal{A}}$.

Access control:
- Upon input ACCESS from $\mathcal{U}_\theta$: If $\mathcal{U}_\theta$ is corrupted, ignore. Else, if $\{send(P_{\mathcal{U}_\theta}, \mathcal{S}),\ rec(P_{\mathcal{U}_\theta}, \mathcal{U}_\theta),\ local(A_{\mathcal{U}_\theta}, \mathcal{U}_\theta)\}$ are matched, output $valid(A_{\mathcal{U}_\theta}, P_{\mathcal{U}_\theta})$ and $access(D_{\mathcal{U}_\theta})$ to $\mathcal{U}_\theta$. Else, if $\{send(P_{\mathcal{U}_\theta}, \mathcal{S}),\ rec(P_{\mathcal{U}_\theta}, \mathcal{U}_\theta),\ local(A_{\mathcal{U}_\theta}, \mathcal{U}_\theta)\}$ are unmatched, output $invalid(A_{\mathcal{U}_\theta}, P_{\mathcal{U}_\theta})$ to $\mathcal{U}_\theta$. Else, record and output $access(D_{\mathcal{U}_\theta})$ to $\tilde{\mathcal{A}}$.

Adversary behaviors:
- Upon request FORWARD($\mathcal{P}$) from $\tilde{\mathcal{A}}$: If $\{send(sid_{\mathcal{P}_\theta}, m_{\mathcal{P}_\theta}, \mathcal{P}),\ rec(sid_{\mathcal{S}_\theta}, m_{\mathcal{U}_\theta}, \mathcal{U}_\theta)/rec(sid_{\mathcal{U}_\theta}, m_{\mathcal{S}_\theta}, \mathcal{S})\}$ are recorded, output $forward(sid_{\mathcal{P}_\theta}, m_{\mathcal{P}_\theta}, \mathcal{P})$ to $\mathcal{P}$.
- Upon request ACCEPT($\mathcal{P}$) from $\tilde{\mathcal{A}}$: If $init(sid_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ is recorded and $forward(sid_{\mathcal{U}_\theta}, m_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ is recorded, output $accept(\mathcal{U}_\theta)$ to $\mathcal{S}$. If $init(sid_{\mathcal{S}_\theta}, \mathcal{S})$ is recorded and $forward(sid_{\mathcal{S}_\theta}, m_{\mathcal{S}_\theta}, \mathcal{S})$ is recorded, output $accept(\mathcal{S})$ to $\mathcal{U}_\theta$. Else, ignore.
- Upon request FORGE($\mathcal{P}$)/REPLAY($\mathcal{P}$) from $\tilde{\mathcal{A}}$: If $access(D_{\mathcal{U}_\theta})$ is recorded and $\mathcal{U}_\theta$ is corrupted, output $accept(\mathcal{U}_\theta)$ to $\mathcal{S}$. Else, ignore.

TABLE 3
Ideal Authority Sharing Functionality: $\mathcal{F}_{\text{SHARE}}$

Activation:
- Upon input ACTIVATE at $\mathcal{P}$: If $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$ is recorded, remove the existing $init(sid_{\mathcal{P}_\theta}, \mathcal{P})$, and apply $\mathcal{F}_{\text{ACCESS}}$ to initialize $\mathcal{P}$. Else, directly apply $\mathcal{F}_{\text{ACCESS}}$ to initialize $\mathcal{P}$.

Access request:
- Upon input CHALLENGE($\mathcal{U}_b$) from $\mathcal{U}_a$: If $\mathcal{U}_a$ is corrupted, ignore. Else, if $chall(R^{\mathcal{U}_x}_{\mathcal{U}_a})$ is recorded, remove the existing $chall(R^{\mathcal{U}_x}_{\mathcal{U}_a})$, extract $R^{\mathcal{U}_b}_{\mathcal{U}_a}$, record and output $chall(R^{\mathcal{U}_b}_{\mathcal{U}_a})$ to $\mathcal{U}_a$. Else, record and output $chall(R^{\mathcal{U}_b}_{\mathcal{U}_a}, \mathcal{U}_a)$ to $\tilde{\mathcal{A}}$.
- Upon input CHALLENGE($\mathcal{U}_a$) from $\mathcal{U}_b$: If $\mathcal{U}_b$ is corrupted, ignore. Else, if $chall(R^{\mathcal{U}_x}_{\mathcal{U}_b})$ is recorded, remove the existing $chall(R^{\mathcal{U}_x}_{\mathcal{U}_b})$, extract $R^{\mathcal{U}_a}_{\mathcal{U}_b}$, record and output $chall(R^{\mathcal{U}_a}_{\mathcal{U}_b})$ to $\mathcal{U}_b$. Else, record and output $chall(R^{\mathcal{U}_a}_{\mathcal{U}_b}, \mathcal{U}_b)$ to $\tilde{\mathcal{A}}$.

Authority sharing:
- Upon input SHARE($\dot{D}_{\mathcal{U}_b}, \mathcal{U}_a$) from $\mathcal{U}_a$: If $\mathcal{U}_a$ is corrupted, ignore. Else, if $\{chall(R^{\mathcal{U}_b}_{\mathcal{U}_a}, \mathcal{U}_a),\ chall(R^{\mathcal{U}_a}_{\mathcal{U}_b}, \mathcal{U}_b)\}$ are recorded and matched, record and output $share(\dot{D}_{\mathcal{U}_b}, \mathcal{U}_a)$ to $\mathcal{U}_a$. Else, if $\{chall(R^{\mathcal{U}_x}_{\mathcal{U}_a}, \mathcal{U}_a),\ chall(R^{\mathcal{U}_x}_{\mathcal{U}_b}, \mathcal{U}_b)\}$ are not matched, record and output $share(D_{\mathcal{U}_a}, \mathcal{U}_a)$ to $\mathcal{U}_a$. Else, record and output $share(D_{null}, \mathcal{U}_a)$ to $\tilde{\mathcal{A}}$.
- Upon input SHARE($\dot{D}_{\mathcal{U}_a}, \mathcal{U}_b$) from $\mathcal{U}_b$: If $\mathcal{U}_b$ is corrupted, ignore. Else, if $\{chall(R^{\mathcal{U}_b}_{\mathcal{U}_a}, \mathcal{U}_a),\ chall(R^{\mathcal{U}_a}_{\mathcal{U}_b}, \mathcal{U}_b)\}$ are recorded and matched, record and output $share(\dot{D}_{\mathcal{U}_a}, \mathcal{U}_b)$ to $\mathcal{U}_b$. Else, if $\{chall(R^{\mathcal{U}_x}_{\mathcal{U}_a}, \mathcal{U}_a),\ chall(R^{\mathcal{U}_x}_{\mathcal{U}_b}, \mathcal{U}_b)\}$ are not matched, record and output $share(D_{null}, \mathcal{U}_b)$ to $\mathcal{U}_b$. Else, record and output $share(D_{null}, \mathcal{U}_b)$ to $\tilde{\mathcal{A}}$.

Adversary behaviors:
- Upon request LISTEN($\mathcal{U}_\theta$) from $\tilde{\mathcal{A}}$: If $chall(R^{\mathcal{U}_x}_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ is recorded and $\mathcal{U}_\theta$ is corrupted, output $listen(R^{\mathcal{U}_x}_{\mathcal{U}_\theta}, \mathcal{U}_\theta)$ to $\mathcal{U}_\theta$. Else, ignore.
- Upon request FORGE($\mathcal{U}_\theta$) or REPLAY($\mathcal{U}_\theta$) from $\tilde{\mathcal{A}}$: If $share(D_{null}, \mathcal{U}_a)$ is recorded and $\mathcal{U}_a$ is corrupted, output $share(\dot{D}_{\mathcal{U}_a}, \mathcal{U}_b)$ to $\mathcal{U}_b$. If $share(D_{null}, \mathcal{U}_b)$ is recorded and $\mathcal{U}_b$ is corrupted, output $share(\dot{D}_{\mathcal{U}_b}, \mathcal{U}_a)$ to $\mathcal{U}_a$. Else, ignore.

communicate with each other, and $\mathcal{A}$ can fully control the interconnections of $\mathcal{P}$ to obtain/modify the exchanged messages. During the protocol execution, $\mathcal{Z}$ is activated first, and dual session identifiers shared by all the involved parties reflects the protocol state.

5.2 Ideal Functionality

Definition 1. Functionality $\mathcal{F}_{\text{ACCESS}}$. *$\mathcal{F}_{\text{ACCESS}}$ is an incorruptible ideal data accessing functionality via available channels, as shown in Table 2.*

In $\mathcal{F}_{\text{ACCESS}}$, a party $\mathcal{P}$ (e.g., $\mathcal{U}_\theta$, $\mathcal{S}$) is initialized (via input INITIALIZE), and thereby initiates a new session along with generating dual session identifiers $\{sid_{\mathcal{U}_\theta}, sid_{\mathcal{S}_\theta}\}$. $\mathcal{P}$ follows the assigned protocol procedure to send (via input SEND) and receive (via input RECEIVE) messages. A random number $r_{\mathcal{P}_\theta}$ is generated by $\mathcal{P}$ for further computation (via input GENERATE). Data access control is realized by checking $\{send(.), rec(.), local(.)\}$ (via input ACCESS). If $\mathcal{P}$ is controlled by an ideal adversary $\tilde{\mathcal{A}}$, four types of behaviors may be performed: $\tilde{\mathcal{A}}$ may record the exchanged messages on listened channels, and may forward the intercepted messages to $\mathcal{P}$ (via request FORWARD); $\tilde{\mathcal{A}}$ may record the state of authentication between $\mathcal{U}_\theta$ and $\mathcal{S}$ to interfere in the normal verification (via request ACCEPT); $\tilde{\mathcal{A}}$ may impersonate an legal party to obtain the full state (via request FORGE), and may replay the formerly intercepted messages to involve the ongoing communications (via request REPLAY).

Definition 2. Functionality $\mathcal{F}_{\text{SHARE}}$. *$\mathcal{F}_{\text{SHARE}}$ is an incorruptible ideal authority sharing functionality, as shown in Table 3.*

$\mathcal{F}_{\text{SHARE}}$ is activated by $\mathcal{P}$ (via input ACTIVATE), and the initialization is performed via INITIALIZE of $\mathcal{F}_{\text{ACCESS}}$. The access request pointers $\{R^{\mathcal{U}_b}_{\mathcal{U}_a}, R^{\mathcal{U}_a}_{\mathcal{U}_b}\}$ are respectively published and challenged by $\{\mathcal{U}_a, \mathcal{U}_b\}$ to indicate their desires (via input CHALLENGE). The authority sharing between $\{\mathcal{U}_a, \mathcal{U}_b\}$ is realized, and the desired data fields $\{\dot{D}_{\mathcal{U}_b}, \dot{D}_{\mathcal{U}_a}\}$ are accordingly obtained by $\{\mathcal{U}_a, \mathcal{U}_b\}$ (via input SHARE). If $\mathcal{P}$ is controlled by an ideal adversary $\tilde{\mathcal{A}}$, $\tilde{\mathcal{A}}$ may detect the exchanged challenged access request pointer $R^{\mathcal{U}_x}_{\mathcal{U}_\theta}$ (via request LISTEN); $\tilde{\mathcal{A}}$ may record the request pointer to interfere in the normal authority sharing between $\mathcal{U}_a$ and $\mathcal{U}_b$ (via request FORGE/REPLAY).

In the UC model, $\mathcal{F}_{\text{ACCESS}}$ and $\mathcal{F}_{\text{SHARE}}$ formally define the basic components of the ideal-world simulation.

- *Party.* Party $\mathcal{P}$ refers to multiple users $\mathcal{U}_\theta$ (e.g., $\mathcal{U}_a$, $\mathcal{U}_b$), and a cloud server $\mathcal{S}$ involved in a session. Through a successful session execution, $\{\mathcal{U}_\theta, \mathcal{S}\}$ establish authentication and access control, and $\{\mathcal{U}_a, \mathcal{U}_b\}$

obtain each other's temp authorized data fields for data authority sharing.

- *Session identifier.* The session identifiers $sid_{\mathcal{U}_\theta}$ and $sid_{\mathcal{S}_\theta}$ are generated for initialization by the environment $\mathcal{Z}$. The ideal adversary $\tilde{\mathcal{A}}$ may control and corrupt the interactions between $\mathcal{U}_\theta$ and $\mathcal{S}$.
- *Access request pointer.* The access request pointer $R_{\mathcal{U}_\theta}^{\mathcal{U}_x}$ is applied to indicate $\mathcal{U}_\theta$'s access request on $\mathcal{U}_x$'s temp authorized data fields $\dot{D}_{\mathcal{U}_x}$.

5.3 Real Protocol π_{SHARE}

A real protocol π_{SHARE} is performed based on the ideal functionalities to realize $\mathcal{F}_{\text{SHARE}}$ in $\mathcal{F}_{\text{ACCESS}}$-hybrid model.

Upon input ACTIVATE($\mathcal{P}$) at $\mathcal{P}$ (e.g., $\mathcal{U}_\theta$, and $\mathcal{S}$), $\mathcal{P}$ is activated via $\mathcal{F}_{\text{SHARE}}$ to trigger a new session, in which INITIALIZE of $\mathcal{F}_{\text{ACCESS}}$ is applied for initialization and assignment. $\{init(sid_{\mathcal{U}_\theta},\mathcal{U}_\theta),\ init(sid_{\mathcal{S}_\theta},\mathcal{S})\}$ are respectively obtained by $\{\mathcal{U}_\theta, \mathcal{S}\}$. Message deliveries are accordingly performed by inputting SEND and RECEIVE. Upon input SEND from $\mathcal{U}_\theta$, $\mathcal{U}_\theta$ records and outputs $send(sid_{\mathcal{U}_\theta},\mathcal{U}_\theta)$ via $\mathcal{F}_{\text{ACCESS}}$. Upon input RECEIVE from $\mathcal{S}$, $\mathcal{S}$ obtains $rec(sid_{\mathcal{U}_\theta},\mathcal{S})$ via $\mathcal{F}_{\text{ACCESS}}$. Upon input GENERATE($\mathcal{S}$) from $\mathcal{S}$, $\mathcal{S}$ randomly chooses a random number $r_{\mathcal{S}_\theta}$ to output $gen(r_{\mathcal{S}_\theta})$ and to establish a ciphertext for access control. Upon input GENERATE($\mathcal{U}_\theta$) from $\mathcal{U}_\theta$, $\mathcal{U}_\theta$ generates a random number $r_{\mathcal{U}_\theta}$ for further checking the validity of $\{A_{\mathcal{U}_\theta}, P_{\mathcal{U}_\theta}\}$. Upon input ACCESS from $\mathcal{U}_\theta$, $\mathcal{U}_\theta$ checks whether $\{send(.), rec(.), local(.)\}$ are matched via $\mathcal{F}_{\text{ACCESS}}$. If it holds, output $valid(A_{\mathcal{U}_\theta}, P_{\mathcal{U}_\theta})$ is valid. Else, output $invalid(A_{\mathcal{U}_\theta}, P_{\mathcal{U}_\theta})$ and terminate the protocol. Upon input CHALLENGE($\mathcal{U}_x$) from $\mathcal{U}_\theta$, $\mathcal{U}_\theta$ generates an access request pointer $R_{\mathcal{U}_\theta}^{\mathcal{U}_x}$, and outputs $chall(R_{\mathcal{U}_\theta}^{\mathcal{U}_x})$ to $\mathcal{U}_x$. Upon input SEND from $\mathcal{U}_\theta$, $\mathcal{U}_\theta$ computes a message $m_{\mathcal{U}_\theta}$, records and outputs $send(m_{\mathcal{U}_\theta},\mathcal{U}_\theta)$ via $\mathcal{F}_{\text{ACCESS}}$, in which $R_{\mathcal{U}_\theta}^{\mathcal{U}_x}$ is wrapped in $m_{\mathcal{U}_\theta}$. Upon input RECEIVE from $\mathcal{S}$, $\mathcal{S}$ obtains $rec(m_{\mathcal{U}_\theta},\mathcal{S})$ for access request matching. Upon input SHARE($\dot{D}_{\mathcal{U}_b},\mathcal{U}_a$) and SHARE($\dot{D}_{\mathcal{U}_a},\mathcal{U}_b$) from $\{\mathcal{U}_a, \mathcal{U}_b\}$, $\mathcal{S}$ checks whether $\{chall(R_{\mathcal{U}_a}^{\mathcal{U}_b},\mathcal{U}_a),\ chall(R_{\mathcal{U}_b}^{\mathcal{U}_a},\mathcal{U}_b)\}$ are matched. If it holds, output $share(\dot{D}_{\mathcal{U}_b},\mathcal{U}_a)$ to $\mathcal{U}_a$ and $share(\dot{D}_{\mathcal{U}_a},\mathcal{U}_b)$ to $\mathcal{U}_b$ to achieve data sharing. Else, output $share(D_{null},\mathcal{U}_a)$ to $\mathcal{U}_a$ and $share(D_{null},\mathcal{U}_b)$ to $\mathcal{U}_b$ for regular data accessing.

5.4 Security Proof of π_{SHARE}

Theorem 3. *The protocol π_{SHARE} UC-realizes the ideal functionality $\mathcal{F}_{\text{SHARE}}$ in the $\mathcal{F}_{\text{ACCESS}}$-hybrid model.*

Proof: Let $\mathcal{A}$ be a real adversary that interacts with the parties running π_{SHARE} in the $\mathcal{F}_{\text{ACCESS}}$-hybrid model. Let $\tilde{\mathcal{A}}$ be an ideal adversary such that any environment $\mathcal{Z}$ cannot distinguish with a non-negligible probability whether it is interacting with $\mathcal{A}$ and π_{SHARE} in REAL or it is interacting with $\tilde{\mathcal{A}}$ and $\mathcal{F}_{\text{SHARE}}$ in IDEAL. It means that there is a simulator Sim that translates π_{SHARE} procedures into REAL such that these cannot be distinguished by $\mathcal{Z}$.

Construction of the ideal adversary $\tilde{\mathcal{A}}$: The ideal adversary $\tilde{\mathcal{A}}$ acts as Sim to run the simulated copies of $\mathcal{Z}$, $\mathcal{A}$, and $\mathcal{P}$. $\tilde{\mathcal{A}}$ correlates runs of π_{SHARE} from REAL into IDEAL: the interactions of $\mathcal{A}$ and $\mathcal{P}$ is corresponding to the interactions of $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{P}}$. The input of $\mathcal{Z}$ is forwarded to $\mathcal{A}$ as $\mathcal{A}$'s input, and the output of $\mathcal{A}$ (after running π_{SHARE}) is copied to $\tilde{\mathcal{A}}$ as $\tilde{\mathcal{A}}$'s output.

Simulating the party $\mathcal{P}$. $\mathcal{U}_\theta$ and $\mathcal{S}$ are activated and initialized by ACTIVATE and INITIALIZATION, and $\tilde{\mathcal{A}}$ simulates as $\mathcal{A}$ during interactions.

- Whenever $\tilde{\mathcal{A}}$ obtains $\{init(sid_{\mathcal{P}_\theta},\mathcal{P}),\ gen(r_{\mathcal{P}_\theta},\mathcal{P})\}$ from $\mathcal{F}_{\text{ACCESS}}$, $\tilde{\mathcal{A}}$ transmits the messages to $\mathcal{A}$.
- Whenever $\tilde{\mathcal{A}}$ obtains $\{rec(.),\ send(.)\}$ from $\mathcal{F}_{\text{ACCESS}}$, $\tilde{\mathcal{A}}$ transmits the messages to $\mathcal{A}$, and forwards $\mathcal{A}$'s response $forward(sid_{\mathcal{P}_\theta}, m_{\mathcal{P}_\theta}, \mathcal{P})$ to $\mathcal{F}_{\text{ACCESS}}$.
- Whenever $\tilde{\mathcal{A}}$ obtains $\{init(.),\ forward(.)\}$ from $\mathcal{F}_{\text{ACCESS}}$, $\mathcal{S}$ transmits the messages to $\mathcal{A}$, and forwards $\mathcal{A}$'s response $accept(\mathcal{P})$ to $\mathcal{F}_{\text{ACCESS}}$.
- Whenever $\tilde{\mathcal{A}}$ obtains $chall(R_{\mathcal{U}_\theta}^{\mathcal{U}_x},\mathcal{U}_\theta)$ from $\mathcal{F}_{\text{SHARE}}$, $\tilde{\mathcal{A}}$ transmits the message to $\mathcal{A}$, and forwards $\mathcal{A}$'s response $listen(R_{\mathcal{U}_\theta}^{\mathcal{U}_x},\mathcal{U}_\theta)$ to $\mathcal{F}_{\text{SHARE}}$.

Simulating the party corruption. Whenever $\mathcal{P}$ is corrupted by $\mathcal{A}$, thereby $\tilde{\mathcal{A}}$ corrupts the corresponding $\tilde{\mathcal{P}}$. $\tilde{\mathcal{A}}$ provides $\mathcal{A}$ with the corrupted parties' internal states.

- Whenever $\tilde{\mathcal{A}}$ obtains $access(D_{\mathcal{U}_\theta})$ from $\mathcal{F}_{\text{ACCESS}}$, $\tilde{\mathcal{A}}$ transmits the message $access(D_{\mathcal{U}_\theta})$ to $\mathcal{A}$, and forwards $\mathcal{A}$'s response $accept(\mathcal{P})$ to $\mathcal{F}_{\text{ACCESS}}$.
- Whenever $\tilde{\mathcal{A}}$ obtains $chall(R_{\mathcal{U}_\theta}^{\mathcal{U}_x},\mathcal{U}_\theta)$ from $\mathcal{F}_{\text{SHARE}}$, $\tilde{\mathcal{A}}$ transmits the message to $\mathcal{A}$, and forwards $\mathcal{A}$'s response $share(D_{null},\mathcal{U}_\theta)$ to $\mathcal{F}_{\text{SHARE}}$.

IDEAL *and* REAL *are indistinguishable*: Assume that $\{\mathcal{CS}, \mathcal{CU}_\theta\}$ respectively indicate the events that corruptions of $\{\mathcal{S}, \mathcal{U}\}$. $\mathcal{Z}$ invokes ACTIVATE and INITIALIZE to launch an interaction. The commands GENERATE and ACCESS are invoked to transmit $access(D_{\mathcal{U}_\theta})$ to $\tilde{\mathcal{A}}$, and $\mathcal{A}$ responds $accept(\mathcal{P})$ to $\tilde{\mathcal{A}}$. Thereafter, CHALLENGE and SHARE are invoked to transmit $share(R_{\mathcal{U}_\theta}^{\mathcal{U}_x},\mathcal{U}_\theta)$, and $\mathcal{A}$ responds $share(D_{null},\mathcal{U}_\theta)$ to $\tilde{\mathcal{A}}$. Note that $init(.)$ independently generates dual session identifiers $\{sid_{\mathcal{U}_\theta}, sid_{\mathcal{S}_\theta}\}$, and the simulations of REAL and IDEAL are consistent even though $\tilde{\mathcal{A}}$ may intervene to prevent the data access control and authority sharing in IDEAL. The pseudorandom number generator (introduced in $\{init(.), gen(.)\}$), and the collision-resistant hash function (introduced in $\{access(.), share(.)\}$) are applied to guarantee that the probability of the environment $\mathcal{Z}$ can distinguish the adversary's behaviors in IDEAL and REAL is at most negligible. The simulation is performed based on the fact that no matter the event $\mathcal{CS}$ or $\mathcal{CU}_\theta$ occurs or not, Therefore, π_{SHARE} UC-realizes the ideal functionality $\mathcal{F}_{\text{SHARE}}$ in the $\mathcal{F}_{\text{ACCESS}}$-hybrid model. □

6 CONCLUSION

In this work, we have identified a new privacy challenge during data accessing in the cloud computing to achieve privacy-preserving access authority sharing. Authentication is established to guarantee data confidentiality and data integrity. Data anonymity is achieved since the wrapped values are exchanged during transmission. User privacy is enhanced by anonymous access requests to privately inform the cloud server about the users' access

desires. Forward security is realized by the session identifiers to prevent the session correlation. It indicates that the proposed scheme is possibly applied for privacy preservation in cloud applications.

Acknowledgments

This work was funded by DNSLAB, China Internet Network Information Center, Beijing 100190, China.

References

[1] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," Nat'l Inst. of Standards and Technology, 2009.
[2] A. Mishra, R. Jain, and A. Durresi, "Cloud Computing: Networking and Communication Challenges," *IEEE Comm. Magazine*, vol. 50, no. 9, pp. 24-25, Sept. 2012.
[3] R. Moreno-Vozmediano, R.S. Montero, and I.M. Llorente, "Key Challenges in Cloud Computing to Enable the Future Internet of Services," *IEEE Internet Computing*, vol. 17, no. 4, pp. 18-25, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6203493, July/Aug. 2013.
[4] K. Hwang and D. Li, "Trusted Cloud Computing with Secure Resources and Data Coloring," *IEEE Internet Computing*, vol. 14, no. 5, pp. 14-22, Sept./Oct. 2010.
[5] J. Chen, Y. Wang, and X. Wang, "On-Demand Security Architecture for Cloud Computing," *Computer*, vol. 45, no. 7, pp. 73-78, 2012.
[6] Y. Zhu, H. Hu, G. Ahn, and M. Yu, "Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231-2244, Dec. 2012.
[7] H. Wang, "Proxy Provable Data Possession in Public Clouds," *IEEE Trans. Services Computing*, vol. 6, no. 4, pp. 551-559, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6357181, Oct.-Dec. 2012.
[8] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717-1726, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6311398, Sept. 2013.
[9] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, no. 5, pp. 847-859-25, May 2011.
[10] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," *IEEE Network*, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.
[11] L.A. Dunning and R. Kresman, "Privacy Preserving Data Sharing with Anonymous ID Assignment," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 2, pp. 402-413, Feb. 2013.
[12] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182-1191, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6374615, June 2013.
[13] S. Grzonkowski and P.M. Corcoran, "Sharing Cloud Services: User Authentication for Social Enhancement of Home Networking," *IEEE Trans. Consumer Electronics*, vol. 57, no. 3, pp. 1424-1432, Aug. 2011.
[14] M. Nabeel, N. Shang, and E. Bertino, "Privacy Preserving Policy Based Content Sharing in Public Clouds," *IEEE Trans. Knowledge and Data Eng.*, vol. 25, no. 11, pp. 2602-2614, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6298891, Nov. 2013.
[15] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," *IEEE Trans. Services Computing*, vol. 5, no. 2, pp. 220-232, Apr.-June 2012.
[16] S. Sundareswaran, A.C. Squicciarini, and D. Lin, "Ensuring Distributed Accountability for Data Sharing in the Cloud," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 4, pp. 556-568, July/Aug. 2012.
[17] Y. Tang, P.C. Lee, J.C.S. Lui, and R. Perlman, "Secure Overlay Cloud Storage with Access Control and Assured Deletion," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 6, pp. 903-916, Nov./Dec. 2012.
[18] Y. Zhu, H. Hu, G. Ahn, D. Huang, and S. Wang, "Towards Temporal Access Control in Cloud Computing," *Proc. IEEE INFOCOM*, pp. 2576-2580, Mar. 2012.
[19] S. Ruj, M. Stojmenovic, and A. Nayak, "Decentralized Access Control with Anonymous Authentication for Securing Data in Clouds," *IEEE Trans. Parallel and Distributed Systems*, vol. 25, no. 2, pp. 384-394, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6463404, Feb. 2014.
[20] R. Sánchez, F. Almenares, P. Arias, D. Díaz-Sánchez, and A. Marín, "Enhancing Privacy and Dynamic Federation in IdM for Consumer Cloud Computing," *IEEE Trans. Consumer Electronics*, vol. 58, no. 1, pp. 95-103, Feb. 2012.
[21] H. Zhuo, S. Zhong, and N. Yu, "A Privacy-Preserving Remote Data Integrity Checking Protocol with Data Dynamics and Public Verifiability," *IEEE Trans. Knowledge and Data Eng.*, vol. 23, no. 9, pp. 1432-1437, Sept. 2011.
[22] Y. Xiao, C. Lin, Y. Jiang, X. Chu, and F. Liu, "An Efficient Privacy-Preserving Publish-Subscribe Service Scheme for Cloud Computing," *Proc. IEEE GLOBECOM '10*, Dec. 2010.
[23] I.T. Lien, Y.H. Lin, J.R. Shieh, and J.L. Wu, "A Novel Privacy Preserving Location-Based Service Protocol with Secret Circular Shift for K-NN Search," *IEEE Trans. Information Forensics and Security*, vol. 8, no. 6, pp. 863-873, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6476681, June 2013.
[24] A. Barsoum and A. Hasan, "Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2375-2385, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6392165, Dec. 2013.
[25] H.Y. Lin and W.G. Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," *IEEE Trans. Parallel and Distributed Systems*, vol. 23, no. 6, pp. 995-1003, June 2012.
[26] J. Yu, P. Lu, G. Xue, and M. Li, "Towards Secure Multi-Keyword Top-k Retrieval over Encrypted Cloud Data," *IEEE Trans. Dependable and Secure Computing*, vol. 10, no. 4, pp. 239-250, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6425381, July/Aug. 2013.
[27] K.W. Park, J. Han, J.W. Chung, and K.H. Park, "THEMIS: A Mutually Verifiable Billing System for the Cloud Computing Environment," *IEEE Trans. Services Computing*, vol. 6, no. 3, pp. 300-313, http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6133267, July-Sept. 2013.
[28] R. Canetti, "Universally Composable Security: A New Paradigm for Cryptographic Protocols," *Proc. 42nd IEEE Symp. Foundations of Computer Science (FOCS '01)*, pp. 136-145, Oct. 2001.

Hong Liu is currently working toward the PhD degree at the School of Electronic and Information Engineering, Beihang University, China. She focuses on the security and privacy issues in radio frequency identification, vehicle-to-grid networks, and Internet of Things. Her research interests include authentication protocol design, and security formal modeling and analysis. She is a student member of the IEEE.

Huansheng Ning received the BS degree from Anhui University in 1996 and the PhD degree from Beihang University in 2001. He is a professor in the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. His current research interests include Internet of Things, aviation security, electromagnetic sensing and computing. He has published more than 50 papers in journals, international conferences/workshops. He is a senior member of the IEEE.

Qingxu Xiong received the PhD degree in electrical engineering from Peking University, Beijing, China, in 1994. From 1994 to 1997, he worked in the Information Engineering Department at the Beijing University of Posts and Telecommunications as a postdoctoral researcher. He is currently a professor in the School of Electrical and Information Engineering at the Beijing University of Aeronautics and Astronautics. His research interests include scheduling in optical and wireless networks, performance modeling of wireless networks, and satellite communication. He is a member of the IEEE.

Laurence T. Yang received the BE degree in computer science from Tsinghua University, China, and the PhD degree in computer science from the University of Victoria, Canada. He is a professor in the School of Computer Science and Technology at the Huazhong University of Science and Technology, China, and in the Department of Computer Science, St. Francis Xavier University, Canada. His research interests include parallel and distributed computing, and embedded and ubiquitous/pervasive computing. His research is supported by the National Sciences and Engineering Research Council and the Canada Foundation for Innovation. He is a member of the IEEE.