An Integrated Privacy Preserving Attribute Based Access Control Framework

Runhua Xu
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
runhua.xu@pitt.edu

James B.D. Joshi
School of Information Sciences
University of Pittsburgh
Pittsburgh, USA
jjoshi@pitt.edu

***Abstract*—Recent advances in IT have enabled many applications that generate/collect huge amounts of personal data. While these advances have made many personalized applications such as personalized user-centric healthcare possible there are significant system maintenance cost related to data management, and security and privacy issues that need to be first addressed. Although cloud computing presents a new paradigm that helps maintaining users aggregated information distributed in different Internet enabled applications in one place, it also introduces new challenges in security and privacy. In this paper, we propose an integrated user-centric (or an organization-centric) privacy preserving attribute based access control approach to protect the security and privacy of a user's(or the organization's) data stored by a cloud service provider. The proposed approach includes a novel privacy-preserving revocable ciphertext policy attribute-based encryption (PR-CP-ABE) scheme. We also propose an extended Path-ORAM protocol that addresses the access pattern privacy as users access the protected data on cloud. We present security and privacy analysis and compare the performance parameters with other existing approaches.**

***Keywords*-Access Control; Data Security; Privacy; Electronic Health Records; Attribute-based Encryption**

I. Introduction

Recent advances in IT have enabled many applications that generate/collect huge amounts of personal data. While these advances have made many personalized applications, such as personalized user-centric healthcare, possible, there are significant system maintenance cost related to data management, and security and privacy issues that need to be first addressed to ensure their successes. In many instances, organizations collect, store and use huge amounts of personal data.

Emerging technologies such as cloud computing provide better platforms for data storage and management [1]. In particular, cloud computing helps in maintaining users' or organizations' aggregated data distributed in different Internet enabled applications in one place [2]. However, they further introduce new security and privacy challenges. In particular, although data confidentiality can be achieved by encrypting data to be sored in the cloud, there are significant challenges with regards to providing fine-grained accesses to critical and privacy-sensitive data stored there. Besides the access control challenges for cloud, the outsourced data model attached to cloud computing also introduces new privacy attacks such as leakage of sensitive information through the patterns of accesses to the stored data [3]. As users and organizations adopt cloud services, integrated solutions to ensure security and privacy are critically needed.

To address the need for fine-grained access to data stored on the cloud various cryptographic access control mechanisms have been recently proposed. These approaches support access to encrypted data stored in the cloud at various levels of granularity. Among these, Ciphertext Policy Attribute based Encryption (CP-ABE) [4] provides better design for fine-grained access control. However, there are still several challenges before CP-ABE schemes can be employed in applications. For instance, original CP-ABE does not support *write* accesses and immediate *revocation* of rights at the attribute level. The access structure, which indicates authorized entities, in the CP-ABE schemes may contain some sensitive attributes (e.g., Social Security Number, affiliation, vocation, age and salary) that may disclose users privacy. Mechanisms that address these need to also ensure fully forward and backward security [5]. A common assumption is that cloud storage provider is *honest-but-curious*. That means we cannot prevent cloud storage providers from gathering information related to stored data and accesses while providing the services that they have agreed to. Although the outsourced sensitive data is encrypted, access pattern disclosure is possible. By using some basic information, cloud storage providers (CSPs) or attackers can analyze the access patterns to infer a good amount of sensitive information [3]. Various Oblivious RAM approaches have been proposed in the literature to address such access pattern privacy issues [6]–[10].

In this paper, we propose an integrated, privacy preserving user-centric (or an organization-centric) attribute based access control approach to protect the privacy and security of a users' (or the organizations') data stored by a CSP. The proposed approach includes a novel access control framework based on privacy-preserving revocable cipher-

text policy attribute-based encryption (PR-CP-ABE). It also includes an extended Path-ORAM protocol that addresses the access pattern privacy as users access the protected data on the cloud. We present analysis about security and privacy and compare the performance parameters with other existing approaches. We also show that our PR-CP-ABE scheme is secure against selectively Chosen Plaintext Attack (CPA) under the decisional parallel Bilinear Diffie-Hellman Exponent (pBDHE) assumption.

Here we gives an example of application scenario in the healthcare domain that we will follow in this paper. We assume a patient/user-centric health application that allows a patient/user to store and manage all his Electronic Health Records (EHRs) by storing them in a CSP. CSP is assumed to be *honest-but-curios*. Using our proposed framework, a patient stores his EHRs in cloud storage. Suppose that he lives in state Y and usually goes to hospital B. One day he travels to state X and goes to a different hospital A. He can easily provide read/write permission to physician M. When he comes back to state Y, he needs to revoke physician M's permission immediately to ensure further access restriction on his sensitive data. Moreover, he can provide/revoke read permission to/from a pharmacist for buying medicine in a pharmacy.

Note that while the example focuses on patient/user centric management of EHRs, we can generalize it to similar user-centric applications or other organization-centric applications employing cloud services; for instance, a similar hospital-centric application can be thought of where the hospital maintained data is stored in the cloud and the hospital needs to manage access to stored data to different users and stakeholders by considering various security and privacy issues.

Existing approaches mentioned earlier provide some parts of the solutions but do not provide an integrated framework that provides read/write access handling capability, immediate revocation at the attributed level, and access pattern privacy.

The key contributions of the proposed work are as follows:

- We propose a privacy-preserving revocable ciphertext-policy attribute-based encryption (PR-CP-ABE) scheme that supports immediate attribute revocation and prevents privacy leaks that may occur through access structure. Moreover, we use *Linear Secret Sharing Scheme (LSSS)* matrix as the access structure, which has been proven to be an expressive policy structure. To our best knowledge, we believe it is the first work that has integrated immediate attribute revocation and privacy-preserving access structure.
- We also propose an extended Path Oblivious RAM (ePath-ORAM) protocol that prevents privacy disclosure of access patterns. That is, a client can hide its data access patterns from an untrusted server in cloud storage applications. Moreover, our ePath-ORAM supports update of both access policies, and encrypted data, i.e., read/write access operations, which are not completely addressed in the literature.
- We present security proof of the PR-CP-ABE scheme. The proposed PR-CP-ABE scheme is proven to be secure against selectively Chosen Plaintext Attack (CPA) under the decisional parallel Bilinear Diffie-Hellman Exponent (pBDHE) assumption, as shown in Appendix A.

The rest of paper is organized as follows. In Section II, we present related work. Our proposed framework is described in Section III. We review some concepts and introduce our PR-CP-ABE construction in Section III-B. Discussion and analysis about our framework is in Section III-E. Finally, we conclude this paper in Section IV.

II. Related Work

Attribute-based Encryption scheme [11], proposed by Sahai and Waters, combines the access control function with encryption by specifying a particular access policy over the users' attributes, which facilitates dynamical control based on users' attribute information. Then Bethencourt et al. give the initial construction of CP-ABE [4], where access structure is associated with ciphertext and users' key is associated with their attributes. Thus, access policy is determined by the encrypting party. The CP-ABE scheme provides a new approach to outsource data in a cloud environment. Meanwhile, researchers have tried to make access policy more flexible. Three types of access structures have been proposed: *AND-gates*, *LSSS* matrix and *tree*. Note that Waters proposes the first LSSS matrix based CP-ABE and points out that its expressiveness is not lower than that of the tree structure [12].

Two kinds of privacy issues have been addressed in the literature. Hur [13] fixes the issue that a private key generator may disclose users' privacy because of their full privilege on users' private keys. Moreover, sensitive attribute information, which is also users' privacy information, in the access structure may be leaked. Recently, a series of CP-ABE schemes [14]–[16] supporting hidden policy has been proposed. However, the limitation of these schemes is their limited policy expressiveness by using *And-gate* access structure. Lai et al. present a CP-ABE scheme which supports policy hiding by inner product predicate encryption, which is proven *fully* secure rather than *selectively* secure. [15]. Moreover, based on LSSS matrix, they present another CP-ABE scheme [16] supporting partial hidden policy.

In earlier work related to revocation issues, researchers add expiration time to each attribute to achieve revocation. However, the solution does not support immediate revocations. Issues of scalability and security degradation in terms of backward and forward security still exist. Recently, researchers have proposed CP-ABE schemes supporting immediate attribute revocation. Hur and Xie et al. [13],

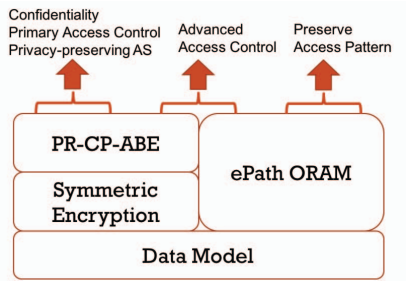

Figure 1. Overview of Access Control Framework

[17], [18] propose efficient attribute revocation schemes that utilize the secure two-party computation to generate private key for each user. Zu et al. [5] also propose a revocable CP-ABE schemes to archive efficient immediate revocation.

As users' access patterns can be disclosed [3], several schemes [6], [7], [10]have been proposed in the literature to avoid the analysis of user's access patterns, which are based on Oblivious RAM [19]. ORAM is a data protection scheme to make the access patterns independent of the inputs to the algorithm. Goodrich et al. [6] proposed practical oblivious storage, but attacker model in their scheme is not strong enough. Nabeel and Bertino [20] also present an approach that is based on two layers of encryption with broadcast encryption, but it needs a policy decomposition. Maffei et al. [10] give a framework based on ORAM with zero-knowledge proof and predict encryption to achieve the privacy and access control goals.

III. The Proposed Framework

A. Overview of Access Control Framework

Our proposed access control framework consist of four parts, as shown in Figure 1.

The base part is our outsourced data model, which defines the structure of outsourced data. Confidentiality of data is protected by symmetric encryption. We use PR-CP-ABE scheme to provide read access service to data by protecting private key of an existing symmetric encryption based mechanism. Moreover, the extended Path ORAM protocol focuses on privacy issues related to disclosure in access pattern. The integration of of ePath ORAM and PR-CP-ABE support advanced access control, such as write operation on data, access policy update, which are neglected in existing CP-ABE schemes.

B. PR-CP-ABE Construction

In this section, we present the proposed PR-CP-ABE scheme. We first present the standard definitions of various elements that we adopt from the existing literature.

1) Preliminaries:

Definition 1: Linear Secret Sharing Schemes [21]. A secret-sharing scheme Π over a set of parties $\mathcal{P}$ is called linear (over $\mathbb{Z}_p$), if

- The shares for each party form a vector over $\mathbb{Z}_p$.
- There exists a matrix, M with l rows and n columns, called the share-generating matrix for Π. For all $i = 1, ..., l$, the ith row of M, let the function ρ define the party labeling row i as $\rho(i)$. When we consider the column vector $v = (s, r_2, ..., r_n)$, where $s \in \mathbb{Z}_p$ is the secret to be shared, and $r_2, ..., r_n \in \mathbb{Z}_p$ are randomly chosen, then Mv is the vector of l shares of the secret s according to Π. The share $(Mv)_i$ belongs to party $\rho(i)$.

Suppose that Π is an LSSS for the access structure $\mathbb{A}$. Let $S \in \mathbb{A}$ be any authorized set, and let $I \subset \{1, 2, ..., l\}$ be defined as $I = i : \rho(i) \in S$. Then there exist some constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such that, if λ_i are valid shares of any secret s according to Π, then $\sum_{i \in I} \omega_i \lambda_i = s$. Constants ω_i can be found in time polynomial in the size of share-generating matrix M [21].

Definition 2: Decisional Parallel Bilinear Diffie-Hellman Exponent Assumption [12]. Choose a group $\mathbb{G}$ of prime order p according to the security parameter. Let $a, s, b_1, ..., b_q \in \mathbb{Z}_p$ be chosen at random and g be a generator of $\mathbb{G}$. If an adversary is given

$$\begin{aligned}\vec{y} =& g, g^s, g^a, ..., g^{a^q}, , g^{a^{q+2}}, ..., g^{a^{2q}},\\ & \{g^{sb_j}, g^{a/b_j}, ..., g^{a^q/b_j}, , g^{a^{q+2}/b_j}, ..., g^{a^{2q}/b_j}\}_{\forall_{1 \le j \le q}},\\ & \{g^{asb_k/b_j}, ..., g^{a^q sb_k/b_j}\}_{\forall_{1 \le k,j \le q}},\end{aligned}$$

it must remain hard to distinguish $e(g, g)^{a^{q+1}s} \in \mathbb{G}_T$ from a random element in $\mathbb{G}_T$. An algorithm β that outputs $z \in \{0, 1\}$ has advantage ϵ in solving q-parallel BDHE in $\mathbb{G}$ if

$$|\Pr[\beta(\vec{y}, T = e(g, g)^{a^{q+1}s}) = 0] - \Pr[\beta(\vec{y}, T = R) = 0]| \ge \epsilon$$

Composite Order Bilinear Groups is first introduced in [22]. Here are some useful properties. The order of bilinear groups is the product of two distinct primes. Let p, r be distinct primes, $\mathbb{G}$ and $\mathbb{G}_T$ be cyclic groups of order $N = pr$. Let $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a map that satisfies the following conditions:

- Bilinear: $\forall g, h \in \mathbb{G}, a, b \in Z_N, e(g^a, h^b) = e(g, h)^{ab}$.
- Non-degenerate: $\exists g \in \mathbb{G}$ such that $e(g, h)$ has order N in $\mathbb{G}_T$.

If the group operation in $\mathbb{G}$ and the bilinear map e are both efficiently computable, the multiplicative cyclic group $\mathbb{G}$ is a bilinear group. Note that the map e is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$. We use $\mathbb{G}_p$ and $\mathbb{G}_r$ to denote the subgroups of $\mathbb{G}$ with order p and r respectively. Note also that if $h_p \in \mathbb{G}_p$ and $h_r \in \mathbb{G}_r$ then $e(h_r, h_p) = 1$.

2) Our Model: The model of proposed PR-CP-ABE scheme has following five components:

Setup. The setup algorithm is run by the authority, which takes a security parameter 1^λ and outputs the public parameters PK and the master key MSK.

Encrypt. The encrypt algorithm is run by the data owner. It uses the public parameters PK, a message M, and an access structure $\mathbb{A} = (A, \rho, \tau)$ over the universe of attributes, and outputs the corresponding ciphertext.

KeyGen. The key generator algorithm is run by the authority. It takes the master key MSK, and a set of attributes S, and then outputs the secret keys sk_1, and delegation key sk_2 for user and cloud service provider, respectively.

Re-encrypt. The re-encrypt algorithm is run by the CSP. This algorithm takes as input the ciphertext and delegation key sk_2. Then it re-encrypts the ciphertext and introduces a new random element into the ciphertext component, which is associated with a set of revoked attributes.

Decrypt. The decrypt algorithm is run by a user accessing the data. It takes as input the re-encrypted ciphertext that contains a partial access structure (A, ρ) and a secret key sk_1 for the user's set of attributes S. If S satisfies the access structure, it will output message M, otherwise it will output a stop sign $\perp$.

3) Privacy-preserving Access Structure: In our PR-CP-ABE scheme, the user can only get the re-encrypted ciphertext from CSPs. The access policy is described as access structure (A, ρ, τ), where A is the $l \times n$ share-generating matrix, ρ is map from each row of A to an attribute name and τ is the value of the associated attribute.

In our construction, the attribute value τ is hidden and the other two parts are associated with the ciphertext. We believe that it is enough to prevent users' privacy disclosure; and here is an example to illustrate that. Suppose that a patient's EHRs are encrypted with an access policy as follow:

(ID: abc@xyz.com `OR` *SSN: 123-45-6789)* `OR` *(Affiliation: University Hospital* `AND` *Vocation: Physician).*

It means that either the owner with the given ID or SSN can access the EHRs, or the physician in University Hospital can access the EHRs. After the encryption, the access policy that is attached to the outsourced EHRs will be as follows:

*(ID: *)* `OR` *SSN: *)* `OR` *(Affiliation: *)* `AND` *Vocation: *).*

Even though, the attacker could find the attribute name, it does not make sense without attribute value.

4) The Detail Construction: Here we present the detail construction of PR-CP-ABE scheme:

Setup$(1^\lambda, U)$. The setup algorithm first runs $\mathcal{G}(1^\lambda)$ to obtain initial parameters $(p_1, p_2, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are cyclic groups with order $N = p_1 p_2$. Thus $\mathbb{G}_{p_1}, \mathbb{G}_{p_2}$ are the subgroups of $\mathbb{G}$, $\mathbb{G} = \mathbb{G}_{p_1} \times \mathbb{G}_{p_2}$. The attribute universe description is $U = \mathbb{Z}_N$.

Then the algorithm chooses $\{g, h, u_1, u_2, ..., u_n\} \in \mathbb{G}_{p_1}$, $\{\alpha_1, \alpha_2, a\} \in \mathbb{Z}_N$ randomly, and sets $\alpha = \alpha_1 + \alpha_2 \bmod N$, and $Z \in \mathbb{G}_{p_2}$. The public key is published as follows:

$$PK = (N, g, g^a, g^\alpha, e(g,g)^\alpha, \{u_i\}_{1 \le i \le n}, H = h \cdot Z).$$

The master key is published as

$$MSK = (h, \alpha_1, \alpha_2).$$

Encrypt$(PK, M, (A, \rho, \tau))$. The encryption algorithm takes the public key PK, a message M and an LSSS access structure (A, ρ, τ) as input. Here, A is an $l \times n$ LSSS matrix, ρ is a map function from each row of A to an attribute name, and $\tau = \{t_{\rho(i)}\}_{1 \le i \le l}$ is the set of values of attributes associated with $\rho(i)$.

Then encryption algorithm chooses two random vectors $\vec{v_1}, \vec{v_2}$ to share the encryption secrets s_1, s_2:

$$\vec{v_j}^T = (s_j, v_{j,2}, ..., v_{j,n})_{1 \le j \le 2}.$$

Let $\{Z_{j,1,i}, Z_{j,2,i}\}_{1 \le i \le l, 1 \le j \le 2} \in \mathbb{G}_{p_2}, \{r_{1,i}, r_{2,i}\}_{1 \le i \le l} \in \mathbb{Z}_N$ be chosen uniformly at random. The algorithm calculates the following ciphertext components:

$$\begin{aligned}
\widetilde{C}_1 =& M \cdot e(g,g)^{\alpha s_1}, \widetilde{C}_2 = e(g,g)^{\alpha s_2}, \\
C_j =& \{g^{s_j}\}_{1 \le j \le 2}, \\
C_{j,i} =& \{g^{a \vec{A}_i \vec{v_j}^T} (u_{\rho(i)}^{t_{\rho(i)}} H)^{r_{j,i}} \cdot Z_{j,1,i}\}_{1 \le i \le l, 1 \le j \le 2}, \\
D_{j,i} =& \{g^{r_{j,i}} \cdot Z_{j,2,i}\}_{1 \le i \le l, 1 \le j \le 2},
\end{aligned}$$

where $\vec{A}_i$ is the vector corresponding to the i-th row of A. Lastly, the output is the ciphertext CT as follows:

$$CT = (\{(A, \rho)\}, \{\widetilde{C}_j, C_j, \{C_{j,i}, D_{j,i}\}_{1 \le i \le l}\}_{1 \le j \le 2}$$

KeyGen(PK, MSK, S). The KeyGen algorithm takes public key PK, master key MSK and user's attribute set $S = \{s_i\}_{1 \le i \le n}$ as input and returns two secret keys, user's private key SK_1 and delegation key SK_2 for the cloud service provider. It randomly chooses $t \in \mathbb{Z}_N$ and $R, R', \{R_i\}_{1 \le i \le n} \in \mathbb{G}_{p_2}$. Thus users' secret keys are generated as $sk_1 = (k, k', \{k_i\}_{1 \le i \le n})$, where $k = R \cdot g^{\alpha_1} \cdot g^{at}, k' = R' \cdot g^t, k_i = R_i \cdot \{(u_i^{s_i})^t\}_{1 \le i \le n}$. The delegation key for cloud service provider is generated as $sk_2 = (g^{\alpha_2})$.

Re-encrypt(CT, sk_2). Re-encryption algorithm takes the initial ciphertext CT and delegation key as input to re-encrypt and returns new ciphertext $\widetilde{CT}$. There are two cases to consider:

- Suppose that there is no revoked attribute. The CSP selects a element $\theta \in \mathbb{Z}_N$ randomly. Then CSP calculates the ciphertext as follows:

$$\begin{aligned}
D =& (sk_2)^\theta = g^{\alpha_2 \theta}, \\
C_j^{'} =& \{C_j^{(1/\theta)}\}_{1 \le j \le 2}, \\
C_{j,i}^{'} =& \{C_{j,i} \cdot (u_{\rho(i)} H)^\theta\}_{1 \le i \le l, 1 \le j \le 2}, \\
D_{j,i}^{'} =& \{D_{j,i} \cdot g^\theta\}_{1 \le i \le l, 1 \le j \le 2},
\end{aligned}$$

  Then the re-encrypted ciphertext is computed as

$$\widetilde{CT} = \{D, \{\widetilde{C}_j, C_j, C_j^{'}, \{C_{j,i}^{'}, D_{j,i}^{'}\}_{1 \le i \le l}\}_{1 \le j \le 2}\}$$

- Suppose that there is a revoked attribute x. As in the previous case, it will select a random element $\theta, \theta_x \in$

$\mathbb{Z}_N$ to encrypt the delegation key and ciphertext, $D, C_1^{'}$, $C_2^{'}, C_{1,i}^{'}, C_{2,i}^{'}$. The components $D_{1,i}^{'}, D_{2,i}^{'}$ are generated as follows:

$$D_{j,i}^{'} = \left\{ \begin{array}{ll} D_{j,i} \cdot g^{\theta} & \text{if } \rho(i) \neq x \\ (D_{j,i} \cdot g^{\theta})^{1/\theta_x} & \text{if } \rho(i) = x \end{array} \right\}_{1 \leq j \leq 2, 1 \leq i \leq l}.$$

The re-encrypted ciphertext is computed as

$$\widetilde{CT} = \{D, \mathbb{C}, \{\widetilde{C}_j, C_j, C_j^{'}, \{C_{j,i}^{'}, D_{j,i}^{'}\}_{1 \leq i \leq l}\}_{1 \leq j \leq 2}\}.$$

Decrypt$(\widetilde{CT}, sk_1)$. The decryption algorithm takes a ciphertext $\widetilde{CT}$ and a secret key sk_1 for a set of attributes S as input. It first calculates $I_{A,\rho}$ from (A, ρ), where $I_{A,\rho}$ denotes the smallest subsets of $\{1, ..., l\}$ that satisfies (A, ρ). Then it checks if there exists an $\mathcal{I} \in I_{A,\rho}$ that satisfies the following equation:

$$\frac{\widetilde{C}_2 \cdot \prod_{i \in \mathcal{I}} e(C_{2,i}^{'}, k^{'})^{\omega_i}}{e(C_2^{'}, D) \cdot e(C_2, K) \cdot \prod_{i \in \mathcal{I}} e(D_{2,i}^{'}, k_i)^{\omega_i}} = 1,$$

where $\sum_{i \in \mathcal{I}} \omega_i \vec{A}_i = (1, 0, ..., 0)$. If the above equation test is not passed, it outputs stop sign $\perp$. Otherwise, it continues to compute:

$$T = \frac{\prod_{i \in \mathcal{I}} e(C_{1,i}^{'}, k^{'})^{\omega_i}}{\prod_{i \in \mathcal{I}} e(D_{1,i}^{'}, k_i)^{\omega_i}} = e(g, g)^{ats_1}.$$

Then the message M is recovered as follows:

$$M = \frac{\widetilde{C}_1 \cdot T}{e(C_1^{'}, D) \cdot e(C_1, k)}.$$

C. Outsourced Data Model

We define outsourced data $\mathcal{D}$ as follows to present the Path-ORAM based protocol:

Definition 3: Let k_δ be a randomly chosen session key and $Enc_{k_\delta}(data)$ be the ciphertext of $data$ produced by a symmetric encryption scheme with k_δ. Let $Enc_\gamma(k_\delta)$ be the ciphertext of the session key that is encrypted by our proposed PR-CP-ABE scheme. Then, we represent the outsourced data $\mathcal{D}$ as follows:

$$\mathcal{D} = (id, \mathcal{P}_r, \mathcal{P}_w, \mathcal{P}_o, Enc_{k_\delta}(data)),$$

where

$$\begin{aligned} \mathcal{P}_r &= (< A_r, \rho_r >, Enc_\gamma(k_\delta)), \\ \mathcal{P}_w &= (< A_w, \rho_w >, Enc_\gamma(s_w), s_w), \\ \mathcal{P}_o &= (< A_o, \rho_o >, Enc_\gamma(s_o), s_o). \end{aligned}$$

Here id is a unique identifier for the outsourced data in the cloud storage environment. $\mathcal{P}_i (i \in \{r, w, o\})$ is the component that is associated with the read, write and owner permissions, respectively. $< A_i, \rho_i > (i \in \{r, w, o\})$ represents privacy-preserving access structures associated with each permission type (i.e., $\mathcal{P}_i$'s policy).

For instance, if a user's attributes satisfy $< A_r, \rho_r >$, he can have read access to the data. If a user wants to update the data, he should prove that he has ability to decrypt $Enc_\gamma(s_w)$, which is associated with the write access policy $< A_w, \rho_w >$. Here s_w is the write permission related random seed that will be updated after each write operation. Similarly, $\mathcal{P}_o$ is component to verify the owner permission. Users with owner permission can update access policy of each part of $\mathcal{D}$.

Protocol 1 ePath-ORAM-Read(Client, Server)

1: Server $\xleftarrow{id, read}$ Client
2: Server: find data $\mathcal{D}$ = Path-ORAM(read, id, NULL)
3: Server $\xrightarrow{\mathcal{P}_r, Enc_{k_\delta}(data)}$ Client

Note: *id* is the identifier; *read* is the operate type.

Protocol 2 ePath-ORAM-Write(Client, Server)

1: Server $\xleftarrow{id, write}$ Client
2: Server: find data $\mathcal{D}$ = Path-ORAM(read, id, NULL)
3: Server $\xrightarrow{<A_w, \rho_w>, Enc_\gamma(s_w)}$ Client
4: Server $\xleftarrow{s_w^{'}}$ Client
5: Server: if $s_w == s_w^{'}$, continue; otherwise, cancel
6: Server $\xrightarrow{\mathcal{P}_r, Enc_{k_\delta}(data), s_w^{''} \in_R \mathbb{Z}}$ Client
7: Server $\xleftarrow{Enc_\gamma(k_\delta^{'}), Enc_{k_\delta^{'}}(data^{'}), Enc_\gamma(s_w^{''}), s_w^{''}}$ Client
8: Server: update $\mathcal{D}$ to $\mathcal{D}^{'}$, Path-ORAM(write, id, $\mathcal{D}^{'}$)

Note: *id* is the identifier; *write* is the operate type; $\mathcal{D}^{'}$ is the updated data.

D. Our ORAM Protocols

Here, we describe our ORAM protocol that is extended from the Path ORAM protocol proposed in [7], which concludes that Path ORAM is asymptotically better than the best known ORAM scheme with small client storage. To support data and policy update in the cloud storage, we extend it to achieve read/write control and protect privacy with regards to the access pattern disclosure. Our ORAM scheme contains three protocols in our access control framework: *ePath-ORAM-Read*, *ePath-ORAM-Write*, *ePath-ORAM-Owner*.

Before the specific description, here are some assumptions that we make. Cloud storage provider is *honest-but-curious*, which is a common assumption in cloud based applications, i.e., the cloud storage provider follows our protocol but seeks to gather additional information, which is regarded as a passive adversary activities. Meanwhile, the verification components, random seeds s_w and s_o (nonces), should be chosen randomly and updated after every interaction. We also assume that communication of new random seeds, $s_w^{''}, s_o^{''}$, is done through a secure channel.

The specific read and write protocols are shown in Protocol 1, 2, respectively. As outsourced data is protected by PR-CP-ABE scheme, the cloud storage provider (server) uses

the Path ORAM mechanism to store and manage a user's (or client's) data. When a client has a *read* request, the server will find $\mathcal{D}$ by Path ORAM mechanism and only send back components $\mathcal{P}_r$ and $Enc_{k_\delta}(data)$. If the server receives a *write* request, it will find data $\mathcal{D}$ and send back components $< A_w, \rho_w >, Enc_\gamma(s_w)$ to verify the client's decryption ability. If the user is verified, the server will update *write* permission seed with new random element $s_w^{''}$ and the user's data encrypted with new session key $k_\delta^{'}$. Then the server writes back updated $\mathcal{D}^{'}$ to the cloud storage using the Path ORAM mechanism.

Note that the *ePath-ORAM-Owner* protocol is similar to *ePath-ORAM-Write* protocol (hence not shown). The key difference is that the owner of data can update access policies in the data tuple $\mathcal{D}$ If the owner updates an access policy, he will be required to update the corresponding components $\mathcal{P}_i, (i \in \{r, w, o\})$ and $Enc_{k_\delta}(data)$. Moreover, to protect the user's privacy the published access structures are only a part of the original ones without attribute value τ. Thus, in our application example, the physician should cooperate with the patient (owner) to get original access structure $< A_w, \rho_w, \tau_w >$ to encrypt $s_w^{''}$, which is reasonable in the real scenario.

E. Discussion and Analysis

1) Tricks of PR-CP-ABE Construction: We use re-encryption technique in our construction to achieve revocation. In our construction, we use composite order bilinear group, rather than using the prime order bilinear group as in existing approaches, to setup the initial elements. By the orthogonal property of subgroup elements in composite order bilinear groups, we can introduce some random elements that correspond to attributes without any influence in decryption. Challenges here are related to introducing random element-pairs into ciphertext components and private key components, respectively, and designing the decryption formula to eliminate the impact of the random elements. Meanwhile, our ciphertext has two similar parts. The first part is only a ciphertext of the protected data, while the second part does not contain that. Here, the second part is used to help a user to decide which attribute set satisfies the access structure. Moreover, we define attribute universe description set in *Setup* algorithm, rather than using hash function; this can help improve the efficiency of our PR-CP-ABE scheme. The limitation is that the system should redo the setup again when new attributes are added into the application's access policy. However, if the required set of attributes are considered in the initialization step itself, this is not of much concern.

2) Protection Features: The key features supported by our access control framework include privacy-preservation, user-centric policy management, and privilege/policy updating. Ciphertext policy attribute based encryption is such a kind of scheme to support user-centric access control. For privilege updating, in CP-ABE schemes, it is easy to grant privileges, but hard to revoke them. Meanwhile, the access structure, which is attached to encrypted data, has the risk of disclosure of users' privacy. While revocation issue and access structure privacy have been separately tackled by other researchers, to our best knowledge, our proposed PR-CP-ABE work is the first work that integrates them, as shown in Table I. Note that the LSSS matrix access structure used in our scheme is also the most expressive access structure in CP-ABE field [12].

3) Correctness Proof of PR-CP-ABE: Note that the decryption step has two parts that are similar. Thus, we only give the proof of one of them, that is, for the recovery of the message M from the ciphertext. The proof of the other part is similar. First, calculate the value T as follows:

$$
\begin{aligned}
T &= \frac{\prod_{i\in\mathcal{I}} e(g^{a\vec{A}_i\vec{v_1}^T}(u_{\rho(i)}^{t_{\rho(i)}}H)^{r_{1,i}} \cdot Z_{1,1,i} \cdot (u_{\rho(i)}H)^{\theta}, g^t R')^{\omega_i}}{\prod_{i\in\mathcal{I}} e(g^{r_{1,i}} \cdot Z_{1,2,i} \cdot g^{\theta}, (u_i^{s_i})^t R_i)^{\omega_i}} \\
&= \frac{e(g,g)^{\sum at\vec{A}_i\vec{v_1}^T\omega_i} \prod_{i\in\mathcal{I}} e((u_{\rho(i)}^{t_{\rho(i)}}h)^{r_{1,i}} \cdot (u_{\rho(i)}h)^{\theta}, g^t)^{\omega_i}}{\prod_{i\in\mathcal{I}} e(g^{r_{1,i}} \cdot g^{\theta}, (u_i^{s_i}h)^t)^{\omega_i}} \\
&= e(g,g)^{\sum at\vec{A}_i\vec{v_1}^T\omega_i} = e(g,g)^{ats_1}
\end{aligned}
$$

Then we can recover the message M as follows:

$$
\begin{aligned}
\frac{\widetilde{C}_1 \cdot T}{e(C_1', D) \cdot e(C_1, K)} &= \frac{M \cdot e(g,g)^{\alpha s_1} \cdot e(g,g)^{ats_1}}{e(g^{s_1/k}, g^{\alpha_2 k}) \cdot e(g^{s_1}, g^{\alpha_1} g^{at} \cdot R)} \\
&= \frac{M \cdot e(g,g)^{\alpha s_1} \cdot e(g,g)^{ats_1}}{e(g,g)^{s_1(at+\alpha_1+\alpha_2)}} = M
\end{aligned}
$$

4) Performance Analysis: As the experimental results in [24] have shown, the time taken for encryption/decryption is in milliseconds, but time cost of the authority related activities is in seconds, i.e., communication time is the main performance cost. Thus, we focus on performance analysis in terms of communication cost.

We compare our scheme with previous schemes [5], [17], [23] in terms of communication cost. As shown in Table II, communication costs related to various schemes are briefly compared. As discussed in Section III-B2, the private keys and public parameters contribute to the communication costs of authority $\leftrightarrow$ user and authority $\leftrightarrow$ owner, respectively. The transmission of the ciphertexts and re-encrypted ciphertexts are the main communication costs in the cloud service provider $\leftrightarrow$ user and the cloud service provider $\leftrightarrow$ owner. As shown in Table II, the scheme in [5] has the best performance. However, our scheme is better than other two schemes and provides more protection features. Moreover, our scheme prevents privacy leaks in access structure when compared to [5].

5) Forward/Backward Secrecy: A secure scheme or protocol is said to be forward secrecy if compromise of long-term keys does not compromise past session keys. That is forward secrecy protects past ciphertext against future

Table I
COMPARISON OF KEY FEATURES

Schemes	Access Structure Type	Immediate Revocation	Privacy-preserving Access Structure
[5]	LSSS Matrix	Yes	No
[23]	And-gate	Yes	No
[17]	Tree-based	Yes	No
[16]	LSSS Matrix	No	Yes
[14]	And-gate	No	Yes
Ours	LSSS Matrix	Yes	Yes

Table II
COMPARISON OF COMMUNICATION COST

Entities	Our scheme	[5]	[23]	[17]
Authority $\leftrightarrow$ User	$(2+n_i)\|\mathbb{G}\|$	$(2+n_i)\|\mathbb{G}\|$	$(1+2n_i)\|\mathbb{G}\|$	$(1+2n_i)\|\mathbb{G}\|$
Authority $\leftrightarrow$ Owner	$(2+n_a)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(1+3n_a)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2\|\mathbb{G}\|+\|\mathbb{G}_T\|$
CSP $\leftrightarrow$ User	$(4m+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$(2m+3)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $+m\|\mathbb{Z}_p\|$	$(3m+2n_i)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$(3m+2n_i)\|\mathbb{G}\|+\|\mathbb{G}_T\|$ $+(m/2\|n_u\|+log(n_u+1))\|\mathbb{Z}_p\|$
CSP $\leftrightarrow$ Owner	$2((2m+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|)$	$(2m+1)\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3m\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$2m\|\mathbb{G}\|+(m+1)\|\mathbb{G}_T\|$

[1] Let $|\mathbb{G}|$, $|\mathbb{G}_T|$ and $|\mathbb{Z}_p|$ be the elements size in $\mathbb{G}, \mathbb{G}_T$ and $\mathbb{Z}_p$, respectively.
[2] Let n_i, n_u, n_a be the attributes number of user i, number of users and universal attributes number.
[3] Let m represent the attached attributes number.

compromises of secret keys. Obviously, if attributes of a user in our scheme have been revoked, the user's attributes will not satisfy the access policy in future. Because the user could not update the private key with random components, which is associated to the revoked attributes. Thus our scheme provides forward secrecy property. Backward secrecy property in our scheme is to be defined as that a additional user to a group is unable to decrypt ciphertext constructed prior before the user's introduction. For instance, a user joins in our scheme with attributes that satisfy the access structure associated with the previous ciphertext. Our scheme makes sure that the new joined user can not decrypt that ciphertext. Even though the user can request private keys that is corresponding to the attributes in the access structure, the random factors, k_x, k, in components (specifying in $D'_{j,i}$) of previous ciphertext are not corresponded to random factors in current private keys. Therefor, the backward secrecy in our scheme is also guaranteed.

6) Access Control: In our framework, the data is encrypted by symmetric encryption algorithm and the symmetric key will change when update operation is triggered. Thus, the confidentiality of data is assured. Meanwhile, the symmetric key is protected by the PR-CP-ABE scheme that provides access control function based on access policy. However, PR-CP-ABE does not distinguish and manage the read and write access for data. Thus, access control feature of PR-CP-ABE could be viewed as primary access control.

Moreover, our framework also provides advanced access control for data by integration of PR-CP-ABE and ePath ORAM. To achieve that, we define a new outsourced data model and propose ePath-ORAM protocols, to achieve write or owner privilege verification by checking the decryption ability of the requesting user based on write/owner access policy. Specifically, the data model contains three key parts: $\mathcal{P}_r, \mathcal{P}_w, \mathcal{P}_o$. $\mathcal{P}_r$ contains access policy to verify users' read permission. $\mathcal{P}_w, \mathcal{P}_o$ are used to support write/owner access for data. The difference between $\mathcal{P}_w$ and $\mathcal{P}_o$ is that $\mathcal{P}_w$ relates to updating data while $\mathcal{P}_o$ relates to updating access policy in the outsourced data. Only when the privilege of owner is verified, does the ePath-ORAM update the data including the access policies.

7) Privacy Preservation: As confidentiality of data is protected by encryption algorithms, the main privacy issue about data has been handled. However, the access policy in PR-CP-ABE and access pattern analysis become the only two main places where privacy disclosure may occur, which is neglected in this domain. In our proposed access control framework, we solve both of two privacy disclosures. To avoid privacy leak in access policy, the PR-CP-ABE strips the attribute value from the access structure when outsourcing the data. In terms of the access pattern disclosure, our framework solves it by achieving an extended path Oblivious RAM protocol that has been proved in previous research.

IV. CONCLUSION

In this paper, we have proposed a novel privacy-preserving attribute-based access control framework for sensitive data with new features like user-centric data and policy management, immediate privilege revocation, and privacy protection. We have shown that the proposed scheme satisfies the security and privacy requirements and has good performance in terms of communication cost. Meanwhile, the security proof shows that our system achieves CPA security under the decisional parallel Bilinear Diffie-Hellman Exponent

assumption. As future work, it can be implemented in real case and extended to apply to the mobile application domain.

V. Acknowledgment

This research work has been supported by the National Science Foundation grant DGE-1438809

References

[1] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.

[2] Daniel J Abadi. Data management in the cloud: limitations and opportunities. *IEEE Data Eng. Bull.*, 32(1):3–12, 2009.

[3] Mohammad Saiful Islam, Mehmet Kuzu, and Murat Kantarcioglu. Access pattern disclosure on searchable encryption: Ramification, attack and mitigation. *Ndss*, 2012.

[4] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In *IEEE Symposium on Security and Privacy, 2007. SP'07.*, pages 321–334. IEEE, 2007.

[5] Longhui Zu, Zhenhua Liu, and Juanjuan Li. New ciphertext-policy attribute-based encryption with efficient revocation. In *Computer and Information Technology (CIT), 2014 IEEE International Conference on*, pages 281–287. IEEE, 2014.

[6] Michael T Goodrich, Michael Mitzenmacher, Olga Ohrimenko, and Roberto Tamassia. Practical oblivious storage. In *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pages 13–24. ACM, 2012.

[7] Emil Stefanov, Marten Van Dijk, Elaine Shi, Christopher Fletcher, Ling Ren, Xiangyao Yu, and Srinivas Devadas. Path oram: An extremely simple oblivious ram protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 299–310. ACM, 2013.

[8] Daniel Apon, Jonathan Katz, Elaine Shi, and Aishwarya Thiruvengadam. Verifiable oblivious storage. In *Public-Key Cryptography–PKC 2014*, pages 131–148. Springer, 2014.

[9] Emil Stefanov and Elaine Shi. Multi-cloud oblivious storage. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 247–258. ACM, 2013.

[10] Matteo Maffei, Giulio Malavolta, Manuel Reinert, and Dominique Schröder. Privacy and access control for outsourced personal records. Technical report, Cryptology ePrint Archive, Report 2015/224, 2015, http://eprint. iacr. org, 2015.

[11] Amit Sahai and Brent Waters. Fuzzy identity-based encryption. In *Advances in Cryptology–EUROCRYPT 2005*, pages 457–473. Springer, 2005.

[12] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In *Public Key Cryptography–PKC 2011*, pages 53–70. Springer, 2011.

[13] Junbeom Hur. Improving security and efficiency in attribute-based data sharing. *Knowledge and Data Engineering, IEEE Transactions on*, 25(10):2271–2282, 2013.

[14] Xiaohui Li, Dawu Gu, Yanli Ren, Ning Ding, and Kan Yuan. Efficient ciphertext-policy attribute based encryption with hidden policy. In *Internet and Distributed Computing Systems*, pages 146–159. Springer, 2012.

[15] Junzuo Lai, Robert H Deng, and Yingjiu Li. Fully secure cipertext-policy hiding cp-abe. In *Information Security Practice and Experience*, pages 24–39. Springer, 2011.

[16] Junzuo Lai, Robert H Deng, and Yingjiu Li. Expressive cp-abe with partially hidden access structures. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pages 18–19. ACM, 2012.

[17] Junbeom Hur and Dong Kun Noh. Attribute-based access control with efficient revocation in data outsourcing systems. *Parallel and Distributed Systems, IEEE Transactions on*, 22(7):1214–1221, 2011.

[18] Xingxing Xie, Hua Ma, Jin Li, and Xiaofeng Chen. New ciphertext-policy attribute-based access control with efficient revocation. In *Information and Communication Technology*, pages 373–382. Springer, 2013.

[19] Oded Goldreich and Rafail Ostrovsky. Software protection and simulation on oblivious rams. *Journal of the ACM (JACM)*, 43(3):431–473, 1996.

[20] Mohamed Nabeel and Elisa Bertino. Privacy preserving delegated access control in public clouds. *Knowledge and Data Engineering, IEEE Transactions on*, 26(9):2268–2280, 2014.

[21] Amos Beimel. *Secure schemes for secret sharing and key distribution*. PhD thesis, Technion-Israel Institute of technology, Faculty of computer science, 1996.

[22] Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of cryptography*, pages 325–341. Springer, 2005.

[23] Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Attribute based data sharing with attribute revocation. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 261–270. ACM, 2010.

[24] Bo Lang, Runhua Xu, and Yawei Duan. Extending the ciphertext-policy attribute based encryption scheme for supporting flexible access control. In *Security and Cryptography (SECRYPT), 2013 International Conference on*, pages 1–11. IEEE, 2013.

[25] Ling Cheung and Calvin Newport. Provably secure ciphertext policy abe. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 456–465. ACM, 2007.

APPENDIX A.
SECURITY PROOF OF PR-CP-ABE

In the following we adopt the proof approaches used by Cheung and Newport [25], to prove that the proposed approach is secure for the attacker model. The security model for our PR-CP-ABE scheme is Indistinguishable Chosen-plaintext Attack (IND-CPA) game with selective attributes. This model is widely used in analyzing ciphertext policy attribute based encryption schemes [5], [12].

There are two roles in the attack game model: adversary $\mathcal{A}$ and simulator $\mathcal{B}$. The adversary $\mathcal{A}$ tries to break our scheme, while simulator $\mathcal{B}$ tries to solve the problem that is based on the computational complexity theory.

Init. $\mathcal{B}$ takes in our secure assumption $\vec{y}, T$. $\mathcal{A}$ prepares a challenge access policy $< A^*_{l^* \times n^*}, \tau^*, \rho^* >$, and sends a set of revoked attributes S^*_x to $\mathcal{B}$.

Setup. $\mathcal{B}$ chooses $\alpha^{'}, \alpha^{''} \in \mathbb{Z}_p$ randomly, and sets $\alpha_1 = \alpha^{'} + a^{q+1}, \alpha_2 = \alpha^{''}$. Let $\alpha = \alpha_1 + \alpha_2 = \alpha^{'} + a^{q+1} + \alpha^{''}$. Then for each attribute x, it chooses a corresponding element $z_x \in \mathbb{Z}_p$ randomly. $\mathcal{B}$ checks the map ρ^*, and if $\rho^*(i) = x$, $\mathcal{B}$ simulates the attribute-related component μ_x as follows:

$$\mu_x = g^{z_x} \prod_{i \in X} g^{a A^*_{i,1}/b_i} \cdot g^{a^2 A^*_{i,2}/b_i} \cdots g^{a^{n^*} A^*_{i,n^*}/b_i}.$$

Otherwise, $\mathcal{B}$ sets $\mu_x = g^{z_x}$, and set other public parameters randomly.

Phase I. In this phase, Simulator $\mathcal{B}$ simulates the private key according to $\mathcal{A}$'s key request for a attributes set S with the restriction that S does not satisfy access structure $< A^*_{l^* \times n^*}, \tau^*, \rho^* >$.

$\mathcal{B}$ selects a vector $\vec{\omega} = (\omega_1, ..., \omega_2) \in \mathbb{Z}_p^{n^*}$ such that $\omega_1 = -1$ and $\vec{\omega} \cdot A^*_i = 0$ for all $\rho^*(i) \in S$. According to the definition of *LSSS* access structure, the vector $\vec{\omega}$ exists. Then $\mathcal{B}$ selects random element $r \in \mathbb{Z}_p$ and defines t as follows:

$$t = r + \omega_1 a^q + \omega_2 a^{q-1} + \cdots + \omega_{n^*} a^{q-n^*+1}.$$

Lastly, $\mathcal{B}$ chooses random element $R, R^{'}, \{R_i\}_{1 \le i \le n} \in \mathbb{G}_{p_2}$ and constructs the private key as follows

$$K' = g^r \prod_{i=1}^{n^*} (g^{a^{q+1-i}})^{\omega_i} R', K = g^{\alpha^{'}} g^{ar} \prod_{i=2}^{n^*} (g^{a^{q+2-i}})^{\omega_i} R.$$

Note that according to the above simulation, $\mathcal{B}$ puts the component of q-parallel BDHE challenge, $g^{-a^{q+1}}$, into private key component g^{at} without any influence on original scheme, i.e., $\mathcal{A}$ would not aware such simulation.

For attribute-related components in private key, there are only two kind of cases: If the attribute is in the challenge access policy, $\mathcal{B}$ just simply sets $K_x = K'^{z_x}$. Otherwise, $\mathcal{B}$ select random element $R_x \in \mathbb{G}_{p_2}$ constructs K_x as follows:

$$K_x = (K')^{z_x} \prod_{i \in X} \prod_{j=1}^{n^*} (g^{(a^j/b_i)r} \prod_{k=1, k \ne j}^{n^*} (g^{a^{q+1+j-k}/b_i})^{\omega_k})^{M^*_{i,j}} R_x.$$

Note that, due to $\vec{\omega} \cdot A^*_i = 0$, the terms of g^{a^{q+1}/b_i} in the simulation would not affect the decrypted result when using these private keys.

Challenge. $\mathcal{A}$ can submit two any random two messages M_0 and M_1 with the equal length to $\mathcal{B}$. Then $\mathcal{B}$ flips a coin to get a random bit $\beta \in \{0, 1\}$ and chooses a random $\mu \in \mathbb{Z}_p$. For privacy preserving purpose, the ciphertext in our scheme constructs by two parts. The only difference is that whether the part contains component, which is associated to original message, or not. Here, we just give simulation on one of them. The other is similar. $\mathcal{B}$ simulates ciphertext as follows:

$$\widetilde{C^*} = \mathcal{M}_\beta \cdot T \cdot e(g^s, g^{\alpha^{'}}) \cdot e(g^s, g^{\alpha^{''}}),$$
$$C^* = g^s, D^{'*} = (g^\alpha)^{1/\mu}, C^{'*} = (g^s)^{1/\mu}.$$

Here $D^{'*}$ is the ciphertext of delegation key.

Then $\mathcal{B}$ chooses random elements $y^{'}_2, ..., y^{'}_{n^*} \in \mathbb{Z}_P$ and simulates secret sharing part as follows:

$$\vec{v} = (s, y^{'}_2 + sa, y^{'}_3 + sa^2, ..., y^{'}_{n^*} + sa^{n-1}) \in \mathbb{Z}_p^{n^*}$$

For the access structure part, $\mathcal{B}$ chooses random elements $r^{'}_1, ..., r^{'}_l$. If the attribute is not revoked, $\mathcal{B}$ selects random elements $H, Z \in \mathbb{G}_{p_2}$ and constructs the challenge ciphertext components as, $D'^*_i = \{g^{-sb_i} g^{-r^{'}_i} Z_{1,i}\}_{1 \le i \le l}$ Otherwise, $\mathcal{B}$ simulates the challenge component as follows: $D'^*_i = \{(g^{-r_i} g^{-sb_i})^{v_{\rho^*(j)}}\}_{1 \le i \le l}$. Note that $C^{'*}$ is the same as before.

Finally, $\mathcal{B}$ sends the following challenge to Adversary $\mathcal{A}$:

$$\widetilde{CT}^{'*} = \{D'^*, \widetilde{C^*}, C^*, C'^*, \{C'^*_i, D'^*_i\}_{1 \le i \le l}\}.$$

Phase II. Same as phases I.

Guess. $\mathcal{A}$ eventually outputs a guess $\beta^{'}$ of β. According to $\mathcal{A}$'s guess, $\mathcal{B}$ gives answer to q-parallel BDHE challenge. If $\beta = \beta'$, $\mathcal{B}$ guess that $T = e(g, g)^{a^{q+1} s}$, otherwise $\mathcal{B}$ guess that T is a random group element. We believe that our simulation is perfect. Here we give the probability analysis.

If T is a valid tuple in q-parallel BDHE challenge, the advantage probability of $\mathcal{B}$ is as follows:

$$\Pr[\mathcal{B}(\vec{y}, T = e(g, g)^{a^{q+1} s}) = 0] = 1/2 + Adv_{\mathcal{A}}.$$

Otherwise, T is just a random element, then the advantage probability of $\mathcal{B}$ is as follows:

$$\Pr[\mathcal{B}(\vec{y}, T = R) = 0] = 1/2.$$

Thus the total advantage probability of $\mathcal{B}$ is as follows:

$$\Pr[\mathcal{B}] = (1/2 + Adv_{\mathcal{A}}) \cdot 1/2 + 1/2 \cdot 1/2 = 1/2 + 1/2 \cdot Adv_{\mathcal{A}}.$$

That is, $\mathcal{B}$ can challenge the q-parallel BDHE game with non-negligible advantage $1/2 \cdot Adv_{\mathcal{A}}$. However, based on the security assumption, no polynomial time algorithm has a non-negligible advantage in solving the decisional q-parallel BDHE challenge, i.e., no polynomial time adversary has non-negligible advantage to break our system.