SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks 2016; **9:**2387–2396
Published online 7 July 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1507

RESEARCH ARTICLE

CP-ABE with outsourced decryption and directionally hidden policy

Zhiwei Wang[1,2]* and Wenyang Liu[2]

[1] Nanjing University of Information Science and Technology, Jiangsu 210044, China
[2] College of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China

ABSTRACT

Ciphertext-policy attribute-based encryption (CP-ABE) is a novel cryptographic primitive for access controlling. However, the existing CP-ABE schemes are very inefficient as the decryptions involve many expensive pairing operations. Another drawback is that the access policy itself may disclose some privacies of the users. In certain applications, access structures also should be protected. In this work, we propose a notion of *CP-ABE with outsourced decryption and directionally hidden policy*, which allows a semi-trusted proxy in the cloud to help a user decrypt a ciphertext, but the proxy cannot learn the plaintext and the access policy. We construct a concrete scheme from Waters' CP-ABE and prove its *security*, *verifiability*, and *directionally hidden policy*. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS
CP-ABE; outsourced decryption; verifiability; directionally hidden policy

***Correspondence**
Zhiwei Wang, College of Computer Science and Technology, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China.
E-mail: zhwwang@njupt.edu.cn

1. INTRODUCTION

For one to many encryptions [1], Sahai *et al*. [2] firstly proposed the concept of attribute-based encryption (ABE). Only when the user's key satisfies the attributes of ciphertext, the ciphertext can be decrypted successfully. There are two kinds of ABE systems: The first one is ciphertext-policy ABE (CP-ABE), where ciphertexts are encrypted with access structures and keys are extracted from attributes; the second one is key-policy ABE, where keys and ciphertexts are just the reverse.

The scheme of Sahai *et al*. [2] was limited to specify as threshold access policies [3] with one threshold gate. Then, Lewko *et al*. [4] used monotone access structure [5] to devise a CP-ABE and a key-policy ABE, which are proved secure in composite bilinear groups. In order to improve the expressiveness, some ABE systems make use of linear secret sharing scheme (LSSS) or Boolean formulas as access structure. Waters [6] employed LSSS matrix as access structure to design CP-ABE under noninteractive hard assumptions. In [7], Goyal *et al*. used Boolean formulas to construct a bounded CP-ABE scheme [8]. Pandit *et al*. [9] used minimal sets to describe general access structure in ABE systems.

However, many existing ABE schemes have an efficiency drawback that is the computational cost of decryption [10]. Green *et al*. [11] proposed a notion of *ABE with outsourced decryption*, which delegates laborious computation overhead to a proxy. Lai *et al*. [12] proposed an improved notion of *ABE with verifiable outsourced decryption*, which allows the user to verify the correctness of partially decrypted ciphertext carried out by the proxy. Recently, Baodong *et al*. [13] proposed an improved ABE scheme with verifiable outsourced decryption. Another drawback is that access policy is associated with a ciphertext that may leak lots of sensitive information of the user. One can obtain CP-ABE with fully hidden access policy from inner-product predicate encryption [14]. However, supporting access policies are expressed in conjunctive normal form (CNF) or disjunctive normal form DNF. Recently, Lai *et al*. [15] propose a CP-ABE scheme with *partially* hidden access policies[†] that can be expressed as an LSSS.

We try to combine the concept of *outsourced decryption* and *hidden policy* together and propose a new notion

[†] The notion of partially hidden policy means that only the attribute values can be hidden, and the attribute names are still explicit.

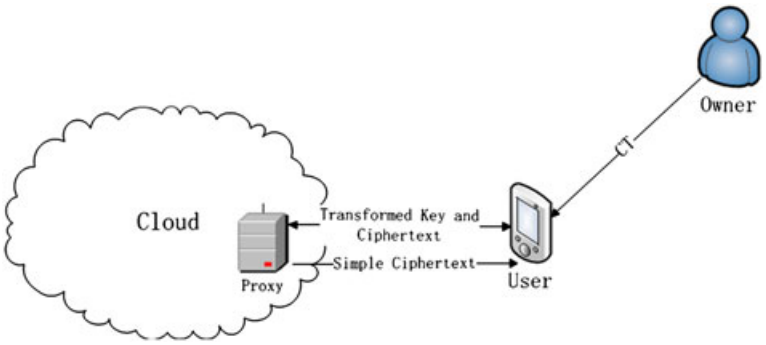

Figure 1. Ciphertext-policy attribute-based encryption with outsourced decryption and directionally hidden policy.

of *CP-ABE with outsourced decryption and directionally hidden policy* (CP-ABE-OD-DHP). The concept of *directionally hidden policy* is different from *fully hidden policy* and *partially hidden policy*, in which *the access policies are only hidden to a semi-trusted (that is, honest-but-curious) proxy and not include the ciphertext receivers.* In this system, a data owner sends a ciphertext *CT* to a user, who only has limitedly computational resource. Then, the user preprocesses the ciphertext and key and sends the transformed key and ciphertext without access policy to a proxy in cloud. The proxy generates a simple ciphertext and returns it to the user. Finally, the user recovers the plaintext only with lightly computational cost. Our scheme is very suitable for the cloud computing adoption framework [16], and our model can be described as Figure 1.

Our model can be used in many applications [17,18]. For example, in smart grid, users with smart devices (e.g., smart meters) usually have limitedly computational resource. They have to delegate laborious computation overhead to a storage center, which has highly computing power. But the storage center is semi-trusted because it might be operated by different electric utilities. Thus, it is curious about the access policies of the ciphertexts in the grid as well as the plain data of them for its own benefit. Disclosing some sensitive access policy of one electric utility to a storage center belonging to another electric utility may cause serious consequences. Thus, not only the plain data but also the access policies of the ciphertexts should be hidden from the storage center's view.

In this paper, we construct a CP-ABE-OD-DHP scheme based on Waters' CP-ABE [6]. We prove the *security*, *verifiability*, and *directionally hidden policy* of our scheme.

Organization. In Section 2, we review two complexity assumptions in composite group and access structure. In Section 3, we provide the definition and security model of CP-ABE-OD-DHP. In Section 4, we devise a basic CP-ABE scheme based on LSSS and proved its security based on Waters' CP-ABE scheme. In Section 5, we construct a concrete scheme and prove its security, verifiability, and directionally hidden policy. In Section 6, we conclude our paper.

2. BACKGROUND

2.1. Hardness assumptions

Bilinear group of composite order [19] is a product of two or more distinct primes. In our construction, we use the group order of $N = pq$, where p and q are two different primes. We denote these groups as $\mathbb{G}$ and admit an efficient bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}_T$'s order is the same as $\mathbb{G}$'s. Any element of $\mathbb{G}$ can be denoted as $g_1^{a_1} g_2^{a_2}$, where g_1 and g_2 are the generators of subgroup $\mathbb{G}_p$ and $\mathbb{G}_q$, respectively. $\mathbb{G}_p$ has the order p, and $\mathbb{G}_q$ has the order q. It is obvious that if an element of $\mathbb{G}_p$ is paired with an element in $\mathbb{G}_q$, the result of the pairing is an identity element. We denote g as a generator of $\mathbb{G}$. Then, g^q in $\mathbb{G}_p$ is a generator, and g^p in $\mathbb{G}_q$ is also a generator. If $u \in \mathbb{G}_p$ and $h \in \mathbb{G}_q$, then we have that $u = g^{qa}$ and $h = g^{pb}$, for some certainly a, b. So

$$e(u,h) = e\left(g^{qa}, g^{pb}\right) = e\left(g^a, g^b\right)^{pq} = 1_{\mathbb{G}_T}$$

The following assumptions, which has been analyzed in [4,20,21], have been used in many constructions [22–24].

Assumption 1. *Given* $\Theta = (N = pq, \mathbb{G}, \mathbb{G}_T, e)$*, if for any probabilistic polynomial time (PPT) algorithm* $\mathfrak{A}$*, the following equation holds*

$$|Pr[\mathfrak{A}(\Theta, g_1, X_1, Y_1Y_2, T_0) = 1] - Pr\left[\mathfrak{A}(\Theta, g_1, X_1, Y_1Y_2, T_1) = 1\right]| \leq \epsilon$$

where the probability ϵ *is negligible over the choice of* $g_1, X_1, Y_1 \in \mathbb{G}_p, Y_2 \in \mathbb{G}_p, T_0 \in \mathbb{G}, T_1 \in \mathbb{G}_p$.

Assumption 2. *Given* $\Theta = (N = pq, \mathbb{G}, \mathbb{G}_T, e)$*, if for any PPT algorithm* $\mathfrak{A}$*, the following equation holds*

$$|Pr[\mathfrak{A}(\Theta, g_1, g_1^{\alpha}X_2, g_1^{s}Y_2, Z_2, T_0) = 1] - Pr\left[\mathfrak{A}\left(\Theta, g_1, g_1^{\alpha}X_2, g_1^{s}Y_2, Z_2, T_1\right) = 1\right]| \leq \epsilon$$

DOI: 10.1002/sec

where the probability ϵ is negligible over the choice of $s, \alpha \in \mathbb{Z}_N, g_1 \in \mathbb{G}_p, X_2, Y_2, Z_2 \in \mathbb{G}_q, T_0 = e\left(g_1^{\alpha}, g_1^{s}\right)$, $T_1 \in \mathbb{G}_T$.

2.2. Access structure

We adapt our definitions that are given by [25].

Definition 1 (Access structure)**.** *Let $\{S_1, \ldots, S_n\}$ be a set of attributes. We call an authorized collection $\mathbb{A} \subset 2^{\{S_1,\ldots,S_n\}}$ monotone on the condition that $\forall B, C$, $B \in \mathbb{A}$, and $B \subseteq C$ then $C \in \mathbb{A}$. A monotone collection $\mathbb{A}$ is a monotone access structure, which is a non-empty set of subsets of $\{S_1, \ldots, S_n\}$. If the sets are not in $\mathbb{A}$, then they are unauthorized sets.*

The definition of LSSS can be seen in [25]. From the discussion in [25], each LSSS scheme Π for the access structure $\mathbb{A}$ can be used to *linear reconstruction*. Let $\mathcal{C} \in \mathbb{A}$ be any authorized set and let $I \subset \{1, \ldots, l\}$ be defined as $I = \{i : \rho(i) \in \mathcal{C}\}$. Then, there exists constants $\{\omega_i \in \mathbb{Z}_N\}_{i \in I}$ such that $\sum_{i \in I} \omega_i \lambda_i = \mu$, if $\{\lambda_i\}$ are valid shares of any μ in Π. These $\{\omega_i\}$ can be found in polynomial time.

3. DEFINITION OF CP-ABE-OD-DHP

3.1. Definition

A CP-ABE-OD-DHP is composed of seven PPT algorithms: **Setup**, **KeyGen**, **Encrypt**, **Preprocess**, **GenTK**, **Outsourced decrypt**, and **Light decrypt**. An authority uses the **Setup** algorithm to produce a master public/secret key pair and uses the **KeyGen** algorithm to produce a secret key for a user. A data owner uses the **Encrypt** algorithm to generate a ciphertext for an access structure and sends it to an user. When the user receives the ciphertext, he uses the **Preprocess** algorithm to produce a transformed ciphertext for hiding the access policy and uses the **GenTK** algorithm to produce a transformed secret key. Then, the user sends the transformed ciphertext and key to the proxy in cloud. The proxy uses the **Outsourced decrypt**, generates a simple ciphertext, and returns it to the user. Finally, the user uses the **Light decrypt** algorithm to recover the message from the simple ciphertext.

(1) **Setup** $(1^{\lambda}, \Sigma)$: The setup algorithm outputs a master public/secret key pair MPK/MSK.
(2) **KeyGen** $(MSK, \mathbb{S})$: It inputs an attribute set $\mathbb{S}$, and MSK and outputs a secret key $SK_{\mathbb{S}}$.
(3) **Encrypt** $(M, \mathbb{A})$: The encryption algorithm inputs a monotone access structure $\mathbb{A}$ and a message M for encryption and outputs a ciphertext $\bar{CT}$.
(4) **Preprocess** $(\bar{CT}, MPK)$: This algorithm takes as input a original ciphertext $\bar{CT}$ for an access structure $\mathbb{A}$ and the master public key MPK. It outputs the transformed ciphertext CT without access structure.
(5) **GenTK** $(\bar{CT}, MPK, SK_{\mathbb{S}})$: It inputs an original ciphertext $\bar{CT}$, MPK, and user's secret key $SK_{\mathbb{S}}$. It outputs a transformed secret key TK and a retrieving number $RK_{\mathbb{S}}$, if $\mathbb{S}$ satisfies $\mathbb{A}$.
(6) **Outsourced decrypt**(MPK, TK, CT): It inputs MPK, a transformed secret key TK and a transformed ciphertext CT. It outputs a partially decrypted ciphertext CT'.
(7) **Light decrypt** $(MPK, CT, CT', RK_{\mathbb{S}})$: It inputs MPK, the original ciphertext CT, a partially decrypted ciphertext CT', and the retrieving number $RK_{\mathbb{S}}$. It outputs an M.

3.2. Security model

In this section, we formally define the *security*, *verifiability*, and *directionally hidden policy* requirements of a CP-ABE-OD-DHP scheme. Informally, *security* ensures that an adversary cannot learn anything about the message; *verifiability* allows the user to check the correctness of outsourced decryption carried out by the proxy in cloud; *directionally hidden policy* ensures the proxy in cloud cannot learn the access policy.

The **security** of CP-ABE-OD-DHP can be described as a game.

Setup: The challenger $\mathfrak{C}$ generates MPK and MSK by using the **Setup** algorithm and sends MPK to $\mathfrak{A}$.

Queries 1: The adversary $\mathfrak{A}$ can issue four kinds of queries in this phase: **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries**. $\mathfrak{C}$ sets an empty table $\mathcal{T}$ and an empty set $\mathcal{D}$.
Key Queries: When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ sends it to $\mathfrak{A}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{C}$ sets $\mathcal{D} = \mathcal{D} \cup \mathbb{S}$ and responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.
Transformation Key Queries: When $\mathfrak{A}$ issues a transformation key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ returns $TK_{\mathbb{S}}$. Otherwise, $\mathfrak{C}$ runs the **KeyGen** algorithm to acquire $SK_{\mathbb{S}}$ and runs the **GenTK** algorithm to obtain $TK_{\mathbb{S}}$ and $RK_{\mathbb{S}}$. Then, it stores the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$ and returns $TK_{\mathbb{S}}$ to $\mathfrak{A}$.
Decrypt Queries: On input a $\mathbb{S}$ and a CT, $\mathfrak{C}$ runs the **KeyGen** algorithm to obtain $SK_{\mathbb{S}}$ and runs the **Decrypt** algorithm to obtain M. **Light Decrypt Queries:** On input a $\mathbb{S}$ and a pair CT, CT', $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ runs the **Light decrypt** algorithm to obtain M; otherwise, it outputs "failure."

Challenge: $\mathfrak{A}$ submits two distinct messages M_0, M_1 with the same length and an access structure $\mathbb{A}$ on the condition that any $\mathbb{S} \in \mathcal{D}$ cannot satisfy

$\mathbb{A}$. $\mathfrak{C}$ selects $\beta \in \{0,1\}$ and runs the **Encrypt** algorithm on M_β to acquire CT^* and sends CT^* to $\mathfrak{C}$.

Queries 2: $\mathfrak{A}$ continues to make **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries** adaptively, and $\mathfrak{C}$ returns the answer as Queries 1. However, $\mathfrak{A}$ cannot make key queries on any attributes set $\mathbb{S}$ that satisfies the access structure $\mathbb{A}$ being added to $\mathcal{D}$.

Outputs: Finally, $\mathfrak{A}$ outputs a guess bit β' for β.

The winning advantage of $\mathfrak{A}$ in this game is defined as $Pr[\beta - \beta'] - 1/2$.

Definition 2. *A CP-ABE-OD-DHP scheme is chosen ciphertext attacks secure if the advantages of all PPT adversaries in the aforementioned game are negligible.*

Chosen plaintext attacks (CPA) secure: *If we remove the decrypt queries from the aforementioned game, then we say a CP-ABE-OD-DHP scheme is CPA secure.*

Selective secure: *If the adversary submits the challenge access structure $\mathbb{A}^*$ before **Setup**, then we say A CP-ABE-OD-DHP scheme is selectively secure.*

The **Verifiability** of CP-ABE-OD-DHP also can be described as a game.

Setup: The challenger $\mathfrak{C}$ runs the **Setup** algorithm to generate *MPK* and *MSK* and sends *MPK* to $\mathfrak{A}$.

Queries 1: The adversary $\mathfrak{A}$ can issue four kinds of queries in this phase: **Key Queries** , **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries**. $\mathfrak{C}$ sets an empty table $\mathcal{T}$ and an empty set $\mathcal{D}$.

Key Queries: When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ sends it to $\mathfrak{A}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{C}$ sets $\mathcal{D} = \mathcal{D} \cup \mathbb{S}$ and responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.

Transformation Key Queries: When $\mathfrak{A}$ issues a transformation key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ returns $TK_{\mathbb{S}}$. Otherwise, $\mathfrak{C}$ runs the **KeyGen** algorithm to acquire $SK_{\mathbb{S}}$ and runs the **GenTK** algorithm to obtain $TK_{\mathbb{S}}$ and $RK_{\mathbb{S}}$. Then, it stores the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$ and returns $TK_{\mathbb{S}}$ to $\mathfrak{A}$.

Decrypt Queries: On input a $\mathbb{S}$ and a CT, $\mathfrak{C}$ runs the **KeyGen** algorithm to obtain $SK_{\mathbb{S}}$ and runs the **Decrypt** algorithm to obtain M.

Light Decrypt Queries: On input a $\mathbb{S}$ and a pair CT, CT', $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ runs the **Light decrypt** algorithm to obtain M; otherwise, it outputs "failure."

Challenge: $\mathfrak{A}$ submits a M^* and an $\mathbb{A}$. $\mathfrak{C}$ runs the **Encrypt** algorithm on M^* to obtain CT^* and sends CT^* to $\mathfrak{C}$.

Queries 2: $\mathfrak{A}$ continues to make **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries** adaptively, and $\mathfrak{C}$ returns the answer as Queries 1.

Outputs: Finally, $\mathfrak{A}$ outputs a $\mathbb{S}^*$ and a partially decrypted ciphertext CT'^*. We assume that the entry $< \mathbb{S}^*, SK_{\mathbb{S}^*}, TK_{\mathbb{S}^*}, RK_{\mathbb{S}^*} >$ exists in $\mathcal{T}$(if not, $\mathfrak{C}$ can generate it by responding the **Transformation Key Query**). If Lightdecrypt$(MPK, CT^*, CT'^*, RK_{\mathbb{S}^*})$ does not output "failure" and

$$\text{Lightdecrypt}\left(MPK, CT^*, CT'^*, RK_{\mathbb{S}^*}\right) \neq M^*$$

then $\mathfrak{A}$ wins the game.

Definition 3. *A CP-ABE-OD-DHP scheme is verifiable if the winning advantages of all PPT adversaries in the aforementioned game are negligible.*

The **Directionally hidden policy** of CP-ABE-OD-DHP also can be described as a game. The notion *directionally hidden policy* means that the access policy of ciphertext is only hidden to the proxy in cloud. Thus, in this game, the adversary $\mathfrak{A}$ is the proxy in cloud.

Setup: The challenger $\mathfrak{C}$ runs the **Setup** algorithm to generate *MPK* and *MSK* and sends *MPK* to $\mathfrak{A}$.

Queries 1: $\mathfrak{A}$ can issue four kinds of queries in this phase: **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries**. $\mathfrak{C}$ sets an empty table $\mathcal{T}$ and an empty set $\mathcal{D}$.

Key Queries: When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ sends it to $\mathfrak{A}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{C}$ sets $\mathcal{D} = \mathcal{D} \cup \mathbb{S}$ and responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.

Transformation Key Queries: When $\mathfrak{A}$ issues a transformation key query on attributes set $\mathbb{S}$, $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ returns $TK_{\mathbb{S}}$. Otherwise, $\mathfrak{C}$ runs the **KeyGen** algorithm to obtain $SK_{\mathbb{S}}$ and runs the **GenTK** algorithm to obtain $TK_{\mathbb{S}}$ and $RK_{\mathbb{S}}$. Then, it stores the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$ and returns $TK_{\mathbb{S}}$ to $\mathfrak{A}$.

Decrypt Queries: On input a $\mathbb{S}$ and a CT, $\mathfrak{C}$ runs the **KeyGen** algorithm to obtain $SK_{\mathbb{S}}$ and runs the **Decrypt** algorithm to obtain M.

Light Decrypt Queries: On input a $\mathbb{S}$ and a pair of CT, CT', $\mathfrak{C}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$, if the entry exists in $\mathcal{T}$, then $\mathfrak{C}$ runs the **Light decrypt** algorithm to obtain M; otherwise, it outputs "failure."

DOI: 10.1002/sec

Challenge: $\mathfrak{A}$ submits two messages M_0, M_1 and two access structures $\mathbb{A}_0$ and $\mathbb{A}_1$ on the condition that any $\mathbb{S} \in \mathcal{D}$ cannot satisfy $\mathbb{A}_0$ and $\mathbb{A}_1$. $\mathfrak{C}$ selects $\beta \in \{0,1\}$ and runs the **Encrypt** algorithm on M_β to acquire CT^* with $\mathbb{A}_\beta$ and sends CT^* to $\mathfrak{C}$.

Queries 2: $\mathfrak{A}$ continues to make **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries** adaptively, and $\mathfrak{C}$ returns the answer as Queries 1. However, $\mathfrak{A}$ cannot make key queries on any attributes set $\mathbb{S}$ that satisfies $\mathbb{A}_0$ and $\mathbb{A}_1$ being added to $\mathcal{D}$.

Outputs: $\mathfrak{A}$ outputs its guess β' for β.

The advantage of the adversary in this game is defined as $Pr[\beta - \beta'] - 1/2$.

Definition 4. *A CP-ABE-OD-DHP scheme is directionally hidden policy if the advantages of all PPT adversaries in the aforementioned game are negligible.*

4. BASIC CONSTRUCTION OF CP-ABE

4.1. Basic construction

Before proposing the CP-ABE-OD-DHP scheme, we design a new basic CP-ABE scheme that is based on Waters' most efficient CP-ABE scheme [6]. In our construction, we add a new algorithm **Preprocess** to Waters' scheme, which adds a verifiable part $\tilde{C}$ and some elements in $\mathbb{G}_q$ to Waters' ciphertext.

Setup $(1^\lambda, \Sigma)$: It inputs λ, which is a security parameter, a monotone universal attribute space Σ. It produces $\Theta = (N = pq, \mathbb{G}, \mathbb{G}_T, e)$, where p and q are two distinct prime numbers. Then, it selects random generators $g_1, h_1, \ldots, h_U \in G_p$ and $g_2 \in G_q$. It picks $a, \alpha \in \mathbb{Z}_p$. Finally, it chooses a hash function $H : \mathbb{G} \to \mathbb{Z}_{p_1}$ with collision resistance. The master public key is

$$MPK :< \Theta, g_1, h_1, \cdots, h_U, g_2, g_1^a, y = e(g_1, g_1)^\alpha, H >$$

and the master secret key is $MSK = g_1^\alpha$.

KeyGen $(MSK, \mathbb{S})$: It inputs a set of attributes $\mathbb{S}$ and MSK. It first chooses $t \in \mathbb{Z}_N$ and creates the private key as

$$\begin{aligned} SK_{\mathbb{S}} &=< K_1, K_2, \{K_x\}_{x\in\mathbb{S}} > \\ &= \left\langle g_1^\alpha g_1^{at}, g_1^t, \{h_x^t\}_{x\in\mathbb{S}} \right\rangle \end{aligned}$$

Encrypt (M, Π, MPK): It inputs MPK, an LSSS scheme $\Pi = (\mathcal{A}, \rho)$ and a message M to encrypt. Here, $\mathcal{A}$ is defined as an $l \times n$ matrix. The function ρ maps rows of $\mathcal{A}$ to attributes. The algorithm first randomly chooses$s \in \mathbb{Z}_N$ and a vector $\vec{v} = (s, v_2, \cdots, v_n) \in \mathbb{Z}_N^n$. For $i = 1$ to l, it computes $\lambda_i = \vec{v} \cdot \mathcal{A}_i$, where $\mathcal{A}_i$ is the vector that is the ith row of $\mathcal{A}$. Furthermore, it chooses random $r_1, \ldots, r_l, \in \mathbb{Z}_p$ and generates ciphertext $\bar{CT}$ as

$$\left\langle \bar{C} = M \cdot y^s, \bar{C}' = g_1^s, \left\{ \bar{C}_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i}, \bar{D}_i = g_1^{r_i} \right\}_{i\in[l]} \right\rangle$$

Preprocess $(\bar{CT}, MPK)$: The algorithm chooses random $\tau_1, \cdots, \tau_l, \delta \in \mathbb{Z}_q$. The converted ciphertext CT is

$$\begin{aligned} &\left\langle C = M \cdot y^s, \tilde{C} = g_1^{H(M)}, C' \right. \\ &\left. = g_1^s g_2^\delta, \left\{ C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, D_i = g_1^{r_i} \right\}_{i\in[l]} \right\rangle \end{aligned}$$

Decrypt (CT, SK): It inputs a CT for an LSSS scheme $\Pi = (\mathcal{A}, \rho)$ and an SK associated with $\mathbb{S}$. If $\mathbb{S} \in \mathbb{A}$ is an authorized set, then let $I \subset [l]$ be defined as $I = \{i : \rho(i) \in \mathbb{S}\}$. Then, it computes a set $\{\omega_i \in \mathbb{Z}_N\}_{i\in I}$ such that $\sum_{i\in I} \omega_i \lambda_i = s$, if $\{\lambda_i\}$ are valid shares according to $\mathcal{A}$. Then, the decryption algorithm computes

$$\frac{e(C', K_1)}{\prod_{i\in I}(e(C_i, K_2)e(D_i, K_{\rho(i)}))^{\omega_i}} = y^s$$

Finally, it can obtain the message M from C. If $\tilde{C} = g_1^{H(M)}$, it outputs M; otherwise, it outputs "invalid."

Decryption Correctness:

$$\begin{aligned} &\frac{e(C', K_1)}{\prod_{i\in I}(e(C_i, K_2)e(D_i, K_{\rho(i)}))^{\omega_i}} \\ &= \frac{e\left(g_1^s g_2^\delta, g_1^\alpha \cdot g_1^{at}\right)}{\prod_{i\in I}\left(e\left(g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, g_1^t\right) e\left(g_1^{r_i}, h_{\rho(i)}^t\right)\right)^{\omega_i}} \\ &= \frac{e(g_1, g_1)^{\alpha s} \cdot e(g_1, g_1)^{ats}}{\prod_{i\in I}\left(e\left(g_1^{a\lambda_i}, g_1^t\right) \cdot e\left(h_{\rho(i)}^{-r_i}, g_1^t\right) \cdot e\left(g_1^{r_i}, h_{\rho(i)}^t\right)\right)^{\omega_i}} \\ &= \frac{e(g_1, g_1)^{\alpha s} \cdot e(g_1, g_1)^{ats}}{e(g_1, g_1)^{at \cdot \sum_{i\in I} \lambda_i \omega_i}} \\ &= e(g_1, g_1)^{\alpha s} = y^s \end{aligned}$$

4.2. Security proof of basic construction

Theorem 1. *If Waters' CP-ABE scheme* [6] *is selectively CPA secure, then the basic construction is also selectively CPA secure.*

Proof. We prove this theorem by two games. The first game Game_0 is the original game of Waters' scheme. The second game Game_1 is the same as Game_0 except that the $\tilde{C}$ in challenge ciphertext is generated randomly and elements in $\mathbb{G}_q$ are added in C', C_i. We prove this theorem by two lemmas. The first lemma is that Game_0 and Game_1 are computationally indistinguishable, while the second lemma is that the advantage probability of adversary in

Game_1 is negligible. Thus, the advantage probability of adversary in Game_0 is also negligible, and Theorem 1 is proved secure. □

Lemma 1. *If Waters' CP-ABE scheme* [6] *is selectively CPA secure, then* Game_0 *and* Game_1 *are computationally indistinguishable.*

Proof. If there is a distinguisher $\mathfrak{A}$ can discriminate Game_0 and Game_1, then there exists an algorithm $\mathfrak{B}$ that can break Waters' CP-ABE scheme. Suppose $\mathfrak{C}$ is a challenger corresponding to $\mathfrak{B}$. $\mathfrak{B}$ runs $\mathfrak{A}$ as follows:

(1) **Setup:** $\mathfrak{A}$ gives $\mathfrak{B}$ its challenge access structure $\mathbb{A}^*$, and then, $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its challenge access structure. $\mathfrak{C}$ provides the public key $MPK' =< p, \mathbb{G}_p, g_1, g_1^a, e(g_1,g_1)^\alpha, \{h_i\}_{i\in[U]} >$. $\mathfrak{B}$ chooses a random prime number q with λ-bit and computes $N = pq$. $\mathfrak{B}$ generates the composite order group $\mathbb{G}$ with order N and the prime order group $\mathbb{G}_q$ with order q. Then, $\mathfrak{B}$ chooses a generator g_2 in $\mathbb{G}_q$ and a collision-resistant hash function $H : \mathbb{G}_p \to \mathbb{Z}_p$. Finally, $\mathfrak{B}$ sends the master public key $MPK =< N, \mathbb{G}, g_1, h_1, \ldots, h_U, g_2, g_1^a, y = e(g_1,g_1)^\alpha, H >$ to $\mathfrak{A}$.
(2) **Key Queries 1:** When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{B}$ sends it to $\mathfrak{C}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{B}$ responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.
(3) **Challenge:** When $\mathfrak{A}$ submits two message M_0 and M_1 (distinct messages but with equal length) to $\mathfrak{B}$, $\mathfrak{B}$ chooses a bit $\beta \in \{0,1\}$ and sends M_0 and M_1 to $\mathfrak{C}$. Then, $\mathfrak{C}$ selects a bit $\mu \in \{0,1\}$ and encrypts M_μ under MPK' and $\mathbb{A}^*$ by using the encrypt algorithm of Waters' scheme and sends $CT'^* =< C, C', \{C_i, D_i\}_{i\in[l]} >$ to $\mathfrak{B}$. $\mathfrak{B}$ randomly chooses $\delta, \tau_1, \ldots, \tau_l \in \mathbb{Z}_q$ and responses $CT^* =< C, \tilde{C} = g_1^{H(M_\beta)}, C' \cdot g_2^\delta, \{C_i \cdot g_2^{\tau_i}, D_i\}_{i\in[l]} >$.
(4) **Key Queries 2:** $\mathfrak{A}$ makes key queries adaptively, and $\mathfrak{B}$ returns the answer as **Key Queries 1**.
(5) **Outputs:** $\mathfrak{A}$ outputs β' to $\mathfrak{B}$, and $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its guess to μ.

If $\beta = \mu$, then $\mathfrak{B}$ has simulated Game_0. Otherwise, it has simulated Game_0. So if $\mathfrak{A}$ can distinguish Game_0 and Game_1, then $\mathfrak{B}$ can break Waters' CP-ABE scheme. □

Lemma 2. *If Waters' CP-ABE scheme* [6] *is selectively CPA secure, then the advantage of adversary* $\mathfrak{A}$ *in* Game_1 *is negligible.*

Proof. If an adversary $\mathfrak{A}$ can win Game_1, then there exists an algorithm $\mathfrak{B}$ that can break Waters' CP-ABE scheme. Suppose $\mathfrak{C}$ is a challenger corresponding to $\mathfrak{B}$. $\mathfrak{B}$ runs $\mathfrak{A}$ as follows:

(1) **Setup:** $\mathfrak{A}$ gives $\mathfrak{B}$ its challenge access structure $\mathbb{A}^*$, and then, $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its challenge access structure. $\mathfrak{C}$ provides the public key $MPK' =< p, \mathbb{G}_p, g_1, g_1^a, e(g_1,g_1)^\alpha, \{h_i\}_{i\in[U]} >$. $\mathfrak{B}$ chooses a random prime number q with λ-bit and computes $N = pq$. $\mathfrak{B}$ generates the composite order group $\mathbb{G}$ with order N, and the prime order group $\mathbb{G}_q$ with order q. Then, $\mathfrak{B}$ chooses a generator g_2 in $\mathbb{G}_q$ and a collision-resistant hash function $H : \mathbb{G}_p \to \mathbb{Z}_p$. Finally, $\mathfrak{B}$ sends the master public key $MPK =< N, \mathbb{G}, g_1, h_1, \ldots, h_U, g_2, g_1^a, y = e(g_1,g_1)^\alpha, H >$ to $\mathfrak{A}$.
(2) **Key Queries 1:** When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{B}$ sends it to $\mathfrak{C}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{B}$ responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.
(3) **Challenge:** The adversary $\mathfrak{A}$ submits two messages M_0 and M_1 to $\mathfrak{B}$(equal length), and $\mathfrak{B}$ sends M_0 and M_1 to $\mathfrak{C}$. Then, $\mathfrak{C}$ selects $\mu \in \{0,1\}$ and encrypts M_μ under MPK' and $\mathbb{A}^*$ by using the encrypt algorithm of Waters and sends $CT'^* =< C, C', \{C_i, D_i\}_{i\in[l]} >$ to $\mathfrak{B}$. $\mathfrak{B}$ randomly chooses $\delta, \tau_1, \ldots, \tau_l \in \mathbb{Z}_q$ and a random element $\hat{C} \in \mathbb{G}_p$ and responses $CT^* =< C, \tilde{C} = \hat{C}, C' \cdot g_2^\delta, \{C_i \cdot g_2^{\tau_i}, D_i\}_{i\in[l]} >$.
(4) **Key Queries 2:** $\mathfrak{A}$ makes key queries adaptively and $\mathfrak{B}$ returns the answer as **Key Queries 1**.
(5) **Outputs:** $\mathfrak{A}$ outputs β' to $\mathfrak{B}$, and $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its guess to μ.

Obviously, $\mathfrak{B}$ has properly simulated Game_1. So, if $\mathfrak{A}$ can win Game_1, then $\mathfrak{B}$ can break Waters' CP-ABE scheme with non-negligible advantage. □

5. CP-ABE-OD-DHP SCHEME

5.1. Construction of CP-ABE-OD-DHP

Based on our basic construction, we construct a new scheme, which has two advantages. The first one is that this scheme can realize verifiable outsourced decryption, while the second one is directionally hidden policy to the proxy in cloud. The **Setup**, **KeyGen**, **Encrypt**, and **Preprocess** algorithms are the same as our basic construction except that we add $e(g_2, g_2)$ to master public key MPK. We now consider the followed algorithms:

GenTK $(MPK, SK_{\mathbb{S}}, \tilde{CT})$: It inputs a master public key MPK, an attribute-based secret key $SK_{\mathbb{S}} =< K_1, K_2, \{K_x\}_{x\in\mathbb{S}} >$, and an original ciphertext $\tilde{CT}$ with $\mathbb{A}$. If $\mathbb{S} \in \mathbb{A}$ is an authorized set, then let $I \subset [l]$ be defined as $I = \{i : \rho(i) \in \mathbb{S}\}$. Then, it computes a set $\{\omega_i \in \mathbb{Z}_N\}_{i\in I}$ such that $\sum_{i\in I} \omega_i \lambda_i = s$, if $\{\lambda_i\}$ are valid shares according to matrix $\mathcal{A}$ in Π. It chooses a random number $z, d \in \mathbb{Z}_q$ and outputs the transformed key $TK =< K_1' = K_1 \cdot g_2^z, K_2' = K_2 \cdot g_2^d, \{(K_{\rho(i)}, \omega_i)\}_{i\in I}, I >$, and the retrieving number $RK_{\mathbb{S}} = z\delta - d\sum_{i\in I} \tau_i$.‡

Outsourced decrypt (MPK, TK, CT): It inputs the master public key MSK, a transformed key $TK =< K_1', K_2', \{(K_{\rho(i)}, \omega_i)\}_{i\in I}, I >$, and a ciphertext

‡ δ and τ_i are the same as our basic construction.

DOI: 10.1002/sec

$CT =< C, \tilde{C}, C', \{C_i, D_i\}_{i\in[l]} >$ without access structure. Then, it computes

$$T = \frac{e\left(C', K_1'\right)}{\prod_{i\in I}\left(e\left(C_i, K_2'\right) e\left(D_i, K_{\rho(i)}\right)\right)^{\omega_i}} = e(g_1, g_1)^{\alpha s} e(g_2, g_2)^{z\delta - d\sum_{i\in I}\tau_i}$$

and outputs the transformed ciphertext as $CT' =< C', \tilde{C}', T >$.

Light decrypt $(MPK, CT, CT', RK_{\mathbb{S}})$: It inputs a master public key MPK, a ciphertext $CT =< C, \tilde{C}, C', \{C_i, D_i\}_{i\in[l]} >$, a transformed ciphertext $CT' < C', \tilde{C}', T >$, and a retrieving number $RK_{\mathbb{S}}$ for an attribute set $\mathbb{S}_k$. If $C \neq C'$ or $\tilde{C} \neq \tilde{C}'$, it outputs "failure." Otherwise, it computes $M = \frac{C\cdot e(g_2,g_2)^{RK_{\mathbb{S}}}}{T}$.§ If $g_1^{H(M)} = \tilde{C}'$, then it outputs M; otherwise, it outputs "invalid."

In our scheme, when the proxy runs the **Outsourced decrypt** algorithm, because the ciphertext is blinded with elements in $\mathbb{G}_q$, the proxy cannot know the access policy and cannot use dictionary attack to guess the access policy. When a user runs the **Light decrypt** algorithm to reveal the message, it only needs three exponentiations. In **Preprocess** and **GenTK** algorithms, it requires three exponentiations and two exponentiations, respectively. This computational cost (eight exponentiations in total) is far less than running the **Decrypt** algorithm to recover the message in the original scheme. The use of $\tilde{C}$ in ciphertext to **verify** whether the transformation carried out by the proxy in cloud is correct.

5.2. Security proof

Theorem 2. *If the basic construction is selectively CPA secure, then the aforementioned scheme is also selectively CPA secure.*

Proof. If an adversary $\mathfrak{A}$ can break the aforementioned scheme in selectively CPA-secure model, then there exists an algorithm $\mathfrak{B}$ that can break the basic construction in selectively CPA-secure model. Suppose $\mathfrak{C}$ is a challenger corresponding to $\mathfrak{B}$. $\mathfrak{B}$ runs $\mathfrak{A}$ as follows:

(1) **Setup:** $\mathfrak{A}$ gives $\mathfrak{B}$ its challenge access structure $\mathbb{A}^*$, and then, $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its challenge access structure. $\mathfrak{C}$ provides the public key $MPK' =< p, \mathbb{G}_p, g_1, g_1^a, e(g_1, g_1)^{\alpha}, \{h_i\}_{i\in[U]} >$. $\mathfrak{B}$ chooses a random prime number q with λ-bit and computes $N = pq$. $\mathfrak{B}$ generates the composite order group $\mathbb{G}$ with order N and the prime order group $\mathbb{G}_q$ with order q. Then, $\mathfrak{B}$ chooses a generator g_2 in $\mathbb{G}_q$ and a collision-resistant hash function $H : \mathbb{G}_p \to \mathbb{Z}_p$. Finally, $\mathfrak{B}$ sends the master public key $MPK =< N, \mathbb{G}, g_1, h_1, \cdots, h_U, g_2, g_1^a, y = e(g_1, g_1)^{\alpha}, H >$ of basic construction to $\mathfrak{A}$.

(2) **Queries 1:** $\mathfrak{A}$ can issue two kinds of queries in this phase: **Key Queries** and **Transformation Key Queries**. $\mathfrak{B}$ sets an empty table $\mathcal{T}$ and an empty set $\mathcal{D}$.

Key Queries: When $\mathfrak{A}$ issues a key query on attributes set $\mathbb{S}$, $\mathfrak{B}$ sends it to $\mathfrak{C}$ and obtains the key $SK_{\mathbb{S}}$. Then, $\mathfrak{B}$ sets $\mathcal{D} = \mathcal{D} \cup \mathbb{S}$ and responses $SK_{\mathbb{S}}$ to $\mathfrak{A}$ as the answer.

Transformation Key Queries: When $\mathfrak{A}$ issues a transformation key query on attributes set $\mathbb{S}$, $\mathfrak{B}$ searches the entry $< \mathbb{S}, SK_{\mathbb{S}}, TK_{\mathbb{S}}, RK_{\mathbb{S}} >$ in $\mathcal{T}$; if the entry exists in $\mathcal{T}$, then $\mathfrak{B}$ returns $TK_{\mathbb{S}}$. Otherwise, $\mathfrak{B}$ selects a random set $I \subset \{1, \cdots, l\}$ and randomly chooses $\omega_i \in \mathbb{Z}_N$ for $i \in I$. Then, $\mathfrak{B}$ chooses $\kappa, t \in \mathbb{Z}_p$ and $z, d \in \mathbb{Z}_q$ at random and sets

$$K_1' = g_1^{\kappa} g_1^{at} g_2^{z}, K_2' = g_1^{t} g_2^{d}, \left\{\left(K_{\rho(i)} = h_{\rho(i)}^{t}, \omega_i\right)\right\}_{i\in I}$$

Then, $\mathfrak{B}$ puts the entry $< \mathbb{S}, *, TK_{\mathbb{S}} =< K_1', K_2', \{(K_{\rho(i)}, \omega_i)\}_{i\in I}, I >, * >$ in $\mathcal{T}$ and returns $TK_{\mathbb{S}}$ to $\mathfrak{A}$.

(3) **Challenge:** $\mathfrak{A}$ submits two messages M_0 and M_1 to $\mathfrak{B}$ (equal length), and $\mathfrak{B}$ sends M_0 and M_1 to $\mathfrak{C}$. Then, $\mathfrak{C}$ selects $\mu \in \{0, 1\}$ and encrypts M_μ under MPK and $\mathbb{A}^*$ by using the encrypt algorithm in the basic construction and sends CT^* to $\mathfrak{B}$. $\mathfrak{B}$ sends CT^* to $\mathbb{A}^*$.

(4) **Queries 2:** $\mathfrak{A}$ continues to make key queries and transformation key queries adaptively, and $\mathfrak{B}$ returns the answer as **Queries 1**.

(5) **Outputs:** $\mathfrak{A}$ outputs β to $\mathfrak{B}$, and $\mathfrak{B}$ sends it to $\mathfrak{C}$ as its guess to μ.

So if $\mathfrak{A}$ can attack the aforementioned scheme in selectively CPA-secure model, then $\mathfrak{B}$ can break the basic construction in selectively CPA-secure model. □

Theorem 3. *If Assumption 1 holds, then the aforementioned scheme has a verifiable outsourced decryption.*

Proof. Suppose an adversary $\mathfrak{A}$ can attack the verifiability of the aforementioned scheme, then there exists an algorithm $\mathfrak{B}$ that can break Assumption 1 also with non-negligible probability. $\mathfrak{B}$ gives $< N = , \mathbb{G}, \mathbb{G}_T, e, g_1, X_1, Y_1Y_2, T >$, whose goal is to determine $T \in \mathbb{G}$ or $T \in \mathbb{G}_p$.

(1) **Setup:** $\mathfrak{B}$ randomly chooses a collision resistant hash function $H : \mathbb{G}_p \to \mathbb{Z}_p$, sends $MPK =< N, \mathbb{G}, g_1, h_1, \cdots, h_U, g_2, g_1^a, y = e(g_1, g_1)^{\alpha}, H >$ to $\mathfrak{A}$. The master secret key is $MSK = g_1^{\alpha}$.

(2) **Queries 1:** $\mathfrak{A}$ can issue **Key Queries**, **Transformation Key Queries**, **Decrypt Queries**, and **Light Decrypt Queries** adaptively. $\mathfrak{B}$ can respond to these queries properly, as $\mathfrak{B}$ knows MSK.

§ $e(g_2, g_2)$ is pre-computed in MPK.

Table I. Performance comparison.

Schemes	GHW [11]	LDGW [12]	QDLM [13]	Ours
$\|CT\|$	$(1+2l)\|\mathbb{G}\|$	$(3+4l)\|\mathbb{G}\|$	$(1+2l)\|\mathbb{G}\|+l_{CRH}$	$(3+2l)\|\mathbb{G}\|$
$\|CT'\|$	$\|\mathbb{G}_T\|$	$2\|\mathbb{G}_T\|+\|\mathbb{G}\|$	$\|\mathbb{G}_T\|+l_{CRH}$	$\|\mathbb{G}_T\|+2\|\mathbb{G}\|$
O.Dec	$(2+l)Pr+2lEx$	$(4+l)Pr+(2+4l)Ex$	$(2+l)Pr+2lEx$	$2lPr+2lEx$
L.Dec	$1Ex$	$4Ex$	$1Ex$	$1Ex$
HP	No	No	No	Directionally

(3) **Challenge:** $\mathfrak{A}$ submits a M^* and $\mathbb{A}$ to $\mathfrak{B}$, and $\mathfrak{B}$ encrypts M^* under *MPK* and $\mathbb{A}$ by using the encrypt algorithm in the aforementioned scheme and sends CT^* to $\mathfrak{A}$.

(4) **Queries 2:** $\mathfrak{A}$ makes key queries and transformation key queries adaptively, and $\mathfrak{B}$ returns the answer as **Queries 1**.

$$\left\langle M_b \cdot T, g_1^{H(M_b)}, g_1^s Y_2, \left(C_i = \left(g_1^s Y_2\right)^{a\tilde{\lambda}_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i} \right)_{i\in[l]} \right\rangle$$

(5) **Outputs:** $\mathfrak{A}$ outputs $\mathbb{S}^*$ and CT'^*.

Because $\mathfrak{B}$ knows $RK_{\mathbb{S}^*}$, she can recover M from CT'^* by using $RK_{\mathbb{S}^*}$. If $\mathfrak{A}$ wins the aforementioned game, then $\mathfrak{B}$ obtains $g_1^{H(M)} = g_1^{H(M^*)}$, where $M \neq M^*$ and $H(M) \neq H\left(M^*\right)$, as H is collision resistant. Thus, $H(M) = H\left(M^*\right)$ mod p, and $\mathfrak{B}$ can compute $a = gcd(H(M) - H\left(M^*\right), N)$. Let b denote N/a. Then, $\mathfrak{B}$ can check whether $a = p$ by $X_1^a = 1$. If the equation holds, then $\mathfrak{B}$ can distinguish between $T \in \mathbb{G}_p$ and $T \in \mathbb{G}$ by using $\hat{e}(Y_1 Y_2, T)^b \stackrel{?}{=} 1$. Thus, if $\mathfrak{A}$ wins, then $\mathfrak{B}$ can break Assumption 1. □

Theorem 4. *If Assumption 2. holds, then the aforementioned scheme is directionally hidden policy to the proxy in cloud.*

Proof. In the aforementioned scheme, we add elements in G_q to ciphertext and transformation key, and they can be considered as "semi-functional" [20]. In the aforementioned application, the proxy in cloud is semi-trusted, and the access policy should be hidden to it. Thus, the proxy acts as an adversary in the real game Game_r, and it only obtains transformed ciphertext and key from the user. Obviously, the transformed key cannot decrypt transformed ciphertext, if the retrieving number $RK_{\mathbb{S}} \neq 0$. Thus, the advantage probability of adversary in Game_r is negligible.

The next game Game_f is the same as Game_r except that the challenge ciphertext CT are chosen from $\mathbb{G}_T$ at random. Thus, if Game_r and Game_f can be proved indistinguishable, then the access policy is proved hidden to the proxy in cloud. □

Lemma 3. *If Assumption 2 holds, then* Game_r *and* Game_f *are indistinguishable.*

Proof. Given an instance $\left(g_1, g_1^\alpha X_2, g_1^s Y_2, Z_2, T\right)$ of Assumption 2, $\mathfrak{B}$ sets $g_2 = Z_2, y = e\left(g_1, g_1^\alpha X_2\right) = e(g_1, g_1)^\alpha$. Then, $\mathfrak{C}$ can answer all key queries. In the challenge phase, $\mathfrak{B}$ randomly chooses $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_l, r_1, \cdots, r_l \in \mathbb{Z}_N$ returns the ciphertext CT^* as

$$\left\langle M_b \cdot T, g_1^{H(M_b)}, g_1^s Y_2, \left(C_i = \left(g_1^s Y_2\right)^{a\tilde{\lambda}_i} h_{\rho(i)}^{-r_i}, D_i = g_1^{r_i} \right)_{i\in[l]} \right\rangle$$

where T is the assumption term. Let $g_1^s Y_2 = g_1^s g_2^c$, then

$$\left\langle M_b \cdot T, g_1^{H(M_b)}, g_1^s g_2^\delta, \left(C_i = g_1^{a\lambda_i} h_{\rho(i)}^{-r_i} g_2^{\tau_i}, D_i = g_1^{r_i} \right)_{i\in[l]} \right\rangle$$

where $\delta = c, \lambda_i = s \cdot \tilde{\lambda}_i, \tau_i = ac\tilde{\lambda}_i$. If $T = e(g_1, g_1)^{\alpha s}$, then $\mathfrak{B}$ can simulate Game_r in this case. However, if $T \in \mathbb{G}_T$ is a random element, then $\mathfrak{B}$ can simulate Game_f. Thus, if the adversary $\mathfrak{A}$ has non-negligible for distinguishing between Game_f and Game_r, then $\mathfrak{B}$ can break the Assumption 2 by using $\mathfrak{A}$'s output with the same probability. □

5.3. Comparison

In this section, we compare the performance of our scheme with other three ABE schemes [11–13] with outsourced decryption in Table I. Let an LSSS access structure be an $l \times n$ matrix. For the scheme of Baodong *et al.* [13], we let l_{CRH} denote the output of collision-resistent hash function. Let *Pr* denote the time of computing a pairing, and let *Ex* denote the times of computing exponentiations in group $\mathbb{G}$ or $\mathbb{G}_T$. Let $|CT|$ and $|CT'|$ denote the length of outsourced ciphertext and transformed ciphertext, respectively. Let *O.Dec* and *L.Dec* denote the computation cost of the outsourced decryption and the light decryption. Finally, we denote *HP* as hidden policy.

From Table I, we can see that the performance of our scheme¶ is similar to the scheme of Baodong *et al.*

¶ In Waters' ABE with fast decryption scheme [26], it still needs two pairings in the decryption phase, while it only needs one exponentiation in the light decryption phase of our scheme. The operation of pairing is much more time-consuming than exponentiation.

DOI: 10.1002/sec

Table II. Properties comparison.

Schemes	KSW [14]	LDL [15]	Ours
Anonymity of access structures	Fully hidden	Partially hidden	Directionally hidden
Expressiveness of access structures	Inner product predicates	LSSS	LSSS
Security	Fully secure	Fully secure	Fully secure
Outsourced decryption	No	No	Yes

LSSS, linear secret sharing scheme.

and superior to the other two schemes. However, our scheme can be directionally hidden policy, which is superior to these schemes.

Furthermore, an overview comparison of our scheme with those of other two CP-ABE schemes [14,15] with hidden access structures is given in Table II.

In the partially hidden policy scheme [15], each attribute includes two parts: attribute name and attribute value, for example, "Doctor" is the attribute name, while "Gynecologist," "Cardiologist," or "Psychologist" may be the attribute value. Lai *et al*. [15] use the orthogonality property in composite order bilinear group to hide the attribute values, but the attribute names are still to be exposed. Thus, the adversary still can speculate that someone was ill if the policy of her ciphertext involves the attribute name "Doctor," although he does not know which kinds of disease she got. In our directionally hidden scheme, the access policy is completely hidden to the semi-trusted proxy in cloud that is better than the partially hidden policy scheme [15]. But compared with the fully hidden policy scheme [14], the shortcoming of our scheme is that the ciphertexts should be kept secretly by the receiver. However, this shortcoming was offset by the efficiency. In [14], the access structure must be converted to an inner-product predicate, and this causes a superpolynomial blowup in ciphertext size. So the CP-ABE scheme in [14] is very inefficient. From Table II, we can conclude that our scheme is superior to the other two schemes, because it is not only with hidden policy but also with outsourced decryption.

6. CONCLUSIONS

In this work, we propose a secure definition of CP-ABE-OD-DHP. In such a CP-ABE system, a proxy in cloud can help the user to decrypt a ciphertext, but it cannot learn the message and the access policy. Finally, the user can verify the correctness of partially decrypted ciphertext from the proxy. Thus, not only the plaintexts but also the access policies are obfuscated in the proxy point of view during the decryption process. Because the access policies are only hidden to the proxy, we call it *directionally hidden policy*. We construct a concrete CP-ABE-OD-DHP scheme and prove its *security* based on Waters' CP-ABE scheme and prove its *verifiability* and *directionally hidden policy*.

ACKNOWLEDGEMENTS

This research is partially supported by the National Natural Science Foundation of China under grant nos. 61373006, 61232016, and U1405254; the Qing Lan Project; the PAPD fund; and Collaborative Innovation Center of Information Security Technology in Anhui University.

REFERENCES

1. Boneh D, Ding X, Tsudik G. Fine-grained control of security capabilities. *ACM Transactions on Internet Technology (TOIT)* 2004; **4**(1): 60–82.
2. Sahai A, Waters B. Fuzzy identity based encryption. *EUROCRYPT'05, LNCS 3494*, Springer-Verlag, Berlin, 2005; 457–473.
3. Libert B, Yung M. Adaptively secure non-interactive threshold cryptosystems. *Proceedings of the 35th international colloquium on Automata, Languages and Programming (ICALP'11), Part II, LNCS 6756*, Springer-Verlag, Berlin, 2011; 588–600.
4. Lewko A, Okamoto T, Sahai A, Takashima T, Waters B. Fully securefunctional encryption: attribute-based encryption and (hierarchical) inner product encryption. *EUROCRYPT'10, LNCS 6110*, Springer-Verlag, Berlin, 2010; 62–91.
5. Beimel A, Gal A, Paterson M. Lower bounds for monotone span programs. *Computational Complexity* 1997; **6**(1): 29–45.
6. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. *Public Key Cryptography*, Taormina, Italy, 2011; 53–70.
7. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute-based encryption. *Proceedings of the 35th international colloquium on Automata, Languages and Programming (ICALP'08) Part II, LNCS 5126*, Springer-Verlag, Berlin, 2008; 579–591.
8. Zhang M, Yang B, Takagi T. Bounded leakage-resilient functional encryption with hidden vector predicate. *The Computer Journal, Oxford* 2013; **56**(4): 464–477.
9. Pandit T, Barua R. Efficient fully secure attribute-based encryption schemes for general access structures. *ProvSec'12, LNCS 7496*, Springer-Verlag, Berlin, 2012; 193–214.

10. Hur J, Noh DK. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Transactions on Parallel and Distributed Systems* 2011; **22**(7): 1214–1221.
11. Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts. *Proceedings of the USENIX Security Symposium*, San Francisco, CA, USA, 2011.
12. Junzuo Lai J, Deng RH, Guan C, Weng J. Attribute-based encryption with verifiable outsourced decryption. *IEEE Transactions on Information Forensics and Security* 2013; **8**(8): 1343–1354.
13. Qin B, Deng R, Liu S, Ma S. Attribute-based encryption with efficient verifiable outsourced decryption. *IEEE Transactions on Information Forensics and Security* 2015; **10**(7): 1384–1392.
14. Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. *Journal of Cryptology* 2013; **26**(2): 191–224.
15. Lai J, Deng RH, Li Y. Expressive CP-ABE with partially hidden access structures. *ASIACCS*, Seoul, South Korea, 2012; 18–19.
16. Chang V, Kuo Y-H, Ramachandran M. Cloud computing adoption framework: a security framework for business clouds. *Future Generation Computer Systems* 2016; **57**: 24–41.
17. Fu Z, Sun X, Liu Q, Zhou L, Shu J. Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Transactions on Communications* 2015; **E98-B**(1): 190–200.
18. Ren Y, Shen J, Wang J, Han J, Lee S. Mutual verifiable provable data auditing in public cloud storage. *Journal of Internet Technology* 2015; **16**(2): 317–323.
19. Boneh D, Goh E, Nissim K. Evaluating 2-DNF formulas on ciphertexts. *TCC*, Cambridge MA, USA, 2005; 325–341.
20. Lewko A, Waters B. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. *TCC'10, LNCS 5978*, Springer-Verlag, Berlin, 2010; 455–479.
21. Waters B. Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In *Advances in Cryptology - CRYPTO 2009 of LNCS*, Vol. 5677. Springer, 2009; 619–636.
22. Yuen TH, Chow SSM, Zhang Y, Yiu SM. Identity-based encryption resilient to continual auxiliary leakage. *EUROCRYPT*, Cambridge, United Kingdom, 2012; 117–134.
23. Zhang M, Shi W, Wang C, Chen Z, Mu Y. Leakage-resilient attribute-based encryption with fast decryption: models, analysis and constructions. *ISPEC 2013 LNCS 7863*, Springer-Verlag, Berlin, 2013; 75–90.
24. Lewko A, Rouselakis Y, Waters B. Achieving leakage resilience through dual system encryption. *TCC 2011, LNCS 6597*, Rhode Island, USA, 2011; 70–88.
25. Beimel A. Secure schemes for secret sharing and key distribution. *PhD thesis*, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
26. Hohenberger S, Waters B. Attribute-based encryption with fast decryption. *Public-Key Cryptography - PKC 2013 of LNCS*, Nara, Japan, 2013; 162–179.