SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks (2016)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/sec.1509

RESEARCH ARTICLE

Anonymous attribute-based proxy re-encryption for access control in cloud computing

Yinghui Zhang[1,2]*, Jin Li[3], Xiaofeng Chen[2,4] and Hui Li[4]

[1] National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China
[2] State Key Laboratory of Cryptology, PO Box 5159, Beijing 100878, China
[3] School of Computer Science and Educational Software, Guangzhou University, Guangzhou, China
[4] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, Xi'an, China

ABSTRACT

As a public key cryptographic primitive, attribute-based encryption (ABE) is promising in implementing fine-grained access control in cloud computing. However, before ABE comes into practical applications, two challenging issues have to be addressed, that is, users' attribute privacy protection and access policy update. In this paper, we tackle the aforementioned challenge for the first time by formalizing the notion of anonymous ciphertext-policy attribute-based proxy re-encryption (anonymous CP-ABPRE) and giving out a concrete construction. We propose a novel technique called match-then-re-encrypt, in which a matching phase is additionally introduced before the re-encryption phase. This technique uses special components of the proxy re-encryption key and ciphertext to anonymously check whether the proxy can fulfill a proxy re-encryption or not. Theoretical analysis and simulation results demonstrate that our anonymous CP-ABPRE scheme is secure and efficient. Copyright © 2016 John Wiley & Sons, Ltd.

KEYWORDS

cloud computing; access control; attribute-based proxy re-encryption; anonymity; access policy update

***Correspondence**

Yinghui Zhang, National Engineering Laboratory for Wireless Security, Xi'an University of Posts and Telecommunications, Xi'an, China.

E-mail: yhzhaang@163.com

1. INTRODUCTION

With the advent of cloud computing technology, it is essential that users' data should be properly controlled and protected on cloud computing platforms. However, traditional access control mechanisms are not suitable for cloud computing. For one thing, in traditional solutions, service providers are fully trusted by data owners, and access policies are defined and enforced by service providers. For another, cloud users often want to share their data with users they specified and to make access policies themselves.

As a public key cryptographic primitive, attribute-based encryption (ABE) is promising in implementing cloud-based access control systems [1–3], where users enjoy different and flexible access privileges. There are two kinds of ABE schemes, that is key-policy ABE (KP-ABE) construction and ciphertext-policy ABE (CP-ABE) scheme [2]. In this paper, our concern is on the latter. Especially, in a CP-ABE system, a data owner encrypts messages by enforcing an access policy that corresponds to a universe of attributes. As a result, a user is able to decrypt a given ciphertext only if his attributes satisfy the specified access policy. In addition, CP-ABE no longer relies on cloud storage servers for resisting the collusion attack from some unauthorized users. The aforementioned properties of CP-ABE are very applicable to fine-grained access control on untrusted storage. On the other hand, in military circumstances and commercial fields, it is necessary to protect the access policy in that it could be sensitive information. Also, access rights delegation is indispensable for many application scenarios in practice. However, typical CP-ABE techniques fail to realize the protection of users' attribute privacy or the update of access policies.

In order to tackle the aforementioned challenge, anonymous ABE has been studied [4–11]. In particular, a novel technique called match-then-decrypt was introduced in [11] to improve decryption efficiency. Although keeping the access policy hidden, anonymous CP-ABE fails to support access policy update. On the other hand, ciphertext-

policy attribute-based proxy re-encryption (CP-ABPRE) [12,13] successfully supports access policy update and hence allows users to delegate access privileges in cloud computing environment. Unfortunately, CP-ABPRE cannot protect users' attribute privacy. To the best of the authors' knowledge, there seems no available anonymous CP-ABPRE schemes, which can protect users' attribute privacy and support access policy update, simultaneously.

1.1. Our contribution

Contributions in this study can be summarized as follows:

(1) In order to realize privacy-preserving access control supporting access policy update in cloud computing, we formalize the notion and security model of anonymous CP-ABPRE and propose the first anonymous CP-ABPRE scheme. It's noted that anonymous CP-ABPRE cannot be realized by trivial combinations of the existing techniques such as CP-ABE and proxy re-encryption (PRE). The main reason is that users' attribute privacy protection requires access policies not be revealed in ciphertexts, which leads to the failure of attribute matching detection in proxy re-encryption and hence access policy update cannot be realized.
(2) A novel technique called match-then-re-encrypt is introduced into the proxy re-encryption of anonymous CP-ABPRE, in which a matching phase is added before the proxy re-encryption phase. This technique works by computing special components in proxy re-encryption keys and ciphertexts, which are used to check whether the proxy can fulfill a proxy re-encryption or not. In addition, the technique of match-then-decrypt is adopted to improve decryption efficiency. It can efficiently determine whether an attribute private key matches the hidden access policy without decryption. In the proposed concrete construction, the novel techniques of match-then-re-encrypt and match-then-decrypt can protect users' attribute privacy and support access policy update, simultaneously.
(3) In the standard model, the proposed anonymous CP-ABPRE scheme is proved to be secure. Even legitimate decryptors cannot obtain any information on the potential access policy in ciphertexts more than the fact that he can recover the corresponding plaintexts. The analysis and simulation results show that the proposed scheme is practical.

1.2. Related work

We summarize the major related work in the areas of typical ABE, anonymous ABE, and attribute-based PRE.

Typical ABE. Since ABE [1] was introduced to realize access control systems, there have been a plenty of researches on ABE constructions [2,3,14–18]. KP-ABE and CP-ABE, as two variants of ABE, were defined by Goyal *et al*. [2]. In a KP-ABE system, a user can obtain a secret key, which corresponds to an access policy, from a trusted attribute authority. Each ciphertext is associated with some attributes, and the access policy indicates whether a secret key can decrypt a ciphertext or not. In CP-ABE, the attribute authority provides users with secret keys, which are associated with sets of descriptive attributes, and a ciphertext corresponds to an access policy. The first KP-ABE scheme was constructed in [2], and the access policy is monotonic. For the sake of expressiveness, Ostrovsky *et al*. [14] proposed a KP-ABE system with non-monotone formulas for the first time. However, CP-ABE is more suitable for the secure cloud-based data sharing environment than KP-ABE in that it empowers data owners the ability of making policies [19]. The first CP-ABE scheme was constructed by Bethencourt *et al*. [3], while it suffers a security weakness that the security proof was given in generic group models. To achieve strengthened security in CP-ABE, Cheung *et al*. [15] constructed an AND-gate policy CP-ABE scheme and proved its security in standard models. Later, a new CP-ABE scheme was given by Goyal *et al*. [16], which enjoys expressiveness and security in standard models. However, the scheme is impractical due to low efficiency. Recently, Hohenberger *et al*. [17] have proposed an ABE construction with fast decryption. Garg *et al*. [18] have given the first construction of ABE for general circuits from multilinear maps. It achieves both KP-ABE and CP-ABE variants. Zhang *et al*. [20] have proposed a CP-ABE scheme with small computation cost and constant-size ciphertexts. The scheme can efficiently support AND-gate access policies with multiple attribute values and wildcards. Although having various attractive features, the aforementioned CP-ABE schemes suffer from a weakness that the access policy has to be sent along with ciphertexts explicitly because recipients must know how to combine secret key components for a successful decryption.

Anonymous ABE. In order to protect users' attribute privacy, anonymous ABE has been studied [4–11]. In anonymous CP-ABE, a decryptor obtains his attribute secret key from an attribute authority. If the secret key does not match the access policy specified for a ciphertext, the decryptor fails to recover the plain data. In particular, it is impossible for him to guess what access policy has been specified for the ciphertext. Kapadia *et al*. [4] constructed a CP-ABE scheme, which can realize hidden ciphertext policies of AND gates on positive and negative values of attributes, but it cannot resist collusion attacks. On the basis of the primitive Hidden Vector Encryption, a predicate encryption scheme was presented by Boneh *et al*. [5], which can be used to construct anonymous CP-ABE schemes with the help of predicates with opposite semantics. Furthermore, an inner product encryption scheme was proposed by Katz *et al*. [6], which can achieve a hidden CP-ABE variant. Later, a more efficient anonymous CP-ABE schemes was constructed [7]. Li *et al*. [8] proposed an anonymous CP-ABE system for realizing accountability. To achieve full security, Lai *et al*. [9,10] proposed

two anonymous CP-ABE schemes under new assumptions based on composite order groups. However, in these anonymous CP-ABE schemes, the user has to decrypt and determine whether his attribute list matches the hidden access policy or not. Such a test has to be repeated until a successful decryption or all of the possible tests have been considered. As a result, this direct decryption method will suffer a severe efficiency drawback. In order to tackle this problem, Zhang *et al*. [11] constructed a new anonymous CP-ABE scheme, in which a novel technique called match-then-decrypt is introduced and decryption efficiency is greatly improved. Note that, although protecting users' attribute privacy, the aforementioned anonymous CP-ABE schemes fail to support access policy update.

Attribute-Based PRE (ABPRE). In practical access control systems, CP-ABPRE can be deployed to realize access rights delegation because it supports access policy update. In CP-ABPRE, a proxy is specified by users in advance. The proxy can receive a proxy re-encryption key, which is used to transform a ciphertext associated with an access policy into another one with a new policy. Blaze *et al*. [21] proposed the notion of PRE for the first time. Later, many research works were dedicated to PRE with nice properties [22–25]. However, the traditional PRE schemes are not appropriate to cloud-based environment. In cloud-based data sharing, Liang *et al*. [12] presented a CP-ABPRE scheme for purpose of access rights delegation, which is proved selective ciphertext-policy and chosen plaintext secure. Subsequently, Luo *et al*. [13] proposed a new CP-ABPRE scheme. Kawai *et al*. [26] proposed an anonymous inner-product PRE scheme and a ciphertext-policy functional PRE scheme. However, both solutions adopt a one-time signature, which is strongly unforgeable, and the latter fails to conceal ciphertext-policy in ciphertexts. Very recently, Liu *et al*. [27,28] have proposed the notion of time-based ABPRE by considering time in access policies. Although supporting access policy update, existing CP-ABPRE schemes cannot protect users' attribute privacy.

1.3. Organization

The rest of this paper is organized as follows. In Section 2, we review some preliminaries mainly consist of cryptographic backgrounds. In Section 3, we display the system model and design goals. In Section 4, we give the formalized security model. In Section 5, our construction for anonymous CP-ABPRE together with its security results and performance comparisons are described. Finally, we draw a conclusion in Section 6.

2. PRELIMINARIES

We review some cryptographic background, and then, access policies are detailed.

2.1. Cryptographic background

2.1.1. Bilinear pairings.

Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic multiplicative groups of some large prime order p, and we denote the identities of $\mathbb{G}$ and $\mathbb{G}_T$ as $1_{\mathbb{G}}$ and $1_{\mathbb{G}_T}$, respectively. Let $g \in_R \mathbb{G}$ be a generator. We call $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ a bilinear pairing if it is a map satisfying properties in the following:

(1) Bilinear: For any $a, b \in \mathbb{Z}_p$, $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.
(2) Non-degenerate: There exists $\hat{e}(g_1, g_2) \neq 1$ for some $g_1, g_2 \in \mathbb{G}$.
(3) Computable: For all $g_1, g_2 \in \mathbb{G}$, $\hat{e}(g_1, g_2)$ can be computed efficiently.

2.1.2. Complexity assumptions.

The Decisional Bilinear Diffie–Hellman (DBDH) Assumption: Let $a, b, c, z \in_R \mathbb{Z}_p$, and $g \in_R \mathbb{G}$ be a generator. The DBDH assumption holds in group $\mathbb{G}$ if no probabilistic polynomial-time (PPT) algorithm can distinguish the tuple $[g, g^a, g^b, g^c, \hat{e}(g, g)^{abc}]$ from $[g, g^a, g^b, g^c, g^z]$ with non-negligible advantage.

The Decision Linear (D-Linear) Assumption: Suppose $g \in_R \mathbb{G}$ is a generator and $a_1, a_2, a_3, a_4, z \in_R \mathbb{Z}_p$. The D-Linear assumption is said to hold in $\mathbb{G}$ if no PPT algorithm can distinguish the tuple $[g, g^{a_1}, g^{a_2}, g^{a_1 a_3}, g^{a_2 a_4}, g^{a_3 + a_4}]$ from $[g, g^{a_1}, g^{a_2}, g^{a_1 a_3}, g^{a_2 a_4}, g^z]$ with an advantage non-negligible .

The Computational Bilinear Diffie-Hellman (CBDH) Assumption: Suppose $g \in_R \mathbb{G}$ is a generator and $a, b, c; \in_R \mathbb{Z}_p$. We say that the CBDH assumption holds in $\mathbb{G}$ if given $[g, g^a, g^b, g^c]$, no PPT algorithm can compute $\hat{e}(g, g)^{abc}$ with an advantage non-negligible.

2.2. Access policy

In CP-ABE, an access policy is a ciphertext policy. Intuitively, an access policy is a rule W over descriptive attributes that returns either "true" or "false" given an attribute list L. L satisfies W only if W returns "true" on L. Usually, notation $L \models W$ is used to represent the fact that L satisfies W, and $L \not\models W$ means L does not match W. In our construction, we consider access policies consisting of an AND-gate supporting multiple values of attributes with wildcards. As a generalization of access policies in [15], the access policy in our construction is the same as those in [7]. For a given access policy $W = [W_1, W_2, \cdots, W_n]$, an attribute list $L = [L_1, L_2, \cdots, L_n]$ satisfies $L \models W$ if $L_i = W_i$ or for all $1 \leq i \leq n$ the equation $W_i = *$ holds, and $L \not\models W$ otherwise. Note that the wildcard * in the access policy plays the same role as the "don't care" value.

3. SYSTEM MODEL AND DESIGN GOALS

We first present the system model in this section and then describe our design goals.

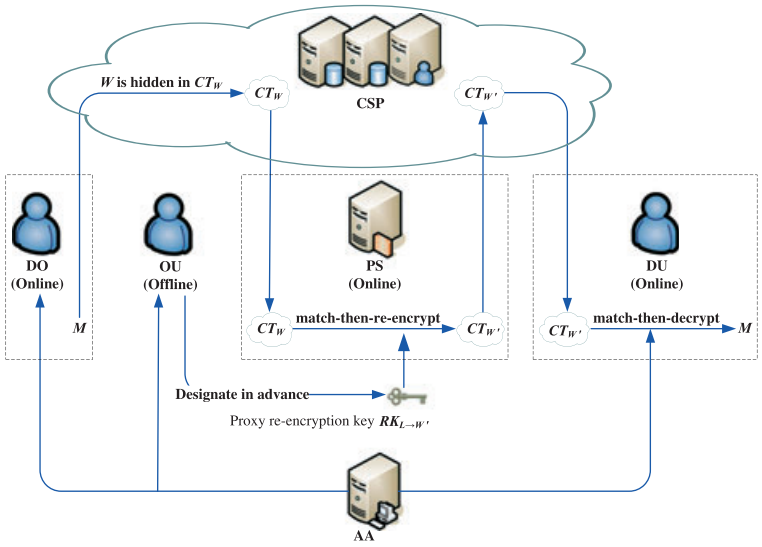

Figure 1. System model of anonymous access control supporting access policy update.

3.1. System model

As shown in Figure 1, the system model of anonymous attribute-based access control supporting access policy update in cloud computing consists of six entities Attribute Authority (AA), Cloud Service Provider (CSP), Data Owner (DO), Original User: the decryptor of original ciphertexts (OU), Proxy Server (PS), and Delegated User: the decryptor of re-encryption ciphertexts (DU). Because AA and CSP are the same as those in typical ABE, we only describe the entities DO, OU, PS, and DU.

- DO is an entity who owns data M and intends to safely store it on cloud storage servers provided by CSP for sharing. DO is responsible for making attribute-based access policy W such that an attribute list L satisfies $L \models W$ and enforcing it on M anonymously by encrypting M under W to achieve original ciphertext CT_W. In anonymous attribute-based access control, W cannot be revealed in CT_W in order to protect users' attribute privacy.
- OU is an entity who has an attribute list L and intends to access the encrypted data CT_W hosted in CSP. If $L \models W$, where W is the underlying access policy hidden in CT_W specified by DO, then OU will succeed in decrypting CT_W. In order to improve the efficiency of decryption in anonymous attribute-based access control, a novel technique called match-then-decrypt is adopted, in which a matching phase is additionally introduced before the decryption phase. To be specific, after downloading CT_W from CSP, OU should check whether his attribute private key matches W or not before full decryption. And OU performs the full decryption if and only if the attribute matching is successful.
- PS[†] is an entity designated by OU in advance. PS can get a proxy re-encryption key $RK_{L\to W'}$ from OU and generate $CT_{W'}$ by re-encrypting CT_W to realize access policy update from W to W'. In order to realize proxy re-encryption, a novel technique called match-then-re-encrypt is introduced, in which a matching phase is additionally introduced before the re-encryption phase. Specifically, before re-encrypting CT_W from CSP, PS should determine whether $RK_{L\to W'}$ can re-encrypt CT_W or not. In other words, PS performs the full proxy re-encryption only if the attribute matching is successful.
- DU[‡] is an entity who intends to access the re-encrypted data $CT_{W'}$ hosted in CSP. If DU has an attribute list matching the underlying new access policy W' hidden in $CT_{W'}$ specified by OU, then he will succeed in decrypting $CT_{W'}$. Similarly, the technique match-then-decrypt can be adopted by DU to improve decryption efficiency.

3.2. Design goals

- *Data Confidentiality.* DO will try his best to prevent unauthorized users from accessing plaintexts if the users' attribute list does not match the access policy specified for corresponding ciphertexts. In addition, CSP and PS are not allowed to access plaintexts of the encrypted data.
- *Collusion-Resistance.* Some users may combine their attribute secret keys to decrypt a ciphertext that cannot be decrypted by any one of them alone. In a practi-

[†] The cloud storage server can act as PS.
[‡] DU may act as OU of some ciphertexts.

cal access control system, the property of collusion-resistance means that these colluders fail to recover the plaintexts.
- *Attribute Privacy Protection.* In attribute-based access control supporting access policy update, access policies may be sensitive and need to be protected. For one thing, PS should complete proxy re-encryption without knowing access policies. For another, even if a user's attribute list matches the ciphertext policy, the user should not learn any information on the policy specified by DO more than the fact that he can access the corresponding plaintext.
- *Master Key Security.* Even if PS colludes with DU, it is infeasible for them to obtain the original decryption key of OU.

Also, the performance-related issue should be taken into consideration. In anonymous CP-ABPRE, efficient attribute matching detections should be made before full proxy re-encryption and decryption.

4. FORMALIZED SECURITY MODEL

Before giving the formalized security model, we first lay out the definition of anonymous CP-ABPRE. An anonymous CP-ABPRE scheme is composed of six algorithms given in the following:

- **Setup**$(1^\lambda) \rightarrow (PK, MK)$: The setup algorithm is run by AA. Taking a security parameter λ as input, it returns a system public key PK and a master key MK. Note that PK is known to all users and SK is kept secret.
- **KeyGen**$(PK, MK, L) \rightarrow SK_L$: The key generation algorithm is run by AA. On input PK, MK, and an attribute list L, it outputs the attribute secret key SK_L associated with L.
- **Encrypt**$(PK, M, W) \rightarrow CT_W$: The encryption algorithm is run by DO. DO specifies an access policy W. On input PK, a message M and W, it generates a ciphertext CT_W as the encryption of M with respect to W. To achieve the anonymity of access policy, W cannot be revealed in CT_W.
- **RKGen**$(PK, SK_L, W') \rightarrow RK_{L \rightarrow W'}$: This algorithm is run by OU or DU to obtain a proxy re-encryption key. On input PK, SK_L associated with L and an access policy W', it outputs $RK_{L \rightarrow W'}$ as a proxy re-encryption key.
- **Reencrypt**$(PK, RK_{L \rightarrow W'}, CT_W) \rightarrow CT_{W'}$ or $\perp$: The proxy re-encryption algorithm is run by PS. It involves two phases, that is, attribute matching detection and re-encryption phase. On input PK, $RK_{L \rightarrow W'}$ and a ciphertext CT_W, PS does the following without knowing W:

  (1) Matching Phase: It returns $\perp$ to terminate re-encryption with overwhelming probability if $L \not\models W$. Otherwise, the Matching Phase ends by initiating the Re-encryption Phase.
  (2) Re-encryption Phase: It generates a proxy re-encryption ciphertext $CT_{W'}$ corresponding to W', where the plaintext is the same as that of CT_W.

- **Decrypt**$(PK, CT_W, SK_L) \rightarrow M$ or $\perp$: The decryption algorithm is run by OU or DU. It involves two phases, that is, attribute matching detection and decryption phase. On input PK, a ciphertext CT_W of a message M under W, and a secret key SK_L associated with L, the ciphertext CT_W is tested and decrypted by a user (OU or DU) with secret key SK_L without knowing W as follows:

  (1) Matching Phase: It returns $\perp$ to terminate decryption with overwhelming probability if $L \not\models W$. Otherwise, the Matching Phase ends by initiating the Decryption Phase.
  (2) Decryption Phase: It returns M.

Subsequently, to achieve the security goals in Section 3.2, we model the capability of adversaries and define corresponding security notions. The design goals of *Data Confidentiality*, *Collusion-Resistance*, and *Attribute Privacy Protection* are reflected in the indistinguishability against selective ciphertext-policy and chosen-plaintext attacks (IND-sCP-CPA) [2,3,15] model. The *Master Key Security* goal is reflected in the selective master key security (sMKS) model. Both models are interactive games between an adversary and a challenger.

4.1. IND-sCP-CPA Game

Init: $\mathcal{A}$ submits two challenge ciphertext policies W_0^* and W_1^* to the challenger.

Setup: The challenger specifies a large security parameter λ, and runs the **Setup** algorithm to get a master key SK and the corresponding public key PK. It keeps SK secretly and makes PK public.

Phase 1: The adversary $\mathcal{A}$ queries the following oracles in polynomial time:

- **KeyGen oracle** $\mathcal{O}_{KeyGen}$: $\mathcal{A}$ submits an attribute list L, and the challenger returns SK_L to $\mathcal{A}$ only if $(L \not\models W_0^* \wedge L \not\models W_1^*)$ or $(L \models W_0^* \wedge L \models W_1^*)$. Otherwise, it outputs $\perp$.
- **RKGen oracle** $\mathcal{O}_{RKGen}$: The adversary $\mathcal{A}$ submits L and W, if $(L \not\models W_0^* \wedge L \not\models W_1^*)$ or $(L \models W_0^* \wedge L \models W_1^*)$, the challenger returns $\mathcal{A}$ the re-encryption key $RK_{L \rightarrow W}$. Otherwise, it outputs $\perp$.
- **Reencrypt oracle** $\mathcal{O}_{ReEnc}$: $\mathcal{A}$ submits L, W', and an anonymous ciphertext CT_W under an access policy W, if $((L \not\models W_0^* \wedge L \not\models W_1^*)$ or $(L \models W_0^* \wedge L \models W_1^*))$ and $L \models W$, the challenger gives $\mathcal{A}$ $CT_{W'}$. Otherwise, it outputs $\perp$.

Challenge: Once **Phase 1** is over, $\mathcal{A}$ outputs two messages M_0 and M_1 of equal length. It is required that $M_0 = M_1$ if any secret key on L satisfying $L \models W_0^* \wedge L \models W_1^*$ has been queried. The challenger randomly chooses a bit $\nu \in \{0,1\}$, computes $CT_{W_\nu^*} = \textbf{Encrypt}(PK, M_\nu, W_\nu^*)$, and sends $CT_{W_\nu^*}$ to $\mathcal{A}$, where W_ν^* is hidden.

Phase 2: It is similar to **Phase 1**.

Guess: $\mathcal{A}$ outputs a bit $\nu' \in \{0,1\}$ as a guess of ν, and it wins the aforementioned game if $\nu' = \nu$.

In the IND-sCP-CPA game, we define the advantage of $\mathcal{A}$ as $\mathsf{Adv}^{\text{IND-sCP-CPA}}_{\text{CP-ABPRE}}(\mathcal{A}) = |\Pr[\ \nu' = \nu\ ] - \frac{1}{2}|$.

Remark 1. In the model, it is required in the challenge phase that $M_0 = M_1$ if the adversary obtains a secret key SK_L matching both W_0^* and W_1^*. Otherwise, the adversary can directly decrypt the challenge ciphertext to get the bit value ν. Hence, it easily follows that the security models of anonymous CP-ABPRE would make no sense without such a restriction.

4.2. sMKS game

Init: $\mathcal{A}$ submits an attribute list L^* to the challenger.

Setup: The same as that of IND-sCP-CPA Game.

Queries: $\mathcal{A}$ queries the following oracles in polynomial time:

- **KeyGen oracle** $\mathcal{O}_{KeyGen}$: $\mathcal{A}$ submits an attribute list L, if $L \neq L^*$, SK_L is returned by the challenger to $\mathcal{A}$. Otherwise, it outputs $\perp$.
- **RKGen oracle** $\mathcal{O}_{RKGen}$: $\mathcal{A}$ submits L and W, the challenger generates a proxy re-encryption key $RK_{L\to W}$ for $\mathcal{A}$.
- **Reencrypt oracle** $\mathcal{O}_{ReEnc}$: The adversary $\mathcal{A}$ submits an a attribute list L, W', and CT_W under an access policy W, the challenger returns $CT_{W'}$ as a re-encryption ciphertext to $\mathcal{A}$. Note that, the access policy W is not revealed in CT_W to achieve anonymity.

Output: $\mathcal{A}$ outputs an attribute secret key SK_{L^*}, and it succeeds if SK_{L^*} is valid for L^*.

In the sMKS game, we define the advantage of $\mathcal{A}$ as $\mathsf{Adv}^{\text{sMKS}}_{\text{CP-ABPRE}}(\mathcal{A}) = \Pr[\ \mathcal{A}\text{ succeeds }]$.

Definition 1. *An anonymous CP-ABPRE scheme is said to be IND-sCP-CPA (resp. sMKS) secure, if no probabilistic polynomial-time adversary can break the aforementioned IND-sCP-CPA (resp. sMKS) game for anonymous CP-ABPRE with non-negligible advantage.*

5. ANONYMOUS CP-ABPRE SCHEME

5.1. Proposed construction

Setup(1^λ): Choose two cyclic multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of large prime order p. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear pairing. Choose g as a generator of $\mathbb{G}$ and $E : \mathbb{G} \to \mathbb{G}_T$ as an encoding between $\mathbb{G}$ and $\mathbb{G}_T$ [12,13]. Assume the universal attribute set is $\mathcal{U} = \{\omega_1, \omega_2, \cdots, \omega_n\}$. Each attribute has multiple values, and the multi-value set for ω_i is denoted by $S_i = \{v_{i,1}, v_{i,2}, \cdots, v_{i,n_i}\}$. AA chooses $g_2, g_3 \in_R \mathbb{G}$ and $y \in_R \mathbb{Z}_p$. For each attribute ω_i with $1 \le i \le n$, AA also chooses $\{T_{i,t} \in_R \mathbb{G}\}_{1\le t\le n_i}$[§] and $\{a_{i,t}, b_{i,t} \in_R \mathbb{Z}_p\}_{1\le t\le n_i}$. Then, AA computes $g_1 = g^y$, $Y = \hat{e}(g_1, g_2)$ and $\{\{A_{i,t} = T_{i,t}^{a_{i,t}}, B_{i,t} = T_{i,t}^{b_{i,t}}\}_{1\le t\le n_i}\}_{1\le i\le n}$. The system public key is $PK = \langle g, g_1, g_2, g_3, Y, \{\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1\le t\le n_i}\}_{1\le i\le n}\rangle$, and $MK = \langle y, \{\{a_{i,t}, b_{i,t}\}_{1\le t\le n_i}\}_{1\le i\le n}\rangle$ is the master key.

KeyGen(PK, MK, L): After receiving an attribute list $L = [L_1, L_2, \cdots, L_n]$, AA chooses $r', \bar{d} \in_R \mathbb{Z}_p$, $\{r_i, \hat{r}_i, \lambda_i \in_R \mathbb{Z}_p\}_{1\le i\le n}$, sets $r = \sum_{i=1}^n r_i$, $\hat{r} = \sum_{i=1}^n \hat{r}_i$, and computes $[D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}] = [g_2^{y-r}, g_2^{y-\hat{r}}, g_3^{\bar{d}}, g^{r'}]$. For$1 \le i \le n$, suppose the indexes satisfy $L_i = v_{i,k_i}$, AA computes $[D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}] = [g_2^{\hat{r}_i} T_{i,k_i}^{r'}, g_2^{r_i} B_{i,k_i}^{\lambda_i a_{i,k_i}}, g^{\lambda_i a_{i,k_i}}, g^{\lambda_i b_{i,k_i}}]$. Then, $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1\le i\le n}\rangle$.

Encrypt(PK, M, W): To encrypt $M \in \mathbb{G}_T$ under an access policy $W = [W_1, W_2, \cdots, W_n]$, DO chooses $s, s', s'' \in_R \mathbb{Z}_p$, and computes $\tilde{C} = MY^s$, $C_0 = g^s$, $C_{\text{RE}} = g_3^s$, $C_\Delta = Y^{s'}$ and $C_0' = g^{s'}$. Also, DO chooses $\{\sigma_i \in_R \mathbb{G} | 1 \le i \le n\}$ such that $\prod_{i=1}^n \sigma_i = 1_{\mathbb{G}}$. For $1 \le i \le n$ and $1 \le t \le n_i$, DO computes $[C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}]$ as $[\sigma_i T_{i,t}^{s'}, B_{i,t}^{s-s''}, A_{i,t}^{s''}]$ if $v_{i,t} \in W_i$. Otherwise, $[C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}]$ are random elements in $\mathbb{G}$. Finally, the ciphertext of M with respect to W is $CT_W = \langle C_\Delta, C_0', \tilde{C}, C_0, C_{\text{RE}}, \{\{C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\}_{1\le t\le n_i}\}_{1\le i\le n}\rangle$.

RKGen(PK, SK_L, W'): Let $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1\le i\le n}\rangle$ and W' be an access policy. In order to obtain a proxy re-encryption key associated with W', it chooses $d \in_R \mathbb{Z}_p$ and computes g^d, $\widehat{D}_0' = \widehat{D}_0(\bar{D}_0)^n$, $\bar{D}_0' = \bar{D}_0 g_3^d$, $D'_{\Delta,i} = D_{\Delta,i} g_3^d$, $D'_{i,0} = D_{i,0} g_3^d$. Meanwhile, it sets $D_0' = D_0$, $D'_{\Delta,0} = D_{\Delta,0}$, $D'_{i,1} = D_{i,1}$, $D'_{i,2} = D_{i,2}$, and computes $\mathbb{C}$=Encrypt$(PK, E(g^d), W')$. Then, $RK_{L\to W'} = \langle L, D_0', \widehat{D}_0', \bar{D}_0', D'_{\Delta,0}, \{D'_{\Delta,i}, D'_{i,0}, D'_{i,1}, D'_{i,2}\}_{1\le i\le n}, \mathbb{C}\rangle$ is the proxy re-encryption key corresponding to W'.

Reencrypt$(PK, RK_{L\to W'}, CT_W)$: Upon receiving $RK_{L\to W'}$ for W', and CT_W under W, PS does the following without knowing W:

(1) Matching Phase: PS computes $C'_\Delta = \hat{e}\left(C_0', \bar{D}_0'\right)^n$ and checks whether $L \models W$ in terms of the following Equality (1). Specifically, $L \models W$ if and only if Equality (1) holds

[§] The attribute authority can obtain $T_{i,t}$ by choosing $\tau_{i,t} \in_R \mathbb{Z}_p$ and computing $T_{i,t} = g^{\tau_{i,t}}$.

DOI: 10.1002/sec

$$C_\Delta C'_\Delta = \frac{\hat{e}\left(C'_0, \widehat{D}'_0 \prod_{i=1}^{n} D'_{\Delta,i}\right)}{\hat{e}\left(\prod_{i=1}^{n} C_{i,k_i,\Delta}, D'_{\Delta,0}\right)}, \quad (1)$$

where suppose the indexes satisfy $L_i = v_{i,k_i}$. It returns $\perp$ if $L \not\models W$. Otherwise, it initiates the Re-encryption Phase.

(2) Re-encryption Phase: For $1 \le i \le n$, suppose the indexes satisfy $L_i = v_{i,t}$, PS computes

$$E_i = \frac{\hat{e}\left(C_0, D'_{i,0}\right)}{\hat{e}\left(C_{i,t,1}, D'_{i,1}\right)\hat{e}\left(C_{i,t,2}, D'_{i,2}\right)} = \hat{e}(g, g_2)^{sr_i}\hat{e}(g, g_3)^{sd}.$$

Subsequently, it computes

$$\overline{C} = \hat{e}\left(C_0, D'_0\right)\prod_{i=1}^{n} E_i = \hat{e}(g, g_2)^{ys}\hat{e}(g, g_3)^{nsd},$$

and outputs a proxy re-encryption ciphertext $CT_{W'} = \langle \tilde{C}, C_{\mathrm{RE}}, \overline{C}, \mathbb{C}\rangle$. It follows from the subsequent Decrypt algorithm that DU only needs g^d to decrypt the proxy re-encryption ciphertext. Thus, we can obtain a two time proxy re-encryption ciphertext $CT_{W''} = \langle \tilde{C}, C_{\mathrm{RE}}, \overline{C}, \mathbb{C}'\rangle$, where $\mathbb{C}'$ is generated based on the algorithm Reencrypt with the ciphertext $\mathbb{C}$ and another proxy re-encryption key $RK_{L'\to W''}$ as inputs. Specifically, $\mathbb{C}' = \mathrm{Reencrypt}(PK, RK_{L'\to W''}, \mathbb{C})$. Similarly, the multiple time proxy re-encryption ciphertexts can be generated.

Decrypt(PK, CT_W, SK_L): The ciphertext CT_W is tested and decrypted by a user with secret key SK_L as follows:

- If CT_W is an original ciphertext, the user does the following:

  (1) Matching Phase: The user checks whether $L \models W$ in terms of the following Equality (2). Suppose the indexes satisfy $L_i = v_{i,k_i}$, $L \models W$ if and only if Equality (2) holds

  $$C_\Delta = \frac{\hat{e}\left(C'_0, \widehat{D}_0 \prod_{i=1}^{n} D_{\Delta,i}\right)}{\hat{e}\left(\prod_{i=1}^{n} C_{i,k_i,\Delta}, D_{\Delta,0}\right)}. \quad (2)$$

  If $L \not\models W$, it returns $\perp$. Otherwise, it initiates the Decryption Phase.

  (2) Decryption Phase: Suppose the indexes satisfy $L_i = v_{i,k_i}$, the user computes

  $$M = \frac{\tilde{C}\prod_{i=1}^{n}\hat{e}(C_{i,k_i,1}, D_{i,1})\hat{e}(C_{i,k_i,2}, D_{i,2})}{\hat{e}(C_0, D_0)\prod_{i=1}^{n}\hat{e}(C_0, D_{i,0})}.$$

- Else if CT_W is a one-time proxy re-encryption ciphertext consists of $\langle \tilde{C}, C_{\mathrm{RE}}, \overline{C}, \mathbb{C}\rangle$, the user does the following:

  (1) Matching Phase: The user checks whether $L \models W$ in accordance with $\mathbb{C}$ by using the method in Equality (2). If $L \not\models W$, the algorithm returns $\perp$. Otherwise, it initiates the Decryption Phase.

  (2) Decryption Phase: The user does

  – Performs a decryption of the original ciphertext $\mathbb{C}$ of $E(g^d)$ using the secret key SK_L and decodes it to g^d.

  – Computes $\frac{\tilde{C}\hat{e}(C_{\mathrm{RE}}, g^d)^n}{\overline{C}} = M$.

- Else, if CT_W is a multi-time proxy re-encryption ciphertext, the decryption is similar to the aforementioned phases. Suppose that CT_W is an $N + 1$ time proxy re-encryption ciphertext consists of $\langle \tilde{C}, C_{\mathrm{RE}}, \overline{C}, \mathbb{C}'\rangle$, where $\mathbb{C}'$ is an N time proxy re-encryption ciphertext of $E(g^d)$, then the user does the following:

  – Performs a decryption of the N time proxy re-encryption ciphertext $\mathbb{C}'$. If the algorithm does not return $\perp$, the user recovers $E(g^d)$, decodes it to g^d and proceeds.

  – Computes $\frac{\tilde{C}\hat{e}(C_{\mathrm{RE}}, g^d)^n}{\overline{C}} = M$.

5.2. Security results

Theorem 1. *The anonymous CP-ABPRE scheme is secure in the IND-sCP-CPA model, under the DBDH assumption and the D-Linear assumption without random oracles. Hence, it achieves the security goals of Data Confidentiality, Collusion-Resistance, and Attribute Privacy Protection.*

Proof. We prove that the proposed anonymous CP-ABPRE scheme is IND-sCP-CPA secure under the DBDH and D-Linear assumptions in standard models. To be specific, if there exists a polynomial-time adversary $\mathcal{A}$, which can attack the proposed scheme in the IND-sCP-CPA model with an advantage ϵ_{IND}, then it follows that ϵ_{IND} is negligible and $\epsilon_{\mathrm{IND}} \le \epsilon_{\mathrm{DBDH}} + n\epsilon_{\mathrm{DL}}$, where ϵ_{DBDH} and ϵ_{DL} respectively denotes the advantage of a distinguisher of a DBDH challenge and a D-Linear challenge, and n is the size of attribute set of the system.

Suppose that $\mathcal{A}$ commits to the challenge ciphertext policies $W_0^* = [W_{0,1}^*, W_{0,2}^*, \cdots, W_{0,n}^*]$ and $W_1^* = [W_{1,1}^*, W_{1,2}^*, \cdots, W_{1,n}^*]$ at the beginning of the game. Based on the IND-sCP-CPA security model, in order to demonstrate that $\mathcal{A}$ cannot achieve non-negligible advantage in the original security game $\mathbf{G}$, a sequence of hybrid games are used. At first, $\mathbf{G}$ is slightly modified into another game

$\mathbf{G}_0$. The definitions of games $\mathbf{G}$ and $\mathbf{G}_0$ are the same except the generation of the challenge ciphertext. In $\mathbf{G}_0$, to be precise, if the adversary did not obtain the attribute secret key SK_L associated with the attribute list L satisfying the condition of $[L \models W_0^* \wedge L \models W_1^*]$, then the challenge ciphertext component $\tilde{C}$ is a random element in $\mathbb{G}_T$ in any case, while other challenge ciphertext components are generated in a normal way. On the other hand, if $\mathcal{A}$ obtained the secret key SK_L satisfying $[L \models W_0^* \wedge L \models W_1^*]$, then all ciphertext components are generated correctly. In this case, we have $\mathbf{G} = \mathbf{G}_0$. Subsequently, we modify the game $\mathbf{G}_0$ by changing the ciphertext components $\{\{C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\}_{1\le t\le n_i}\}_{1\le i\le n}$ and define the following games in sequence.

If $v_{i,t}$ satisfies $(v_{i,t} \notin W_{0,i}^* \wedge v_{i,t} \notin W_{1,i}^*)$ or $(v_{i,t} \in W_{0,i}^* \wedge v_{i,t} \in W_{1,i}^*)$, $\{C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\}$ are generated in all the games as in the real game. In the case that there exists $v_{i,t}$ satisfying $(v_{i,t} \notin W_{0,i}^* \wedge v_{i,t} \in W_{1,i}^*)$ or $(v_{i,t} \in W_{0,i}^* \wedge v_{i,t} \notin W_{1,i}^*)$, for any random coin, random elements from $\mathbb{G}$ are assigned to $\{C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\}$ in the new modified game $\mathbf{G}_l$, although they are generated normally in $\mathbf{G}_{l-1}$. The process is repeated until there is no component $v_{i,t}$ satisfying $(v_{i,t} \notin W_{0,i}^* \wedge v_{i,t} \in W_{1,i}^*)$ or $(v_{i,t} \in W_{0,i}^* \wedge v_{i,t} \notin W_{1,i}^*)$. In the last game, $\mathcal{A}$'advantage is zero because the challenge ciphertext is independent of the random coin. Based on the aforementioned way, we replace well-formed components in $\mathbf{G}_{l-1}$ with random elements from $\mathbb{G}$ in $\mathbf{G}_l$, then a D-Linear challenge is embedded into the ciphertext and a distinguisher of the D-Linear challenge can be generated with the help of a distinguisher of $\mathbf{G}_{l-1}$ and $\mathbf{G}_l$.

Suppose that the aforementioned games sequentially compose $\{\mathbf{G}, \mathbf{G}_0, \mathbf{G}_1, \cdots, \mathbf{G}_{l_{max}}\}$, where $\mathbf{G}$ is the original attack game and $\mathbf{G}_{l_{max}}$ is the last one. Let $\mathcal{E}$ be the event that $\nu' = \nu$ in the original game $\mathbf{G}$ and $\mathcal{E}_l$ be the event $\nu' = \nu$ in $\mathbf{G}_l$ for $0 \le l \le l_{max}$. Then, $\epsilon_{\text{IND}} = \left|\Pr[\mathcal{E}] - \frac{1}{2}\right| = \left|\Pr[\mathcal{E}] - \Pr[\mathcal{E}_{l_{max}}]\right|$. From the triangle inequality, it follows that

$$\epsilon_{\text{IND}} \le \left|\Pr[\mathcal{E}] - \Pr[\mathcal{E}_0]\right| + \sum_{l=1}^{l_{max}} \left|\Pr[\mathcal{E}_{l-1}] - \Pr[\mathcal{E}_l]\right|.$$

In the following, we have concluded that $\left|\Pr[\mathcal{E}] - \Pr[\mathcal{E}_0]\right| \le \epsilon_{\text{DBDH}}$ and $\left|\Pr[\mathcal{E}_{l-1}] - \Pr[\mathcal{E}_l]\right| \le \epsilon_{\text{DL}}$ for $1 \le l \le l_{max}$ by proving Lemma 2 and Lemma 3, respectively. Then it is obvious that $\epsilon_{\text{IND}} \le \epsilon_{\text{DBDH}} + n\epsilon_{\text{DL}}$, which is negligible, and hence, the proposed anonymous CP-ABPRE construction is IND-sCP-CPA secure under the DBDH and D-Linear assumptions. □

Lemma 2. *Under the DBDH assumption, the difference between the advantage of $\mathcal{A}$ in $\mathbf{G}$ and that in $\mathbf{G}_0$ is negligible in λ. To be specific, we have $\left|Pr[\mathcal{E}] - Pr[\mathcal{E}_0]\right| \le \epsilon_{DBDH}$.*

Proof. Suppose that $\mathcal{A}$ has a difference ϵ_0 between its advantages in $\mathbf{G}$ and $\mathbf{G}_0$. We shall build a simulator $\mathcal{S}_0$ such that it plays the DBDH game with advantage ϵ_0. Suppose the DBDH challenger gives a challenge $[g, g^a, g^b, g^c, Z]$, where the component Z is either $\hat{e}(g,g)^{abc}$ or a random element in $\mathbb{G}_T$ with equal probability, the simulation proceeds as follows:

Init: $\mathcal{S}_0$ runs $\mathcal{A}$ and receives from $\mathcal{A}$ two challenge ciphertext policies $W_0^* = [W_{0,1}^*, W_{0,2}^*, \cdots, W_{0,n}^*]$ and $W_1^* = [W_{1,1}^*, W_{1,2}^*, \cdots, W_{1,n}^*]$. Then $\mathcal{S}_0$ flips a random coin $\nu \in \{0,1\}$.

Setup: In order to generate PK to $\mathcal{A}$, the simulator $\mathcal{S}_0$ computes $g_1 = g^a$, $g_2 = g^b$ and sets $Y = \hat{e}(g_1, g_2) = \hat{e}(g,g)^{ab}$, which implies that $y = a$. Also, $\mathcal{S}_0$ chooses $\mu \in_R \mathbb{Z}_p$ and computes $g_3 = g^{\mu}$. For $1 \le i \le n$, $\mathcal{S}_0$ chooses $\{\tau_{i,t}, a_{i,t}, b_{i,t} \in_R \mathbb{Z}_p\}_{1\le t\le n_i}$, and generates $\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1\le t\le n_i}$ as follows:

- If $v_{i,t} \in W_{v,i}^*$, then $\mathcal{S}_0$ sets $T_{i,t} = g^{\tau_{i,t}}$, and $A_{i,t} = T_{i,t}^{a_{i,t}}$, $B_{i,t} = T_{i,t}^{b_{i,t}}$;
- If $v_{i,t} \notin W_{v,i}^*$, then $\mathcal{S}_0$ sets $T_{i,t} = g_2^{\tau_{i,t}}$, and $A_{i,t} = T_{i,t}^{a_{i,t}}$, $B_{i,t} = T_{i,t}^{b_{i,t}}$.

Then $PK = \langle g, g_1, g_2, g_3, Y, \{\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1\le t\le n_i}\}_{1\le i\le n}\rangle$ and $\mathcal{S}_0$ sends $\mathcal{A}$ the public key PK.

Phase 1: The queries in the following are made by $\mathcal{A}$.

- **KeyGen query** $\mathcal{O}_{KeyGen}(L)$: Suppose $\mathcal{A}$ submits $L = [L_1, L_2, \cdots, L_n]$ in order to obtain a corresponding secret key. Only the case where $L \not\models W_0^* \wedge L \not\models W_1^*$ is taken into consideration. The reason is according to the definition of our security model, the message M_0 is equal to M_1 if $L \models W_0^* \wedge L \models W_1^*$. As a result, $\mathbf{G}$ is the same as $\mathbf{G}_0$, and there is no difference between advantages in $\mathbf{G}$ and $\mathbf{G}_0$ of the adversary $\mathcal{A}$. Accordingly, $\mathcal{S}_0$ just aborts and adopts a random guess.

  In the case that $L \not\models W_0^* \wedge L \not\models W_1^*$, there must be an integer $j \in \{1, 2, \cdots, n\}$ such that $L_j = v_{j,k_j}$ and $L_j \notin W_{v,j}^*$. For $1 \le i \le n$, $\mathcal{S}_0$ chooses $r_i', \hat{r}_i' \in \mathbb{Z}_p$, then sets $r_j = a + r_j'$, $\hat{r}_j = a + \hat{r}_j'$ and for every $i \ne j$, sets $r_i = r_i'$, $\hat{r}_i = \hat{r}_i'$. Finally, $\mathcal{S}_0$ computes $r = \sum_{i=1}^n r_i = a + \sum_{i=1}^n r_i'$, $\hat{r} = \sum_{i=1}^n \hat{r}_i = a + \sum_{i=1}^n \hat{r}_i'$. The components $[D_0, \widehat{D}_0]$ can be computed as

$$\begin{cases} D_0 = \prod_{i=1}^n g_2^{-r_i'} = g_2^{-\sum_{i=1}^n r_i'} = g_2^{a-r} = g_2^{y-r}, \\ \widehat{D}_0 = \prod_{i=1}^n g_2^{-\hat{r}_i'} = g_2^{-\sum_{i=1}^n \hat{r}_i'} = g_2^{a-\hat{r}} = g_2^{y-\hat{r}}. \end{cases}$$

  $\mathcal{S}_0$ also chooses $\bar{d} \in_R \mathbb{Z}_p$ and computes $\bar{D}_0 = g_3^{\bar{d}}$. In addition, $\mathcal{S}_0$ can compute the components $\left[D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\right]$ as follows:

– For j, $\mathcal{S}_0$ computes

$$\begin{cases} D_{\Delta,j} = g_2^{\hat{r}_j} T_{j,k_j}^{r'} = g_2^{\hat{r}'_j + a + r'\tau_{j,k_j}} = g_2^{\hat{r}'_j + \beta}, \\ D_{j,0} = g_2^{r_j} B_{j,k_j}^{a_{j,k_j}\lambda_j} = g_2^{r'_j + \gamma}, \\ D_{j,1} = g^{a_{j,k_j}\lambda_j} = \left(g^{\lambda_j}\right)^{a_{j,k_j}}, \\ D_{j,2} = g^{b_{j,k_j}\lambda_j} = \left(g^{\lambda_j}\right)^{b_{j,k_j}}, \end{cases}$$

where $\beta, \gamma \in_R \mathbb{Z}_p$ are chosen by $\mathcal{S}_0$. It sets $r' = \frac{\beta - a}{\tau_{j,k_j}}$ and $\lambda_j = \frac{\gamma - a}{a_{j,k_j} b_{j,k_j} \tau_{j,k_j}}$, hence $g^{\lambda_j} = g^{\frac{\gamma}{a_{j,k_j} b_{j,k_j} \tau_{j,k_j}}} g_1^{-\frac{1}{a_{j,k_j} b_{j,k_j} \tau_{j,k_j}}}$, which is utilized in the computation of $D_{j,1}$ and $D_{j,2}$.

– In addition, $\mathcal{S}_0$ sets $D_{\Delta,0} = g^{r'} = g^{\frac{\beta - a}{\tau_{j,k_j}}} = g^{\frac{\beta}{\tau_{j,k_j}}} g_1^{-\frac{1}{\tau_{j,k_j}}}$.

– For the case of $i \neq j$, $\mathcal{S}_0$ chooses $\lambda_i \in \mathbb{Z}_p$ and sets $\left[D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\right]$ as follows:

$$\begin{cases} D_{\Delta,i} = g_2^{\hat{r}_i} T_{i,k_i}^{r'} = g_2^{\hat{r}'_i} \left(g^{r'}\right)^{\tau_{i,k_i}}, \\ D_{i,0} = g_2^{r_i} B_{i,k_i}^{a_{i,k_i}\lambda_i} = g_2^{r'_i} g^{a_{i,k_i} b_{i,k_i} \lambda_i \tau_{i,k_i}}, \\ D_{i,1} = g^{a_{i,k_i}\lambda_i} = \left(g^{a_{i,k_i}}\right)^{\lambda_i}, \\ D_{i,2} = g^{b_{i,k_i}\lambda_i} = \left(g^{b_{i,k_i}}\right)^{\lambda_i}. \end{cases}$$

Finally, the generated secret key is $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1 \le i \le n} \rangle$.

- **RKGen query** $\mathcal{O}_{RKGen}(L, W)$: $\mathcal{A}$ submits $L = [L_1, L_2, \cdots, L_n]$ and a ciphertext policy $W = [W_1, W_2, \cdots, W_n]$ in a re-encryption key query. For the same reason as in the secret key query $\mathcal{O}_{RKGen}(\cdot)$, we consider only the case where $L \not\models W_0^* \wedge L \not\models W_1^*$. In this case, $\mathcal{S}_0$ submits L to $\mathcal{O}_{KeyGen}$ oracle and obtains $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1 \le i \le n} \rangle$. In order to obtain a re-encryption key corresponding to W, $\mathcal{S}_0$ chooses $d \in_R \mathbb{Z}_p$ and computes g^d, $\widehat{D}'_0 = \widehat{D}_0 (\bar{D}_0)^n$, $\bar{D}'_0 = \bar{D}_0 g_3^d$, $D'_{\Delta,i} = D_{\Delta,i} g_3^d$, $D'_{i,0} = D_{i,0} g_3^d$. Meanwhile, it sets $D'_0 = D_0$, $D'_{\Delta,0} = D_{\Delta,0}$, $D'_{i,1} = D_{i,1}$, $D'_{i,2} = D_{i,2}$, and computes $\mathbb{C}$ as the corresponding ciphertext of $E(g^d)$ with respect to W, that is, $\mathbb{C}$=Encrypt$(PK, E(g^d), W)$. Finally, the re-encryption key $RK_{L \to W}$ for W is $\langle L, D'_0, \widehat{D}'_0, \bar{D}'_0, D'_{\Delta,0}, \{D'_{\Delta,i}, D'_{i,0}, D'_{i,1}, D'_{i,2}\}_{1 \le i \le n}, \mathbb{C} \rangle$.
- **Reencrypt query** $\mathcal{O}_{ReEnc}(L, W', CT_W)$: For the same reason as in the query $\mathcal{O}_{KeyGen}(\cdot)$, only $L \not\models W_0^* \wedge L \not\models W_1^*$ is taken into account. In this case, $\mathcal{S}_0$ submits (L, W') to the re-encryption key query $\mathcal{O}_{RKGen}(\cdot)$ and achieved the re-encryption key $RK_{L \to W'}$. Then, $\mathcal{S}_0$ proceeds according to the corresponding algorithm.

Challenge: The adversary $\mathcal{A}$ submits two challenge messages M_0 and M_1. The simulator $\mathcal{S}_0$ sets $\tilde{C} = M_\nu Z$ and $C_0 = g^c$ that implies $s = c$ and computes $C_{\mathrm{RE}} = g_3^s = \left(g^\mu\right)^s = (g^s)^\mu = (g^c)^\mu$. In addition, $\mathcal{S}_0$ chooses $s', s'' \in_R \mathbb{Z}_p$, $\{\sigma_i \in_R \mathbb{G} | 1 \le i \le n\}$ such that $\prod_{i=1}^n \sigma_i = 1$, and sets $C_\Delta = Y^{s'}$ and $C'_0 = g^{s'}$. Then for $1 \le i \le n$ and $1 \le t \le n_i$, the simulator $\mathcal{S}_0$ computes $\left[C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\right]$ with respect to the access policy W_ν^* as follows:

- If $v_{i,t} \in W_{\nu,i}^*$, then $C_{i,t,\Delta} = \sigma_i T_{i,t}^{s'} = \sigma_i g^{\tau_{i,t} s'}$, $C_{i,t,1} = B_{i,t}^{s - s''} = T_{i,t}^{b_{i,t}(s - s'')} = g^{\tau_{i,t} b_{i,t} (s - s'')}$ and $C_{i,t,2} = A_{i,t}^{s''} = T_{i,t}^{a_{i,t} s''} = g^{\tau_{i,t} a_{i,t} s''}$;
- If $v_{i,t} \notin W_{\nu,i}^*$, then $\left[C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\right]$ are random elements in $\mathbb{G}$.

Obviously, $\mathcal{S}_0$ correctly generates the challenge ciphertext of M_ν with respect to W_ν^*, and that is $CT_{W_\nu^*} = \langle C_\Delta, C'_0, \tilde{C}, C_0, C_{\mathrm{RE}}, \{\{C_{i,t,\Delta}, C_{i,t,1}, C_{i,t,2}\}_{1 \le t \le n_i}\}_{1 \le i \le n} \rangle$.

Phase 2: The simulator $\mathcal{S}_0$ proceeds as in Phase 1.

Guess: For ν, the adversary $\mathcal{A}$ outputs a guess ν'. Then, $\mathcal{S}_0$ outputs 1 if $\nu' = \nu$ and 0 otherwise. According to our assumption, there is a difference ϵ_0 between the probability of $\mathcal{A}$ correctly guessing ν in $\mathbf{G}$ and that in $\mathbf{G}_0$. Note that when $Z = \hat{e}(g, g)^{abc}$, $\mathcal{A}$ is in $\mathbf{G}$, and when Z is a random element in $\mathbb{G}_T$, $\mathcal{A}$ is in $\mathbf{G}_0$. Hence, $\mathcal{S}_0$ has advantage ϵ_0 in the DBDH game.

It follows from the DBDH assumption that $\epsilon_0 \le \epsilon_{\mathrm{DBDH}}$. Therefore, the difference between the advantage of $\mathcal{A}$ in $\mathbf{G}$ and that in $\mathbf{G}_0$ is negligible in λ. To be specific, we have $\left|\Pr[\mathcal{E}] - \Pr[\mathcal{E}_0]\right| = \epsilon_0 \le \epsilon_{\mathrm{DBDH}}$. □

Lemma 3. *Under the D-Linear assumption, the difference between advantages of $\mathcal{A}$ in $\mathbf{G}_{l-1}$ and $\mathbf{G}_l$ is negligible in λ for $1 \le l \le l_{max}$. To be specific, $\left|Pr[\mathcal{E}_{l-1}] - Pr[\mathcal{E}_l]\right| \le \epsilon_{DL}$ for $1 \le l \le l_{max}$.*

Proof. Suppose that the adversary $\mathcal{A}$ has non-negligible difference ϵ_l between its advantage in $\mathbf{G}_{l-1}$ and that in $\mathbf{G}_l$. In the following, we shall build a simulator $\mathcal{S}_l$ that can get advantage ϵ_l in the D-Linear game. Suppose the D-Linear challenger gives a challenge $[g, g^{z_1}, g^{z_2}, Z, g^{z_2 z_4}, g^{z_3 + z_4}]$, where Z is either $g^{z_1 z_3}$ or a random element in $\mathbb{G}$ with the same probability, $\mathcal{S}_l$ creates the following simulation.

As mentioned in Theorem 1, the ciphertext components $\{C_{i_l,t_l,\Delta}, C_{i_l,t_l,1}, C_{i_l,t_l,2}\}$ with $k_{i_l} = t_l$ in game $\mathbf{G}_{l-1}$ are computed as in the real scheme, while the components in game $\mathbf{G}_l$ are random elements from the group $\mathbb{G}$ for any random coin ν. Without loss of generality, assume $(v_{i_l,t_l} \notin W_{0,i_l}^* \wedge v_{i_l,t_l} \in W_{1,i_l}^*)$. The simulation proceeds as follows:

Init: $\mathcal{S}_l$ runs $\mathcal{A}$ and receives two challenge ciphertext policies $W_0^* = [W_{0,1}^*, W_{0,2}^*, \cdots, W_{0,n}^*]$ and $W_1^* = [W_{1,1}^*, W_{1,2}^*, \cdots, W_{1,n}^*]$ from $\mathcal{A}$. Subsequently, $\mathcal{S}_l$ randomly flips a coin $\nu \in \{0, 1\}$. If $\nu = 0$, $\mathcal{S}_l$ just aborts and uses a

random guess. The reason is that if $\nu = 0$ where ($v_{i_l,t_l} \notin W^*_{0,i_l} \wedge v_{i_l,t_l} \in W^*_{1,i_l}$), we have $\mathbf{G}_{l-1} = \mathbf{G}_l$ according to the definition of games. Then, the challenge ciphertext in $\mathbf{G}_{l-1}$ is the same as that in $\mathbf{G}_l$; hence, $\mathcal{A}$ has the same advantage in $\mathbf{G}_{l-1}$ as that in $\mathbf{G}_l$. Accordingly, assume $\nu = 1$ in the following.

Setup: In order to generate PK for $\mathcal{A}$, the simulator $\mathcal{S}_l$ sets $g_1 = g^{z_1}$, which implies that $y = z_1$. Then, $\mathcal{S}_l$ chooses $\mu_2, \mu_3 \in_R \mathbb{Z}_p$ and computes $g_2 = g^{\mu_2}$, $g_3 = g^{\mu_3}$ and $Y = \hat{e}(g_1, g_2)$. For each attribute i with $1 \le i \le n$, $\mathcal{S}_l$ chooses $\{\tau_{i,t} \in_R \mathbb{Z}_p\}_{1\le t\le n_i}$ and generates $\{T_{i,t}\}_{1\le t\le n_i}$ as $T_{i,t} = g^{\tau_{i,t}}$ if $v_{i,t} \in W^*_{\nu,i}$, otherwise, $T_{i,t} = (g^{z_1})^{\tau_{i,t}}$. In addition, for $1 \le i \le n$, $\mathcal{S}_l$ chooses $\{a_{i,t}, b_{i,t} \in_R \mathbb{Z}_p\}_{1\le t\le n_i}$ and generates $\{A_{i,t}, B_{i,t}\}_{1\le t\le n_i}$ as follows:

- In the case of a_{i_l,t_l}, b_{i_l,t_l}, $\mathcal{S}_l$ sets $a_{i_l,t_l} = z_1$, $b_{i_l,t_l} = z_2$, and computes A_{i_l,t_l} and B_{i_l,t_l} without knowing z_1 and z_2 as follows:

$$\begin{cases} A_{i_l,t_l} = T_{i_l,t_l}^{a_{i_l,t_l}} = \left(g^{\tau_{i_l,t_l}}\right)^{a_{i_l,t_l}} = (g^{z_1})^{\tau_{i_l,t_l}}, \\ B_{i_l,t_l} = T_{i_l,t_l}^{b_{i_l,t_l}} = \left(g^{\tau_{i_l,t_l}}\right)^{b_{i_l,t_l}} = (g^{z_2})^{\tau_{i_l,t_l}}. \end{cases}$$

- In other cases, $\mathcal{S}_l$ computes $A_{i,t} = T_{i,t}^{a_{i,t}}$ and $B_{i,t} = T_{i,t}^{b_{i,t}}$ in a straightforward way.

Then $PK = \langle g, g_1, g_2, g_3, Y, \{\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1\le t\le n_i}\}_{1\le i\le n}\rangle$ and $\mathcal{S}_l$ sends PK to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ makes queries as follows.

- **KeyGen query** $\mathcal{O}_{KeyGen}(L)$: Suppose that $\mathcal{A}$ submits $L = [L_1, L_2, \cdots, L_n]$ to this query. In order to compute the secret key $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1\le i\le n}\rangle$, $\mathcal{S}_l$ takes into account the following two circumstances:

  Case 1. $L_{i_l} \neq v_{i_l,t_l}$: $\mathcal{S}_l$ chooses $r', \bar{d} \in_R \mathbb{Z}_p$ and $\{r_i, \hat{r}_i, \lambda_i \in_R \mathbb{Z}_p\}_{1\le i\le n}$, sets $r = \sum_{i=1}^n r_i$, $\hat{r} = \sum_{i=1}^n \hat{r}_i$, and computes the secret components $[D_0, \widehat{D}_0, D_{\Delta,0}]$ as follows:

$$\begin{cases} D_0 = g_2^{y-r} = (g^{\mu_2})^{z_1} g_2^{-r} = (g^{z_1})^{\mu_2} g_2^{-r}, \\ \widehat{D}_0 = g_2^{y-\hat{r}} = (g^{\mu_2})^{z_1} g_2^{-\hat{r}} = (g^{z_1})^{\mu_2} g_2^{-\hat{r}}, \\ \bar{D}_0 = g_3^{\bar{d}}, \\ D_{\Delta,0} = g^{r'}. \end{cases}$$

  For $1 \le i \le n$, $\mathcal{S}_l$ computes the secret components

$$\begin{aligned} &[D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}] \\ &= \left[g_2^{\hat{r}_i} T_{i,k_i}^{r'}, g_2^{r_i} B_{i,k_i}^{\lambda_i a_{i,k_i}}, g^{\lambda_i a_{i,k_i}}, g^{\lambda_i b_{i,k_i}}\right] \end{aligned}$$

  Case 2. $L_{i_l} = v_{i_l,t_l}$: Note that $a_{i_l,t_l} = z_1$, $b_{i_l,t_l} = z_2$ and $v_{i_l,t_l} \in W^*_{\nu,i_l}$ in this case. $\mathcal{S}_l$ does the following:

  (1) $\mathcal{S}_l$ computes $\left[D_{\Delta,i_l}, D_{i_l,0}, D_{i_l,1}, D_{i_l,2}\right]$ as

$$\begin{cases} D_{\Delta,i_l} = g_2^{\hat{r}_{i_l}} T_{i_l,t_l}^{r'} = g^{\mu_2 \hat{r}_{i_l}} g^{\tau_{i_l,t_l} r'} = g^{\hat{r}'_{i_l}}, \\ D_{i_l,0} = g_2^{r_{i_l}} B_{i_l,t_l}^{\lambda_{i_l} a_{i_l,t_l}} = g^{r'_{i_l}}, \\ D_{i_l,1} = g^{\lambda_{i_l} a_{i_l,t_l}} = \left(g^{a_{i_l,t_l}}\right)^{\lambda_{i_l}} = (g^{z_1})^{\lambda_{i_l}}, \\ D_{i_l,2} = g^{\lambda_{i_l} b_{i_l,t_l}} = \left(g^{b_{i_l,t_l}}\right)^{\lambda_{i_l}} = (g^{z_2})^{\lambda_{i_l}}, \end{cases}$$

  where $r', \lambda_{i_l}, \hat{r}'_{i_l}, r'_{i_l} \in_R \mathbb{Z}_p$ are chosen by $\mathcal{S}_l$, and it sets $\hat{r}_{i_l} = \frac{\hat{r}'_{i_l} - r'\tau_{i_l,t_l}}{\mu_2}$ and $r_{i_l} = \frac{r'_{i_l} - \lambda_{i_l} a_{i_l,t_l} b_{i_l,t_l} \tau_{i_l,t_l}}{\mu_2} = \frac{r'_{i_l} - \lambda_{i_l} z_1 z_2 \tau_{i_l,t_l}}{\mu_2}$.

  (2) Because $L_{i_l} = v_{i_l,t_l} \wedge v_{i_l,t_l} \notin W^*_{0,i_l}$, it follows that $L \not\models W^*_0$. According to the definition of our security model, we have $L \not\models W^*_1$, that is, $L \not\models W^*_\nu$. Consequently, there must exist an integer $j \neq i_l$ satisfying $j \in \{1, 2, \cdots, n\}$ such that $L_j = v_{j,k_j} \wedge v_{j,k_j} \notin W^*_{\nu,j}$ in that $v_{i_l,t_l} \in W^*_{1,i_l} = W^*_{\nu,i_l}$. Then $\mathcal{S}_l$ sets $\hat{r}_j = \hat{r}'_j + \frac{r'\tau_{i_l,t_l}}{\mu_2}$ and $r_j = r'_j + \frac{\lambda_{i_l} a_{i_l,t_l} b_{i_l,t_l} \tau_{i_l,t_l}}{\mu_2} = r'_j + \frac{\lambda_{i_l} z_1 z_2 \tau_{i_l,t_l}}{\mu_2}$, where $\hat{r}'_j$ and r'_j are chosen by $\mathcal{S}_l$. It computes the secret components $\left[D_{\Delta,j}, D_{j,0}, D_{j,1}, D_{j,2}\right]$:

$$\begin{cases} D_{\Delta,j} = g_2^{\hat{r}_j} T_{j,k_j}^{r'} = g_2^{\hat{r}'_j} g^{r'\tau_{i_l,t_l}} T_{j,k_j}^{r'}, \\ D_{j,0} = g_2^{r_j} B_{j,k_j}^{\lambda_j a_{j,k_j}} \\ \qquad = g_2^{r'_j} g_1^{z_2 \tau_{i_l,t_l} \lambda_{i_l} + \tau_{j,k_j} \lambda_j a_{j,k_j} b_{j,k_j}} \\ \qquad = g_2^{r'_j} g_1^{\lambda'_j}, \\ D_{j,1} = g^{\lambda_j a_{j,k_j}} = \left(g^{\lambda_j}\right)^{a_{j,k_j}}, \\ D_{j,2} = g^{\lambda_j b_{j,k_j}} = \left(g^{\lambda_j}\right)^{b_{j,k_j}}, \end{cases}$$

  where r', λ_{i_l} are known to $\mathcal{S}_l$ as mentioned previously, and $\lambda'_j \in_R \mathbb{Z}_p$ are chosen by $\mathcal{S}_l$ such that $\lambda_j = \frac{\lambda'_j}{\tau_{j,k_j} a_{j,k_j} b_{j,k_j}} - \frac{z_2 \tau_{i_l,t_l} \lambda_{i_l}}{\tau_{j,k_j} a_{j,k_j} b_{j,k_j}}$. Accordingly, $\mathcal{S}_l$ can successfully compute $g^{\lambda_j} = g^{\frac{\lambda'_j}{\tau_{j,k_j} a_{j,k_j} b_{j,k_j}}} - g_2^{\frac{\tau_{i_l,t_l} \lambda_{i_l}}{\tau_{j,k_j} a_{j,k_j} b_{j,k_j}}}$ without knowing z_2.

  (3) For $i \neq i_l, j$, $\mathcal{S}_l$ computes the secret components $[D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}]$ as in the first case in a straightforward way.

  (4) Finally, $\mathcal{S}_l$ computes the secret components $[D_0, \widehat{D}_0, D_{\Delta,0}]$. Note that

$$r = \sum_{i=1}^{n} r_i = r_{i_l} + r_j + \sum_{i \neq i_l, j} r_i$$
$$= \frac{r'_{i_l} - \lambda_{i_l} z_1 z_2 \tau_{i_l,t_l}}{\mu_2} + r'_j + \frac{\lambda_{i_l} z_1 z_2 \tau_{i_l,t_l}}{\mu_2}$$
$$+ \sum_{i \neq i_l, j} r_i = \frac{r'_{i_l}}{\mu_2} + r'_j + \sum_{i \neq i_l, j} r_i.$$

Hence, $\mathcal{S}_l$ can compute g^r, and $D_0 = g_2^{y-r} = ((g^{z_1})\, g^{-r})^{\mu_2}$. Similarly,

$$\hat{r} = \sum_{i=1}^{n} \hat{r}_i = \frac{\hat{r}'_{i_l}}{\mu_2} + \hat{r}'_j + \sum_{i \neq i_l, j} \hat{r}_i,$$

and $\mathcal{S}_l$ can compute $g^{\hat{r}}$ and $\widehat{D}_0 = g_2^{y-\hat{r}} = g^{\mu_2(z_1-\hat{r})} = \left((g^{z_1})\, g^{-\hat{r}}\right)^{\mu_2}$. It is straightforward for $\mathcal{S}_l$ to compute $\bar{D}_0 = g_3^{\bar{d}}$ and $D_{\Delta,0} = g^{r'}$ owing to it knows $\bar{d}$ and r'.

- **RKGen query** $\mathcal{O}_{RKGen}(L, W)$: The simulator $\mathcal{S}_l$ proceeds as in Lemma 2.
- **Reencrypt query** $\mathcal{O}_{ReEnc}(L, W', CT_W)$: $\mathcal{S}_l$ proceeds as in Lemma 2.

Challenge: The adversary $\mathcal{A}$ submits two messages M_0 and M_1. The simulator $\mathcal{S}_l$ sets $C_0 = g^{z_3+z_4}$ that implies $s = z_3 + z_4$ and computes $C_{\mathrm{RE}} = g_3^s = \left(g^{\mu_3}\right)^s = (g^s)^{\mu_3} = \left(g^{z_3+z_4}\right)^{\mu_3}$. If $L \not\models W_0^* \wedge L \not\models W_1^*$, $\mathcal{S}_l$ sets $\tilde{C}$ to be a random element in $\mathbb{G}_T$ and otherwise sets $\tilde{C} = M_\nu \hat{e}(g^{z_1}, g^{z_3+z_4})^{\mu_2} = M_\nu \hat{e}(g_1, g_2)^s$. In addition, $\mathcal{S}_l$ chooses $s' \in_R \mathbb{Z}_p$, $\{\sigma_i \in_R \mathbb{G} | 1 \leq i \leq n\}$ such that $\prod_{i=1}^n \sigma_i = 1$ and computes $C_\Delta = Y^{s'}$ and $C'_0 = g^{s'}$. Then for $1 \leq i \leq n$ and $1 \leq t \leq n_i$, $\mathcal{S}_l$ computes $C_{i,t,\Delta} = \sigma_i T_{i,t}^{s'}$. Also, $\mathcal{S}_l$ generates the ciphertext components $\{\{C_{i,t,1}, C_{i,t,2}\}_{1 \leq t \leq n_i}\}_{1 \leq i \leq n}$ as in game $\mathbf{G}_{l-1}$ except that $\{C_{i_l,t_l,1}, C_{i_l,t_l,2}\}$ in the challenge ciphertext are computed as follows without knowing $z_2 z_4$ and $z_1 z_3$:

$$\begin{cases} C_{i_l,t_l,1} = B_{i_l,t_l}^{s-s''} = g^{\tau_{i_l,t_l} z_2 (s-s'')} = (g^{z_2 z_4})^{\tau_{i_l,t_l}}, \\ C_{i_l,t_l,2} = A_{i_l,t_l}^{s''} = g^{\tau_{i_l,t_l} z_1 s''} = Z^{\tau_{i_l,t_l}}. \end{cases}$$

This implies that $s'' = z_3$. If $Z = g^{z_1 z_3}$, the components are well-formed and $\mathcal{A}$ is in $\mathbf{G}_{l-1}$, otherwise Z is a random element from $\mathbb{G}_T$ and $\mathcal{A}$ is in $\mathbf{G}_l$.

Phase 2: $\mathcal{S}_l$ proceeds as in Phase 1.

Guess: For ν, the adversary $\mathcal{A}$ outputs a guess ν'. $\mathcal{S}_l$ outputs 1 if $\nu' = \nu$ and 0 otherwise. Based on our assumption, there is a difference ϵ_l between the probability of $\mathcal{A}$ correctly guessing ν in $\mathbf{G}_{l-1}$ and that in $\mathbf{G}_l$. Note that if $Z = g^{z_1 z_3}$, $\mathcal{A}$ is in $\mathbf{G}_{l-1}$, and if Z is a random element in $\mathbb{G}_T$, $\mathcal{A}$ is in $\mathbf{G}_l$. Accordingly, in the D-Linear game, $\mathcal{S}_l$ has advantage ϵ_l. It follows from the D-Linear assumption that $\epsilon_l \leq \epsilon_{\mathrm{DL}}$. Hence, the difference between advantages of $\mathcal{A}$ in $\mathbf{G}_{l-1}$ and $\mathbf{G}_l$ is negligible in λ. Specifically, $|\Pr[\mathcal{E}_{l-1}] - \Pr[\mathcal{E}_l]| = \epsilon_l \leq \epsilon_{\mathrm{DL}}$ for $1 \leq l \leq l_{max}$. □

Theorem 2. *Under the CBDH assumption, the anonymous CP-ABPRE scheme is sMKS secure and hence has Master Key Security without random oracles.*

Proof. Under the CBDH assumption, we prove that the proposed anonymous CP-ABPRE scheme has master key security in the sMKS model without random oracles. To be specific, if there exists a polynomial-time adversary $\mathcal{A}$, which can attack the proposed scheme with advantage ϵ_{MKS} in the sMKS model, then it follows that $\epsilon_{\mathrm{MKS}} \leq \epsilon_{\mathrm{CBDH}}$, where ϵ_{CBDH} denotes the advantage of a solver of a CBDH challenge. Suppose the CBDH challenger gives a challenge $[g, A, B, C] = [g, g^a, g^b, g^c,]$, the simulation is described in the following.

Init: $\mathcal{S}$ runs the adversary $\mathcal{A}$ and receives an attribute list L^* from $\mathcal{A}$.

Setup: In order to generate PK for $\mathcal{A}$, the simulator $\mathcal{S}$ computes $g_1 = A$, $g_2 = B$, $g_3 = B$, and computes $Y = \hat{e}(g_1, g_2) = \hat{e}(g, g)^{ab}$, which implies that $y = a$. Also, for each attribute i with $1 \leq i \leq n$, $\mathcal{S}$ chooses $\{\tau_{i,t}, a_{i,t}, b_{i,t} \in_R \mathbb{Z}_p\}_{1 \leq t \leq n_i}$, and generates $\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1 \leq t \leq n_i}$ as follows:

$$T_{i,t} = \begin{cases} g^{\tau_{i,t}}, & \text{if } t = j \\ B^{\tau_{i,t}}, & \text{if } t \neq j \end{cases}, \; A_{i,t} = T_{i,t}^{a_{i,t}}, B_{i,t} = T_{i,t}^{b_{i,t}}.$$

Then $PK = \langle g, g_1, g_2, g_3, Y, \{\{T_{i,t}, A_{i,t}, B_{i,t}\}_{1 \leq t \leq n_i}\}_{1 \leq i \leq n} \rangle$ and $\mathcal{S}$ sends PK to $\mathcal{A}$.

Queries: The following queries are made by $\mathcal{A}$.

- **KeyGen query** $\mathcal{O}_{KeyGen}(L)$: The adversary $\mathcal{A}$ submits an attribute list $L = [L_1, L_2, \cdots, L_n]$. The attribute list must satisfy $L \neq L^*$, and otherwise $\mathcal{S}$ just aborts, and a random guess is taken.

  In the case of $L \neq L^*$, there is some j satisfying $L_j \neq L_j^*$. After j is determined, $\mathcal{S}$ just generates SK_L in the method in Lemma 2.
- **RKGen query** $\mathcal{O}_{RKGen}(L, W)$: Suppose $\mathcal{A}$ submits $L = [L_1, L_2, \cdots, L_n]$ and $W = [W_1, W_2, \cdots, W_n]$. There are two circumstances to be taken into account.

  Case 1. $L \neq L^*$: $\mathcal{S}$ submits L to a query $\mathcal{O}_{KeyGen}(\cdot)$ and obtains a secret key $SK_L = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1 \leq i \leq n} \rangle$. In order to obtain a re-encryption key associated with W, $\mathcal{S}$ proceeds as in Lemma 2.

  Case 2. $L = L^*$: $\mathcal{S}$ computes the re-encryption key by doing the following:

  (1) For $1 \leq i \leq n$, $\mathcal{S}$ chooses $r'_i, \hat{r}'_i \in \mathbb{Z}_p$, and computes $r_i = \frac{a}{n} + r'_i$, $\hat{r}_i = \frac{a}{n} + \hat{r}'_i$. Then it sets $r = \sum_{i=1}^n r_i = a + \sum_{i=1}^n r'_i$, $\hat{r} = \sum_{i=1}^n \hat{r}_i = a + \sum_{i=1}^n \hat{r}'_i$.

Table I. Comparison of CP-ABE Schemes

Schemes	Expressiveness	Complexity assumption	Type of bilinear group	Anonymity	Policy update	Attribute matching detection cost	
						Re-encryption	Decryption
BSW [3]	Type 1[a]	Generic group model	Any	×	×	×	$\mathcal{O}(1)$
LCLS [12]	Type 2[b]	ADBDH[e], CTDH[f]	Any	×	✓	$\mathcal{O}(1)$	$\mathcal{O}(1)$
LHC [13]	Type 3[c]	DBDH, CBDH	Any	×	✓	$\mathcal{O}(1)$	$\mathcal{O}(1)$
YK [26]	Type4[d]	DBDH, Unforgeability[g]	Any	×[i]	✓	$\mathcal{O}(1)$	$\mathcal{O}(1)$
NYO [7]	Type 3	DBDH, D-Linear	Any	✓	×	×	$\mathcal{O}(n)$
LDL1 [9]	Type 3	New assumptions[h]	Composite order $N = p_1p_2p_3$	✓	×	×	$\mathcal{O}(n)$
LDL2 [10]	Type 4	New assumptions	Composite order $N = p_1p_2p_3p_4$	✓	×	×	$\mathcal{O}(n)$
ZCLW [11]	Type 3	DBDH, D-Linear	Any	✓	×	×	$\mathcal{O}(1)$
This work	Type 3	DBDH, D-Linear, CBDH	Any	✓	✓	$\mathcal{O}(1)$	$\mathcal{O}(1)$

[a] Tree-based structure.
[b] AND-gates on positive and negative attributes with wildcards.
[c] AND-gates on multi-valued attributes with wildcards.
[d] Linear secret sharing schemes.
[e] Augment Decisional Bilinear Diffie–Hellman Assumption (ADBDH).
[f] Complex Triple Diffie–Hellman Assumption (CTDH).
[g] Strong unforgeability of one-time signature.
[h] New assumptions based on composite order groups.
[i] The scheme only supports payload-hiding for original and re-encrypted ciphertexts, which just requires that a ciphertext conceal the plaintext.

Security Comm. Networks (2016)
DOI: 10.1002/sec

The components $[D'_0, \widehat{D}'_0]$ of the re-encryption key is computed as

$$\begin{cases} D'_0 = \prod_{i=1}^{n} B^{-r'_i} = g_2^{a-r} = g_2^{y-r}, \\ \widehat{D}'_0 = \prod_{i=1}^{n} B^{-\hat{r}'_i} = g_2^{a-\hat{r}} = g_2^{y-\hat{r}}. \end{cases}$$

(2) $\mathcal{S}$ chooses $\bar{d}, r' \in \mathbb{Z}_p$, sets $\bar{D}_0 = g_3^{\bar{d}}$ and $D'_{\Delta,0} = g^{r'}$.

(3) $\mathcal{S}$ chooses $d' \in \mathbb{Z}_p$ and sets $d = -\frac{a}{n} + d'$ that means $g^d = g^{-\frac{a}{n}+d'} = A^{-\frac{1}{n}} g^{d'}$. For $1 \le i \le n$, suppose the indexes satisfy $L_i = v_{i,k_i}$, $\mathcal{S}$ chooses $\lambda_i \in \mathbb{Z}_p$ and computes

$$\begin{cases} D'_{\Delta,i} = D_{\Delta,i} g_3^d = g_2^{\hat{r}_i} T_{i,k_i}^{r'} g_3^d = B^{\hat{r}'_i + d'} T_{i,k_i}^{r'}, \\ D'_{i,0} = D_{i,0} g_3^d = B^{r'_i + d'} B_{i,k_i}^{\lambda_i a_{i,k_i}}, \\ D'_{i,1} = g^{\lambda_i a_{i,k_i}} = \left(g^{a_{i,k_i}}\right)^{\lambda_i}, \\ D'_{i,2} = g^{\lambda_i b_{i,k_i}} = \left(g^{b_{i,k_i}}\right)^{\lambda_i}. \end{cases}$$

(4) $\mathcal{S}$ computes $\mathbb{C}$ as in Case 1.

- **Reencrypt query** $\mathcal{O}_{ReEnc}(L, W', CT_W)$: $\mathcal{S}_l$ proceeds as in Lemma 2.

Output: $\mathcal{A}$ returns the secret key $SK_{L^*} = \langle D_0, \widehat{D}_0, \bar{D}_0, D_{\Delta,0}, \{D_{\Delta,i}, D_{i,0}, D_{i,1}, D_{i,2}\}_{1 \le i \le n} \rangle$ for the attribute list L^*. If SK_{L^*} is valid, then the following equation holds:

$$\hat{e}\left(g^s, \widehat{D}_0\right) \prod_{i=1}^{n} \frac{\hat{e}(g^s, D_{\Delta,i})}{\hat{e}(X_i, D_{\Delta,0})} = \hat{e}(g_1, g_2)^s,$$

where $s \in_R \mathbb{Z}_p$ and $X_i = T_{i,k_i}^s$ with $L_i^* = v_{i,k_i}$.

Suppose $L_i^* = v_{i,k_i}$, $\mathcal{A}$ sets $T_i = C^{\tau_{i,k_i}}$, and then, it can solve the CBDH problem by outputting

$$\hat{e}(C, \widehat{D}_0) \prod_{i=1}^{n} \frac{\hat{e}(C, D_{\Delta,i})}{\hat{e}(T_i, D_{\Delta,0})} = \hat{e}(g, g)^{abc}.$$

Therefore, $\mathcal{S}$ has advantage ϵ_{MKS} in the CBDH game. It follows from the CBDH assumption that $\epsilon_{\text{MKS}} \le \epsilon_{\text{CBDH}}$, which is negligible in the security parameter. □

5.3. Performance analysis

To precisely evaluate the performance of the proposed anonymous CP-ABPRE scheme, we implement it based on the Stanford Pairing-Based Crypto library (version 0.5.12) [29] on a Linux machine with 2.40 GHz Intel Core(TM)2 Duo CPU and 2GB of RAM. The order p of groups $\mathbb{G}$ and $\mathbb{G}_T$ is set as a prime of length 160 bits. In our experiments, we consider the worst case of the access policy, which ensures that all the ciphertext components are involved in decryption. Specifically, we generate 100 distinct access policies in the form of $(W_1 \wedge W_2 \wedge \cdots \wedge W_n)$ with n increasing from 1 to 100. For each access policy, the experiment is repeated for 30 times, and the final results adopt the average value.

The properties of previous CP-ABE schemes [3,7,9–13,26] and the new scheme are compared in Table I, where n represents the total number of attributes in universe. It's noted that, anonymity can be applied to protect users' attribute privacy and access policy update means the supporting of access rights delegation. From the comparison in Table I, it is clear that only the proposed construction can protect users' attribute privacy and support access rights delegation, simultaneously.

Although protecting users' attribute privacy, the schemes [7,9–11] fail to support access policy update. In these schemes, only the scheme [11] adopts the novel technique match-then-decrypt, and the schemes [7,9,10] need $\mathcal{O}(n)$ paring operations in attribute matching detection

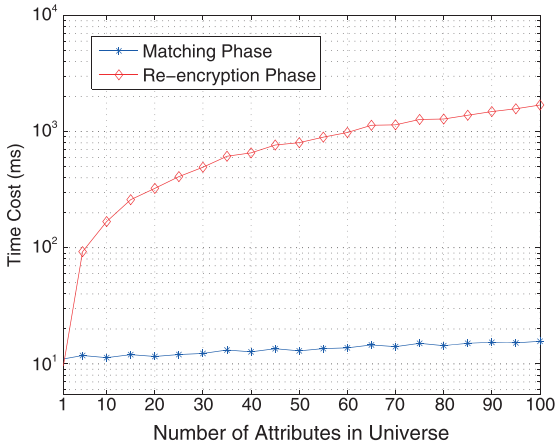

Figure 2. Re-encryption cost for anonymous CP-ABPRE.

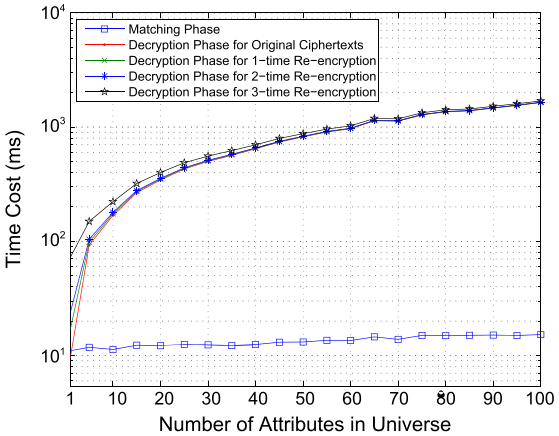

Figure 3. Decryption cost for anonymous CP-ABPRE.

of decryption. Besides, in order to achieve full security, the schemes [9,10] are constructed based on composite order groups and the security proofs are given under many new assumptions. The schemes [12,13,26] support access policy update; however, these schemes cannot protect users' attribute privacy. In the proposed scheme, the novel techniques match-then-re-encrypt and match-then-decrypt are respectively adopted in proxy re-encryption and decryption to improve the corresponding efficiency. It is shown in Table I that our scheme only needs $\mathcal{O}(1)$ paring operations in attribute matching detections of proxy re-encryption and decryption while protecting users' attribute privacy. The simulation results are shown in Figure 2 and Figure 3. It easily follows that the proposed scheme can efficiently implement attribute matching detections before proxy re-encryption and decryption, which alleviates the computation burden of proxy servers and users, especially in the case of invalid ciphertexts. Specifically, if a user's attributes associated with his secret key do not match the ciphertext policy, the user only requires $\mathcal{O}(1)$ pairing operations in proxy re-encryption and decryption without knowing the access policy in the proposed anonymous CP-ABPRE scheme. Generally, only the proposed scheme can simultaneously protect users' attribute privacy and support access policy update, and it is secure and efficient.

6. CONCLUSION

In this paper, we simultaneously consider two important requirements of ABE: anonymity and access policy update. We formalize the notion and security model of anonymous CP-ABPRE and present a concrete scheme. Our construction is provably secure without random oracles. We achieve this by introducing a novel technique called match-then-re-encrypt. The results in simulation experiments indicate that the proposed scheme can be utilized to realize anonymous attribute-based access control supporting access rights delegation of data on untrusted cloud.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (grant nos 61402366, 61272455, 61272457, 61472091, 61272037, and 61472472), the Natural Science Foundation of Shaanxi Province (grant no. 2015JQ6236), the Doctoral Fund of Ministry of Education of China (grant no. 20130203110004) and the Scientific Research Program funded by Shaanxi Provincial Education Department (grant no. 15JK1686). Yinghui Zhang is supported by New Star Team of Xi'an University of Posts and Telecommunications.

REFERENCES

1. Sahai A, Waters B. Fuzzy identity-based encryption. In *Advances in cryptology-EUROCRYPT'05*, vol. 3494, LNCS. Springer: Heidelberg, 2005; 557–557.
2. Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. *CCS'06*, ACM, New York, 2006; 89–98.
3. Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. *SP'07*, IEEE, Oakland, 2007; 321–334.
4. Kapadia A, Tsang PP, Smith SW. Attribute-based publishing with hidden credentials and hidden policies. *NDSS'07*, The Internet Society, San Diego, 2007; 179–192.
5. Boneh D, Waters B. Conjunctive, subset, and range queries on encrypted data. In *TCC'07*, vol. 4392, LNCS. Springer: Heidelberg, 2007; 535–554.
6. Katz J, Sahai A, Waters B. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Advances in Cryptology-EUROCRYPT'08*, vol. 4965, LNCS. Springer: Heidelberg, 2008; 146–162.
7. Nishide T, Yoneyama K, Ohta K. Abe with partially hidden encryptor-specified access structure. In *ACNS'08*, vol. 5037, LNCS. Springer: Heidelberg, 2008; 111–129.
8. Li J, Ren K, Zhu B, Wan Z. Privacy-aware attribute-based encryption with user accountability. In *ISC'09*, vol. 5735, LNCS. Springer: Heidelberg, 2009; 347–362.
9. Lai J, Deng RH, Li Y. Fully secure cipertext-policy hiding cp-abe. *ISPEC'11*, Springer, Heidelberg, 2011; 24–39.
10. Lai J, Deng RH, Li Y. Expressive cp-abe with partially hidden access structures. *ASIACCS'12*, ACM, Seoul, 2012; 18–19.
11. Zhang Y, Chen X, Li J, Wong DS, Li H. Anonymous attribute-based encryption supporting efficient decryption test. *ASIACCS'13*, ACM, Hangzhou, 2013; 511–516.
12. Liang X, Cao Z, Lin H, Shao J. Attribute based proxy re-encryption with delegating capabilities. *ASIACCS'09*, ACM, New York, 2009; 276–286.
13. Luo S, Hu J, Chen Z. Ciphertext policy attribute-based proxy re-encryption. In *ICICS'10*, vol. 6476, LNCS. Springer: Heidelberg, 2010; 401–415.
14. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. *CCS'07*, ACM, New York, 2007; 195–203.
15. Cheung L, Newport C. Provably secure ciphertext policy abe. *CCS'07*, ACM, New York, 2007; 456–465.
16. Goyal V, Jain A, Pandey O, Sahai A. Bounded ciphertext policy attribute based encryption. In *ICALP'08*, vol. 5126, LNCS. Springer: Heidelberg, 2008; 579–591.
17. Hohenberger S, Waters B. Attribute-based encryption with fast decryption. In *Public-Key Cryptography–PKC 2013*. Springer: Heidelberg, 2013; 162–179.

18. Garg S, Gentry C, Halevi S, Sahai A, Waters B. Attribute-based encryption for circuits from multilinear maps. In *Advances in Cryptology-CRYPTO'13*, vol. 8042, LNCS. Springer: Heidelberg, 2013; 479–499.
19. Zhang Y, Chen X, Li J, Li H, Li F. Attribute-based data sharing with flexible and direct revocation in cloud computing. *KSII Transactions on Internet & Information Systems* 2014; **8**(11): 4028–4049.
20. Zhang Y, Zheng D, Chen X, Li J, Li H. Computationally efficient ciphertext-policy attribute-based encryption with constant-size ciphertexts. In *Provable Security-ProvSec'14*, vol. 8782, LNCS. Springer: Heidelberg, 2014; 259–273.
21. Blaze M, Bleumer G, Strauss M. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology-EUROCRYPT'98*, vol. 1403, LNCS. Springer: Heidelberg, 1998; 127–144.
22. Ateniese G, Fu K, Green M, Hohenberger S. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security* 2006; **9**(1): 1–30.
23. Canetti R, Hohenberger S. Chosen-ciphertext secure proxy re-encryption. *CCS'07*, ACM, New York, 2007; 185–194.
24. Matsuo T. Proxy re-encryption systems for identity-based encryption. In *Pairing'07*, vol. 4575, LNCS. Springer: Heidelberg, 2007; 247–267.
25. Libert B, Vergnaud D. Unidirectional chosen-ciphertext secure proxy re-encryption. In *Public Key Cryptography–PKC 2008*, Vol. 4939. Springer: Heidelberg, 2008; 360–379.
26. Kawai Y, Takashima K. *Fully-anonymous functional proxy-re-encryption*. Available from: https://eprint.iacr.org/2013/318.pdf [Accessed on 11 October 2013].
27. Liu Q, Wang G, Wu J. Clock-based proxy re-encryption scheme in unreliable clouds. *ICPPW'12*, IEEE, Pittsburgh, 2012; 304–305.
28. Liu Q, Wang G, Wu J. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information Sciences* 2014; **258**: 355–370.
29. Lynn B. *The Stanford Pairing Based Crypto Library*: Pittsburgh. Available from: http://crypto.stanford.edu/pbc [Accessed on 2 June 2011].