Dual-Policy Attribute Based Encryption

Nuttapong Attrapadung and Hideki Imai

Research Center for Information Security (RCIS),
National Institute of Advanced Industrial Science and Technology (AIST)
Akihabara-Daibiru Room 1003, 1-18-13, Sotokanda,
Chiyoda-ku, Tokyo 101-0021 Japan
{n.attrapadung,h-imai}@aist.go.jp

Abstract. We present a new variant of Attribute based encryption (ABE) called Dual-Policy ABE. Basically, it is a conjunctively combined scheme between Key-Policy and Ciphertext-Policy ABE, the two previous available types of ABE. Dual-Policy ABE allows *simultaneously* two access control mechanisms over encrypted data: one involves policies over *objective* attributes ascribed to data and the other involves policies over *subjective* attributes ascribed to user credentials. The previous two types of ABE can only allow either functionality above one at a time.

Keywords: Attribute-based encryption, Ciphertext policy, Key policy.

1 Introduction

Attribute-based encryption (ABE) enables an access control mechanism over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. ABE comes in two flavors called Ciphertext-Policy ABE and Key-Policy ABE.

In Ciphertext-Policy ABE, an encryptor can express any access policy, stating what kind of receivers will be able to decrypt the message, directly in the encryption algorithm (which can be run by anyone knowing the universal public key issued priorly by an authority). Such a policy is specified in terms of access structure over attributes. A user is ascribed by an attribute set, in the sense that each attribute corresponds to one of her credential, and is priorly given the private key from the authority. Such a user can decrypt a ciphertext if her attribute satisfies the access policy associated to the ciphertext. An example application of CP-ABE is secure mailing list system with access policy. There, a private key will be assigned for an attribute set, such as {"MANAGER", "AGE:30", "INSTITUTE:ABC"}, while policies over attributes such as "MANAGER" $\vee$ ("TRAINEE" $\wedge$ "AGE:25") will be associated to ciphertexts.

In Key-Policy ABE, the roles of an attribute set and an access policy are swapped from what we described for CP-ABE. Attribute sets are used to annotate the ciphertexts and access policies over these attributes are associated to users' secret keys. An example application of KP-ABE is Pay-TV system with package policy (called target broadcast system in [6]). There, a ciphertext

M. Abdalla et al. (Eds.): ACNS 2009, LNCS 5536, pp. 168–185, 2009.

will associate with an attribute set, such as {"TITLE:24", "GENRE:SUSPENSE", "SEASON:2", "EPISODE:13" }, while policies over attributes such as "SOCCER" ∨ ("TITLE:24" ∧ "SEASON:5") will be associated to TV program package keys that user receives when subscribes.

A drawback of the above two previous types of ABE is that we must choose whether attributes will be used to annotate either the ciphertexts, which we call *objects* (since they are to be decrypted), or the users' credentials, which we call *subjects* (since users are to decrypt); after setup we must also stick with such condition throughout the entire application. To see why this is inconvenient, we give an example in the Pay-TV application above. Since we are using KP-ABE, the encrypted movie can only be ascribed by *objective* attributes. Thus, the broadcast station, which is the encryptor, cannot directly specify *subjective* access policy, *i.e.,* who can or cannot decrypt. It might want to do so, since it may want, for example, to directly include or revoke some user credentials. The same inconvenience happens also for CP-ABE complimentarily.

In this paper, we present a new type of ABE called Dual-Policy ABE, which resolves the above problem affirmatively. Basically, it is a conjunctively combined scheme between KP and CP ABE. Dual-Policy ABE works as follows. An encryptor can associate the data simultaneously with both a set *objective* attributes that annotate the data itself and a *subjective* access policy that states what kind of receivers will be able to decrypt. On the other hand, a user is given a private key assigned simultaneously for both a set of *subjective* attributes that annotate user's credentials and a *subjective* access policy that states what kind of data she can decrypt. The decryption can be done if and only if the objective attribute set satisfies the objective policy *and* the subjective attribute set satisfies the subjective policy.

Previous Works. ABE was introduced by Sahai and Waters [9] in the context of a generalization of ID-based encryption (IBE) [2] called Fuzzy IBE, which is an ABE that allows only single threshold access structures. The first (and still being state-of-the-art) KP-ABE scheme that allows any monotone access structures was proposed by Goyal et al. [6], while the first such CP-ABE scheme, albeit with the security proof in the generic bilinear group model, was proposed by Bethencourt, Sahai, and Waters [1]. Ostrovsky, Sahai, and Waters [8] then subsequently extended both to handle also any non-monotone structures. Goyal et al. [5] presented bounded CP-ABE in the standard model. Waters [10] recently proposed the first fully expressive CP-ABE in the standard model.

Our Approach. Our DP-ABE scheme is based on an algebraic combination of CP-ABE by Waters [10] and KP-ABE by Goyal et al. [6]. We note that such a combination is non-trivial at the first place, since, for example, one may think of obtaining DP-ABE by using AND-double encryption (even in a secure way) of KP-ABE and CP-ABE. However, one can easily find out that this mislead method is insecure due to collusion attacks. Our scheme utilizes more sophisticated techniques for secure integration.

Our DP-ABE subsumes both KP-ABE and CP-ABE in the sense that when neglecting objective attributes our scheme becomes CP-ABE of Waters [10] and when neglecting subjective attributes it becomes KP-ABE of Goyal et al. [6].

Furthermore, our DP-ABE scheme also realizes the delegation of private keys. An interesting property is that we can also delegate the key of pure KP-ABE to a key of DP-ABE, where subjective attribute dimension is added, and the key of pure CP-ABE to a key of DP-ABE, where objective attribute dimension is added. Therefore our DP-ABE scheme is extended seamlessly from both KP-ABE of [6] and CP-ABE of [10].

Another feature of our DP-ABE scheme is that even such a scheme has been already set-up to be used as DP-ABE, it can also be used as if it were KP-ABE or CP-ABE on-the-fly by using encryption in what we call single-policy modes. This flexibility provides great convenience since the same instantiated key can be used for all three variants of ABE.

More Related Works. Recently, Boneh and Hamburg [3] formalized a very general framework called Generalized IBE (GIBE) which also includes both of ABE variants as special cases. DP-ABE also falls into their framework: it can be casted as a product scheme between KP-ABE and CP-ABE. However, their instantiated construction for KP-ABE seems to have large key size that is linear to the access structure collection size, which could be super-polynomially large.

Another similar general framework called predicate encryption was proposed previously by Katz, Sahai, and Waters [7]. Their system achieves also anonymity property, where the information about access structures or attribute sets associated with ciphertexts itself is kept hidden. However, their system tends to handle only less expressive access structures than systems without anonymity.

Organization of the Paper. We first provide preliminary materials such as the notion of linear secret sharing and bilinear pairing in Section 2. We then present the definition and the security notion of Dual-Policy ABE in Section 3. In Section 4, we present our concrete DP-ABE scheme called DPABE. In Section 5, we describe the key delegation of our DP-ABE scheme. In Section 6, we present both generic and specific enhanced schemes for DP-ABE that admit single-policy modes. We then conclude in Section 7. The security proofs of the schemes with key delegation and single-policy modes are given in Appendix A.1,A.2.

2 Preliminaries

We first provide the notion of access structure and linear secret sharing scheme as follows. Such formalization is recapped from [10].

Definition 1 (Access Structure). *Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\mathcal{P}}$ is monotone if for all B, C we have that if $B \in \mathbb{A}$ and $B \subseteq C$ then $C \in \mathbb{A}$. An access structure (respectively, monotonic access structure) is a collection (respectively, monotone collection) $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$.*

Definition 2 (Linear Secret Sharing Schemes (LSSS)). *Let $\mathcal{P}$ be a set of parties. Let M be a matrix of size $\ell \times k$. Let $\rho : \{1, \ldots, \ell\} \to \mathcal{P}$ be a function that maps a row to a party for labeling. A secret sharing scheme Π for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$ is a linear secret-sharing scheme in $\mathbb{Z}_p$ and is represented by (M, ρ) if it consists of two polynomial-time algorithms:*

$\mathsf{Share}_{(M,\rho)}$**:** *The algorithm takes as input $s \in \mathbb{Z}_p$ which is to be shared. It randomly chooses $y_2, \ldots, y_k \in \mathbb{Z}_p$ and let $\boldsymbol{v} = (s, y_2, \ldots, y_k)$. It outputs $M\boldsymbol{v}$ as the vector of ℓ shares. The share $\lambda_{\rho(i)} := \boldsymbol{M_i} \cdot \boldsymbol{v}$ belongs to party $\rho(i)$, where we denote $\boldsymbol{M_i}$ as the ith row in M.*

$\mathsf{Recon}_{(M,\rho)}$**:** *The algorithm takes as input $S \in \mathbb{A}$. Let $I = \{i |\ \rho(i) \in S\}$. It outputs reconstruction constants $\{(i, \mu_i)\}_{i \in I}$ which has a linear reconstruction property: $\sum_{i \in I} \mu_i \cdot \lambda_{\rho(i)} = s$.*

Proposition 1. *Let (M, ρ) be a LSSS for access structure $\mathbb{A}$ over a set of parties $\mathcal{P}$, where M is a matrix of size $\ell \times k$. For all $S \notin \mathbb{A}$, there exists a polynomial time algorithm that outputs a vector $\boldsymbol{w} = (w_1, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $w_1 = -1$ and for all $x \in S$ it holds that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$.*

Bilinear Maps. We briefly review facts about bilinear maps. Let $\mathbb{G}, \mathbb{G}_T$ be multiplicative groups of prime order p. Let g be a generator of $\mathbb{G}$. A bilinear map is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ for which the following hold: (1) e is bilinear; that is, for all $u, v \in \mathbb{G}$, $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$. (2) The map is non-degenerate: $e(g, g) \neq 1$. We say that $\mathbb{G}$ is a bilinear group if the group action in $\mathbb{G}$ can be computed efficiently and there exists $\mathbb{G}_T$ for which the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is efficiently computable.

Decision BDHE Assumption. Let $\mathbb{G}$ be a bilinear group of prime order p. The Decision q-BDHE (Bilinear Diffie-Hellman Exponent) problem [4] in $\mathbb{G}$ is stated as follows: given a vector

$$\left(g, h, g^{\alpha}, g^{(\alpha^2)}, \ldots, g^{(\alpha^q)}, g^{(\alpha^{q+2})}, \ldots, g^{(\alpha^{2q})}, Z\right)$$

$\in \mathbb{G}^{2q+1} \times \mathbb{G}_T$ as input, determine if $Z = e(g, h)^{(\alpha^{q+1})}$. We denote $g_i = g^{(\alpha^i)} \in \mathbb{G}$ for shorthand. Let $\boldsymbol{y}_{g,\alpha,q} = (g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q})$. An algorithm $\mathcal{A}$ that outputs $b \in \{0, 1\}$ has advantage ϵ in solving Decision q-BDHE in $\mathbb{G}$ if

$$|\Pr[\mathcal{A}(g, h, \boldsymbol{y}_{g,\alpha,q}, e(g_{q+1}, h)) = 0] - \Pr[\mathcal{A}(g, h, \boldsymbol{y}_{g,\alpha,q}, Z) = 0]| \geq \epsilon,$$

where the probability is over the random choice of generators $g, h \in \mathbb{G}$, the random choice of $\alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_T$, and the randomness of $\mathcal{A}$. We refer to the distribution on the left as $\mathcal{P}_{BDHE}$ and on the right as $\mathcal{R}_{BDHE}$. We say that the Decision q-BDHE assumption holds in $\mathbb{G}$ if no polynomial-time algorithm has a non-negligible advantage in solving the problem.

3 Definitions

A Dual-policy attribute-based encryption scheme consists of four algorithms.

Setup**:** This is a randomized algorithm that takes no input other than the implicit security parameter. It outputs public key pk and master key msk.

$\mathsf{Encrypt}(\mathsf{pk}, \mathcal{M}, (\mathbb{S}, \omega))$**:** This is a randomized algorithm that takes as input the public key pk, a message $\mathcal{M}$, a subjective access structure $\mathbb{S}$, a set of objective attributes ω. It outputs the ciphertext ct.

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, (\psi, \mathbb{O}))$**:** This is a randomized algorithm that takes as input the public key pk, the master key msk, a set of subjective attributes ψ, an objective access structure $\mathbb{O}$. It outputs a private decryption key sk.

$\mathsf{Decrypt}(\mathsf{pk}, (\psi, \mathbb{O}), \mathsf{sk}, (\mathbb{S}, \omega), \mathsf{ct})$**:** This algorithm takes as input the public key pk, a decryption key sk and its associated pair of set of subjective attributes ψ and objective access structure $\mathbb{O}$, a ciphertext ct and its associated pair of subjective access structure $\mathbb{S}$ and set of objective attributes ω. It outputs the message $\mathcal{M}$ if it holds that the set ω of objective attributes satisfies the objective access structure $\mathbb{O}$ and that the set ψ of subjective attributes satisfies the subjective access structure $\mathbb{S}$, *i.e.,* $\omega \in \mathbb{O}$ *and* $\psi \in \mathbb{S}$.

We require the standard correctness of decryption: if $\mathsf{Setup} \rightarrow (\mathsf{pk}, \mathsf{msk})$ then $\mathsf{Decrypt}\Big(\mathsf{pk}, (\psi, \mathbb{O}), \mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, (\psi, \mathbb{O})), (\mathbb{S}, \omega), \mathsf{Encrypt}(\mathsf{pk}, \mathcal{M}, (\mathbb{S}, \omega))\Big) \rightarrow \mathcal{M}$, for all $\mathcal{M}$ in the message space and all $\omega \in \mathbb{O}$ and $\psi \in \mathbb{S}$.

The selective security notion for DP-ABE is defined in the following game.

Init. The adversary declares the target subjective access structure $\mathbb{S}^\star$ and the target objective attribute set $\omega^\star$.

Setup. The challenger runs the Setup algorithm of DP-ABE and gives the public key pk to the adversary.

Phase 1. The adversary is allowed to issue queries for private keys for pairs of subjective attribute set and objective access structure $(\psi, \mathbb{O})$ such that $\omega^\star \notin \mathbb{O}$ *or* $\psi \notin \mathbb{S}^\star$, *i.e.,* the negated condition of that of a legitimate key which can be used to decrypt a challenge ciphertext.

Challenge. The adversary submits two equal length messages $\mathcal{M}_0$ and $\mathcal{M}_1$. The challenger flips a random bit b and computes the challenge ciphertext $\mathsf{ct}^\star$ on the target pair $(\mathbb{S}^\star, \omega^\star)$ of subjective access structure and objective attribute set and then gives $\mathsf{ct}^\star$ to the adversary.

Phase 2. Phase 1 is repeated.

Guess. The adversary outputs a guess b' of b.

The advantage of an adversary in this game is defined as $\Pr[b = b'] - \frac{1}{2}$. Note that this can be extended to handle chosen-ciphertext attacks by allowing decryption queries in Phase 1,2.

Definition 3. *A DP-ABE scheme is secure in the selective-set security notion if all polynomial time adversaries have at most a negligible advantage in the above game.*

4 Dual-Policy ABE Scheme

Our DP-ABE scheme will be based on a combination of CP-ABE by Waters [10] and KP-ABE by Goyal et al. [6]. Both subjective and objective access structures are those which there exist linear secret sharing schemes that realize them. We denote by (M, ρ) a LSSS scheme that represents a subjective access structure $\mathbb{S}$ and by (N, π) a LSSS scheme that represents a objective access structure $\mathbb{O}$. We will restrict ρ to be an injective function as in Waters [10] scheme, but we can extend to an unrestricted scheme, also similarly as in [10].

4.1 Main Construction

Let m be the maximum size of subjective attribute set allowed to be assigned to a key, *i.e.,* we restrict $|\psi| \leq m$. Let n be the maximum size of objective attribute set allowed to be associated with a ciphertext, *i.e.,* we restrict $|\omega| \leq n$. Let $\ell_{\mathsf{s,max}}$ be the maximum number of rows allowed in a subjective access structure matrix. Let $m' = m + \ell_{\mathsf{s,max}} - 1$ and $n' = n - 1$. Our main scheme DPABE is described as follows. Let $\mathcal{U}_{\mathsf{s}}, \mathcal{U}_{\mathsf{o}}$ be the universe of subjective and objective attributes, respectively.

▶ Setup: The algorithm first picks a random generator $g \in \mathbb{G}$ and random exponent $\gamma, a \in \mathbb{Z}_p$. It then defines two functions $F_{\mathsf{s}} : \mathbb{Z}_p \to \mathbb{G}$ and $F_{\mathsf{o}} : \mathbb{Z}_p \to \mathbb{G}$ by first randomly choosing $h_0, \ldots, h_{m'}, t_0, \ldots, t_{n'} \in \mathbb{G}$ and setting

$$F_{\mathsf{s}}(x) = \prod_{j=0}^{m'} h_j^{x^j}, \quad F_{\mathsf{o}}(x) = \prod_{j=0}^{n'} t_j^{x^j}.$$

It assigns the public key as $\mathsf{pk} = (g, e(g,g)^{\gamma}, g^a, h_0, \ldots, h_{m'}, t_0, \ldots, t_{n'})$. The master key is $\mathsf{msk} = (\gamma, a)$.

▶ Encrypt: Inputs to the encryption algorithm are a LSSS access structure (M, ρ) for subjective policy and a objective attribute set $\omega \subset \mathcal{U}_{\mathsf{o}}$. Let M be $\ell_{\mathsf{s}} \times k_{\mathsf{s}}$ matrix. The algorithm first randomly chooses $s, y_2, \ldots, y_{k_{\mathsf{s}}} \in \mathbb{Z}_p$ and lets $\boldsymbol{u} = (s, y_2, \ldots, y_{k_{\mathsf{s}}})$. For $i = 1$ to ℓ_{s}, it calculates $\lambda_i = \boldsymbol{M_i} \cdot \boldsymbol{u}$, where $\boldsymbol{M_i}$ is the vector corresponding to ith row of M. The ciphertext ct is set to $\mathsf{ct} = (C, \hat{C}, \{C_i\}_{i=1,\ldots,\ell_{\mathsf{s}}}, \{C'_x\}_{x \in \omega})$, where

$$C = \mathcal{M} \cdot (e(g,g)^{\gamma})^s, \qquad \hat{C} = g^s,$$
$$C_i = g^{a\lambda_i} F_{\mathsf{s}}(\rho(i))^{-s}, \qquad C'_x = F_{\mathsf{o}}(x)^s.$$

▶ KeyGen: Inputs to the encryption algorithm are a LSSS access structure (N, π) for objective policy and a subjective attribute set $\psi \subset \mathcal{U}_{\mathsf{s}}$. Let N be $\ell_{\mathsf{o}} \times k_{\mathsf{o}}$ matrix. The algorithm first randomly chooses $r, z_2 \ldots, z_{k_{\mathsf{o}}} \in \mathbb{Z}_p$ and lets $\boldsymbol{v} = (\gamma + ar, z_2, \ldots, z_{k_{\mathsf{o}}})$. For $i = 1$ to ℓ_{o}, it calculates $\sigma_i = \boldsymbol{N_i} \cdot \boldsymbol{v}$, where $\boldsymbol{N_i}$ is the vector corresponding to ith row of N. It also randomly chooses $r_1, \ldots, r_{\ell_{\mathsf{o}}} \in \mathbb{Z}_p$.

It creates the private decryption key as $\mathsf{sk} = (K, \{K_x\}_{x\in\psi}, \{\hat{K}_i, K'_i\}_{i=1,\ldots,\ell_\mathsf{o}})$, where

$$K = g^r, \qquad K_x = F_\mathsf{s}(x)^r,$$
$$\hat{K}_i = g^{\sigma_i} F_\mathsf{o}(\pi(i))^{-r_i}, \qquad K'_i = g^{r_i}.$$

▶ Decrypt: The decryption algorithm takes as input the ciphertext ct which contains a subjective access structure (M, ρ) and a set of objective attributes ω, and a decryption key sk which contains a set of subjective attributes ψ and an objective access structure (N, π). Suppose that the set ψ for subjective attribute satisfies (M, ρ) and that the set ω for objective attribute satisfies (N, π) (so that the decryption is possible). We then let $I_\mathsf{s} = \{i|\ \rho(i) \in \psi\}$ and $I_\mathsf{o} = \{i|\ \pi(i) \in \omega\}$. It then calculates corresponding sets of reconstruction constants $\{(i, \mu_i)\}_{i\in I_\mathsf{s}} = \mathsf{Recon}_{(M,\rho)}(\psi)$ and $\{(i, \nu_i)\}_{i\in I_\mathsf{o}} = \mathsf{Recon}_{(N,\pi)}(\omega)$. The decryption algorithm then computes

$$C \cdot \frac{\prod_{i\in I_\mathsf{s}} \Big(e(C_i, K) \cdot e(\hat{C}, K_{\rho(i)})\Big)^{\mu_i}}{\prod_{j\in I_\mathsf{o}} \Big(e(\hat{K}_j, \hat{C}) \cdot e(K'_j, C'_{\pi(j)})\Big)^{\nu_j}} = \mathcal{M}. \tag{1}$$

Correctness. We verify the correctness of the decryption as follows. Let sk and ct be defined as in the scheme above. We first note that from linear reconstruction property of the LSSS schemes, we have

$$\sum_{i\in I_\mathsf{s}} \mu_i \lambda_i = s, \quad \sum_{i\in I_\mathsf{o}} \nu_i \sigma_i = \gamma + ar. \tag{2}$$

The correctness can then be verified as

$$\begin{aligned}
&C \cdot \frac{\prod_{i\in I_\mathsf{s}} \Big(e(C_i, K) \cdot e(\hat{C}, K_{\rho(i)})\Big)^{\mu_i}}{\prod_{j\in I_\mathsf{o}} \Big(e(\hat{K}_j, \hat{C}) \cdot e(K'_j, C'_{\pi(j)})\Big)^{\nu_j}} \\
&= C \cdot \frac{\prod_{i\in I_\mathsf{s}} \Big(e(g^{a\lambda_i} F_\mathsf{s}(\rho(i))^{-s}, g^r) \cdot e(g^s, F_\mathsf{s}(\rho(i))^r)\Big)^{\mu_i}}{\prod_{j\in I_\mathsf{o}} \Big(e(g^{\sigma_j} F_\mathsf{o}(\pi(j))^{-r_j}, g^s) \cdot e(g^{r_j}, F_\mathsf{o}(\pi(j))^s)\Big)^{\nu_j}} \\
&= C \cdot \frac{\prod_{i\in I_\mathsf{s}} e(g^{a\lambda_i}, g^r)^{\mu_i}}{\prod_{j\in I_\mathsf{o}} e(g^{\sigma_j}, g^s)^{\nu_j}} = C \cdot \frac{e(g^{as}, g^r)}{e(g^{\gamma+ar}, g^s)} = C \cdot \frac{1}{e(g,g)^{\gamma s}} = \mathcal{M}.
\end{aligned}$$

Remark 1. The above decryption algorithm of Eq.(1) was written only for ease of visualizing. A more efficient computation with the less number of applications of pairing can be done as follows. Note that Eq.(3) requires only $|\omega|+2$ applications of pairing, while Eq.(1) requires $2(|\omega| + |\psi|)$ such applications.

$$C \cdot \frac{e\Big(\big(\prod_{i\in I_\mathsf{s}} C_i^{\mu_i}\big), K\Big)}{\prod_{j\in I_\mathsf{o}} e\Big(K'_j, C'_{\pi(j)}\Big)^{\nu_j}} \cdot e\left(\hat{C}, \frac{\Big(\prod_{i\in I_\mathsf{s}} K_{\rho(i)}^{\mu_i}\Big)}{\Big(\prod_{j\in I_\mathsf{o}} \hat{K}_j^{\nu_j}\Big)}\right) = \mathcal{M}. \tag{3}$$

4.2 Security Proof

Theorem 1. *If an adversary can break the* DPABE *scheme with advantage* ϵ *in the selective-set security model for DP-ABE with a challenge subjective access structure matrix of size* $\ell_{\mathsf{s}}^{\star} \times k_{\mathsf{s}}^{\star}$*, then a simulator with advantage* ϵ *in solving the Decision q-BDHE problem can be constructed, where* $m + k_{\mathsf{s}}^{\star} \leq q$.

The proof follows mostly from [6,10] with some non-trivial adaptation mostly in simulating the private keys.

Proof. Suppose there exists an adversary, $\mathcal{A}$, that has advantage ϵ in attacking the DPABE scheme. We build a simulator $\mathcal{B}$ that solves the Decision q-BDHE problem in $\mathbb{G}$. $\mathcal{B}$ is given as input a random q-BDHE challenge $(g, h, \boldsymbol{y}_{g,\alpha,q}, Z)$, where $\boldsymbol{y}_{g,\alpha,q} = (g_1, \ldots, g_q, g_{q+2}, \ldots, g_{2q})$ and Z is either $e(g_{q+1}, h)$ or a random element in $\mathbb{G}_1$ (recall that $g_j = g^{(\alpha^j)}$). $\mathcal{B}$ proceeds as follows.

Init. The selective-set game begins with $\mathcal{A}$ first outputting $((M^{\star}, \rho^{\star}), \omega^{\star})$, where $(M^{\star}, \rho^{\star})$ is a target subjective access structure in the form of LSSS matrix and $\omega^{\star}$ is a target objective attribute set. Let $M^{\star}$ be of size $\ell_{\mathsf{s}}^{\star} \times k_{\mathsf{s}}^{\star}$, where $m + k_{\mathsf{s}}^{\star} \leq q$. Wlog, we can assume that $\ell_{\mathsf{s}}^{\star} = \ell_{\mathsf{s},\mathsf{max}}$ and $|\omega^{\star}| = n$.

Setup. $\mathcal{B}$ chooses random $\gamma' \in \mathbb{Z}_p$ and implicitly sets $\gamma = \gamma' + \alpha^{q+1}$ by letting $e(g,g)^{\gamma} = e(\alpha, \alpha^q) e(g,g)^{\gamma'}$. It also lets $g^a = g^{\alpha}$.

The simulator then programs the function F_{s} by defining $F_{\mathsf{s}}(x) = g^{p(x)}$, where p is a polynomial in $\mathbb{Z}_p[x]$ of degree $m + \ell_{\mathsf{s}}^{\star} - 1$ which is implicitly defined as follows. It first chooses $k_{\mathsf{s}}^{\star} + m + 1$ polynomial $p_0, \ldots, p_{k_{\mathsf{s}}^{\star}+m}$ in $\mathbb{Z}_p[x]$ of degree $m + \ell_{\mathsf{s}}^{\star} - 1$ in such a way that for x such that there exists an i where $x = \rho^{\star}(i)$ (there are exactly $\ell_{\mathsf{s}}^{\star}$ values of such x, since $\rho^{\star}$ is injective) we set

$$p_j(x) = \begin{cases} M_{i,j}^{\star} & \text{for } j \in [1, k_{\mathsf{s}}^{\star}], \\ 0 & \text{for } j \in [k_{\mathsf{s}}^{\star} + 1, k_{\mathsf{s}}^{\star} + m], \end{cases}$$

and random for x elsewhere (by randomly picking values at some other m points for each polynomial) and p_0 is chosen completely randomly. Write coefficients in each polynomial as $p_j(x) = \sum_{i=0}^{m+\ell_{\mathsf{s}}^{\star}-1} p_{j,i} \cdot x^i$. It then conceptually defines

$$p(x) = \sum_{j=0}^{k_{\mathsf{s}}^{\star}+m} p_j(x) \cdot \alpha^j.$$

by setting $h_i = \prod_{j=0}^{k_{\mathsf{s}}^{\star}+m} g_j^{p_{j,i}}$ for $i \in [0, m + \ell_{\mathsf{s}}^{\star} - 1]$. From the definition of F_{s} in the scheme, one can verify that

$$F_{\mathsf{s}}(x) = \prod_{i=0}^{m+\ell_{\mathsf{s}}^{\star}-1} h_i^{x^i} = g^{p(x)}.$$

The simulator then programs the next function F_{o} as follows. It randomly picks a polynomial in $\mathbb{Z}_p[x]$ of degree $n-1$, $f'(x) = \sum_{j=0}^{n-1} f'_j x^j$. Next it defines

$f(x) = \prod_{k \in \omega^\star}(x-k) = \sum_{j=0}^{n-1} f_j x^j$. We note that f_j's terms can be computed completely from $\omega^\star$. From this we can ensure that $f(x) = 0$ if and only if $x \in \omega^\star$. It then lets $t_j = g_q^{f_j} g^{f'_j}$ for $j = [0, n-1]$. We thus have

$$F_{\mathsf{o}}(x) = \prod_{j=0}^{n-1} t_j^{(x^j)} = g_q^{f(x)} \cdot g^{f'(x)}.$$

It then gives the public key $\mathsf{pk} = (g, e(g,g)^\gamma, g_1, h_0, \ldots, h_{m'}, t_0, \ldots, t_{n'})$ to $\mathcal{A}$.

Phase 1. The adversary makes requests for private keys corresponding to objective access structure and subjective attribute set pair $((N, \pi), \psi)$ subjected to condition that ψ does not satisfy $M^\star$ *or* $\omega^\star$ does not satisfy N. We distinguish two cases due to the latter condition.

[**Case 1**: $\omega^\star$ does not satisfy N].

The simulator randomly chooses $r \in \mathbb{Z}_p$. It then lets $K = g^r$ and for all $x \in \psi$ lets $K_x = F_{\mathsf{s}}(x)^r$ as in the construction.

Due to the condition in this case and by Proposition 1, there must exist a vector $\boldsymbol{a} = (a_1, \ldots, a_{k_{\mathsf{o}}}) \in \mathbb{Z}_p^{k_{\mathsf{o}}}$ such that $a_1 = -1$ and that for all i where $\pi(i) \in \omega^\star$, it holds that $\boldsymbol{N_i} \cdot \boldsymbol{a} = 0$.

The simulator randomly chooses $z'_2, \ldots, z'_{k_{\mathsf{o}}} \in \mathbb{Z}_p$ and lets $\boldsymbol{v'} = (0, z'_2, \ldots, z'_{k_{\mathsf{o}}})$. It implicitly defines a vector $\boldsymbol{v} = -(\gamma' + \alpha^{q+1} + \alpha r)\boldsymbol{a} + \boldsymbol{v'}$, which will be used for creating shares of $\gamma + \alpha r$ as in the construction.

For i where $\pi(i) \in \omega^\star$, it randomly chooses $r_i \in \mathbb{Z}_p$ and computes $K'_i = g^{r_i}$ and

$$\hat{K}_i = g^{\boldsymbol{N_i} \cdot \boldsymbol{v'}} F_{\mathsf{o}}(\pi(i))^{-r_i} = g^{\boldsymbol{N_i} \cdot \boldsymbol{v}} F_{\mathsf{o}}(\pi(i))^{-r_i},$$

where the right equality is due to $\boldsymbol{N_i} \cdot \boldsymbol{a} = 0$.

For i where $\pi(i) \notin \omega^\star$, it randomly chooses $r'_i \in \mathbb{Z}_p$. Observe that

$$\boldsymbol{N_i} \cdot \boldsymbol{v} = (\boldsymbol{N_i} \cdot \boldsymbol{a})\alpha^{q+1} + (r\boldsymbol{N_i} \cdot \boldsymbol{a})\alpha + \boldsymbol{N_i} \cdot (\boldsymbol{v'} - \gamma' \boldsymbol{a})$$

contains the term α^{q+1}, thus we cannot compute $g^{\boldsymbol{N_i} \cdot \boldsymbol{v}}$ as usual. We will use the term $F_{\mathsf{o}}(\pi(i))^{-r_i}$ to cancel out the unknown value. To do this it implicitly defines $r_i = r'_i + \frac{\alpha(\boldsymbol{N_i} \cdot \boldsymbol{a})}{f(\pi(i))}$. This can be done by setting

$$\hat{K}_i = g_1^{\left(r\boldsymbol{N_i} \cdot \boldsymbol{a} - \frac{(\boldsymbol{N_i} \cdot \boldsymbol{a}) f'(\pi(i))}{f(\pi(i))}\right)} \cdot g^{\boldsymbol{N_i} \cdot (\boldsymbol{v'} - \gamma' \boldsymbol{a})} \cdot F_{\mathsf{o}}(\pi(i))^{-r'_i},$$

$$K'_i = g^{r'_i} g_1^{\frac{(\boldsymbol{N_i} \cdot \boldsymbol{a})}{f(\pi(i))}} = g^{r_i},$$

which can be computed since $\pi(i) \notin \omega^\star$ hence $f(\pi(i)) \neq 0$. The correctness of $\hat{K}_i$ can be verified as:

$$\begin{aligned}\hat{K}_i &= (g_{q+1})^{\boldsymbol{N_i a}} g_1^{r\boldsymbol{N_i} \cdot \boldsymbol{a}} g^{\boldsymbol{N_i} \cdot (\boldsymbol{v'} - \gamma' \boldsymbol{a})} \cdot \left((g_{q+1})^{-\boldsymbol{N_i} \cdot \boldsymbol{a}} g_1^{-\frac{(\boldsymbol{N_i} \cdot \boldsymbol{a}) f'(\pi(i))}{f(\pi(i))}}\right) \cdot F_{\mathsf{o}}(\pi(i))^{-r'_i} \\ &= g^{\boldsymbol{N_i} \cdot \boldsymbol{v}} \cdot F_{\mathsf{o}}(\pi(i))^{-\frac{\alpha(\boldsymbol{N_i} \cdot \boldsymbol{a})}{f(\pi(i))}} \cdot F_{\mathsf{o}}(\pi(i))^{-r'_i} = g^{\boldsymbol{N_i} \cdot \boldsymbol{v}} \cdot F_{\mathsf{o}}(\pi(i))^{-r_i}.\end{aligned}$$

[**Case 2**: $\omega^\star$ satisfies N].

In this case, we must have that ψ does not satisfy $M^\star$. Therefore, by Proposition 1 and our definition of p_j above, there must exist a vector $(w_1, \ldots, w_{k_{\mathsf{s}}^\star}) \in \mathbb{Z}_p^{k_{\mathsf{s}}^\star}$ such that $w_1 = -1$ and for all $x \in \psi$ such that there exist i where $x = \rho^\star(i)$, we have $(p_1(x), \ldots, p_{k_{\mathsf{s}}^\star}(x)) \cdot (w_1, \ldots, w_{k_{\mathsf{s}}^\star}) = 0$. Next it also computes one possible solution of variables $w_{k_{\mathsf{s}}^\star+1}, \ldots, w_{k_{\mathsf{s}}^\star+m}$ from the system of $|\psi|$ equations: for all $x \in \psi$,

$$(p_1(x), \ldots, p_{k_{\mathsf{s}}^\star+m}(x)) \cdot (w_1, \ldots, w_{k_{\mathsf{s}}^\star+m}) = 0,$$

which is possible since $|\psi| \leq m$. Now we define $\boldsymbol{b_x} = (p_1(x), \ldots, p_{k_{\mathsf{s}}^\star+m}(x))$ and $\boldsymbol{w} = (w_1, \ldots, w_{k_{\mathsf{s}}^\star+m})$. Thus, for all $x \in \psi$ we have $\boldsymbol{b_x} \cdot \boldsymbol{w} = 0$.

The simulator then randomly chooses $r' \in \mathbb{Z}_p$ and implicitly defining

$$r = r' + w_1 \cdot \alpha^q + w_2 \cdot \alpha^{q-1} + \cdots w_{k_{\mathsf{s}}^\star+m} \cdot \alpha^{q-(k_{\mathsf{s}}^\star+m)+1}, \tag{4}$$

by setting $K = g^{r'} \prod_{k=1}^{k_{\mathsf{s}}^\star+m} (g_{q+1-k})^{w_k} = g^r$. From our definition of r, we have

$$\gamma + \alpha r = \gamma' + \alpha r' + w_2 \alpha^q + \cdots + w_{k_{\mathsf{s}}^\star+m} \cdot \alpha^{q-(k_{\mathsf{s}}^\star+m)+2},$$

where we observe that the important term α^{q+1} in γ is canceled out. It randomly chooses $z_2 \ldots, z_{k_{\mathsf{o}}} \in \mathbb{Z}_p$ and implicitly lets $\boldsymbol{v} = (\gamma + \alpha r, z_2, \ldots, z_{k_{\mathsf{o}}})$ as in the construction. It also randomly chooses $r_1, \ldots, r_{\ell_{\mathsf{o}}} \in \mathbb{Z}_p$. It then computes for $i = 1$ to ℓ_{o}, $K_i' = g^{r_i}$ and

$$\hat{K}_i = \big(g^{\gamma'} g_1^{r'} \prod_{k=2}^{k_{\mathsf{s}}^\star+m} (g_{q-k+2})^{w_k}\big)^{N_{i,1}} \cdot \prod_{j=2}^{k_{\mathsf{o}}} g^{N_{i,j} z_j} \cdot F_{\mathsf{o}}(\pi(i))^{-r_i},$$

where one can verify that $\hat{K}_i = g^{\boldsymbol{N_i} \cdot \boldsymbol{v}} \cdot F_{\mathsf{o}}(\pi(i))^{-r_i}$. We can compute this since g_{q+1} is not contained. The simulator then creates K_x for all $x \in \psi$ as:

$$K_x = K^{p_0(x)} \cdot \prod_{j=1}^{k_{\mathsf{s}}^\star+m} \left(g_j^{r'} \prod_{\substack{k \in [1, k_{\mathsf{s}}^\star+m] \\ k \neq j}} (g_{q+1-k+j})^{w_k} \right)^{p_j(x)},$$

where one can verify that $K_x = F_{\mathsf{s}}(x)^r$ by observing that since for all $x \in \psi$, we have $\boldsymbol{b_x} \cdot \boldsymbol{w} = 0$; therefore,

$$\begin{aligned} K_x = K_x \cdot (g_{q+1})^{\boldsymbol{b_x} \cdot \boldsymbol{w}} &= K_x \cdot \prod_{j=1}^{k_{\mathsf{s}}^\star+m} (g_{q+1-j+j})^{w_j p_j(x)} \\ &= K^{p_0(x)} \cdot \prod_{j=1}^{k_{\mathsf{s}}^\star+m} \left(g_j^{r'} \prod_{k=1}^{k_{\mathsf{s}}^\star+m} (g_{q+1-k+j})^{w_k} \right)^{p_j(x)} \\ &= (g^r)^{p_0(x)} \cdot \prod_{j=1}^{k_{\mathsf{s}}^\star+m} (g^r)^{\alpha^j p_j(x)} = (g^r)^{p(x)} = F_{\mathsf{s}}(x)^r. \end{aligned}$$

Challenge. The adversary gives two message $\mathcal{M}_0, \mathcal{M}_1$ to the simulator. The simulator flips a coin b and creates $C = \mathcal{M}_b \cdot Z \cdot e(h, g^{\gamma'})$, $\hat{C} = h$, and for $x \in \omega^\star$, $C'_x = h^{f'(x)}$. Write $h = g^s$ for some unknown s. The simulator chooses randomly $y'_2, \ldots, y_{k_{\mathsf{s}}^\star} \in \mathbb{Z}_p$. Let $\boldsymbol{y}' = (0, y'_2, \ldots, y_{k_{\mathsf{s}}^\star})$. It will then implicitly share the secret s using the vector

$$\boldsymbol{v} = (s, s\alpha + y'_2, s\alpha^2 + y'_3, \ldots, s\alpha^{k_{\mathsf{s}}^\star - 1} + y'_{k_{\mathsf{s}}^\star}),$$

by setting for $i = 1, \ldots, \ell_{\mathsf{s}}^\star$, $C_i = (g_1)^{\boldsymbol{M}_i^\star \cdot \boldsymbol{y}'} \cdot (g^s)^{-p_0(\rho^\star(i))}$.

We claim that if when $Z = e(g_{q+1}, h)$, then the above ciphertext is a valid challenge. The term $C, \hat{C}$ is trivial. For all $x \in \omega'$, we have $f(x) = 0$, hence

$$C'_x = (g^s)^{f'(x)} = (g_q^{f(x)} g^{f'(x)})^s = F_{\mathsf{o}}(x)^s.$$

For $i = 1, \ldots, \ell_{\mathsf{s}}^\star$, we have

$$\begin{aligned} C_i &= (g^\alpha)^{\boldsymbol{M}_i^\star \cdot \boldsymbol{y}'} \prod_{j=1}^{k_{\mathsf{s}}^\star} g^{M_{i,j}^\star s\alpha^j} \cdot (g^s)^{-p_0(\rho^\star(i))} \prod_{j=1}^{k_{\mathsf{s}}^\star} (g^s)^{-M_{i,j}^\star \alpha^j} \\ &= g^{\alpha \boldsymbol{M}_i^\star \cdot \boldsymbol{v}} \cdot (g^s)^{-p(\rho^\star(i))} = g^{\alpha \boldsymbol{M}_i^\star \cdot \boldsymbol{v}} F_{\mathsf{s}}(\rho^\star(i))^{-s}, \end{aligned}$$

which concludes our claim.

Phase 2. $\mathcal{B}$ performs exactly as it did in Phase 1.

Guess. $\mathcal{A}$ outputs $b' \in \{0, 1\}$ for the guess of b. If $b = b'$ then $\mathcal{B}$ outputs 1 (meaning $Z = e(g_{q+1}, h)$). Else, it outputs 0 (meaning Z is random in $\mathbb{G}_T$).

We see that if $(g, h, \boldsymbol{y}_{g,\alpha,q}, Z)$ is sampled from $\mathcal{R}_{BDHE}$ then $\Pr[\mathcal{B}(g, h, \boldsymbol{y}_{g,\alpha,q}, Z) = 0] = \frac{1}{2}$. On the other hand, if $(g, h, \boldsymbol{y}_{g,\alpha,q}, Z)$ is sampled from $\mathcal{P}_{BDHE}$ then we have $|\Pr[\mathcal{B}(g, h, \boldsymbol{y}_{g,\alpha,q}, Z) = 0] - \frac{1}{2}| \geq \epsilon$. It follows that $\mathcal{B}$ has advantage at least ϵ in solving q-BDHE problem in $\mathbb{G}$. This concludes the proof.

4.3 Some Extended Constructions

We note that an unrestricted scheme where ρ is not necessarily injective, a scheme with CCA security, a scheme based only on Decision Bilinear Diffie-Hellman (DBDH) assumption can be realized similarly to [10]. We can also model $F_{\mathsf{s}}, F_{\mathsf{o}}$ as random oracles and achieve better efficiency and simpler proof as in [10]. In Goyal et al. [6] paper, the KP-ABE for LSSS realizable structures does not have delegation property; while the one for access-tree structures have. We can also base our DP-ABE scheme on the access-tree based KP-ABE. Finally, we can extend the access structures to include non-monotone type ones as in [8].

5 Key Delegation in DP-ABE

We now extend the definition and scheme realizations of DP-ABE to obtain the delegation of keys. We begin with the definition of Delegate algorithm to be added on.

Delegate: It takes as inputs a private key $\mathsf{sk}_{(\psi,\mathbb{O})}$ of subjective attribute set and objective access structure pair $(\psi, \mathbb{O})$, and another new pair $(\psi', \mathbb{O}')$ intended to derive its key. It outputs the key $\mathsf{sk}_{(\psi',\mathbb{O}')}$ if and only if $\psi' \subseteq \psi$ and $\mathbb{O}' \subseteq \mathbb{O}$.

In other words, key delegation can be realized when the new subjective attribute set is a subset of the original set and the new objective access structure is more restrictive than the original one (or either one condition holds while the other remains the same). In defining this algorithm, we require its correctness that the private key $\mathsf{sk}_{(\psi',\mathbb{O}')}$ output from Delegate has the same distribution as the one from KeyGen algorithm.

Recall that $\mathcal{U}_\mathsf{s}$ is the universe of subjective attributes and $2^{\mathcal{U}_\mathsf{o}}$ is the full objective access structure. The delegation will start from the master key, which can be considered equivalently as the private key for $(\mathcal{U}_\mathsf{s}, 2^{\mathcal{U}_\mathsf{o}})$. From that, we can consider two types of intermediate states: $(\psi, 2^{\mathcal{U}_\mathsf{o}})$ which can be considered as a key in pure CP-ABE scheme and $(\mathcal{U}_\mathsf{s}, \mathbb{O})$ which can be considered as a key in pure KP-ABE scheme.

Such intermediate keys are indeed already defined generically in any DP-ABE scheme (by instantiating $\mathsf{sk}_{(\psi,\mathbb{O})}$ with $\mathbb{O} = 2^{\mathcal{U}_\mathsf{o}}$ for the first type and $\psi = \mathcal{U}_\mathsf{s}$ for the second type). However, both $2^{\mathcal{U}_\mathsf{o}}$ and $\mathcal{U}_\mathsf{s}$ are of super-polynomial size; therefore, the size of instantiated keys could be very large for any DP-ABE constructions (including our basic DPABE construction). To resolve this, we thus newly define KeyGen for only those two specific types of keys below.

We now describe the delegation scheme for our DPABE scheme as follows. The security proof is postponed to Section A.1.

5.1 Delegating CP-ABE to DP-ABE

$$(\mathcal{U}_\mathsf{s}, 2^{\mathcal{U}_\mathsf{o}}) \to (\psi, 2^{\mathcal{U}_\mathsf{o}}) \to (\psi, \mathbb{O})$$

From the master key $\mathsf{msk} = (\gamma, a)$, it randomly chooses $r \in \mathbb{Z}_p$ and creates a private key for $(\psi, 2^{\mathcal{U}_\mathsf{o}})$ as $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})} = (K, \{K_x\}_{x\in\psi}, \hat{K})$ where

$$K = g^r, \quad K_x = F_\mathsf{s}(x)^r, \quad \hat{K} = g^{\gamma + ar}. \tag{5}$$

Note that this is exactly the key in the CP-ABE of Waters [10]. This means that one can seamlessly extend Waters' CP-ABE to ours DP-ABE without having to setup again. The decryption using this key can be done by Eq.(1) but neglecting all the terms related to objective attribute set, ω. Thus, Eq.(1) is simplified to

$$C \cdot \frac{\prod_{i\in I_\mathsf{s}} \Big(e(C_i, K) \cdot e(\hat{C}, K_{\rho(i)})\Big)^{\mu_i}}{e(\hat{K}, \hat{C})} = \mathcal{M}.$$

From the above private key for $(\psi, 2^{\mathcal{U}_\mathsf{o}})$, we can further delegate to obtain a private key for $(\psi, \mathbb{O})$. Let $\mathbb{O}$ be represented by a LSSS (N, π) as usual. Let N be $\ell_\mathsf{o} \times k_\mathsf{o}$ matrix. The algorithm randomly chooses $r', z_2 \ldots, z_{k_\mathsf{o}}, r_1, \ldots, r_{\ell_\mathsf{o}} \in \mathbb{Z}_p$.

It implicitly lets $\boldsymbol{v} = (\gamma + a(r+r'), z_2, \ldots, z_{k_\mathsf{o}})$. It creates the private key $\mathsf{sk}_{(\psi,\mathbb{O})} = (K^{\mathsf{new}}, \{K_x^{\mathsf{new}}\}_{x\in\psi}, \{\hat{K}_i^{\mathsf{new}}, K_i'^{\mathsf{new}}\}_{i=1,\ldots,\ell_\mathsf{o}})$ as

$$K^{\mathsf{new}} = K \cdot g^{r'}, \qquad K_x^{\mathsf{new}} = K_x \cdot F_\mathsf{s}(x)^{r'},$$
$$\hat{K}_i^{\mathsf{new}} = (\hat{K} \cdot (g^a)^{r'})^{N_{i,1}} g^{\sum_{j=2}^{k_\mathsf{o}} N_{i,j} z_j} F_\mathsf{o}(\pi(i))^{-r_i}, \qquad K_i'^{\mathsf{new}} = g^{r_i},$$

which distributes exactly the same as in our main scheme; in particular, one can verify that $\hat{K}_i^{\mathsf{new}} = g^{\boldsymbol{N_i}\cdot\boldsymbol{v}} F_\mathsf{o}(\pi(i))^{-r_i}$.

5.2 Delegating KP-ABE to DP-ABE

$\boxed{(\mathcal{U}_\mathsf{s}, 2^{\mathcal{U}_\mathsf{o}}) \to (\mathcal{U}_\mathsf{s}, \mathbb{O}) \to (\psi, \mathbb{O})}$

From the master key $\mathsf{msk} = (\gamma, a)$, the algorithm will create a private key for $(\mathcal{U}_\mathsf{s}, \mathbb{O})$ as follows. Let $\mathbb{O}$ be represented by a LSSS (N, π) as usual. Let N be $\ell_\mathsf{o} \times k_\mathsf{o}$ matrix. The algorithm randomly chooses $z_2 \ldots, z_{k_\mathsf{o}}, r_1, \ldots, r_{\ell_\mathsf{o}} \in \mathbb{Z}_p$. It lets $\boldsymbol{z} = (\gamma, z_2, \ldots, z_{k_\mathsf{o}})$. It then creates $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})} = (\{\hat{K}_i, K_i'\}_{i=1,\ldots,\ell_\mathsf{o}})$ where

$$\hat{K}_i = g^{\boldsymbol{N_i}\cdot\boldsymbol{z}} F_\mathsf{o}(\pi(i))^{-r_i}, \quad K_i' = g^{r_i}, \tag{6}$$

Note that this is exactly the key in the KP-ABE of Goyal et al. [10]. This means that one can seamlessly extend such KP-ABE schemes to ours DP-ABE without having to setup again. The decryption using this key can be done by Eq.(1) but neglecting all the terms related to subjective attribute set, ψ. Thus, Eq.(1) is simplified to

$$C \cdot \frac{1}{\prod_{j\in I_\mathsf{o}} \left(e(\hat{K}_j, \hat{C}) \cdot e(K_j', C'_{\pi(j)})\right)^{\nu_j}} = \mathcal{M}.$$

From the above private key for $(\mathcal{U}_\mathsf{s}, \mathbb{O})$, we can further delegate to obtain a private key for $(\psi, \mathbb{O})$.The algorithm randomly chooses $r, z_2', \ldots, z_{k_\mathsf{o}}', r_1', \ldots, r_{\ell_\mathsf{o}}' \in \mathbb{Z}_p$. It creates $\mathsf{sk}_{(\psi,\mathbb{O})} = (K^{\mathsf{new}}, \{K_x^{\mathsf{new}}\}_{x\in\psi}, \{\hat{K}_i^{\mathsf{new}}, K_i'^{\mathsf{new}}\}_{i=1,\ldots,\ell_\mathsf{o}})$ as

$$K^{\mathsf{new}} = g^r, \qquad K_x^{\mathsf{new}} = F_\mathsf{s}(x)^r,$$
$$\hat{K}_i^{\mathsf{new}} = \hat{K}_i \cdot (g^a)^{N_{i,1} r} g^{\sum_{j=2}^{k_\mathsf{o}} N_{i,j} z_j'} F_\mathsf{o}(\pi(i))^{-r_i'}, \qquad K_i'^{\mathsf{new}} = K_i' \cdot g^{r_i'},$$

which distributes exactly the same as in our main scheme.

5.3 Delegating in DP-ABE

$\boxed{(\psi, \mathbb{O}) \to (\psi', \mathbb{O}')}$

The delegation from $(\psi, \mathbb{O}) \to (\psi', \mathbb{O})$, where $\psi' \subset \psi$, can be done by deleting the elements K_x where $x \in \psi \setminus \psi'$ and then re-randomizing the other remaining elements in a similar way as delegations stated previously. More

precisely, from $\mathsf{sk}_{(\psi,\mathbb{O})} = (K, \{K_x\}_{x\in\psi}, \{\hat{K}_i, K'_i\}_{i=1,\ldots,\ell_\mathsf{o}})$, the algorithm creates $\mathsf{sk}_{(\psi',\mathbb{O})} = (K^{\mathsf{new}}, \{K_x^{\mathsf{new}}\}_{x\in\psi'}, \{\hat{K}_i^{\mathsf{new}}, K_i'^{\mathsf{new}}\}_{i=1,\ldots,\ell_\mathsf{o}})$ as follows. It first randomly chooses $r', z'_2, \ldots, z'_{k_\mathsf{o}}, r'_1, \ldots, r'_{\ell_\mathsf{o}} \in \mathbb{Z}_p$ and then computes

$$K^{\mathsf{new}} = K \cdot g^{r'}, \qquad K_x^{\mathsf{new}} = K_x \cdot F_\mathsf{s}(x)^{r'},$$
$$\hat{K}_i^{\mathsf{new}} = \hat{K}_i \cdot (g^a)^{N_{i,1}r'} g^{\sum_{j=2}^{k_\mathsf{o}} N_{i,j} z'_j} F_\mathsf{o}(\pi(i))^{-r'_i}, \qquad K_i'^{\mathsf{new}} = K'_i \cdot g^{r'_i},$$

which distributes exactly the same as a key for $(\psi', \mathbb{O})$.

The delegation from $(\psi, \mathbb{O}) \to (\psi, \mathbb{O}')$, where $\mathbb{O}'$ is more restrictive than $\mathbb{O}$, can be done on the access-tree based DP-ABE in a similar way to the KP-ABE scheme of Goyal et al. [6], with proper re-randomization.

6 Single-Policy Modes of DP-ABE

In this section, we describe a feature of DP-ABE called encryption in single-policy modes. Suppose that a DP-ABE scheme has been set-up already. The single-policy encryption mode allows an encryptor to still encrypt his message as if it were a KP-ABE or CP-ABE on-the-fly. More specifically, when a message is encrypted in KP-ABE mode with objective attribute set ω, any user with key for $(\psi, \mathbb{O})$ where $\omega \in \mathbb{O}$ can decrypt it regardless of whatever subjective attribute set ψ. Analogously, when a message is encrypted in CP-ABE mode with subjective policy $\mathbb{S}$, any user with key for $(\psi, \mathbb{O})$ where $\psi \in \mathbb{S}$ can decrypt it regardless of whatever objective policy $\mathbb{O}$.

We now describe a simple generic construction and then a more efficient direct construction as follows.

6.1 Generic Construction

As a first attempt, we describe a trivial approach to generically realize encryption in single-policy modes as follows. To encrypt in KP-ABE mode with objective attribute set ω, one just encrypt to $(2^{\mathcal{U}_\mathsf{s}}, \omega)$. To encrypt in CP-ABE mode with subjective policy $\mathbb{S}$, one just encrypt to $(\mathbb{S}, \mathcal{U}_\mathsf{o})$. However, $2^{\mathcal{U}_\mathsf{s}}$ and $\mathcal{U}_\mathsf{o}$ are of super-polynomial size; therefore, the size of instantiated ciphertext could be very large for any DP-ABE constructions (including our basic DPABE construction).

To resolve this, we propose a simple generic conversion from any DP-ABE scheme S to a new DP-ABE scheme S' that admits efficient single-policy modes as follows. The idea is to use dummy attributes: one for subjective and one for objective attribute.

$\mathsf{S}'.\mathsf{Setup}$ is exactly the same as $\mathsf{S}.\mathsf{Setup}$ except that it additionally chooses a special subjective attribute $T_\mathsf{s} \in \mathcal{U}_\mathsf{s}$ and a special objective attribute $T_\mathsf{o} \in \mathcal{U}_\mathsf{o}$ and adds them into the public key. Both $T_\mathsf{s}, T_\mathsf{o}$ will not be used as attributes in S'. Next we define

$$\mathsf{S}'.\mathsf{KeyGen}(\mathsf{pk}, \mathsf{msk}, (\psi, \mathbb{O})) = \mathsf{S}.\mathsf{KeyGen}\big(\mathsf{pk}, \mathsf{msk}, (\psi \cup \{T_\mathsf{s}\}, \mathbb{O} \cup \{\{T_\mathsf{o}\}\})\big).$$

$\mathsf{S}'.\mathsf{Encrypt}$ is done as usual except in the single-policy modes where we define

$$\mathsf{S}'.\mathsf{Encrypt}(\mathsf{pk}, \mathcal{M}, (2^{\mathcal{U}_\mathsf{s}}, \omega)) = \mathsf{S}.\mathsf{Encrypt}\big(\mathsf{pk}, \mathcal{M}, (\{\{T_\mathsf{s}\}\}, \omega)\big),$$
$$\mathsf{S}'.\mathsf{Encrypt}(\mathsf{pk}, \mathcal{M}, (\mathbb{S}, \mathcal{U}_\mathsf{o})) = \mathsf{S}.\mathsf{Encrypt}\big(\mathsf{pk}, \mathcal{M}, (\mathbb{S}, \{T_\mathsf{o}\})\big),$$

which corresponds to KP-ABE and CP-ABE mode respectively. Decryption can be done exactly in the same way as usual.

6.2 Direct Construction

When applying the above generic conversion to our proposed DPABE, the resulting scheme seems to contain some redundancy, in particular, involving using the dummy subjective attribute and the LSSS scheme for the augmented objective access structure $\mathbb{O} \cup \{\{T_\mathsf{o}\}\}$. In this section, we thus also present a direct construction DPABE2 by tweaking the main DPABE construction as follows.

DPABE2.Setup is exactly the same as that of DPABE except that it also includes a special objective attribute $T_\mathsf{o} \in \mathcal{U}_\mathsf{o}$ in the public key. DPABE2.KeyGen is also exactly the same as before except the following. To generate the key $\mathsf{sk}_{(\psi,\mathbb{O})}$, it also includes two new elements $(\hat{K}_{(\mathsf{o})}, K'_{(\mathsf{o})})$ which are computed by first randomly choosing $\tilde{r} \in \mathbb{Z}_p$ and setting

$$\hat{K}_{(\mathsf{o})} = g^{\gamma + ar} F_\mathsf{o}(T_\mathsf{o})^{-\tilde{r}}, \quad K'_{(\mathsf{o})} = g^{\tilde{r}}. \tag{7}$$

Hence the key will be $\mathsf{sk}_{(\psi,\mathbb{O})} = (K, \{K_x\}_{x\in\psi}, \{\hat{K}_i, K'_i\}_{i=1,\ldots,\ell_\mathsf{o}}, \hat{K}_{(\mathsf{o})}, K'_{(\mathsf{o})})$.

For the intermediate states, the key $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}$ is unchanged from Eq.(5), while the key $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ is exactly the same as defined in Eq.(6) except that it additionally includes the two above new elements of Eq.(7) albeit setting $r = 0$. The delegation can be done as usual with proper re-randomization.

The encryption DPABE2.Encrypt is exactly the same as usual DPABE except in the single-policy modes which we describe below. To encrypt in KP-ABE mode, *i.e.,* to encrypt to $(2^{\mathcal{U}_\mathsf{s}}, \omega)$, one randomly chooses $s \in \mathbb{Z}_p$ and set the ciphertext to $\mathsf{ct} = (C, \hat{C}, C_0, \{C'_x\}_{x\in\omega})$, where

$$C = \mathcal{M} \cdot (e(g,g)^\gamma)^s, \qquad \hat{C} = g^s,$$
$$C_0 = g^{as}, \qquad C'_x = F_\mathsf{o}(x)^s.$$

The decryption in this case is done by simplifying Eq.(1) to

$$C \cdot \frac{e(C_0, K)}{\prod_{j\in I_\mathsf{o}} \Big(e(\hat{K}_j, \hat{C}) \cdot e(K'_j, C'_{\pi(j)})\Big)^{\nu_j}} = \mathcal{M}.$$

On the other hand, to encrypt in CP-ABE mode, *i.e.,* to encrypt to $(\mathbb{S}, \mathcal{U}_\mathsf{o})$, one just compute as in the usual DPABE.Encrypt but set the ciphertext to $\mathsf{ct} = (C, \hat{C}, \{C_i\}_{i=1,\ldots,\ell_\mathsf{s}}, C')$, where

$$C = \mathcal{M} \cdot (e(g,g)^\gamma)^s, \qquad \hat{C} = g^s,$$
$$C_i = g^{a\lambda_i} F_\mathsf{s}(\rho(i))^{-s}, \qquad C' = F_\mathsf{o}(T_\mathsf{o})^s.$$

The decryption in this case is done by simplifying Eq.(1) to

$$C \cdot \frac{\prod_{i \in I_s} \left(e(C_i, K) \cdot e(\hat{C}, K_{\rho(i)}) \right)^{\mu_i}}{e(\hat{K}_{(\mathsf{o})}, \hat{C}) \cdot e(K'_{(\mathsf{o})}, C')} = \mathcal{M}.$$

The security proof of DPABE2 is given in Section A.2.

7 Conclusions

We presented a new variant of Attribute based encryption (ABE) called Dual-Policy ABE. It is a useful primitive that combines two access control functionalities from Ciphertext-policy ABE and Key-policy ABE. We formalized the notion of Dual-policy ABE and presented an efficient concrete scheme based on an algebraic combination between Goyal et al. KP-ABE [6] and Waters' CP-ABE [10], which are the state-of-the-art schemes for ABE of respective kinds. We further proposed two add-on features: key delegation and single-policy modes of encryption. Key delegation has an interesting property that it also allows the delegation from KP-ABE key of Goyal et al. scheme or CP-ABE key of Waters' scheme to our DP-ABE. Therefore, one can extend those two existing ABE schemes by delegating to DP-ABE seamlessly. Single-policy mode feature allows users to use DP-ABE keys as if it were the vanilla KP-ABE or CP-ABE on-the-fly. This feature allows great flexibility since one DP-ABE key can be used for all three types of ABE (KP,CP,DP ABE).

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy 2007, pp. 321–334 (2007)
2. Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. SIAM Journal of Computing 32(3), 586–615 (2003); In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
3. Boneh, D., Hamburg, M.: Generalized Identity Based and Broadcast Encryption Schemes. In: Pieprzyk, J. (ed.) Asiacrypt 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
4. Boneh, D., Gentry, C., Waters, B.: Collusion resistant broadcast encryption with short ciphertexts and private keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
5. Goyal, V., Jain, A., Pandey, O., Sahai, A.: Bounded ciphertext policy attribute-based encryption. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008 (Track C), Part I. LNCS, vol. 5125, pp. 579–591. Springer, Heidelberg (2008)
6. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communications Security 2006, pp. 89–98 (2006)
7. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)

8. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-Based Encryption with Non-Monotonic Access Structures. In: ACM Conference on Computer and Communications Security 2007, pp. 195–203 (2007)
9. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
10. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. Cryptology ePrint archive: report 2008/290

A Security Proofs of Schemes with Extended Features

A.1 Security Proof of the Scheme with Delegation

In this section, we describe the security proof of the scheme with delegation given in Section 5. The only difference from our basic construction in Section 4.1 is that we newly re-define the private key $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}, \mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$, for the intermediate states. According to the security definition, the adversary can also query for the key $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ if $\omega^\star$ does not satisfy $\mathbb{O}$ and the key $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}$ if ψ does not satisfy $\mathbb{S}^\star$. Here we recall that $(\mathbb{S}^\star, \omega^\star)$ is the target subjective access structure and objective attribute set pair. Therefore, it suffices to show how to simulate these two types of keys in Phase 1 (and 2), in addition to the proof of the main scheme (*cf.* Section 4.2).

For the first type, the simulator $\mathcal{B}$ answers the query for $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ such that $\omega^\star$ does not satisfy $\mathbb{O}$ by simulating the private key elements in exactly the same way as in the Case 1 in Phase 1 in the proof of the main scheme, albeit setting $r = 0$ and neglecting the term K, K_x. The resulting simulated key $(\{\hat{K}_i, K'_i\}_{i=1,\ldots,\ell_\mathsf{o}})$ is distributed as the key $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ in the real scheme (*cf.* Eq.(6)). This holds thanks to the correctness of simulation for $\mathsf{sk}_{(\psi,\mathbb{O})}$ in the proof of our main scheme and the fact that $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ as defined in Eq.(6) simplifies $\mathsf{sk}_{(\psi,\mathbb{O})}$ as defined in the main scheme with r being set to $r = 0$.

For the second type, the simulator $\mathcal{B}$ answers the query for $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}$ such that ψ does not satisfy $\mathbb{S}^\star$ as follows. Since the elements $(K, \{K_x\}_{x\in\psi})$ in both the key $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}$ defined in Eq.(5) and the key $\mathsf{sk}_{(\psi,\mathbb{O})}$ of the main scheme are the same, we just simulate $(K, \{K_x\}_{x\in\psi})$ exactly as in the Case 2 in Phase 1 in the proof of the main scheme. It then computes $\hat{K}$ as $\hat{K} = g^{\gamma'} g_1^{r'} \prod_{k=2}^{k^\star_\mathsf{s}+m} (g_{q-k+2})^{w_k}$,which can be verified that $\hat{K} = g^{\gamma+\alpha r}$ as required (recall that in the simulation, r is implicitly defined in Eq.(4) and $a = \alpha$).

Remark 2. In the security proof of the main scheme in Section 4.2, we could have done a simpler simulation if the key delegation were already defined there. For Case 1, it suffices to compute the key $\mathsf{sk}_{(\mathcal{U}_\mathsf{s},\mathbb{O})}$ and then delegate to $\mathsf{sk}_{(\psi,\mathbb{O})}$ to answer the query. For Case 2, it suffices to compute the key $\mathsf{sk}_{(\psi,2^{\mathcal{U}_\mathsf{o}})}$ and then delegate to $\mathsf{sk}_{(\psi,\mathbb{O})}$. However, we believe that separating the key delegation feature from the basic scheme makes its description easier to follow.

A.2 Security Proof of the Scheme with Single-Policy Modes

In this section, we give a sketch of the security proof for this tweaked scheme DPABE2 given in Section 6.2. Note that the only differences from the main proof

are as follows. First we must also consider two new possible target pair types of $(2^{\mathcal{U}_\mathsf{s}}, \omega)$ and $(\mathbb{S}, \mathcal{U}_\mathsf{o})$ for the challenge ciphertext. Second, we must also simulate the two new private key elements for each query.

We first consider the normal case where the adversary announces the target pair of type $(\mathbb{S}^\star, \omega^\star)$ in the Init phase. In this case, the proof follows exactly the main proof except that the simulator also simulates additional key components. For Case 1 of Phase 1 in the main proof, it computes the additional keys as

$$\hat{K}_{(\mathsf{o})} = g^{\gamma'} g^{\alpha r} g_1^{-f'(T_\mathsf{o})/f(T_\mathsf{o})} F_\mathsf{o}(T_\mathsf{o})^{-\tilde{r}'}, \quad K'_{(\mathsf{o})} = g^{\tilde{r}'} g_1^{1/f(T_\mathsf{o})}, \tag{8}$$

where it randomly chooses $\tilde{r}' \in \mathbb{Z}_p$. It can be verified that this distributes as in Eq.(7) with $\tilde{r} = \tilde{r}' + 1/f(T_\mathsf{o})$. For Case 2, the simulator can compute $g^{\gamma+\alpha r}$ and thus can generate the elements of Eq.(7) above.

Next, we consider the case where the adversary announces the target pair of type $(2^{\mathcal{U}_\mathsf{s}}, \omega^\star)$ in the Init phase, *i.e.,* the challenge ciphertext will be in KP-ABE mode. In Setup phase, the simulator chooses $a \in \mathbb{Z}_p$ and $h_0, \ldots, h_{m'} \in \mathbb{G}$ randomly (in particular, instead of setting $a = \alpha$ as previously done). The remaining elements of the public key are simulated as in the main proof. In Phase 1, it suffices to simulate the key for $(\mathcal{U}_\mathsf{s}, \mathbb{O})$ such that $\omega^\star$ does not satisfy $\mathbb{O}$. This can be done in exactly the same way as before (*cf.* Section A.1, first type), albeit it also includes two new elements as in Eq.(8) with $r = 0$. In Challenge phase, the term $C, \hat{C}, C'_x$ can be simulated as usual. In addition, it just sets $C_0 = \hat{C}^a$. The rest follows from the main proof.

Finally, we consider the case where the adversary announces the target pair of type $(\mathbb{S}^\star, \mathcal{U}_\mathsf{o})$ in the Init phase, *i.e.,* the challenge ciphertext will be in CP-ABE mode. In this case, the proof follows exactly the main proof that is instantiated with the selective target pair $(\mathbb{S}^\star, \{T_\mathsf{o}\})$. Note also that it suffices to simulate the key for $\mathsf{sk}_{(\psi, 2^{\mathcal{U}_\mathsf{o}})}$ such that ψ does not satisfy $\mathbb{S}^\star$. Such a key does not include the two new elements of Eq.(7).