Computationally Efficient Expressive Key-Policy Attribute Based Encryption Schemes with Constant-Size Ciphertext

Y. Sreenivasa Rao and Ratna Dutta

Indian Institute of Technology Kharagpur
Kharagpur-721302, India
{ysrao,ratna}@maths.iitkgp.ernet.in

Abstract. In this paper, we present two attribute based encryption (ABE) schemes for monotone access structure (MAS) in the key-policy setting, where secret key is generated according to a MAS, ciphertext is associated with a set of attributes and decryption is possible only if the attribute set satisfies the MAS. The first scheme is secure against chosen plaintext attacks (i.e., CPA secure) while the second scheme is secure against chosen ciphertext attacks (i.e., CCA secure). The security proofs are free from the random oracle heuristic. The most interesting features of both schemes are *constant-size* ciphertext, *constant* number of bilinear pairing evaluations and *low* computation cost (in terms of exponentiations) compared with previous schemes. We further propose two non-monotone access structure (nonMAS) variants, one is CPA secure and another is CCA secure, by using the idea of transforming a non-MAS over attributes to a MAS over attributes and their negation. These key-policy ABE schemes for nonMAS preserve the same functionality as that of MAS primitives. While the secret key in all our constructions has quadratic size in the number of attributes, the number of pairing evaluations is constant. The (CPA and CCA) security of all our schemes are proved under the decisional n-Bilinear Diffie-Hellman Exponent assumption over prime order groups in the selective model.

Keywords: key-policy, attribute-based encryption, constant-size ciphertext, (non-)monotone access structure, chosen ciphertext security.

1 Introduction

Functional Encryption (FE) [4] is a new version of public key encryption that facilitates sophisticated and flexible relations between the "parameters" of secret keys and ciphertexts where either (i) secret key is generated according to a parameter $\mathbb{A}$ and ciphertext is associated with another parameter W, yielding *Key-Policy* FE (KP-FE) or (ii) ciphertext is created according to a parameter $\mathbb{A}$ and secret key is associated with another parameter L, yielding *Ciphertext-Policy* FE (CP-FE). Decryption is successful in key-policy (or ciphertext-policy) FE if and only if a relation $\mathcal{R}^{KP}(\mathbb{A}, W)$ (or $\mathcal{R}^{CP}(L, \mathbb{A})$) holds. A FE is an Attribute

S. Qing et al. (Eds.): ICICS 2013, LNCS 8233, pp. 346–362, 2013.

Based Encryption (ABE) [1,2,3] if one of the parameters for ciphertext and secret key is a tuple of attributes, and the other is an access structure or monotone span program over a set of attributes, wherein the relation $\mathcal{R}^{KP}$ (or $\mathcal{R}^{CP}$) is an "inclusion" relation, i.e., $\mathcal{R}^{KP}(\mathbb{A}, W)$ (or $\mathcal{R}^{CP}(L, \mathbb{A})$) holds if and only if $W \in \mathbb{A}$ (or $L \in \mathbb{A}$). In this case, KP-FE (or CP-FE) is called as Key-Policy ABE (KP-ABE) [2] (or Ciphertext-Policy ABE (CP-ABE) [3]).

The first ABE system introduced by Sahai and Waters [1] is considered as a KP-ABE with threshold access policy. Later, Goyal et al. [2] designed the first KP-ABE for Monotone Access Structure (MAS). There are quite a number of KP-ABE schemes [6,5,4] that allow Non-Monotone Access Structure (nonMAS). While all the schemes mentioned so far are proven to be *selectively* Chosen Plaintext Attacks (CPA) secure where the adversary commits to her target before the simulation is set up, the works presented in [7,4] achieve *full* CPA security. Attrapadung et al. [8] proposed the first constant-size ciphertext selectively CPA secure KP-ABE for MAS as well as nonMAS over prime order groups with constant number of bilinear pairings, but secret key size is quadratic in the number of attributes. Independent of this work, Wang and Luo [9] proposed another KP-ABE for MAS with the same functionality as that of [8]. However, their scheme is proven to be secure in the random oracle assumption, while [8] does not use any such random oracle heuristic.

Security against Chosen Ciphertext Attacks (CCA) for ABE is a challenging task and has received little attention so far. The KP/CP-ABE schemes [2,11,4] used CHK (Canetti-Halevi-Katz) technique [10] to achieve CCA security in the standard model (without random oracles). They associate one-time signature keys with each encryption operation in combination with the delegation mechanism that uses key of one access structure $\mathbb{A}$ to construct a key for another access structure $\mathbb{A}'$ which is more restricted than $\mathbb{A}$. Resulting CCA secure ABE schemes have linear-size ciphertexts. Generalizing this idea, Yamada et al. [12] proposed a generic construction of CCA secure ABE and proved that any CPA secure ABE scheme preserving either delegatability or verifiability generically yields a CCA secure ABE primitive in the standard model. Note that it is easy to extend CPA security to CCA security in the random oracle model by applying Fujisaki-Okamoto transformation [13]. To the best of our knowledge, there is no constant-size ciphertext KP-ABE for expressive access policies (MAS as well as nonMAS) that is CCA secure in the standard model.

Our Contribution. The main focus of this article is to construct computationally efficient constant-size ciphertext KP-ABE schemes for Linear Secret-Sharing Scheme (LSSS)-realizable MAS as in [8,9] as well as nonMAS providing both CPA and CCA security in the standard model. To this end, we propose four KP-ABE schemes having the following unique features: (i) constant-size ciphertext, (ii) constant number of bilinear pairing evaluations, (iii) constant computation cost during encryption, (iv) $\mathcal{O}(|I|)$ exponentiations in decryption, where $|I|$ is the number of rows of LSSS matrix used in the decryption, and (v) secret key size $\mathcal{O}(\ell \cdot n)$ group elements, where ℓ is the number of rows in the user LSSS matrix, n is the number of attributes in the attribute space.

Table 1. Comparison of constant-size ciphertext KP-ABE for MAS and nonMAS

		SK Size	CT Size	Enc. Cost		Dec. Cost		
	Scheme	$E_{\mathbb{G}}$	$E_{\mathbb{G}} + E_{\mathbb{G}_T} + E_{\mathbb{Z}}$	$\mathsf{Ex}_{\mathbb{G}}$	$\mathsf{Ex}_{\mathbb{G}_T}$	$\mathsf{Ex}_{\mathbb{G}}$	Pairings	Security
MAS	[8,9]	$\mathcal{O}(\ell \cdot \overline{n})$	$2+1+0$	$\mathcal{O}(\phi)$	1	$\mathcal{O}(\|I\| \cdot \phi)$	2	sCPA
	Scheme I	$\mathcal{O}(\ell \cdot n)$	$2+1+0$	2	1	$\mathcal{O}(\|I\|)$	2	sCPA
	Scheme II	$\mathcal{O}(\ell \cdot n)$	$3+1+1$	5	1	$\mathcal{O}(\|I\|)$	6	sCCA
nonMAS	[8]	$\mathcal{O}(\ell \cdot \overline{n})$	$3+1+0$	$\mathcal{O}(\phi)$	1	$\mathcal{O}(\|I\| \cdot \phi)$	3	sCPA
	Scheme III	$\mathcal{O}(\ell \cdot n)$	$3+1+0$	3	1	$\mathcal{O}(\|I\|)$	3	sCPA
	Scheme IV	$\mathcal{O}(\ell \cdot n)$	$4+1+1$	6	1	$\mathcal{O}(\|I\|)$	9	sCCA

$E_{\mathbb{G}}$ (resp. $E_{\mathbb{G}_T}, E_{\mathbb{Z}}$) = number of elements in a group $\mathbb{G}$ (resp. $\mathbb{G}_T, \mathbb{Z}_p$), $\mathsf{Ex}_{\mathbb{G}}$ (resp. $\mathsf{Ex}_{\mathbb{G}_T}$) = number of exponentiations in a group $\mathbb{G}$ (resp. $\mathbb{G}_T$), ℓ = number of rows in the user LSSS access structure matrix, n = number of attributes used in the system, ϕ = number of attributes in a ciphertext, $\overline{n}$ = maximum number of attributes that can be associated with a ciphertext, $|I|$ = number of rows of LSSS matrix used in the decryption, sCPA (resp. sCCA) = selective CPA (resp. CCA) security, SK *= Secret Key and* CT *= Ciphertext. Note that $\overline{n} = n$ in the small universe setting.*

We use the threshold public key encryption framework of [14] to design our basic construction, referred as Scheme I, which realizes monotone LSSS access structure. We further extend our monotone KP-ABE approach to non-monotone KP-ABE by using the technique of [6] for transforming a nonMAS over attributes to a MAS over attributes and their negation. The resulting nonMAS KP-ABE construction is referred as Scheme III. Both the Scheme I and Scheme III are proven to be selectively CPA secure (as [8,9]) in the standard model under the decisional n-Bilinear Diffie-Hellman Exponent (n-BDHE) assumption over prime order bilinear groups. Finally, to enhance the CPA security of our basic constructions for MAS and nonMAS to CCA security, we incorporate the technique of CCA secure public key encryption of [15]. The generic conversions proposed in [12] transform the existing constant-size ciphertext KP-ABE schemes [8,9] to CCA secure schemes which no longer exhibit constant ciphertext-size as the conversion appends additional (dummy) attributes to the ciphertext. This new attribute addition incurs additional overhead which is linear to the number of attached attributes. In sum, we believe that our Scheme II for MAS and Scheme IV for nonMAS are the *first* CCA secure KP-ABE schemes with all the properties listed above.

In Table 1, we provide a detailed comparison between our schemes and the previous KP-ABE schemes [8,9] with constant-size ciphertext proposed so far. As the number, n, of attributes in the attribute universe is a factor of the secret key size, our constructions deal only with small attribute universe, thereby the attributes are fixed at system setup phase as in [7,1,2,16,8]. The KP-ABE schemes [8,9] are large universe constructions with a bound, $\overline{n}$, on the number of attributes that can be annotated to a ciphertext. For a fair comparison, we consider the small universe variants of the schemes [8,9]. Under this assumption, $\overline{n} = n$, i.e., there is no bound on the number of ciphertext attributes. Our

Table 2. Comparison of our large universe KP-ABE for MAS with [8,9]

		Public	SK Size	CT Size	Enc. Cost		Dec. Cost		
	Scheme	Key Size	$E_{\mathbb{G}}$	$E_{\mathbb{G}} + E_{\mathbb{G}_T}$	$\mathsf{Ex}_{\mathbb{G}}$	$\mathsf{Ex}_{\mathbb{G}_T}$	$\mathsf{Ex}_{\mathbb{G}}$	Pairings	Security
MAS	[8,9]	$\mathcal{O}(\overline{n})$	$\mathcal{O}(\ell \cdot \overline{n})$	$2+1$	$\mathcal{O}(\overline{n})$	1	$\mathcal{O}(\lvert I\rvert \cdot \overline{n})$	2	sCPA
	Scheme V	$\mathcal{O}(1)$	$\mathcal{O}(\ell^2)$	$(\phi+1)+1$	$\mathcal{O}(\phi)$	1	$\mathcal{O}(\lvert I\rvert)$	2	sCPA

schemes need only $\mathcal{O}(|I|)$ exponentiations and 2 pairing computations to decrypt any ciphertext, $|I|$ being the number of rows of LSSS matrix used in the decryption. On the contrary, the existing constant-size ciphertext KP-ABE schemes [8,9] perform $\mathcal{O}(|I| \cdot \phi)$ exponentiations followed by 2 pairing computations to decrypt a ciphertext, where ϕ denotes the number of attributes associated with a ciphertext. This could be very expensive in terms of exponentiations in certain situations. For instance, if a decryptor receives a ciphertext with 1000 attributes, our schemes require 20 exponentiations (if $|I| = 10$) and 2 pairing operations to decrypt that ciphertext. On the other hand, the schemes [8,9] require 10,000 exponentiations and 2 pairing operations to decrypt the same ciphertext. The encryptor executes 1000 exponentiations to compute the above ciphertext in [8,9], while that for our Scheme I is only 2. Thus, the schemes [8,9] in the large universe setting cannot yield directly KP-ABE constructions for small attribute universe that are computationally efficient, supporting expressive access policies and achieving constant-size ciphertext. We believe that our new constructions are of independent interest in the small universe setting as they outperform the KP-ABE schemes of [8,9] in terms of exponentiations, thereby can efficiently be deployed in practice.

By using the similar ideas in [16,14], our basic construction, Scheme I, can be extended to large universe setting (referred as Scheme V, see Section 5) wherein the attribute parameters are dynamically computed after the system setup by using a hash function, while the ciphertext-size is proportional to the number of attributes in it. However, it still preserves the decryption efficiency analogous to our small universe construction. The large universe constructions of [8,9] place a bound, $\overline{n}$, on the maximum number of attributes to encrypt each message in the system. This makes the system infeasible and the size of public key is proportional to this bound $\overline{n}$. On the other hand, Scheme V is free from any such system-wide limitations and exhibits constant-size public parameters. But, as in [9], the scheme is secure in the random oracle model. The secret key size of [8,9] increases by a factor of $\overline{n}$, while that for our Scheme V *only* increases by a factor of the number of attributes in user secret key. In sum, while all the proposed schemes present faster decryption capabilities over previous proposals, we achieve a controllable trade-off between the ciphertext size and the attribute universe size. In Table 2, we compare our large universe construction with the previous schemes [8,9]. Even though we show some of the improvements over previous schemes [8,9], the work of Attrapadung et al. [8] is a major step forward in designing expressive KP-ABE schemes with constant-size ciphertexts.

2 Background

Notation. Let $x \in_R X$ denote the operation of picking an element x uniformly at random from the set X. We denote the set $\{1, 2, \ldots, n\}$ as $[n]$.

In this section, we recall necessary background from [16,7].

Definition 1 (Access Structure). *Let U be the universe of attributes and $\mathcal{P}(U)$ be the collection of all subsets of U. Every subset $\mathbb{A}$ of $\mathcal{P}(U) \setminus \{\emptyset\}$ is called an access structure. An access structure $\mathbb{A}$ is said to be monotone access structure (MAS) if for any $C \in \mathcal{P}(U)$, with $C \supseteq B$ where $B \in \mathbb{A}$ implies $C \in \mathbb{A}$.*

2.1 Linear Secret-Sharing Schemes (LSSS)

Let U be the universe of attributes. A secret-sharing scheme $\Pi_{\mathbb{A}}$ for the access structure $\mathbb{A}$ over U is called *linear* (in $\mathbb{Z}_p$) if $\Pi_{\mathbb{A}}$ consists of the following two polynomial-time algorithms, where $\mathbb{M}$ is a matrix of size $\ell \times k$, called the *share-generating matrix* for $\Pi_{\mathbb{A}}$ and $\rho : [\ell] \to I_U$ is a row labeling function that maps each row of the matrix $\mathbb{M}$ to an attribute in $\mathbb{A}$, I_U being the index set of U.

(i) $\mathsf{Distribute}(\mathbb{M}, \rho, \alpha)$: This algorithm takes as input the share-generating matrix $\mathbb{M}$, row labeling function ρ and a secret $\alpha \in \mathbb{Z}_p$ which is to be shared. It randomly selects $z_2, z_3, \ldots, z_k \in_R \mathbb{Z}_p$ and sets $\boldsymbol{v} = (\alpha, z_2, z_3, \ldots, z_k) \in \mathbb{Z}_p^k$. It outputs a set $\{\boldsymbol{M_i} \cdot \boldsymbol{v} : i \in [\ell]\}$ of ℓ shares, where $\boldsymbol{M_i} \in \mathbb{Z}_p^k$ is the i-th row of matrix $\mathbb{M}$. The share $\lambda_{\rho(i)} = \boldsymbol{M_i} \cdot \boldsymbol{v}$ belongs to an attribute $\rho(i)$.

(ii) $\mathsf{Reconstruct}(\mathbb{M}, \rho, W)$: This algorithm will accept as input $\mathbb{M}, \rho$ and a set of attributes $W \in \mathbb{A}$. Let $I = \{i \in [\ell] : \rho(i) \in I_W\}$, where I_W is index set of the attribute set W. It returns a set $\{\omega_i : i \in I\}$ of secret reconstruction constants such that $\sum_{i \in I} \omega_i \lambda_{\rho(i)} = \alpha$, if $\{\lambda_{\rho(i)} : i \in I\}$ is a valid set of shares of the secret α according to $\Pi_{\mathbb{A}}$.

Lemma 1. *Let $(\mathbb{M}, \rho)$ be a LSSS access structure realizing an access structure $\mathbb{A}$ over the universe U of attributes, where $\mathbb{M}$ is share-generating matrix of size $\ell \times k$, and $W \subset U$. If $W \notin \mathbb{A}$ (in other words, W does not satisfy $\mathbb{M}$), there exists a polynomial time algorithm that outputs a vector $\boldsymbol{w} = (-1, w_2, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$, for each row i of $\mathbb{M}$ for which $\rho(i) \in I_W$.*

2.2 Bilinear Maps and Hardness Assumption

We use multiplicative cyclic groups $(\mathbb{G}, \mathbb{G}_T)$ of prime order p with an efficiently computable mapping $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ such that $e(u^a, v^b) = e(u, v)^{ab}$, $\forall\, u, v \in \mathbb{G}$, $a, b \in \mathbb{Z}_p$ and $e(g, g) \neq 1_T$, where 1_T is the unit element in $\mathbb{G}_T$.

Decisional n-BDHE Assumption. An algorithm (or distinguisher) $\mathfrak{D}$ for solving the decisional n-BDHE (Bilinear Diffie-Hellman Exponent) problem in $(\mathbb{G}, \mathbb{G}_T)$ takes as input a tuple $(\overrightarrow{y}_{a,s}, Z) \in \mathbb{G}^{2n+1} \times \mathbb{G}_T$, where $a, s \in_R \mathbb{Z}_p, g \in_R \mathbb{G}, g_i = g^{a^i}, \forall i \in [2n]$, $\overrightarrow{y}_{a,s} = (g, g^s, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$ and determines

whether $Z = e(g_{n+1}, g^s)$ or a random element in $\mathbb{G}_T$. The advantage of a 0/1-valued algorithm $\mathfrak{D}$ in solving the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is defined to be $\mathsf{Adv}_{\mathfrak{D}}^{n\text{-dBDHE}} = |\Pr[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1 | Z = e(g_{n+1}, g^s)]$
$- \Pr[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1 | Z \text{ is random}]\,|$.

Definition 2. *The decisional n-BDHE problem in* $(\mathbb{G}, \mathbb{G}_T)$ *is said to be* $(\mathcal{T}, \epsilon)$*-hard if the advantage* $\mathsf{Adv}_{\mathfrak{D}}^{n\text{-dBDHE}} \leq \epsilon$, *for any probabilistic polynomial-time (PPT) distinguisher* $\mathfrak{D}$ *running in time at most* $\mathcal{T}$.

2.3 KP-ABE Template

Let U be the attribute universe. A single trusted central authority (CA) manages all the attributes and its keys, and is responsible for issuing secret keys to users according to access structure of user attributes. The KP-ABE scheme consists of the following four algorithms.

Setup(κ, U)**.** This algorithm is run by the CA and takes as input a security parameter κ and the attribute universe U. It returns public key PK and master secret key MK. The secret key MK is kept secret by CA and the public key PK is made public.

KeyGen$(\mathsf{PK}, \mathsf{MK}, \mathbb{A})$**.** The CA runs this algorithm with the input PK, MK and an access structure $\mathbb{A}$. It outputs the secret key $\mathsf{SK}_{\mathbb{A}}$ associated with $\mathbb{A}$.

Encrypt(PK, M, W)**.** An encryptor will execute this algorithm with the input PK, a message M to be encrypted under a set W of attributes. It then returns a ciphertext CT_W in such a way that only the user with access structure $\mathbb{A}$ satisfied by W can decrypt CT_W.

Decrypt$(\mathsf{PK}, \mathsf{SK}_{\mathbb{A}}, \mathsf{CT}_W)$**.** This algorithm is run by decryptor and takes as input $\mathsf{PK}, \mathsf{SK}_{\mathbb{A}}$ and CT_W. It outputs the message M encrypted under a set W of attributes if the access structure $\mathbb{A}$ embedded in decryptor's secret key $\mathsf{SK}_{\mathbb{A}}$ is satisfied by W, otherwise decryption will fail.

2.4 Selective-Set Security Model for KP-ABE

We describe IND-sCPA (ciphertext indistinguishability under selective-set chosen plaintext attacks) security model in terms of a game $\mathsf{Game}^{\mathsf{IND-sCPA}}$ carried out between a challenger and an adversary. The challenger executes the relevant KP-ABE algorithms in order to answer the queries from the adversary. The game is as follows:

Init. The adversary announces a set W^* of attributes that he wishes to be challenged upon.

Setup. The challenger executes the **Setup** algorithm and gives public key PK to the adversary.

Query Phase 1. The adversary is allowed to make secret key queries for an access structure $\mathbb{A}$ subject to the constraint that W^* must not satisfy the access structure $\mathbb{A}$. The challenger then runs **KeyGen** algorithm and returns the corresponding secret key $\mathsf{SK}_{\mathbb{A}}$ to the adversary. This process can be repeated polynomial number of times.

Challenge. The adversary submits two equal length messages M_0, M_1. The challenger flips a random coin $\mu \in \{0,1\}$ and runs **Encrypt** algorithm in order to encrypt M_μ under W^*. The resulting challenge ciphertext CT_{W^*} is given to the adversary.

Query Phase 2. Query Phase 1 is repeated.

Guess. The adversary outputs a guess bit $\mu' \in \{0,1\}$ for the challenger's secret coin μ and wins if $\mu' = \mu$.

The advantage of an adversary $\mathcal{A}$ in the IND-sCPA game is defined to be $\mathsf{Adv}_{\mathcal{A}}(\mathsf{Game}^{\mathsf{IND-sCPA}}) = |\Pr[\mu' = \mu] - \frac{1}{2}|$, where the probability is taken over all random coin tosses of both adversary and challenger.

We note that the foregoing security model can easily be extended to IND-sCCA (ciphertext indistinguishability under selective-set chosen ciphertext attacks) security model by allowing decryption queries in Query Phase 1, 2, with the restriction that no decryption query is allowed on challenge ciphertext CT_{W^*}.

Definition 3. *A KP-ABE scheme is said to be $(\mathcal{T}, q, \epsilon)$-IND-sCPA secure if the advantage $\mathsf{Adv}_{\mathcal{A}}(\mathsf{Game}^{\mathsf{IND-sCPA}}) \leq \epsilon$, for any PPT adversary $\mathcal{A}$ running in time at most $\mathcal{T}$ that makes at most q secret key queries in the foregoing selective-set CPA security game.*

Definition 4. *A KP-ABE scheme is said to be $(\mathcal{T}, q, q_D, \epsilon)$-IND-sCCA secure if the advantage $\mathsf{Adv}_{\mathcal{A}}(\mathsf{Game}^{\mathsf{IND-sCCA}}) \leq \epsilon$, for any PPT adversary $\mathcal{A}$ running in time at most $\mathcal{T}$ that makes at most q secret key queries and q_D decryption queries in the selective-set CCA security game.*

3 KP-ABE for Monotone Access Structures

In this section, we first present our efficient KP-ABE scheme with constant-size ciphertext that provides selective CPA (sCPA) security. We further enhance the sCPA security to selective CCA (sCCA) security by using the technique of CCA secure public key encryption of [15]. In these two constructions, every monotone access structure (MAS) is represented by LSSS access structure $(\mathbb{M}, \rho)$.

3.1 Scheme I: Basic sCPA Secure Scheme

Setup(κ, U). On receiving the implicit security parameter κ, this algorithm generates a prime number p, a bilinear group $\mathbb{G}$, a generator $g \in_R \mathbb{G}$ and a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative groups of order p. It then chooses a random $\alpha \in_R \mathbb{Z}_p$ and $h_0 \in_R \mathbb{G}$, and for each attribute $att_j \in U$, it randomly chooses $h_j \in_R \mathbb{G}$, for all $j \in [n]$. The public key is $\mathsf{PK} = \langle p, g, h_0, Y = e(g,g)^\alpha, h_1, h_2, \ldots, h_n \rangle$ and the master secret key is $\mathsf{MK} = \alpha$.

KeyGen($\mathsf{PK}, \mathsf{MK}, (\mathbb{M}, \rho)$). Here $\mathbb{M}$ is a share-generating matrix of size $\ell \times k$ and ρ is a mapping from each row i of $\mathbb{M}$ to an attribute $att_{\rho(i)}$. The CA first executes $\mathsf{Distribute}(\mathbb{M}, \rho, \alpha)$ and obtains a set $\{\lambda_{\rho(i)} = \boldsymbol{M_i} \cdot \boldsymbol{v} : i \in [\ell]\}$ of ℓ shares, where $\boldsymbol{v} \in_R \mathbb{Z}_p^k$ such that $\boldsymbol{v} \cdot \mathbf{1} = \alpha$ (here, $\mathbf{1} = (1, 0, \ldots, 0)$ is a vector of length k). For each row $i \in [\ell]$, it chooses a random exponent $r_i \in_R \mathbb{Z}_p$ and computes

$D_i = g^{\lambda_{\rho(i)}}(h_0 h_{\rho(i)})^{r_i}, D'_i = g^{r_i}, D''_i = \left\{D''_{i,j} : D''_{i,j} = h_j^{r_i}, \forall j \in [n] \setminus \{\rho(i)\}\right\}$. The CA then returns the secret key $\mathsf{SK}_{(\mathbb{M},\rho)} = \langle(\mathbb{M},\rho), \{D_i, D'_i, D''_i : i \in [\ell]\}\rangle$ associated with $(\mathbb{M},\rho)$.

Encrypt(PK, M, W). To encrypt a message $M \in \mathbb{G}_T$ under a set W of attributes, the encryptor selects $s \in_R \mathbb{Z}_p$ and computes $C = MY^s, C_1 = g^s$ and $C_2 = (h_0 \prod_{att_j \in W} h_j)^s$. It outputs the ciphertext $\mathsf{CT}_W = \langle W, C, C_1, C_2\rangle$.

Decrypt$(\mathsf{PK}, \mathsf{SK}_{(\mathbb{M},\rho)}, \mathsf{CT}_W)$. The decryptor first runs $\mathsf{Reconstruct}(\mathbb{M}, \rho, W)$ to obtain a set $\{\omega_i : i \in I\}$ of reconstruction constants, where $I = \{i \in [\ell] : att_{\rho(i)} \in W\}$. If W satisfies the access structure $(\mathbb{M},\rho)$, then $\sum_{i\in I} \omega_i \lambda_{\rho(i)} = \alpha$. The decryptor computes E_1, E_2 as follows:

$$E_1 = \prod_{i\in I}\left(D_i \cdot \prod_{att_j \in W, j\neq\rho(i)} D''_{i,j}\right)^{\omega_i}, \; E_2 = \prod_{i\in I}(D'_i)^{\omega_i}.$$

The message M can be obtained by computing $C \cdot e(C_2, E_2)/e(C_1, E_1)$.

Theorem 1 (*Security Proof*). *If the attribute universe U has n attributes then our Scheme I is $(\mathcal{T}, q, \epsilon)$-IND-sCPA secure, assuming that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e$ and $\epsilon' = \epsilon/2$. Here, $\mathcal{T}_e$ denotes the running time of one exponentiation in $\mathbb{G}$.*

Proof. Suppose that an adversary $\mathcal{A}$ can $(\mathcal{T}, q, \epsilon)$-*break* our Scheme I in the IND-sCPA security model. We will show that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is *not* $(\mathcal{T}', \epsilon')$-hard.

Suppose a distinguisher $\mathfrak{D}$ is given the decisional n-BDHE challenge $(\overrightarrow{y}_{a,s}, Z)$, where $\overrightarrow{y}_{a,s} = (g, g^s, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$, $g_i = g^{a^i}$, and $Z = e(g_{n+1}, g^s)$ or Z is a random element of $\mathbb{G}_T$. Now, the distinguisher $\mathfrak{D}$ plays the role of a challenger in $\mathsf{Game}^{\mathsf{IND-sCPA}}$ and interacts with $\mathcal{A}$ in order to solve the decisional n-BDHE problem (i.e., $\mathfrak{D}$ attempts to output 1 if $Z = e(g_{n+1}, g^s)$ and 0 otherwise) as follows.

Init. The adversary $\mathcal{A}$ outputs the target attribute set W^*.

Setup. The distinguisher $\mathfrak{D}$ selects a random value $\alpha' \in_R \mathbb{Z}_p$ and implicitly sets $\alpha = \alpha' + a^{n+1}$ by letting $Y = e(g,g)^\alpha = e(g,g)^{\alpha'} e(g^a, g^{a^n})$.

The distinguisher $\mathfrak{D}$ then programs the parameters $\{h_i : i \in [n]\}$ as follows. For $i \in [n]$, $\mathfrak{D}$ chooses a random value $t_i \in_R \mathbb{Z}_p$ and computes $h_i = g^{t_i} g_{n+1-i}$. Furthermore, to program h_0, the distinguisher selects a random $t_0 \in_R \mathbb{Z}_p$ and computes $h_0 = g^{t_0} \prod_{att_j \in W^*} h_j^{-1}$. We note that the parameters h_i are distributed randomly due to the g^{t_i} factor, for $i = 0, 1, \ldots, n$.

Finally, the public key $\mathsf{PK} = \langle p, g, h_0, Y, h_1, h_2, \ldots, h_n\rangle$ will be given to the adversary $\mathcal{A}$.

Query Phase 1. In this phase, the adversary $\mathcal{A}$ requests for secret keys corresponding to the LSSS access structures $(\mathbb{M}, \rho)$ subject to the condition that W^* does not satisfy $\mathbb{M}$ and then the distinguisher responds as follows.

Let the size of a share-generating matrix $\mathbb{M}$ be $\ell \times k$. Since W^* does not satisfy $\mathbb{M}$, by Lemma 1, there exists a vector $\boldsymbol{w} = (-1, w_2, \ldots, w_k) \in \mathbb{Z}_p^k$ such that $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$, for all rows i where $att_{\rho(i)} \in W^*$.

The distinguisher randomly selects $y'_2, y'_3, \ldots, y'_k \in_R \mathbb{Z}_p$ and implicitly sets

$$\boldsymbol{v} = (\alpha' + a^{n+1}, -(\alpha' + a^{n+1})w_2 + y'_2, \ldots, -(\alpha' + a^{n+1})w_k + y'_k) \in \mathbb{Z}_p^k$$

which will be used for generating shares of α as in the original scheme. Note that $\boldsymbol{v}$ can be written as $\boldsymbol{v} = -(\alpha' + a^{n+1})\boldsymbol{w} + \boldsymbol{v'}$, where $\boldsymbol{v'} = (0, y'_2, \ldots, y'_k) \in \mathbb{Z}_p^k$. Observe that $\lambda_{\rho(i)} = \boldsymbol{M}_i \cdot \boldsymbol{v}$ contains the term a^{n+1} and hence $g^{\lambda_{\rho(i)}}$ contains terms of the form $g^{a^{n+1}} = g_{n+1}$ which is unknown to $\mathfrak{D}$. Therefore, $\mathfrak{D}$ must make sure that there are no terms of the form g_{n+1} involved in secret key components. To this end, the distinguisher implicitly creates suitable r_i values in such a way that the unknown terms will be canceled out automatically. Now, the secret key corresponding to each row $\boldsymbol{M_i}, i \in [\ell]$, of $\mathbb{M}$ is computed as one of the following two cases:

Case 1: For i where $att_{\rho(i)} \in W^*$.

In this case, the distinguisher randomly chooses $r'_i \in_R \mathbb{Z}_p$ and implicitly sets $r_i = r'_i - a^{\rho(i)}$. Since $att_{\rho(i)} \in W^*$, $\boldsymbol{M_i} \cdot \boldsymbol{w} = 0$ and hence $\boldsymbol{M_i} \cdot \boldsymbol{v} = -(\alpha' + a^{n+1})\boldsymbol{M_i} \cdot \boldsymbol{w} + \boldsymbol{M_i} \cdot \boldsymbol{v'} = \boldsymbol{M_i} \cdot \boldsymbol{v'}$. Then the distinguisher computes

$$D_i = g^{\boldsymbol{M_i} \cdot \boldsymbol{v'}} (h_0 h_{\rho(i)})^{r'_i} g_{\rho(i)}^{-t_0} \prod_{att_j \in W^*,\ j \neq \rho(i)} \left(g_{\rho(i)}^{t_j} \cdot g_{n+1-j+\rho(i)} \right),$$

$$D'_i = g^{r'_i} g_{\rho(i)}^{-1}, \quad D''_i = \left\{ D''_{i,j} : D''_{i,j} = h_j^{r'_i} g_{\rho(i)}^{-t_j} g_{n+1-j+\rho(i)}^{-1}, \forall j \in [n] \setminus \{\rho(i)\} \right\}.$$

Case 2: For i where $att_{\rho(i)} \notin W^*$, i.e., $\rho(i) \neq j$, for all $att_j \in W^*$.

Note that $\boldsymbol{M_i} \cdot \boldsymbol{v} = \boldsymbol{M_i} \cdot (\boldsymbol{v'} - \alpha' \boldsymbol{w}) - (\boldsymbol{M_i} \cdot \boldsymbol{w}) a^{n+1}$. In this case, the distinguisher selects a random $r'_i \in_R \mathbb{Z}_p$ and implicitly sets $r_i = r'_i + (\boldsymbol{M_i} \cdot \boldsymbol{w}) a^{\rho(i)}$. Then the secret key components are computed as

$$D_i = g^{\boldsymbol{M_i} \cdot (\boldsymbol{v'} - \alpha' \boldsymbol{w})} (h_0 h_{\rho(i)})^{r'_i} g_{\rho(i)}^{(\boldsymbol{M_i} \cdot \boldsymbol{w})(t_0 + t_{\rho(i)})} \cdot \prod_{att_j \in W^*} \left(g_{\rho(i)}^{-(\boldsymbol{M_i} \cdot \boldsymbol{w}) t_j} g_{n+1-j+\rho(i)}^{-(\boldsymbol{M_i} \cdot \boldsymbol{w})} \right),$$

$$D'_i = g^{r'_i} g_{\rho(i)}^{(\boldsymbol{M_i} \cdot \boldsymbol{w})}, \quad D''_i = \left\{ D''_{i,j} = h_j^{r'_i} g_{\rho(i)}^{(\boldsymbol{M_i} \cdot \boldsymbol{w}) t_j} g_{n+1-j+\rho(i)}^{(\boldsymbol{M_i} \cdot \boldsymbol{w})}, \forall j \in [n] \setminus \{\rho(i)\} \right\}.$$

Since $1 \leq \rho(i) \leq n$ and $j \neq \rho(i)$, the secret key components D_i, D'_i and D''_i do not contain any term which implicitly contains g_{n+1} and hence the distinguisher can correctly distribute the secret key components. Therefore, the distribution of the secret key is identical to that of the original scheme. Finally, the distinguisher sends the secret key $\mathsf{SK}_{(\mathbb{M},\rho)} = \langle (\mathbb{M}, \rho), \{D_i, D'_i, D''_i : i \in [\ell]\} \rangle$ associated with $(\mathbb{M}, \rho)$ to the adversary.

Challenge. The adversary $\mathcal{A}$ submits two equal length messages M_0 and M_1 to the distinguisher $\mathfrak{D}$. Now, the distinguisher flips a random coin $\mu \in \{0, 1\}$ and encrypts M_μ under the challenge attribute set W^*. The components of challenge ciphertext CT_{W^*} are computed as follows: $C = M_\mu Z \cdot e(g^s, g^{\alpha'}), C_1 = g^s, C_2 = (g^s)^{t_0}$. The challenge ciphertext $\mathsf{CT}_{W^*} = \langle W^*, C, C_1, C_2 \rangle$ is returned to $\mathcal{A}$.

If $Z = e(g_{n+1}, g^s)$, then the challenge ciphertext CT_{W^*} is a valid encryption of the message M_μ under the attribute set W^* as $C_1 = g^s$, $C_2 = (g^s)^{t_0} = (g^{t_0})^s =$

$(h_0 \prod_{att_j \in W^*} h_j)^s$ and $C = M_\mu Z \cdot e(g^s, g^{\alpha'}) = M_\mu \cdot e(g_{n+1}, g^s) \cdot e(g^s, g^{\alpha'}) = M_\mu \cdot e(g,g)^{(\alpha' + a^{n+1})s} = M_\mu \cdot e(g,g)^{\alpha s}$.

On the contrary, if Z is a random element in $\mathbb{G}_T$, then the challenge ciphertext CT_{W^*} is independent of μ in the adversary's view.

Query Phase 2. $\mathfrak{D}$ proceeds exactly as it did in Query Phase 1.

Guess. The adversary $\mathcal{A}$ outputs his guess $\mu' \in \{0,1\}$ on μ. If $\mu' = \mu$, then $\mathfrak{D}$ outputs 1 in the decisional n-BDHE game to guess that $Z = e(g_{n+1}, g^s)$; otherwise it outputs 0 to indicate that Z is a random element in $\mathbb{G}_T$.

If $Z = e(g_{n+1}, g^s)$, then the adversary's view in the above game is identical to that in a real attack. In that case $|\Pr[\mu = \mu'] - 1/2| > \epsilon$. On the other hand, if Z is a random element in $\mathbb{G}_T$, then $\mathcal{A}$ cannot obtain any information about M_μ and hence $\Pr[\mu = \mu'] = 1/2$. Since the events $Z = e(g_{n+1}, g^s)$ and Z is random element in $\mathbb{G}_T$ are equiprobable, it is easy to see that $\mathsf{Adv}_{\mathfrak{D}}^{n\text{-dBDHE}} > \epsilon/2$. Thus, the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is not $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e$ and $\epsilon' = \epsilon/2$. □

3.2 Scheme II: Extension to sCCA Security

Setup(κ, U). This algorithm generates a tuple $(p, \mathbb{G}, g, \mathbb{G}_T, e)$ according to the implicit security parameter κ. It then chooses a random $\alpha \in_R \mathbb{Z}_p$ and $h_0, h_1, \ldots, h_n, \delta_1, \delta_2, \delta_3 \in_R \mathbb{G}$. It also selects a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$. Now, it outputs the public key and master secret key as $\mathsf{PK} = \langle p, g, h_0, Y = e(g,g)^\alpha, h_1, h_2, \ldots, h_n, \delta_1, \delta_2, \delta_3, \mathcal{H} \rangle$ and $\mathsf{MK} = \alpha$, respectively.

KeyGen($\mathsf{PK}, \mathsf{MK}, (\mathbb{M}, \rho)$). This algorithm is similar to the KeyGen algorithm of sCPA secure construction given in Section 3.1.

Encrypt(PK, M, W). To encrypt a message $M \in \mathbb{G}_T$ under a set W of attributes, the encryptor selects at random $s, \gamma \in_R \mathbb{Z}_p$ and computes

$$C = MY^s,\ C_1 = g^s,\ C_2 = (h_0 \textstyle\prod_{att_j \in W} h_j)^s,\ C_3 = (\delta_1^\beta \delta_2^\gamma \delta_3)^s,$$

where $\beta = \mathcal{H}(W, C, C_1, C_2)$. The encryptor outputs the ciphertext CT_W as $\mathsf{CT}_W = \langle W, C, C_1, C_2, C_3, \gamma \rangle$.

Decrypt($\mathsf{PK}, \mathsf{SK}_{(\mathbb{M},\rho)}, \mathsf{CT}_W$). The decryptor first checks the following two identities: $e(g, C_2) \overset{?}{=} e(C_1, h_0 \prod_{att_j \in W} h_j)$ and $e(g, C_3) \overset{?}{=} e(C_1, \delta_1^\beta \delta_2^\gamma \delta_3)$, where $\beta = \mathcal{H}(W, C, C_1, C_2)$. If one of the two identities does not hold, decryption will fail. Otherwise, it will proceed similar to the Decrypt algorithm of sCPA secure construction given in Section 3.1.

Theorem 2 (*Security Proof*). *Assume that the attribute universe U has n attributes and collision-resistant hash function exists. Then our Scheme II is $(\mathcal{T}, q, q_D, \epsilon)$-IND-sCCA secure, assuming that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e + \mathcal{O}(1) \cdot q_D \cdot \mathcal{T}_p$ and $\epsilon' = (1 - q_D/p) \cdot \epsilon$. Here, $\mathcal{T}_e$ denotes the running time of one exponentiation in $\mathbb{G}$ and $\mathcal{T}_p$ denotes the running time of one pairing computation in $\mathbb{G}_T$.*

Proof. Suppose that there exists an adversary $\mathcal{A}$ which can $(\mathcal{T}, q, q_D, \epsilon)$-*break* our Scheme II in the IND-sCCA security model. We can then build a distinguisher $\mathfrak{D}$ which uses $\mathcal{A}$ to show that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is *not* $(\mathcal{T}', \epsilon')$-hard. On input the decisional n-BDHE challenge $(\overrightarrow{y}_{a,s}, Z)$, where $\overrightarrow{y}_{a,s} = (g, g^s, g_1, \ldots, g_n, g_{n+2}, \ldots, g_{2n})$, $g_i = g^{a^i}$, and $Z = e(g_{n+1}, g^s)$ or Z is a random element of $\mathbb{G}_T$, the distinguisher $\mathfrak{D}$ attempts to output 1 if $Z = e(g_{n+1}, g^s)$ and 0 otherwise. Now, $\mathfrak{D}$ plays the role of a challenger in $\mathsf{Game}^{\mathsf{IND-sCCA}}$ and interacts with $\mathcal{A}$ as follows.

Init. The adversary $\mathcal{A}$ outputs the target attribute set W^* that he wishes to be challenged upon.

Setup. This Setup phase is same as the Setup phase described in the proof of Theorem 1. In addition, the distinguisher $\mathfrak{D}$ randomly chooses $\tau_2, \tau_3, \theta_1, \theta_2, \theta_3 \in_R \mathbb{Z}_p$ and sets $\delta_1 = g_1 g^{\theta_1}, \delta_2 = g_1^{\tau_2} g^{\theta_2}, \delta_3 = g_1^{\tau_3} g^{\theta_3}$. Note here that $\delta_1, \delta_2, \delta_3$ are distributed randomly due to the g^{θ_i} factor. The public key $\mathsf{PK} = \langle p, g, h_0, Y, h_1, h_2, \ldots, h_n, \delta_1, \delta_2, \delta_3, \mathcal{H}\rangle$ will be given to the adversary $\mathcal{A}$, where $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$ is a collision-resistant hash function.

Query Phase 1. In this phase, the distinguisher $\mathfrak{D}$ answers secret key queries as well as decryption queries from the adversary.

Secret Key Query: On adversary's secret key query, the distinguisher proceeds exactly as it did in Query Phase 1 in the proof of Theorem 1.

Decryption Query: When $\mathfrak{D}$ is given a ciphertext $\mathsf{CT}_W = \langle W, C, C_1, C_2, C_3, \gamma\rangle$ as an input to decryption query, $\mathfrak{D}$ first computes $\beta = \mathcal{H}(W, C, C_1, C_2)$ and performs the following pairing test on ciphertext components

$$e(g, C_2) \overset{?}{=} e(C_1, h_0 \textstyle\prod_{att_j \in W} h_j) \text{ and } e(g, C_3) \overset{?}{=} e(C_1, \delta_1^\beta \delta_2^\gamma \delta_3).$$

If one of the two pairing test identities does not hold, it returns $\perp$. Otherwise, it checks whether $\beta + \gamma\tau_2 + \tau_3 = 0$ (this happens with probability at most $1/p$). If so, the distinguisher $\mathfrak{D}$ aborts (we refer to this event as abort) and outputs a random bit, else it returns

$$C \cdot e\left(C_3/C_1^{\beta\theta_1+\gamma\theta_2+\theta_3}, g_n^{(\beta+\gamma\tau_2+\tau_3)^{-1}}\right)^{-1} \cdot e\left(C_1, g^{\alpha'}\right)^{-1} = M.$$

Challenge. The adversary $\mathcal{A}$ submits two equal length messages M_0 and M_1 to the distinguisher $\mathfrak{D}$. Now, the distinguisher flips a random binary coin $\mu \in \{0,1\}$ and encrypts M_μ under the challenge attribute set W^*. The components of challenge ciphertext CT_{W^*} are computed as follows

$$C^* = M_\mu Z \cdot e(g^s,\ g^{\alpha'}),\ C_1^* = g^s,\ C_2^* = (g^s)^{t_0},\ C_3^* = (g^s)^{\beta^*\theta_1+\gamma^*\theta_2+\theta_3},$$

where $\beta^* = \mathcal{H}(W^*, C^*, C_1^*, C_2^*)$ and $\gamma^* = -(\beta^*+\tau_3)/\tau_2$. The challenge ciphertext $\mathsf{CT}_{W^*} = \langle W^*, C^*, C_1^*, C_2^*, C_3^*, \gamma^*\rangle$ is returned to the adversary $\mathcal{A}$.

If $Z = e(g_{n+1}, g^s)$, then the challenge ciphertext CT_{W^*} is a valid encryption of the message M_μ under the attribute set W^* as explained below.

$$C_1^* = g^s,\ C_2^* = (g^s)^{t_0} = (g^{t_0})^s = (h_0 \textstyle\prod_{att_j \in W^*} h_j)^s.$$

Since $\gamma^* = -(\beta^* + \tau_3)/\tau_2$, we have $\beta^* + \gamma^*\tau_2 + \tau_3 = 0$ and hence $C_3^* = (g^s)^{\beta^*\theta_1+\gamma^*\theta_2+\theta_3} = (g_1^s)^{\beta^*+\gamma^*\tau_2+\tau_3}(g^s)^{\beta^*\theta_1+\gamma^*\theta_2+\theta_3} = (\delta_1^{\beta^*}\delta_2^{\gamma^*}\delta_3)^s$. Finally, $C^* = M_\mu Z \cdot e(g^s, g^{\alpha'}) = M_\mu \cdot e(g_{n+1}, g^s) \cdot e(g^s, g^{\alpha'}) = M_\mu \cdot e(g,g)^{(a^{n+1}+\alpha')s} = M_\mu \cdot e(g,g)^{\alpha s}$.

If Z is a random element in $\mathbb{G}_T$, then the challenge ciphertext CT_{W^*} is independent of μ in the adversary's view.

Query Phase 2. The adversary $\mathcal{A}$ issues more secret key and decryption queries and the distinguisher $\mathfrak{D}$ responds as in **Query Phase 1**.

We point out here a couple of facts. First, the adversary is not allowed to make a decryption query on challenge ciphertext CT_{W^*}. If so, $\mathfrak{D}$ aborts. Second, if the adversary is able to create a ciphertext $\mathsf{CT}_W = \langle W^*, C, C_1, C_2, C_3, \gamma\rangle$ with $\beta^* = \mathcal{H}(W^*, C, C_1, C_2)$ such that $\mathsf{CT}_W \neq \mathsf{CT}_{W^*}$, this represents a collision in the hash function $\mathcal{H}$. However, the probability that this event happens is negligible since $\mathcal{H}$ is a collision-resistant hash function.

Guess. The adversary $\mathcal{A}$ outputs his guess $\mu' \in \{0,1\}$ on μ. If any abort happens, the distinguisher $\mathfrak{D}$ outputs 0. Otherwise, $\mathfrak{D}$ outputs 1 in the n-dBDHE game to guess that $Z = e(g_{n+1}, g^s)$ if $\mu' = \mu$, and it outputs 0 to indicate that Z is a random element in $\mathbb{G}_T$ if $\mu' \neq \mu$. Therefore, as long as $\mathfrak{D}$ does not abort in the simulation, $\mathfrak{D}$ can use the $\mathcal{A}$'s advantage to show that the decisional n-BDHE problem is not $(\mathcal{T}', \epsilon')$-hard. This can be checked as follows.

If $Z = e(g_{n+1}, g^s)$, then the distinguisher $\mathfrak{D}$ provides a perfect simulation and hence

$$\begin{aligned}\epsilon < \mathsf{Adv}_{\mathcal{A}}(\mathsf{Game}^{\mathsf{IND-CPA}}) &= \Pr\left[\mu' = \mu|[Z = e(g_{n+1}, g^s)] \wedge \overline{\mathsf{abort}}\ \right] - \tfrac{1}{2}\\ &= \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = e(g_{n+1}, g^s)] \wedge \overline{\mathsf{abort}}\ \right] - \tfrac{1}{2},\end{aligned}$$

i.e., $\Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = e(g_{n+1}, g^s)] \wedge \overline{\mathsf{abort}}\ \right] > \epsilon + 1/2$.

If Z is a random element $X \in \mathbb{G}_T$, then $\mathcal{A}$ cannot obtain any information about M_μ and therefore, $\Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = X] \wedge \overline{\mathsf{abort}}\ \right] = \frac{1}{2}$.

Since the event abort is independent of whether $Z = e(g_{n+1}, g^s)$ or a random element $X \in \mathbb{G}_T$, we have that

$$\Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = e(g_{n+1}, g^s)] \wedge \mathsf{abort}\ \right] = \tfrac{1}{2} \text{ and}$$
$$\Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = X] \wedge \mathsf{abort}\ \right] = \tfrac{1}{2}.$$

The probability of the event abort in the simulation is $\Pr[\mathsf{abort}] = q_D/p$, where q_D is the maximum number of decryption queries the adversary can make during simulation. Now,

$$\begin{aligned}\mathsf{Adv}_{\mathfrak{D}}^{n\text{-}\mathsf{dBDHE}} &= \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|Z = e(g_{n+1}, g^s)\right] - \Pr[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|Z = X]\\ &= \Pr[\mathsf{abort}] \cdot \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = e(g_{n+1}, g^s)] \wedge \mathsf{abort}\ \right]\\ &\quad + \Pr[\overline{\mathsf{abort}}] \cdot \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = e(g_{n+1}, g^s)] \wedge \overline{\mathsf{abort}}\ \right]\\ &\quad - \Pr[\mathsf{abort}] \cdot \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = X] \wedge \mathsf{abort}\ \right]\\ &\quad - \Pr[\overline{\mathsf{abort}}] \cdot \Pr\left[\mathfrak{D}(\overrightarrow{y}_{a,s}, Z) = 1|[Z = X] \wedge \overline{\mathsf{abort}}\ \right]\\ &> \frac{q_D}{p} \cdot \frac{1}{2} + (1 - \frac{q_D}{p}) \cdot (\epsilon + \frac{1}{2}) - \frac{q_D}{p} \cdot \frac{1}{2} - (1 - \frac{q_D}{p}) \cdot \frac{1}{2} = (1 - \frac{q_D}{p})\epsilon.\end{aligned}$$

Thus, the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is not $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e + \mathcal{O}(1) \cdot q_D \cdot \mathcal{T}_p$ and $\epsilon' = (1 - q_D/p) \cdot \epsilon$. □

4 KP-ABE Variants for Non-monotone Access Structures

This section is dedicated to the presentation of our constant-size ciphertext KP-ABE schemes for Non-Monotone Access Structure (nonMAS) that provide both sCPA and sCCA security.

To build a KP-ABE for nonMAS with constant-size ciphertext, we employ the *moving from MAS to nonMAS* technique [6] that represents non-monotone access structures in terms of monotone access structures with *negative* attributes ($\mathsf{NOT}\texttt{crypto}$ is a negative attribute of the attribute $\texttt{crypto}$). We discuss here the technique for completeness. For ease of reference, we call the attribute $\texttt{crypto}$, a *positive* attribute and we denote its negation $\mathsf{NOT}\texttt{crypto}$ by $\neg\texttt{crypto}$. Let U be a positive attribute universe.

Given a family $\mathfrak{F} = \{\Pi_{\mathbb{A}} : \mathbb{A} \in \mathsf{MA}\}$ of linear secret-sharing schemes for a set of possible monotone access structures MA, and $\widetilde{U} = U \bigcup \{\neg att : att \in U\}$ is the underlying attribute universe for each monotone access structure $\mathbb{A} \in \mathsf{MA}$, a family NM of non-monotone access structures can be defined as follows. For each access structure $\mathbb{A} \in \mathsf{MA}$ over $\widetilde{U}$, one defines a possibly non-monotone access structure $N_{\mathbb{A}}$ over U in the following way.

- For every set $W \subset U$, form $N(W) = W \bigcup \{\neg att : att \in U \setminus W\} \subset \widetilde{U}$.
- Now, define $N_{\mathbb{A}}$ by saying that W is authorized in $N_{\mathbb{A}}$ if and only if $N(W)$ is authorized in $\mathbb{A}$, i.e., $W \in N_{\mathbb{A}}$ iff $N(W) \in \mathbb{A}$.

The family of non-monotone access structures is $\mathsf{NM} = \{N_{\mathbb{A}} : \Pi_{\mathbb{A}} \in \mathfrak{F}\}$. Note that the non-monotone access structure $N_{\mathbb{A}}$ will have only positive attributes in its access sets.

We combine the foregoing methodology with our KP-ABE schemes for MAS in order to construct desired KP-ABE schemes for nonMAS.

4.1 Scheme III: sCPA Secure Construction

Setup(κ, U). This algorithm first generates $p, \mathbb{G}, \mathbb{G}_T, e$ according to the implicit security parameter κ. It then picks a random generator $g \in_R \mathbb{G}$, random elements $h_0, k_0 \in_R \mathbb{G}$ and a random exponent $\alpha \in_R \mathbb{Z}_p$. For each attribute $att_j \in U$, it randomly chooses $h_j, k_j \in_R \mathbb{G}$, for all $j \in [n]$. Now, it outputs the public key and master secret key respectively as

$$\mathsf{PK} = \langle p, g, h_0, k_0, Y = e(g,g)^{\alpha}, \{h_j, k_j\}_{j \in [n]} \rangle \text{ and } \mathsf{MK} = \alpha.$$

KeyGen($\mathsf{PK}, \mathsf{MK}, \widetilde{\mathbb{A}}$). Given a non-monotone access structure $\widetilde{\mathbb{A}}$ such that we have $\widetilde{\mathbb{A}} = N_{\mathbb{A}}$ for some monotone access structure $\mathbb{A}$ over $\widetilde{U} = U \bigcup \{\neg att : att \in U\}$ and associated with a linear secret sharing scheme $\Pi_{\mathbb{A}} = (\mathbb{M}_{\ell \times k}, \rho)$, this algorithm first runs $\mathsf{Distribute}(\mathbb{M}, \rho, \alpha)$ and obtains a set $\{\lambda_{\rho(i)} = \boldsymbol{M_i} \cdot \boldsymbol{v} : i \in [\ell]\}$ of ℓ shares, where $\boldsymbol{v} \in_R \mathbb{Z}_p^k$ such that $\boldsymbol{v} \cdot \mathbf{1} = \alpha$ (here, $\mathbf{1} = (1, 0, \ldots, 0)$ is a vector of length k). Note that each row $i \in [\ell]$ of $\mathbb{M}$ is associated with an attribute $\widetilde{att}_{\rho(i)} \in \{att_{\rho(i)}, \neg att_{\rho(i)}\}$. For each row $i \in [\ell]$, it chooses a random exponent $r_i \in_R \mathbb{Z}_p$ and computes

$$D_i = g^{\lambda_{\rho(i)}} (\widetilde{h}_0 \widetilde{h}_{\rho(i)})^{r_i}, D'_i = g^{r_i}, D''_i = \left\{ D''_{i,j} : D''_{i,j} = \widetilde{h}_j^{r_i}, \forall j \in [n] \setminus \{\rho(i)\} \right\},$$

where, for each $j = 0, 1, \ldots, n$, $\widetilde{h}_j = \begin{cases} h_j, \text{ if } \widetilde{att}_{\rho(i)} = att_{\rho(i)}, \\ k_j, \text{ if } \widetilde{att}_{\rho(i)} = \neg att_{\rho(i)}. \end{cases}$ It then returns the secret key $\mathsf{SK}_{\widetilde{\mathbb{A}}} = \langle \widetilde{\mathbb{A}}, \{D_i, D'_i, D''_i : i \in [\ell]\} \rangle$ associated with the non-monotone access structure $\widetilde{\mathbb{A}}$.

Encrypt(PK, M, W)**.** To encrypt a message $M \in \mathbb{G}_T$ under a set $W \subset U$ of attributes, this algorithm selects at random $s \in_R \mathbb{Z}_p$ and computes $C = MY^s, C_1 = g^s$, $C_2 = (h_0 \prod_{att_j \in W} h_j)^s$ and $C_3 = (k_0 \prod_{att_j \in W} k_j)^s$. It outputs the ciphertext $\mathsf{CT}_W = \langle W, C, C_1, C_2, C_3 \rangle$.

Decrypt$(\mathsf{PK}, \mathsf{SK}_{\widetilde{\mathbb{A}}}, \mathsf{CT}_W)$**.** This algorithm first checks whether $W \in \widetilde{\mathbb{A}}$. If not, it outputs $\perp$. Otherwise, since $\widetilde{\mathbb{A}} = N_{\mathbb{A}}$ for some monotone access structure $\mathbb{A}$ over $\widetilde{U}$ associated with a linear secret sharing scheme $\Pi_{\mathbb{A}} = (\mathbb{M}_{\ell \times k}, \rho)$, we have $N(W) \in \mathbb{A}$. It runs $\mathsf{Reconstruct}(\mathbb{M}, \rho, N(W))$ and obtains a set $\{\omega_i : i \in I\}$ of reconstruction constants such that $\sum_{i \in I} \omega_i \lambda_{\rho(i)} = \alpha$, where $I = \{i \in [\ell] : \widetilde{att}_{\rho(i)} \in N(W)\}$. Let $I^+ = \{i \in [\ell] : \widetilde{att}_{\rho(i)} = att_{\rho(i)} \in N(W)\}$ and $I^- = \{i \in [\ell] : \widetilde{att}_{\rho(i)} = \neg att_{\rho(i)} \in N(W)\}$. Then $I = I^+ \bigcup I^-$. It now computes E_1, E_2, E_3 as follows:

$$E_1 = \prod_{i \in I} \left(D_i \cdot \prod_{att_j \in W, j \neq \rho(i)} D''_{i,j} \right)^{\omega_i}, \ E_2 = \prod_{i \in I^+} (D'_i)^{\omega_i}, \ E_3 = \prod_{i \in I^-} (D'_i)^{\omega_i}.$$

The message M is obtained by computing $C \cdot e(C_2, E_2) \cdot e(C_3, E_3) / e(C_1, E_1)$.

Security Proof: The proof of the following theorem is straightforward from the proof of Theorem 1 with the modification that in the simulation, the secret key generation uses h_j elements for positive attributes and k_j elements for negative attributes. Due to page limitation, the detailed proof is omitted.

Theorem 3. *If the attribute universe U has n attributes then Scheme III is $(\mathcal{T}, q, \epsilon)$-IND-sCPA secure, assuming that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e$ and $\epsilon' = \epsilon/2$. Here, $\mathcal{T}_e$ denotes the running time of one exponentiation in $\mathbb{G}$.*

4.2 Scheme IV: Extension to sCCA Security

Similar to KP-ABE schemes for MAS, we can extend our Scheme III to sCCA secure KP-ABE construction for non-monotone access structure by employing the same technique used in Scheme II. We describe the sCCA secure scheme as a set of the following four algorithms.

Setup(κ, U)**.** This algorithm randomly selects $\delta_1, \delta_2, \delta_3 \in_R \mathbb{G}$ and a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$. The other public parameters and master secret are chosen analogous to the Setup algorithm of Scheme III. It finally outputs the public key and master secret key respectively as
$\mathsf{PK} = \langle p, g, h_0, k_0, Y = e(g,g)^\alpha, \{h_j, k_j\}_{j \in [n]}, \delta_1, \delta_2, \delta_3, \mathcal{H} \rangle$ and $\mathsf{MK} = \alpha$.

KeyGen$(\mathsf{PK}, \mathsf{MK}, \widetilde{\mathbb{A}})$**.** This algorithm acts as KeyGen algorithm of Scheme III.

Encrypt(PK, M, W)**.** To generate the ciphertext, this algorithm selects at random $s, \gamma \in_R \mathbb{Z}_p$ and computes $C = MY^s, C_1 = g^s$, $C_2 = (h_0 \prod_{att_j \in W} h_j)^s$, $C_3 = (k_0 \prod_{att_j \in W} k_j)^s$ and $C_4 = (\delta_1^\beta \delta_2^\gamma \delta_3)^s$, where $\beta = \mathcal{H}(W, C, C_1, C_2, C_3)$. It outputs the ciphertext $\mathsf{CT}_W = \langle W, C, C_1, C_2, C_3, C_4, \gamma \rangle$.

Decrypt$(\mathsf{PK}, \mathsf{SK}_{\widetilde{\mathbb{A}}}, \mathsf{CT}_W)$**.** This algorithm first checks the following identities: $e(g, C_2) \overset{?}{=} e(C_1, h_0 \prod_{att_j \in W} h_j)$, $e(g, C_3) \overset{?}{=} e(C_1, k_0 \prod_{att_j \in W} k_j)$ and $e(g, C_4) \overset{?}{=} e(C_1, \delta_1^\beta \delta_2^\gamma \delta_3)$, where $\beta = \mathcal{H}(W, C, C_1, C_2, C_3)$. If one of the three identities does not hold, decryption will fail. Otherwise, it will proceed similar to the Decrypt algorithm of Scheme III in order to recover the message M.

Theorem 4 **(*Security Proof*).** *Assume that the attribute universe U has n attributes and collision-resistant hash function exists. Then our Scheme IV is $(\mathcal{T}, q, q_D, \epsilon)$-IND-sCCA secure, assuming that the decisional n-BDHE problem in $(\mathbb{G}, \mathbb{G}_T)$ is $(\mathcal{T}', \epsilon')$-hard, where $\mathcal{T}' = \mathcal{T} + \mathcal{O}(n^2) \cdot q \cdot \mathcal{T}_e + \mathcal{O}(1) \cdot q_D \cdot \mathcal{T}_p$ and $\epsilon' = (1 - q_D/p) \cdot \epsilon$. Here, $\mathcal{T}_e$ denotes the running time of one exponentiation in $\mathbb{G}$ and $\mathcal{T}_p$ denotes the running time of one pairing computation in $\mathbb{G}_T$.*

5 Scheme V: Large Universe KP-ABE for MAS

In this section, we extend our basic construction Scheme I to the large attribute universe setting as the set of the following four algorithms.

Setup(κ)**.** Let $(p, \mathbb{G}, \mathbb{G}_T, e)$ be as in Section 4.1 and $U = \{0,1\}^*$ is assumed to be the attribute universe. Choose a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$, which will be modeled as a random oracle, to compute attribute values dynamically. Pick $\alpha \in_R \mathbb{Z}_p$ and set $Y = e(g,g)^\alpha$. The public key and master secret key are $\mathsf{PK} = \langle p, g, Y, H \rangle$ and $\mathsf{MK} = \alpha$, respectively.

KeyGen$(\mathsf{PK}, \mathsf{MK}, (\mathbb{M}, \rho))$**.** Here $\mathbb{M}$ is a share-generating matrix of size $\ell \times k$ and ρ is a mapping from each row i of $\mathbb{M}$ to an attribute $\rho(i) \in \{0,1\}^*$. Let L be the set of attributes appeared in LSSS access structure $(\mathbb{M}, \rho)$. The CA first executes $\mathsf{Distribute}(\mathbb{M}, \rho, \alpha)$ and obtains a set $\{\lambda_{\rho(i)} = \boldsymbol{M_i} \cdot \boldsymbol{v} : i \in [\ell]\}$ of ℓ shares, where $\boldsymbol{v} \in_R \mathbb{Z}_p^k$ such that $\boldsymbol{v} \cdot \mathbf{1} = \alpha$. For each row $i \in [\ell]$, it chooses $r_i \in_R \mathbb{Z}_p$ and computes
$D_i = g^{\lambda_{\rho(i)}} H(\rho(i))^{r_i}, D'_i = g^{r_i}, D''_i = \{D''_{i,y} : D''_{i,y} = H(y)^{r_i}, \forall y \in L \setminus \{\rho(i)\}\}$
The CA then returns the secret key $\mathsf{SK}_{(\mathbb{M},\rho)} = \langle (\mathbb{M}, \rho), \{D_i, D'_i, D''_i : i \in [\ell]\} \rangle$.

Encrypt(PK, M, W)**.** To encrypt a message $M \in \mathbb{G}_T$ under a set W of attributes, the encryptor selects $s \in_R \mathbb{Z}_p$ and computes

$$C = MY^s, C_1 = g^s, C_2 = \{C_{2,y} : C_{2,y} = H(y)^s, \forall y \in W\}.$$

It outputs the ciphertext $\mathsf{CT}_W = \langle W, C, C_1, C_2 \rangle$.

Decrypt$(\mathsf{PK}, \mathsf{SK}_{(\mathbb{M},\rho)}, \mathsf{CT}_W)$**.** The decryptor first runs $\mathsf{Reconstruct}(\mathbb{M}, \rho, W)$ to obtain a set $\{\omega_i : i \in I\}$ of reconstruction constants, where $I = \{i \in [\ell] : \rho(i) \in W\}$. If W satisfies the access structure $(\mathbb{M}, \rho)$, then $\sum_{i \in I} \omega_i \lambda_{\rho(i)} = \alpha$. The decryptor computes E_1, E_2, C'_2 as follows:

$$E_1 = \prod_{i \in I} \left(D_i \cdot \prod_{y \in W', y \neq \rho(i)} D''_{i,y} \right)^{\omega_i}, \; E_2 = \prod_{i \in I} (D'_i)^{\omega_i}, \; C'_2 = \prod_{y \in W'} C_{2,y},$$

where $W' = \{y \in W : \exists\ j \in I$ such that $\rho(j) = y\}$. The message M can be obtained by computing $C \cdot e(C_2', E_2)/e(C_1, E_1)$.

Note. Due to lack of space, the security proof will be given in the full version.

6 Conclusion

In this paper, we proposed efficient CPA as well as CCA secure KP-ABE schemes for both MAS and nonMAS with constant-size ciphertext and constant number of bilinear pairing computations. Security of all our schemes against selective adversary has been proven under the decisional n-BDHE assumption in the standard model. Our schemes outperform the existing schemes in terms of computation cost during encryption and decryption.

References

1. Sahai, A., Waters, B.: Fuzzy Identity-Based Encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
2. 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, pp. 89–98 (2006)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-Policy Attribute-Based Encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007)
4. Okamoto, T., Takashima, K.: Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 191–208. Springer, Heidelberg (2010)
5. Lewko, A., Sahai, A., Waters, B.: Revocation Systems with Very Small Private Keys. In: IEEE Symposium on Security and Privacy, pp. 273–285 (2010)
6. Ostrovksy, R., Sahai, A., Waters, B.: Attribute Based Encryption with Non-Monotonic Access Structures. In: ACM Conference on Computer and Communications Security, pp. 195–203 (2007)
7. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption. Cryptology ePrint report 2010/110 (2010)
8. Attrapadung, N., Herranz, J., Laguillaumie, F., Libert, B., de Panafieu, E., Ràfols, C.: Attribute-Based Encryption Schemes with Constant-Size Ciphertexts. Theor. Comput. Sci. 422, 15–38 (2012)
9. Chang-Ji, W., Jian-Fa, L.: A Key-policy Attribute-based Encryption Scheme with Constant Size Ciphertext. In: CIS 2012, pp. 447–451. IEEE (2012)
10. Canetti, R., Halevi, S., Katz, J.: Chosen-Ciphertext Security from Identity-Based Encryption. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 207–222. Springer, Heidelberg (2004)
11. Cheung, L., Newport, C.: Provably Secure Ciphertext Policy ABE. In: ACM Conference on Computer and Communications Security, pp. 456–465 (2007)
12. Yamada, S., Attrapadung, N., Hanaoka, G., Kunihiro, N.: Generic Constructions for Chosen-Ciphertext Secure Attribute Based Encryption. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds.) PKC 2011. LNCS, vol. 6571, pp. 71–89. Springer, Heidelberg (2011)

13. Fujisaki, E., Okamoto, T.: How to Enhance the Security of Public-Key Encryption at Minimum Cost. In: Imai, H., Zheng, Y. (eds.) PKC 1999. LNCS, vol. 1560, pp. 53–68. Springer, Heidelberg (1999)
14. Qin, B., Wu, Q., Zhang, L., Domingo-Ferrer, J.: Threshold Public-Key Encryption with Adaptive Security and Short Ciphertexts. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 62–76. Springer, Heidelberg (2010)
15. Lai, J., Deng, R.H., Liu, S., Kou, W.: Efficient CCA-Secure PKE from Identity-Based Techniques. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 132–147. Springer, Heidelberg (2010)
16. Waters, B.: Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization. Cryptology ePrint report 2008/290 (2008)