Decentralized attribute-based encryption (ABE) is a special form of multiauthority ABE systems, in which no central authority and global coordination are required other than creating the common reference parameters. In this paper, we propose a new decentralized ABE in prime-order groups by using extended dual system groups. We formulate some assumptions used to prove the security of our scheme. Our proposed scheme is fully secure under the standard $k$-Lin assumption in random oracle model and can support any monotone access structures. Compared with existing fully secure decentralized ABE systems, our construction has shorter ciphertexts and secret keys. Moreover, fast decryption is achieved in our system, in which ciphertexts can be decrypted with a constant number of pairings.

1. Introduction

Attribute-based encryption (ABE), which enables fine-grained access control, was first introduced by Sahai and Waters [1]. Subsequently, Goyal et al. [2] classified ABE as key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE). In KP-ABE, ciphertexts are associated with a set of attributes and secret keys are associated with access policies, while the opposite is true for CP-ABE. The ciphertext can be decrypted by secret keys if and only if the attributes satisfy the access policy.

Over the past decade, there have been a number of ABE schemes [3–9] proposed for supporting fairly expressive policies. However, the classical ABE system has only a single authority, which manages all attributes and issues private keys for all users. This may be unable to meet the requirements of some applications due to the lack of flexibility. There are three major aspects that impact the application value of single authority ABE systems. First, the single authority system failed to achieve the collaboration between different institutions since it cannot verify attributes across different organizations. Second, there exists key escrow problem in single authority system. The authority must be highly trustworthy as it can decrypt any ciphertext. Finally, key generation for all users that relied on a single authority is a huge workload and can easily become a performance bottleneck in the system. Furthermore, failure of the authority affects the whole system.

Multiauthority or decentralized ABE [10, 11] systems are put forward to address this issue. Lewko and Waters [11] provided the first fully secure decentralized ABE system. In their system, any party can become an authority by creating a public key. Authorities can issue private keys independently, and some authorities that go wrong will only affect the attributes in their domain and not the system as a whole. In addition, the scheme in [11] supports any monotone access structures.

Though the Lewko-Waters decentralized ABE scheme is expressive, the construction is based on composite-order bilinear group. The current research [12] showed that prime-order bilinear groups outperform composite-order groups in terms of both time efficiency and space efficiency. To be specific, elements with 3072 or 3248 bits are required for a 128-bit security level in composite-order groups according to NIST or ECRYPT II recommendations, while elements with
256 bits are sufficient in prime-order groups for the same security level. As for the time efficiency, [12] indicated that a pairing over an elliptic curve of composite order is 254 times slower than over a prime-order elliptic curve for the 128-bit security level. For the above reasons, it is preferable to design schemes on prime-order groups. In a subsequent work by Okamoto and Takashima [13], a decentralized ABE system on prime-order groups was presented by using dual pairing vector spaces [5]. The construction improves the efficiency of decentralized ABE systems, but there is still a significant performance penalty due to the required size of the vectors. Hence, it is worth constructing a more compact decentralized ABE system in prime-order setting.

We present a new construction of decentralized ABE by using extended dual system group (EDSG). Our proposed scheme is built on prime-order groups with better space and time efficiency and can be proved fully secure under standard $k$-Lin assumption in the random oracle model.

To prove that full security of decentralized ABE system is a challenging job, even using the powerful dual system encryption methodology [14, 15], [11] used two subgroups for semifunctional space. The first subgroup is used to hide nominal semifunctionality from the attacker’s view by appending blinding factors to each key at a time. The second subgroup is used to avoid leakage of information about the first one by switching the semifunctional components from the first subgroup to it.

Dual system groups (DSG) [16] are an attractive tool for simulating composite-order groups in the prime-order setting. In contrast to prior works [17–19], which attempted to maximize the properties satisfied by both composite-order and prime-order groups, the dual system groups seek to investigate the minimal properties needed for the application to dual system encryption. The benefit is that we can obtain more efficient and compact schemes, and that is why our scheme can reduce the size of ciphertext compared with previous work [13]. Unfortunately, we observe that dual system groups in [16] are insufficient for constructing fully secure decentralized ABE since it only has one semifunctional space. To overcome this, we extend the basis of dual system groups from $2k \times 2k$ matrix to $3k \times 3k$ matrix inspired by [20]. The first $k$-dimensional subspace is the normal space, the next $k$-dimension subspace is used to construct type 1 semifunctional secret keys, and the last $k$-dimension subspace is used to construct type 2 semifunctional secret keys. In addition, we also realize the left subgroup indistinguishability, right subgroup indistinguishability 1, and right subgroup indistinguishability 2. These assumptions are used to mimic the effect of the subgroup decision assumption in composite-order groups.

The paper is organized as follows. In Section 2, we introduced the related works. In Section 3, a brief summary of the relevant concepts in multiauthority CP-ABE and prime-order bilinear groups was presented. In Section 4, we gave our revised definition of dual system groups and realized it in the prime-order setting in Section 5. In Section 6, we gave our decentralized CP-ABE system, outlined the security proof, and discussed its efficiency. In Section 7, we concluded the paper.

2. Related Works

Attribute-based encryption was introduced by Sahai and Waters [1], which can encrypt a message for multiple receivers by their attributes, rather than designating recipient in advance. Subsequently, Goyal et al. [2] extended this idea and classified ABE system into two categories: key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). The first fully secure ABE system was presented by Lewko et al. [4]; all ABE systems can only be proved to be selective secure ones before that. In addition, several variants of ABE have been proposed. Ostrovsky et al. [21] showed how to realize negation by incorporating specific revocation schemes into the construction of [2]. Lewko et al. [22] provided a fully secure ABE system which is resilient to continual leakage. With regard to the public parameter optimization problems, large universe ABE system, in which the size of the attribute universe can be exponentially large, was proposed in [23, 24]. The first multi-authority ABE system was introduced in [10] by Chase, which has one central authority (CA) and multiple attribute authorities (AAs). Subsequently, Chase and Chow [25] removed the CA by using a distributed pseudorandom function. Both of [10, 25] can only support AND-gates policy. A multi-authority ABE that supports threshold policy was provided by Lin et al. [26]. CA is not required for their system. However, the authorities are fixed and they must interact with each other during setup. The multi-authority ABE proposed in [10, 25, 26] looked only at the KP-ABE setting. Müller et al. [27] proposed the first multi-authority CP-ABE supported policies written in disjunctive normal form (DNF) with one CA and multiple AAs. The system can be only proved to be secure in generic group model. In addition, all these above systems can only defend selective attacks; that is, the attacker must commit to a target access structure before setup phase. Lewko and Waters [11] first obtained a fully secure multi-authority CP-ABE by using dual system encryption technique [14, 15]. Their system is decentralized; that is, the authorities are equal and with no need for CA and can support any monotone access structures. They proved security under static assumptions in the random oracle model. Liu et al. [28] proposed a multi-authority CP-ABE where there are multiple CAs and AAs. In their system, all of the CAs must work together to issue an identity-related key to the user. They used $(n, n)$ threshold policy to distribute the master secret to prevent the authority decrypting ciphertexts independently. The system can be proved fully secure in the standard model. Scheme [11] is built on the composite-order group, which resulted in low efficiency of the systems. An improvement design was carried out in prime-order bilinear groups in [13]. Recently, Rouselakis and Waters [29] proposed an efficient large universe decentralized ABE system. However, the scheme only achieved static security, in which all queries (about both ciphertexts and secret keys) done by the attacker should be sent to the challenger immediately after seeing the global parameters.

In addition, some extension researches on multiauthority ABE have been proposed. Ma et al. [30] presented a multi-authority ABE with traitor tracing. The system is not
practical due to infeasible large sizes of public key and ciphertext. Li et al. [31] proposed a multi-authority CP-ABE scheme with accountability, which allows tracing the identity of a misbehaving user who leaked the decryption key to others. The system supported AND-gates policy. A large universe decentralized KP-ABE scheme was proposed in [32]. The system supported any monotone access policy and can be proved as selectively secure in the standard model. Gorasia et al. [33] presented a multi-authority CP-ABE scheme with fast decryption, which only supports threshold policy. Zhong et al. [34] proposed a decentralized CP-ABE scheme with hidden policy. It also supported user revocation but only achieved selective security. An adaptively secure multi-authority CP-ABE scheme with verifiable outsourced decryption was given in [35].

3. Preliminaries

Notation. We use $s \leftarrow_r S$ to denote that $s$ is picked randomly from a set $S$. We denote probabilistic polynomial-time by PPT. $[n]$ denotes the set $\{1, \ldots, n\}$ for any $n \in \mathbb{Z}^+$.

3.1. Prime-Order Bilinear Groups and Computational Assumptions

Prime-Order Bilinear Groups. The asymmetric prime-order group generator $\mathcal{G}$ takes a security parameter $\lambda$ as input and outputs $(p, G_1, G_2, G_T, g_1, g_2, e)$, where $G_1$, $G_2$, and $G_T$ are cyclic groups of prime order $p$, $g_1$, $g_2$ are generators of $G_1$, $G_2$, respectively, $e : G_1 \times G_2 \rightarrow G_T$ is an effective computable non-degenerate bilinear pairing, that is, $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$, and $g_T = e(g_1, g_2) \neq 1$.

Assumption 1 ($k$-Lin: the $k$-linear assumption in $G_1$). For any PPT adversary $A$, the advantage of $A$ is negligible in $\lambda$:

$$\text{Adv}^{(k)\text{-Lin}}_A = |\Pr[A(D, T_0) = 1] - \Pr[A(D, T_1) = 1]|,$$

where

$$D = (p, G_1, G_2, G_T, e, g_1, g_2, g_1^{a_1}, \ldots, g_1^{a_k}, g_1^{a_{k+1}}, g_1^{a_{k+2}}, \ldots),$$

$$T_0 = g_1^{a_{k+1}s_1 + \cdots + s_k},$$

$$T_1 = g_1^{a_1s_1 + \cdots + s_{k+1} + s_{k+2}},$$

$$s_1, \ldots, s_k \leftarrow \mathbb{Z}_p, a_1, \ldots, a_k, s_{k+1} \leftarrow \mathbb{Z}_p^*.$$

Assumption 2 ($(k, \ell)$-LLIN: the $(k, \ell)$-lifted linear assumption in $G_1$). For any PPT adversary $A$, the advantage of $A$ is negligible in $\lambda$:

$$\text{Adv}^{(k, \ell)\text{-LLIN}}_A = |\Pr[A(D, T_0) = 1] - \Pr[A(D, T_1) = 1]|,$$

where

$$D = (p, G_1, G_2, G_T, e, g_1, g_2, g_1^{a_1}, \ldots, g_1^{a_k}, g_1^{a_{k+1}}, \ldots),$$

$$g_1^{a_{k+1}s_1 + \cdots + s_k},$$

$$T_0 = g_1^{a_{k+1}s_1 + \cdots + s_k},$$

$$T_1 = g_1^{a_1s_1 + \cdots + s_k + s_{k+1}},$$

$$s_1, \ldots, s_k \leftarrow \mathbb{Z}_p, a_1, \ldots, a_k, s_{k+1} \leftarrow \mathbb{Z}_p^*,$$

$$(4)$$
Key Query Phase 1. In this phase, $\mathcal{A}$ makes key queries by submitting $(\text{GID}, \text{att}_i)$ to $\mathcal{B}$, where $\text{att}_i$ belonged to un corrupted authorities. $\mathcal{B}$ returns $SK_{\text{GID}_j}$ to $\mathcal{A}$.

Challenge. $\mathcal{A}$ submits two equal-length messages $m_0, m_1$ and an access policy $(\mathcal{M}, \rho)$ with the following constraint. We let $V$ denote the subset of attributes controlled by corrupt AAs. For each identity $\text{GID}, V_{\text{GID}}$ denotes the subset of attributes $\text{att}_i$ which $\mathcal{A}$ has queried. For each $\text{GID}$, we require that $V \cup V_{\text{GID}}$ cannot satisfy $(\mathcal{M}, \rho)$. $\mathcal{B}$ randomly chooses $\beta \in \{0, 1\}$ and encrypts $m_\beta$ under $(\mathcal{M}, \rho)$. It sends the ciphertext to $\mathcal{A}$.

Key Query Phase 2. $\mathcal{A}$ continually queries $\mathcal{B}$ as in phase 1 in the same constraint.

Guess. $\mathcal{A}$ outputs a guess $\beta'$ for $\beta$.

The adversary’s advantage is defined to be $| \Pr[\beta' = \beta] - 1/2 |$.

Definition 4. A multiauthority CP-ABE scheme is secure if, for all PPT adversaries, the advantage is negligible in the above security game.

4. Extended Dual System Groups

(i) $\text{sampP}(1^\lambda, 1^n)$: output:

- (a) Public parameter, $pp$, contains group description $(G, H, G_T)$, a nondegenerate bilinear map $e : G \times H \rightarrow G_T$, a linear map $\mu$ defined on $H$, and some additional parameters for $\text{sampG}$ and $\text{sampH}$.
- (b) Secret parameter, $sp$, contains $\tilde{h}^*, \tilde{h}^* \in H$ (where $\tilde{h}^*, \tilde{h}^* \neq 1$), and some parameters for $\text{sampG}$, $\text{sampH}$, $\text{sampG}$, and $\text{sampH}$.

(ii) $\text{sampGT}$: $\text{Im}(\mu) \rightarrow G_T$.

(iii) $\text{sampG}(pp)$: output $\tilde{g} = (g_0, g_1, \ldots, g_n) \in G^{n+1}$.

(iv) $\text{sampH}(pp)$: output $\tilde{h} = (h_0, h_1, \ldots, h_n) \in H^{n+1}$.

(v) $\text{sampG}(pp, sp)$: output $\tilde{g} = (\tilde{g}_0, \tilde{g}_1, \ldots, \tilde{g}_n) \in G^{n+1}$.

(vi) $\text{sampH}(pp, sp)$: output $\tilde{h} = (\tilde{h}_0, \tilde{h}_1, \ldots, \tilde{h}_n) \in H^{n+1}$.

(vii) $\text{sampG}(pp, sp)$: output $\tilde{g} = (\tilde{g}_0, \tilde{g}_1, \ldots, \tilde{g}_n) \in G^{n+1}$.

(viii) $\text{sampH}(pp, sp)$: output $\tilde{h} = (\tilde{h}_0, \tilde{h}_1, \ldots, \tilde{h}_n) \in H^{n+1}$.

The first four algorithms are used for normal ciphertexts and secret keys in the real system, while the remaining are only used for semifunctional ones in the security proof. We use $\text{sampG}_0$ to indicate the first element of $\tilde{g}$, that is, $g_0$.

Correctness. It needs to meet the following conditions.

- (Projective). For $h \in H$ and a random variable $s$, $\text{sampG}(\mu(h); s) = e(\text{sampG}_0(pp, s), h)$.

\begin{equation}
\text{(Associative). For all } \overrightarrow{g} \leftarrow \text{sampG}(pp) \text{ and } \overrightarrow{h} \leftarrow \text{sampH}(pp),
\quad e(g_0, h_i) = e(g_i, h_0), \quad i = 1, \ldots, n. \quad (6)
\end{equation}

Security. It needs to meet the following conditions.

- (Orthogonality) $(i) \mu(\tilde{h}^*) = \mu(\tilde{h}^*) = 1$. $(ii) e(\tilde{g}_0, \tilde{h}^*) = 1$. $(iii) e(\tilde{g}_0, \tilde{h}^*) = 1$.

- (Nondegeneracy). For all $\tilde{g}_0 \leftarrow \text{sampG}_0(pp, sp)$ and $\tilde{g}_0 \leftarrow \text{sampG}(pp, sp), e(\tilde{g}_0, \tilde{h}^*)$ and $e(\tilde{g}_0, \tilde{h}^*)$ are distributed uniformly over $G_T$.

- (H-Subgroup). The output of $\text{sampH}(pp)$ is distributed uniformly over a subgroup of $H^{n+1}$.

- (Left Subgroup Indistinguishability). For any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in $\lambda$:

\begin{equation}
\text{Adv}_{\mathcal{A}}^{\text{LS}}(\lambda) = \left| \Pr \left[ \mathcal{A} \left( pp, \overrightarrow{g} \right) = 1 \right] - \Pr \left[ \mathcal{A} \left( pp, \overrightarrow{g} \cdot \overrightarrow{\tilde{g}} \cdot \overrightarrow{\tilde{g}} \right) = 1 \right] \right| ,
\end{equation}

where

\begin{equation}
\text{(pp, sp)} \leftarrow \text{sampP}(1^\lambda, 1^n);
\quad \overrightarrow{g} \leftarrow \text{sampP}(pp);
\quad \overrightarrow{\tilde{g}} \leftarrow \text{sampG}(pp, sp);
\quad \overrightarrow{\tilde{g}} \leftarrow \text{sampG}(pp, sp). \quad (8)
\end{equation}

- (Right Subgroup Indistinguishability 1). For any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ is negligible in $\lambda$:

\begin{equation}
\text{Adv}_{\mathcal{A}}^{\text{RS1}}(\lambda) = \left| \Pr \left[ \mathcal{A} \left( pp, \overrightarrow{\tilde{g}} \cdot \overrightarrow{\tilde{g}} \cdot \overrightarrow{\tilde{g}}, \overrightarrow{\tilde{h}} \right) = 1 \right] - \Pr \left[ \mathcal{A} \left( pp, \overrightarrow{\tilde{g}} \cdot \overrightarrow{\tilde{g}} \cdot \overrightarrow{\tilde{g}}, \overrightarrow{\tilde{h}} \cdot \overrightarrow{\tilde{h}} \right) = 1 \right] \right| ,
\end{equation}

where

\begin{equation}
\text{(pp, sp)} \leftarrow \text{sampP}(1^\lambda, 1^n);
\quad \overrightarrow{\tilde{g}} \leftarrow \text{sampP}(pp);
\quad \overrightarrow{\tilde{g}} \leftarrow \text{sampG}(pp, sp);
\quad \overrightarrow{\tilde{g}} \leftarrow \text{sampG}(pp, sp);
\quad \overrightarrow{\tilde{h}} \leftarrow \text{sampH}(pp);
\quad \overrightarrow{\tilde{h}} \leftarrow \text{sampH}(pp, sp). \quad (9)
\end{equation}
\[ (Right \text{ Subgroup Indistinguishability 2}) \text{. For any PPT adversary } \mathcal{A}, \text{ the advantage of } \mathcal{A} \text{ is negligible in } \lambda: \]
\[
\text{Adv}^{\text{RS2}}_{\mathcal{A}}(\lambda) = \left| \Pr\left[ \mathcal{A}\left( \text{pp, } \vec{g}, \vec{h} \cdot \vec{g}, \vec{h} \cdot \vec{h} \right) = 1 \right] - \Pr\left[ \mathcal{A}\left( \text{pp, } \vec{g}, \vec{g} \cdot \vec{g}, \vec{h} \cdot \vec{h} \right) = 1 \right] \right|,
\]
where \[
\text{(pp, sp)} \leftarrow \text{SampP}\left(1^\lambda, 1^n\right) ;
\]
\[
\vec{g} \leftarrow \text{SampG}(\text{pp}) ;
\]
\[
\vec{h} \leftarrow \text{SampH}(\text{pp}) ;
\]
\[
\text{(ii) Define }
\]
\[
\mathcal{A}\left( \text{pp, } \vec{g}, \vec{g} \cdot \vec{g}, \vec{h} \cdot \vec{h} \right) \text{ where }
\]
\[
\mathcal{A}\left( \text{pp, } \vec{g}, \vec{g} \cdot \vec{g} \right) = \mathcal{A}\left( \text{pp, } \vec{g}, \vec{h} \cdot \vec{h} \right) = 1
\]

\[ (Parameter \text{ Hiding}) \text{. The following two distributions are identical: } \]
\[
\left\{ \begin{array}{l}
\text{pp, } \vec{g} \cdot \vec{g}, \vec{h} \cdot \vec{h} \\
\text{sp, } \vec{g} \cdot \vec{g} \cdot \vec{g} \cdot \vec{g}, \vec{h} \cdot \vec{h} \cdot \vec{h} \cdot \vec{h} \\
\end{array} \right\},
\]
where \[
\text{(pp, sp)} \leftarrow \text{SampP}\left(1^\lambda, 1^{n+1}\right) ;
\]
\[
\vec{g} = (\vec{g}_0, \ldots) \leftarrow \text{SampG}(\text{pp, sp}) ;
\]
\[
\vec{h} = (\vec{h}_0, \ldots) \leftarrow \text{SampH}(\text{pp, sp}) ;
\]
\[
\text{(iii) Sample } B \leftarrow \text{GL}_{2k}(\mathbb{Z}_p), \text{ and set } B^* = (B^{-1})^T, \]
\[
Y_1, \ldots, Y_n \leftarrow \mathbb{Z}_p^{3k \times 3k} ; R \text{ is a random full-rank diagonal matrix in } \mathbb{Z}_p^k \text{ whose bottom-right entry is a } 2k \text{-dimensional unit matrix; define }
\]
\[
D = \pi_t(B),
\]
\[
D_t = Y_1^T \pi_t(B),
\]
\[
E = \pi_M(B),
\]
\[
E_t = Y_1^T \pi_M(B),
\]
\[
F = \pi_R(B),
\]
\[
F_t = Y_1^T \pi_R(B).
\]
\[ (iv) \text{ define } \mu([k]_2) = [D^T k]_T \text{ for all } k \in \mathbb{Z}_p^{3k}. \]
\]
\[ Output \]
\[
\text{pp} := \left( (p, G, H, G_T, e) ,
\right.
\]
\[
\text{[D]_1, [Y_1^T D]_1, \ldots, [Y_n^T D]_1 },
\]
\[
[D^*_2, [Y_1^T D^*_2]_2, \ldots, [Y_n^T D^*_2]_2 \right). \]
\[ \text{SampGT(}[\bar{p}]_T). \text{ Pick } \vec{s} \leftarrow \mathbb{Z}_p^k \text{ and output } [\vec{s}^T \bar{p}]_T \in G_T. \]
\[ \text{SampG(pp). Pick } \vec{s} \leftarrow \mathbb{Z}_p^k \text{ and output }
\]
\[
([D^T \vec{s}]_1, [Y_1^T D^T \vec{s}]_1, \ldots, [Y_n^T D^T \vec{s}]_1) \in \left( G_1^k \right)^{n+1}. \]
\[ \text{SampH(pp). Pick } \vec{s} \leftarrow \mathbb{Z}_p^k \text{ and output }
\]
\[
([D^* \vec{s}]_2, [Y_1^T D^* \vec{s}]_2, \ldots, [Y_n^T D^* \vec{s}]_2) \in \left( G_2^k \right)^{n+1}. \]

5. Instantiating EDSG

We let \( \pi_t(\cdot), \pi_M(\cdot) \) and \( \pi_R(\cdot) \) be functions mapping from a \( 3k \times 3k \) matrix to its left-most \( k \) columns, the middle \( k \) columns, and the right-most \( k \) columns, respectively.

\[ \text{SampP}(1^\lambda, 1^n) \]
\[ (i) \text{ Run } (p, G, G_T, g_t, g_T, e) \leftarrow \mathcal{G}(1^\lambda). \]
\[ (ii) \text{ Define } (G, H, H_T, e) = (G_1^{3k}, G_2^{3k}, G_T, e). \]
\[ (iii) \text{ Sample } B \leftarrow \text{GL}_{2k}(\mathbb{Z}_p), \text{ and set } B^* = (B^{-1})^T, \]
\[ Y_1, \ldots, Y_n \leftarrow \mathbb{Z}_p^{3k \times 3k} ; R \text{ is a random full-rank diagonal matrix in } \mathbb{Z}_p^k \text{ whose bottom-right entry is a } 2k \text{-dimensional unit matrix; define }
\]
\[ D = \pi_t(B),
\]
\[ D_t = Y_1^T \pi_t(B),
\]
\[ E = \pi_M(B),
\]
\[ E_t = Y_1^T \pi_M(B),
\]
\[ F = \pi_R(B),
\]
\[ F_t = Y_1^T \pi_R(B).
\]
SampG(pp, sp). Pick ̂s ← Z_pk and output
\[
\left( \left[ E_1^s \right]_2, \left[ Y_1^i E_1^s \right]_2, \ldots, \left[ Y_n^i E_1^s \right]_2 \right) \in \left( G_2^{3k} \right)^{n+1}.
\]
(19)

SampH(pp, sp). Pick ̂r ← Z^k_p and output
\[
\left( \left[ E_1^r \right]_2, \left[ Y_1^r E_1^r \right]_2, \ldots, \left[ Y_n^r E_1^r \right]_2 \right) \in \left( G_2^{3k} \right)^{n+1}.
\]
(20)

SampG(pp, sp). Pick ̂s ← Z^k_p and output
\[
\left( \left[ F_1^s \right]_2, \left[ Y_1^s F_1^s \right]_2, \ldots, \left[ Y_n^s F_1^s \right]_2 \right) \in \left( G_2^{3k} \right)^{n+1}.
\]
(21)

SampH(pp, sp). Pick ̂r ← Z^k_p and output
\[
\left( \left[ F_1^r \right]_2, \left[ Y_1^r F_1^r \right]_2, \ldots, \left[ Y_n^r F_1^r \right]_2 \right) \in \left( G_2^{3k} \right)^{n+1}.
\]
(22)

Set ̃r* := [E* r]_2, ̃r* := [F* r]_2.

Correctness. We check correctness properties as follows.

(Projective). For all ̂k ∈ Z^k_p, ̂s ∈ Z^k_p:
\[
\text{SampGT} \left( \mu \left( \left[ \bar{k} \right]_2 \right), \bar{s} \right) = \left[ (D^s)^T \bar{k} \right]_T \times \left[ \left( D^s \right)^T \bar{r} \right]_T
\]
\[
= \left[ \left( D^s \right)^T \bar{k} \right]_T \times \left( \text{SampG0} (pp; \bar{s}) \right) \left( \left[ \bar{k} \right]_2 \right).
\]
(23)

(Associative). For all ̃s ∈ Z^k_p, ̃r ∈ Z^k_p, W ∈ Z^k×3k:
\[
e(\bar{g}_0, h_0) = \left( (D^s)^T \bar{y} \left( D^r \right)^T \bar{s} \right) = \left( D^s \bar{y} \bar{r} \right)^T = e(\bar{g}_0, h_0), \quad i = 1, \ldots, n.
\]
(24)

Security. We check the following security properties.

(Orthogonality)

(i) \( \mu(\bar{h}^*) = \mu(\bar{r}^*) = (1, \ldots, 1)^T \in \mathbb{C}_1 \).

(ii) \( e([E^s]_1, [E^r]_1) = 1_{G_1} \).

(iii) \( e([F^s]_1, [E^r]_1) = 1_{G_1} \).

(Nondegeneracy)

(i) \( e([E^s]_1, [E^r]_1) = e(g_1, g_2) \left( \begin{array}{c} 1 \\ 1 \end{array} \right) \).

(ii) \( e([F^s]_1, [E^r]_1) = e(g_1, g_2) \left( \begin{array}{c} 1 \\ 1 \end{array} \right) \).

With overwhelming probability, the inner product \( \bar{s} \cdot \bar{r} \) is distributed uniformly over \( Z_p \) and therefore \( e([E^s]_1, [E^r]_1) \) is distributed uniformly over \( G_1 \), and the same is true for \( e([F^s]_1, [E^r]_1) \).

(\H-Subgroup). This follows from the fact that \( Z_p^{3k} \) is an additive group.

Lemma 5 (left subgroup indistinguishability). For any PPT adversary \( \mathcal{A} \), there exists an adversary \( \mathcal{B} \) such that
\[
\text{Adv}_{\mathcal{A}}^{LS} (\lambda) \leq \text{Adv}_{\mathcal{B}}^{(k,2k)-\text{Lin}}.
\]
(25)

We may rewrite the LS advantage function as follows:
\[
\text{Adv}_{\mathcal{A}}^{LS} (\lambda) = \left| \Pr_{\mathcal{A} (pp; \bar{g})} \left[ \mathcal{A} \left( pp, \bar{g} \cdot \bar{g} \cdot \bar{g} \right) = 1 \right] \right|,
\]
(26)

where
\[
\left( pp, \bar{g} \right) \leftarrow \text{SampP} \left( \bar{I}^{k}, \bar{I}^{n} \right);
\]

\[
\left[ \begin{array}{c}
\bar{s} \\bar{s} \\bar{s}
\end{array} \right] \leftarrow \mathcal{B} \left( \bar{Z}_p \right),
\]

\[
\bar{g} := \left[ \begin{array}{c}
\bar{B} \\bar{B} \\bar{B}
\end{array} \right], \quad \bar{Y}_i \bar{B} \quad \bar{Y}_i \bar{B} \quad \bar{Y}_i \bar{B}
\]
(27)

(28)

Proof. Given an instance of \((k,2k)-\text{Lin}\) problem (i.e., \( \ell = 2k \)): \( g_1, g_2, g_1, \ldots, g_k, \beta \), \( \{ g_i \}_{i \in [2k]} \), \( \{ h_i \}_{i \in [2k]}, \{ a_i \}_{i \in [k]}, \{ a_i \}_{i \in [k]} \),

as input, where all \( s_{k+i} \) are either 0 or uniformly chosen from \( Z_p^{3k} \). \( \mathcal{B} \) implicitly sets
\[
\bar{s} = (s_1, \ldots, s_k)^T,
\]
(29)

Define \( W \in Z_p^{3k×3k} \) as
Sample $\overline{B} \leftarrow \text{GL}_{3k}(Z_p)$, $Y_1, \ldots, Y_n \leftarrow Z_p^{3k \times 3k}$, $\overline{r}_1, \ldots, \overline{r}_k \leftarrow Z_p^*$, set $\overline{B}^* := (\overline{B}^\top)^\top$, and implicitly set $(\overline{B}, \overline{B}^*) = (\overline{B}W, \overline{B}^*W^*)$.

$W := \left(\begin{array}{c|c}
\begin{array}{cccc}
a_1 & \ldots & a_k \\
b_{1,1} & \ldots & b_{1,k} & 1 \\
\vdots & \vdots & \ddots & \vdots \\
b_{k,1} & \ldots & b_{k,k} & 1 \\
b_{k+1,1} & \ldots & b_{k+1,k} & 1 \\
\vdots & \vdots & \ddots & \vdots \\
b_{2k,1} & \ldots & b_{2k,k} & 1
\end{array}
\end{array}\right)$

$W^* := (W^{-1})^\top :=
\left(\begin{array}{c|c}
\begin{array}{cccc}
a_{1}^{-1} & -a_1^{-1}b_{1,1} & \ldots & -a_1^{-1}b_{1,k} \\
\vdots & \vdots & \ddots & \vdots \\
a_{k}^{-1} & -a_k^{-1}b_{k,1} & \ldots & -a_k^{-1}b_{k,k} \\
1 & \ldots & \ldots & 1
\end{array}
\end{array}\right)$.

Simulating $pp$

$[\pi_L(B)]_1 = [\overline{B} \pi_L(W)]_1$, $[\pi_L(B^*)]_2 = [\overline{B}^* \pi_L(W^*R)]_2$.

Simulating the Challenge. $B$ simulates the challenge as

$\pi_L(W^*R) := \left(\begin{array}{c}
\overline{r}_1 \\
\vdots \\
\overline{r}_k \\
0^{2k \times k}
\end{array}\right)$.

$B \left(\begin{array}{c}
\frac{s}{s} \\
\frac{s}{s}
\end{array}\right) = \overline{B}W \left(\begin{array}{c}
\frac{s}{s} \\
\frac{s}{s}
\end{array}\right)$.

$Y_i^\top B \left(\begin{array}{c}
\frac{s}{s} \\
\frac{s}{s}
\end{array}\right) = Y_i^\top \overline{B}W \left(\begin{array}{c}
\frac{s}{s} \\
\frac{s}{s}
\end{array}\right)$.
If \( s_{k+i} = 0, i = 1, \ldots, 2k \), that is, \( \overrightarrow{s} = \overrightarrow{s} = 0 \), the output is \( \overrightarrow{g} \); otherwise, the output is \( \overrightarrow{g} \cdot \overrightarrow{g} \).

**Lemma 6** (right subgroup indistinguishability 1). For any PPT adversary \( \mathcal{A} \), there is an adversary \( \mathcal{B} \) such that

\[
\text{Adv}_{\mathcal{A}}^{\text{RSI}}(\lambda) \leq \text{Adv}_{\mathcal{B}}^{(k,k)-\text{LLin}}(\lambda).
\]

We may rewrite the RSI advantage function as follows:

\[
\text{Adv}_{\mathcal{A}}^{\text{RSI}}(\lambda) = \left| \Pr \left[ \mathcal{A} \left( \text{pp, g} \cdot \overrightarrow{s}, \overrightarrow{h} \right) = 1 \right] - \Pr \left[ \mathcal{A} \left( \text{pp, g} \cdot \overrightarrow{s}, \overrightarrow{h} \right) = 1 \right] \right|,
\]

where

\[
(\text{pp, sp}) \leftarrow \text{SampP} \left( 1^k, 1^n \right);
\]

\[
\overrightarrow{s}, \overrightarrow{s}, \overrightarrow{r}, \overrightarrow{r} \leftarrow_{\text{R}} \mathbb{Z}_p^{2k};
\]

\[
\overrightarrow{g} \cdot \overrightarrow{g} = \begin{bmatrix} \overrightarrow{s} \\ \overrightarrow{s} \end{bmatrix}, \quad Y_n^T \begin{bmatrix} \overrightarrow{s} \\ \overrightarrow{s} \end{bmatrix} = \ldots,
\]

\[
\begin{bmatrix} Y_n^T \begin{bmatrix} \overrightarrow{s} \\ \overrightarrow{s} \end{bmatrix} \end{bmatrix} \ldots,
\]

\[
\overrightarrow{g} = \begin{bmatrix} \overrightarrow{0} \\ \overrightarrow{0} \\ \overrightarrow{s} \end{bmatrix}, \quad Y_n^T \begin{bmatrix} \overrightarrow{0} \\ \overrightarrow{0} \\ \overrightarrow{s} \end{bmatrix} = \ldots,
\]

\[
\begin{bmatrix} Y_n^T \begin{bmatrix} \overrightarrow{0} \\ \overrightarrow{0} \\ \overrightarrow{s} \end{bmatrix} \end{bmatrix} \ldots,
\]

\[
\overrightarrow{h} = \begin{bmatrix} B^T \begin{bmatrix} \overrightarrow{r} \\ \overrightarrow{0} \end{bmatrix} \\ Y_1 B^T \begin{bmatrix} \overrightarrow{r} \\ \overrightarrow{0} \end{bmatrix} \end{bmatrix} = \ldots,
\]

\[
\begin{bmatrix} Y_1 B^T \begin{bmatrix} \overrightarrow{r} \\ \overrightarrow{0} \end{bmatrix} \end{bmatrix} = \ldots.
\]
\[
W = \begin{pmatrix}
1 & -a_1^{-1}b_{1,1} & \cdots & -a_1^{-1}b_{1,k} \\
\vdots & \vdots & \ddots & \vdots \\
1 & -a_k^{-1}b_{k,1} & \cdots & -a_k^{-1}b_{k,k}
\end{pmatrix}
\].

(40)

Simulating pp

\[
[\pi_L(B)]_1 = [\overline{B}\pi_L(W)]_1,
\]

\[
[Y_i^\pi L(B)]_1 = [Y_i^\pi \overline{B}\pi_L(W)]_1,
\]

\[
[\pi_L(B^*R)]_2 = [\overline{B}\pi_L(W^*R)]_2,
\]

\[
[Y_i\pi_L(B^*R)]_2 = [Y_i\overline{B}\pi_L(W^*R)]_2.
\]

(43)

Simulating \( g \cdot \overline{g}, \overline{g} \). Sample \( \overline{s}, \overline{s}, \overline{s} \leftarrow \mathbb{Z}_p^K \), and implicitly set

\[
(B, B^*) = (\overline{B}W, \overline{B}^*W^*),
\]

(41)

Then we can compute

\[
\begin{pmatrix}
\overline{s} \\
\overline{s}' \\
\overline{0}
\end{pmatrix} = W^{-1}\begin{pmatrix}
\overline{s}' \\
\overline{0}
\end{pmatrix},
\]

(44)

Then we can compute

\[
W^*R
\]

\[
= \begin{pmatrix}
a_1\overline{r}_1 & \cdots & a_k\overline{r}_k \\
\vdots & \ddots & \vdots \\
b_{k,1}\overline{r}_1 & \cdots & b_{k,k}\overline{r}_k
\end{pmatrix}
\].

(42)

(45)
Simulating the Challenge. \( \mathcal{B} \) simulates the challenge as

\[
\begin{bmatrix}
B^* R \begin{pmatrix}
\overrightarrow{r} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix}
= \begin{bmatrix}
\overrightarrow{B} \overrightarrow{W} R \begin{pmatrix}
\overrightarrow{r} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix}
\]

\[
\begin{bmatrix}
Y B^* \begin{pmatrix}
\overrightarrow{r} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix}
= \begin{bmatrix}
\overrightarrow{Y} \overrightarrow{B} \overrightarrow{W} R \begin{pmatrix}
\overrightarrow{r} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix}
\]

If \( r_{k+i} = 0, i = 1, \ldots, 2k \), that is, \( \overrightarrow{r} = \overrightarrow{0} \), the output is \( \overrightarrow{h} \); otherwise, the output is \( \overrightarrow{h} \cdot \overrightarrow{h} \).

Lemma 7 (right subgroup indistinguishability 2'). For any PPT adversary \( \mathcal{A} \), there exists an adversary \( \mathcal{B} \) such that

\[
\operatorname{Adv}^{\text{RS2}}_{\mathcal{B}} (\lambda) \leq \operatorname{Adv}^{(k,k)}_{\mathcal{B}} (\lambda) - \operatorname{LLin}.
\] (47)

The RS2' advantage function:

\[
\operatorname{Adv}^{\text{RS2}'}_{\mathcal{A}} (\lambda) := \left| \Pr \left[ \mathcal{A} \left( \operatorname{pp}, \overrightarrow{g}, \overrightarrow{g} \cdot \overrightarrow{g}, \overrightarrow{h} \cdot \overrightarrow{h} \right) = 1 \right] - \Pr \left[ \mathcal{A} \left( \operatorname{pp}, \overrightarrow{g}, \overrightarrow{g} \cdot \overrightarrow{g}, \overrightarrow{h} \cdot \overrightarrow{h} \right) = 1 \right] \right|
\] (48)

where

\( \operatorname{pp}, \operatorname{sp} \leftarrow \operatorname{SampP} \left( 1^\lambda, 1^n \right) \);

\( \overrightarrow{s}, \overrightarrow{s}, \overrightarrow{r}, \overrightarrow{r}, \overrightarrow{r} \leftarrow \mu \mathbb{Z}_p^k \);

\( \overrightarrow{g} = \begin{bmatrix}
\overrightarrow{B} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix} \\
\overrightarrow{Y} \overrightarrow{B} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix} \\
\overrightarrow{Y} \overrightarrow{g} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix} \)

\( \overrightarrow{r} = \begin{pmatrix}
\overrightarrow{r}_1 \\
\overrightarrow{r}_2 \\
\vdots \\
\overrightarrow{r}_k
\end{pmatrix} \)

\( \overrightarrow{g} \cdot \overrightarrow{g} = \begin{bmatrix}
\overrightarrow{B} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix} \\
\overrightarrow{Y} \overrightarrow{B} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix} \\
\overrightarrow{Y} \overrightarrow{g} \begin{pmatrix}
\overrightarrow{s} \\
\overrightarrow{0}
\end{pmatrix}
\end{bmatrix} \)

\( \overrightarrow{h} = \begin{pmatrix}
\overrightarrow{r}_1 \\
\overrightarrow{r}_2 \\
\vdots \\
\overrightarrow{r}_k
\end{pmatrix} \)

Proof. Given an instance of \((k,k)\)-LLin problem (i.e., \( \ell = k \))

\[
\begin{bmatrix}
\overrightarrow{g}_1, \overrightarrow{g}_2, \overrightarrow{g}_2, \ldots, \overrightarrow{g}_2, \overrightarrow{y}_1, \overrightarrow{y}_2, \overrightarrow{y}_3, \ldots, \overrightarrow{y}_k
\end{bmatrix}
\]

as input, where all \( s_{k+i} \) are either 0 or uniformly chosen from \( \mathbb{Z}_p^* \). \( \mathcal{B} \) samples \( r_1', \ldots, r_k' \leftarrow \mathbb{Z}_p^* \) and implicitly sets

\( \overrightarrow{r} = (r_1', \ldots, r_k')^\top \)

\( \overrightarrow{r} = (r_1, \ldots, r_k)^\top \)

\( \overrightarrow{r} = (r_{k+1}, \ldots, r_{2k})^\top \).
Define $W^* \in \mathbb{Z}_p^{3k \times 3k}$ as

$$W^* = \begin{pmatrix} 1 & \cdots & 1 \\ & \ddots & \vdots \\ & & 1 \end{pmatrix}$$

$$W^{-1} = \begin{pmatrix} 1 & \cdots & 1 \\ & \ddots & \vdots \\ & & 1 \end{pmatrix}$$

Then we can compute

$$W^* R$$

$$= \begin{pmatrix} r_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_{kk} \end{pmatrix}$$

(54)

Simulating $pp$

$$[\pi_L(B)]_1 = [\bar{B}\pi_L(W)]_1,$$

$$[Y_i\pi_L(B)]_1 = [Y_i\bar{B}\pi_L(W)]_1,$$

$$[\pi_L(B^* R)]_2 = [\bar{B}^* \pi_L(W^* R)]_2,$$

$$[Y_i\pi_L(B^* R)]_2 = [Y_i\bar{B}^* \pi_L(W^* R)]_2,$$

$$[\pi_R(B^* R)]_2 = [\bar{B}^* \pi_R(W^* R)]_2,$$

$$[Y_i\pi_R(B^* R)]_2 = [Y_i\bar{B}^* \pi_R(W^* R)]_2.$$
Then we can compute
\[
\begin{align*}
\begin{bmatrix}
B & 0 \\
0 & 0 \\
\end{bmatrix} & = \begin{bmatrix}
B & 0 \\
0 & 0 \\
\end{bmatrix}, \\
Y, B & = \begin{bmatrix}
Y, B & 0 \\
0 & 0 \\
\end{bmatrix}, \\
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
\end{bmatrix} & = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
\end{bmatrix}, \\
Y, B & = \begin{bmatrix}
Y, B & 0 \\
0 & 0 \\
\end{bmatrix}.
\end{align*}
\]
(59)

Simulating the Challenge. \( \mathcal{B} \) simulates the challenge as
\[
\begin{align*}
\begin{bmatrix}
B^* R & 0 \\
0 & 0 \\
\end{bmatrix} & = \begin{bmatrix}
B^* R & 0 \\
0 & 0 \\
\end{bmatrix}, \\
Y, B^* & = \begin{bmatrix}
Y, B^* & 0 \\
0 & 0 \\
\end{bmatrix}, \\
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
\end{bmatrix} & = \begin{bmatrix}
0 & 0 \\
0 & 0 \\
\end{bmatrix}, \\
Y, B^* & = \begin{bmatrix}
Y, B^* & 0 \\
0 & 0 \\
\end{bmatrix}.
\end{align*}
\]

Hence, right subgroup indistinguishability 2 is true.

Lemma 8 (parameter hiding). The following are identically distributed:

\[
\begin{align*}
\left\{ pp, \left[ \pi_M \left( B, s \right) + \pi_R \left( B, s \right) \right]_{R \in \mathbb{Z}_p^k} \right\}, & \left\{ Y_i \left[ \pi_M \left( B, s \right) + \pi_R \left( B, s \right) \right]_{R \in \mathbb{Z}_p^k} \right\}, \\
\left\{ \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) \right\}, & \left\{ Y_i \left[ \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) \right] \right\}, \\
\left\{ \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) \right\}, & \left\{ Y_i \left[ \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) \right] \right\}, \\
\left\{ pp, \left[ \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) \right]_{R \in \mathbb{Z}_p^k} \right\}, & \left\{ Y_i \pi_M \left( B^* R, \hat{r} \right) + \pi_R \left( B^* R, \hat{r} \right) + \tilde{u}_i \pi_M \left( B^* R, \hat{r} \right) + \tilde{u}_i \pi_R \left( B^* R, \hat{r} \right) \right\}
\end{align*}
\]
(62)

where \( \tilde{s}, \tilde{s}, \tilde{r}, \tilde{r} \) and \( \tilde{r}, \tilde{r} \) are \( \mathbb{Z}_p^k \), and \( \tilde{u}_i, \tilde{u}_i \) are \( \mathbb{Z}_p^k \).

Proof. Sample \( B \leftarrow \mathbb{G}_k \) and set \( B^* = \left( B^{-1} \right)^\top \), \( Y_1, \ldots, Y_n \leftarrow \mathbb{Z}_p^{3k \times 3k} \); \( R \) is a random full-rank diagonal matrix in \( \mathbb{Z}_p^k \) whose bottom-right entry is a 2k-dimensional unit matrix:

\[
\begin{align*}
D = \pi_k \left( B \right), \\
D_1 = Y_i \pi_k \left( B \right), \\
E = \pi_k \left( B \right), \\
E_1 = Y_i \pi_k \left( B \right), \\
F = \pi_k \left( B \right),
\end{align*}
\]
Hence,

\[ D^* = \pi_L(B^*) \],

\[ E^* = \pi_M(B^*) \],

\[ F^* = \pi_R(B^*) \].

Define \( V_1 = \pi_M(B^*)\pi_M(B)^\top \), \( V_2 = \pi_R(B^*)\pi_R(B)^\top \), and \( Y_i = Y_i + \tilde{u}_iV_1 + \bar{u}_iV_2 \). Then

\[ Y_i^\top B = Y_i^\top B + \bar{u}_iV_1^\top B + \tilde{u}_iV_2^\top B \]

\[ Y_i' B^* R = Y_iB^* R + \bar{u}_iV_1B^* R + \tilde{u}_iV_2B^* R \].

Observe that

\[ V_i^\top B = \pi_M(B) \left( \pi_M(B^*) \right)^\top \]

\[ = \pi_M(B) \begin{pmatrix} 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & \cdots & 0 \end{pmatrix} \]

\[ = \begin{pmatrix} \pi_M(B) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \pi_M(B) & 0 & \cdots & 0 \end{pmatrix} \]

\[ V_iB^* R = \pi_M(B^*) \left( \pi_M(B) \right)^\top R \]

\[ = \begin{pmatrix} \pi_M(B^*) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \pi_M(B^*) & 0 & \cdots & 0 \end{pmatrix} \]

\[ Y_i' B^* R = \pi_M(B^*) \left( \pi_M(B) \right)^\top R \]

\[ = \begin{pmatrix} \pi_M(B^*) & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \pi_M(B^*) & 0 & \cdots & 0 \end{pmatrix} \]

Hence, \( Y_i^\top B = (D_i^* + \tilde{u}_iE_i + \bar{u}_iF_i) \)

\[ Y_i' B^* R = (D_i^* + \tilde{u}_iE_i + \bar{u}_iF_i) \]

(i) If \( \tilde{u}_i = \bar{u}_i = 0 \), then we obtain the first distribution.

(ii) If \( \tilde{u}_i, \bar{u}_i \leftarrow \mathbb{Z}_p \), then we obtain the second distribution.

6. Our Scheme

This section presents our decentralized CP-ABE system. Recall that \( \pi_L(\cdot) \), \( \pi_M(\cdot) \), and \( \pi_R(\cdot) \) are functions mapping from a \( 3k \times 3k \) matrix to its left \( k \) columns, middle \( k \) columns, and right \( k \) columns, respectively. We use the left \( k \)-dimension subspaces to generate the normal ciphertexts and secret keys. The next two are only used in the security proof. The hash function \( H \) maps global identities to random elements in \( \mathbb{F}_l \), which is used as a random oracle in the security proof.

6.1. Construction

GlobalSetup(\( 1^k \)). Sample \( B \leftarrow \text{GL}_{3k}(\mathbb{Z}_p) \) and set \( B^* = (B^*)^\top \). Output

\[ \text{GP} = \left( B, B^*, \tilde{u}_i, \bar{u}_i, \pi_L(B), \pi_R(B), H \right) \].

Authority Setup(GP). For each attribute \( att_i \) belonging to the authority, the authority samples \( k_i \leftarrow \mathbb{Z}_p^{3k} \), \( Y_i \leftarrow \mathbb{Z}_p^{3k \times 3k} \) and outputs

\[ \text{APK} := \left( k_i, Y_i \right) \].

\[ \text{Enc}_{\{\text{APK}\}}(\text{GP}, \text{M}, \rho, m) \). Input a message \( m \), a matrix \( M \in \mathbb{Z}_{p^k}^{n \times d} \) with \( \rho \) (in our system, we restrict the fact that \( \rho \) is injective) mapping its rows to attributes, the global parameters, and the public keys of the relevant authorities. Pick \( U_1, \ldots, U_t \leftarrow \mathbb{Z}_p^{3k \times 3k} \), \( Z \leftarrow \mathbb{Z}_p^{3k} \), \( V = (v_0, v_2, \ldots, v_t) \leftarrow \mathbb{Z}_p^t \). We let \( M_x \) denote row \( x \) of \( M \), and \( \lambda_x = M_x \cdot V \). The ciphertext is

\[ C = m \cdot e(g_1, g_2)^{\lambda_x}, \]

\[ C_0 = \left[ \pi_L(B) \right]_1, \]

\[ C_{1,x} = e(g_1, g_2)^{\lambda_x} e(g_1, g_2)^{\pi_L(B) \bar{s}}, \]

\[ C_{2,x} = \left( e(g_1, g_2)^{\pi_L(B) \bar{s}} \right)^{ \pi_M(B) \bar{s}} M_x^\top + Y_i^\top \left( \pi_M(B) \bar{s} \right)_1 \]

KeyGen(GP, GID, ASK, \( att_i \)). Compute a key for GID for attribute \( att_i \) belonging to authority \( AA_i \) as follows:

\[ \text{SK}_{\text{GID},i} = \tilde{g}_2^\lambda H(GID)^{Y_i}. \]

Dec(GP, SK, CT). The secret keys \( \{\text{SK}_{\text{GID},i}\} \) correspond to a subset of rows \( M_x \) of \( M \). If \( (1, 0, \ldots, 0) \) is in the span of \( M_x \), then \( \omega_1, \ldots, \omega_i \in \mathbb{Z}_p \) is computed such that
Then, compute
\[
\prod_x \left( C_{1,x} \cdot e \left( H(GID), C_{2,x} \right)^{w_x} \right) = \prod_x C_{1,x} \cdot e \left( H(GID), \prod_x C_{2,x} \right) e \left( \prod_x C_{2,x} \right) = e \left( g_1, g_2 \right)^{\sum_x w_x M_2} = e \left( g_1, g_2 \right)^{\pi \cdot t},
\]
(72)

6.2. Security Proof. We define the semifunctional ciphertext and secret key as follows.

Semifunctional Ciphertext. We let \( C', C'_0, C'_{1,x}, C'_{2,x} \) denote the normal ciphertext. The semifunctional ciphertext takes the following form:
\[
C = C',
C_0 = C'_0 \cdot \left[ \pi_M(B) \cdot s + \pi_R(B) \cdot s \right]^1,
C_{1,x} = C'_{1,x} \cdot e(\gamma_2, \eta_2) \cdot \left[ \pi_M(B) \cdot s + \pi_R(B) \cdot s \right],
C_{2,x} = C'_{2,x} \cdot e(\gamma_2, \eta_2) \cdot \left[ \pi_M(B) \cdot s + \pi_R(B) \cdot s \right],
\]
(73)
\[
\sum_x \omega_x M_2 = (1, 0, \ldots, 0).
\]

We define the semifunctional ciphertext \( \tilde{C} \) and secret key as follows.

Semifunctional Secret Key. There are two types of semifunctional keys. Type 1 semifunctional key takes the following form:
\[
H(GID) = \left[ \pi_L(B^R) + \pi_M(B^R) \cdot \gamma_2 \right]_2,
\]
(74)
\[
SK_{GID,1} = g_2^T \cdot H(GID) \cdot \gamma_1,
= \left[ k_j + \gamma_1 \left( \pi_L(B^R) + \pi_M(B^R) \cdot \gamma_2 \right) \right]_2.
\]
Type 2 semifunctional key takes the following form:
\[
H(GID) = \left[ \pi_L(B^R) + \pi_R(B^R) \cdot \gamma_2 \right]_2,
\]
(75)
\[
SK_{GID,2} = g_2^T \cdot H(GID) \cdot \gamma_1,
= \left[ k_j + \gamma_1 \left( \pi_L(B^R) + \pi_R(B^R) \cdot \gamma_2 \right) \right]_2.
\]

When a semifunctional key is used to decrypt a semifunctional ciphertext, the additional terms
(i) type 1 semifunctional key: \( e(g_1, g_2)^{\pi_2 (B^R) \cdot \pi_2 (B) \cdot \gamma_2} \),
(ii) type 2 semifunctional key: \( e(g_1, g_2)^{\pi_2 (B^R) \cdot \pi_2 (B) \cdot \gamma_2} \)
prevent decryption.

Game Sequence. We let \( Adv_{\mathcal{A}}^{Game_x} \) denote the advantage of \( \mathcal{A} \) in \( Game_x \).

(i) \( Game_0 \): it is the real security game.
(ii) \( Game_1 \): there is no difference with \( Game_0 \) except that challenge ciphertext becomes semifunctional.
(iii) \( Game_{2,1} \) for \( j = 1, \ldots, q \); there is no difference with \( Game_1 \) except that the first \( j \) keys revealed to \( \mathcal{A} \) become semifunctional of type 2, and the \( j \)'th key becomes semifunctional of type 1.
(iv) \( Game_{2,2} \) for \( j = 1, \ldots, q \); there is no difference with \( Game_1 \) except that the first \( j \) keys revealed to \( \mathcal{A} \) become semifunctional of type 2. We let \( Game_{2,0,2} \) denote \( Game_1 \).
(v) \( Game_3 \): there is no difference with \( Game_{2,0,2} \) except that we generate a semifunctional ciphertext of a random message \( m' \in \mathbb{G}_T \) as the challenge ciphertext.

Lemma 9 (from \( Game_0 \) to \( Game_1 \)). For any PPT adversary \( \mathcal{A} \), there exists an adversary \( \mathcal{B} \) such that \( |Adv_{\mathcal{A}}^{Game_0}(\lambda) - Adv_{\mathcal{A}}^{Game_1}(\lambda)| \leq Adv_{\mathcal{A}}^{Game_0}(\lambda) \).

Proof. The adversary \( \mathcal{B} \) gets input
\[
(\mathbb{P}, T'),
\]
(77)
where \( T' \) is \( g' \) or \( g' \cdot g' \cdot g' \).

Setup. Pick \( k_j \) and output
\[
APK = \left( \mathbb{P}, e \left( \pi_L(B), \gamma_1 \right) \right),
\]
(78)

Key Queries. When \( \mathcal{A} \) queries the random oracle for \( H(GID) \), \( \mathcal{B} \) chooses \( T' \leftarrow Z_p^* \), sets \( H(GID) = [\gamma_1 \cdot (B^R) \cdot \gamma_2]_2 \), and stores this value. \( \mathcal{B} \) creates secret keys as follows:
\[
SK_{GID,1} = \left[ k_j + \gamma_1 \left( \pi_L(B^R) + \pi_R(B^R) \cdot \gamma_2 \right) \right]_2.
\]
(79)

Challenge. Upon receiving \( (M, \rho), m_0, \) and \( m_1 \), \( \mathcal{B} \) can compute the ciphertext by using \( T' \). We note that the ciphertext
is properly distributed except $C_{2,x}$, which take the following forms:

$$C_{2,x} = \left[ \begin{array}{c} 0 \\ \vdots \\ U_2^T \pi_L(B) \widetilde{s} \\ \vdots \\ U_L^T \pi_L(B) \widetilde{s} \end{array} \right] M^T_x + Y^T_i \pi_M(B) \widetilde{s},$$

and

$$\left[ \begin{array}{c} 0 \\ \vdots \\ \widehat{\rho} \pi_M(B) \widetilde{s} \\ \vdots \\ \widehat{\rho} \pi_M(B) \widetilde{s} \end{array} \right] M^T_x + Y^T_i \pi_R(B) \widetilde{s}.$$

are identically distributed. This follows readily from the fact that

(i) the space spanned by rows of $M$ whose corresponding attributes belong to corrupt authorities cannot include the vector $(1, 0, \ldots, 0)$; it reveals no information about $\widehat{\rho} \pi_M(B) \widetilde{s}$ and $\widehat{\rho} \pi_R(B) \widetilde{s}$;

(ii) rows of $M$ whose corresponding attributes belong to good authorities are masked by $\widehat{\rho} \pi_M(B) \widetilde{s}$ and $\widehat{\rho} \pi_R(B) \widetilde{s}$, respectively.

If $\overrightarrow{t} = \overrightarrow{g}$, $\mathcal{B}$ properly simulates $\text{Game}_{ij}$; if $\overrightarrow{t} = \overrightarrow{g} \cdot \overrightarrow{g} \cdot \overrightarrow{g}$, $\mathcal{B}$ properly simulates $\text{Game}_{i}$. Hence, $\mathcal{B}$ can determine the distribution of $\overrightarrow{t}$ by using adversary $\mathcal{A}$. \hfill $\Box$

**Lemma 10** (from $\text{Game}_{ij-1,2}$ to $\text{Game}_{i,iii}$). For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $|\text{Adv}_{\mathcal{A}}(\text{Game}_{ij-1,2}(\lambda)) - \text{Adv}_{\mathcal{B}}(\text{Game}_{i,iii}(\lambda))| \leq \text{Adv}_{\mathcal{B}^{RSI}}(\lambda)$.

**Proof.** The adversary $\mathcal{B}$ gets input

$$\left( pp, g, \overrightarrow{g}, \overrightarrow{h}, \overrightarrow{t} \right),$$

where $\overrightarrow{t}$ is $h$ or $\overrightarrow{h} \cdot \overrightarrow{h}$.

Setup. Pick $k_i$ and output

$$\text{APK} = \left( pp, e \left( \pi_L(B), \overrightarrow{t} \right) \right).$$

**Key Queries.** We let $\text{GID}_k$ denote the $k$th identity queried by $\mathcal{A}$.

(i) $k < j$: $\mathcal{B}$ chooses $\overrightarrow{r} \leftarrow Z_p^k$ and sets $H(\text{GID}_k) = \left( [\pi_L(B') R] + \pi_R(B') \overrightarrow{r} \right) \overrightarrow{r}.$

(ii) $k > j$: $\mathcal{B}$ chooses $\overrightarrow{r} \leftarrow Z_p^k$, and sets $H(\text{GID}_k) = \left( [\pi_L(B') R] \overrightarrow{r} \right) \overrightarrow{r}.$

(iii) $k = j$: $\mathcal{B}$ creates secret keys as follows:

$$\text{SK}_{\text{GID}_j} = g_k, H(GID)^{Y_i}. \quad (84)$$

Challenge. Upon receiving $(M, \rho), m_0, m_1, m_2$, $\mathcal{B}$ computes the ciphertext by using $\overrightarrow{g} \cdot \overrightarrow{g} \cdot \overrightarrow{g}$ as follows:

$$C' = m \cdot e(g_1, g_2)^{y_0},$$

$$C_0 = \left[ \pi_L(B) \overrightarrow{s} + \pi_M(B) \overrightarrow{s} + \pi_R(B) \overrightarrow{s} \right],$$

$$C_{1,x} = e(g_1, g_2)^{y_1} \cdot e(g_1, g_2)^{y_2} \cdot \left( (\pi_L(B) \overrightarrow{s} + \pi_R(B) \overrightarrow{s} + \pi_L(B) \overrightarrow{s}) \right),$$

$$C_{2,x} = \left[ \begin{array}{c} 0 \\ \vdots \\ \overrightarrow{U}_1 \pi_L(B) \overrightarrow{s} \\ \vdots \\ \overrightarrow{U}_L \pi_L(B) \overrightarrow{s} \end{array} \right] M^T_x + Y^T_i \pi_R(B) \overrightarrow{s},$$

$$\left[ \begin{array}{c} 0 \\ \vdots \\ \overrightarrow{U}_1 \pi_M(B) \overrightarrow{s} \\ \vdots \\ \overrightarrow{U}_L \pi_M(B) \overrightarrow{s} \end{array} \right] M^T_x + Y^T_i \pi_R(B) \overrightarrow{s},$$

$$\left[ \begin{array}{c} 0 \\ \vdots \\ \overrightarrow{U}_1 \pi_R(B) \overrightarrow{s} \\ \vdots \\ \overrightarrow{U}_L \pi_R(B) \overrightarrow{s} \end{array} \right] M^T_x + Y^T_i \pi_R(B) \overrightarrow{s}$$

where $\overrightarrow{s}, \overrightarrow{s} \sim Z_p^k, \overrightarrow{U}_2, \ldots, \overrightarrow{U}_L \sim Z_p^{m \times k}$. The ciphertext is properly distributed except that the second components of
$C_{2x}$ are shares of 0. We must argue that there is no difference in $\mathcal{A}$'s view.

By parameter hiding, it suffices to show that

$$\begin{bmatrix}
0 & \cdots & \tilde{\mathbf{M}}_x + \tilde{\mathbf{u}}_M(B) \tilde{s} \\
\tilde{s} \tilde{\pi}_M(B) \tilde{s} & \cdots & \tilde{\mathbf{M}}_x + \tilde{\mathbf{u}}_M(B) \tilde{s}
\end{bmatrix} \iff$$

are identically distributed. This follows readily from the fact that

(i) for $k > j$ and $k < j$, $H(\text{GID}_i)$ have nothing to do with $\pi_M(B) \tilde{s}$;

(ii) the space spanned by rows of $\mathbf{M}$ whose corresponding attributes belong to corrupted authorities or queried with $\text{GID}_j$ cannot include the vector $(1,0,\ldots,0)$; it reveals no information about $\tilde{s} \pi_M(B) \tilde{s}$;

(iii) the remaining rows of $\mathbf{M}$ are masked by $\tilde{\mathbf{u}}_M(B) \tilde{s}$.

If $\overrightarrow{t} = \overrightarrow{h}$, $\mathcal{B}$ properly simulates $\text{Game}_{2,1;2}$; if $\overrightarrow{t} = \overrightarrow{v} \neq \overrightarrow{h}$, $\mathcal{B}$ properly simulates $\text{Game}_{2,1;1}$. Hence, $\mathcal{B}$ can determine the distribution of $\overrightarrow{t}$ by using adversary $\mathcal{A}$.

Lemma 11 (from $\text{Game}_{2,1;1}$ to $\text{Game}_{2,1;2}$). For any PPT adversary $\mathcal{A}$, there exists an adversary $\mathcal{B}$ such that $|\text{Adv}_{\mathcal{A}}^{\text{Game}_{2,1;1}}(\lambda) - \text{Adv}_{\mathcal{A}}^{\text{Game}_{2,1;2}}(\lambda)| < \text{Adv}_{\mathcal{A}}^{R_{\text{Game}}}(\lambda)$.

Proof. The adversary $\mathcal{B}$ gets input

$$\begin{bmatrix}
\mathbf{p}, \overrightarrow{g}, \overrightarrow{\tilde{g}}, \overrightarrow{t}
\end{bmatrix},$$

where $\overrightarrow{t} = \overrightarrow{h} \cdot \overrightarrow{h}$ or $\overrightarrow{t} = \overrightarrow{h} \cdot \overrightarrow{v}$.

Setup. Pick $\overrightarrow{k}_2$ and output

$$\begin{bmatrix}
\mathbf{p}, \mathbf{e}(\pi_L(B)_{1,1}, \overrightarrow{k}_2)
\end{bmatrix}.$$  

Key Queries. We let $\text{GID}_k$ denote the $k$th identity queried by $\mathcal{A}$.

(i) $k < j$: $\mathcal{B}$ chooses $\overrightarrow{t} \leftarrow Z_p^k$ and sets $H(\text{GID}_k) = \pi_L(B^*) R + \pi_0(B^*) \overrightarrow{r}$.

(ii) $k > j$: $\mathcal{B}$ chooses $\overrightarrow{t} \leftarrow Z_p^k$ and sets $H(\text{GID}_k) = \pi_L(B^*) \overrightarrow{r}.$

(iii) $k = j$: $H(\text{GID}) = \overrightarrow{t}_0$, where $\overrightarrow{t}_0$ is the first element in $\overrightarrow{t}$.

$\mathcal{B}$ creates secret keys as follows:

$$K_{\text{GID}_j} = \overrightarrow{g}_2 H(\text{GID}) Y_j.$$  

Challenge. Upon receiving $(\mathbf{M}, p)$, $m_0$, and $m_1$, $\mathcal{B}$ computes the ciphertext as follows:

$$\begin{align*}
C' &= m \cdot e(g_1, g_2)^{h}, \\
C_0 &= \pi_L(B) \overrightarrow{s} + \pi_M(B) \overrightarrow{s} + \pi_R(B) \overrightarrow{s}, \\
C_{1x} &= e(g_1, g_2)^{l_x} e(g_1, g_2)^{l_x} \cdot (\pi_L(B)^T + \pi_M(B)^T + \pi_R(B)^T), \\
C_{2x} &= \left[\begin{array}{c}
0 \\
\overrightarrow{s} \\
\cdots \\
\overrightarrow{s}
\end{array}\right] + \left[\begin{array}{c}
\mathbf{U}_1^T \pi_L(B) \overrightarrow{s} + \mathbf{U}_2^T \pi_M(B) \\
\cdots + \mathbf{U}_n^T \pi_R(B) \overrightarrow{s}
\end{array}\right] \\
&\cdot \left[\begin{array}{c}
\mathbf{M}_x \\
\cdots \\
\mathbf{M}_x
\end{array}\right].
\end{align*}$$

Output

$$\begin{bmatrix}
\mathbf{g}_1, \mathbf{g}_2, \mathbf{e}(\pi_L(B)_{1,1}, \overrightarrow{k}_2), H
\end{bmatrix}.$$  

$\mathcal{B}$ also samples $\overrightarrow{k}_j \leftarrow Z_p^k$, $\overrightarrow{Y}_j \leftarrow Z_p^{k \times k}$ for each attribute and sets

APK

$$= \left[\begin{array}{c}
\mathbf{g}_1, \mathbf{g}_2, \mathbf{e}(\pi_L(B)_{1,1}, \overrightarrow{k}_2), H
\end{array}\right].$$

Key Queries. In both games, the secret keys take the following form:

$$\begin{bmatrix}
\overrightarrow{k}_i + \overrightarrow{Y}_i \pi_L(B^*) \overrightarrow{r} + \overrightarrow{R}_i \pi_R(B^*) \overrightarrow{r}
\end{bmatrix}.$$  

which means they leak no information whatsoever about $\pi_M(B)$. 
Challenge. Upon receiving \((\mathbf{M}, \rho), m_0, m_1, \mathcal{B}\) computes the semifunctional ciphertext of \(m_0\) or \(m_1\). Observe that

\[
C_{1x} = e(g_1, g_2)^{\lambda} \cdot e(g_1, g_2)^{k_{\tau} \cdot (\pi_x(B)\tau + \pi_x(B)\tau)}
\]

and the quantity \(e(g_1, g_2)^{k_{\tau} \cdot \pi_x(B)\tau}\) is uniformly distributed over \(G_T\). This implies the challenge ciphertext is identically distributed to a semifunctional encryption of a random message in \(G_T\), as in Game3.

6.3. Performance Discussions. In this section, we provided analysis regarding the space and computation cost of the proposed scheme by comparing it with existing decentralized ABE schemes.

As shown in Table 1, [11] is built on composite-order groups. We recall that composite-order elements are 12 times larger than prime-order ones and pairing is 250 times slower in composite-order groups than in prime-order ones [12]. Though [29] is efficient, the scheme can only prove static security under a \(q\)-type assumption. Both [13] and ours are based on prime-order groups; the secret key size and the ciphertext size in ours are reduced by about 40% compared with [13] under the same assumption (DLIN). We will see further improvement if we instantiate our construction under the SXDH assumption. In addition, the ciphertexts in our setting can be decrypted with a constant number of pairings at the cost of increasing some exponentiations. We believe that this is a good deal since pairing is about 5 times slower than group exponentiation according to [29]. The advantage of decryption performance in our scheme will become more and more obvious as the number of attributes used for decryption increases.

7. Conclusions

In this paper, we presented a fully secure decentralized CP-ABE scheme under the standard \(k\)-Lin assumptions in prime-order groups. To prove the security of our scheme, we extended the basis of dual system groups from \(2k \times 2k\) matrix to \(3k \times 3k\) matrix and realized some assumptions to mimic the effect of the subgroup decision assumption in composite-order groups. Our scheme achieved lower computational cost thanks to decryption which only needs constant number of pairing operations. We discussed the performance of our scheme from the theoretical points of view. Compared with other existing decentralized CP-ABE schemes, our scheme is more compact to implement and can provide better efficiency in terms of the communication and computation cost.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (nos. 61502529, 61472142, and 61602512) and the Science and Technology Commission of Shanghai Municipality (no. 14YFI404200).

References


