A CP-ABE Scheme Supporting Arithmetic Span Programs

Chao Ma, Haiying Gao, and Duo Wei

PLA SSF Information Engineering University, Zhengzhou, Henan, China

Correspondence should be addressed to Chao Ma; mac__88@163.com

Received 18 September 2019; Revised 25 January 2020; Accepted 3 February 2020; Published 16 March 2020

Academic Editor: Bruce M. Kapron

Copyright © 2020 Chao Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Attribute-based encryption achieves fine-grained access control, especially in a cloud computing environment. In a ciphertext-policy attribute-based encryption (CP-ABE) scheme, the ciphertexts are associated with the access policies, while the secret keys are determined by the attributes. In recent years, people have tried to find more effective access structures to improve the efficiency of encryption systems. This paper presents a ciphertext-policy attribute-based encryption scheme that supports arithmetic span programs. On the composite-order bilinear group, the security of the scheme is proven by experimental sequence based on the combination of composite-order bilinear entropy expansion lemma and subgroup decision (SD) assumption. And, it is an adaptively secure scheme with constant-size public parameters.

1. Introduction

In the cloud computing environment, the traditional public key encryption system cannot meet the realistic needs due to the feature that it only achieves one-to-one encrypted data sharing. In 2006, Goyal et al. [1] proposed attribute-based encryption (ABE), which can achieve one-to-many encryption, making the sharing of encrypted data more convenient. Besides, the encrypter does not need to know the specific identifying information of the visitors but only needs to use the access structure to complete the access control of the user’s identity on the fine-grained level, which provides a new idea for data sharing. ABE is divided into two types based on ciphertexts or keys being marked as attributes. For example, in a CP-ABE scheme, keys are marked as attributes and the ciphertexts are linked with access policies. Conversely, the key-policy ABE (KP-ABE) means that keys are linked with access policies and the ciphertexts are marked as a series of attributes.

In 2006, Goyal et al. [1] came up with a KP-ABE scheme that supports an access tree. The size of the public parameters is linearly related to the size of the attributes, that is, the size is not constant. In 2008, Katz et al. [2] put forward the first KP-ABE scheme based on the inner product on the composite-order bilinear group. It is a selectively secure scheme, and the length of the ciphertext increases linearly with the vector’s dimension. In 2010, Herranz et al. [3] proposed a CP-ABE scheme with a constant-size ciphertext, but it only supports the threshold access control. In 2011, based on dual pairing vector space, Okamoto and Takshima [4] presented a zero-inner product encryption scheme and a nonzero inner product encryption scheme which are fully secure under the standard model, in which the ciphertext’s length or the key’s length can reach a constant. In 2011, Attrapadung et al. [5] first proposed a KP-ABE scheme that supports the nonmonotonic access control. The scheme has a constant-size ciphertext, but it can only be proved under the selective model. In 2013, Chen et al. [6] gave a general construction method from inner product encryption to ABE and presented an ABE scheme supporting threshold access control based on inner product encryption. This scheme achieves adaptive security with constant-size ciphertext. In 2014, Wee [7] first proposed an ABE scheme supporting the arithmetic span programs [8], but did not give a specific scheme (just a framework). In 2015, Attrapadung et al. [9] proposed a general conversion between the ABE scheme supporting the arithmetic span programs and the KP-ABE scheme when we do not limit the size of the span programs, but the size of the attributes is limited. This scheme achieves adaptive security with a constant-size ciphertext, but the
length of the public parameters is still not constant. In 2017, Chen et al. [10] first proposed a KP-ABE scheme supporting arithmetic span programs via bilinear entropy expansion, and the scheme is adaptive security with constant-size parameters. In particular, Table 1 illustrates the development of ABE about the access structure. Besides, the existing ABE scheme can be converted into a scheme supporting the arithmetic span program. Compared with the ABE scheme, the computational complexity and parameter size of the scheme supporting the arithmetic span program are relatively small. Therefore, based on the fact that the composite-order bilinear group has fewer algorithm components and the algorithm represents simple and clear advantages, we naturally think of the following question about ABE:

“Can we design a CP-ABE scheme that supports arithmetic span program on a bilinear group?”

1.1. Our Contribution. Although CP-ABE and KP-ABE have many similarities in structure, even a dual relationship, the application scenarios are very different. In the CP-ABE scheme, because the policy is embedded in ciphertext, the data owner can set policies to determine which properties can access the ciphertext. That is, encrypted access control for this data can be refined to the attribute level. The application scenario of CP-ABE is usually data encryption storage and fine-grained sharing on the public cloud, while the application scenario of KP-ABE is more inclined to pay video websites, log encryption management, and so on. Inspired by [10], we consider designing an adaptively secure CP-ABE scheme. There are some schemes supporting arithmetic span programs [10, 11], where [10, 11] are KP-ABE schemes. However, considering that the composite-order group has fewer algorithm components and the algorithm represents simple and clear advantages, it is meaningful to construct a CP-ABE scheme on composite-order groups. Specifically, to reduce the parameter size, we first give the composite-order bilinear entropy expansion lemma, which contains the specific form of public parameters, ciphertext, and the key. In the setup, we use some random numbers as the specific form of public parameters, ciphertext, and the key. In the KeyGen, we combine the master secret key and use the master secret key to calculate the master public key. In the Enc, we subtly embed the strategy into certain components of the ciphertext in combination with the public parameters and the bilinear entropy extension vector. In the KeyGen, we combine the attribute vector, the public parameter, and the bilinear entropy extension vector to generate the secret key. In the Dec, the arithmetic span program is used as a standard for decryption and the user can decrypt normally. Finally, based on SD assumption and composite-order bilinear entropy expansion lemma, the scheme is proved to have adaptive security.

2. Preliminaries

Notation. We let \( \mathbb{Z}_p \) denote a ring of algebraic integers modulo a prime number \( p \) and \( \mathbb{Z}_p^n \) denote an m-dimension vector in \( \mathbb{Z}_p \). \( G_N \) and \( e \) represent a group of order \( N \) and a bilinear map, respectively. We denote \( [n] \) as the set \( \{1, 2, \ldots, n\} \) and n-dimensional vector as the bold letter \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \).

2.1. Bilinear Maps

Definition 1 (see [12, 13] bilinear maps). Let \( G_N, H_N, G_T \) be bilinear groups of order \( N = p_1p_2p_3 \), where \( p_1, p_2, \) and \( p_3 \) are primes. Let \( g \) be the generator of \( G_N \) and \( g_1, g_2, \) and \( g_3 \) the generators of \( g_{p_1}, g_{p_2}, \) and \( g_{p_3} \), respectively. Let \( h \) be the generator of \( H_N \), and \( h_1, h_2, \) and \( h_3 \) are the generators of \( h_{p_1}, h_{p_2}, \) and \( h_{p_3} \), respectively.

\[ e: (G_N, H_N) \rightarrow G_T \] is a bilinear map, if it satisfies the following three properties:

1. Bilinearity: \( e(g_0^a, h_0^b) = e(g_0, h_0)^{ab} \) for all \( a, b \in \mathbb{Z}_p \), \( g_0 \in G_N \), \( h_0 \in H_N \).
2. Nondegeneracy: there exists \( g_0 \in G_N \), \( h_0 \in H_N \), such that the order of \( e(g_0, h_0) \) is \( N \).
3. Computability: for all \( g_0 \in G_N \), \( h_0 \in H_N \), there is an efficient algorithm to compute \( e(g_0, h_0) \).

Also, the composite-order bilinear map satisfies the orthogonality \( e(g_i, h_j) = 1 \), for all \( i, j \in \{1, 2, 3\} \), \( i \neq j \).

2.2. Arithmetic Span Program

Definition 2 (arithmetic span program [8]). An arithmetic span program \((u, \rho)\) is a map \( \rho: [l] \rightarrow [n] \), and a collection of row vectors \( v = \{y_j, z_j\}: j \in [l], y_j, z_j \in Z_p^l \), for \( \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n, \mathbf{x} \in Z_p^n \), satisfies \((u, \rho)\) iff there exists constants \( \omega_1, \ldots, \omega_l \in \mathbb{Z}_p \), such that

\[ \sum_{j=1}^l \omega_j (y_j + x_{p(j)}z_j) = 1, \tag{1} \]

where \( 1 = (1, 0, \ldots, 0) \in Z_p^l \).

Like in paper [9], we limit \( \rho \) to be an identity map and \( l = n \).

2.3. Computational Assumptions

Assumption 1 (SD \([\mathbb{Z}_p]^N \rightarrow \mathbb{Z}_p^{p_1p_2p_3} \) [12, 13]). We define the subgroup decision assumption (denoted by SD \([\mathbb{Z}_p]^N \rightarrow \mathbb{Z}_p^{p_1p_2p_3} \) ) holds if for all probability polynomial time (PPT) adversaries \( A \), and the following advantage function is negligible in \( \lambda \):
① the security of the scheme against selectively chosen plaintext attacks can be proven in the standard model; ② this paper did not give a specific scheme (just a framework).

\[
\text{Adv}_{\mathcal{A}}^{\text{DDH}_{P2}^{\text{cov}}} (\lambda) = |\Pr[\mathcal{A}(G, D, T_0)] - \Pr[\mathcal{A}(G, D, T_1)]|,
\]

where

\[
D = (h_1, h_2, h_3, g_1, g_2, g_3),
T_0 = (g_2^x, g_2^y, g_2^z),
T_1 = (g_2^x, g_2^y, g_2^z),
\]

Assumption 2. (DDH_{P2}^{cov}) The P2-DDH assumption (denoted by DDH_{P2}^{cov}), holds if for all probability polynomial time (PPT) adversaries \(\mathcal{A}\), and the following advantage function is negligible in \(\lambda\):

\[
\text{Adv}_{\mathcal{A}}^{\text{DDH}_{P2}^{\text{cov}}} (\lambda) = |\Pr[\mathcal{A}(G, D, T_0)] - \Pr[\mathcal{A}(G, D, T_1)]|,
\]

2.4. Bilinear Entropy Expansion Lemma. For an adversary \(A\), the advantage of distinguishing the following two distributions in any polynomial time is negligible:

\[
\begin{align*}
\text{Adv}_{\mathcal{A}}^{\text{DDH}_{P2}^{\text{cov}}} & (\lambda) = |\Pr[\mathcal{A}(G, D, T_0)] - \Pr[\mathcal{A}(G, D, T_1)]|,
\end{align*}
\]
See Appendix for details about the proof of this lemma.

3. CP-ABE Supporting Arithmetic Span Programs

3.1. Formal Definition of the CP-ABE Scheme Supporting Arithmetic Span Program

Setup \((1^λ, p^*)\): input security parameters \((1^λ, p^*)\) and output the master public key \(\text{mpk}\) and the master secret key \(\text{msk}\).

\[
\text{Enc}(\text{mpk}, \upsilon, m) : \text{input access structure } \upsilon = \{(y_j, z_j) : j \in [n], y_j, z_j \in \mathbb{Z}_p^l\} \text{ and plaintext } m \text{ and output ciphertext } \text{ct}_\upsilon.
\]

KeyGen(\(\text{mpk}, \text{msk}, x\)): input the vector \(x \in \mathbb{Z}_p^n\) and output the secret key \(\text{sk}_x\).

\[
\text{Dec}(\text{mpk}, \text{sk}_x, \text{ct}_\upsilon) : \text{input } \text{sk}_x \text{ and } \text{ct}_\upsilon \text{ and output } m \text{ if } x, \upsilon \text{ satisfies } \sum_{j=1}^n \omega_j(y_j + x_jz_j) = 1.
\]

3.2. Adaptively Security Model for CP-ABE Schemes Supporting Arithmetic Span Programs. We present an adaptive security model of the CP-ABE scheme that supports the arithmetic span program through the games about the challenger \(B\) and adversary \(A\).

Setup 1: challenger \(B\) runs the initialization algorithm and sends \(\text{mpk}\) to adversary \(A\).

Stage 1: adversary \(A\) chooses \(x\) to perform multiple secret key queries. Challenger \(B\) runs the KeyGen and sends the secret key to the adversary \(A\).

Challenge: adversary \(A\) sends two equal-length plaintexts \((m_0, m_1)\) and the challenge access structure \(\upsilon^* = \{(y_j, z_j) : j \in [n], y_j, z_j \in \mathbb{Z}_p^l\}\) to challenger \(B\) (any query vector \(x\) and the challenge access structure \(\upsilon^* = \{(y_j, z_j) : j \in [n], y_j, z_j \in \mathbb{Z}_p^l\}\) do not satisfy \(\sum_{j=1}^l \omega_j(y_j + x_jz_j) = 1\). Challenger \(B\) chooses \(b \in \{0, 1\}\) randomly and computes \(\text{ct}_{\upsilon^*} = \text{Enc}(\text{mpk}, \upsilon^*, \text{mpk})\). Then, Challenger \(B\) sends the challenge ciphertext \(\text{ct}_{\upsilon^*}\) to the adversary \(A\).

Stage 2: same as Stage 1.

Guess: adversary \(A\) outputs the guess \(b'\) about \(b\).

We say adversary \(A\) wins this game iff \(b' = b\), and the advantage of adversary \(A\) is \(\text{Adv}_A(\lambda) = |\text{Pr}(b' = b) - 1/2|\).

The encryption scheme is adaptively secure if the advantages of winning the above games are negligible, for all PPT adversaries.

3.3. Our Construction

Setup \((1^λ, 1^n)\): input the number of security parameters \(\lambda\) and attributes \(n\) and select \(G = (N = p_1p_2p_3, G_N, H_N, e) \leftarrow G(1^λ)\). Pick random generators \(g_1, h_1\), and \(h_{123}\) of \(G_{p_1}, H_{p_1}\), and \(H_N\), respectively. Sample \(w, w_0, w_1, w', w_0, w'_1, u_0, u_0, u' \rightarrow \mathbb{Z}_N\) and output the master public key

\[
\text{mpk} = (h_{112}, h_1, g_1, g_1, g_1, g_1, g_1, e(g_1, h_{123})^\alpha),
\]

and the master secret key

\[
\text{msk} = (h_{112}, h_1, a, w, w_0, w_1, w', w_0, w'_1, u_0).
\]

\[
\text{Enc}(\text{mpk}, \upsilon, m) : \text{input the access structure } \upsilon = \{(y_j, z_j) : j \in [l], y_j, z_j \in \mathbb{Z}_p^l\} \text{ and the message } m \in \{0, 1\}^l. \text{ Select } s, s_j \rightarrow \mathbb{Z}_N, u \rightarrow \mathbb{Z}_N^{-1} \text{ for all } j \in [n]. \text{ Compute and output }
\]

\[
\text{ct}_\upsilon = \begin{cases} C_0 = g_1^\upsilon, & \text{if } \upsilon \text{ satisfies } \sum_{j=1}^l \omega_j(y_j + u_jz_j) = 1, \\ C_{0,j} = g_1^{\omega_j}, & \text{for all } j \in [n], \\ C_{1,j} = g_1^{s_j}, & \text{for all } j \in [n], \\ C = e(g_1, h_{123})^\upsilon \cdot m \\ \end{cases}
\]

\[
\text{KeyGen}(\text{mpk}, \text{msk}, x) : \text{input the master secret key } \text{msk} \text{ and vector } x = (x_1, x_2, \ldots, x_n) \in \mathbb{Z}_p^n. \text{ Select } r, r_j, r_j' \rightarrow \mathbb{Z}_N \text{ for all } j \in [n] \text{ and output }
\]

\[
\text{sk}_x = \begin{cases} K_0 = h_{123}^{u_0}, & K_1 = h_1^{u_0}, \\ K_{1,j} = h_1^{s_j}, & \text{for all } j \in [n], \\ K_{2,j} = h_1^{r_j}, & \text{for all } j \in [n], \\ \end{cases}
\]

\[
\text{Dec}(\text{mpk}, \text{sk}_x, \text{ct}_\upsilon) : \text{input secret key } \text{sk}_x \text{ and ciphertext } \text{ct}_{\upsilon^*}. \text{ If } x, \upsilon \text{ satisfies } \sum_{j=1}^n \omega_j(y_j + x_jz_j) = 1, \text{ then compute } m = C \cdot C' / e(C_0, K_0)
\]
where
\[
C' = \prod_{j=1}^{n} \left( e\left(C_{0,j} \cdot (C_{0,j})^{x_j}, K_1 \right) \cdot e\left(C_{1,j}, K_{1,j} \right)^{-1} \cdot e\left(C_{2,j}, K_{2,j} \right) \cdot e\left((C_{2,j})^{x_j}, K_{2,j}' \right) \right)^{\omega_j}.
\]

(12)

3.4. Correctness. For all \((x, \nu)\) satisfies \(\sum_{j=1}^{l} \omega_j (y_j + x_j z_j) = 1\), we compute
\[
e\left(C_{0,j} \cdot (C_{0,j})^{x_j}, K_1 \right) \cdot e\left(C_{1,j}, K_{1,j} \right)^{-1} \cdot e\left(C_{2,j}, K_{2,j} \right) \cdot e\left((C_{2,j})^{x_j}, K_{2,j}' \right) = m.
\]
3.5. Security. The proof of the security relies on a series of games that cannot be distinguished. We first define the ciphertext and secret key distributions that are needed in the process of the proof.

3.5.1. Ciphertext Distributions

Standard ciphertext: generated by the encryption algorithm:

\[
\begin{align*}
\text{ct}_w &= \left\{ \begin{array}{cl}
C_0 &= g_1^i \\
C_{0,j} &= g_1^{i_j} \\
C_{1,j} &= g_1^i, C_{2,j} &= g_1^j (w_{0}+jw_1) \\
C &= e(g_1, h_{123})^a \cdot m
\end{array} \right\}.
\end{align*}
\]

Entropy expansion ciphertext: the difference between it and standard ciphertext is given as follows: \( w \rightarrow v \mod p_2, w' \rightarrow v' \mod p_2, w_0 + jw_1 \rightarrow u_j \mod p_2, w'_0 + jw'_1 \rightarrow u'_j \mod p_2 (\forall j \in [n]), \)

\[
\begin{align*}
\text{ct}_u &= \left\{ \begin{array}{cl}
C_0 &= g_1^i g_2^j \\
C_{0,j} &= g_1^{i_j} g_2^{j'_j} \\
C_{1,j} &= g_1^i, C_{2,j} &= g_1^j (w_{0}+jw_1) \\
C &= e(g_1^i g_2^j, h_{123})^a \cdot m
\end{array} \right\}.
\end{align*}
\]

3.5.2. Secret Key Distributions

Standard secret key: it is generated by the secret key generation algorithm:

\[
\text{sk}_x = \left\{ \begin{array}{cl}
K_0 &= h_{123}^a h_{1}^{\text{sk}_x} \cdot K_1 = h_{1}' \\
K_{1,j} &= h_1^{i_j} (w_{0}+jw_1) + x_j w_1 \\
K_{2,j} &= h_{1}'^{i'_j} \cdot j \in [n]
\end{array} \right\}.
\]

Entropy expansion secret key: compared to the standard secret key, we make a copy of \( \{ h_{1}', h_{1}'^{i_j} \}_{j \in [n]} \) in \( H_{F_{1}} \):

\[
\begin{align*}
\text{sk}_x &= \left\{ \begin{array}{cl}
K_0 &= h_{123}^a h_{1}^{\text{sk}_x} \cdot K_1 = h_{1}' h_{2}' \\
K_{1,j} &= h_1^{i_j} (w_{0}+jw_1) + x_j w_1 \cdot \left( w_{0}+jw_1 \right) \\
K_{2,j} &= h_{1}'^{i'_j} \cdot h_{2}'^{i'_j} \cdot j \in [n]
\end{array} \right\}.
\end{align*}
\]
Pseudostandard secret key: compared to the entropy expansion secret key, we make a copy of 
\[ \{h'_1, h'_2 \}_{j \in [n]} \] in \( H_{p_i} \):

\[
\text{sk}_x = \begin{cases} 
    K_0 &= h'_1 h'_2 h'_3 \\
    K_{1,j} &= h'_1 (w + x, w') + r_j (w + jw, x), h'_2 (v + x, v') + r_j (v + jv, x'), h'_3 (v + x, v') + r_j (v + jv, x') \\
    K_{2,j} &= h'_1 h'_2 h'_3 \\
\end{cases}_{j \in [n]} \].
\]

Pseudo-semi-functional secret key: compared to the pseudostandard secret key, we sample \( \tilde{\alpha} \rightarrow H_{p_i} \):

\[
\text{sk}_x = \begin{cases} 
    K_0 &= h'_1 h'_2 h'_3 \\
    K_{1,j} &= h'_1 (w + x, w') + r_j (w + jw, x), h'_2 (v + x, v') + r_j (v + jv, x'), h'_3 (v + x, v') + r_j (v + jv, x') \\
    K_{2,j} &= h'_1 h'_2 h'_3 \\
\end{cases}_{j \in [n]} \].
\]

Semifunctional secret key: compared to the pseudo-semi-functional secret key, we remove 
\[ \{h'_j, h'_2 \}_{j \in [n]} \] :

\[
\text{sk}_x = \begin{cases} 
    K_0 &= h'_1 h'_2 h'_3 \\
    K_{1,j} &= h'_1 (w + x, w') + r_j (w + jw, x), h'_2 (v + x, v') + r_j (v + jv, x'), h'_3 (v + x, v') + r_j (v + jv, x') \\
    K_{2,j} &= h'_1 h'_2 h'_3 \\
\end{cases}_{j \in [n]} \].
\]

3.5.3. Games. Assume that an adversary \( A \) makes at most \( Q \) secret key queries. Let the advantage of \( A \) in Game_{xx} be denoted by \( \text{Adv}_{xx}(\lambda) \). In the following, we describe in detail the specific details of the games, and the comparison of Game_{xx} is given in Table 2.

Game_{0}: the challenge ciphertext and secret keys are generated by Enc and KeyGen, respectively.

Game_{i}: compared to Game_{0}, all challenge ciphertext and secret keys are entropy expansion.

Game_{i,1}: compared to Game_{0}, the first \( i - 1 \) secret keys are semifunctional and the last \( Q - i + 1 \) are entropy expansion.

Game_{i,2}: compared to Game_{i,1}, modify the \( i' \) key to the pseudo-semifunctional key.

Game_{i,3}: compared to Game_{i,2}, modify the \( i' \) key to the semifunctional key.

Game_{final}: challenge ciphertext is the entropy expansion ciphertext about a random message, while the secret keys are semifunctional.

**Lemma 1** (Game_{0} \( \approx \) Game_{0}). There exists a challenger \( B_0 \) who can distinguish the left and right distributions in the bilinear entropy expansion lemma with a non-negligible advantage if \( |\text{Adv}_{0}(\lambda) - \text{Adv}_{0}(\lambda)| > \epsilon \), that is, \( \text{Time}(B_0) \approx \text{Time}(A) \).
### Proof

Challenger $B_0$ obtains the following distribution:

$$
\begin{align*}
\text{aux: } & \begin{cases} 
& g_1, g_1^{w_i}, g_1^{w_1}, g_1^{w_2}, g_1^{w_3}, g_1^{w_4}, g_1^{w_5}, g_1^{w_6} \\
& \end{cases} \\
\text{ct: } & \begin{cases} 
& C_0 = C_0^{\ast}_{0,j}, C_0^{\ast}_{0,j} = C_0^{\ast}_{0,j} \\
& C_{0,j} = C_{0,j}^{\ast}_{0,j}, C_{0,j}^{\ast}_{0,j} = C_{0,j}^{\ast}_{0,j} \\
& C_{1,j} = C_{1,j}^{\ast}_{0,j}, C_{1,j}^{\ast}_{0,j} = C_{1,j}^{\ast}_{0,j} \\
& C = e(C_0^{\ast}_{0,j}, \alpha_{\text{sk}, j}) \\
& \end{cases} \\
\text{sk: } & \begin{cases} 
& K_0 = h_{i, j}^{\ast}, K_1 = K_1^{\ast} \\
& K_{1,j} = K_{1,j}^{\ast}, K_{1,j}^{\ast} = K_{1,j}^{\ast} \\
& K_{2,j} = K_{2,j}^{\ast}, K_{2,j}^{\ast} = K_{2,j}^{\ast} \\
& \end{cases}
\end{align*}
$$

(21)

$B_0$ needs to distinguish whether it is left distribution or right in the bilinear entropy expansion lemma.

Setup: pick a random generator $h_{123}$ of $H_N$. Sample $\alpha \sim \mathbb{R}Z_N$ and output

$$
\text{mpk} = (\text{aux}, e(g_1, h_{123}^\alpha)).
$$

(22)

Stage 1: adversary $A$ queries the secret key corresponding to the vector $x^f = (x_1^f, x_2^f, \ldots, x_n^f)$.

$$
\begin{align*}
\text{ct}_{x^f}: & \begin{cases} 
& C_0 = C_0^{\ast}_{0,j}, C_0^{\ast}_{0,j} = C_0^{\ast}_{0,j} \\
& C_{0,j} = C_{0,j}^{\ast}_{0,j}, C_{0,j}^{\ast}_{0,j} = C_{0,j}^{\ast}_{0,j} \\
& C_{1,j} = C_{1,j}^{\ast}_{0,j}, C_{1,j}^{\ast}_{0,j} = C_{1,j}^{\ast}_{0,j} \\
& C = e(C_0^{\ast}_{0,j}, \alpha_{\text{sk}, j}) \\
& \end{cases}
\end{align*}
$$

(24)

Stage 2: same as Stage 1.

Guess: adversary $A$ outputs the guess $b'$ about $b$.

Note: the output is the standard secret key and the standard challenge ciphertext if $B_0$ obtains the left distribution. Conversely, the output is the entropy expansion secret key and the entropy expansion challenge ciphertext if $B_0$ obtains the right distribution. Challenger $B_0$ also distinguishes the left and right distributions of the entropy expansion lemma with a non-negligible advantage if $|\text{Adv}_{\text{SD}}(\lambda) - \text{Adv}_{\text{sk}, j}(\lambda)| \geq \epsilon$. Game$_0$ and Game$_0$ cannot be distinguished due to the indistinguishability of the left and right distributions.

### Lemma 3

(Game$_0$ $\equiv$ Game$_1$). We know it in Table 2 easily.

| Games for proving the adaptive security of our scheme. | SK |
|------------------------------------------------------|-----|
| $0$ | $0'$ | Standard entropy expansion |
| $i$ | Semifunctional |
| $i, 1$ | Pseudostandard |
| $i, 2$ | Pseudo-semifunctional |
| $i, 3$ | Semifunctional |
| Final | Random message |

Challenger $B_0$ simulates the secret key generation algorithm and picks $\bar{r} \sim \mathbb{R}Z_N$ for all $j \in [n]$. Output $\text{sk}_{x, j} = \left\{ \begin{array}{ll} 
K_0 = h_{123}^{\epsilon}, K_1 = K_1^{\epsilon} \\
K_{1,j} = K_{1,j}^{\epsilon}, K_{1,j}^{\epsilon} = K_{1,j}^{\epsilon} \\
K_{2,j} = K_{2,j}^{\epsilon}, K_{2,j}^{\epsilon} = K_{2,j}^{\epsilon} \\
\end{array} \right\}_{j \in [n]}$. (23)

Challenge: adversary $A$ sends two equal-length plaintexts $(m_0, m_1)$ and the challenge access structure $v' = \{(y_j, z_j) : j \in [n], y_j, z_j \in Z_p^n\}$ to challenger $B_0$ (any query vector $x'$ in Phase 1 and the challenge access structure $v' = \{(y_j, z_j) : j \in [n], y_j, z_j \in Z_p^n\}$ do not satisfy $\sum_{j=1}^n \omega_j = x' \cdot z_j = \mathbf{1}$). Challenger $B_0$ picks $b \in \{0, 1\}$ and $u = \mathbb{R}Z_N^{-1}$. and outputs the challenge ciphertext:

$$
\begin{align*}
\text{ct}_{x^f}: & \begin{cases} 
& C_0 = C_0^{\ast}_{0,j}, C_0^{\ast}_{0,j} = C_0^{\ast}_{0,j} \\
& C_{0,j} = C_{0,j}^{\ast}_{0,j}, C_{0,j}^{\ast}_{0,j} = C_{0,j}^{\ast}_{0,j} \\
& C_{1,j} = C_{1,j}^{\ast}_{0,j}, C_{1,j}^{\ast}_{0,j} = C_{1,j}^{\ast}_{0,j} \\
& C = e(C_0^{\ast}_{0,j}, \alpha_{\text{sk}, j}) \\
& \end{cases}
\end{align*}
$$

(24)

Lemma 3 (Game$_0$ $\equiv$ Game$_1$). There exists a tester $B_1$ who can solve $\text{SD}_{\mathbb{R}^{P_1 \rightarrow P_2}}^{\mathbb{R}^{P_1 \rightarrow P_2}}$ with a non-negligible advantage if $|\text{Adv}_{\lambda}(\lambda) - \text{Adv}_{\text{sk}, j}(\lambda)| \geq \epsilon$, that is, $\text{Time}(B_1) \approx \text{Time}(A)$.

Proof. Firstly, from the SD$_{\mathbb{R}^{P_1 \rightarrow P_2}}^{\mathbb{R}^{P_1 \rightarrow P_2}}$ assumption, sample $\bar{r}, \bar{r}_j, \bar{r}_j' \sim \mathbb{R}Z_N^{123}$ for all $j \in [n]$ and we have $\left\{ \bar{h}_{1,j}^{\epsilon}, \bar{h}_{1,j}^{\epsilon}, \bar{h}_{1,j}^{\epsilon}, \bar{h}_{1,j}^{\epsilon}, \bar{h}_{1,j}^{\epsilon}, \bar{h}_{1,j}^{\epsilon} \right\}_{j \in [n]}$ given $g_1, g_2, h_1, h_2, h_3$.

Challenger $B_1$ samples $u_j, v_j, u'_j, v'_j$, and $\bar{a}$ for all $j \in [n]$ and obtains $\left\{ T, T_j, T_j' \right\}$ with $g_1, g_2, h_1$. Then, $B_1$ needs to
distinguish whether \( \{T, T', T''\} \) is the left distribution or right.

Setup: pick random generator \( h_{123} \) of \( H_N \). Sample \( w_0, w_1, w', w_0, w_1, \alpha, u_0 \sim \mu \mathbb{Z}_N \) and output

\[
\text{mpk} = \left( g_1^w, g_1^{w_0}, g_1^{w_1}, g_1^w, g_1, g_1^w, g_1, e(g_1, h_{123})^\alpha \right).
\]

(25)

Stage 1: adversary \( A \) queries the secret key corresponding to vector \( x = (x_1', x_2', \ldots, x_n') \). Challenger \( B_1 \) simulates the secret key generation algorithm and samples \( \tilde{r}, \tilde{r}_j, \tilde{T}_j \sim \mu \mathbb{Z}_N \) for all \( j \in [n] \) and outputs

\[
sk_{x_1'} = \left\{ \begin{array}{l}
K_0 = h_{123}^w h_2^{w_0} h_2^{w_1}, K_1 = h_1^2 h_2^{w_0} h_2^{w_1} \\
K_{1,j} = h_1^w (w_{x_1', w_{x_2'}} \cdot x_j) (w_{x_{j+1}', w_{x_{j+2}'}} \cdot x_{j+1}'), \tau_j(x_j, y_j) h_2^{w_{x_1', w_{x_2'}}} h_2^{w_{x_{j+1}', w_{x_{j+2}'}}} & \text{if } k < i \\
K_{2,j} = h_1^2 h_2^{w_0} h_2^{w_1}, K_{2,j} = h_1^2 h_2^{w_0} h_2^{w_1} & j \in [n]
\end{array} \right.
\]

(26)

Challenge: adversary \( A \) sends two equal-length plaintexts \((m_0, m_1)\) and the challenge access structure \( \nu^* = \{(y_j, z_j): j \in [n], y_j, z_j \in \mathbb{Z}_N^\ast\} \) to challenger \( B_1 \) (any query vector \( x' \) in Phase 1 and the challenge access structure \( \nu^* = \{(y_j, z_j): j \in [n], y_j, z_j \in \mathbb{Z}_N^\ast\} \) do not satisfy \( \sum_{j=1}^n \omega_j (y_j + x_{j'} \cdot z_j) = 1 \)). Challenger \( B_1 \) picks \( b \in \{0, 1\} \) and \( u \sim \mu \mathbb{Z}_N \) and outputs the challenge ciphertext:

\[
ct_{\nu'} = \left\{ \begin{array}{l}
C_0 = g_1^u g_2^{s_1} \\
C_{0,j} = g_1^{s_{x_1'} u_{x_2'}} g_2^{s_{x_{j+1}'} u_{x_{j+2}'}}, C_{0,j} = g_1^{s_{x_1'} u_{x_2'}} g_2^{s_{x_{j+1}'} u_{x_{j+2}'}} & j \in [n]
\end{array} \right.
\]

(27)

Stage 2: same as Stage 1.

Guess: adversary \( A \) outputs the guess \( b' \) about \( b \).

Note: the output is the entropy secret key if \( B_1 \) obtains the left distribution, which is \( \{T = h_2^{-1/2}, T_j = h_2^{-1/2} \} \). The output is the entropy pseudostandard secret key if \( B_1 \) obtains the right distribution, which is \( \{T = h_2^{-1/2}, T_j = h_2^{-1/2}, T'' = h_2^{-1/2} \} \). Challenger \( B_1 \) also solves \( \text{SD}_{P_1} \) with a non-negligible advantage if

\[
|\text{Adv}_1(\lambda) - \text{Adv}_1(\lambda)| > \varepsilon. \text{ Therefore, } \text{Game}_1 \text{ and Game}_{1,1} \text{ cannot be distinguished due to } \text{SD}_{P_1} \rightarrow P_2, P_2'. \]

\[ \square \]

\textbf{Lemma 4} (Game$_{i,1} \equiv$ Game$_{i,2}$). \textit{The advantage in Game$_{i,1}$ and Game$_{i,2}$ satisfies }\( |\text{Adv}_{i,1}(\lambda) - \text{Adv}_{i,2}(\lambda)| > \varepsilon \text{ for any adversaries } A. \)

\textbf{Proof.} Challenger \( B_2 \) samples \( u_j, v_j, u_{j'}, v_{j'} \), and \( \hat{u} \) for all \( j \in [n] \). The difference between Game$_{i,1}$ and Game$_{i,2}$ is only
the \(i\)th secret key query. The following shows that the challenger \(B_2\) cannot distinguish these two games.

Setup: pick random generator \(h_{123}\) of \(H_N\). Sample \(w, w_0, w_1, w', w'_0, w'_1, \alpha, u_0 \rightarrow_{R} Z_N\) and output

\[
\text{mpk} = (g_1, g'_1, g_t, g'_t, g_1, g'_1, g_0, g'_0, e(g_1, h_{123})^a).
\]

(28)

\[
\text{sk}_{\pi} \xrightarrow{k \leftarrow i} \begin{cases}
K_0 = h_{123}^a h_1^0 r, K_1 = h_2^0 h_3^0 \\
K_{1,j} = h_1(t w x^j u^j w) \tau_j (w_0 u_j w_0 v^j u_j v) h_2^{-r_j} (v_r^j u_r^j) \tau_j u_j v^j u_j v^j h_3^{-r_j} (v_r^j u_r^j) \\
K_{2,j} = h_1^0 h_2^0 h_3^0, K_{2,j} = h_1^0 h_2^0 h_3^0
\end{cases},
\]

(29)

and Game_{1,2} outputs

\[
\begin{cases}
K_0 = h_{123}^a h_1^0 r, K_1 = h_2^0 h_3^0 \\
K_{1,j} = h_1(t w x^j u^j w) \tau_j (w_0 u_j w_0 v^j u_j v) h_2^{-r_j} (v_r^j u_r^j) \tau_j u_j v^j u_j v^j h_3^{-r_j} (v_r^j u_r^j) \\
K_{2,j} = h_1^0 h_2^0 h_3^0, K_{2,j} = h_1^0 h_2^0 h_3^0
\end{cases}
\]

(30)

Adversary \(A\) observes that the only difference between Game_{1,1} and Game_{1,2} is the \(i\)th secret key query in \(K_i\). Firstly, \(h_1^0 h_2^0 r_j\) and \(h_1^0 h_2^0 r_j\) have the same distribution due to the random number \(\alpha, \tau, \tau\). Secondly, the output of the decryption algorithm in Game_{1,1} and Game_{1,2} is the same because \(e(g_1^0, h_2^0) = 1\). Therefore, the adversary \(A\) cannot distinguish these two secret keys.

Challenge: Game_{1,1} and Game_{1,2} have the same distribution because their outputs are entropy expansion challenge ciphertext.

Obtained from the above analysis, we have

\[|\text{Adv}_{1,1}(\lambda) - \text{Adv}_{1,2}(\lambda)| = 0. \]
Lemma 6 (GameGA ≡ GameGA−1). We know it in Table 2 easily (in fact, they have the same secret key and challenge ciphertext).

Lemma 7 (GameGA+1 ≡ GameGA−1).

Proof. Challenge B4 samples u0, vj, u0′, vj′ for all j ∈ [n]. The difference between these two games is the challenge ciphertext. In GameGA+1, the challenge ciphertext is obtained by m, while the challenge ciphertext in GameGA−1 is obtained by a random message. Let us prove that the two games are indistinguishable. Pick random generator h123 and h3 of HN and HPN respectively. Select a, ̃a ← RN and define h0 123 := h0 123/h0 3. We simulate GameGA+1 as follows:

Setup: pick random generator h123 of HN. Sample w, w0, w1, w′, w0′, w1′, α, u0 ← RN and output

\[
mpk = (g_1, g_1^w, g_1^w_0, g_1^w_1, g_1^{w_0}, g_1^{w_1}, g_1^{w_0’}, g_1^{w_1’}, e(g_1, h_{123}^{\bar{a}})).
\]

(32)

We can remove h0 3 because e(g1, h0 3) = 1.

Stage 1: adversary A queries the secret key corresponding to vector x* = (x′1, x′2, . . . , x′n). Challenger B4 simulates the secret key generation algorithm and picks r, rj, rj′ ← RN for all j ∈ [n], Output

\[
sk_x = \begin{cases}
K_0 = h_{123}^a h_1^{w0} h_2^{w0'}, K_1 = h_1^{w1} h_2^{w1'}\\
K_{1, j} = h_1^{w+x, w} r_j (u_j + jw_0) + x r_j (u_j + jw_1) h_2^{r_j (r_j + x, r_j') + r_j x (r_j + x', r_j')}\\
K_{2, j} = h_1^{w,x} h_2^{w,x'} r_j, K_0’ = h_1^{w,x} h_2^{w,x'} r_j.
\end{cases}
\]

(33)

The challenge: adversary A sends two equal-length plaintexts (m0 and m1) and the challenge access structure ν* = \{(yj, zj): j ∈ [n], yj, zj ∈ ZN\} to challenger B4 (any query vector x in Phase 1 and the challenge access structure ν* = \{(yj, zj): j ∈ [n], yj, zj ∈ ZN\} do not satisfy \(\sum_{j=1}^{n} \omega_j (y_j + x'_j \cdot z_j) = 1\)). Challenger B4 picks b ∈ {0, 1} and u ← RN, and the outputs challenge the ciphertext:

\[
ct_{x*} := \begin{cases}
C_0 = g_1^w g_2^{w'}\\
C_{0, j} = g_1^z_j (u_j)^{s_j, w} g_2^{s_j, w'} g_2^{s_j, w'}, C_{0, j}’ = g_1^z_j (u_j)^{s_j, w} g_2^{s_j, w’} g_2^{s_j, w’}\\
C_{1, j} = g_1^z_j g_2^{z_j}, C_{2, j} = g_1^z_j (u_j + jw_0) g_2^{z_j} g_2^{z_j}, C_{2, j}’ = g_1^z_j (u_j + jw_1) g_2^{z_j} g_2^{z_j’}\\
C = e(g_1^w g_2^{w’}, h_{123}^{\bar{a}}) \cdot e(g_1^w g_2^{w’}, h_{123}^{\bar{a}}) \cdot m_b
\end{cases}
\]

(34)

Guess: adversary A outputs the guess b’ about b.

We have e(g1^s1 g2^s2, h_{123}^{\bar{a}}), e(g1^s1 g2^s2, h_{123}^{\bar{a}}) = e(g1^s1 g2^s2, h_{123}^{\bar{a}}) e(g1^s1 g2^s2, h_{123}^{\bar{a}}), e(g1^s1 g2^s2, h_{123}^{\bar{a}}) in the entropy expansion challenge ciphertext. The distribution of e(g1^s1 g2^s2, h_{123}^{\bar{a}}) in G2 is a uniform distribution due to the random number ̃a, that is, the ciphertext which encrypted from a random number and the ciphertext which encrypted from m have the same distribution. Therefore, adversary A cannot distinguish these two entropy expansion ciphertexts.

Obtained from the above analysis, we have [AdvGA+1 (λ) − AdvGA−1 (λ)] = 0.

□
Theorem 1. Our CP-ABE scheme supporting arithmetic span programs is adaptively secure under the entropy expansion lemma and subgroup decision assumption decision. Also,

$$\max\{\text{Time}(B_0), \text{Time}(B_1), \text{Time}(B_2)\} \approx \text{Time}(A).$$  \quad (35)

Proof. The advantage of adversary A in our scheme is equivalent to the advantage in Game0 under the adaptively secure model. By Lemmas 1–7, we obtain

$$\text{Adv}_0(\lambda) = \text{Adv}_0(\lambda) - \text{Adv}_{\text{优势}}(\lambda) + \text{Adv}_{\text{优势}}(\lambda) - \text{Adv}_1(\lambda)$$

$$+ \cdots + \text{Adv}_Q(\lambda) - \text{Adv}_{Q+1}(\lambda)$$

$$+ \text{Adv}_{Q+1}(\lambda) - \text{Adv}_{\text{Final}}(\lambda) + \text{Adv}_{\text{Final}}(\lambda) \quad (36)$$

By Lemma 1, we know

$$\text{Adv}_0(\lambda) - \text{Adv}_{\text{优势}}(\lambda) \leq \epsilon.$$ \quad (37)

By Lemma 2, we know

$$\text{Adv}_{\text{优势}}(\lambda) - \text{Adv}_1(\lambda) = 0.$$ \quad (38)

The indistinguishability between Game$_i$ and Game$_{i+1}$ is due to

$$\text{Game}_i - \text{Game}_{i+1}$$

$$= (\text{Game}_i - \text{Game}_{i-1}) + (\text{Game}_{i-1} - \text{Game}_{i,2})$$

$$+ (\text{Game}_{i,2} - \text{Game}_{i,3}) + (\text{Game}_{i,3} - \text{Game}_{i+1}).$$ \quad (39)

By Lemma 3–6, we know

$$\text{Adv}_i(\lambda) - \text{Adv}_{i+1}(\lambda)$$

$$
\leq \text{Adv}_i(\lambda) - \text{Adv}_{i+1}(\lambda)
+ \text{Adv}_{i+1}(\lambda) - \text{Adv}_{i+2}(\lambda)
+ \text{Adv}_{i+2}(\lambda) - \text{Adv}_{i+3}(\lambda)
+ \text{Adv}_{i+3}(\lambda) - \text{Adv}_{i+1}(\lambda)
\leq 2\epsilon.$$ \quad (40)

By Lemma 7, we know

$$|\text{Adv}_{Q+1}(\lambda) - \text{Adv}_{\text{Final}}(\lambda)| = 0.$$ \quad (41)

Obviously, we have Adv$_{\text{Final}}(\lambda) = 0$.

In summary, the advantage of the adversary A in Game$_0$ is

$$\text{Adv}_0(\lambda) \leq (2Q + 1)\epsilon.$$ \quad (42)

That is, our scheme is adaptively secure under the entropy expansion lemma and subgroup decision assumption.

\[\square\]

3.6. Performance Analysis. At last, we show the difference between our scheme and the existing schemes that support arithmetic span programs in Table 3 (where “T” represents the operation time of the bilinear mapping). Compared with [11], the size of the public parameters of our scheme is smaller (from O (n) to O (1)) and adaptive security is achieved. Compared with [10], our scheme chooses the CP-ABE suitable for more flexible application scenarios and is based on the SD assumption to prove its adaptive security.

4. Conclusion

In this paper, we present a ciphertext-policy attribute-based encryption scheme that supports arithmetic span programs on composite-order bilinear groups. Firstly, we prove our entropy expansion lemma with a sequence of games and seven lemmas. Secondly, we prove that our scheme is adaptively secure under the conditions that entropy expansion lemma and subgroup decision assumption are true.

Appendix

Proof for the Bilinear Entropy Expansion Lemma

We first list the proof frame through a series of indistinguishable distributions:

| Scheme | Access policy | MPK | Assumption | Decryption complexity | Security |
|--------|---------------|-----|------------|-----------------------|----------|
| [11]   | KP            | O (n) | DBDH       | 2nT                   | Selective|
| [10]   | KP            | O (1) | MDDH       | 12nT                  | Adaptive |
| Our    | CP            | O (1) | SD         | 12nT                  | Adaptive |
Lemma 8

\[\begin{align*}
\text{ct: } C_0 &= g_1^r, C_{0,j} = g_1^{s^0}, C_{0,j} = g_1^{s^0}, C_{1,j} = g_1^s \\
C_{2,j} &= g_1^{s^j (w_j u_j w_j)} C_{2,j} = g_1^{s^j (w_j u_j w_j)} \quad j \in [n], \\
\text{sk: } K_1 &= h_1^r, K_{1,j} = h_2^{r u_{w_j} (w_j u_j w_j)} K_{2,j} = h_1^r \quad j \in [n].
\end{align*}\]

\[\begin{align*}
\text{sd_{aux}}^{n(r)}_{\mathcal{g}_1} \ ensued \quad \begin{align*}
\text{ct: } C_0 &= g_1^r g_2^r, C_{0,j} = g_1^{s^0} g_2^{s^0}, C_{0,j} = g_1^{s^0}, C_{1,j} = g_1^s g_2^s \\
C_{2,j} &= g_1^{s^j (w_j u_j w_j)} g_2^{s^j (w_j u_j w_j)} C_{2,j} = g_1^{s^j (w_j u_j w_j)} \quad j \in [n] , \\
\text{sk: } K_1 &= h_1^r, K_{1,j} = h_2^{r u_{w_j} (w_j u_j w_j)} h_2^{r u_{w_j} (w_j u_j w_j)} K_{2,j} = h_1^r h_2^r \quad j \in [n].
\end{align*}\]

(A.1)

\[\begin{align*}
\text{dhn_{aux}}^{n^2} \ ensued \quad \begin{align*}
\text{ct: } C_0 &= g_1^r g_2^r, C_{0,j} = g_1^{s^0} g_2^{s^0}, C_{0,j} = g_1^{s^0}, C_{1,j} = g_1^s g_2^s \\
C_{2,j} &= g_1^{s^j (w_j u_j w_j)} g_2^{s^j (w_j u_j w_j)} C_{2,j} = g_1^{s^j (w_j u_j w_j)} \quad j \in [n] , \\
\text{sk: } K_1 &= h_1^r h_2^r, K_{1,j} = h_2^{r u_{w_j} (w_j u_j w_j)} h_2^{r u_{w_j} (w_j u_j w_j)} K_{2,j} = h_1^r h_2^r \quad j \in [n].
\end{align*}\]
The following highlights the proof of Lemmas A.1 and A.2.

\[
\text{aux: } g_1, g_2, g_1^u, g_1^u, g_1^u, g_1^u,
\text{ct: } \left\{ g_1^{s_j}, g_1^{s_j(u_j^r w_j^r)}, g_1^{s_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

\[
\text{sk: } \left\{ h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

**Lemma A.1.** Under the DDH**\(H_N\)**, DDH**\(H_N\)**, SD**\(p_i\)**, and SD**\(p_i\)** assumptions, we have

\[
\text{Game}_{0}: \text{it is the same as the left distribution in Lemma A.1:}
\]

\[
\text{Game}_{0}': \text{modify sk as follows:}
\text{sk: } \left\{ h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

\[
\text{ct: } \left\{ g_1^{s_j}, g_1^{s_j(u_j^r w_j^r)}, g_1^{s_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

Now, we briefly explain that Game$_0$ = Game$_0'$. Under the DDH**\(H_N\)** assumption, we have

\[
\left\{ h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)} \right\}_{j \in [n]} = \left\{ h_2^{r_j(v_j^r w_j^r)}, h_2^{r_j(v_j^r w_j^r)}, h_2^{r_j(v_j^r w_j^r)}, h_2^{r_j(v_j^r w_j^r)} \right\}_{j \in [n]}, \text{ given } g_1, g_2, h_{13}, \]

where $v_j, v_j^r \rightarrow \mathbb{Z}_N$, and set $u_j = v_j + jw_1, u_j^r = v_j^r + jw_1$.

Game$_i$ ($i = 1, 2, \ldots, n + 1$): modify ct as follows:

\[
\text{ct: } \left\{ g_1^{s_j}, g_1^{s_j(u_j^r w_j^r)}, g_1^{s_j(u_j^r w_j^r)} \right\}_{j \in i}, \left\{ g_1^{s_j(u_j^r w_j^r)}, g_1^{s_j(u_j^r w_j^r)} \right\}_{j \in i} \]

\[
\text{sk: } \left\{ h_1^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)} \right\}_{j \in [n]}, \text{ given } g_1, g_2, h_{13}, h_2. \]

It is easy to know that Game$_0 = \text{Game}_i$. Then, we will prove that Game$_0 \approx \text{Game}_{i+1}$ through the following game sequence.

Game$_{i+1}$: modify ct as follows:

\[
\text{ct: } \left\{ g_1^{s_j}, g_1^{s_j(u_j^r w_j^r)}, g_1^{s_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

Now, we briefly explain that Game$_3 = \text{Game}_{i+1}$. Under the SD**\(p_i\)** assumption, we have

\[
g_3^{s_j} = g_1^{s_j}, \text{ given } g_1, g_2, h_{13}, h_2. \]

Game$_{i+2}$: modify sk as follows:

\[
\text{sk: } \left\{ h_1^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)}, h_2^{r_j(u_j^r w_j^r)} \right\}_{j \in [n]} \]

Now, we briefly explain that Game$_{i+2} \approx \text{Game}_{i+1}$. Under the DDH**\(p_i\)** assumption, we have
\[ \left\{ h_3^{r_{u,0}}, h_2^{r_{u_0}}, h_1, h_3^{r_{u_1}} \right\} \] 

\[ \approx \left\{ h_3^{r_{u,v_j}}, h_2^{r_{v_j}}, h_1, h_3^{r_{v_j}} \right\} \] 

given \( g_1, g_2, g_3, h_1, h_2, h_3 \), \hspace{1cm} (A.10)

where \( v_j, v_j' \leftarrow_{r} Z_{N_e} \), and set \( u_j = \tilde{u}_0 + (j - i)\nu_j \), \( u'_j = \tilde{u}_0 + (j - i)\nu_j' \), where

\[
\begin{align*}
\omega_0 &= \tilde{u}_0 \mod p_1 p_2, \\
\omega'_0 &= \tilde{u}_0' \mod p_1 p_2, \\
\omega_i &= \tilde{u}_0 - iw_i \mod p_3, \\
\omega'_i &= \tilde{u}_0' - iw'_i \mod p_3.
\end{align*}
\]

(A.11)

Game_{i,3}: modify \( \sk \) and \( \ct \) as follows:

\[
\begin{align*}
\ct_i := & \left\{ g_1^{z_i} g_2^{z_i} g_3^{s_i (u_0+iw_i)} g_3^{s_i u_i}, g_1^{s_i (u_0+iw_i)}, g_3^{s_i u_i} \right\}, \\
\sk_i := & \left\{ h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i}, h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i} \right\} \\
& h_1^{r_i}, h_2^{r_i}, h_3^{r_i}, r_i, r_i', r_i''.
\end{align*}
\]

(A.12)

It is easy to know that \( \text{Game}_{i,2} \simeq \text{Game}_{i,3} \) based on the fact \( w_0 + iw_i = u_i \mod p_2 \) and \( w'_0 + iw'_i = u'_i \mod p_2 \).

Game_{i,4}: modify \( \ct \) as follows:

\[
\begin{align*}
\ct_i := & \left\{ g_1^{n_i} g_2^{z_i}, g_1^{s_i (u_0+iw_i)} g_3^{s_i u_i}, g_1^{s_i (u_0+iw_i)} g_3^{s_i u_i} \right\}.
\end{align*}
\]

(A.13)

Now, we briefly explain that \( \text{Game}_{i,3} \simeq \text{Game}_{i,4} \). Under the SD_{P_3 \rightarrow P_1 P_2} assumption, we have

\[
g_i^2 \approx g_i^2 g_i^5, \quad \text{given } g_1, g_2, h_1, h_2, h_3.
\]

(A.14)

Game_{i,5}: modify \( \sk_i \) and \( \ct_i \) as follows:

\[
\begin{align*}
\ct_i := & \left\{ g_1^{z_i} g_2^{z_i} g_3^{s_i (u_0+iw_i)}, g_1^{s_i (u_0+iw_i)} g_2^{s_i u_i} \right\}, \\
\sk_i := & \left\{ h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i (u_0+iw_i)} h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i (u_0+iw_i)} \right\}.
\end{align*}
\]

(A.15)

It is easy to know that \( \text{Game}_{i,4} \simeq \text{Game}_{i,5} \) based on the fact \( \text{Game}_{i,2} \simeq \text{Game}_{i,3} \).

Game_{i,6}: modify \( \sk_i \) as follows:

\[
\sk_i := \left\{ h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i (u_0+iw_i)} h_1^{r_i (u_0+iw_i)} h_2^{r_i u_i} h_3^{r_i (u_0+iw_i)} \right\}_j,
\]

(A.16)

It is easy to know that \( \text{Game}_{i,5} \simeq \text{Game}_{i,6} \) based on the fact \( \text{Game}_{i,3} \simeq \text{Game}_{i,2} \).

Game_{i,7}: modify \( \ct_i \) as follows:

\[
\begin{align*}
\ct_i := & \left\{ g_1^{n_i} g_2^{z_i}, g_1^{s_i (u_0+iw_i)} g_2^{s_i u_i}, g_1^{s_i (u_0+iw_i)} g_2^{s_i u_i} \right\}.
\end{align*}
\]

(A.17)

It is easy to know that \( \text{Game}_{i,6} \simeq \text{Game}_{i,7} \) based on the fact \( \text{Game}_{i} \simeq \text{Game}_{i,1} \).

---

The rough proof of Lemma A.1 is as above. For more details, please refer to Lemma 22~29 which are in paper [10]. □
Proof. Under the DDH$_{g_1}$ assumption, we have
\[
\{g_2^{s_j}, g_2^{s_{\omega_j}}, g_2^{s_{\omega_j}}\}_{j \in [n]} = \{g_2^{s_j}, g_2^{s_{\omega_j}}, g_2^{s_{\omega_j}}\}_{j \in [n]}.
\] (A.20)
Suppose the adversary $A$ inputs $\{g_2^{s_j}, T_j, T'_j\}_{j \in [n]}$ and sets $u_j = r'_j \cdot (\bar{u}_j - rw)$, $u_j = (r'_j)^{-1} \cdot (\bar{u}_j - rw)$ where $r, r'_j, \bar{w}, \bar{w}, \bar{u}_j, \bar{u}_j \leftarrow_r \mathbb{Z}_N$. Then, the system outputs
\[
\begin{align*}
&\text{aux: } g_1, g_2, h_1, g_2^{s_j}, g_2^{s_{\omega_j}}, h_1, h_2^{s_{\omega_j}} \\
&\text{ct: } T_j, T_j', g_2^{s_j}, g_2^{s_{\omega_j}}, g_2^{s_{\omega_j}} (\bar{u}_j, r_j), T_j (r'_j), g_2^{s_j} (\bar{u}_j, r'_j), T_j (r'_j) \\
&\text{sk: } h_2^{s_j}, h_2^{s_{\omega_j}}, h_2^{s_{\omega_j}}, h_2^{s_{\omega_j}} \}_{j \in [n]}.\end{align*}
\] (A.21)

Now, we observe the above output and use this to illustrate the correctness of Lemma A.2.

1. If $T_j = g_2^{s_{\omega_j}}$ and $T_j' = g_2^{s_{\omega_j}}$ and we write $s_j u_j = (s_j / r_j) \cdot (\bar{u}_j - rw)$ and $s_j u'_j = (s_j / r'_j) \cdot (\bar{u}_j - rw)$, we get $\bar{u}_j = u_j r_j + ru_j$, $\bar{u}_j = u'_j r'_j + ru'_j$ and the left distribution.

2. If $T_j = g_2^{s_{\omega_j}}$ and $T_j' = g_2^{s_{\omega_j}}$ and we write $s_j u_j = (s_j / r_j) \cdot (\bar{u}_j - rv_j)$ and $s_j u'_j = (s_j / r'_j) \cdot (\bar{u}_j - rv'_j)$, we get $\bar{u}_j = u'_j r'_j + rv'_j$, $\bar{u}_j = u'_j r'_j + rv'_j$ and the right distribution.

That is, if we can determine $\{T_j, T_j'\}_{j \in [n]}$, then the DDH$_{g_1}$ problem will be solved. □

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (nos. 61702548 and 61601515) and the Fundamental and Frontier Technology Research of Henan Province (no. 162300410192).

References

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proceedings of 13th ACM Conference on Computer and Communications Security, pp. 89–98, ACM, Alexandria, VA, USA, November 2006.

[2] J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in Proceedings of EUROCRYPT 2008, vol. 4965, pp. 146–162, Springer-Verlag, Turkey, April 2008.

[3] J. Herranz, F. Llaguallamie, and C. Ráfols, "Constant size ciphertexts in threshold attribute-based encryption," in Proceedings of Public Key Cryptography, PKC 2010, pp. 19–34, Springer, Paris, France, May 2010.

[4] T. Okamoto and K. Takashima, "Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption," in Proceedings of CANS 2011, vol. 7092, pp. 138–159, Springer-Verlag, Sanya, China, December 2011.

[5] N. Attrapadung, B. Libert, and E. de Panafieu, "Expressive key-policy attribute-based encryption with constant-size ciphertexts," in Proceedings of Public Key Cryptography, PKC 2011, pp. 90–108, Springer, Taormina, Italy, March 2011.

[6] C. Chen, J. Chen, H. W. Lim et al., "Fully secure attribute-based systems with short ciphertexts/signatures and threshold access structures," in Proceedings of Cryptology CT-RSA 2013, pp. 50–67, Springer, San Francisco, CA, USA, February 2013.

[7] H. Wee, "Dual system encryption via predicate encodings," in Proceedings of TCC 2014, vol. 8349, pp. 616–637, Springer, San Diego, CA, USA, February 2014.

[8] M. Karchmer and A. Wigderson, "On span programs," in Proceedings of the 8th Annual IEEE Computer Society, pp. 102–111, IEEE, San Diego, CA, USA, May 1993.

[9] N. Attrapadung, G. Hanaoka, and S. Yamada, "Conversions among several classes of predicate encryption and applications to ABE with various compactness tradeoffs," in Proceedings of Cryptology-ASIACRYPT 2015, International Conference on the Theory and Application of Cryptology and Information Security, pp. 575–601, Springer, Auckland, New Zealand, November 2015.

[10] J. Chen, J. Gong, L. Kowalczyk, and H. Wee, "Unbounded ABE via bilinear entropy expansion, revisited," in Proceedings of Cryptology-CRYPTO 2018, Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 503–534, Springer, Tel Aviv, Israel, April 2018.

[11] Y. Ishai and H. Wee, "Partial garbling schemes and their applications," in Proceedings of ICALP 2014, vol. 8572, pp. 650–662, Springer, Copenhagen, Denmark, July 2014.

[12] A. Lewko and B. Waters, "New techniques for dual system encryption," in Proceedings of TCC 2005, vol. 3378, pp. 305–325, Springer, Cambridge, MA, USA, February 2005.