Hindawi
Security and Communication Networks
Volume 2020, Article ID 3265871, 16 pages
https://doi.org/10.1155/2020/3265871

Research Article

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

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 no 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 Takashima [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 achieved by the Boolean circuit, 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 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.

1.2. Organization. We first list some relevant knowledge in Section 2. Then, we present the formal definition of our scheme in Section 3.1 and propose the adaptive security model in Section 3.2. Specifically, we present our scheme in Section 3.3 and verify its correctness in Section 3.4. Finally, we prove its adaptive security by a series of experiments in Section 3.5.

2. Preliminaries

Notation. We let Z_p denote a ring of algebraic integers modules a prime number p and Z_p^n denote an m-dimension vector in 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, and G_T be bilinear groups of order $N = p_1 p_2 p_3$, 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_{p1}, g_{p2}, and g_{p3}, respectively. Let h be the generator of H_N, and h_1, h_2, and h_3 are the generators of H_{p1}, H_{p2}, and H_{p3}, respectively.

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

(1) Bilinearity: $e(g_0^a, h_0^a) = e(g_0, h_0)^{ab}$ for all $a, b \in Z_p$, $g_0 \in G_N$, $h_0 H_N$.

(2) Nondegeneracy: there exists $g_0 \in G_N$, $h_0 H_N$, such that the order of $e(g_0, h_0)$ is N.

(3) Computability: for all $g_0 \in G_N$, $h_0 H_N$, 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 (υ, ρ) is a map ρ: $[l] \longrightarrow [n]$, and a collection of row vectors $\upsilon = \{(\mathbf{y}_j, \mathbf{z}_j)\colon j \in [l], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$, for $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$, $\mathbf{x} \in Z_p^n$ satisfies (υ, ρ) iff there exists constants $\omega_1, \ldots, \omega_1 \in Z_p$, such that

$$\sum_{j=1}^{l} \omega_j \left(\mathbf{y}_j + x_{\rho(j)} \mathbf{z}_j\right) = \mathbf{1}, \tag{1}$$

where $\mathbf{1} \coloneqq (1, 0, \ldots,) \in Z_p^{l'}$.

Like in paper [9], we limit ρ to be an identity map and $l = n$.

2.3. Computational Assumptions

Assumption 1 ($\mathrm{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$ [12, 13]). We define the subgroup decision assumption (denoted by $\mathrm{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$) holds if for all probability polynomial time (PPT) adversaries A, and the following advantage function is negligible in λ:

Table 1: Development of ABE about access structure.

Scheme	Access policy	Access structure	Parameter	Security
[1]	KP	AND and OR gates	Not constant	Selective
[2]	KP	Inner product	Not constant	Selective
[3]	CP	Threshold	Constant	①
[4]	KP	Inner product	Constant	Adaptive
[5]	KP	Nonmonotonic	Constant	Selective
[6]	Both	Threshold	Constant	Adaptive
[7]	②	Arithmetic span programs		
[10]	KP	Arithmetic span programs	Constant	Adaptive
Our	CP	Arithmetic span programs	Constant	Adaptive

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).

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{SD}_{p_2\to p_2p_3}^{H_N}}(\lambda) := \left|\Pr\left[\mathscr{A}(G,D,T_0)=1\right]-\Pr\left[\mathscr{A}(G,D,T_1)=1\right]\right|, \tag{2}$$

where

$$\begin{aligned} D &:= (h_1,h_2,h_3,g_1,g_2,g_{23}),\\ g_{12} &\longleftarrow_R G_{p_2p_3}, T_0 \longleftarrow_R H_{p_2}, T_1 \longleftarrow_R H_{p_2p_3}. \end{aligned} \tag{3}$$

Assumption 2. ($\mathrm{DDH}_{p_2}^{G_N}$)The p_2-DDH assumption (denoted by $\mathrm{DDH}_{p_2}^{G_N}$), holds if for all probability polynomial time (PPT) adversaries $\mathscr{A}$, and the following advantage function is negligible in λ:

$$\mathrm{Adv}_{\mathscr{A}}^{\mathrm{DDH}_{p_2}^{G_N}}(\lambda) = \left|\Pr\left[\mathscr{A}(G,D,T_0)=1\right]-\Pr\left[\mathscr{A}(G,D,T_1)=1\right]\right|, \tag{4}$$

where

$$\begin{aligned} D &= (h_1,h_2,h_3,g_1,g_2,g_3),\\ T_0 &= (g_2^x,g_2^y,g_2^{xy}),\\ T_1 &= (g_2^x,g_2^y,g_2^z),\\ x,y,z &\longleftarrow_R Z_N. \end{aligned} \tag{5}$$

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

$$\left\{\begin{array}{c} \text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'} \\ \text{ct: } C_0 = g_1^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w}, C_{0,j}' = g_1^{s_j w'}, C_{1,j} = g_1^{s_j} \\ C_{2,j} = g_1^{s_j(w_0+jw_1)}, C_{2,j}' = g_1^{s_j(w_0'+jw_1')} \end{array}\right\}_{j\in[n]} \\ \text{sk: } K_1 = h_1^{r}, \left\{\begin{array}{c} K_{1,j} = h_1^{rw+r_j(w_0+jw_1)}, K_{2,j} = h_1^{r_j} \\ K_{1,j}' = h_1^{rw'+r_j'(w_0'+jw_1')}, K_{2,j}' = h_1^{r_j'} \end{array}\right\}_{j\in[n]} \end{array}\right\} \approx$$

$$\left\{\begin{array}{c} \text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'} \\ \text{ct: } C_0 = g_1^{s} g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j v_j}, C_{0,j}' = g_1^{s_j w'} g_2^{s_j v_j'}, C_{1,j} = g_1^{s_j} g_2^{s_j} \\ C_{2,j} = g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, C_{2,j}' = g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'} \end{array}\right\}_{j\in[n]} \\ \text{sk: } K_1 = h_1^{r} h_2^{r}, \left\{\begin{array}{c} K_{1,j} = h_1^{rw+r_j(w_0+jw_1)} h_2^{rv_j+r_j u_j}, K_{2,j} = h_1^{r_j} h_2^{r_j}, \\ K_{1,j}' = h_1^{rw'+r_j'(w_0'+jw_1')} h_2^{rv_j'+r_j u_j'}, K_{2,j}' = h_1^{r_j'} h_2^{r_j'} \end{array}\right\}_{j\in[n]} \end{array}\right\}, \tag{6}$$

where

$$w, w_0, w_1, w', w_0', w_1' \longleftarrow_R Z_N, v_j, u_j, v_j', u_j' \longleftarrow_R Z_N, s, s_j \longleftarrow_R Z_N, r_j, r_j' \longleftarrow_R Z_N. \tag{7}$$

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 (l^λ, l^n): input security parameters (l^λ, l^n) and output the master public key mpk and the master secret key msk.

Enc (mpk, v, m): input access structure $v = \{(\mathbf{y}_j, \mathbf{z}_j): j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$ and plaintext m and output ciphertext ct_v.

KeyGen $(\text{mpk}, \text{msk}, x)$: input the vector $\mathbf{x} \in Z_p^n$ and output the secret key $\text{sk}_\mathbf{x}$.

Dec $(\text{mpk}, \text{sk}_\mathbf{x}, \text{ct}_v)$: input $\text{sk}_\mathbf{x}$ and ct_v and output m if $\mathbf{x}, v$ satisfies $\sum_{j=1}^{l} \omega_j (\mathbf{y}_j + x_j \mathbf{z}_j) = \mathbf{1}$.

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 mpk to adversary A.

Stage 1: adversary A chooses $\mathbf{x}\prime$ 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 and m_1) and the challenge access structure $v^* = \{(\mathbf{y}_j, \mathbf{z}_j): j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$ to challenger B (any query vector $\mathbf{x}\prime$ and the challenge access structure $v^* = \{(\mathbf{y}_j, \mathbf{z}_j): j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$ do not satisfy $\sum_{j=1}^{l} \omega_j (\mathbf{y}_j + x_j \mathbf{z}_j) = \mathbf{1}$). Challenger B chooses $b \in \{0, 1\}$ randomly and computes $\text{ct}_{v^*} = \text{Enc}(\text{mpk}, v^*, m_b)$. Then, Challenger B sends the challenge ciphertext ct_{v^*} 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 $\text{Adv}_A(\lambda) = |\Pr(b' = b) - 1/2|$.

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 $(1^\lambda, 1^n)$: input the number of security parameters λ and attributes n and select $G := (N = p_1 p_2 p_3, G_N, H_N, e) \longleftarrow G(1^\lambda)$. Pick random generators g_1, h_1, and h_{123} of G_{p_1}, H_{p_1}, and H_N, respectively. Sample $w, w_0, w_1, w', w_0', w_1', \alpha, u_0 \longleftarrow_R Z_N$ and output the master public key

$$\text{mpk} = \Big((N, G_N, H_N, e); g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}, g_1^{u_0}, e(g_1, h_{123})^{\alpha}\Big), \tag{8}$$

and the master secret key

$$\text{msk} = (h_{123}, h_1, \alpha, w, w_0, w_1, w', w_0', w_1', u_0). \tag{9}$$

Enc (mpk, v, m): input the access structure $v = \{(\mathbf{y}_j, \mathbf{z}_j): j \in [l], \mathbf{y}_j, \mathbf{z}_j \in \mathbb{Z}_p^{l'}\}$ and the message $m \in \{0, 1\}^\lambda$. Select $s, s_j \longleftarrow_R Z_N$, $\mathbf{u} \longleftarrow_R Z_N^{l'-1}$ for all $j \in [n]$. Compute and output

$$\text{ct}_v = \left\{ \begin{array}{c} C_0 = g_1^s \\ \left\{ \begin{array}{c} C_{0,j} = g_1^{s\mathbf{y}_j \binom{u_0}{\mathbf{u}} + s_j w}, C_{0,j}' = g_1^{s\mathbf{z}_j \binom{u_0}{\mathbf{u}} + s_j w'} \\ C_{1,j} = g_1^{s_j}, C_{2,j} = g_1^{s_j (w_0 + j w_1)}, C_{2,j}' = g_1^{s_j (w_0' + j w_1')} \end{array} \right\}_{j \in [n]} \\ C = e(g_1, h_{123})^{\alpha s} \cdot m \end{array} \right\}. \tag{10}$$

KeyGen $(\text{mpk}, \text{msk}, \mathbf{x})$: input the master secret key msk and vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in Z_p^n$. Select $r, r_j, r_j' \longleftarrow_R Z_N$ for all $j \in [n]$ and output

$$\text{sk}_\mathbf{x} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 r}, K_1 = h_1^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r(w + x_j w') + r_j (w_0 + j w_1) + x_j r_j' (w_0' + j w_1')} \\ K_{2,j} = h_1^{r_j}, K_{2,j}' = h_1^{r_j'} \end{array} \right\}_{j \in [n]} \end{array} \right\}. \tag{11}$$

Dec $(\text{mpk}, \text{sk}_\mathbf{x}, \text{ct}_v)$: input secret key $\text{sk}_\mathbf{x}$ and ciphertext ct_v. If $(\mathbf{x}, v)$ satisfies $\sum_{j=1}^{l} \omega_j (\mathbf{y}_j + x_j \mathbf{z}_j) = \mathbf{1}$, then compute $m = C \cdot C' / e(C_0, K_0)$

where

$$C' = \prod_{j=1}^{n}\Big(e\big(C_{0,j}\cdot\big(C'_{0,j}\big)^{x_j},K_1\big)\cdot e\big(C_{1,j},K_{1,j}\big)^{-1} \cdot e\big(C_{2,j},K_{2,j}\big)\cdot e\big(\big(C_{2,j}\big)^{x_j},K'_{2,j}\big)\Big)^{\omega_j}. \tag{12}$$

3.4. Correctness. For all $(\mathbf{x}, v)$ satisfies $\sum_{j=1}^{l}\omega_j(\mathbf{y}_j + x_j\mathbf{z}_j) = \mathbf{1}$, we compute

$$\begin{aligned}
&e\big(C_{0,j}\cdot\big(C'_{0,j}\big)^{x_j},K_1\big)\cdot e\big(C_{1,j},K_{1,j}\big)^{-1}\cdot e\big(C_{2,j},K_{2,j}\big)\cdot e\big(\big(C_{2,j}\big)^{x_j},K'_{2,j}\big)\\
&\quad= e\left(g_1^{s\mathbf{y}_j\binom{u_0}{\mathbf{u}}+s_jw+x_js\mathbf{z}_j\binom{u_0}{\mathbf{u}}+x_js_jw'},h_1^{r}\right)\cdot e\left(g_1^{s_j},h_1^{r\left(w+x_jw'\right)+r_j\left(w_0+jw_1\right)+x_jr'_j\left(w'_0+jw'_1\right)}\right)^{-1}\\
&\qquad\cdot e\left(g_1^{s_j\left(w_0+jw_1\right)},h_1^{r_j}\right)\cdot e\left(g_1^{x_js_j\left(w'_0+jw'_1\right)},h_1^{r'_j}\right)\\
&\quad= e\left(g_1,h_1\right)^{rs\mathbf{y}_j\binom{u_0}{\mathbf{u}}+rs_jw+rx_js\mathbf{z}\binom{u_0}{\mathbf{u}}+rx_js_jw'}\cdot e\left(g_1,h_1\right)^{-rs_j\left(w+x_jw'\right)-r_js_j\left(w_0+jw_1\right)-x_jr'_js_j\left(w'_0+jw'_1\right)}\\
&\qquad\cdot e\left(g_1,h_1\right)^{r_js_j\left(w_0+jw_1\right)}\cdot e\left(g_1,h_1\right)^{x_jr'_js_j\left(w'_0+jw'_1\right)}\\
&\quad= e\left(g_1,h_1\right)^{rs\mathbf{y}_j\binom{u_0}{\mathbf{u}}+rs_jw+rx_js\mathbf{z}_j\binom{u_0}{\mathbf{u}}+rx_js_jw'-rs_j\left(w+x_jw'\right)-r_js_j\left(w_0+jw_1\right)-x_jr'_js_j\left(w'_0+jw'_1\right)+r_js_j\left(w_0+jw_1\right)+x_jr_{jl}s_j\left(w'_0+jw'_1\right)}\\
&\quad= e\left(g_1,h_1\right)^{rs\left(\mathbf{y}_j\binom{u_0}{\mathbf{u}}+x_j\mathbf{z}\binom{u_0}{\mathbf{u}}\right)},\\
&C' = \prod_{j=1}^{n}\Big(e\big(C_{0,j}\cdot\big(C'_{0,j}\big)^{x_j},K_1\big)\cdot e\big(C_{1,j},K_{1,j}\big)^{-1}\cdot e\big(C_{2,j},K_{2,j}\big)\cdot e\big(\big(C_{2,j}\big)^{x_j},K'_{2,j}\big)\Big)^{\omega_j}\\
&\quad= \prod_{j=1}^{n}\left(e\left(g_1,h_1\right)^{rs\left(\mathbf{y}_j\binom{u_0}{\mathbf{u}}+x_j\mathbf{z}\binom{u_0}{\mathbf{u}}\right)}\right)^{\omega_j}\\
&\quad= e\left(g_1,h_1\right)^{rs\sum_{j=1}^{n}\omega_j\left(\mathbf{y}_j\binom{u_0}{\mathbf{u}}+x_j\mathbf{z}\binom{u_0}{\mathbf{u}}\right)}\\
&\quad= e\left(g_1,h_1\right)^{rsu_0},\\
&\frac{C\cdot C'}{e\left(C_0,K_0\right)}\\
&\quad= \frac{e\left(g_1,h_{123}\right)^{\alpha s}\cdot m\cdot e\left(g_1,h_1\right)^{rsu_0}}{e\left(g_1^{s},h_{123}^{\alpha}\cdot h_1^{u_0r}\right)}\\
&\quad= \frac{e\left(g_1,h_{123}\right)^{\alpha s}\cdot m\cdot e\left(g_1,h_1\right)^{rsu_0}}{e\left(g_1^{s},h_{123}^{\alpha}\right)\cdot e\left(g_1^{s},h_1^{u_0r}\right)}\\
&\quad= \frac{e\left(g_1,h_{123}\right)^{\alpha s}\cdot e\left(g_1,h_1\right)^{rsu_0}\cdot m}{e\left(g_1,h_{123}\right)^{\alpha s}\cdot e\left(g_1,h_1\right)^{rsu_0}}\\
&\quad= m.
\end{aligned} \tag{13}$$

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:

$$\mathrm{ct}_v = \left\{ \begin{array}{c} C_0 = g_1^s \\ \left\{ \begin{array}{c} C_{0,j} = g_1^{s\mathbf{y}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + S_j w}, C'_{0,j} = g_1^{s\mathbf{z}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w'} \\ C_{1,j} = g_1^{s_j}, C_{2,j} = g_1^{s_j (w_0 + j w_1)}, C'_{2,j} = g_1^{s_j (w'_0 + j w'_1)} \end{array} \right\}_{j \in [n]} \\ C = e(g_1, h_{123})^{\alpha s} \cdot m \end{array} \right\}. \tag{14}$$

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

$$\mathrm{ct}_v = \left\{ \begin{array}{c} C_0 = g_1^s g_2^s \\ \left\{ \begin{array}{c} C_{0,j} = g_1^{s\mathbf{y}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + S_j w} g_2^{s v_j}, C'_{0,j} = g_1^{s\mathbf{z}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w'} g_2^{s v'_j} \\ C_{1,j} = g_1^{s_j} g_2^{s_j}, C_{2,j} = g_1^{s_j (w_0 + j w_1)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j (w'_0 + j w'_1)} g_2^{s_j u'_j} \end{array} \right\}_{j \in [n]} \\ C = e(g_1^s g_2^s, h_{123})^{\alpha} \cdot m \end{array} \right\}. \tag{15}$$

3.5.2. Secret Key Distributions

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

$$\mathrm{sk}_{\mathbf{x}} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 r}, K_1 = h_1^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r(w + x_j w') + r_j (w_0 + j w_1) + x_j r'_j (w'_0 + j w'_1)} \\ K_{2,j} = h_1^{r_j}, K'_{2,j} = h_1^{r'_j} \end{array} \right\}_{j \in [n]} \end{array} \right\}. \tag{16}$$

Entropy expansion secret key: compared to the standard secret key, we make a copy of $\left\{ h_1^r, h_1^{r(w + x_j w') + r_j (w_0 + j w_1) + x_j r'_j (w'_0 + j w'_1)}, h_1^{r_j}, h_1^{r'_j} \right\}_{j \in [n]}$ in H_{p_3}:

$$\mathrm{sk}_{\mathbf{x}} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 r}, K_1 = h_1^r h_2^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r(w + x_j w') + r_j (w_0 + j w_1) + x_j r'_j (w'_0 + j w'_1)} h_2^{r(v_j + x_j v'_j) + r_j u_j + x_j r'_j r'_j} \\ K_{2,j} = h_1^{r_j} h_2^{r_j}, K'_{2,j} = h_1^{r'_j} h_2^{r'_j} \end{array} \right\}_{j \in [n]} \end{array} \right\}. \tag{17}$$

Pseudostandard secret key: compared to the entropy expansion secret key, we make a copy of $\left\{h_2^r, h_2^{r(v_j+x_jv_j')+r_ju_j+x_jr'\mu_j'}, h_2^{r_j}, h_2^{r_j'}\right\}_{j\in[n]}$ in H_{p_3}:

$$\mathrm{sk}_{\mathbf{x}} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 r}, K_1 = h_1^r h_2^r h_3^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r\left(w+x_jw'\right)+r_j\left(w_0+jw_1\right)+x_jr_j'\left(w_0'+jw_1'\right)} h_2^{r\left(v_j+x_jv_j'\right)+r_ju_j+x_jr'\mu_j'} h_3^{r\left(v_j+x_jv_j'\right)+r_ju_j+x_jr'\mu_j'} \\ K_{2,j} = h_1^{r_j} h_2^{r_j} h_3^{r_j}, K_{2,j}' = h_1^{r_j'} h_2^{r_j'} h_3^{r_j'} \end{array} \right\}_{j\in[n]} \end{array} \right\}. \tag{18}$$

Pseudosemi-functional secret key: compared to the pseudostandard secret key, we sample $\widehat{\alpha} \longleftarrow H_{p_3}$:

$$\mathrm{sk}_{\mathbf{x}} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{\widehat{\alpha}} h_1^{u_0 r}, K_1 = h_1^r h_2^r h_3^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r\left(w+x_jw'\right)+r_j\left(w_0+jw_1\right)+x_jr_j'\left(w_0'+jw_1'\right)} h_2^{r\left(v_j+x_jv_j'\right)+r_ju_j+x_jr'\mu_j'} h_3^{r\left(v_j+x_jv_j'\right)+r_ju_j+x_jr'\mu_j'} \\ K_{2,j} = h_1^{r_j} h_2^{r_j} h_3^{r_j}, K_{2,j}' = h_1^{r_j'} h_2^{r_j'} h_3^{r_j'} \end{array} \right\}_{j\in[n]} \end{array} \right\}. \tag{19}$$

Semifunctional secret key: compared to the pseudo-semi-functional secret key, we remove $\left\{h_3^r, h_3^{r(v_j+x_jv_j')+r_ju_j+x_jr'\mu_j'}, h_3^{r_j}, h_3^{r_j'}\right\}_{j\in[n]}$:

$$\mathrm{sk}_{\mathbf{x}} = \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{\widehat{\alpha}} h_1^{u_0 r}, K_1 = h_1^r h_2^r \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{r\left(w+x_jw'\right)+r_j\left(w_0+jw_1\right)+x_jr_j'\left(w_0'+jw_1'\right)} h_2^{r\left(v_j+x_jv_j'\right)+r_ju_j+x_jr'\mu_j'} \\ K_{2,j} = h_1^{r_j} h_2^{r_j}, K_{2,j}' = h_1^{r_j'} h_2^{r_j'} \end{array} \right\}_{j\in[n]} \end{array} \right\}. \tag{20}$$

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 $\mathrm{Adv}_{xx}(\lambda)$. 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.

$\mathrm{Game}_{0'}$: compared to Game_0, all challenge ciphertext and secret keys are entropy expansion.

Game_i: compared to $\mathrm{Game}_{0'}$, the first $i-1$ secret keys are semifunctional and the last $Q-i+1$ are entropy expansion.

$\mathrm{Game}_{i,1}$: compared to Game_i, modify the i' key to the pseudostandard key.

$\mathrm{Game}_{i,2}$: compared to $\mathrm{Game}_{i,1}$, modify the i' key to the pseudo-semifunctional key.

$\mathrm{Game}_{i,3}$: compared to $\mathrm{Game}_{i,2}$, modify the i' key to the semifunctional key.

$\mathrm{Game}_{\mathrm{Final}}$: challenge ciphertext is the entropy expansion ciphertext about a random message, while the secret keys are semifunctional.

Lemma 1 ($\mathrm{Game}_0 \approx_c \mathrm{Game}_{0'}$). *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 $|Adv_0(\lambda) - Adv_{0'}(\lambda)| > \varepsilon$, that is, $Time(B_0) \approx Time(A)$.*

Table 2: Games for proving the adaptive security of our scheme.

Game	CT	SK		
		$\kappa < i$	$\kappa = i$	$\kappa > i$
0	Standard entropy expansion		Standard entropy expansion	
$0'$				
i	—	Semifunctional	Entropy expansion	Entropy expansion
i,1	—	—	Pseudostandard	—
i,2	—	—	Pseudo-semifunctional	—
i,3	—	—	Semifunctional	—
Final	Random message		Semifunctional	

Proof. Challenger B_0 obtains the following distribution:

$$\left\{\begin{array}{c} \text{aux: } \left(g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w'_0}, g_1^{w'_1}, g_1^{u_0}\right) \\ \text{ct: } C_0, \left\{C_{0,j}, C_{0,j}{}', C_{1,j}, C_{2,j}, C_{2,j}{}'\right\}_{j\in[n]} \\ \text{sk: } K_1, \left\{K_{1,j}, K_{1,j}{}', K_{2,j}, K_{2,j}{}'\right\}_{j\in[n]} \end{array}\right\}. \tag{21}$$

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 $\alpha \longleftarrow_R Z_N$ and output

$$\text{mpk} = \left(\text{aux}, e\left(g_1, h_{123}\right)^{\alpha}\right). \tag{22}$$

Stage 1: adversary A queries the secret key corresponding to the vector $\mathbf{x}\prime = (x_1{}', x_2{}', \ldots, x_n{}')$. Challenger B_0 simulates the secret key generation algorithm and picks $\widetilde{r}_j \longleftarrow_R Z_N$ for all $j \in [n]$. Output

$$\text{sk}_{\mathbf{x}\prime} = \left\{\begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 r}, K_1 = K_1^{\widetilde{r}_j} \\ \left\{\begin{array}{c} K_{1,j} = K_{1,j}^{\widetilde{r}_j} \cdot K_{1,j}{}'^{x_j \widetilde{r}_j} \\ K_{2,j} = K_{2,j}^{\widetilde{r}_j}, K_{2,j}{}' = K_{2,j}{}'^{\widetilde{r}_j} \end{array}\right\}_{j\in[n]} \end{array}\right\}. \tag{23}$$

Challenge: adversary A sends two equal-length plaintexts (m_0 and m_1) and the challenge access structure $v^* = \left\{(\mathbf{y}_j, \mathbf{z}_j): j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\right\}$ to challenger B_0 (any query vector $\mathbf{x}'$ in Phase 1 and the challenge access structure $v^* = \left\{(\mathbf{y}_j, \mathbf{z}_j): j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\right\}$ do not satisfy $\sum_{j=1}^{n} \omega_j (\mathbf{y}_j + x_j^{\prime} \cdot \mathbf{z}_j) = \mathbf{1}$). Challenger B_0 picks $b \in \{0, 1\}$ and $\mathbf{u} \longleftarrow_R Z_N^{l'-1}$ and outputs the challenge ciphertext:

$$\text{ct}_{v^*}: \left\{\begin{array}{c} C_0 = \mathrm{C}_0 \\ \left\{\begin{array}{c} C_{0,j} = g_1^{s\mathbf{y}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix}} \cdot \mathrm{C}_{0,j}, C'_{0,j} = g_1^{s\mathbf{z}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix}} \cdot C'_{0,j} \\ C_{1,j} = \mathrm{C}_{1,j}, C_{2,j} = \mathrm{C}_{2,j}, C'_{2,j} = C'_{2,j} \end{array}\right\}_{j\in[n]} \\ C = e\left(\mathrm{C}_0, h_{123}^{\alpha}\right) \cdot m_b \end{array}\right\}. \tag{24}$$

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 $|\text{Adv}_0(\lambda) - \text{Adv}_{0\prime}(\lambda)| \geq \varepsilon$. Game_0 and $\text{Game}_{0\prime}$ cannot be distinguished due to the indistinguishability of the left and right distributions. □

Lemma 2 ($\text{Game}_{0\prime} \equiv \text{Game}_i$). *We know it in Table 2 easily.*

Lemma 3 ($\text{Game}_i \approx \text{Game}_{i,1}$). *There exists a challenger B_1 who can solve $SD_{p_2 \longrightarrow p_2 p_3}^{H_N}$ with a non-negligible advantage if $|Adv(\lambda) - Adv_{i,1}(\lambda)| > \varepsilon$, that is, $Time(B_1) \approx Time(A)$.*

Proof. Firstly, from the $\text{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$ assumption, sample $\widetilde{r}, \widetilde{r}_j, \widetilde{r}_j{}' \longleftarrow_R Z_N$ for all $j \in [n]$ and we have $\left(h_2^{\widetilde{r}}, \left\{h_2^{\widetilde{r}_j}, h_2^{\widetilde{r}_j{}'}\right\}_{j\in[n]}\right) \approx_c \left(h_2^{\widetilde{r}} h_3^{\widetilde{r}}, \left\{h_2^{\widetilde{r}_j} h_3^{\widetilde{r}}, h_2^{\widetilde{r}_j{}'} h_3^{\widetilde{r}}\right\}_{j\in[n]}\right)$ given $g_1, g_{23}, h_1, h_2, h_3$

Challenger B_1 samples u_j, v_j, u'_j, v'_j, and $\widehat{\alpha}$ for all $j \in [n]$ and obtains $\left\{T, T_j, T'_j\right\}$ with g_1, g_2, and h_1. Then, B_1 needs to

distinguish whether $\{T, T_j, T'_j\}$ is the left distribution or right.

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

$$\text{mpk} := \left(g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w'_0}, g_1^{w'_1}, g_1^{u_0}, e(g_1, h_{123})^{\alpha}\right). \tag{25}$$

Stage 1: adversary A queries the secret key corresponding to vector $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$. Challenger B_1 simulates the secret key generation algorithm and samples $\tilde{r}, \tilde{r}_j, \tilde{r}_j{}' \longleftarrow_R Z_N$ for all $j \in [n]$ and outputs

$$\text{sk}_{\mathbf{x}'} = \left\{ \begin{array}{l} \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{\widehat{\alpha}} h_1^{u_0 r}, K_1 = h_1^{r} h_2^{r} \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\bar{r}\left(w + x'_j w'\right) + \bar{r}_j\left(w_0 + j w_1\right) + x_j \bar{r}'_j\left(w'_0 + j w'_1\right)} h_2^{\bar{r}\left(v_j + x_j v'_j\right) + \bar{r}_j u_j + x_j \bar{r}'_j u'_j} \\ K_{2,j} = h_1^{\bar{r}_j} h_2^{\bar{r}_j}, K'_{2,j} = h_1^{\bar{r}'_j} h_2^{\bar{r}'_j} \end{array} \right\}_{j \in [n]} \end{array} \right\} \text{ if } \kappa < i \\ \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{u_0 \bar{r}}, K_1 = h_1^{\bar{r}} T \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\bar{r}\left(w + x_j w'\right) + \bar{r}_j\left(w_0 + j w_1\right) + x'_j \bar{r}'_j\left(w'_0 + j w'_1\right)} T^{v_j + x'_j v'_j} T_j^{u_j} T_j{}'^{x'_j u'_j} \\ K_{2,j} = h_1^{\bar{r}_j} T_j, K'_{2,j} = h_1^{\bar{r}'_j} T'_j \end{array} \right\}_{j \in [n]} \end{array} \right\} \text{ if } \kappa = i \\ \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 \bar{r}}, K_1 = h_1^{\bar{r}} h_2^{\bar{r}} \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\bar{r}\left(w + x'_j w'\right) + \bar{r}_j\left(w_0 + j w_1\right) + x'_j \bar{r}'_j\left(w'_0 + j w'_1\right)} h_2^{\bar{r}\left(v_j + x'_j v'_j\right) + \bar{r}_j u_j + x'_j \bar{r}'_j u'_j} \\ K_{2,j} = h_1^{\bar{r}_j} h_2^{\bar{r}_j}, K'_{2,j} = h_1^{\bar{r}'_j} h_2^{\bar{r}'_j} \end{array} \right\}_{j \in [n]} \end{array} \right\} \text{ if } \kappa > i \end{array} \right\}. \tag{26}$$

Challenge: adversary A sends two equal-length plaintexts (m_0 and m_1) and the challenge access structure $v^* = \{(\mathbf{y}_j, \mathbf{z}_j) \colon j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$ to challenger B_1 (any query vector $\mathbf{x}'$ in Phase 1 and the challenge access structure $v^* = \{(\mathbf{y}_j, \mathbf{z}_j) \colon j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\}$ do not satisfy $\sum_{j=1}^{n} \omega_j (\mathbf{y}_j + x'_j \cdot \mathbf{z}_j) = \mathbf{1}$). Challenger B_1 picks $b \in \{0, 1\}$ and $\mathbf{u} \longleftarrow_R Z_N^{l'-1}$ and outputs the challenge ciphertext:

$$ct_{v^*} := \left\{ \begin{array}{c} C_0 = g_1^{s} g_2^{s} \\ \left\{ \begin{array}{c} C_{0,j} = g_1^{s\mathbf{y}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w} g_2^{s v_j}, C'_{0,j} = g_1^{s\mathbf{z}_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w'} g_2^{s v'_j} \\ C_{1,j} = g_1^{s_j} g_2^{s_j}, C_{2,j} = g_1^{s_j\left(w_o + j w_1\right)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j\left(w_o + j w'_1\right)} g_2^{s_j u_1{}'} \end{array} \right\}_{j \in [n]} \\ C = e\left(C_0, h_{123}^{\alpha}\right) \cdot m_b \end{array} \right\}. \tag{27}$$

Stage 2: same as Stage 1.

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

Note: the output is the entropy expansion secret key if B_1 obtains the left distribution, which is $\left\{T = h_2^{\tilde{r}}, T_j = h_2^{\tilde{r}_j}, T'_j = h_2^{\tilde{r}_j{}'}\right\}$ The output is the entropy pseudostandard secret key if B_1 obtains the right distribution, which is $\left\{T = h_2^{\tilde{r}} h_3^{\tilde{r}}, T_j = h_2^{\tilde{r}_j} h_3^{\tilde{r}_j}, T'_j = h_2^{\tilde{r}_j{}'} h_3^{\tilde{r}_j{}'}\right\}$. Challenger B_1 also solves $\text{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$ with a non-negligible advantage if $|\text{Adv}_i(\lambda) - \text{Adv}_{i,1}(\lambda)| > \varepsilon$. Therefore, Game_i and $\text{Game}_{i,1}$ cannot be distinguished due to $\text{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$. □

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

Proof. Challenger B_2 samples u_j, v_j, u'_j, v'_j, and $\widehat{\alpha}$ for all $j \in [n]$. The difference between $\text{Game}_{i,1}$ and $\text{Game}_{i,2}$ 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 $w, w_0, w_1, w', w_0', w_1', \alpha, u_0 \longleftarrow_R Z_N$ and output

$$\mathrm{mpk} := \left(g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}, g_1^{u_0}, e(g_1, h_{123})^{\alpha}\right). \tag{28}$$

Stage 1, 2: adversary A queries the secret key corresponding to vector $\mathbf{x}' = (x_1', x_2', \ldots, x_n')$. Challenger B_2 simulates the secret key generation algorithm and samples $\tilde{r}, \tilde{r}_j, \tilde{r}_j' \longleftarrow_R Z_N$ for all $j \in [n]$. $\mathrm{Game}_{i,1}$ outputs

$$\mathrm{sk}_{\mathbf{x}'} \xrightarrow{\kappa = i} \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_1^{u_0 \tilde{r}}, K_1 = h_1^{\tilde{r}} h_2^{\tilde{r}} h_3^{\tilde{r}} \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\tilde{r}\left(w + x_j' w'\right) + \tilde{r}_j \left(w_0 + j w_1\right) + x_j' \tilde{r}_j' \left(w_0' + w_1'\right)} h_2^{\tilde{r}'\left(v_j + x_j' v_j'\right) + \tilde{r}_j u_j + x_j' \tilde{r}_j' u_j'} h_3^{\tilde{r}\left(v_j + x_j' v_j'\right) + \tilde{r}_j u_j + x_j' \tilde{r}_j' u_j'} \\ K_{2,j} = h_1^{\tilde{r}_j} h_2^{\tilde{r}_j} h_3^{\tilde{r}_j}, K_{2,j}' = h_1^{\tilde{r}_j'} h_2^{\tilde{r}_j'} h_3^{\tilde{r}_j'} \end{array} \right\}_{j \in [n]} \end{array} \right\}, \tag{29}$$

and $\mathrm{Game}_{i,2}$ outputs

$$\mathrm{sk}_{\mathbf{x}'} \xrightarrow{\kappa = i} \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{\hat{\alpha}} h_1^{u_0 \tilde{r}}, K_1 = h_1^{\tilde{r}} h_2^{\tilde{r}} h_3^{\tilde{r}} \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\tilde{r}\left(w + x_j' w'\right) + \tilde{r}_j \left(w_0 + j w_1\right) + x_j' \tilde{r}_j' \left(w_0' + w_1'\right)} h_2^{\tilde{r}'\left(v_j + x_j' v_j'\right) + \tilde{r}_j u_j + x_j' \tilde{r}_j' u_j'} h_3^{\tilde{r}\left(v_j + x_j' v_j'\right) + \tilde{r}_j u_j + x_j' \tilde{r}_j' u_j'} \\ K_{2,j} = h_1^{\tilde{r}_j} h_2^{\tilde{r}_j} h_3^{\tilde{r}_j}, K_{2,j}' = h_1^{\tilde{r}_j'} h_2^{\tilde{r}_j'} h_3^{\tilde{r}_j'} \end{array} \right\}_{j \in [n]} \end{array} \right\}. \tag{30}$$

Adversary A observes that the only difference between $\mathrm{Game}_{i,1}$ and $\mathrm{Game}_{i,2}$ is i'th secret key query in K_0. Firstly, $h_3^{\hat{\alpha}} h_1^{u_0 \tilde{r}}$ and $h_1^{u_0 \tilde{r}}$ have the same distribution due to the random number $\hat{\alpha}, \tilde{r}$. Secondly, the output of the decryption algorithm in $\mathrm{Game}_{i,1}$ and $\mathrm{Game}_{i,2}$ is the same because $e(g_1^s, h_3^{\hat{\alpha}}) = 1$. Therefore, the adversary A cannot distinguish these two secret keys.

Challenge: $\mathrm{Game}_{i,1}$ and $\mathrm{Game}_{i,2}$ have the same distribution because their outputs are entropy expansion challenge ciphertext.

Obtained from the above analysis, we have $|\mathrm{Adv}_{i,1}(\lambda) - \mathrm{Adv}_{i,2}(\lambda)| = 0$. □

Lemma 5 ($\mathrm{Game}_{i,2} \approx \mathrm{Game}_{i,3}$). *There exists a challenger B_3 who can solve $SD_{p_2 \longrightarrow p_2 p_3}^{H_N}$ with a non-negligible advantage if $|Adv_{i,2}(\lambda) - Adv_{i,3}(\lambda)| > \varepsilon$. That is, $Time(B_3) \approx Time(A)$.*

Proof. Same as Lemma 3, challenger B_3 samples $u_j, v_j, u_j', v_j', \hat{\alpha}$ for all $j \in [n]$ and obtains $\{T, T_j, T_j'\}$ with g_1, g_2, and h_1. Then, B_3 needs to distinguish whether $\{T, T_j, T_j'\}$ is the left distribution or right. And challenger B_3 outputs

$$\mathrm{sk}_{\mathbf{x}'} \xrightarrow{\kappa = i} \left\{ \begin{array}{c} K_0 = h_{123}^{\alpha} h_3^{\hat{\alpha}} h_1^{u_0 \hat{r}}, K_1 = h_1^{\tilde{r}} T_j^{\tilde{r}} \\ \left\{ \begin{array}{c} K_{1,j} = h_1^{\tilde{r}\left(w + x_j w'\right) + \tilde{r}_j \left(w_0 + j w_1\right) + x_j \tilde{r}_j' \left(w_0' + j w_1'\right)} T^{v_j + x_j v_j'} T_j^{u_j} T_j'^{x_j u_j'} \\ K_{2,j} = h_1^{\tilde{r}_j} T_j, K_{2,j}' = h_1^{\tilde{r}_j'} T_j' \end{array} \right\}_{j \in [n]} \end{array} \right\}. \tag{31}$$

The output is a pseudo-semifunctional secret key if B_3 obtains the left distribution, which is $\left\{T = h_2^{\tilde{r}} h_3^{\tilde{r}}, T_j = h_2^{\tilde{r}_j} h_3^{\tilde{r}_j}, T_j' = h_2^{\tilde{r}_j'} h_3^{\tilde{r}_j'}\right\}$. The output is an entropy semifunctional secret key if B_3 obtains the right distribution, which is $\left\{T = h_2^{\tilde{r}}, T_j = h_2^{\tilde{r}_j}, T_j' = h_2^{\tilde{r}_j}\right\}$. Challenger B_3 also solves $\mathrm{SD}_{p_2 \longrightarrow p_2 p_3}^{H_N}$ with a non-negligible

advantage if $|\mathrm{Adv}_i(\lambda) - \mathrm{Adv}_{i,1}(\lambda)| > \varepsilon$. Therefore, $\mathrm{Game}_{i,2}$ and $\mathrm{Game}_{i,3}$ cannot be distinguished due to $\mathrm{SD}^{H_N}_{p_2 \longrightarrow p_2 p_3}$. □

Lemma 6 ($\mathrm{Game}_i \equiv \mathrm{Game}_{i-1,3}$). *We know it in Table 2 easily (in fact, they have the same secret key and challenge ciphertext).*

Lemma 7 ($\mathrm{Game}_{Q+1} \equiv \mathrm{Game}_{\mathrm{Final}}$)

Proof. Challenger B_4 samples u_j, v_j, u'_j, v'_j for all $j \in [n]$. The difference between these two games is the challenge ciphertext. In Game_{Q+1}, the challenge ciphertext is obtained by m, while the challenge ciphertext in $\mathrm{Game}_{\mathrm{Final}}$ is obtained by a random message. Let us prove that the two games are indistinguishable. Pick random generator h_{123} and h_3 of H_N and H_{p_3}, respectively. Select $\tilde{\alpha}, \hat{\alpha} \longleftarrow_R Z_N$ and define $h^{\alpha}_{123} := h^{\tilde{\alpha}}_{123}/h^{\hat{\alpha}}_3$. We simulate Game_{Q+1} as follows:

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

$$\mathrm{mpk} := \left(g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w'_0}, g_1^{w'_1}, g_1^{u_0}, e\left(g_1, h^{\tilde{\alpha}}_{123}\right)\right). \tag{32}$$

We can remove $h^{\hat{\alpha}}_3$ because $e(g_1, h^{\tilde{\alpha}}_3) = 1$.

Stage 1: adversary A queries the secret key corresponding to vector $\mathbf{x}\prime = (x'_1, x'_2, \ldots, x'_n)$. Challenger B_4 simulates the secret key generation algorithm and picks $r, r_j, r'_j \longleftarrow_R Z_N$ for all $j \in [n]$. Output

$$\mathrm{sk}_{\mathbf{x}\prime} = \left\{ \begin{array}{c} K_0 = h^{\alpha}_{123} h_1^{u_0 r}, K_1 = h_1^r h_2^r \\ K_{1,j} = h_1^{r\left(w + x_j w'\right) + r_j\left(w_0 + j w_1\right) + x_j r'_j\left(w'_0 + j w'_1\right)} h_2^{r\left(v_j + x_j v'_j\right) + r_j u_j + x_j r' \mu'_j} \\ K_{2,j} = h_1^{r_j} h_2^{r_j}, K'_{2,j} = h_1^{r'_j} h_2^{r'_j} \end{array} \right\}_{j \in [n]}. \tag{33}$$

Challenge: adversary A sends two equal-length plaintexts (m_0 and m_1) and the challenge access structure $v^* = \left\{(\mathbf{y}_j, \mathbf{z}_j) \colon j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\right\}$ to challenger B_4 (any query vector $\mathbf{x}\prime$ in Phase 1 and the challenge access structure $v^* = \left\{(\mathbf{y}_j, \mathbf{z}_j) \colon j \in [n], \mathbf{y}_j, \mathbf{z}_j \in Z_p^{l'}\right\}$ do not satisfy $\sum_{j=1}^{n} \omega_j(\mathbf{y}_j + x'_j \cdot \mathbf{z}_j) = \mathbf{1}$). Challenger B_4 picks $b \in \{0, 1\}$ and $\mathbf{u} \longleftarrow_R Z_N^{l'-1}$, and the outputs challenge the ciphertext:

$$\mathrm{ct}_{v^*} := \left\{ \begin{array}{c} C_0 = g_1^s g_2^s \\ \left\{ \begin{array}{c} C_{0,j} = g_1^{s y_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w} g_2^{s v_j}, C'_{0,j} = g_1^{s z_j \begin{pmatrix} u_0 \\ \mathbf{u} \end{pmatrix} + s_j w'} g_2^{s v_j'} \\ C_{1,j} = g_1^{s_j} g_2^{s_j}, C_{2,j} = g_1^{s_j\left(w_0 + j w_1\right)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j\left(w'_0 + j w_1'\right)} g_2^{s_j u_j'} \end{array} \right\}_{j \in [n]} \\ C = e\left(g_1^s g_2^s, h^{\tilde{\alpha}}_{123}\right) \cdot e\left(g_1^s g_2^s, h_3^{-\hat{\alpha}}\right) \cdot m_b \end{array} \right\}. \tag{34}$$

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

We have $e(g_1^S g_2^S, h^{\tilde{\alpha}}_{123}) \cdot e(g_1^S g_2^S, h_3^{-\hat{\alpha}}) = e(g_1^S g_2^S, h^{\tilde{\alpha}}_{123}) \cdot e(g_1^S g_2^S, h_3)^{-\hat{\alpha}}$ in the entropy expansion challenge ciphertext. The distribution of $e(g_1^S g_2^S, h_3)^{-\hat{\alpha}}$ in G_T is a uniform distribution due to the random number $\hat{\alpha}$, 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 $|\mathrm{Adv}_{Q+1}(\lambda) - \mathrm{Adv}_{\mathrm{Final}}(\lambda)| = 0$. □

Table 3: Comparison of schemes supporting arithmetic span program.

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

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

$$\max\{\mathrm{Time}(B_0), \mathrm{Time}(B_1), \mathrm{Time}(B_3)\} \approx \mathrm{Time}(A). \quad (35)$$

Proof. The advantage of adversary A in our scheme is equivalent to the advantage in Game_0 under the adaptively secure model. By Lemmas 1~7, we obtain

$$\begin{aligned}
&\mathrm{Adv}_0(\lambda) \\
&\quad = \mathrm{Adv}_0(\lambda) - \mathrm{Adv}_{0,}(\lambda) + \mathrm{Adv}_{0'}(\lambda) - \mathrm{Adv}_1(\lambda) \\
&\qquad + \cdots + \mathrm{Adv}_Q(\lambda) - \mathrm{Adv}_{Q+1}(\lambda) \\
&\qquad + \mathrm{Adv}_{Q+1}(\lambda) - \mathrm{Adv}_{\mathrm{Final}}(\lambda) + \mathrm{Adv}_{\mathrm{Final}}(\lambda) \\
&\quad \le \left|\mathrm{Adv}_0(\lambda) - \mathrm{Adv}_{0'}(\lambda)\right| + \left|\mathrm{Adv}_{0'}(\lambda) - \mathrm{Adv}_1(\lambda)\right| \\
&\qquad + \cdots + \left|\mathrm{Adv}_Q(\lambda) - \mathrm{Adv}_{Q+1}(\lambda)\right| \\
&\qquad + \left|\mathrm{Adv}_{Q+1}(\lambda) - \mathrm{Adv}_{\mathrm{Final}}(\lambda)\right| + \mathrm{Adv}_{\mathrm{Final}}(\lambda).
\end{aligned} \quad (36)$$

By Lemma 1, we know

$$\left|\mathrm{Adv}_0(\lambda) - \mathrm{Adv}_{0'}(\lambda)\right| \le \varepsilon. \quad (37)$$

By Lemma 2, we know

$$\left|\mathrm{Adv}_{0'}(\lambda) - \mathrm{Adv}_1(\lambda)\right| = 0. \quad (38)$$

The indistinguishability between Game_i and Game_{i+1} is due to

$$\begin{aligned}
&\mathrm{Game}_i - \mathrm{Game}_{i+1} \\
&\quad = \left(\mathrm{Game}_i - \mathrm{Game}_{i,1}\right) + \left(\mathrm{Game}_{i,1} - \mathrm{Game}_{i,2}\right) \\
&\quad + \left(\mathrm{Game}_{i,2} - \mathrm{Game}_{i,3}\right) + \left(\mathrm{Game}_{i,3} - \mathrm{Game}_{i+1}\right).
\end{aligned} \quad (39)$$

By Lemma 3~6, we know

$$\begin{aligned}
&\left|\mathrm{Adv}_i(\lambda) - \mathrm{Adv}_{i+1}(\lambda)\right| \\
&\quad \le \left|\mathrm{Adv}_i(\lambda) - \mathrm{Adv}_{i+1}(\lambda)\right| \\
&\qquad + \left|\mathrm{Adv}_{i,1}(\lambda) - \mathrm{Adv}_{i,2}(\lambda)\right| \\
&\qquad + \left|\mathrm{Adv}_{i,2}(\lambda) - \mathrm{Adv}_{i,3}(\lambda)\right| \\
&\qquad + \left|\mathrm{Adv}_{i,3}(\lambda) - \mathrm{Adv}_{i+1}(\lambda)\right| \\
&\quad \le 2\varepsilon.
\end{aligned} \quad (40)$$

By Lemma 7, we know

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

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

In summary, the advantage of the adversary A in Game_0 is

$$\mathrm{Adv}_0(\lambda) \le (2Q+1)\varepsilon. \quad (42)$$

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

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:

$$\left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w}, C'_{0,j} = g_1^{s_j w'}, C_{1,j} = g_1^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_1^{r}, \left\{\begin{array}{c} K_{1,j} = h_1^{rw+r_j(w_0+jw_1)}, K_{2,j} = h_1^{r_j}\\ K'_{1,j} = h_1^{rw'+r_j'(w_0'+jw_1')}, K'_{2,j} = h_1^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\},$$

$$\mathrm{SD}_{P_1}^{H_N} \underset{\approx}{\longrightarrow} {}_{P_1P_2P_3} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w}, C'_{0,j} = g_1^{s_j w'}, C_{1,j} = g_1^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_{123}^{r}, \left\{\begin{array}{c} K_{1,j} = h_{123}^{rw+r_j(w_0+jw_1)}, K_{2,j} = h_{123}^{r_j}\\ K'_{1,j} = h_{123}^{rw'+r_j'(w_0'+jw_1')}, K'_{2,j} = h_{123}^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\},$$

$$\mathrm{SD}_{P_1}^{G_N} \underset{\approx}{\longrightarrow} {}_{P_1P_2} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j w}, C'_{0,j} = g_1^{s_j w'} g_2^{s_j w'}, C_{1,j} = g_1^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_{123}^{r}, \left\{\begin{array}{c} K_{1,j} = h_{123}^{rw+r_j(w_0+jw_1)}, K_{2,j} = h_{123}^{r_j}\\ K'_{1,j} = h_{123}^{rw'+r_j'(w_0'+jw_1')}, K'_{2,j} = h_{123}^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\},$$

$$\mathrm{DDH}_{P_3}^{H_N} \underset{\approx}{} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j w}, C'_{0,j} = g_1^{s_j w'} g_2^{s_j w'}, C_{1,j} = g_1^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_{13}^{r}h_2^{r}, \left\{\begin{array}{c} K_{1,j} = h_{13}^{rw+r_j(w_0+jw_1)}h_2^{rw+r_j(w_0+jw_1)}, K_{2,j} = h_{13}^{r_j}h_2^{r_j}\\ K'_{1,j} = h_{13}^{rw'+r_j'(w_0'+jw_1')}h_2^{rw'+r_j'(w_0'+jw_1')}, K'_{2,j} = h_{13}^{r_j'}h_2^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\}, \tag{A.1}$$

$$\underset{\approx}{\text{Lemma 8}} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j w}, C'_{0,j} = g_1^{s_j w'} g_2^{s_j w_j'}, C_{1,j} = g_1^{s_j} g_2^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_{13}^{r}h_2^{r}, \left\{\begin{array}{c} K_{1,j} = h_{13}^{rw+r_j(w_0+jw_1)}h_2^{rw+r_j u_j}, K_{2,j} = h_{13}^{r_j}h_2^{r_j}\\ K'_{1,j} = h_{13}^{rw'+r_j'(w_0'+jw_1')}h_2^{rw+r_j u_j'}, K'_{2,j} = h_{13}^{r_j'}h_2^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\},$$

$$\mathrm{SD}_{P_1}^{H_N} \underset{\approx}{\longrightarrow} {}_{P_1P_3} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j w}, C'_{0,j} = g_1^{s_j w'} g_2^{s_j w_j'}, C_{1,j} = g_1^{s_j} g_2^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_1^{r}h_2^{r}, \left\{\begin{array}{c} K_{1,j} = h_1^{rw+r_j(w_0+jw_1)}h_2^{rw+r_j u_j}, K_{2,j} = h_1^{r_j}h_2^{r_j}\\ K'_{1,j} = h_1^{rw'+r_j'(w_0'+jw_1')}h_2^{rw+r_j u_j'}, K'_{2,j} = h_1^{r_j'}h_2^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\},$$

$$\underset{\approx}{\text{Lemma 9}} \left\{\begin{array}{c}\text{aux: } g_1, g_1^{w}, g_1^{w_0}, g_1^{w_1}, g_1^{w'}, g_1^{w_0'}, g_1^{w_1'}\\ \text{ct: } C_0 = g_1^{s}g_2^{s}, \left\{\begin{array}{c} C_{0,j} = g_1^{s_j w} g_2^{s_j v_j}, C'_{0,j} = g_1^{s_j w'} g_2^{s_j v_j'}, C_{1,j} = g_1^{s_j} g_2^{s_j}\\ C_{2,j} = g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, C'_{2,j} = g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\end{array}\right\}_{j\in[n]}\\ \text{sk: } K_1 = h_1^{r}h_2^{r}, \left\{\begin{array}{c} K_{1,j} = h_1^{rw+r_j(w_0+jw_1)}h_2^{rv_j+r_j u_j}, K_{2,j} = h_1^{r_j}h_2^{r_j}\\ K'_{1,j} = h_1^{rw'+r_j'(w_0'+jw_1')}h_2^{rv_j'+r_j u_j'}, K_{2,j}' = h_1^{r_j'}h_2^{r_j'}\end{array}\right\}_{j\in[n]}\end{array}\right\}.$$

The following highlights the proof of Lemmas A.1 and A.2.

Lemma A.1. *Under the* $DDH_{p_2}^{H_N}$, $DDH_{p_3}^{H_N}$, $SD_{p_1 \longrightarrow p_1p_3}^{G_N}$, *and* $SD_{p_3 \longrightarrow p_2p_3}^{G_N}$ *assumptions, we have*

$$\left\{\begin{array}{c} \text{aux: } g_1, g_2, g_1^{w}, g_1^{w'}, g_1^{w_0}, g_1^{w_0'} \\ \text{ct: } \left\{g_1^{s_j}, g_1^{s_j(w_0+jw_1)}, g_1^{s_j(w_0'+jw_1')}\right\}_{j\in[n]} \\ \text{sk: } \left\{h_{13}^{r_j(w_0+jw_1)} h_2^{r_j(w_0+jw_1)}, h_{13}^{r_j'(w_0'+jw_1')} h_2^{r_j'(w_0'+jw_1')}, h_{13}^{r_j} h_2^{r_j}, h_{13}^{r_j'} h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\} \overset{\mathrm{DDH}_{p_2}^{G_N}}{\approx}$$
$$\left\{\begin{array}{c} \text{aux: } g_1, g_2, g_1^{w}, g_1^{w'}, g_1^{w_0}, g_1^{w_0'} \\ \text{ct: } \left\{g_1^{s_j} g_2^{s_j}, g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\right\}_{j\in[n]} \\ \text{sk: } \left\{h_{13}^{r_j(w_0+jw_1)} h_2^{r_j u_j}, h_{13}^{r_j'(w_0'+jw_1')} h_2^{r_j' \mu_j'}, h_{13}^{r_j} h_2^{r_j}, h_{13}^{r_j'} h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\}. \tag{A.2}$$

Proof. The proof is similar to Lemma 3 in paper [10], and we first modify the game sequence in Lemma 3.

Game_0: it is the same as the left distribution in Lemma A.1:

$$\left\{\begin{array}{c} \text{aux: } g_1, g_2, g_1^{w}, g_1^{w'}, g_1^{w_0}, g_1^{w_0'} \\ \text{ct: } \left\{g_1^{s_j}, g_1^{s_j(w_0+jw_1)}, g_1^{s_j(w_0'+jw_1')}\right\}_{j\in[n]} \\ \text{sk: } \left\{h_{13}^{r_j(w_0+jw_1)} h_2^{r_j(w_0+jw_1)}, h_{13}^{r_j'(w_0'+jw_1')} h_2^{r_j'(w_0'+jw_1')}, h_{13}^{r_j} h_2^{r_j}, h_{13}^{r_j'} h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\}. \tag{A.3}$$

$\mathrm{Game}_{0_\prime}$: modify sk as follows:

$$\text{sk: } \left\{h_{13}^{r_j(w_0+jw_1)} h_2^{r_j u_j}, h_{13}^{r_j'(w_0'+jw_1')} h_2^{r_j' \mu_j'}, h_{13}^{r_j} h_2^{r_j}, h_{13}^{r_j'} h_2^{r_j'}\right\}_{j\in[n]}. \tag{A.4}$$

Now, we briefly explain that $\mathrm{Game}_0 \approx \mathrm{Game}_{0'}$. Under the $\mathrm{DDH}_{p_2}^{H_N}$ assumption, we have

$$\begin{aligned} &\left\{h_2^{r_j(w_0+jw_1)}, h_2^{r_j'(w_0'+jw_1')}, h_2^{r_j}, h_2^{r_j'}\right\}_{j\in[n]} \\ &\quad \approx \left\{h_2^{r_j v_j}, h_2^{r_j' v_j'}, h_2^{r_j}, h_2^{r_j'}\right\}_{j\in[n]}, \quad \text{given } g_1, g_2, h_{13}, \end{aligned} \tag{A.5}$$

where $v_j, v_j' \longleftarrow_R Z_N$, and set $u_j = v_j + jw_1$, $u_j' = v_j' + jw_1'$.

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

$$\text{ct: } \left\{\begin{array}{l} \left\{g_1^{s_j} g_2^{s_j}, g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\right\}_{j<i}, \\ \left\{g_1^{s_j}, g_1^{s_j(w_0+jw_1)}, g_1^{s_j(w_0'+jw_1')}\right\}_{j\geq i}. \end{array}\right. \tag{A.6}$$

It is easy to know that $\mathrm{Game}_{0_\prime} \approx \mathrm{Game}_1$. Then, we will prove that $\mathrm{Game}_i \approx \mathrm{Game}_{i+1}$ through the following game sequence.

$\mathrm{Game}_{i,1}$: modify ct_i as follows:

$$\mathrm{ct}_i\text{: } \left\{g_1^{s_i} g_3^{s_i}, g_1^{s_i(w_0+iw_1)} g_3^{s_i(w_0+iw_1)}, g_1^{s_i(w_0'+iw_1')} g_3^{s_i(w_0'+iw_1')}\right\}. \tag{A.7}$$

Now, we briefly explain that $\mathrm{Game}_i \approx \mathrm{Game}_{i,1}$. Under the $\mathrm{SD}_{p_1 \longrightarrow p_1p_3}^{G_N}$ assumption, we have

$$g_1^{s_i} \approx g_1^{s_i} g_3^{s_i}, \quad \text{given } g_1, g_2, h_{13}, h_2. \tag{A.8}$$

$\mathrm{Game}_{i,2}$: modify sk_j as follows:

$$\begin{aligned} \mathrm{sk}_j\text{: } &\left\{h_1^{r_j(w_0+jw_1)} h_2^{r_j u_j} h_3^{r_j u_j}, h_1^{r_j'(w_0'+jw_1')} h_2^{r_j u_j'} h_3^{r_j u_j'}, \right. \\ &\left. h_1^{r_j} h_2^{r_j} h_3^{r_j}, h_1^{r_j'} h_2^{r_j'} h_3^{r_j'}\right\}_{j\neq i}. \end{aligned} \tag{A.9}$$

Now, we briefly explain that $\mathrm{Game}_{i,1} \approx \mathrm{Game}_{i,2}$. Under the $\mathrm{DDH}_{p_3}^{H_N}$ assumption, we have

$$\left\{h_3^{r_j w_1}, h_2^{r_j' w_1'}, h_3^{r_j}, h_3^{r_j'}\right\}_{j\in[n]} \approx \left\{h_3^{r_j v_j}, h_3^{r_j' v_j'}, h_3^{r_j}, h_3^{r_j'}\right\}_{j\in[n]}, \quad \text{given } g_1, g_2, g_3, h_1, h_2, h_3, \tag{A.10}$$

where $v_j, v_j' \longleftarrow_R Z_N$, and set $u_j = \widehat{w}_0 + (j-i)v_j$, $u_j' = \widehat{w}_0{}' + (j-i)v_j'$, where

$$\begin{cases} w_0 = \widehat{w}_0 \bmod p_1 p_2, w_0 = \widehat{w}_0 - iw_1 \bmod p_3, \\ w_0' = \widehat{w}_0{}' \bmod p_1 p_2, w_0' = \widehat{w}_0{}' - iw_1' \bmod p_3. \end{cases} \tag{A.11}$$

$\text{Game}_{i,3}$: modify sk_i and ct_i as follows:

$$\begin{aligned} \text{ct}_i&: \left\{g_1^{s_i} g_3^{s_i}, g_1^{s_i(w_0+iw_1)} g_3^{s_i u_i}, g_1^{s_i(w_0'+iw_1')} g_3^{s_i u_i'}\right\}, \\ \text{sk}_i&: \left\{h_1^{r_i(w_0+iw_1)} h_2^{r_i u_i} h_3^{r_i u_i}, h_1^{r_i'(w_0'+iw_1')} h_2^{r_i u_i'} h_3^{r_i u_i'}, \right. \\ &\qquad \left. h_1^{r_i} h_2^{r_i} h_3^{r_i}, h_1^{r_i'} h_2^{r_i'} h_3^{r_i'}\right\}. \end{aligned} \tag{A.12}$$

It is easy to know that $\text{Game}_{i,2} \approx \text{Game}_{i,3}$ based on the fact $w_0 + iw_1 = u_i \bmod p_3$ and $w_0' + iw_1' = u_i' \bmod p_3$.

$\text{Game}_{i,4}$: modify ct_i as follows:

$$\text{ct}_i: \left\{g_1^{s_i} g_2^{s_i} g_3^{s_i}, g_1^{s_i(w_0+iw_1)} g_2^{s_i u_i} g_3^{s_i u_i}, g_1^{s_i(w_0'+iw_1')} g_2^{s_i u_i'} g_3^{s_i u_i'}\right\}. \tag{A.13}$$

Now, we briefly explain that $\text{Game}_{i,3} \approx \text{Game}_{i,4}$. Under the $\text{SD}_{p_3 \longrightarrow p_2 p_3}^{G_N}$ assumption, we have

$$g_3^{s_i} \approx g_2^{s_i} g_3^{s_i}, \quad \text{given } g_1, g_2, h_1, h_{23}. \tag{A.14}$$

$\text{Game}_{i,5}$: modify sk_i and ct_i as follows:

$$\begin{aligned} \text{ct}_i&: \left\{g_1^{s_i} g_2^{s_i} g_3^{s_i}, g_1^{s_i(w_0+iw_1)} g_2^{s_i u_i} g_3^{s_i(w_0+iw_1)}, \right. \\ &\qquad \left. g_1^{s_i(w_0'+iw_1')} g_2^{s_i u_i'} g_3^{s_i(w_0'+iw_1')}\right\}, \\ \text{sk}_i&: \left\{h_1^{r_i(w_0+iw_1)} h_2^{r_i u_i} h_3^{r_i(w_0+iw_1)}, h_1^{r_i'(w_0'+iw_1')} h_2^{r_i u_i'} h_3^{r_i'(w_0'+iw_1')}, \right. \\ &\qquad \left. h_1^{r_i} h_2^{r_i} h_3^{r_i}, h_1^{r_i'} h_2^{r_i'} h_3^{r_i'}\right\}. \end{aligned} \tag{A.15}$$

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

$\text{Game}_{i,6}$: modify sk_j as follows:

$$\begin{aligned} \text{sk}_j&: \left\{h_1^{r_j(w_0+jw_1)} h_2^{r_j u_j} h_3^{r_j(w_0+jw_1)}, h_1^{r_j'(w_0'+jw_1')} h_2^{r_j u_j'} h_3^{r_j(w_0+jw_1)}, \right. \\ &\qquad \left. h_1^{r_j} h_2^{r_j} h_3^{r_j}, h_1^{r_j'} h_2^{r_j'} h_3^{r_j'}\right\}_{j\neq i}. \end{aligned} \tag{A.16}$$

It is easy to know that $\text{Game}_{i,5} \approx \text{Game}_{i,6}$ based on the fact $\text{Game}_{i,1} \approx \text{Game}_{i,2}$.

$\text{Game}_{i,7}$: modify ct_i as follows:

$$\text{ct}_i: \left\{g_1^{s_i} g_2^{s_i}, g_1^{s_i(w_0+iw_1)} g_2^{s_i u_i}, g_1^{s_i(w_0'+iw_1')} g_2^{s_i u_i'}\right\}. \tag{A.17}$$

It is easy to know that $\text{Game}_{i,6} \approx \text{Game}_{i,7}$ based on the fact $\text{Game}_i \approx \text{Game}_{i,1}$:

$$\text{Game}_{n+1}: \left\{\begin{array}{c} \text{aux: } g_1, g_2, g_1^{w}, g_1^{w'}, g_1^{w_0}, g_1^{w_0'} \\ \text{ct: } \left\{g_1^{s_j} g_2^{s_j}, g_1^{s_j(w_0+jw_1)} g_2^{s_j u_j}, g_1^{s_j(w_0'+jw_1')} g_2^{s_j u_j'}\right\}_{j\in[n]} \\ \text{sk: } \left\{h_{13}^{r_j(w_0+jw_1)} h_2^{r_j u_j}, h_{13}^{r_j'(w_0'+jw_1')} h_2^{r_j' u_j'}, h_{13}^{r_j} h_2^{r_j}, h_{13}^{r_j'} h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\}. \tag{A.18}$$

The rough proof of Lemma A.1 is as above. For more details, please refer to Lemma 22~29 which are in paper [10]. □

Lemma A.2. *Sample $w, u_j, v_j, s, s_j, r_j \longleftarrow_R Z_N$, and we have*

$$\left\{\begin{array}{c} \text{aux: } g_1, g_2, h_1, g_1^{w}, g_1^{w'}, h_1^{w}, h_1^{w'} \\ \text{ct: } \left\{g_2^{s_j w}, g_2^{s_j w'}, g_2^{s_j}, g_2^{s_j u_j}, g_2^{s_j u_j'}\right\}_{j\in[n]} \\ \text{sk: } h_2^{r}, \left\{h_2^{rw+r_j u_j}, h_2^{r_j}, h_2^{rw'+r_j u_j'}, h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\} \overset{\text{DDH}_{p_2}^{G_N}}{\approx} \left\{\begin{array}{c} aux: g_1, g_2, h_1, g_1^{w}, g_1^{w'}, h_1^{w}, h_1^{w'} \\ ct: \left\{g_2^{s_j v_j}, g_2^{s_j v_j'}, g_2^{s_j}, g_2^{s_j u_j}, g_2^{s_j u_j'}\right\}_{j\in[n]} \\ sk: h_2^{r}, \left\{h_2^{rv_j+r_j u_j}, h_2^{r_j}, h_2^{rv_j'+r_j u_j'}, h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\}. \tag{A.19}$$

Proof. Under the $\mathrm{DDH}_{p_2}^{G_N}$ assumption, we have

$$\left\{g_2^{s_j}, g_2^{s_j w}, g_2^{s_j w'}\right\}_{j\in[n]} \approx \left\{g_2^{s_j}, g_2^{s_j v_j}, g_2^{s_j v_j'}\right\}_{j\in[n]}. \tag{A.20}$$

Suppose the adversary A inputs $\left\{g_2^{s_j}, T_j, T_j'\right\}_{j\in[n]}$ and sets $u_j = r_j^{-1}\cdot(\widehat{u}_j - rw)$, $u_j' = (r_j')^{-1}\cdot(\widehat{u}_j{}' - rw')$ where $r, r_j, \widehat{w}, \widehat{w}', \widehat{u}_j, \widehat{u}_j{}' \longleftarrow_R Z_N$. Then, the system outputs

$$\left\{\begin{array}{c} \text{aux: } g_1, g_2, h_1, g_1^{\widehat{w}}, g_1^{\widehat{w}'}, h_1^{\widehat{w}}, h_1^{\widehat{w}'} \\ \text{ct: } \left\{T_j, T_j', g_2^{s_j}, g_2^{s_j\cdot\left(\widehat{u}_j/r_j\right)}\cdot T_j^{-\left(r/r_j\right)}, g_2^{s_j\cdot\left(\widehat{u}_j{}'/r_j'\right)}\cdot T_j^{-\left(r/r_j'\right)}\right\}_{j\in[n]} \\ \text{sk: } h_2^r, \left\{h_2^{r_j\widehat{u}_j}, h_2^{r_j}, h_2^{r_j\widehat{u}_j{}'}, h_2^{r_j'}\right\}_{j\in[n]} \end{array}\right\}. \tag{A.21}$$

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

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

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

That is, if we can determine $\left\{T_j, T_j'\right\}_{j\in[n]}$, then the $\mathrm{DDH}_{p_2}^{G_N}$ 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, Istanbul, Turkey, April 2008.

[3] J. Herranz, F. Laguillaumie, and C. Ràfols, "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, Decemeber 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-EUROCRYPT 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 and fully secure HIBE with short ciphertexts," in *Proceedings of TCC 2010*, vol. 5978, pp. 455–479, Springer, Zurich, Switzerland, February 2010.

[13] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of TCC 2005*, vol. 3378, pp. 325–341, Springer, Cambridge, MA, USA, February 2005.