Hierarchical Predicate Encryption for Inner-Products

Tatsuaki Okamoto[1] and Katsuyuki Takashima[2]

[1] NTT, 3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
okamoto.tatsuaki@lab.ntt.co.jp
[2] Mitsubishi Electric, 5-1-1, Ofuna, Kamakura, Kanagawa, 247-8501 Japan
Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp

Abstract. This paper presents a hierarchical predicate encryption (HPE) scheme for inner-product predicates that is secure (selectively attribute-hiding) in the standard model under new assumptions. These assumptions are non-interactive and of fixed size in the number of adversary's queries (i.e., not "q-type"), and are proven to hold in the generic model. To the best of our knowledge, this is the first HPE (or delegatable PE) scheme for inner-product predicates that is secure in the standard model. The underlying techniques of our result are based on a new approach on bilinear pairings, which is extended from bilinear pairing groups over linear spaces. They are quite different from the existing techniques and may be of independent interest.

1 Introduction

1.1 Background

The notion of *predicate encryption* (PE) was explicitly presented by Katz, Sahai and Waters [16] as a generalized (fine-grained) notion of encryption that covers identity-based encryption (IBE) [2,3,5,9,10,15], hidden-vector encryption (HVE) [7] and attribute-based encryption (ABE) [1,13,19,20,21].

Informally, secret keys in a predicate encryption scheme correspond to predicates in some class $\mathcal{F}$, and a sender associates a ciphertext with an attribute in a set Σ; a ciphertext associated with the attribute $I \in \Sigma$ can be decrypted by secret key sk_f corresponding to the predicate $f \in \mathcal{F}$ if and only if $f(I) = 1$.

In addition, a stronger security notion for PE, *attribute-hiding*, than basic security requirement, *payload-hiding*, was defined in [16]. Roughly speaking, attribute-hiding requires that a ciphertext conceal the associated attribute as well as the plaintext, while payload-hiding only requires that a ciphertext conceal the plaintext. If attributes are identities, i.e., PE is IBE, attribute hiding PE implies *anonymous* IBE.

Katz, Sahai and Waters [16] also presented a concrete construction of PE for a class of predicates called *inner-product* predicates, which represents a wide class of predicates that includes an equality test (for IBE and HVE), disjunctions or conjunctions of equality tests, and, more generally, arbitrary CNF or

M. Matsui (Ed.): ASIACRYPT 2009, LNCS 5912, pp. 214–231, 2009.

DNF formulas (for ABE). Informally, an attribute of inner-product predicates is expressed as vector $\overrightarrow{x}$ and predicate $f_{\overrightarrow{v}}$ is associated with vector $\overrightarrow{v}$, where $f_{\overrightarrow{v}}(\overrightarrow{x}) = 1$ iff $\overrightarrow{x} \cdot \overrightarrow{v} = 0$. (Here, $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the standard inner-product.)

Although the Katz-Sahai-Waters scheme [16] is the most expressive attribute-hiding PE among the existing schemes, no delegation functionality was proposed. Shi and Waters [22] presented a delegation mechanism for a class of PE, but the admissible predicates of the system, which is a class of equality tests for HVE, are more limited than inner-product predicates in [16]. Okamoto and Takashima [18] presented hierarchical delegation of PE for inner-product predicates, but the security proof was only given in the generic model.

1.2 Our Results

This paper addresses the above problems in [16,22,18].

- This paper proposes a *hierarchical* predicate encryption (HPE) scheme for *inner-product* predicates, where a (natural) hierarchical delegation system of inner-product predicates is provided e.g., our hierarchical system is consistent with that for hierarchical IBE (HIBE) [4,8,11,12] (i.e., our HPE is specialized to anonymous HIBE, if the predicate of HPE is specified to the equality test of identities).
- The proposed HPE scheme is selectively attribute-hiding against chosen-plaintext-attacks (CPA) in the standard model under two new assumptions, the RDSP and IDSP assumptions. These assumptions are non-interactive, falsifiable and of fixed size in the number of adversary's queries (i.e., not "q-type"), and are proven to hold in the generic model.
- To achieve the result, this paper advances an approach recently developed in [17,18]. This approach is extended from bilinear pairing groups into higher dimensional vector spaces, and a notion, *dual pairing vector spaces* (DPVS), is employed in this paper. (We will explain this approach below.)

  One of the most basic decisional assumptions in this approach is the decisional subspace problem (DSP) assumption. (It is a higher-dimensional generalization of the decisional DH and Linear assumptions, and the relationships of this assumption with the traditional ones are studied in [17].)

  The assumptions introduced in this paper, the RDSP and IDSP assumptions, are variants of the DSP assumption in DPVS.
- The performance of the proposed HPE scheme is almost the same as (or slightly worse than) that in [18], where the dimension of DPVS for our HPE scheme is $n + 3$, whereas that for [18] is $n + 2$, when n is the dimension of predicate/attribute vectors.
- Since HPE is a generalized (fine-grained) version of anonymous HIBE (AHIBE) (or includes AHIBE as a special case), HPE covers (a generalized version of) applications described in [8], fully private communication and search on encrypted data. For example, we can use a two-level HPE scheme where the first level corresponds to the predicate/attribute of (single-layer) PE and the second level corresponds to those of "attribute search by a predicate" (generalized "key-word search").

1.3 A New Approach – Dual Pairing Vector Spaces

We now explain how the approach works by using a typical construction example on direct products of pairing groups $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, g_T, e)$, where q is a prime, $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of order q, g_i is a generator of $\mathbb{G}_i$ $(i = 1, 2)$, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear pairing operation, and $g_T := e(g_1, g_2) \neq 1$. Here we denote the group operation of $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ by multiplication. Note that this construction also works on *symmetric* pairing groups, where $\mathbb{G}_1 = \mathbb{G}_2$. As for the definitions of some notations, see Section 1.5.

Vector spaces $\mathbb{V}$ and $\mathbb{V}^*$: $\mathbb{V} := \overbrace{\mathbb{G}_1 \times \cdots \times \mathbb{G}_1}^{N}$ and $\mathbb{V}^* := \overbrace{\mathbb{G}_2 \times \cdots \times \mathbb{G}_2}^{N}$, whose elements are expressed by N-dimensional vectors, $\boldsymbol{x} := (g_1^{x_1}, \ldots, g_1^{x_N})$ and $\boldsymbol{y} := (g_2^{y_1}, \ldots, g_2^{y_N})$, respectively $(x_i, y_i \in \mathbb{F}_q$ for $i = 1, \ldots, N)$.

Canonical bases $\mathbb{A}$ and $\mathbb{A}^*$: $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ of $\mathbb{V}$, where $\boldsymbol{a}_1 := (g_1, 1, \ldots, 1)$, $\boldsymbol{a}_2 := (1, g_1, 1, \ldots, 1), \ldots, \boldsymbol{a}_N := (1, \ldots, 1, g_1)$. $\mathbb{A}^* := (\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_N^*)$ of $\mathbb{V}^*$, where $\boldsymbol{a}_1^* := (g_2, 1, \ldots, 1)$, $\boldsymbol{a}_2^* := (1, g_2, 1, \ldots, 1), \ldots, \boldsymbol{a}_N^* := (1, \ldots, 1, g_2)$.

Pairing operation: $e(\boldsymbol{x}, \boldsymbol{y}) := \prod_{i=1}^{N} e(g_1^{x_i}, g_2^{y_i}) = e(g_1, g_2)^{\sum_{i=1}^{N} x_i y_i} = g_T^{\vec{x} \cdot \vec{y}} \in \mathbb{G}_T$ for the above $\boldsymbol{x} \in \mathbb{V}$ and $\boldsymbol{y} \in \mathbb{V}^*$.

Base change: Canonical basis $\mathbb{A}$ is changed to basis $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$ of $\mathbb{V}$ using a uniformly chosen (regular) linear transformation, $X := (\chi_{i,j}) \xleftarrow{\cup} GL(N, \mathbb{F}_q)$, such that $\boldsymbol{b}_i = \sum_{j=1}^{N} \chi_{i,j} \boldsymbol{a}_j$, $(i = 1, \ldots, N)$. $\mathbb{A}^*$ is also changed to basis $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$ of $\mathbb{V}^*$, such that $(\vartheta_{i,j}) := (X^T)^{-1}$, $\boldsymbol{b}_i^* = \sum_{j=1}^{N} \vartheta_{i,j} \boldsymbol{a}_j^*$, $(i = 1, \ldots, N)$. We see that $e(\boldsymbol{b}_i, \boldsymbol{b}_j^*) = g_T^{\delta_{i,j}}$, $(\delta_{i,j} = 1$ if $i = j$, and $\delta_{i,j} = 0$ if $i \neq j)$ i.e., $\mathbb{B}$ and $\mathbb{B}^*$ are dual orthonormal bases of $\mathbb{V}$ and $\mathbb{V}^*$.

Intractable Problem: One of the most natural *decisional* problems in our approach is the *decisional subspace problem* (DSP) [17]. The $\mathrm{DSP}_{(N_1, N_2)}$ assumption is: it is hard to tell $\boldsymbol{v} := v_{N_2+1} \boldsymbol{b}_{N_2+1} + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$ from $\boldsymbol{u} := v_1 \boldsymbol{b}_1 + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$, where $(v_1, \ldots, v_{N_1}) \xleftarrow{\cup} \mathbb{F}_q^{N_1}$ and $N_2 + 1 < N_1$. DSP is intractable if the generalized DDH or DLIN problem is intractable [17].

Trapdoor: Although the DSP problem is assumed to be intractable, it can be efficiently solved by using *trapdoor* $\boldsymbol{t}^* \in \mathsf{span}\langle \boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{N_2}^* \rangle$. Given $\boldsymbol{v} := v_{N_2+1} \boldsymbol{b}_{N_2+1} + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$ or $\boldsymbol{u} := v_1 \boldsymbol{b}_1 + \cdots + v_{N_1} \boldsymbol{b}_{N_1}$, we can tell $\boldsymbol{v}$ from $\boldsymbol{u}$ using $\boldsymbol{t}^*$ since $e(\boldsymbol{v}, \boldsymbol{t}^*) = 1$ and $e(\boldsymbol{u}, \boldsymbol{t}^*) \neq 1$ with high probability.

1.4 Related Works on Our Approach

Higher dimensional vector treatment of bilinear pairing groups have been already employed in the literature especially in the areas of IBE, ABE and BE (e.g., [4,1,6,8,13,14,21]). For example, in a typical vector treatment, two vector forms of $P := (g_1^{x_1}, \ldots, g_1^{x_n})$ and $Q := (g_2^{y_1}, \ldots, g_2^{y_n})$ are set and pairing for P and Q is operated as $e(P, Q) := \prod_{i=1}^{n} e(g_1^{x_i}, g_2^{y_i})$. Such a treatment can be rephrased in our approach using the (symmetric pairing) notations shown in Section 1.3 such that $P = x_1 \boldsymbol{a}_1 + \cdots + x_n \boldsymbol{a}_n$ and $Q = y_1 \boldsymbol{a}_1^* + \cdots + y_n \boldsymbol{a}_n^*$ over canonical basis $\mathbb{A}$ and $\mathbb{A}^*$.

The major drawback of this approach is the easily *decomposable* property over $\mathbb{A}$ (and $\mathbb{A}^*$). That is, it is easy to decompose $x_i \boldsymbol{a}_i = (1, \ldots, 1, g_1^{x_i}, 1, \ldots, 1)$ from $P := x_1 \boldsymbol{a}_1 + \cdots x_n \boldsymbol{a}_n = (g_1^{x_1}, \ldots, g_1^{x_n})$.

In contrast, the current approach employs basis $\mathbb{B}$ that is linearly transformed from $\mathbb{A}$ using a secret random matrix $X \in \mathbb{F}_q^{n \times n}$. A remarkable property over $\mathbb{B}$ is that it seems hard to decompose $x_i \boldsymbol{b}_i$ from $P' := x_1 \boldsymbol{b}_1 + \cdots x_n \boldsymbol{b}_n$. In addition, the dual orthonormal basis $\mathbb{B}^*$ of $\mathbb{V}^*$ can be used as a source of the trapdoors to the decomposability (see Section 1.3) through the pairing operation over $\mathbb{B}$ and $\mathbb{B}^*$. The hard decomposability and its trapdoors are the key trick in this paper. Note that composite order pairing groups are often employed with similar tricks, hard decomposability of a composite order group into the prime order subgroups and its trapdoors through factoring (e.g., [16,22]).

1.5 Notations

When A is a random variable or distribution, $y \xleftarrow{\mathsf{R}} A$ denotes that y is randomly selected from A according to its distribution. When A is a set, $y \xleftarrow{\mathsf{U}} A$ denotes that y is uniformly selected from A. $y := z$ denotes that y is set, defined or substituted by z. When a is a fixed value, $A(x) \to a$ (e.g., $A(x) \to 1$) denotes the event that machine (algorithm) A outputs a on input x. A function $f : \mathbb{N} \to \mathbb{R}$ is *negligible* in λ, if for every constant $c > 0$, there exists an integer n such that $f(\lambda) < \lambda^{-c}$ for all $\lambda > n$.

We denote the finite field of order q by $\mathbb{F}_q$. A vector symbol denotes a vector representation over $\mathbb{F}_q$, e.g., $\overrightarrow{x}$ denotes $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$. $\overrightarrow{x} \cdot \overrightarrow{v}$ denotes the inner-product $\sum_{i=1}^n x_i v_i$ of two vectors $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{v} = (v_1, \ldots, v_n)$. X^{T} denotes the transpose of matrix X. A bold face letter denotes an element of vector space $\mathbb{V}$ (resp. $\mathbb{V}^*$), e.g., $\boldsymbol{x} \in \mathbb{V}$ (resp. $\boldsymbol{x}^* \in \mathbb{V}^*$). $\mathsf{span}\langle \boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \rangle$ (resp. $\mathsf{span}\langle \overrightarrow{x}_1, \ldots, \overrightarrow{x}_n \rangle$) denotes the subspace generated by $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ (resp. $\overrightarrow{x}_1, \ldots, \overrightarrow{x}_n$).

2 Dual Pairing Vector Spaces

Definition 1. *"Dual pairing vector spaces (DPVS)"* $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*)$ *are a tuple of a prime* q*, two* N*-dimensional vector spaces* $\mathbb{V}$ *and* $\mathbb{V}^*$ *over* $\mathbb{F}_q$*, a cyclic group* $\mathbb{G}_T$ *of order* q*, and their canonical bases i.e.,* $\mathbb{A} := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_N)$ *of* $\mathbb{V}$ *and* $\mathbb{A}^* := (\boldsymbol{a}_1^*, \ldots, \boldsymbol{a}_N^*)$ *of* $\mathbb{V}^*$ *that satisfy the following conditions:*

1. *[Non-degenerate bilinear pairing] There exists a polynomial-time computable nondegenerate bilinear pairing* $e : \mathbb{V} \times \mathbb{V}^* \to \mathbb{G}_T$ *i.e.,* $e(s\boldsymbol{x}, t\boldsymbol{y}) = e(\boldsymbol{x}, \boldsymbol{y})^{st}$ *and if* $e(\boldsymbol{x}, \boldsymbol{y}) = 1$ *for all* $\boldsymbol{y} \in \mathbb{V}$*, then* $\boldsymbol{x} = \boldsymbol{0}$*.*
2. *[Dual orthonormal bases]* $\mathbb{A}$*,* $\mathbb{A}^*$*, and* e *satisfy* $e(\boldsymbol{a}_i, \boldsymbol{a}_j^*) = g_T^{\delta_{i,j}}$ *for all* i *and* j*, where* $\delta_{i,j} = 1$ *if* $i = j$*, and* 0 *otherwise, and* $g_T \neq 1 \in \mathbb{G}_T$*.*
3. *[Distortion maps] Endomorphisms* $\phi_{i,j}$ *of* $\mathbb{V}$ *s.t.* $\phi_{i,j}(\boldsymbol{a}_j) = \boldsymbol{a}_i$ *and* $\phi_{i,j}(\boldsymbol{a}_k) = \boldsymbol{0}$ *if* $k \neq j$ *are polynomial-time computable. Moreover, endomorphisms* $\phi_{i,j}^*$ *of* $\mathbb{V}^*$ *s.t.* $\phi_{i,j}^*(\boldsymbol{a}_j^*) = \boldsymbol{a}_i^*$ *and* $\phi_{i,j}^*(\boldsymbol{a}_k^*) = \boldsymbol{0}$ *if* $k \neq j$ *are also polynomial-time computable. We call* $\phi_{i,j}$ *and* $\phi_{i,j}^*$ *"distortion maps".*

Three typical constructions are given in [17]; a product of bilinear pairing groups, or a Jacobian variety of a supersingular curve of genus ≥ 1 [23]. See Section 1.3 as well (where the description of distortion maps is omitted).

3 Assumptions

This section defines two variants of the DSP assumption, the RDSP and IDSP assumptions. An intuition behind these assumptions are given in Remark below.

DPVS generation algorithm $\mathcal{G}_{\mathsf{dpvs}}$ takes input 1^λ ($\lambda \in \mathbb{N}$) and $N \in \mathbb{N}$, and outputs a description of $\mathsf{param} := (q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*)$ with security parameter λ and N-dimensional $\mathbb{V}$ and $\mathbb{V}^*$. It can be constructed in a manner shown in [17]. We describe a random orthonormal basis generator $\mathcal{G}_{\mathsf{ob}}$ below, which is used as a subroutine in the RDSP and IDSP instance generators.

$\mathcal{G}_{\mathsf{ob}}(1^\lambda, N)$: $\mathsf{param} := (q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N)$,
$X := (\chi_{i,j}) \xleftarrow{\mathsf{U}} GL(N, \mathbb{F}_q)$, $(\vartheta_{i,j}) := (X^{\mathrm{T}})^{-1}$,
$\boldsymbol{b}_i := \sum_{j=1}^N \chi_{i,j} \boldsymbol{a}_j$, $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_N)$, $\boldsymbol{b}_i^* := \sum_{j=1}^N \vartheta_{i,j} \boldsymbol{a}_j^*$, $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_N^*)$,
return $(\mathsf{param}, \mathbb{B}, \mathbb{B}^*)$

We now define the RDSP and IDSP instance generators, $\mathcal{G}_\beta^{\mathsf{RDSP}}$ and $\mathcal{G}_\beta^{\mathsf{IDSP}}$.

$\mathcal{G}_\beta^{\mathsf{RDSP}}(1^\lambda, n)$: $(\mathsf{param}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, n+3)$, $\overrightarrow{y} := (y_1, \ldots, y_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$,
$\delta_1, \delta_2, \zeta_1, \zeta_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}$, $\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3})$,
$(\omega^{(k)}, \gamma_1^{(k)}, \gamma_2^{(k)})_{k=1,2,3} \xleftarrow{\mathsf{U}} GL(\mathbb{F}_q, 3)$,
For $i = 1, \ldots, n$; $k = 1, 2, 3$;
$\boldsymbol{h}_i^{(k)*} := \omega^{(k)} \boldsymbol{b}_i^* + \gamma_1^{(k)} y_i \boldsymbol{b}_{n+1}^* + \gamma_2^{(k)} y_i \boldsymbol{b}_{n+2}^*$, $\tau_i^{(k)} := (\gamma_1^{(k)} + \gamma_2^{(k)}) y_i$,
$\boldsymbol{e}_0 := \delta_1 (\sum_{i=1}^n y_i \boldsymbol{b}_i) + \delta_2 \boldsymbol{b}_{n+3}$,
$\boldsymbol{e}_1 := \delta_1 (\sum_{i=1}^n y_i \boldsymbol{b}_i) + \zeta_1 \boldsymbol{b}_{n+1} + \zeta_2 \boldsymbol{b}_{n+2} + \delta_2 \boldsymbol{b}_{n+3}$,
return $(\mathsf{param}, \widehat{\mathbb{B}}, \{\boldsymbol{h}_i^{(k)*}, \tau_i^{(k)}\}_{i=1,\ldots,n;k=1,2,3}, \overrightarrow{y}, \boldsymbol{e}_\beta)$.

$\mathcal{G}_\beta^{\mathsf{IDSP}}(1^\lambda, n)$: $(\mathsf{param}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, n+3)$,
$\overrightarrow{y} := (y_1, \ldots, y_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$, $\overrightarrow{u} := (u_1, \ldots, u_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$,
$\delta_1, \delta_2, \zeta_1, \zeta_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}$, $\widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3})$,
For $i = 1, \ldots, n$; $(\omega^{(k)}, \gamma_{i,1}^{(k)}, \gamma_{i,2}^{(k)})_{k=1,2,3} \xleftarrow{\mathsf{U}} GL(\mathbb{F}_q, 3)$,
For $i = 1, \ldots, n$; $k = 1, 2, 3$;
$\boldsymbol{h}_i^{(k)*} := \omega^{(k)} \boldsymbol{b}_i^* + \gamma_{i,1}^{(k)} \boldsymbol{b}_{n+1}^* + \gamma_{i,2}^{(k)} \boldsymbol{b}_{n+2}^*$, $\tau_i^{(k)} := \gamma_{i,1}^{(k)} + \gamma_{i,2}^{(k)}$,
$\boldsymbol{e}_0 := \delta_1 (\sum_{i=1}^n y_i \boldsymbol{b}_i) + \zeta_1 \boldsymbol{b}_{n+1} + \zeta_2 \boldsymbol{b}_{n+2} + \delta_2 \boldsymbol{b}_{n+3}$,
$\boldsymbol{e}_1 := \delta_1 (\sum_{i=1}^n u_i \boldsymbol{b}_i) + \zeta_1 \boldsymbol{b}_{n+1} + \zeta_2 \boldsymbol{b}_{n+2} + \delta_2 \boldsymbol{b}_{n+3}$,
return $(\mathsf{param}, \widehat{\mathbb{B}}, \{\boldsymbol{h}_i^{(k)*}, \tau_i^{(k)}\}_{i=1,\ldots,n;k=1,2,3}, \overrightarrow{y}, \boldsymbol{e}_\beta)$.

Definition 2 (RDSP: Decisional Subspace Problem with Relevant Dual Vector Tuples). *For all security parameter $\lambda \in \mathbb{N}$, we define RDSP advantage of a probabilistic machine $\mathcal{B}$ as follows:*

$$\mathsf{Adv}^{\mathsf{RDSP}}_{\mathcal{B}}(\lambda) := \left| \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \rho) \to 1 \;\middle|\; \rho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{RDSP}}_0(1^\lambda, n)\right] - \mathsf{Pr}\left[\mathcal{B}(1^\lambda, \rho) \to 1 \;\middle|\; \rho \xleftarrow{\mathsf{R}} \mathcal{G}^{\mathsf{RDSP}}_1(1^\lambda, n)\right]\right|.$$

The RDSP assumption is: for any probabilistic polynomial-time adversary $\mathcal{B}$, $\mathsf{Adv}^{\mathsf{RDSP}}_{\mathcal{B}}(\lambda)$ is negligible in λ.

Definition 3 (IDSP: Decisional Subspace Problem with Irrelevant Dual Vector Tuples). *The IDSP advantage of $\mathcal{B}$, $\mathsf{Adv}^{\mathsf{IDSP}}_{\mathcal{B}}(\lambda)$, and the IDSP assumption are defined similarly as in Definition 2.*

In the generic DPVS model, basic operations in $\mathbb{V}, \mathbb{V}^*$, and $\mathbb{G}_T$, i.e., vector additions in $\mathbb{V}$ and $\mathbb{V}^*$, multiplication in $\mathbb{G}_T$, pairing, and distortion maps w.r.t. $\mathbb{A}$ or $\mathbb{A}^*$, are given by "generic" algorithms that act independently of the representations of vectors or group elements.

Theorem 1. *The advantages $\mathsf{Adv}^{\mathsf{RDSP}}_{\mathcal{B}}(\lambda)$ and $\mathsf{Adv}^{\mathsf{IDSP}}_{\mathcal{B}}(\lambda)$ are $O(d/2^\lambda)$ for any adversary $\mathcal{B}$ in the generic DPVS model, where d is the maximum of the degrees of polynomials of formal variables (in the generic model game).*

We will describe the proof of Theorem 1 in the full version of this paper.

Remark (Intuition behind the Assumptions)
Here we informally explain the RDSP assumption by using a simplified one. In the simplified RDSP assumption, $(\boldsymbol{h}^*_1, \ldots, \boldsymbol{h}^*_n)$ is given to $\mathcal{A}$ in addition to $(\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n+2}), \overrightarrow{y} := (y_1, \ldots, y_n), \boldsymbol{e}_\beta)$, such that $\boldsymbol{h}^*_i := \omega \boldsymbol{b}^*_i + y_i \boldsymbol{b}^*_{n+1}$ $(i = 1, \ldots, n;\ \omega \xleftarrow{\mathsf{U}} \mathbb{F}_q)$ and $\boldsymbol{e}_\beta := \delta_1(\sum_{i=1}^n y_i \boldsymbol{b}_i) + \beta\zeta \boldsymbol{b}_{n+1} + \delta_2 \boldsymbol{b}_{n+2}$ $(\beta \xleftarrow{\mathsf{U}} \{0,1\}$, $\delta_1, \delta_1, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q)$. The simplified RDSP assumption is that it is hard for any adversary $\mathcal{A}$, given $(\mathbb{B}, \overrightarrow{y}, \boldsymbol{e}_\beta)$ along with $(\boldsymbol{h}^*_1, \ldots, \boldsymbol{h}^*_n)$, to correctly guess β. (In the DSP assumption, only $(\mathbb{B}, \overrightarrow{y}, \boldsymbol{e}_\beta)$ is given to $\mathcal{A}$.)

$(\boldsymbol{h}^*_1, \ldots, \boldsymbol{h}^*_n)$ is added in the RDSP assumption in order to simulate the key generation oracle in the security proof of our encryption scheme as follows: for any $\overrightarrow{v} := (v_1, \ldots, v_n)$ with $\overrightarrow{v} \cdot \overrightarrow{y} \neq 0$, the simulator can compute a secret key $\boldsymbol{k}^*$ for $\overrightarrow{v}$ such that $\boldsymbol{k}^* := \frac{1}{\overrightarrow{v} \cdot \overrightarrow{y}} \sum_{i=1}^n v_i \boldsymbol{h}_i = \frac{\omega}{\overrightarrow{v} \cdot \overrightarrow{y}} (\sum_{i=1}^n v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{n+1} = \omega' (\sum_{i=1}^n v_i \boldsymbol{b}^*_i) + \boldsymbol{b}^*_{n+1}$ where $\omega' := \frac{\omega}{\overrightarrow{v} \cdot \overrightarrow{y}}$.

This secret key generation procedure, however, does not work for $\overrightarrow{v}$ with $\overrightarrow{v} \cdot \overrightarrow{y} = 0$, since $\frac{1}{\overrightarrow{v} \cdot \overrightarrow{y}}$ cannot be computed. Therefore, $(\boldsymbol{h}^*_1, \ldots, \boldsymbol{h}^*_n)$ does not seem helpful to break the RDSP assumption, since a secret-key $\boldsymbol{k}^*$ for $\overrightarrow{v}$ with "$\overrightarrow{v} \cdot \overrightarrow{y} = 0$" is of use to guess β by checking whether $e(\boldsymbol{e}_\beta, \boldsymbol{k}^*) = 1$ or not. Hence, the RDSP assumption seems to hold if the DSP assumption does.

Similarly the IDSP assumption is introduced as a variant of the DSP assumption. In the RDSP and IDSP assumptions employed in this paper, we use a public element $\boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}$ (in place of $\boldsymbol{b}_{n+1}$ in basis $\mathbb{B}$ in the simplified

one), and $\boldsymbol{b}_{n+1}$ and $\boldsymbol{b}_{n+2}$ are not published. Such a modification is required for the IDSP assumption since the simplified IDSP assumption does not hold.

In addition, in our RDSP (and IDSP) assumption, $\{\boldsymbol{h}_i^{(k)*}\}_{i=1,\ldots,n;\ k=1,2,3}$ is employed in place of $\{\boldsymbol{h}_i^*\}_{i=1,\ldots,n}$. This modification is introduced to *re-randomize* the coefficients for each key generation of the simulation by a random linear combination of $\boldsymbol{h}_i^{(1)*}$, $\boldsymbol{h}_i^{(2)*}$ and $\boldsymbol{h}_i^{(3)*}$.

4 Definition of Hierarchical Predicate Encryption (HPE)

This section defines hierarchical predicate encryption (HPE) for the class of hierarchical inner-product predicates and its security.[1]

In a delegation system, it is required that a user who has a capability can delegate to another user a more restrictive capability. In addition to this requirement, our hierarchical inner-product encryption introduces a format of hierarchy $\overrightarrow{\mu}$ to define common delegation structure in a system.

We call a tuple of positive integers $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ s.t. $\mu_0 = 0 < \mu_1 < \mu_2 < \cdots < \mu_d = n$ a format of hierarchy of depth d attribute spaces. Let Σ_ℓ $(\ell = 1, \ldots, d)$ be the sets of attributes, where each $\Sigma_\ell := \mathbb{F}_q^{\mu_\ell - \mu_{\ell-1}} \setminus \{\overrightarrow{0}\}$. Let the hierarchical attributes $\Sigma := \cup_{\ell=1}^{d}(\Sigma_1 \times \ldots \times \Sigma_\ell)$, where the union is a disjoint union. Then, for $\overrightarrow{v}_i \in \mathbb{F}_q^{\mu_i - \mu_{i-1}} \setminus \{\overrightarrow{0}\}$, the hierarchical predicate $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$ on hierarchical attributes $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) \in \Sigma$ is defined as follows: $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) = 1$ iff $\ell \leq h$ and $\overrightarrow{x}_i \cdot \overrightarrow{v}_i = 0$ for all i s.t. $1 \leq i \leq \ell$.

Let the space of hierarchical predicates $\mathcal{F} := \{f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)} \mid \overrightarrow{v}_i \in \mathbb{F}_q^{\mu_i - \mu_{i-1}} \setminus \{\overrightarrow{0}\}\}$. We call h (resp. ℓ) the level of $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h)$ (resp. $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$).

Definition 4. *Let $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)$ s.t. $\mu_0 = 0 < \mu_1 < \mu_2 < \cdots < \mu_d = n$ be a format of hierarchy of depth d attribute spaces. A hierarchical predicate encryption (HPE) scheme for the class of hierarchical inner-product predicates $\mathcal{F}$ over the set of hierarchical attributes Σ consists of probabilistic polynomial-time algorithms* Setup, GenKey, Enc, Dec, *and* Delegate$_\ell$ *for $\ell = 1, \ldots, d-1$. They are given as follows:*

- Setup *takes as input security parameter 1^λ and format of hierarchy $\overrightarrow{\mu}$, and outputs (master) public key* pk *and (master) secret key* sk.
- GenKey *takes as input the master public key* pk*, secret key* sk*, and predicate vectors $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$. It outputs a corresponding secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$.
- Enc *takes as input the master public key* pk*, attribute vectors $(\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h)$, where $1 \leq h \leq d$, and plaintext m in some associated plaintext space,* msg. *It returns ciphertext c.*
- Dec *takes as input the master public key* pk*, secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$*, where $1 \leq \ell \leq d$, and ciphertext c. It outputs either plaintext m or the distinguished symbol $\perp$.*

[1] More general delegation structures (partial order structures) than tree hierarchical structures can be easily realized in our HPE scheme. See Remark in Section 5.

– $\mathsf{Delegate}_\ell$ *takes as input the master public key* pk*,* ℓ*-th level secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$*, and* $(\ell+1)$*-th level predicate vector* $\overrightarrow{v}_{\ell+1}$*. It returns* $(\ell+1)$*-th level secret key* $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_{\ell+1})}$*.*

A HPE scheme should have the following correctness property: for all correctly generated pk and $\mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}$, generate $c \xleftarrow{R} \mathsf{Enc}(\mathsf{pk}, m, (\overrightarrow{x}_1,\ldots,\overrightarrow{x}_h))$ and $m' := \mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}, c)$. If $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\overrightarrow{x}_1,\ldots,\overrightarrow{x}_h) = 1$, then $m' = m$. Otherwise, $m' \neq m$ except for negligible probability.

For f and f' in $\mathcal{F}$, we denote $f' \leq f$ if the predicate vector for f is a prefix of that for f'. For the following definition for key queries, see [22].

Remark: We will explain the hierarchical structure by using a small (toy) example that has three levels and each level consists of 2-dimensional space, i.e., 6-dimensional space is employed in total. That is, $\overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d) = (6, 3; 2, 4, 6)$ in this example.

A user who possesses a secret key sk_1 in the top level, associated with the top level predicate vector $\overrightarrow{v}_1 := (v_1, v_2)$, can delegate any value (say $\overrightarrow{v}_2 := (v_3, v_4)$) of the second level key sk_2 such that the predicate vector for sk_2 is $(\overrightarrow{v}_1, \overrightarrow{v}_2)$. Similarly, a user who possesses a secret key in the second level, sk_2 with $(\overrightarrow{v}_1, \overrightarrow{v}_2)$, can delegate any value (say $\overrightarrow{v}_3 := (v_5, v_6)$) of the third level key sk_3 with $(\overrightarrow{v}_1, \overrightarrow{v}_2, \overrightarrow{v}_3)$.

Secret key sk_1 with $\overrightarrow{v}_1$, can decrypt a ciphertext associated with attribute vector $(\overrightarrow{x}_1, (*,*), (*,*)) := ((x_1, x_2), (*,*), (*,*))$ if $\overrightarrow{x}_1 \cdot \overrightarrow{v}_1 = 0$, where $*$ denotes an arbitrary value. Secret key sk_2 with $(\overrightarrow{v}_1, \overrightarrow{v}_2)$ can decrypt a ciphertext with attribute vector $(\overrightarrow{x}_1, \overrightarrow{x}_2, (*,*))$ if $\overrightarrow{x}_1 \cdot \overrightarrow{v}_1 = 0$ and $\overrightarrow{x}_2 \cdot \overrightarrow{v}_2 = 0$. However sk_2 cannot decrypt a ciphertext with higher level (top level) attribute vector $\overrightarrow{x}_1 := (x_1, x_2)$ (or $(\overrightarrow{x}_1, (*,*), (*,*))$). Therefore, the capability of a delegated key sk_2 is more limited than the parent key sk_1.

Hence, when $(\overrightarrow{v}_1, \overrightarrow{v}_2) := ((v_1, v_2), (v_3, v_4))$ is a predicate vector for a secret key, $(\overrightarrow{v}_1, \overrightarrow{v}_2)$ is considered to be $(\overrightarrow{v}_1, \overrightarrow{v}_2, (0,0))$, and when $\overrightarrow{x}_1 := (x_1, x_2)$ is an attribute vector for a ciphertext, $\overrightarrow{x}_1$ is considered to be $(\overrightarrow{x}_1, (*,*), (*,*)))$, where $(*,*) \cdot (0,0) = 0$ and $(*,*) \cdot \overrightarrow{v}_2 \neq 0$ unless $\overrightarrow{v}_2 = (0,0)$.

Definition 5. *A hierarchical inner-product predicate encryption scheme for hierarchical predicates* $\mathcal{F}$ *over hierarchical attributes* Σ *is* selectively attribute-hiding (AH) against chosen plaintext attacks *if for all probabilistic polynomial-time adversaries* $\mathcal{A}$*, the advantage of* $\mathcal{A}$ *in the following experiment is negligible in the security parameter.*

1. $\mathcal{A}$ *outputs challenge attribute vectors* $\mathcal{X}^{(0)} := (\overrightarrow{x}_1^{(0)}, \ldots, \overrightarrow{x}_{h^{(0)}}^{(0)}), \mathcal{X}^{(1)} := (\overrightarrow{x}_1^{(1)}, \ldots, \overrightarrow{x}_{h^{(1)}}^{(1)})$.
2. Setup *is run to generate keys* pk *and* sk*, and* pk *is given to* $\mathcal{A}$*.*
3. $\mathcal{A}$ *may adaptively makes a polynomial number of queries of the following type:*
   – *[Create key]* $\mathcal{A}$ *asks the challenger to create a secret key for a predicate* $f \in \mathcal{F}$*. The challenger creates a key for* f *without giving it to* $\mathcal{A}$*.*

- *[Create delegated key]* $\mathcal{A}$ *specifies a key for predicate* f *that has already been created, and asks the challenger to perform a delegation operation to create a child key for* $f' \leq f$. *The challenger computes the child key without giving it to the adversary.*
- *[Reveal key]* $\mathcal{A}$ *asks the challenger to reveal an already-created key for predicate* f *s.t.* $f(\mathcal{X}^{(0)}) = f(\mathcal{X}^{(1)}) = 0$.

Note that when key creation requests are made, $\mathcal{A}$ *does not automatically see the created key.* $\mathcal{A}$ *sees a key only when it makes a reveal key query.*

4. $\mathcal{A}$ *outputs challenge plaintexts* $m^{(0)}, m^{(1)}$.
5. *A random bit* b *is chosen.* $\mathcal{A}$ *is given* $c^{(b)} \xleftarrow{R} \mathsf{Enc}(\mathsf{pk}, m^{(b)}, \mathcal{X}^{(b)})$.
6. *The adversary may continue to request keys for additional predicate vectors subject to the restrictions given in step 3.*
7. $\mathcal{A}$ *outputs a bit* b', *and succeeds if* $b' = b$.

We define the advantage of $\mathcal{A}$ *as the quantity* $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HPE,AH}}(\lambda) := |\mathsf{Pr}[b' = b] - 1/2|$.

Remark: In Definition 5, adversary $\mathcal{A}$ is not allowed to ask a key-query for $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$ such that $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\mathcal{X}^{(b)}) = 1$ for some $b \in \{0,1\}$, while in the security definition in [16], such a key-query is allowed provided that $m^{(0)} = m^{(1)}$ and $f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\mathcal{X}^{(0)}) = f_{(\overrightarrow{v}_1,\ldots,\overrightarrow{v}_\ell)}(\mathcal{X}^{(1)}) = 1$. This restriction is introduced to prove the security of the proposed HPE scheme only under the RDSP and IDSP assumptions. If we introduce another variant of the assumptions, we can relax this restriction. We will describe this case in the full version of this paper.

5 The Proposed HPE Scheme

5.1 Key Idea in Constructing the Proposed HPE

We will explain a key idea of the proposed HPE scheme.

First, as a special (1-level) case of the proposed construction of HPE, we will show a predicate encryption (PE) construction for the inner-product predicate. Through the orthonormal property of (random) dual bases ($\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_{n+3})$, $\mathbb{B}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_{n+3}^*)$) in DPVS, $(q, \mathbb{V}, \mathbb{V}^*, \mathbb{G}_T, \mathbb{A}, \mathbb{A}^*)$, (Sections 1.3, 2 and 3), the PE scheme for the (n-dimensional) inner-product predicate can be constructed as below, where $\mathbb{V}$ and $\mathbb{V}^*$ are $(n+3)$-dimensional spaces, the public parameter is $(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}, \boldsymbol{b}_{n+3})$ as well as the parameters of DPVS, and the master secret key is (X and) $\mathbb{B}^*$. Ciphertext $(\boldsymbol{c}_1, c_2)$ for attribute $\overrightarrow{x} := (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and plaintext $m \in \mathbb{G}_T$ is $\boldsymbol{c}_1 := \delta_1(x_1\boldsymbol{b}_1 + \cdots + x_n\boldsymbol{b}_n) + \zeta\boldsymbol{d}_{n+1} + \delta_2\boldsymbol{b}_{n+3}$ and $c_2 := g_T^\zeta m$, where $\delta_1, \delta_2, \zeta \xleftarrow{U} \mathbb{F}_q$. Secret key $\boldsymbol{k}^*$ with predicate $\overrightarrow{v} := (v_1, \ldots, v_n) \in \mathbb{F}_q^n$ is $\boldsymbol{k}^* := \sigma(v_1\boldsymbol{b}_1^* + \cdots + v_n\boldsymbol{b}_n^*) + \eta\boldsymbol{b}_{n+1}^* + (1-\eta)\boldsymbol{b}_{n+2}^*$, where $\sigma, \eta \xleftarrow{U} \mathbb{F}_q$. If $\overrightarrow{x} \cdot \overrightarrow{v} = 0$, plaintext m can be computed by $m = c_2/e(\boldsymbol{c}_1, \boldsymbol{k}^*)$, since $e(\boldsymbol{c}_1, \boldsymbol{k}^*) = (\prod_{i=1}^n e(\delta_1 x_i \boldsymbol{b}_i, \sigma v_i \boldsymbol{b}_i^*)) \cdot e(\zeta\boldsymbol{b}_{n+1}, \eta\boldsymbol{b}_{n+1}^*) \cdot e(\zeta\boldsymbol{b}_{n+2}, (1-\eta)\boldsymbol{b}_{n+2}^*) = g_T^{\delta_1\sigma(\sum_{i=1}^n x_i v_i) + \zeta\eta + \zeta(1-\eta)} = g_T^{\delta_1\sigma(\overrightarrow{x}\cdot\overrightarrow{v})+\zeta} = g_T^\zeta$.

We now explain the key idea of the proposed HPE scheme by using a small (toy) example. Let the dimension of (predicate/attribute) vectors be 6, in which

there are three levels and each level has 2-dimensions, $\mathbb{V}$ and $\mathbb{V}^*$ be 9-dimensional spaces, the public parameter be $\mathbb{B} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_6, \boldsymbol{d}_7, \boldsymbol{b}_9)$ as well as the parameters of DPVS, and the master secret key be (X and) $\mathbb{B}^* := (\boldsymbol{b}^*_1, \ldots, \boldsymbol{b}^*_9)$, where $\boldsymbol{d}_7 := \boldsymbol{b}_7 + \boldsymbol{b}_8$.

Ciphertext $(\boldsymbol{c}_1, c_2)$ for attribute $\overrightarrow{x} := (\overrightarrow{x}_1, \overrightarrow{x}_2, \overrightarrow{x}_3) := ((x_1, x_2), (x_3, x_4), (x_5, x_6)) \in \mathbb{F}_q^6$ and plaintext m is constructed as $\boldsymbol{c}_1 := \delta_1(x_1\boldsymbol{b}_1 + x_2\boldsymbol{b}_2) + \cdots + \delta_3(x_5\boldsymbol{b}_5 + x_6\boldsymbol{b}_6) + \zeta\boldsymbol{d}_7 + \delta_4\boldsymbol{b}_8$ and $c_2 := g_T^{\zeta} m$, where $\delta_1, \ldots, \delta_4, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$. If the attribute is a higher level such as $\overrightarrow{x}_1 := (x_1, x_2)$, generate a modified attribute $\overrightarrow{x}^+ := ((x_1, x_2), (x_3^+, x_4^+), (x_5^+, x_6^+))$, where $(x_3^+, x_4^+, x_5^+, x_6^+) \xleftarrow{\mathsf{U}} \mathbb{F}_q^4$. Then, ciphertext $\boldsymbol{c}_1$ for attribute $\overrightarrow{x}_1$ is computed as ciphertext $\boldsymbol{c}_1$ for the modified attribute $\overrightarrow{x}^+$.

Top level secret key $\overrightarrow{\boldsymbol{k}}^*_1 := (\boldsymbol{k}^*_{1,0}, \ldots, \boldsymbol{k}^*_{1,6})$, for predicate $\overrightarrow{v} := (v_1, v_2) \in \mathbb{F}_q^2$ consists of three parts, $\boldsymbol{k}^*_{1,0}$, $(\boldsymbol{k}^*_{1,1}, \boldsymbol{k}^*_{1,2})$ and $(\boldsymbol{k}^*_{1,3}, \ldots, \boldsymbol{k}^*_{1,6})$, where the first one is used for decryption of ciphertexts, the second one for re-randomization (of delegated key), and the last one for delegation. Each part is: $\boldsymbol{k}^*_{1,0} := \sigma_{1,0}(v_1\boldsymbol{b}^*_1 + v_2\boldsymbol{b}^*_2) + \eta_0\boldsymbol{b}^*_7 + (1-\eta_0)\boldsymbol{b}^*_8$, $\boldsymbol{k}^*_{1,j} := \sigma_{1,j}(v_1\boldsymbol{b}^*_1 + v_2\boldsymbol{b}^*_2) + \eta_j\boldsymbol{b}^*_7 - \eta_j\boldsymbol{b}^*_8$ $(j = 1, 2)$, and $\boldsymbol{k}^*_{1,j} := \sigma_{1,j}(v_1\boldsymbol{b}^*_1 + v_2\boldsymbol{b}^*_2) + \psi\boldsymbol{b}_j + \eta_j\boldsymbol{b}^*_7 - \eta_j\boldsymbol{b}^*_8$ $(j = 3, \ldots, 6)$, where $\sigma_{1,j}, \psi \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $j = 0, \ldots, 6$. The first one, $\boldsymbol{k}^*_{1,0}$, can decrypt ciphertext $(\boldsymbol{c}_1, c_2)$ by $c_2/e(\boldsymbol{c}_1, \boldsymbol{k}^*_{1,0})$, since $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{1,0}) = g_T^{\zeta}$ if an attribute of $\boldsymbol{c}_1$ is $((x_1, x_2), (*, *), (*, *))$ with $(x_1, x_2) \cdot (v_1, v_2) = 0$. To delegate a secret key for the 2nd level vector (v_3, v_4), $\sigma_{2,j}(v_3\boldsymbol{k}^*_{1,3} + v_4\boldsymbol{k}^*_{1,4})$ is added to $\boldsymbol{k}^*_{1,0}$ $(j = 0)$, $\boldsymbol{0}$ $(j = 1, 2, 3)$, and $\psi^+\boldsymbol{k}^*_{1,j}$ $(j = 5, 6)$. To re-randomize the coefficients of $(v_1\boldsymbol{b}^*_1 + v_2\boldsymbol{b}^*_2)$, $\boldsymbol{b}^*_7$ and $\boldsymbol{b}^*_8$ in the delegated key, $(\alpha_{j,1}\boldsymbol{k}^*_{1,1} + \alpha_{j,2}\boldsymbol{k}^*_{1,2})$ is also added. So, the delegated key (the second level key) $\overrightarrow{\boldsymbol{k}}^*_2 := (\boldsymbol{k}^*_{2,0}, \ldots, \boldsymbol{k}^*_{2,3}, \boldsymbol{k}^*_{2,5}, \boldsymbol{k}^*_{2,6})$, (where $\boldsymbol{k}^*_{2,0}$ is for decryption, $(\boldsymbol{k}^*_{2,1}, \ldots, \boldsymbol{k}^*_{2,3})$ for re-randomization, and $(\boldsymbol{k}^*_{2,5}, \boldsymbol{k}^*_{2,6})$ for delegation) is computed as $\boldsymbol{k}^*_{2,0} := \boldsymbol{k}^*_{1,0} + (\alpha_{0,1}\boldsymbol{k}^*_{1,1} + \alpha_{0,2}\boldsymbol{k}^*_{1,2}) + \sigma_{2,0}(v_3\boldsymbol{k}^*_{1,3} + v_4\boldsymbol{k}^*_{1,4})$, $\boldsymbol{k}^*_{2,j} := (\alpha_{j,1}\boldsymbol{k}^*_{1,1} + \alpha_{j,2}\boldsymbol{k}^*_{1,2}) + \sigma_{2,j}(v_3\boldsymbol{k}^*_{1,3} + v_4\boldsymbol{k}^*_{1,4})$ $(j = 1, 2, 3)$, and $\boldsymbol{k}^*_{2,j} := \psi^+\boldsymbol{k}^*_{1,j} + (\alpha_{j,1}\boldsymbol{k}^*_{1,1} + \alpha_{j,2}\boldsymbol{k}^*_{1,2}) + \sigma_{2,j}(v_3\boldsymbol{k}^*_{1,3} + v_4\boldsymbol{k}^*_{1,4})$ $(j = 5, 6)$, where $\alpha_{j,1}, \alpha_{j,2}, \sigma_{2,j}, \psi^+ \xleftarrow{\mathsf{U}} \mathbb{F}_q$ $(j = 0, 1, 2, 3, 5, 6)$. Then, the distribution of the delegated key (by $\mathsf{Delegate}$) is equivalent to that obtained by the key generation query (GenKey) except negligible probability (i.e., the simulation of 'create delegated key query' can be equivalent to that of 'create key query'.)

In general, as for the ℓ-th level secret key, $\overrightarrow{\boldsymbol{k}}^*_{\ell} := (\boldsymbol{k}^*_{\ell,0}, \ldots, \boldsymbol{k}^*_{\ell,\ell+1}, \boldsymbol{k}^*_{\ell,\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,n})$, the first one, $\boldsymbol{k}^*_{\ell,0}$, is used for decryption, the second part of components, $\boldsymbol{k}^*_{\ell,1}, \ldots, \boldsymbol{k}^*_{\ell,\ell+1}$, are for re-randomization (of a delegated key), and the last part of components, $\boldsymbol{k}^*_{\ell,\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,n}$, are for delegation.

5.2 HPE Scheme

$$\begin{aligned}
&\mathsf{Setup}(1^\lambda, \overrightarrow{\mu} := (n, d; \mu_1, \ldots, \mu_d)) : (\mathsf{param}, \mathbb{B}, \mathbb{B}^*) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, n+3),\\
&\quad \boldsymbol{d}_{n+1} := \boldsymbol{b}_{n+1} + \boldsymbol{b}_{n+2}, \quad \widehat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3}),\\
&\quad \mathsf{return}\ \ \mathsf{sk} := (X, \mathbb{B}^*),\ \ \mathsf{pk} := (1^\lambda, \mathsf{param}, \widehat{\mathbb{B}}).
\end{aligned}$$

$\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell) := ((v_1, \ldots, v_{\mu_1}), \ldots, (v_{\mu_{\ell-1}+1}, \ldots, v_{\mu_\ell})) :$

$\sigma_{j,i}, \psi, \eta_j \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad$ for $j = 0, \ldots, \ell+1, \mu_\ell+1, \ldots, n;\ i = 1, \ldots, \ell,$

$\boldsymbol{k}^*_{\ell,0} := \sum_{t=1}^{\ell} \sigma_{0,t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \eta_0 \boldsymbol{b}^*_{n+1} + (1-\eta_0)\boldsymbol{b}^*_{n+2},$

$\boldsymbol{k}^*_{\ell,j} := \sum_{t=1}^{\ell} \sigma_{j,t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \eta_j \boldsymbol{b}^*_{n+1} - \eta_j \boldsymbol{b}^*_{n+2}$
for $j = 1, \ldots, \ell+1,$

$\boldsymbol{k}^*_{\ell,j} := \sum_{t=1}^{\ell} \sigma_{j,t}(\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_i) + \psi \boldsymbol{b}^*_j + \eta_j \boldsymbol{b}^*_{n+1} - \eta_j \boldsymbol{b}^*_{n+2}$
for $j = \mu_\ell+1, \ldots, n,$

return $\overrightarrow{\boldsymbol{k}}^*_\ell := (\boldsymbol{k}^*_{\ell,0}, \ldots, \boldsymbol{k}^*_{\ell,\ell+1}, \boldsymbol{k}^*_{\ell,\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,n}).$

$\mathsf{Enc}(\mathsf{pk}, m \in \mathbb{G}_T, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_\ell) := ((x_1, \ldots, x_{\mu_1}), \ldots, (x_{\mu_{\ell-1}+1}, \ldots, x_{\mu_\ell})) :$

$(\overrightarrow{x}_{\ell+1}, \ldots, \overrightarrow{x}_d) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\mu_{\ell+1}-\mu_\ell} \times \cdots \times \mathbb{F}_q^{n-\mu_{d-1}},\ \delta_1, \ldots, \delta_d, \delta_{n+3}, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,$

$\boldsymbol{c}_1 := \sum_{t=1}^{d} \delta_t(\sum_{i=\mu_{t-1}+1}^{\mu_t} x_i \boldsymbol{b}_i) + \zeta \boldsymbol{d}_{n+1} + \delta_{n+3}\boldsymbol{b}_{n+3}, \quad c_2 := g_T^\zeta m,$

return $(\boldsymbol{c}_1, c_2).$

$\mathsf{Dec}(\mathsf{pk}, \boldsymbol{k}^*_{\ell,0}, \boldsymbol{c}_1, c_2) : m' := c_2 / e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,0}),$

return $m'.$

$\mathsf{Delegate}_\ell(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1} := (v_{\mu_\ell+1}, \ldots, v_{\mu_{\ell+1}})) :$

$\alpha_{j,i}, \sigma_j, \psi' \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad$ for $j = 0, \ldots, \ell+2, \mu_{\ell+1}+1, \ldots, n;\ i = 1, \ldots, \ell+1,$

$\boldsymbol{k}^*_{\ell+1,0} := \boldsymbol{k}^*_{\ell,0} + \sum_{i=1}^{\ell+1} \alpha_{0,i} \boldsymbol{k}^*_{\ell,i} + \sigma_0(\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,i}),$

$\boldsymbol{k}^*_{\ell+1,j} := \sum_{i=1}^{\ell+1} \alpha_{j,i} \boldsymbol{k}^*_{\ell,i} + \sigma_j(\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,i})$ for $j = 1, \ldots, \ell+2,$

$\boldsymbol{k}^*_{\ell+1,j} := \sum_{i=1}^{\ell+1} \alpha_{j,i} \boldsymbol{k}^*_{\ell,i} + \sigma_j(\sum_{i=\mu_\ell+1}^{\mu_{\ell+1}} v_i \boldsymbol{k}^*_{\ell,i}) + \psi' \boldsymbol{k}^*_{\ell,j}$ for $j = \mu_{\ell+1}+1, \ldots, n,$

return $\overrightarrow{\boldsymbol{k}}^*_{\ell+1} := (\boldsymbol{k}^*_{\ell+1,0}, \ldots, \boldsymbol{k}^*_{\ell+1,\ell+2}, \boldsymbol{k}^*_{\ell+1,\mu_{\ell+1}+1}, \ldots, \boldsymbol{k}^*_{\ell+1,n}).$

[Correctness] Assume that ciphertext $(\boldsymbol{c}_1, c_2)$ is generated by $\mathsf{Enc}(\mathsf{pk}, m, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h))$ and secret key $\boldsymbol{k}^*_{\ell,0}$ is generated by $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell))$. Note that $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,0}) = g_T^{\sum_{1 \le i \le \ell} \sigma_i \delta_i \overrightarrow{x}_i \cdot \overrightarrow{v}_i + \zeta}$. If $\ell \le h$ and $\overrightarrow{x}_i \cdot \overrightarrow{v}_i = 0$ for all i s.t. $1 \le i \le \ell$, then $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,0}) = g_T^\zeta$. Otherwise, $e(\boldsymbol{c}_1, \boldsymbol{k}^*_{\ell,0})$ is uniformly distributed. Hence, correctness holds for secret keys generated by GenKey, and it also holds for keys generated by $\mathsf{Delegate}$ by Claim 1.

Remark: A generalized delegation (not limited to a hierarchical delegation) system can be constructed on (1-level) PE described in the first part of Section 5.1, where the parameters are the same as above.

In the generalized delegatable PE scheme, secret key generation procedure $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, \overrightarrow{v}_1 := (v_{1,1}, \ldots, v_{1,n}))$ outputs $\overrightarrow{\boldsymbol{k}}^*_1 := (\boldsymbol{k}^*_{1,\mathsf{dec}}, \boldsymbol{k}^*_{1,\mathsf{ran},1}, \boldsymbol{k}^*_{1,\mathsf{ran},2}, \boldsymbol{k}^*_{1,\mathsf{del},1}, \ldots, \boldsymbol{k}^*_{1,\mathsf{del},n})$, where $\boldsymbol{k}^*_{1,\mathsf{dec}} := \sigma_{\mathsf{dec}}(\sum_{i=1}^{n} v_{1,i} \boldsymbol{b}^*_i) + \eta_{\mathsf{dec}} \boldsymbol{b}^*_{n+1} + (1-\eta_{\mathsf{dec}})\boldsymbol{b}^*_{n+2}$; $\boldsymbol{k}^*_{1,\mathsf{ran},j} := \sigma_{\mathsf{ran},j}(\sum_{i=1}^{n} v_{1,i} \boldsymbol{b}^*_i) + \eta_{\mathsf{ran},j} \boldsymbol{b}^*_{n+1} - \eta_{\mathsf{ran},j} \boldsymbol{b}^*_{n+2}$ $(j = 1, 2)$; $\boldsymbol{k}^*_{1,\mathsf{del},j} := \sigma_{\mathsf{del},j}(\sum_{i=1}^{n} v_{1,i} \boldsymbol{b}^*_i) + \psi \boldsymbol{b}^*_j + \eta_{\mathsf{del},j} \boldsymbol{b}^*_{n+1} - \eta_{\mathsf{del},j} \boldsymbol{b}^*_{n+2}$ $(j = 1, \ldots, n)$.

To delegate secret key $\overrightarrow{\boldsymbol{k}}^*_1$ for $\overrightarrow{v}_2 := (v_{2,1}, \ldots, v_{2,n})$, where $\overrightarrow{v}_2 \notin \mathsf{span}\langle \overrightarrow{v}_1 \rangle$, $\mathsf{Delegate}_1(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_1, \overrightarrow{v}_2)$ outputs $\overrightarrow{\boldsymbol{k}}^*_2 := (\boldsymbol{k}^*_{2,\mathsf{dec}}, \boldsymbol{k}^*_{2,\mathsf{ran},1}, \boldsymbol{k}^*_{2,\mathsf{ran},2}, \boldsymbol{k}^*_{2,\mathsf{del},1}, \ \ldots,$

$\boldsymbol{k}^*_{2,\mathsf{del},n}$). Here, $\boldsymbol{k}^*_{2,\mathsf{dec}} := \boldsymbol{k}^*_{1,\mathsf{dec}} + \sum_{i=1}^{2} \alpha_{\mathsf{dec},i}\boldsymbol{k}^*_{1,\mathsf{ran},i} + \sigma_{2,\mathsf{dec}}(\sum_{i=1}^{n} v_{2,i}\boldsymbol{k}^*_{1,\mathsf{del},i})$; $\boldsymbol{k}^*_{2,\mathsf{ran},j} := \sum_{i=1}^{2} \alpha_{\mathsf{ran},i}\boldsymbol{k}^*_{1,\mathsf{ran},i} + \sigma_{2,\mathsf{ran},j}(\sum_{i=1}^{n} v_{2,i}\boldsymbol{k}^*_{1,\mathsf{del},i})$ $(j = 1, 2, 3)$; $\boldsymbol{k}^*_{2,\mathsf{del},j} := \sum_{i=1}^{2} \alpha_{\mathsf{del},i}\ \boldsymbol{k}^*_{1,\mathsf{ran},i} + \sigma_{2,\mathsf{del},j}(\sum_{i=1}^{n} v_{2,i}\boldsymbol{k}^*_{1,\mathsf{del},i}) + \psi'\boldsymbol{k}^*_{1,\mathsf{del},j}$ $(j = 1, \ldots, n)$. Further delegation for $\overrightarrow{\boldsymbol{k}}^*_\ell$ $(\ell = 2, 3, \ldots)$ can be done in the same manner.

Ciphertext $(\boldsymbol{c}_1, c_2)$ for attribute $\overrightarrow{x} := (x_1, \ldots, x_n)$ and plaintext $m \in \mathbb{G}_T$ is the same as that of the 1-level PE. Key $\overrightarrow{\boldsymbol{k}}^*_1$ can decrypt $(\boldsymbol{c}_1, c_2)$ if $\overrightarrow{v}_1 \cdot \overrightarrow{x} = 0$, and key $\overrightarrow{\boldsymbol{k}}^*_2$ can decrypt $(\boldsymbol{c}_1, c_2)$ if $(\overrightarrow{v}_1 \cdot \overrightarrow{x} = 0) \wedge (\overrightarrow{v}_2 \cdot \overrightarrow{x} = 0)$. Namely the capability of delegated key $\overrightarrow{\boldsymbol{k}}^*_2$ is more limited than that of its parent key $\overrightarrow{\boldsymbol{k}}^*_1$. In general, the ℓ-th delegated secret key $\overrightarrow{\boldsymbol{k}}^*_\ell$ can decrypt $(\boldsymbol{c}_1, c_2)$ if $(\overrightarrow{v}_1 \cdot \overrightarrow{x} = 0) \wedge \cdots \wedge (\overrightarrow{v}_\ell \cdot \overrightarrow{x} = 0)$, where $\overrightarrow{v}_j \notin \mathsf{span}\langle \overrightarrow{v}_1, \ldots, \overrightarrow{v}_{j-1} \rangle$ for $2 \leq j \leq \ell$.

5.3 Security

Theorem 2. *The proposed HPE scheme is selectively attribute-hiding against chosen plaintext attacks under the RDSP and IDSP assumptions. For any adversary $\mathcal{A}$, there exist probabilistic machines $\mathcal{B}_1$ and $\mathcal{B}_2$, whose running times are essentially the same as that of $\mathcal{A}$, such that for any security parameter λ,*

$$\mathsf{Adv}^{\mathsf{HPE},\mathsf{AH}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{RDSP}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{IDSP}}_{\mathcal{B}_2}(\lambda) + 3\nu/q$$

where ν is the number of adversary's queries.

Proof Outline: To prove the security, we employ five games, Game 0 (original selective-security game) to Game 4 whose advantage is 0, where, roughly, Game 1 is conceptually changed (the timing of challenger's coin flips is changed) from Game 0, a delegated key query (i.e., a reveal query of an already-created delegated key) is replied by using GenKey (in place of $\mathsf{Delegate}$) in Game 2, the plaintext part of the target ciphertext is randomized in Game 3, and the attribute vector part of the target ciphertext is randomized in Game 4.

Since the distribution regarding each revealed key query in Game 2 is equivalent to that in Game 1 except with probability at most $3/q$, the gap between Games 1 and 2 is bounded by $3\nu/q$.

To prove that the gap between Games 2 and 3 is bounded by the advantage of the RDSP assumption, target ciphertext $(\boldsymbol{c}_1, c_2)$ for $m^{(b)}$ is generated by using $\boldsymbol{e}_\beta$ from the RDSP assumption such that $\boldsymbol{c}_1 := \boldsymbol{e}_\beta + \zeta \boldsymbol{d}_{n+1}$ and $c_2 := g_T^\zeta m^{(b)}$. Then $(\boldsymbol{c}_1, c_2)$ is a ciphertext in Game 2 when $\beta = 0$, and it is a ciphertext in Game 3 when $\beta = 1$. The key generation oracle simulation can be perfectly executed by using $\{\boldsymbol{h}_i^{(k)*}, \tau_i^{(k)}\}_{i=1,\ldots,n;k=1,2,3}$ from the RDSP assumption (see Remark after Theorem 1). It can be done similarly to evaluate the gap between Games 3 and 4 (through the IDSP assumption).

Proof of Theorem 2
To prove Theorem 2, we consider the following five games.

Game 0: Original game (Definition 5).

Game 1: Game 1 is the same as Game 0 except the following procedures:

1. When challenger $\mathcal{C}$ gets challenge attributes $(\overrightarrow{x}_1^{(0)}, \ldots, \overrightarrow{x}_{h^{(0)}}^{(0)})$ and $(\overrightarrow{x}_1^{(1)}, \ldots, \overrightarrow{x}_{h^{(1)}}^{(1)})$ in the first step of the game, $\mathcal{C}$ selects (challenge) bit $b \xleftarrow{\mathsf{U}} \{0,1\}$, and computes
$$(x_1^+, \ldots, x_n^+) := (\delta_1 \overrightarrow{x}_1, \ldots, \delta_d \overrightarrow{x}_d),$$
where $h := h^{(b)}, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) := (\overrightarrow{x}_1^{(b)}, \ldots, \overrightarrow{x}_h^{(b)}), (\overrightarrow{x}_{h+1}, \ldots, \overrightarrow{x}_d) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\mu_{h+1}-\mu_h} \times \cdots \times \mathbb{F}_q^{n-\mu_{d-1}}$, and $\delta_1, \ldots, \delta_d \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
2. When $\mathcal{C}$ gets challenge plaintexts $(m^{(0)}, m^{(1)})$ from adversary $\mathcal{A}$, challenger $\mathcal{C}$ computes $(\boldsymbol{c}_1, c_2)$ as below and returns it to $\mathcal{A}$.
$$\boldsymbol{c}_1 := \textstyle\sum_{i=1}^n x_i^+ \boldsymbol{b}_i + \zeta \boldsymbol{d}_{n+1} + \delta_{n+3} \boldsymbol{b}_{n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$
where $\delta_{n+3}, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Game 2: Game 2 is the same as Game 1 except the following procedures.

1. When a create key query is issued by $\mathcal{A}$, challenger $\mathcal{C}$ only records the specified predicates, and when a create delegated key query is issued, $\mathcal{C}$ only records the specified keys and predicates. In this step, $\mathcal{C}$ just records, but creates no corresponding keys.
2. When a reveal key query is issued for a hierarchical (level-ℓ) predicate $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$ which has been already recorded, $\mathcal{C}$ creates the queried key by using GenKey. In addition, there is a special rule such that $(\sigma_{0,1}, \ldots, \sigma_{0,\ell}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\ell}$ is selected again if $\sum_{t=1}^{\ell} \sigma_{0,t} \delta_t \overrightarrow{x}_t \cdot \overrightarrow{v}_t = 0$ in the computation process of GenKey.

Game 3: Game 3 is the same as Game 2 except the target ciphertext $(\boldsymbol{c}_1, c_2)$ is generated as follows:
$$\boldsymbol{c}_1 := \textstyle\sum_{i=1}^n x_i^+ \boldsymbol{b}_i + \zeta_1 \boldsymbol{b}_{n+1} + \zeta_2 \boldsymbol{b}_{n+2} + \delta_{n+3} \boldsymbol{b}_{n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$
where $\delta_{n+3}, \zeta, \zeta_1, \zeta_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Game 4: Game 4 is the same as Game 3 except the target ciphertext $(\boldsymbol{c}_1, c_2)$ is generated as follows:
$$\boldsymbol{c}_1 := \textstyle\sum_{i=1}^n u_i \boldsymbol{b}_i + \zeta_1 \boldsymbol{b}_{n+1} + \zeta_2 \boldsymbol{b}_{n+2} + \delta_{n+3} \boldsymbol{b}_{n+3}, \quad c_2 := g_T^{\zeta} m^{(b)},$$
where $\delta_{n+3}, \zeta, \zeta_1, \zeta_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $\overrightarrow{u} := (u_1, \ldots, u_n) \xleftarrow{\mathsf{U}} \mathbb{F}_q^n \setminus \{\overrightarrow{0}\}$.

Let $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda)$ be $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HPE,AH}}(\lambda)$ in Game 0, and $\mathsf{Adv}_{\mathcal{A}}^{(i)}(\lambda)$ $(i = 1, \ldots, 4)$ be the advantage of $\mathcal{A}$ in Game i. It is clear that $\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda)$, since it is a conceptual change. It is also clear that $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 0$ by Lemma 4.

We will show three lemmas (Lemmas 1, 2, 3) that evaluate the gaps between pairs of $\mathsf{Adv}_{\mathcal{A}}^{(i)}(\lambda)$ $(i = 1, 2, 3, 4)$. From these lemmas, we obtain $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HPE,AH}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(1)}(\lambda) \leq \sum_{i=1}^{3} \left| \mathsf{Adv}_{\mathcal{A}}^{(i)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(i+1)}(\lambda) \right| + \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{RDSP}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{IDSP}}(\lambda) + 3\nu/q$. □

Lemma 1. *For any adversary* $\mathcal{A}$, $|\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)| \leq 3\nu/q$.

Proof. The distribution of $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ generated by GenKey for a level-$(\ell+1)$ predicate is equivalent to that by the combination of GenKey for the level-ℓ predicate and $\mathsf{Delegate}_\ell$ except with probability $2/q$, from Claim 1. Moreover, the special rule in Game 2 causes probability gap at most $1/q$ for each GenKey operation. Therefore, the revealed key distribution in Game 1 is equivalent to that in Game 2 except with probability at most $(1-(1-3/q)^\nu) \leq 3\nu/q$, since the number of delegate queries is upper-bounded by ν. Hence (by using Shoup's difference lemma), the difference of $\mathsf{Adv}^{(1)}_{\mathcal{A}}(\lambda)$ and $\mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda)$ is upper-bounded by $3\nu/q$. □

Claim 1. *If* $\overrightarrow{\boldsymbol{k}}^*_\ell$ *is generated by* $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell))$, *the distribution of* $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ *generated by* $\mathsf{Delegate}(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1})$ *is equivalent to that of* $\overrightarrow{\boldsymbol{k}}^*_{\ell+1}$ *generated by* $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, .., \overrightarrow{v}_\ell, \overrightarrow{v}_{\ell+1}))$ *except with probability at most* $2/q$.

Proof. The distribution of level-ℓ key $\boldsymbol{k}^*_{\ell,j}$ $(j = 1, \ldots, \ell+1)$ is represented by that of the $\ell+1$ coefficients, $(\sigma_{j,1}, \ldots, \sigma_{j,\ell}, \eta_j)$, of $\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i \boldsymbol{b}^*_{\ell,i}$ $(t = 1, \ldots, \ell)$ and $\boldsymbol{b}^*_{n+1}$ (and the coefficient, ψ, of $\boldsymbol{b}^*_j$ in addition when $j = \mu_\ell + 1, \ldots, n$), since the coefficient of $\boldsymbol{b}^*_{n+2}$ is dependent of that of $\boldsymbol{b}^*_{n+1}$.

Similarly, the distribution of level-$(\ell+1)$ key $\boldsymbol{k}^*_{\ell+1,j}$ $(j = 1, \ldots, \ell+2)$ is represented by that of the $\ell+2$ coefficients, $(\sigma_{j,1}, \ldots, \sigma_{j,\ell+1}, \eta_j)$.

When level-ℓ key $\boldsymbol{k}^*_{\ell,j}$ $(j = 1, \ldots, \ell+1)$ is generated by $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell))$, $(\sigma_{j,1}, \ldots, \sigma_{j,\ell}, \eta_j)_{j=1,\ldots,\ell+1}$ is uniformly distributed.

If coefficient matrix $(\sigma_{j,1}, \ldots, \sigma_{j,\ell}, \eta_j)_{j=1,\ldots,\ell+1}$ $((\ell+1) \times (\ell+1)$ matrix) of $(\boldsymbol{k}^*_{\ell,j})_{j=1,\ldots,\ell+1}$ is regular and $\psi \neq 0$, then the coefficients, $(\sigma_{j,1}, \ldots, \sigma_{j,\ell+1}, \eta_j)$, of $\mathsf{Delegate}(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1})$ is uniformly distributed, i.e., $\mathsf{Delegate}(\mathsf{pk}, \overrightarrow{\boldsymbol{k}}^*_\ell, \overrightarrow{v}_{\ell+1})$ is equivalently distributed as $\mathsf{GenKey}(\mathsf{pk}, \mathsf{sk}, (\overrightarrow{v}_1, \ldots, \overrightarrow{v}_{\ell+1}))$.

Here, $(\sigma_{j,1}, \ldots, \sigma_{j,\ell}, \eta_j)_{j=1,\ldots,\ell+1}$ $((\ell+1) \times (\ell+1)$ matrix) of $(\boldsymbol{k}^*_{\ell,j})_{j=1,\ldots,\ell+1}$ is regular and $\psi \neq 0$ except with probability at most $2/q$. □

Lemma 2. *For any adversary* $\mathcal{A}$, *there exists a probabilistic machine* $\mathcal{B}_1$, *whose running time is essentially the same as that of* $\mathcal{A}$, *such that for any security parameter* λ, $|\mathsf{Adv}^{(2)}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{A}}(\lambda)| = \mathsf{Adv}^{\mathsf{RDSP}}_{\mathcal{B}_1}(\lambda)$.

Proof. In order to prove Lemma 2, we construct a probabilistic machine $\mathcal{B}_1$ against the RDSP problem by using any adversary $\mathcal{A}$ in a security game (Game 2 or 3) as a black box as follows:

1. $\mathcal{B}_1$ is given RDSP instance $(\mathsf{param}, \widehat{\mathbb{B}}, \{\boldsymbol{h}^{(k)*}_i, \tau^{(k)}_i\}_{i=1,\ldots,n;k=1,2,3}, \overrightarrow{y}, \boldsymbol{e}_\beta)$.
2. $\mathcal{B}_1$ plays a role of challenger $\mathcal{C}$ in the security game against adversary $\mathcal{A}$.
3. When $\mathcal{B}_1$ (or challenger $\mathcal{C}$) gets challenge attributes $(\overrightarrow{x}^{(0)}_1, \ldots, \overrightarrow{x}^{(0)}_{h^{(0)}})$ and $(\overrightarrow{x}^{(1)}_1, \ldots, \overrightarrow{x}^{(1)}_{h^{(1)}})$ in the first step of the game, $\mathcal{B}_1$ selects (challenge) bit $b \xleftarrow{\mathsf{U}} \{0,1\}$, and computes

$$(x^+_1, \ldots, x^+_n) := (\delta_1 \overrightarrow{x}_1, \ldots, \delta_d \overrightarrow{x}_d),$$

where $h := h^{(b)}, (\overrightarrow{x}_1, \ldots, \overrightarrow{x}_h) := (\overrightarrow{x}_1^{(b)}, \ldots, \overrightarrow{x}_h^{(b)}), (\overrightarrow{x}_{h+1}, \ldots, \overrightarrow{x}_d) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{\mu_{h+1}-\mu_h} \times \cdots \times \mathbb{F}_q^{n-\mu_{d-1}}$, and $\delta_1, \ldots, \delta_d \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Let $(\pi_{i,j}) \xleftarrow{\mathsf{U}} \{\Pi \in GL(n, \mathbb{F}_q) \mid \overrightarrow{y} = \overrightarrow{x}^+ \cdot \Pi,\ \Pi^{\mathrm{T}} = \Pi\}$, and $\Pi^* := (\pi^*_{i,j}) := ((\pi_{i,j})^{\mathrm{T}})^{-1}$. Note that $\overrightarrow{x}^+ = \overrightarrow{y} \cdot \Pi^*$. Public parameter pk is then calculated as follows and $\mathcal{B}_1$ returns pk to $\mathcal{A}$:

$$\widetilde{\boldsymbol{b}}_j := \textstyle\sum_{\varrho=1}^n \pi_{j,\varrho} \boldsymbol{b}_\varrho,\ \widetilde{\boldsymbol{b}}^*_j := \sum_{\varrho=1}^n \pi^*_{j,\varrho} \boldsymbol{b}^*_\varrho \quad (j = 1, \ldots, n),$$
$$\widetilde{\mathbb{B}} := (\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3}), \quad \mathsf{pk} := (1^\lambda, \mathsf{param}, \widetilde{\mathbb{B}}).$$

4. When a reveal key query is issued for a hierarchical (level-ℓ) predicate $(\overrightarrow{v}_1, \ldots, \overrightarrow{v}_\ell)$ which has been already recorded, $\mathcal{B}_1$ answers as follows: for $j = 0, \ldots, \ell+1, \mu_\ell + 1, \ldots, n$, $\mathcal{B}_1$ calculates

$$\overrightarrow{v}^+_j := (v^+_{j,1}, \ldots, v^+_{j,\mu_\ell}) := (\sigma_{j,1} \overrightarrow{v}_1, \ldots, \sigma_{j,\ell} \overrightarrow{v}_\ell), \tag{1}$$

where $\sigma_{j,1}, \ldots, \sigma_{j,\ell} \xleftarrow{\mathsf{U}} \mathbb{F}_q$. Then, $\mathcal{B}_1$ calculates and returns $\overrightarrow{\boldsymbol{k}}^*_\ell := (\boldsymbol{k}^*_{\ell,0}, \ldots, \boldsymbol{k}^*_{\ell+1,\ell+1}, \boldsymbol{k}^*_{\ell,\mu_\ell+1}, \ldots, \boldsymbol{k}^*_{\ell,n})$ using $\{\boldsymbol{h}^{(k)*}_i, \tau^{(k)}_i\}$ in the RDSP instance:

$\theta_0 := \sum_{k=1}^3 a_{0,k} \sum_{i=1}^{\mu_\ell} v^+_{0,i} \sum_{\varrho=1}^n \pi^*_{i,\varrho} \tau^{(k)}_\varrho$,
$\boldsymbol{k}^*_{\ell,0} := \theta_0^{-1} \sum_{k=1}^3 a_{0,k} \sum_{i=1}^{\mu_\ell} v^+_{0,i} \sum_{\varrho=1}^n \pi^*_{i,\varrho} \boldsymbol{h}^{(k)*}_\varrho$,
For $j = 1, \ldots, \ell+1, \mu_\ell+1, \ldots, n; s = 1, 2$,
  $\theta_{j,s} := \sum_{k=1}^3 a_{j,k,s} \sum_{i=1}^{\mu_\ell} v^+_{j,i} \sum_{\varrho=1}^n \pi^*_{i,\varrho} \tau^{(k)}_\varrho$,
  $\boldsymbol{f}^*_{j,s} := \sum_{k=1}^3 a_{j,k,s} \sum_{i=1}^{\mu_\ell} v^+_{j,i} \sum_{\varrho=1}^n \pi^*_{i,\varrho} \boldsymbol{h}^{(k)*}_\varrho$,
  $\boldsymbol{k}^*_{\ell,j} := \theta_{j,2} \boldsymbol{f}^*_{j,1} - \theta_{j,1} \boldsymbol{f}^*_{j,2}$,
For $j = \mu_\ell + 1, \ldots, n$,
  For $i = 1, \ldots, \mu_\ell, j$,
    $\varphi_i := \sum_{k=1}^3 \widetilde{a}_k \sum_{\varrho=1}^n \pi^*_{i,\varrho} \tau^{(k)}_\varrho$, $\quad \boldsymbol{m}^*_i := \sum_{k=1}^3 \widetilde{a}_k \sum_{\varrho=1}^n \pi^*_{i,\varrho} \boldsymbol{h}^{(k)*}_\varrho$,
  $z_j := \varphi_j \left(\sum_{i=1}^{\mu_\ell} v^+_{j,i} \varphi_i\right)^{-1}$, $\quad \boldsymbol{k}^*_{\ell,j} := \boldsymbol{k}^*_{\ell,j} + \boldsymbol{m}^*_j - z_j \sum_{i=1}^{\mu_\ell} v^+_{j,i} \boldsymbol{m}^*_i$,

where $a_{0,k}, a_{j,k,s}, \widetilde{a}_k \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $j = 1, \ldots, \ell+1, \mu_\ell+1, \ldots, n; k = 1, 2, 3; s = 1, 2$.

If $\theta_0 = 0$, $\{\sigma_{0,t}, a_{0,k} \xleftarrow{\mathsf{U}} \mathbb{F}_q\}_{k=1,2,3;t=1,\ldots,\ell}$ is selected again. For $j = \mu_\ell + 1, \ldots, n$, if $\sum_{i=1}^{\mu_\ell} v^+_{j,i} \varphi_i = 0$, $\{\sigma_{j,t}, \widetilde{a}_k \xleftarrow{\mathsf{U}} \mathbb{F}_q\}_{k=1,2,3;t=1,\ldots,\ell}$ is selected again.

5. When $\mathcal{B}_1$ (or $\mathcal{C}$) gets challenge plaintexts $(m^{(0)}, m^{(1)})$ (from $\mathcal{A}$), $\mathcal{B}_1$ calculates and returns $(\boldsymbol{c}_1, c_2)$ s.t. $\boldsymbol{c}_1 := \boldsymbol{e}_\beta + \zeta \boldsymbol{d}_{n+1}$ and $c_2 := g_T^\zeta m^{(b)}$ using $\boldsymbol{e}_\beta$ in the RDSP instance, ζ, and $m^{(b)}$, where $\zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
6. After the encryption query, GenKey oracle simulation for a reveal key query is executed as above.
7. $\mathcal{A}$ outputs bit b'. If $b = b'$, $\mathcal{B}_1$ outputs $\beta' := 1$. Otherwise, $\mathcal{B}_1$ outputs $\beta' := 0$.

To prove Lemma 2, we show Claims 2, 3, and 4.

Claim 2. *Public parameter* pk *generated in step 3 above has the same distribution as that in Game 2 (and Game 3).*

Proof. Let $D := \begin{pmatrix} \varPi & 0 \\ 0 & I_3 \end{pmatrix}$ be square $(n+3) \times (n+3)$ matrix composed of $\varPi$ and the identity matrix I_3. Then basis $(\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_n, \boldsymbol{b}_{n+1}, \boldsymbol{b}_{n+2}, \boldsymbol{b}_{n+3})$ of $\mathbb{V}$ is obtained from basis $\mathbb{B}$ by the linear transformation determined by D. Hence, its distribution is uniform. Therefore, $\widetilde{\mathbb{B}} = (\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_n, \boldsymbol{d}_{n+1}, \boldsymbol{b}_{n+3})$ in step 3 has the same distribution as that in Game 2 (and Game 3). □

Claim 3. *Secret key* $\overrightarrow{\boldsymbol{k}}^*_\ell$ *generated in steps 4 and 6 above has the same distribution as that in Game 2 (and Game 3).*

Proof. First, we verify that basis $(\widetilde{\boldsymbol{b}}^*_1, \ldots, \widetilde{\boldsymbol{b}}^*_n, \boldsymbol{b}^*_{n+1}, \boldsymbol{b}^*_{n+2}, \boldsymbol{b}^*_{n+3})$ of $\mathbb{V}^*$ is obtained by the linear transformation $(D^{\mathrm{T}})^{-1}$, where D is defined in the proof of Claim 2. That is, it is dual orthonormal to basis $(\widetilde{\boldsymbol{b}}_1, \ldots, \widetilde{\boldsymbol{b}}_n, \boldsymbol{b}_{n+1}, \boldsymbol{b}_{n+2}, \boldsymbol{b}_{n+3})$. Therefore, we can consider $\boldsymbol{k}^*_{\ell,j}$ w.r.t. this dual orthonormal basis.

Secret key $\boldsymbol{k}^*_{\ell,0}$ generated in steps 4 and 6 is $\theta_0^{-1}(\sum_{k=1}^3 a_{0,k}\omega^{(k)}) \sum_{i=1}^{\mu_\ell} v^+_{0,i}\widetilde{\boldsymbol{b}}^*_i + \theta_0^{-1}\theta_1\boldsymbol{b}^*_{n+1} + \theta_0^{-1}\theta_2\boldsymbol{b}^*_{n+2}$, where $\theta_1 := (\sum_{k=1}^3 a_{0,k}\gamma_1^{(k)})\overrightarrow{v}^+_0 \cdot \overrightarrow{x}^+$, $\theta_2 := (\sum_{k=1}^3 a_{0,k}\gamma_2^{(k)})\overrightarrow{v}^+_0 \cdot \overrightarrow{x}^+$, and $\theta_0 = \theta_1 + \theta_2$. Let $\sigma := \theta_0^{-1}(\sum_{k=1}^3 a_{0,k}\omega^{(k)})$. Then, $\sigma, \theta_1, \theta_2$ are independently uniform, since $a_{0,k}$ are independently uniform, and $\theta_0^{-1}\theta_1 + \theta_0^{-1}\theta_2 = 1$. Also, from (1), the coefficients of $\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i\widetilde{\boldsymbol{b}}^*_i$ in $\boldsymbol{k}^*_{\ell,0}$ for each $1 \le t \le \ell$ are all uniformly and independently distributed. Therefore, generated $\boldsymbol{k}^*_{\ell,0}$ has the same distribution as in Game 2 and Game 3.

Similarly, for $j = 1, \ldots, \ell+1, \mu_\ell+1, \ldots, n$, the j-th key $\boldsymbol{k}^*_{\ell,j}$ has independently uniform coefficients w.r.t. $\sum_{i=\mu_{t-1}+1}^{\mu_t} v_i\widetilde{\boldsymbol{b}}^*_i$ for each $1 \le t \le \ell$, and the sum of the coefficients of $\widetilde{\boldsymbol{b}}^*_{n+1}$ and $\widetilde{\boldsymbol{b}}^*_{n+2}$ is zero.

Finally, we investigate the distribution of the coefficients of $\widetilde{\boldsymbol{b}}^*_j$ in $\boldsymbol{k}^*_{\ell,j}$ for $j = \mu_\ell + 1, \ldots, n$. The additional term $\boldsymbol{m}^*_j - z_j \sum_{i=1}^{\mu_\ell} v^+_{j,i}\boldsymbol{m}^*_i$ is

$$\begin{aligned} &-z_j \left(\sum_{k=1}^3 \widetilde{a}_k\omega^{(k)}\right) \sum_{i=1}^{\mu_\ell} v^+_{j,i}\widetilde{\boldsymbol{b}}^*_i + \left(\sum_{k=1}^3 \widetilde{a}_k\omega^{(k)}\right)\widetilde{\boldsymbol{b}}^*_j \\ &+ \left(\varphi_{1,j} - z_j \sum_{i=1}^{\mu_\ell} v^+_{j,i}\varphi_{1,i}\right)\boldsymbol{b}^*_{n+1} + \left(\varphi_{2,j} - z_j \sum_{i=1}^{\mu_\ell} v^+_{j,i}\varphi_{2,i}\right)\boldsymbol{b}^*_{n+2}, \quad (2) \end{aligned}$$

where $\varphi_{1,i} := \left(\sum_{k=1}^3 \widetilde{a}_k\gamma_1^{(k)}\right) x^+_i$, $\varphi_{2,i} := \left(\sum_{k=1}^3 \widetilde{a}_k\gamma_2^{(k)}\right) x^+_i$ and $\varphi_i = \varphi_{1,i} + \varphi_{2,i}$. Therefore, for $j = \mu_\ell + 1, \ldots, n$, the sum of the coefficients of $\boldsymbol{b}^*_{n+1}$ and $\boldsymbol{b}^*_{n+2}$ in (2) is zero, and the coefficients of $\widetilde{\boldsymbol{b}}^*_j$ in $\boldsymbol{k}^*_{\ell,j}$ are common, $\sum_{k=1}^3 \widetilde{a}_k\omega^{(k)}$, which is uniformly distributed. □

Claim 4. *If* $\beta = 0$*, the distribution of* $(\boldsymbol{c}_1, c_2)$ *generated in step 5 is the same as that in Game 2. If* $\beta = 1$*, the distribution of* $(\boldsymbol{c}_1, c_2)$ *generated in step 5 is the same as that in Game 3.*

Proof. If $\beta = 0$, $\boldsymbol{c}_1 = \delta_1 \sum_{i=1}^n y_i\boldsymbol{b}_i + \zeta\boldsymbol{d}_{n+1} + \delta_2\boldsymbol{b}_{n+3} = \delta_1 \sum_{i=1}^n x^+_i\widetilde{\boldsymbol{b}}_i + \zeta\boldsymbol{d}_{n+1} + \delta_2\boldsymbol{b}_{n+3}$ and $c_2 := g_T^{\zeta} m^{(b)}$. This is the target ciphertext in Game 2 with $\mathsf{pk} :=$

$(1^\lambda, \mathsf{param}, \widetilde{\mathbb{B}})$. If $\beta = 1$, $\boldsymbol{c}_1 = \delta_1 \sum_{i=1}^{n} x_i^+ \widetilde{\boldsymbol{b}}_i + (\zeta+\zeta_1)\boldsymbol{b}_{n+1} + (\zeta+\zeta_2)\boldsymbol{b}_{n+2} + \delta_2 \boldsymbol{b}_{n+3}$ and $c_2 := g_T^{\zeta} m^{(b)}$. Because $\zeta + \zeta_1, \zeta + \zeta_2$, and ζ are independently uniform, this is the target ciphertext in Game 3 with $\mathsf{pk} := (1^\lambda, \mathsf{param}, \widetilde{\mathbb{B}})$. □

From Claims 2, 3, and 4, when $\beta = 0$, the advantage of $\mathcal{A}$ in the above game is equal to that in Game 2, i.e., $\mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda)$, and also is equal to $\mathsf{Pr}_0 := \mathsf{Pr}\left[\mathcal{B}_1(1^\lambda, \rho) \to 1 \;\middle|\; \rho \xleftarrow{\mathsf{R}} \mathcal{G}_0^{\mathsf{RDSP}}(1^\lambda, n)\right]$. Similarly, when $\beta = 1$, we see that the advantage of $\mathcal{A}$ in the above game is equal to $\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$, and also is equal to $\mathsf{Pr}_1 := \mathsf{Pr}\left[\mathcal{B}_1(1^\lambda, \rho) \to 1 \;\middle|\; \rho \xleftarrow{\mathsf{R}} \mathcal{G}_1^{\mathsf{RDSP}}(1^\lambda, n)\right]$. Therefore, $|\mathsf{Adv}_{\mathcal{A}}^{(2)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)| = |\mathsf{Pr}_0 - \mathsf{Pr}_1| = \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{RDSP}}(\lambda)$. This completes the proof of Lemma 2. □

Lemma 3. *For any adversary $\mathcal{A}$, there exists a probabilistic machine $\mathcal{B}_2$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda)| = \mathsf{Adv}_{\mathcal{B}_2}^{\mathsf{IDSP}}(\lambda)$.*

Proof. Lemma 3 is similarly proved as Lemma 2. The proof will be given in the full version of this paper. □

Lemma 4. *For any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 0$.*

Proof. The value of b is independent from the adversary's view in Game 4. Hence, $\mathsf{Adv}_{\mathcal{A}}^{(4)}(\lambda) = 0$. □

References

1. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: 2007 IEEE Symposium on Security and Privacy, pp. 321–334. IEEE Press, Los Alamitos (2007)
2. Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 223–238. Springer, Heidelberg (2004)
3. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)
4. Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
5. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001)
6. Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption scheme. In: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455–470. Springer, Heidelberg (2008)
7. Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. In: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535–554. Springer, Heidelberg (2007)

8. Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290–307. Springer, Heidelberg (2006)
9. Cocks, C.: An identity based encryption scheme based on quadratic residues. In: Honary, B. (ed.) Cryptography and Coding 2001. LNCS, vol. 2260, pp. 360–363. Springer, Heidelberg (2001)
10. Gentry, C.: Practical identity-based encryption without random oracles. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 445–464. Springer, Heidelberg (2006)
11. Gentry, C., Halevi, S.: Hierarchical identity-based encryption with polynomially many levels. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437–456. Springer, Heidelberg (2009)
12. Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 548–566. Springer, Heidelberg (2002)
13. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM Conference on Computer and Communication Security 2006, pp. 89–98. ACM, New York (2006)
14. Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 415–432. Springer, Heidelberg (2008)
15. Horwitz, J., Lynn, B.: Towards hierarchical identity-based encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 466–481. Springer, Heidelberg (2002)
16. Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pp. 146–162. Springer, Heidelberg (2008)
17. Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. In: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57–74. Springer, Heidelberg (2008)
18. Okamoto, T., Takashima, K.: A geometric approach on pairings and hierarchical predicate encryption. In: Poster session, EUROCRYPT 2009 (2009)
19. Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. In: ACM Conference on Computer and Communication Security 2007, pp. 195–203. ACM, New York (2007)
20. Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. In: ACM Conference on Computer and Communication Security 2006, pp. 99–112. ACM, New York (2006)
21. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005)
22. Shi, E., Waters, B.: Delegating capability in predicate encryption systems. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 560–578. Springer, Heidelberg (2008)
23. Takashima, K.: Efficiently computable distortion maps for supersingular curves. In: van der Poorten, A.J., Stein, A. (eds.) ANTS-VIII 2008. LNCS, vol. 5011, pp. 88–101. Springer, Heidelberg (2008)
24. Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. ePrint, IACR, `http://eprint.iacr.org/2008/290`