Des. Codes Cryptogr. (2011) 58:235–257
DOI 10.1007/s10623-010-9405-9

Inner-product encryption under standard assumptions

Jong Hwan Park

Received: 12 March 2010 / Revised: 30 April 2010 / Accepted: 4 May 2010 /
Published online: 18 May 2010

Abstract Predicate encryption is a generalized notion for public key encryption that enables one to encrypt attributes as well as a message. In this paper, we present a new inner-product encryption (IPE) scheme, as a specialized predicate encryption scheme, whose security relies on the well-known Decision Bilinear Diffie-Hellman (BDH) and Decision Linear assumptions. Our IPE scheme uses prime order groups equipped with a bilinear map and works in both symmetric and asymmetric bilinear maps. Our result is the first construction of IPE under the standard assumptions. Prior to our work, all IPE schemes known to date require non-standard assumptions to prove security, and moreover some of them use composite-order groups. To achieve our goal, we introduce a novel technique for attribute-hiding, which may be of independent interest.

Keywords Predicate encryption · Inner-product encryption · Bilinear maps

Mathematics Subject Classification (2000) 68P25 · 94A60

1 Introduction

Recently, predicate encryption [15] has been proposed as a generalized notion of public key encryption that allows one to encrypt a message M in a message space $\mathcal{M}$ as well as an attribute I in a set Σ. In a predicate encryption scheme, secret keys correspond to predicates in some class $\mathcal{F}$, and ciphertexts correspond to attributes in Σ. A ciphertext associated with the attribute $I \in \Sigma$ can then be decrypted by a secret key SK_f corresponding to the predicate $f \in \mathcal{F}$ if and only if $f(I) = 1$. Depending on how the class of the predicate f over Σ is defined, predicate encryption implies different types of primitives

Communicated by S. D. Galbraith.

J. H. Park (✉)
Department of Applied Mathematics, Kyung Hee University, Youngin, Korea
e-mail: jonghpark@khu.ac.kr

including identity-based encryption (IBE) [6,11,3,10,21,7,22], attribute-based encryption (ABE) [19,13,18], or searchable encryption on encrypted data [5,1,8]. For instance, if f is a class of an equality predicate, predicate encryption can lead to IBE where an attribute $I \in \Sigma$ corresponds to an identity.

Security for predicate encryption is considered in two aspects: *payload-hiding* and *attribute-hiding*. For a ciphertext associated with both M and I, the former requires that the ciphertext hides all information about only M, and the latter requires that the ciphertext hides all information about both M and I. Naturally, the former is weaker than the latter. Also, the former is related to the standard security notion for the IBE and ABE, and the latter for anonymous IBE [12,9,20] and searchable encryption on encrypted data. In particular, the attribute-hiding property guarantees that the ciphertext reveals no information about the associated I to even a holder of $\mathsf{SK}_{f_1}, \ldots, \mathsf{SK}_{f_\ell}$, except evaluation results $f_1(I), \ldots, f_\ell(I)$ (and the message M only if $f_i(I) = 1$ for some i). Thus, if the ciphertext is associated with only I, the attribute-hiding property allows for predicate encryption where private keys reveal only evaluation results.

Predicate encryption can be realized into various constructions according to how $(\Sigma, \mathcal{F})$ is defined (as mentioned above). Currently, the evaluation of inner products is considered to be the most important class $\mathcal{F}$ of predicates, as the result of [15]. The class $\mathcal{F}$ of inner-product predicates is defined as: if we take $\Sigma = (\mathbb{Z}_p)^n$ for some integer p and n, a ciphertext (and a private key) is associated with a vector $\overrightarrow{x}$ (and $\overrightarrow{y}$) in $(\mathbb{Z}_p)^n$ and the ciphertext can be opened by the private key $\mathsf{SK}_{\overrightarrow{y}}$ where $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$ if and only if $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$.[1] The reason why the inner-product predicate is important is that, as shown in [15], the predicate suffices to express a wide class of predicates including CNF/DNF formulae and polynomial evaluations. Thus, realizing a predicate encryption supporting inner-products, which we call *inner-product encryption* (IPE) for short, is the main goal in predicate encryption.

Until now, there have been three IPE schemes suggested [15,17,16]. The first IPE scheme [15], which is presented by Katz, Sahai, and Waters, shows how inner-product predicates can be realized in composite-order groups, and describes how an IPE scheme can be used as a building block to construct a wide class of predicates. The second IPE scheme [17], which is suggested by Okamoto and Takashima, introduces a new method of using n-dimensional vector spaces in prime order groups, and shows how a delegation functionality is realized in an IPE scheme. The third IPE scheme [16] presents a fully secure IPE scheme in prime-order groups. However, although all the previous IPE constructions show elegant techniques to achieve attribute-hiding properties for IPE schemes, a drawback is that the security of their schemes is all based on new non-standard assumptions. Naturally, a question is to construct an IPE scheme that is secure under well-known standard assumptions, even if it is selectively secure. Such an IPE scheme has not been proposed yet.

1.1 Our results

In this paper, we present a new IPE scheme that is attribute-hiding under the Decision Bilinear Diffie-Hellman (BDH) and Decision Linear assumptions. Our construction uses prime order groups equipped with a bilinear map (i.e., pairing) and furthermore works in both symmetric and asymmetric bilinear maps. To achieve our goal, we introduce a new technique which may be useful for achieving the attribute-hiding and similar properties. Roughly speaking, as in prior works [15,17,16], our technique also uses random elements to mask terms related

[1] $\langle \overrightarrow{x}, \overrightarrow{y} \rangle$ denotes the inner product $\sum_{i=1}^{n} x_i \cdot y_i \bmod p$ for two vectors $\overrightarrow{x} = (x_1, \ldots, x_n)$ and $\overrightarrow{y} = (y_1, \ldots, y_n)$ in $(\mathbb{Z}_p)^n$.

to attributes in a ciphertext. However, a difference occurs when the random elements (for masking) are canceled out in decryption procedures. The KSW scheme used the algebraic property that computing bilinear maps over different subgroups is always an identity, and the two schemes [17,16] used the algebraic property that inner-products of orthogonal vectors are always zero (in exponents). To obtain a similar cancellation effect, our technique is to generate two relative pairings (for one random element) which are positive and negative, respectively, and remove them. In Sect. 3, we will describe the new technique in detail using our IPE scheme.

Based on the result of [15], our IPE scheme can be extended to construct predicate encryption schemes supporting different types of predicates. Since the underlying IPE scheme is attribute-hiding (i.e., secure) under standard assumptions, so are all the extended schemes including a new anonymous IBE scheme and hidden-vector encryption (HVE) scheme. In particular, the new HVE scheme is the first such scheme that is secure under standard assumptions and also supports conjunctions of *equality*, comparison, and subset predicates. Prior to our result, the HVE scheme of Iovino and Persiano [14] is the only scheme that achieves same security as in ours, but their scheme does not support an equality predicate which is essential for expressing more complex predicates.

1.2 Organization

In Sect. 2, we define the notion of an IPE scheme and its security model. In Sect. 3, we present a useful variant of an IPE scheme, called a predicate-only IPE scheme. We also discuss ideas behind our construction by comparing the KSW scheme with ours. In Sect. 4, we suggest a new IPE scheme that works in prime order groups. All security proofs for our proposed schemes are given without using random oracle heuristics.

2 Preliminaries

2.1 Inner-product encryption

Let Σ be the set of attributes involving vectors $\overrightarrow{v}$ of dimension n, and $\mathcal{F}$ be the class of predicates involving inner-products over vectors, i.e., $\mathcal{F} = \{f_{\overrightarrow{x}} \,|\, \overrightarrow{x} \in \Sigma\}$ such that $f_{\overrightarrow{x}}(\overrightarrow{y}) = 1$ iff $\langle \overrightarrow{x}, \overrightarrow{y} \rangle = 0$. An inner-product encryption (IPE) scheme for the class of predicate $\mathcal{F}$ over the set of attributes Σ consists of four algorithms as follows:

Setup: takes as input the security parameter λ and the dimension n of vectors. It outputs a public key PK and a secret key SK.
Encrypt: takes as input the public key PK, a vector $\overrightarrow{x} \in \Sigma$, and a message $M \in \mathcal{M}$ in some associated message space $\mathcal{M}$. It outputs a ciphertext $\mathsf{CT} \leftarrow \texttt{Encrypt}(\mathsf{PK}, \overrightarrow{x}, M)$.
KeyGen: takes as input the secret key SK and a vector $\overrightarrow{v} \in \Sigma$. It outputs a private key $\mathsf{SK}_{\overrightarrow{v}} \leftarrow \texttt{KeyGen}(\mathsf{SK}, \overrightarrow{v})$.
Decrypt: takes as input a private key $\mathsf{SK}_{\overrightarrow{v}}$ and a ciphertext CT. It outputs either a message M if $f_{\overrightarrow{x}}(\overrightarrow{v}) = 1$, i.e., $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = 0$, or the distinguished symbol $\perp$ if $f_{\overrightarrow{x}}(\overrightarrow{v}) = 0$.

For correctness, we require that for all n, all $(\mathsf{PK}, \mathsf{SK})$ generated by $\texttt{Setup}(\lambda, n)$, all $\overrightarrow{v}, \overrightarrow{x} \in \Sigma$, and any private key $\mathsf{SK}_{\overrightarrow{v}} \leftarrow \mathsf{KeyGen}(\mathsf{SK}, \overrightarrow{v})$:

- If $f_{\overrightarrow{x}}(\overrightarrow{v}) = 1$,
  then $\texttt{Decrypt}(\mathsf{SK}_{\overrightarrow{v}}, \texttt{Encrypt}(\mathsf{PK}, \overrightarrow{x}, M)) = M$.

- If $f_{\vec{x}}(\vec{v}) = 0$,
  then $\texttt{Decrypt}(\mathsf{SK}_{\vec{v}}, \texttt{Encrypt}(\mathsf{PK}, \vec{x}, M)) = \perp$ with all but negligible probability.

We further consider a variant of the above, called a *predicate-only* IPE scheme. Compared with the above IPE scheme, the differences are that an encryption algorithm takes as input only a vector $\vec{x}$ and a corresponding decryption algorithm outputs $\texttt{Decrypt}(\mathsf{SK}_{\vec{v}}, \texttt{Encrypt}(\mathsf{PK}, \vec{x})) = f_{\vec{x}}(\vec{v})$ (except possibly with negligible probability). As stated in [15], both the IPE scheme and the predicate-only IPE scheme achieve computational correctness.

2.2 Security model for IPE

Following [15], we define the security, i.e., attribute-hiding property, of the IPE scheme. The security is defined by the following game interacted between an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$. We assume that $(\Sigma, \mathcal{F})$ are given to both $\mathcal{A}$ and $\mathcal{C}$ in advance.

Init: $\mathcal{A}$ outputs two vectors $\vec{x}, \vec{y} \in \Sigma$.

Setup: $\mathcal{C}$ runs $\texttt{Setup}$ to obtain the public key PK and secret key SK. $\mathcal{A}$ is given PK.

Query Phase 1: $\mathcal{A}$ adaptively issues private key queries for any vectors $\vec{v}_1, \ldots, \vec{v}_\ell \in \Sigma$, subject to the restriction that, for all i, $\langle \vec{v}_i, \vec{x} \rangle = 0$ if and only if $\langle \vec{v}_i, \vec{y} \rangle = 0$. $\mathcal{C}$ responds with $\mathsf{SK}_{\vec{v}_i} \leftarrow \texttt{KeyGen}(\mathsf{SK}, \vec{v}_i)$.

Challenge: $\mathcal{A}$ outputs two messages $M_0, M_1 \in \mathcal{M}$. If $M_0 \neq M_1$, it is required that $\langle \vec{v}_i, \vec{x} \rangle \neq 0 \neq \langle \vec{v}_i, \vec{y} \rangle$ for all queried vectors $\vec{v}_i$. $\mathcal{C}$ picks a random bit $b \in \{0, 1\}$. If $b = 0$ then $\mathcal{C}$ gives $\mathsf{CT} \leftarrow \texttt{Encrypt}(\mathsf{PK}, \vec{x}, M_0)$ to $\mathcal{A}$, and if $b = 1$ then $\mathcal{C}$ gives $\mathsf{CT} \leftarrow \texttt{Encrypt}(\mathsf{PK}, \vec{y}, M_1)$ to $\mathcal{A}$.

Query Phase 2: $\mathcal{A}$ continues to issue private key queries for additional vectors, subject to the same restriction as before.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$. $\mathcal{A}$ wins if $b = b'$.

As usual, the advantage of $\mathcal{A}$ in attacking the IPE scheme is defined as $\mathsf{Adv}(\mathcal{A}) = \left|\Pr\left[b = b'\right] - \frac{1}{2}\right|$. In the case where $M_0 \neq M_1$, $\mathcal{A}$ is not permitted to issue private key queries for vectors $\vec{v}_i$ such that $\langle \vec{v}_i, \vec{x} \rangle = 0 = \langle \vec{v}_i, \vec{y} \rangle$, throughout two query phases. Otherwise, $\mathcal{A}$ can obtain a private key $\mathsf{SK}_{\vec{v}_i}$ for such a vector $\vec{v}_i$ and decrypt the challenge ciphertext using $\mathsf{SK}_{\vec{v}_i}$. Note that in the case where $M_0 = M_1$ (as well as in case of the predicate-only IPE scheme), this restriction is not required.

Definition 1 We say that an IPE scheme is attribute-hiding if for all polynomial time adversaries $\mathcal{A}$, we have that $\mathsf{Adv}(\mathcal{A})$ is negligible.

The game above can be used to define the attribute-hiding property for the predicate-only IPE scheme if the adversary does not output messages in the challenge phase. Naturally, the challenge ciphertext is given to $\mathcal{A}$ as: if $b = 0$ then $\mathsf{CT} \leftarrow \texttt{Encrypt}(\mathsf{PK}, \vec{x})$ and if $b = 1$ then $\mathsf{CT} \leftarrow \texttt{Encrypt}(\mathsf{PK}, \vec{y})$. Under this slight modification as well as similar $\mathsf{Adv}(\mathcal{A})$ to the one above, we say that a predicate-only IPE scheme is attribute-hiding if for all polynomial time adversaries $\mathcal{A}$, we have that $\mathsf{Adv}(\mathcal{A})$ is negligible.

2.3 Bilinear maps and complexity assumptions

Bilinear maps: We follow the notation in [6,3]. Let $\mathbb{G}$ and $\mathbb{G}_T$ be two (multiplicative) cyclic groups of prime order p. We assume that g is a generator of $\mathbb{G}$. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a function that has the following properties:

1. Bilinear: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: $e(g, g) \neq 1$.
3. Computable: there is an efficient algorithm to compute the map e.

Then, we say that the map e is a bilinear map in $\mathbb{G}$. Note that $e(,)$ is symmetric since $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

The Decision Bilinear Diffie-Hellman (BDH) Problem: The Decision BDH problem [6] is defined as follows: given $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$ as input, determine whether $Z = e(g, g)^{abc}$ or Z is random in $\mathbb{G}_T$.

The Decision Linear Problem: The Decision Linear problem [4] was originally stated as follows: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_3+z_4}$ or Z is random in $\mathbb{G}$. We consider an equivalently modified version such as: given $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$ as input, determine whether $Z = g^{z_2(z_3+z_4)}$ or Z is random in $\mathbb{G}$. This was already used in [9].

Definition 2 We say that the {Decision BDH, Decision Linear} assumption holds in $\mathbb{G}$ if the advantage of any polynomial time algorithm in solving the {Decision BDH, Decision Linear} problem is negligible.

3 Our predicate-only inner-product encryption scheme

In this section, we present a predicate-only IPE scheme that is attribute-hiding under the Decision Linear assumption. We describe in the next section how to extend the present scheme to obtain our full-fledged IPE scheme.

3.1 Scheme

We assume that $\Sigma = (\mathbb{Z}_p)^n$ for some positive integer n is the set of attributes. For a vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in \Sigma$, each component v_i belongs to the set $\mathbb{Z}_p$. If we want to handle arbitrary attributes $v_i \in \{0, 1\}^*$, we can use a collusion resistant hash function $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$ to hash each v_i prior to key generation and encryption. The predicate-only IPE scheme works as follows:

Setup (λ, n): Given a security parameter $\lambda \in \mathbb{Z}^+$, the setup algorithm runs $\mathcal{G}(\lambda)$ to obtain a tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$. The algorithm picks a random generator $g \in \mathbb{G}$, random exponents δ_1, $\delta_2, \theta_1, \theta_2, \{w_{1,i}\}_{i=1}^n, \{t_{1,i}\}_{i=1}^n, \{f_{1,i}, f_{2,i}\}_{i=1}^n, \{h_{1,i}, h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$. It picks a random $\Omega \in \mathbb{Z}_p$ and obtains $\{w_{2,i}\}_{i=1}^n, \{t_{2,i}\}_{i=1}^n$ under constraints that

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \qquad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

For $i = 1, \ldots, n$, the algorithm sets

$$W_{1,i} = g^{w_{1,i}}, \quad W_{2,i} = g^{w_{2,i}}, \quad T_{1,i} = g^{t_{1,i}}, \quad T_{2,i} = g^{t_{2,i}},$$
$$F_{1,i} = g^{f_{1,i}}, \quad F_{2,i} = g^{f_{2,i}}, \quad H_{1,i} = g^{h_{1,i}}, \quad H_{2,i} = g^{h_{2,i}}.$$

Next, it sets

$$U_1 = g^{\delta_1},\ \ U_2 = g^{\delta_2},\ \ V_1 = g^{\theta_1},\ \ V_2 = g^{\theta_2},\ \ g_1 = g^{\Omega}.$$

The public key PK (including the description of $(p, \mathbb{G}, \mathbb{G}_T, e)$) and secret key SK are set to be

$$\begin{aligned}\mathsf{PK} = \big(g,\ g_1,\ &\big\{W_{1,i}, W_{2,i}, F_{1,i}, F_{2,i}\big\}_{i=1}^{n},\\ &\big\{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\big\}_{i=1}^{n},\ \ \{U_i, V_i\}_{i=1}^{2}\big) \in \mathbb{G}^{8n+6},\\ \mathsf{SK} = \big(\big\{&w_{1,i}, w_{2,i}, f_{1,i}, f_{2,i}, t_{1,i}, t_{2,i}, h_{1,i}, h_{2,i}\big\}_{i=1}^{n},\ \ \{\delta_i, \theta_i\}_{i=1}^{2}\big) \in \mathbb{Z}_p^{8n+4}.\end{aligned}$$

Encrypt(PK, $\overrightarrow{x}$): To encrypt a vector $\overrightarrow{x} = (x_1, \ldots, x_n) \in (\mathbb{Z}_p)^n$ under the public key PK, the encryption algorithm picks random exponents $s_1, s_2, s_3, s_4 \in \mathbb{Z}_p$ and computes the ciphertext CT as follows:

$$\Bigg(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\Big\}_{i=1}^{n} \Bigg) \in \mathbb{G}^{4n+2}.$$

KeyGen(SK, $\overrightarrow{v}$): To create a private key $\mathsf{SK}_{\overrightarrow{v}}$ for a vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p)^n$, the key generation algorithm first picks random exponents $\lambda_1, \lambda_2, \{r_i\}_{i=1}^{n}, \{\phi_i\}_{i=1}^{n}$ in $\mathbb{Z}_p$. The algorithm computes the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n}) \in \mathbb{G}^{4n+2}$ as follows:

$$\begin{aligned}&\big\{K_{1,i} = g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}},\ \ K_{2,i} = g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\big\}_{i=1}^{n},\\ &\big\{K_{3,i} = g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}},\ \ K_{4,i} = g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\big\}_{i=1}^{n},\\ &K_A = \prod_{i=1}^{n} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}}, \qquad K_B = \prod_{i=1}^{n} g^{-(r_i + \phi_i)}.\end{aligned}$$

Decrypt($\mathsf{CT}, \mathsf{SK}_{\overrightarrow{v}}$): To decrypt a ciphertext $\mathsf{CT} = (A, B, \{C_{1,i}, C_{2,i}\}_{i=1}^{n}, \{C_{3,i}, C_{4,i}\}_{i=1}^{n})$ using a private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$, the decryption algorithm outputs 1 iff

$$e(A, K_A) \cdot e(B, K_B) \textstyle\prod_{i=1}^{n} \big[e(C_{1,i}, K_{1,i}) \cdot e(C_{2,i}, K_{2,i}) \cdot e(C_{3,i}, K_{3,i}) \cdot e(C_{4,i}, K_{4,i})\big] \stackrel{?}{=} 1.$$

3.2 Correctness

Assume the ciphertext CT is well-formed for the vector $\overrightarrow{x} = (x_1, \ldots, x_n)$. Then, we first check the following calculation.

$$\begin{aligned}&\prod_{i=1}^{n} e(C_{1,i}, K_{1,i}) e(C_{2,i}, K_{2,i}) e(C_{3,i}, K_{3,i}) e(C_{4,i}, K_{4,i})\\ &\quad= \prod_{i=1}^{n} e\left(g^{w_{1,i}s_1} g^{f_{1,i}s_2} g^{\delta_1 x_i s_3},\ g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right) \cdot e\left(g^{w_{2,i}s_1} g^{f_{2,i}s_2} g^{\delta_2 x_i s_3},\ g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right)\\ &\qquad \cdot e\left(g^{t_{1,i}s_1} g^{h_{1,i}s_2} g^{\theta_1 x_i s_4},\ g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right) \cdot e\left(g^{t_{2,i}s_1} g^{h_{2,i}s_2} g^{\theta_2 x_i s_4},\ g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right)\\ &\quad= \prod_{i=1}^{n} e\left(g^{w_{1,i}s_1},\ g^{-\delta_2 r_i}\right) \cdot e\left(g^{f_{1,i}s_2},\ g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right) \cdot e\left(g^{\delta_1 x_i s_3},\ g^{\lambda_1 v_i w_{2,i}}\right)\end{aligned}$$

$$
\begin{aligned}
&\cdot e\left(g^{w_{2,i}s_1},\ g^{\delta_1 r_i}\right)\cdot e\left(g^{f_{2,i}s_2},\ g^{\delta_1 r_i}g^{-\lambda_1 v_i w_{1,i}}\right)\cdot e\left(g^{\delta_2 x_i s_3},\ g^{-\lambda_1 v_i w_{1,i}}\right)\\
&\cdot e\left(g^{t_{1,i}s_1},\ g^{-\theta_2\phi_i}\right)\cdot e\left(g^{h_{1,i}s_2},\ g^{-\theta_2\phi_i}g^{\lambda_2 v_i t_{2,i}}\right)\cdot e\left(g^{\theta_1 x_i s_4},\ g^{\lambda_2 v_i t_{2,i}}\right)\\
&\cdot e\left(g^{t_{2,i}s_1},\ g^{\theta_1\phi_i}\right)\cdot e\left(g^{h_{2,i}s_2},\ g^{\theta_1\phi_i}g^{-\lambda_2 v_i t_{1,i}}\right)\cdot e\left(g^{\theta_2 x_i s_4},\ g^{-\lambda_2 v_i t_{1,i}}\right)\\
&=\prod_{i=1}^{n} e\left(g^{-\delta_2 w_{1,i}},\ g^{r_i s_1}\right)\cdot e\left(g^{s_2},\ (g^{-\delta_2 r_i}g^{\lambda_1 v_i w_{2,i}})^{f_{1,i}}\right)\cdot e(g,g)^{\lambda_1\delta_1 w_{2,i}\cdot x_i v_i\cdot s_3}\\
&\cdot e\left(g^{\delta_1 w_{2,i}},\ g^{r_i s_1}\right)\cdot e\left(g^{s_2},\ (g^{\delta_1 r_i}g^{-\lambda_1 v_i w_{1,i}})^{f_{2,i}}\right)\cdot e(g,g)^{-\lambda_1\delta_2 w_{1,i}\cdot x_i v_i\cdot s_3}\\
&\cdot e\left(g^{-\theta_2 t_{1,i}},\ g^{\phi_i s_1}\right)\cdot e\left(g^{s_2},\ (g^{-\theta_2\phi_i}g^{\lambda_2 v_i t_{2,i}})^{h_{1,i}}\right)\cdot e(g,g)^{\lambda_2\theta_1 t_{2,i}\cdot x_i v_i\cdot s_4}\\
&\cdot e\left(g^{\theta_1 t_{2,i}},\ g^{\phi_i s_1}\right)\cdot e\left(g^{s_2},\ (g^{\theta_1\phi_i}g^{-\lambda_2 v_i t_{1,i}})^{h_{2,i}}\right)\cdot e(g,g)^{-\lambda_2\theta_2 t_{1,i}\cdot x_i v_i\cdot s_4}\\
&=\prod_{i=1}^{n} e\left(g^{\delta_1 w_{2,i}-\delta_2 w_{1,i}},\, g^{r_i s_1}\right)\cdot e\left(g^{\theta_2 t_{1,i}-\theta_1 t_{2,i}},\, g^{\phi_i s_1}\right)\cdot e\left(g^{s_2},\, K_{1,i}^{f_{1,i}}K_{2,i}^{f_{2,i}}K_{3,i}^{h_{1,i}}K_{4,i}^{h_{2,i}}\right)\\
&\cdot e(g,g)^{[\lambda_1(\delta_1 w_{2,i}-\delta_2 w_{1,i})s_3+\lambda_2(\theta_1 t_{2,i}-\theta_2 t_{1,i})s_4]\cdot x_i v_i}\\
&=\prod_{i=1}^{n} e\left(g^{\Omega},\, g^{r_i s_1}\right)\cdot e\left(g^{\Omega},\, g^{\phi_i s_1}\right)\cdot e\left(g^{s_2},\, K_{1,i}^{f_{1,i}}K_{2,i}^{f_{2,i}}K_{3,i}^{h_{1,i}}K_{4,i}^{h_{2,i}}\right)\cdot e(g,g)^{(\lambda_1\Omega s_3+\lambda_2\Omega s_4)x_i v_i}\\
&=e\left(g^{\Omega s_1},\prod_{i=1}^{n} g^{r_i+\phi_i}\right)\cdot e\left(g^{s_2},\prod_{i=1}^{n} K_{1,i}^{f_{1,i}}K_{2,i}^{f_{2,i}}K_{3,i}^{h_{1,i}}K_{4,i}^{h_{2,i}}\right)\cdot e(g,g)^{\Omega(\lambda_1 s_3+\lambda_2 s_4)\cdot\langle\overrightarrow{x},\overrightarrow{v}\rangle}.
\end{aligned}
$$

Recall that $A = g^{s_2}$, $B = g^{\Omega s_1}$,

$$K_A = \textstyle\prod_{i=1}^{n} K_{1,i}^{-f_{1,i}}K_{2,i}^{-f_{2,i}}K_{3,i}^{-h_{1,i}}K_{4,i}^{-h_{2,i}},\ \ K_B = \prod_{i=1}^{n} g^{-(r_i+\phi_i)}.$$

When multiplying two pairings $e(A, K_A)\cdot e(B, K_B)$ with the result above, we then have the following equation

$$
\begin{aligned}
e(A, K_A)\cdot e(B, K_B)\cdot\prod_{i=1}^{n} e(C_{1,i}, K_{1,i})\cdot e(C_{2,i}, K_{2,i})&\\
\cdot e(C_{3,i}, K_{3,i})\cdot e(C_{4,i}, K_{4,i}) = e(g,g)^{\Omega(\lambda_1 s_3+\lambda_2 s_4)\langle\overrightarrow{x},\overrightarrow{v}\rangle}&.
\end{aligned}
$$

Thus, the above evaluates 1 if $\langle\overrightarrow{x},\overrightarrow{v}\rangle = 0$ in $\mathbb{Z}_p$. If $\langle\overrightarrow{x},\overrightarrow{v}\rangle \neq 0$ in $\mathbb{Z}_p$, then there is only one case that $\lambda_1 s_3 + \lambda_2 s_4 = 0$ in $\mathbb{Z}_p$. In that case, the probability of being the identity is at most $1/p$ as the exponential values λ_1, λ_2, s_3, and s_4 are all chosen randomly in $\mathbb{Z}_p$.

3.3 Discussion

At a high level, the encryption technique of our predicate-only IPE scheme is very similar to that of the KSW scheme [15]. We explain the idea behind our technique by comparing ours with [15]. In the IPE scheme, the ciphertext is associated with a vector $\overrightarrow{x}$ in such a way that it reveals nothing about $\overrightarrow{x}$ to a computationally bounded adversary. To do this, we make use of the masking elements $\{W_{1,i}^{s_1}, W_{2,i}^{s_1}, T_{1,i}^{s_1}, T_{2,i}^{s_1}\}$ to hide each component x_i of a vector $\overrightarrow{x}$. Then, for instance, one ciphertext term $C_{1,i}$ for x_i is on the form of $W_{1,i}^{s_1}F_{1,i}^{s_2}U_1^{x_i s_3}$. One might think that even if $W_{1,i}^{s_1}$ is not used, the resulting term $F_{1,i}^{s_2}U_1^{x_i s_3}$ is enough for hiding x_i component. In that case, however, if $x_i = 0$ in $\mathbb{Z}_p$, then the term becomes $F_{1,i}^{s_2}$ so that it can be tested as $e(A, F_{1,i}) \overset{?}{=} e(g, C_{1,i})$ using bilinear maps. It seems that the multiplied form of $W_{1,i}^{s_1} F_{1,i}^{s_2} U_1^{x_i s_3}$ is not easily tested even if we use prime order groups equipped with a *symmetric* bilinear map. This is only for intuition and, in proving security, we show that the term $W_{1,i}^{s_1} F_{1,i}^{s_2} U_1^{x_i s_3}$ is computationally indistinguishable from the term $W_{1,i}^{s_1} F_{1,i}^{s_2}$ under the Decision Linear assumption. Compared with the KSW scheme, the masking elements in our scheme correspond to random elements $\{R_i\}$ in $\mathbb{G}_r$, where $\mathbb{G}_r$ is a subgroup of $\mathbb{G} = \mathbb{G}_p \times \mathbb{G}_q \times \mathbb{G}_r$ for three primes p, q, and r. Then, the ciphertext terms including x_i are generated in the form of $R_i h_p^{s_2} g_q^{x_i s_3}$, where $h_p \in \mathbb{G}_p$ and $g_q \in \mathbb{G}_q$. It is easy to see that the random elements $\{R_i\}$ have the same effect as in our scheme. Also, the attribute-hiding property is obtained from the fact that (informally) a random element of any of these subgroups is computationally indistinguishable from a random element of $\mathbb{G}$.

On the other hand, the masking elements have to be canceled out in the decryption procedure. This is easily achieved in the KSW scheme using the fact that $e(g_r, g_p) = 1$ and $e(g_r, g_q) = 1$, where $g_p \in \mathbb{G}_p$, $g_q \in \mathbb{G}_q$, and $g_r \in \mathbb{G}_r$. Roughly speaking, when generating a private key for a vector $\overrightarrow{v}$ in the KSW scheme, the terms including v_i are computed using elements in two groups $\mathbb{G}_p$ and $\mathbb{G}_q$, in the form of $g_p^{r_i} g_q^{\lambda_1 v_i}$. Thus, in computing $e(R_i h_p^{s_2} g_q^{x_i s_3}, g_p^{r_i} g_q^{\lambda_1 v_i})$ during the decryption procedure, the masking element R_i will disappear. Our construction also needs such a similar cancellation effect, and we solve this problem by generating two relative pairing values that are positive and negative, respectively. This can be checked by the following equality

$$
\begin{aligned}
& e(C_{1,i}, K_{1,i}) \cdot e(C_{2,i}, K_{2,i}) \\
&\quad = e\left(g^{w_{1,i} s_1} g^{f_{1,i} s_2} g^{\delta_1 x_i s_3},\ g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}}\right) e\left(g^{w_{2,i} s_1} g^{f_{2,i} s_2} g^{\delta_2 x_i s_3},\ g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\right),
\end{aligned}
$$

where both $e(g^{w_{1,i} s_1}, g^{\lambda_1 v_i w_{2,i}})$ and $e(g^{\delta_1 x_i s_3}, g^{-\delta_2 r_i})$ are canceled out. Additionally, we need to remove $e(g^{w_{1,i} s_1}, g^{-\delta_2 r_i}) \cdot e(g^{w_{2,i} s_1}, g^{\delta_1 r_i})$ that are changed into one pairing as $e(g^{\Omega s_1}, g^{r_i})$. This value is also eliminated by the additional computation of $e(B, K_B)$ in the decryption procedure.

As in the KSW scheme, we can see that our encryption algorithm consists of two parallel sub-systems even if one vector $\overrightarrow{x}$ is encrypted. Such redundancy is not avoidable in our construction as well, and plays an important role in proving security.

3.4 Proof of security

Theorem 1 *Assume the Decision Linear assumption holds in $\mathbb{G}$. Then, our predicate-only IPE scheme is attribute-hiding.*

In the security game described in Sect. 2, the adversary has to output two vectors $\overrightarrow{x}, \overrightarrow{y}$ at the outset of the game. The goal of the adversary is to decide which one of two vectors $\overrightarrow{x}$ and $\overrightarrow{y}$ is associated with the challenge ciphertext. Similarly to [15], our IPE scheme contains two parallel sub-systems. Thus, we can adapt the proof idea of the KSW construction, so that we create a sequence of hybrid games using $\overrightarrow{0}$ vector in intermediate games. In our proofs, we say that the challenge ciphertext is encrypted under $(\overrightarrow{a}, \overrightarrow{b})$ in a sense that $\overrightarrow{a}$ is encrypted in the first sub-system and $\overrightarrow{b}$ in the second sub-system. We consider a sequence of hybrid games as follows:

Game_1: The challenge ciphertext CT_1 is generated under $(\vec{x}, \vec{x})$ as a real encryption. The ciphertext CT_1 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\Big\}_{i=1}^{n}\Big).$$

Game_2: The challenge ciphertext CT_2 is generated under $(\vec{x}, \vec{0})$. The ciphertext CT_2 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2}\Big\}_{i=1}^{n}\Big).$$

Game_3: The challenge ciphertext CT_3 is generated under $(\vec{x}, \vec{y})$. The ciphertext CT_3 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n}\Big).$$

Game_4: The challenge ciphertext CT_4 is generated under $(\vec{0}, \vec{y})$. The ciphertext CT_4 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n}\Big).$$

Game_5: The challenge ciphertext CT_5 is generated under $(\vec{y}, \vec{y})$ as a real encryption. The ciphertext CT_5 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{y_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{y_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n}\Big).$$

The CT_1 corresponding to $\vec{x}$ is the challenge ciphertext given to the adversary as a real encryption (Game_1). On the other hand, the CT_5 corresponding to $\vec{y}$ is the challenge ciphertext given to the adversary as a real encryption (Game_5). We will show that no polynomial time adversary is able to distinguish between Game_1 and Game_5 by proving that the transitions between the sequence of games above are all computationally indistinguishable under the Decision Linear assumption.

Lemma 1 *Under the (t, ϵ)-Decision Linear assumption, there is no adversary running in time t that distinguishes between the games* Game_1 *and* Game_2 *with advantage greater than ϵ.*

Proof Suppose that there exists an adversary $\mathcal{A}$ which can attack our predicate-only encryption scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the Decision Linear game with advantage ϵ. On input $(g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_4}, Z) \in \mathbb{G}^6$, $\mathcal{B}$'s goal is to output 1 if $Z = g^{z_2(z_3+z_4)}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Public Parameters: $\mathcal{B}$ selects random exponents δ_1, δ_2, θ_1, θ_2, $\{w_{1,i}\}_{i=1}^n$, $\{t_{1,i}\}_{i=1}^n$, $\{f_{1,i}\}_{i=1}^n$, $\{f_{2,i}\}_{i=1}^n$, $\{h_{1,i}\}_{i=1}^n$, $\{h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$. $\mathcal{B}$ picks a random $\Omega \in \mathbb{Z}_p$ and obtains $\{w_{2,i}\}_{i=1}^n$, $\{t_{2,i}\}_{i=1}^n$, under constraints that

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \qquad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

$\mathcal{B}$ sets

$$\begin{aligned}
&W_{1,i} = (g^{z_2})^{\delta_1 x_i}(g^{z_1})^{w_{1,i}}, \quad W_{2,i} = (g^{z_2})^{\delta_2 x_i}(g^{z_1})^{w_{2,i}},\\
&F_{1,i} = g^{f_{1,i}}, \quad F_{2,i} = g^{f_{2,i}},\\
&T_{1,i} = (g^{z_2})^{\theta_1 x_i}(g^{z_1})^{t_{1,i}}, \quad T_{2,i} = (g^{z_2})^{\theta_2 x_i}(g^{z_1})^{t_{2,i}},\\
&H_{1,i} = (g^{z_2})^{\theta_1 x_i} g^{h_{1,i}}, \quad H_{2,i} = (g^{z_2})^{\theta_2 x_i} g^{h_{2,i}},
\end{aligned}$$

for $i = 1, \ldots, n$. Next, $\mathcal{B}$ sets

$$U_1 = g^{\delta_1}, \quad U_2 = g^{\delta_2}, \quad V_1 = g^{\theta_1}, \quad V_2 = g^{\theta_2}, \quad g_1 = (g^{z_1})^{\Omega}.$$

Observe that each public key element is independently and uniformly distributed in $\mathbb{Z}_p$ under the following random exponents:

$$\begin{aligned}
&\widetilde{w}_{1,i} = \delta_1 x_i z_2 + w_{1,i} z_1, \quad \widetilde{w}_{2,i} = \delta_2 x_i z_2 + w_{2,i} z_1,\\
&\widetilde{t}_{1,i} = \theta_1 x_i z_2 + t_{1,i} z_1, \quad \widetilde{t}_{2,i} = \theta_2 x_i z_2 + t_{2,i} z_1,\\
&\widetilde{h}_{1,i} = \theta_1 x_i z_2 + h_{1,i}, \quad \widetilde{h}_{2,i} = \theta_2 x_i z_2 + h_{2,i},
\end{aligned}$$

for $i = 1, \ldots, n$ and $\widetilde{\Omega} = \Omega z_1$. Note that

$$\begin{aligned}
\delta_1 \widetilde{w}_{2,i} - \delta_2 \widetilde{w}_{1,i} &= \delta_1(\delta_2 x_i z_2 + w_{2,i} z_1) - \delta_2(\delta_1 x_i z_2 + w_{1,i} z_1)\\
&= (\delta_1 w_{2,i} - \delta_2 w_{1,i}) z_1 = \Omega z_1,\\
\theta_1 \widetilde{t}_{2,i} - \theta_2 \widetilde{t}_{1,i} &= \theta_1(\theta_2 x_i z_2 + t_{2,i} z_1) - \theta_2(\theta_1 x_i z_2 + t_{1,i} z_1)\\
&= (\theta_1 t_{2,i} - \theta_2 t_{1,i}) z_1 = \Omega z_1.
\end{aligned}$$

Key Derivation: $\mathcal{A}$ issues private keys queries for vectors. Consider a query for a vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p)^n$. Here, although Definition 2 restricts the vector $\overrightarrow{v}$ for which $\mathcal{A}$ makes a private key query, we do not need to rely on this restriction in distinguishing between two games Game_1 and Game_2.

$\mathcal{B}$ picks random exponents λ_1, λ_2, $\{r_i\}_{i=1}^n$, and $\{\phi_i\}_{i=1}^n$ in $\mathbb{Z}_p$. In generating a private key $\mathsf{SK}_{\overrightarrow{v}}$, $\mathcal{B}$ implicitly will be setting:

$$\widetilde{r}_i = -\frac{\lambda_2 v_i x_i \cdot z_2}{z_1} + r_i, \qquad \widetilde{\phi}_i = \frac{\lambda_2 v_i x_i \cdot z_2}{z_1} + \phi_i z_1,$$

for $i = 1, \ldots, n$ and

$$\widetilde{\lambda}_1 = -\frac{\lambda_2}{z_1} + \lambda_1, \qquad \widetilde{\lambda}_2 = \frac{\lambda_2}{z_1}.$$

Next, $\mathcal{B}$ generates $K_{1,i}$, $K_{2,i}$, $K_{3,i}$, $K_{4,i}$ elements as follows:

$$\begin{aligned}
K_{1,i} &= (g^{z_1})^{\lambda_1 v_i w_{2,i}} (g^{z_2})^{\lambda_1 v_i x_i \delta_2} g^{-\delta_2 r_i} g^{-\lambda_2 v_i w_{2,i}},\\
K_{2,i} &= (g^{z_1})^{-\lambda_1 v_i w_{1,i}} (g^{z_2})^{-\lambda_1 v_i x_i \delta_1} g^{\delta_1 r_i} g^{\lambda_2 v_i w_{1,i}},\\
K_{3,i} &= (g^{z_1})^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}},\\
K_{4,i} &= (g^{z_1})^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}},
\end{aligned}$$

for all $i = 1, \dots, n$. The K_B element is computed as $K_B = \prod_{i=1}^{n} (g^{z_1})^{\phi_i} g^{r_i}$ and the K_A element is computed as $K_A = \prod_{i=1}^{n} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$. Notice that two exponents $f_{1,i}$ and $f_{2,i}$ are constant numbers known to $\mathcal{B}$, so that computing the component $K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}}$ is done as follows:

$$\begin{aligned} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} &= \left((g^{z_1})^{\lambda_1 v_i w_{2,i}} (g^{z_2})^{\lambda_1 v_i x_i \delta_2} g^{-\delta_2 r_i} g^{-\lambda_2 v_i w_{2,i}}\right)^{-f_{1,i}} \\ &\qquad \cdot \left((g^{z_1})^{-\lambda_1 v_i w_{1,i}} (g^{z_2})^{-\lambda_1 v_i x_i \delta_1} g^{\delta_1 r_i} g^{\lambda_2 v_i w_{1,i}}\right)^{-f_{2,i}} \\ &= (g^{z_1})^{\lambda_1 v_i (w_{1,i} f_{2,i} - w_{2,i} f_{1,i})} \cdot (g^{z_2})^{\lambda_1 v_i x_i (\delta_1 f_{2,i} - \delta_2 f_{1,i})} \\ &\qquad \cdot g^{r_i (\delta_2 f_{1,i} - \delta_1 f_{2,i})} \cdot g^{\lambda_2 v_i (w_{2,i} f_{1,i} - w_{1,i} f_{2,i})}. \end{aligned}$$

Next, to see that the component $K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is also computable, observe that

$$\begin{aligned} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}} &= \left((g^{z_1})^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right)^{-(\theta_1 x_i z_2 + h_{1,i})} \\ &\qquad \cdot \left((g^{z_1})^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right)^{-(\theta_2 x_i z_2 + h_{2,i})} \\ &= (g^{z_1 z_2})^{\theta_1 \theta_2 x_i \phi_i - \theta_1 \theta_2 x_i \phi_i} \cdot (g^{z_1})^{\theta_2 \phi_i h_{1,i} - \theta_1 \phi_i h_{2,i}} \\ &\qquad \cdot (g^{z_2})^{-\lambda_2 \theta_1 x_i v_i t_{2,i} + \lambda_2 \theta_2 x_i v_i t_{1,i}} \cdot g^{-\lambda_2 v_i t_{2,i} h_{1,i} + \lambda_2 v_i t_{1,i} h_{2,i}} \\ &= (g^{z_1})^{\phi_i (\theta_2 h_{1,i} - \theta_1 h_{2,i})} \cdot (g^{z_2})^{-\lambda_2 x_i v_i \Omega} \cdot g^{\lambda_2 v_i (t_{1,i} h_{2,i} - t_{2,i} h_{1,i})}. \end{aligned}$$

Note that the unknown term $g^{z_1 z_2}$ is canceled out regardless of the queried vector $\overrightarrow{v}$. As a result, the K_A element can be computed as follows:

$$\begin{aligned} \prod_{i=1}^{n} \Big[&(g^{z_1})^{\lambda_1 v_i \Upsilon_i + \phi_i (\theta_2 h_{1,i} - \theta_1 h_{2,i})} \cdot (g^{z_2})^{\lambda_1 v_i x_i \Psi_i - \lambda_2 x_i v_i \Omega} \\ &\qquad \cdot g^{-r_i \Psi_i - \lambda_2 v_i \Upsilon_i + \lambda_2 v_i (t_{1,i} h_{2,i} - t_{2,i} h_{1,i})} \Big], \end{aligned}$$

where $\Upsilon_i = w_{1,i} f_{2,i} - w_{2,i} f_{1,i}$ and $\Psi_i = \delta_1 f_{2,i} - \delta_2 f_{1,i}$. $\mathcal{B}$ gives $\mathcal{A}$ the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$ for the queried vector $\overrightarrow{v}$.

The Challenge Ciphertext: To generate a challenge ciphertext, $\mathcal{B}$ picks a random $s_3 \in \mathbb{Z}_p$. In generating the ciphertext, $\mathcal{B}$ implicitly will be setting:

$$\widetilde{s}_1 = z_3, \quad \widetilde{s}_2 = z_4, \quad \widetilde{s}_3 = -z_2 z_3 + s_3.$$

$\mathcal{B}$ computes (A, B) elements as $A = g^{z_4} = g^{s_2}$ and $B = (g^{z_1 z_3})^{\Omega} = g^{\Omega z_1 \cdot z_3} = g_1^{s_1}$. Next, for $i = 1, \dots, n$, $\mathcal{B}$ computes $(C_{1,i}, C_{2,i})$ elements as:

$$\begin{aligned} C_{1,i} &= (g^{z_1 z_3})^{w_{1,i}} \cdot (g^{z_4})^{f_{1,i}} \cdot g^{\delta_1 x_i s_3} \\ &= (g^{z_2 \delta_1 x_i} g^{z_1 w_{1,i}})^{z_3} \cdot (g^{f_{1,i}})^{z_4} \cdot (g^{\delta_1})^{x_i (-z_2 z_3 + s_3)} = W_{1,i}^{\widetilde{s}_1} \cdot F_{1,i}^{\widetilde{s}_2} \cdot U_1^{x_i \widetilde{s}_3}, \\ C_{2,i} &= (g^{z_1 z_3})^{w_{2,i}} \cdot (g^{z_4})^{f_{2,i}} \cdot g^{\delta_2 x_i s_3} \\ &= (g^{z_2 \delta_2 x_i} g^{z_1 w_{2,i}})^{z_3} \cdot (g^{f_{2,i}})^{z_4} \cdot (g^{\delta_2})^{x_i (-z_2 z_3 + s_3)} = W_{2,i}^{\widetilde{s}_1} \cdot F_{2,i}^{\widetilde{s}_2} \cdot U_2^{x_i \widetilde{s}_3}. \end{aligned}$$

Next, $\mathcal{B}$ computes $(C_{3,i}, C_{4,i})$ elements as

$$C_{3,i} = (g^{z_1 z_3})^{t_{1,i}} (g^{z_4})^{h_{1,i}} Z^{\theta_1 x_i}, \quad C_{4,i} = (g^{z_1 z_3})^{t_{2,i}} (g^{z_4})^{h_{2,i}} Z^{\theta_2 x_i},$$

for $i = 1, \dots, n$. If $Z = g^{z_2(z_3+z_4)}$, then

$$\begin{aligned} C_{3,i} &= (g^{z_1z_3})^{t_{1,i}}(g^{z_4})^{h_{1,i}}(g^{z_2(z_3+z_4)})^{\theta_1 x_i} \\ &= g^{(\theta_1 x_i z_2 + t_{1,i} z_1) z_3} \cdot g^{(\theta_1 x_i z_2 + h_{1,i}) z_4} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2}, \\ C_{4,i} &= (g^{z_1z_3})^{t_{2,i}}(g^{z_4})^{h_{2,i}}(g^{z_2(z_3+z_4)})^{\theta_2 x_i} \\ &= g^{(\theta_2 x_i z_2 + t_{2,i} z_1) z_3} \cdot g^{(\theta_2 x_i z_2 + h_{2,i}) z_4} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2}. \end{aligned}$$

In this case, $\mathcal{B}$ is playing Game_2 with $\mathcal{A}$. On the other hand, if $Z = g^{z_2(z_3+z_4)}g^r$ for some (non-zero) random $r \in \mathbb{Z}_p$, then

$$\begin{aligned} C_{3,i} &= (g^{z_1z_3})^{t_{1,i}}(g^{z_4})^{h_{1,i}}(g^{z_2(z_3+z_4)}g^r)^{\theta_1 x_i} \\ &= g^{(\theta_1 x_i z_2 + t_{1,i} z_1) z_3} \cdot g^{(\theta_1 x_i z_2 + h_{1,i}) z_4} \cdot g^{\theta_1 x_i r} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2} \cdot V_1^{x_i \widetilde{s}_4}, \\ C_{4,i} &= (g^{z_1z_3})^{t_{2,i}}(g^{z_4})^{h_{2,i}}(g^{z_2(z_3+z_4)}g^r)^{\theta_2 x_i} \\ &= g^{(\theta_2 x_i z_2 + t_{2,i} z_1) z_3} \cdot g^{(\theta_2 x_i z_2 + h_{2,i}) z_4} \cdot g^{\theta_2 x_i r} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2} \cdot V_2^{x_i \widetilde{s}_4}, \end{aligned}$$

where the exponent r plays a role of $\widetilde{s}_4$. In this case, $\mathcal{B}$ is playing Game_1 with $\mathcal{A}$.

Analysis: As mentioned above, if $Z = g^{z_2(z_3+z_4)}$ the challenge ciphertext is distributed exactly as in Game_2, whereas if $Z = g^{z_2(z_3+z_4)}g^r$ the challenge ciphertext is distributed exactly as in Game_1. It follows that under the Decision Linear assumption, these two games are indistinguishable. □

Lemma 2 *Under the* (t, ϵ)*-Decision Linear assumption, there is no adversary running in time t that distinguishes between the games* Game_2 *and* Game_3 *with advantage greater than* ϵ.

Proof Suppose that there exists an adversary $\mathcal{A}$ which can attack our predicate-only encryption scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the Decision Linear game with advantage ϵ. On input $(g, g^{z_1}, g^{z_2}, g^{z_1z_3}, g^{z_4}, Z) \in \mathbb{G}^6$, $\mathcal{B}$'s goal is to output 1 if $Z = g^{z_2(z_3+z_4)}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Public Parameters: $\mathcal{B}$ selects random exponents δ_1, δ_2, θ_1, θ_2, $\{w_{1,i}\}_{i=1}^n$, $\{t_{1,i}\}_{i=1}^n$, $\{f_{1,i}\}_{i=1}^n$, $\{f_{2,i}\}_{i=1}^n$, $\{h_{1,i}\}_{i=1}^n$, $\{h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$. $\mathcal{B}$ picks a random $\Omega \in \mathbb{Z}_p$ and obtains $\{w_{2,i}\}_{i=1}^n$, $\{t_{2,i}\}_{i=1}^n$, under constraints that

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \qquad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

For $i = 1, \dots, n$, $\mathcal{B}$ sets

$$\begin{aligned} W_{1,i} &= (g^{z_2})^{\delta_1 x_i}(g^{z_1})^{w_{1,i}}, \quad W_{2,i} = (g^{z_2})^{\delta_2 x_i}(g^{z_1})^{w_{2,i}}, \\ F_{1,i} &= g^{f_{1,i}}, \quad F_{2,i} = g^{f_{2,i}}, \\ T_{1,i} &= (g^{z_2})^{\theta_1 y_i}(g^{z_1})^{t_{1,i}}, \quad T_{2,i} = (g^{z_2})^{\theta_2 y_i}(g^{z_1})^{t_{2,i}}, \\ H_{1,i} &= (g^{z_2})^{\theta_1 y_i} g^{h_{1,i}}, \quad H_{2,i} = (g^{z_2})^{\theta_2 y_i} g^{h_{2,i}}, \end{aligned}$$

Compared to the proof of Lemma 1, the differences occur in the setting of $\{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^n$ where y_i is inserted instead of x_i. Next, $\mathcal{B}$ sets

$$U_1 = g^{\delta_1}, \quad U_2 = g^{\delta_2}, \quad V_1 = g^{\theta_1}, \quad V_2 = g^{\theta_2}, \quad g_1 = (g^{z_1})^{\Omega}.$$

Each public key element is independently and uniformly distributed in $\mathbb{Z}_p$ under the following random exponents:

$$\begin{aligned} \widetilde{w}_{1,i} &= \delta_1 x_i z_2 + w_{1,i} z_1, \quad \widetilde{w}_{2,i} = \delta_2 x_i z_2 + w_{2,i} z_1, \\ \widetilde{t}_{1,i} &= \theta_1 y_i z_2 + t_{1,i} z_1, \quad \widetilde{t}_{2,i} = \theta_2 y_i z_2 + t_{2,i} z_1, \\ \widetilde{h}_{1,i} &= \theta_1 y_i z_2 + h_{1,i}, \quad \widetilde{h}_{2,i} = \theta_2 y_i z_2 + h_{2,i}, \end{aligned}$$

for $i = 1, \ldots, n$ and $\widetilde{\Omega} = \Omega z_1$. As before, we see that

$$\begin{aligned} \delta_1 \widetilde{w}_{2,i} - \delta_2 \widetilde{w}_{1,i} &= \delta_1(\delta_2 x_i z_2 + w_{2,i} z_1) - \delta_2(\delta_1 x_i z_2 + w_{1,i} z_1) \\ &= (\delta_1 w_{2,i} - \delta_2 w_{1,i}) z_1 = \Omega z_1, \\ \theta_1 \widetilde{t}_{2,i} - \theta_2 \widetilde{t}_{1,i} &= \theta_1(\theta_2 y_i z_2 + t_{2,i} z_1) - \theta_2(\theta_1 y_i z_2 + t_{1,i} z_1) \\ &= (\theta_1 t_{2,i} - \theta_2 t_{1,i}) z_1 = \Omega z_1. \end{aligned}$$

Key Derivation: $\mathcal{A}$ issues private keys queries for vectors. Consider a query for a vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p)^n$. Here $\mathcal{A}$ should make the private key query under the restriction imposed by Definition 2. That is, $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = 0 \bmod p$ if and only if $\langle \overrightarrow{v}, \overrightarrow{y} \rangle = 0 \bmod p$. We consider two cases according to whether $\langle \overrightarrow{v}, \overrightarrow{x} \rangle$ and $\langle \overrightarrow{v}, \overrightarrow{y} \rangle$ are both zero or whether they are both non-zero.

Case I This is the former case where $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = 0 = \langle \overrightarrow{v}, \overrightarrow{y} \rangle$. $\mathcal{B}$ picks random exponents $\lambda_1, \lambda_2, \{r_i\}_{i=1}^n$, and $\{\phi_i\}_{i=1}^n$ in $\mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\widetilde{r}_i = -\frac{\lambda_2 v_i x_i \cdot z_2}{z_1} + r_i, \quad \widetilde{\phi}_i = \frac{\lambda_2 v_i y_i \cdot z_2}{z_1} + \phi_i z_1,$$

for $i = 1, \ldots, n$ and

$$\widetilde{\lambda}_1 = -\frac{\lambda_2}{z_1} + \lambda_1, \quad \widetilde{\lambda}_2 = \frac{\lambda_2}{z_1}.$$

Then, $\mathcal{B}$ generates $K_{1,i}, K_{2,i}, K_{3,i}, K_{4,i}$ elements as

$$\begin{aligned} K_{1,i} &= (g^{z_1})^{\lambda_1 v_i w_{2,i}} (g^{z_2})^{\lambda_1 v_i x_i \delta_2} g^{-\delta_2 r_i} g^{-\lambda_2 v_i w_{2,i}}, \\ K_{2,i} &= (g^{z_1})^{-\lambda_1 v_i w_{1,i}} (g^{z_2})^{-\lambda_1 v_i x_i \delta_1} g^{\delta_1 r_i} g^{\lambda_2 v_i w_{1,i}}, \\ K_{3,i} &= (g^{z_1})^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}, \\ K_{4,i} &= (g^{z_1})^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}, \end{aligned}$$

for all $i = 1, \ldots, n$. Next, we show that $K_A = \prod_{i=1}^n K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is computable. The component $K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}}$ is computed as in the proof of Lemma 1:

$$\begin{aligned} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} &= (g^{z_1})^{\lambda_1 v_i (w_{1,i} f_{2,i} - w_{2,i} f_{1,i})} \cdot (g^{z_2})^{\lambda_1 v_i x_i (\delta_1 f_{2,i} - \delta_2 f_{1,i})} \\ &\quad \cdot g^{r_i (\delta_2 f_{1,i} - \delta_1 f_{2,i})} \cdot g^{\lambda_2 v_i (w_{2,i} f_{1,i} - w_{1,i} f_{2,i})}. \end{aligned}$$

Next, the component $K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is expanded as

$$\begin{aligned} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}} &= \left((g^{z_1})^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}}\right)^{-(\theta_1 y_i z_2 + h_{1,i})} \\ &\quad \cdot \left((g^{z_1})^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\right)^{-(\theta_2 y_i z_2 + h_{2,i})} \\ &= (g^{z_1 z_2})^{\theta_1 \theta_2 y_i \phi_i - \theta_1 \theta_2 y_i \phi_i} \cdot (g^{z_1})^{\theta_2 \phi_i h_{1,i} - \theta_1 \phi_i h_{2,i}} \\ &\quad \cdot (g^{z_2})^{-\lambda_2 \theta_1 y_i v_i t_{2,i} + \lambda_2 \theta_2 y_i v_i t_{1,i}} \cdot g^{-\lambda_2 v_i t_{2,i} h_{1,i} + \lambda_2 v_i t_{1,i} h_{2,i}} \\ &= (g^{z_1})^{\phi_i (\theta_2 h_{1,i} - \theta_1 h_{2,i})} \cdot (g^{z_2})^{-\lambda_2 y_i v_i \Omega} \cdot g^{\lambda_2 v_i (t_{1,i} h_{2,i} - t_{2,i} h_{1,i})}. \end{aligned}$$

As before, the unknown term $g^{z_1 z_2}$ is canceled out regardless of the queried vector $\overrightarrow{v}$. Also, the term $(g^{z_2})^{-\lambda_2 y_i v_i \Omega}$ will be $(g^{z_2})^{-\lambda_2 \Omega \langle \overrightarrow{y}, \overrightarrow{v} \rangle}$ in the multiplied form of all $K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$. Since $\langle \overrightarrow{y}, \overrightarrow{v} \rangle = 0$, it is also canceled out. As a result, the K_A element can be computed as

$$K_A = \prod_{i=1}^{n} \Big[(g^{z_1})^{\lambda_1 v_i \Upsilon_i + \phi_i (\theta_2 h_{1,i} - \theta_1 h_{2,i})} \cdot (g^{z_2})^{\lambda_1 v_i x_i \Psi_i} \\ \cdot g^{-r_i \Psi_i - \lambda_2 v_i \Upsilon_i + \lambda_2 v_i (t_{1,i} h_{2,i} - t_{2,i} h_{1,i})} \Big],$$

where $\Upsilon_i = w_{1,i} f_{2,i} - w_{2,i} f_{1,i}$ and $\Psi_i = \delta_1 f_{2,i} - \delta_2 f_{1,i}$.

The K_B element is supposed to be calculated as $K_B = \prod_{i=1}^{n} g^{-\widetilde{r}_i + \widetilde{\phi}_i}$, but by using the fact that $\langle \overrightarrow{x}, \overrightarrow{v} \rangle = \langle \overrightarrow{y}, \overrightarrow{v} \rangle = 0$, we obtain

$$\begin{aligned} K_B &= \prod_{i=1}^{n} g^{\left(-\frac{\lambda_2 v_i x_i \cdot z_2}{z_1} + r_i \right) + \left(\frac{\lambda_2 v_i y_i \cdot z_2}{z_1} + \phi_i z_1 \right)} \\ &= \prod_{i=1}^{n} (g^{\frac{z_2}{z_1}})^{\lambda_2 (-\langle \overrightarrow{x}, \overrightarrow{v} \rangle + \langle \overrightarrow{y}, \overrightarrow{v} \rangle)} \cdot (g^{z_1})^{\phi_i} g^{r_i} = \prod_{i=1}^{n} (g^{z_1})^{\phi_i} g^{r_i}. \end{aligned}$$

Thus K_B is also computable. $\mathcal{B}$ then gives $\mathcal{A}$ the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$ for the queried vector $\overrightarrow{v}$.

Case II We consider the latter case where $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = c_x \neq 0$ and $\langle \overrightarrow{v}, \overrightarrow{y} \rangle = c_y \neq 0$. $\mathcal{B}$ picks random exponents $\lambda_1, \lambda_2, \{r_i\}_{i=1}^{n}$, and $\{\phi_i\}_{i=1}^{n}$ in $\mathbb{Z}_p$. $\mathcal{B}$ implicitly sets

$$\widetilde{r}_i = -\frac{c_y \cdot \lambda_2 v_i x_i \cdot z_2}{z_1} + r_i, \ \widetilde{\phi}_i = \frac{c_x \cdot \lambda_2 v_i y_i \cdot z_2}{z_1} + \phi_i z_1,$$

for $i = 1, \ldots, n$ and

$$\widetilde{\lambda}_1 = -\frac{c_y \lambda_2}{z_1} + \lambda_1, \quad \widetilde{\lambda}_2 = \frac{c_x \lambda_2}{z_1}.$$

Then, $\mathcal{B}$ generates $K_{1,i}, K_{2,i}, K_{3,i}, K_{4,i}$ elements as

$$\begin{aligned} K_{1,i} &= (g^{z_1})^{\lambda_1 v_i w_{2,i}} (g^{z_2})^{\lambda_1 v_i x_i \delta_2} g^{-\delta_2 r_i} g^{-c_y \lambda_2 v_i w_{2,i}}, \\ K_{2,i} &= (g^{z_1})^{-\lambda_1 v_i w_{1,i}} (g^{z_2})^{-\lambda_1 v_i x_i \delta_1} g^{\delta_1 r_i} g^{c_y \lambda_2 v_i w_{1,i}}, \\ K_{3,i} &= (g^{z_1})^{-\theta_2 \phi_i} g^{c_x \lambda_2 v_i t_{2,i}}, \\ K_{4,i} &= (g^{z_1})^{\theta_1 \phi_i} g^{-c_x \lambda_2 v_i t_{1,i}}, \end{aligned}$$

for all $i = 1, \ldots, n$. Next, we show that $K_A = \prod_{i=1}^{n} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is computable. The component $K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}}$ is computed as

$$K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} = (g^{z_1})^{\lambda_1 v_i (w_{1,i} f_{2,i} - w_{2,i} f_{1,i})} \cdot (g^{z_2})^{\lambda_1 v_i x_i (\delta_1 f_{2,i} - \delta_2 f_{1,i})} \\ \cdot g^{r_i (\delta_2 f_{1,i} - \delta_1 f_{2,i})} \cdot g^{c_y \lambda_2 v_i (w_{2,i} f_{1,i} - w_{1,i} f_{2,i})},$$

and the component $K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is expanded as

$$\begin{aligned}
&K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}\\
&\quad = \left((g^{z_1})^{-\theta_2\phi_i} g^{c_x\lambda_2 v_i t_{2,i}}\right)^{-(\theta_1 y_i z_2 + h_{1,i})} \cdot \left((g^{z_1})^{\theta_1\phi_i} g^{-c_x\lambda_2 v_i t_{1,i}}\right)^{-(\theta_2 y_i z_2 + h_{2,i})}\\
&\quad (g^{z_1 z_2})^{\theta_1\theta_2 y_i\phi_i - \theta_1\theta_2 y_i \phi_i} \cdot (g^{z_1})^{\theta_2\phi_i h_{1,i} - \theta_1\phi_i h_{2,i}} \cdot (g^{z_2})^{-c_x\lambda_2\theta_1 y_i v_i t_{2,i} + c_x\lambda_2\theta_2 y_i v_i t_{1,i}}\\
&\qquad\quad \cdot g^{-c_x\lambda_2 v_i t_{2,i} h_{1,i} + c_x\lambda_2 v_i t_{1,i} h_{2,i}}\\
&\quad (g^{z_1})^{\phi_i(\theta_2 h_{1,i} - \theta_1 h_{2,i})} \cdot (g^{z_2})^{-c_x\lambda_2 y_i v_i \Omega} \cdot g^{c_x\lambda_2 v_i (t_{1,i} h_{2,i} - t_{2,i} h_{1,i})}.
\end{aligned}$$

As a result, the K_A element can be computed as follows:

$$\begin{aligned}
\prod_{i=1}^{n} \Big[&(g^{z_1})^{\lambda_1 v_i \Upsilon_i + \phi_i(\theta_2 h_{1,i} - \theta_1 h_{2,i})} (g^{z_2})^{\lambda_1 v_i x_i \Psi_i - c_x\lambda_2 y_i v_i \Omega}\\
&\qquad \cdot g^{-r_i\Psi_i - c_y\lambda_2 v_i \Upsilon_i + c_x\lambda_2 v_i(t_{1,i} h_{2,i} - t_{2,i} h_{1,i})} \Big],
\end{aligned}$$

where $\Upsilon_i = w_{1,i} f_{2,i} - w_{2,i} f_{1,i}$ and $\Psi_i = \delta_1 f_{2,i} - \delta_2 f_{1,i}$.

In computing the $K_B = \prod_{i=1}^{n} g^{-\widetilde{r}_i + \widetilde{\phi}_i}$, we use the fact that $c_y\langle \overrightarrow{x}, \overrightarrow{v}\rangle = c_x\langle \overrightarrow{y}, \overrightarrow{v}\rangle$. We then obtain

$$\begin{aligned}
K_B &= \prod_{i=1}^{n} g^{\left(-\frac{c_y\cdot\lambda_2 v_i x_i \cdot z_2}{z_1} + r_i\right) + \left(\frac{c_x\cdot\lambda_2 v_i y_i\cdot z_2}{z_1} + \phi_i z_1\right)}\\
&= \prod_{i=1}^{n} (g^{\frac{z_2}{z_1}})^{\lambda_2(-c_y\langle\overrightarrow{x}, \overrightarrow{v}\rangle + c_x\langle\overrightarrow{y}, \overrightarrow{v}\rangle)} \cdot (g^{z_1})^{\phi_i} g^{r_i} = \prod_{i=1}^{n} (g^{z_1})^{\phi_i} g^{r_i},
\end{aligned}$$

which is also computable. $\mathcal{B}$ then gives $\mathcal{A}$ the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^{n}, \{K_{3,i}, K_{4,i}\}_{i=1}^{n})$ for the queried vector $\overrightarrow{v}$.

The Challenge Ciphertext: $\mathcal{B}$ picks a random $s_3 \in \mathbb{Z}_p$. In generating the ciphertext, $\mathcal{B}$ implicitly sets

$$\widetilde{s}_1 = z_3, \quad \widetilde{s}_2 = z_4, \quad \widetilde{s}_3 = -z_2 z_3 + s_3.$$

$\mathcal{B}$ computes (A, B) elements as $A = g^{z_4} = g^{s_2}$ and $B = (g^{z_1 z_3})^{\Omega} = g^{\Omega z_1\cdot z_3} = g_1^{s_1}$. Next, for $i = 1, \ldots, n$, $\mathcal{B}$ computes $(C_{1,i}, C_{2,i})$ elements as:

$$\begin{aligned}
C_{1,i} &= (g^{z_1 z_3})^{w_{1,i}} \cdot (g^{z_4})^{f_{1,i}} \cdot g^{\delta_1 x_i s_3}\\
&= (g^{z_2\delta_1 x_i} g^{z_1 w_{1,i}})^{z_3} \cdot g^{f_{1,i} z_4} \cdot (g^{\delta_1})^{x_i(-z_2 z_3 + s_3)} = W_{1,i}^{\widetilde{s}_1} \cdot F_{1,i}^{\widetilde{s}_2} \cdot U_1^{x_i\widetilde{s}_3},\\
C_{2,i} &= (g^{z_1 z_3})^{w_{2,i}} \cdot (g^{z_4})^{f_{2,i}} \cdot g^{\delta_2 x_i s_3}\\
&= (g^{z_2\delta_2 x_i} g^{z_1 w_{2,i}})^{z_3} \cdot g^{f_{2,i} z_4} \cdot (g^{\delta_2})^{x_i(-z_2 z_3 + s_3)} = W_{2,i}^{\widetilde{s}_1} \cdot F_{2,i}^{\widetilde{s}_2} \cdot U_2^{x_i\widetilde{s}_3}.
\end{aligned}$$

Next, $\mathcal{B}$ computes $(C_{3,i}, C_{4,i})$ elements as

$$C_{3,i} = (g^{z_1 z_3})^{t_{1,i}} (g^{z_4})^{h_{1,i}} Z^{\theta_1 y_i}, \quad C_{4,i} = (g^{z_1 z_3})^{t_{2,i}} (g^{z_4})^{h_{2,i}} Z^{\theta_2 y_i},$$

for $i = 1, \ldots, n$. If $Z = g^{z_2(z_3+z_4)}$, then

$$\begin{aligned} C_{3,i} &= (g^{z_1 z_3})^{t_{1,i}} (g^{z_4})^{h_{1,i}} (g^{z_2(z_3+z_4)})^{\theta_1 y_i} \\ &= g^{(\theta_1 y_i z_2 + t_{1,i} z_1) z_3} \cdot g^{(\theta_1 y_i z_2 + h_{1,i}) z_4} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2}, \\ C_{4,i} &= (g^{z_1 z_3})^{t_{2,i}} (g^{z_4})^{h_{2,i}} (g^{z_2(z_3+z_4)})^{\theta_2 y_i} \\ &= g^{(\theta_2 y_i z_2 + t_{2,i} z_1) z_3} \cdot g^{(\theta_2 y_i z_2 + h_{2,i}) z_4} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2}. \end{aligned}$$

In this case, $\mathcal{B}$ is playing Game_2 with $\mathcal{A}$. On the other hand, if $Z = g^{z_2(z_3+z_4)} g^r$ for some (non-zero) random $r \in \mathbb{Z}_p$, then

$$\begin{aligned} C_{3,i} &= (g^{z_1 z_3})^{t_{1,i}} (g^{z_4})^{h_{1,i}} (g^{z_2(z_3+z_4)} g^r)^{\theta_1 y_i} \\ &= g^{(\theta_1 y_i z_2 + t_{1,i} z_1) z_3} \cdot g^{(\theta_1 y_i z_2 + h_{1,i}) z_4} \cdot g^{\theta_1 y_i r} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2} \cdot V_1^{y_i \widetilde{s}_4}, \\ C_{4,i} &= (g^{z_1 z_3})^{t_{2,i}} (g^{z_4})^{h_{2,i}} (g^{z_2(z_3+z_4)} g^r)^{\theta_2 y_i} \\ &= g^{(\theta_2 y_i z_2 + t_{2,i} z_1) z_3} \cdot g^{(\theta_2 y_i z_2 + h_{2,i}) z_4} \cdot g^{\theta_2 y_i r} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2} \cdot V_2^{y_i \widetilde{s}_4}, \end{aligned}$$

where the exponent r plays a role of $\widetilde{s}_4$. In this case, $\mathcal{B}$ is playing Game_3 with $\mathcal{A}$.

Analysis: As mentioned above, if $Z = g^{z_2(z_3+z_4)}$ the challenge ciphertext is distributed exactly as in Game_2, whereas if $Z = g^{z_2(z_3+z_4)} g^r$ the challenge ciphertext is distributed exactly as in Game_3. It follows that under the Decision Linear assumption, these two games are indistinguishable. □

Lemma 3 *Under the (t, ϵ)-Decision Linear assumption, there is no adversary running in time t that distinguishes between the games* Game_3 *and* Game_4 *with advantage greater than ϵ.*

Lemma 4 *Under the (t, ϵ)-Decision Linear assumption, there is no adversary running in time t that distinguishes between the games* Game_4 *and* Game_5 *with advantage greater than ϵ.*

As in the KSW construction [15], our proof is also symmetric in that the proof that Game_3 and Game_4 are indistinguishable exactly parallels the proof that Game_2 and Game_3 are indistinguishable, and Game_4 and Game_5 are indistinguishable exactly parallels the proof that Game_1 and Game_2 are indistinguishable. This concludes the proof of Theorem 1.

4 Our inner-product encryption scheme

We now describe our IPE scheme which is attribute-hiding under the Decision BDH and Decision Linear assumptions, by slightly modifying the predicate-only IPE scheme. The main difference is that the present IPE scheme should be allowed to encrypt a message $M \in \mathcal{M}$ as well as a vector $\overrightarrow{v} \in \Sigma$, where we assume that $\Sigma = (\mathbb{Z}_p)^n$ and $\mathcal{M} = \mathbb{G}_T$. To do this, we need to modify each algorithm appropriately. In our scheme, the modified parts are underlined.

4.1 Scheme

Setup(λ, n): Given a security parameter $\lambda \in \mathbb{Z}^+$, the setup algorithm runs $\mathcal{G}(\lambda)$ to obtain a tuple $(p, \mathbb{G}, \mathbb{G}_T, e)$. The algorithm picks a random generator $g \in \mathbb{G}$, random exponents

$\delta_1, \delta_2, \theta_1, \theta_2, \{w_{1,i}\}_{i=1}^n, \{t_{1,i}\}_{i=1}^n, \{f_{1,i}, f_{2,i}\}_{i=1}^n, \{h_{1,i}, h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$. It also picks a random $\underline{g_2 \in \mathbb{G}}$. It picks a random $\Omega \in \mathbb{Z}_p$ and obtains $\{w_{2,i}\}_{i=1}^n$, $\{t_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$ under constraints that

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \qquad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

For $i = 1, \dots, n$, the algorithm sets

$$W_{1,i} = g^{w_{1,i}}, \quad W_{2,i} = g^{w_{2,i}}, \quad T_{1,i} = g^{t_{1,i}}, \quad T_{2,i} = g^{t_{2,i}},$$
$$F_{1,i} = g^{f_{1,i}}, \quad F_{2,i} = g^{f_{2,i}}, \quad H_{1,i} = g^{h_{1,i}}, \quad H_{2,i} = g^{h_{2,i}}.$$

Next, it sets

$$U_1 = g^{\delta_1}, \ U_2 = g^{\delta_2}, \ V_1 = g^{\theta_1}, \ V_2 = g^{\theta_2}, \ g_1 = g^{\Omega}, \ \underline{\Lambda = e(g, g_2)}.$$

The public key PK and secret key SK are set to be

$$\begin{aligned} \mathsf{PK} = \big(&g,\ g_1,\ \{W_{1,i}, W_{2,i}, F_{1,i}, F_{2,i}\}_{i=1}^n,\ \{T_{1,i}, T_{2,i}, H_{1,i}, H_{2,i}\}_{i=1}^n, \\ &\{U_i, V_i\}_{i=1}^2,\ \underline{\Lambda}\big) \in \mathbb{G}^{8n+6} \times \mathbb{G}_T, \\ \mathsf{SK} = \Big(&\big\{w_{1,i}, w_{2,i}, t_{1,i}, t_{2,i}, f_{1,i}, f_{2,i}, h_{1,i}, h_{2,i}\big\}_{i=1}^n,\ \{\delta_i, \theta_i\}_{i=1}^2,\ \underline{g_2}\Big) \in \mathbb{Z}_p^{8n+4} \times \mathbb{G}. \end{aligned}$$

Encrypt(PK, $\overrightarrow{x}$,$\underline{M}$): To encrypt a vector $\overrightarrow{x} = (x_1, \dots, x_n) \in (\mathbb{Z}_p)^n$ and a message $\underline{M \in \mathbb{G}_T}$ under the public key PK, the encryption algorithm picks random exponents $s_1, s_2, s_3, s_4 \in \mathbb{Z}_p$ and computes the ciphertext CT as follows:

$$\begin{aligned} \Big(&g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^n, \\ &\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\Big\}_{i=1}^n,\ \underline{\Lambda^{-s_2} M}\Big) \in \mathbb{G}^{4n+2} \times \mathbb{G}_T. \end{aligned}$$

KeyGen(SK, $\overrightarrow{v}$): To create a private key $\mathsf{SK}_{\overrightarrow{v}}$ for a vector $\overrightarrow{v} = (v_1, \dots, v_n) \in (\mathbb{Z}_p)^n$, the key generation algorithm first picks random exponents $\lambda_1, \lambda_2, \{r_i\}_{i=1}^n, \{\phi_i\}_{i=1}^n$ in $\mathbb{Z}_p$. The algorithm computes the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^n, \{K_{3,i}, K_{4,i}\}_{i=1}^n) \in \mathbb{G}^{4n+2}$ as:

$$\begin{aligned} &\big\{K_{1,i} = g^{-\delta_2 r_i} g^{\lambda_1 v_i w_{2,i}},\ \ K_{2,i} = g^{\delta_1 r_i} g^{-\lambda_1 v_i w_{1,i}}\big\}_{i=1}^n, \\ &\big\{K_{3,i} = g^{-\theta_2 \phi_i} g^{\lambda_2 v_i t_{2,i}},\ \ K_{4,i} = g^{\theta_1 \phi_i} g^{-\lambda_2 v_i t_{1,i}}\big\}_{i=1}^n, \\ &\underline{K_A = g_2 \prod_{i=1}^n K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}},}\quad K_B = \prod_{i=1}^n g^{-(r_i + \phi_i)}. \end{aligned}$$

Decrypt($\mathsf{CT}, \mathsf{SK}_{\overrightarrow{v}}$): To decrypt a ciphertext $\mathsf{CT} = (A, B, \{C_{1,i}, C_{2,i}\}_{i=1}^n, \{C_{3,i}, C_{4,i}\}_{i=1}^n, \underline{D})$ using a private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^n, \{K_{3,i}, K_{4,i}\}_{i=1}^n)$, the decryption algorithm outputs

$$\begin{aligned} M \leftarrow\ &\underline{D} \cdot e(A, K_A) \cdot e(B, K_B) \\ &\cdot \prod_{i=1}^n e(C_{1,i}, K_{1,i}) e(C_{2,i}, K_{2,i}) e(C_{3,i}, K_{3,i}) e(C_{4,i}, K_{4,i}). \end{aligned}$$

4.2 Correctness

Assume the ciphertext CT is well-formed for the vector $\overrightarrow{x} = (x_1, \ldots, x_n)$. Then, we have

$$\begin{aligned}&\underline{D} \cdot e(A, K_A) \cdot e(B, K_B) \prod_{i=1}^{n} e(C_{1,i}, K_{1,i}) e(C_{2,i}, K_{2,i}) e(C_{3,i}, K_{3,i}) e(C_{4,i}, K_{4,i})\\&\quad = \underline{\Lambda^{-s_2} M \cdot e(g, g_2)^{s_2}} \cdot e(g, g)^{\Omega(\lambda_1 s_3 + \lambda_2 s_4) \cdot \langle \overrightarrow{x}, \overrightarrow{v} \rangle}.\end{aligned}$$

Note that the term $e(g, g_2)^{s_2} (= \Lambda^{s_2})$ is generated from the pairing computation of $e(A, K_A) = e(g^{s_2}, g_2 \prod_{i=1}^{n} K_{1,i}^{-f_{1,i}} K_{2,i}^{-f_{2,i}} K_{3,i}^{-h_{1,i}} K_{4,i}^{-h_{2,i}})$. Thus, the result above outputs M if $\langle \overrightarrow{x}, \overrightarrow{v} \rangle = 0$ in $\mathbb{Z}_p$. If $\langle \overrightarrow{x}, \overrightarrow{v} \rangle \neq 0$ in $\mathbb{Z}_p$, then there is only one such case that $\lambda_1 s_3 + \lambda_2 s_4 = 0$ in $\mathbb{Z}_p$ with probability at most $1/p$, as in the predicate-only IPE scheme.

4.3 Proof of security

Theorem 2 *Assume the Decision BDH and Decision Linear assumptions hold in $\mathbb{G}$. Then, our IPE scheme is attribute-hiding.*

To prove the security of the IPE scheme defined in Sect. 2, we consider two cases when $M_0 = M_1$ and when $M_0 \neq M_1$. In the former case, the overall proof can be analogously done by adapting the proof techniques given in the predicate-only IPE scheme. The necessary modification is that the simulator needs to generate the included extra term $D = e(g, g_2)^{-s_2} \cdot M_0$ into the challenge ciphertext. The computation is not a problem because, in our proofs of Lemmas 1 and 2, the exponent s_2 becomes $z_4 \in \mathbb{Z}_p$ and the element $g^{z_4} \in \mathbb{G}$ is also given to the simulator, so that $e(g, g_2)^{-s_2}$ is easily computed as $e(g^{z_4}, g_2^{-1})$ for any random $g_2 \in \mathbb{G}$ chosen by the simulator.

We next consider the latter case where $M_0 \neq M_1$. As pointed out in [15], we need the additional restriction that the adversary is not permitted to make private key queries for vectors $\overrightarrow{v}$ such that $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = 0 = \langle \overrightarrow{v}, \overrightarrow{y} \rangle$, where $\overrightarrow{x}$ and $\overrightarrow{y}$ are two vectors chosen by the adversary at the beginning of the security game. This is required since otherwise the adversary decrypts the challenge ciphertext and recovers an encrypted message using a private key for such a vector $\overrightarrow{v}$. Under the restriction, we create the following hybrid games.

Game_0: The challenge ciphertext CT_0 is generated under $(\overrightarrow{x}, \overrightarrow{x})$ and M_0 as a real encryption. The ciphertext CT_0 is computed as follows:

$$\begin{aligned}\Big(g^{s_2},\ g_1^{s_1},\ &\Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},\\ &\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\Big\}_{i=1}^{n},\ \Lambda^{-s_2} M_0\Big).\end{aligned}$$

Game_1: The challenge ciphertext CT_1 is generated under $(\overrightarrow{x}, \overrightarrow{x})$ and a random message $R \in \mathbb{G}_T$ as a real encryption. The ciphertext CT_1 is computed as follows:

$$\begin{aligned}\Big(g^{s_2},\ g_1^{s_1},\ &\Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},\\ &\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{x_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{x_i s_4}\Big\}_{i=1}^{n},\ R'\Big).\end{aligned}$$

Game_2: The challenge ciphertext CT_2 is generated under $(\vec{x}, \vec{0})$ and a random message $R \in \mathbb{G}_T$. The ciphertext CT_2 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2}\Big\}_{i=1}^{n},\ R'\Big).$$

Game_3: The challenge ciphertext CT_3 is generated under $(\vec{x}, \vec{y})$ and a random message $R \in \mathbb{G}_T$. The ciphertext CT_3 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{x_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{x_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n},\ R'\Big).$$

Game_4: The challenge ciphertext CT_4 is generated under $(\vec{0}, \vec{y})$ and a random message $R \in \mathbb{G}_T$. The ciphertext CT_4 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n},\ R'\Big).$$

Game_5: The challenge ciphertext CT_5 is generated under $(\vec{y}, \vec{y})$ and a random message $R \in \mathbb{G}_T$. The ciphertext CT_5 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{y_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{y_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n},\ R'\Big).$$

Game_6: The challenge ciphertext CT_6 is generated under $(\vec{y}, \vec{y})$ and M_1 as a real encryption. The ciphertext CT_6 is computed as follows:

$$\Big(g^{s_2},\ g_1^{s_1},\ \Big\{W_{1,i}^{s_1} \cdot F_{1,i}^{s_2} \cdot U_1^{y_i s_3},\ W_{2,i}^{s_1} \cdot F_{2,i}^{s_2} \cdot U_2^{y_i s_3}\Big\}_{i=1}^{n},$$
$$\Big\{T_{1,i}^{s_1} \cdot H_{1,i}^{s_2} \cdot V_1^{y_i s_4},\ T_{2,i}^{s_1} \cdot H_{2,i}^{s_2} \cdot V_2^{y_i s_4}\Big\}_{i=1}^{n},\ \Lambda^{-s_2} M_1\Big).$$

We show that no polynomial time adversary is able to distinguish between Game_0 and Game_6 by proving that the transitions between the sequence of games above are all computationally indistinguishable. The proof is completely done only by proving that Game_0 and Game_1 is indistinguishable, which is given by Lemma 5. This is because the indistinguishableness of Game_0 and Game_1 indicates that of Game_5 and Game_6 by the symmetric property of the hybrid games. Also, the transitional games from Game_1 and Game_5 are exactly the same as in the security proof of the predicate-only IPE scheme, except that the extra random term $R' \in \mathbb{G}_T$ is attached.

Lemma 5 *Under the (t, ϵ)-Decision BDH assumption, there is no adversary running in time t that distinguishes between the games* Game_0 *and* Game_1 *with advantage greater than ϵ.*

Proof Suppose that there exists an adversary $\mathcal{A}$ which can attack our full-fledged inner-product encryption scheme with non-negligible advantage ϵ. We describe an algorithm $\mathcal{B}$ which uses $\mathcal{A}$ to solve the Decision BDH game with advantage ϵ. On input $(g, g^a, g^b, g^c, Z) \in \mathbb{G}^4 \times \mathbb{G}_T$, $\mathcal{B}$'s goal is to output 1 if $Z = g^{abc}$ and 0 otherwise. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows:

Public Parameters: $\mathcal{B}$ selects random exponents δ_1, δ_2, θ_1, θ_2, γ, $\{w_{1,i}\}_{i=1}^n$, $\{t_{1,i}\}_{i=1}^n$, $\{f_{1,i}\}_{i=1}^n$, $\{f_{2,i}\}_{i=1}^n$, $\{h_{1,i}\}_{i=1}^n$, $\{h_{2,i}\}_{i=1}^n$ in $\mathbb{Z}_p$. $\mathcal{B}$ picks a random $\Omega \in \mathbb{Z}_p$ and obtains $\{w_{2,i}\}_{i=1}^n$, $\{t_{2,i}\}_{i=1}^n$, under constraints that

$$\Omega = \delta_1 w_{2,i} - \delta_2 w_{1,i}, \qquad \Omega = \theta_1 t_{2,i} - \theta_2 t_{1,i}.$$

In the unlikely event that $\Omega = 0$ in $\mathbb{Z}_p$, $\mathcal{B}$ tries again with new random exponents. $\mathcal{B}$ sets

$$\begin{aligned} &W_{1,i} = g^{w_{1,i}}, \quad W_{2,i} = g^{w_{2,i}}, \quad F_{1,i} = (g^b)^{x_i \delta_1} g^{f_{1,i}}, \quad F_{2,i} = (g^b)^{x_i \delta_2} g^{f_{2,i}}, \\ &T_{1,i} = g^{t_{1,i}}, \quad T_{2,i} = g^{t_{2,i}}, \quad H_{1,i} = (g^b)^{x_i \theta_1} g^{h_{1,i}}, \quad H_{2,i} = (g^b)^{x_i \theta_2} g^{h_{2,i}}, \end{aligned}$$

for $i = 1, \ldots, n$. Next, $\mathcal{B}$ sets

$$\begin{aligned} &U_1 = g^{\delta_1}, \quad U_2 = g^{\delta_2}, \quad V_1 = g^{\theta_1}, \quad V_2 = g^{\theta_2}, \\ &g_1 = g^{\Omega}, \quad \Lambda = e(g^a, g^b)^{-\Omega} \cdot e(g, g)^{\gamma}. \end{aligned}$$

Observe that each public key element is independently and uniformly distributed in $\mathbb{Z}_p$ under the following random exponents:

$$\begin{aligned} \widetilde{f_{1,i}} &= x_i \delta_1 b + f_{1,i}, \qquad \widetilde{f_{2,i}} = x_i \delta_2 b + f_{2,i}, \\ \widetilde{h}_{1,i} &= x_i \theta_1 b + h_{1,i}, \qquad \widetilde{h_{2,i}} = x_i \theta_2 b + h_{2,i}, \end{aligned}$$

for $i = 1, \ldots, n$ and $g_2 = g^{-\Omega ab} g^{\gamma}$.

Key Derivation: $\mathcal{A}$ issues private keys queries for vectors. Consider a query for a vector $\overrightarrow{v} = (v_1, \ldots, v_n) \in (\mathbb{Z}_p)^n$. $\mathcal{A}$ can request the private key query as long as $\langle \overrightarrow{v}, \overrightarrow{x} \rangle = c_x \neq 0$ (we do not need to use the fact that $\langle \overrightarrow{v}, \overrightarrow{y} \rangle \neq 0$ here).

$\mathcal{B}$ picks random exponents λ_1, λ_2, $\{r_i\}_{i=1}^n$, and $\{\phi_i\}_{i=1}^n$ in $\mathbb{Z}_p$. In generating a private key $\mathsf{SK}_{\overrightarrow{v}}$, $\mathcal{B}$ implicitly sets

$$\widetilde{\lambda_1} = \tfrac{1}{2c_x} a + \lambda_1, \qquad \widetilde{\lambda_2} = \tfrac{1}{2c_x} a + \lambda_2.$$

$\mathcal{B}$ generates $K_{1,i}$, $K_{2,i}$, $K_{3,i}$, $K_{4,i}$ elements as

$$\begin{aligned} &K_{1,i} = (g^a)^{v_i w_{2,i}/2c_x} g^{-\delta_2 r_i + \lambda_1 v_i w_{2,i}}, \quad K_{2,i} = (g^a)^{-v_i w_{1,i}/2c_x} g^{\delta_1 r_i - \lambda_1 v_i w_{1,i}}, \\ &K_{3,i} = (g^a)^{v_i t_{2,i}/2c_x} g^{-\theta_2 \phi_i + \lambda_2 v_i t_{2,i}}, \quad K_{4,i} = (g^a)^{-v_i t_{1,i}/2c_x} g^{\theta_1 \phi_i - \lambda_2 v_i t_{1,i}}, \end{aligned}$$

for all $i = 1, \ldots, n$. The K_B element is computed as $K_B = \prod_{i=1}^n g^{r_i + \phi_i}$ and the K_A element will be computed as $K_A = g_2 \prod_{i=1}^n K_{1,i}^{-\widetilde{f}_{1,i}} K_{2,i}^{-\widetilde{f}_{2,i}} K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$. Notice that $K_{1,i}^{-\widetilde{f}_{1,i}} K_{2,i}^{-\widetilde{f}_{2,i}}$ is expanded as

$$\begin{aligned} &K_{1,i}^{-\widetilde{f}_{1,i}} K_{2,i}^{-\widetilde{f}_{2,i}} \\ &\quad = \left((g^a)^{v_i w_{2,i}/2c_x} g^{-\delta_2 r_i + \lambda_1 v_i w_{2,i}}\right)^{x_i \delta_1 b + f_{1,i}} \\ &\qquad\quad \cdot \left((g^a)^{-v_i w_{1,i}/2c_x} g^{\delta_1 r_i - \lambda_1 v_i w_{1,i}}\right)^{x_i \delta_2 b + f_{2,i}} \\ &\quad = (g^{ab})^{v_i x_i \Omega/2c_x} (g^a)^{v_i (w_{2,i} f_{1,i} - w_{1,i} f_{2,i})/2c_x} g^{r_i(\delta_1 f_{2,i} - \delta_2 f_{1,i}) + \lambda_1 v_i (w_{2,i} f_{1,i} - w_{1,i} f_{2,i})}. \end{aligned}$$

Next, we see that the component $K_{3,i}^{-\widetilde{h}_{1,i}} K_{4,i}^{-\widetilde{h}_{2,i}}$ is expanded as

$$\begin{aligned} & K_{3,i}^{-\widetilde{f}_{1,i}} K_{4,i}^{-\widetilde{f}_{2,i}} \\ & = \left((g^a)^{v_i t_{2,i}/2c_x} g^{-\theta_2\phi_i+\lambda_2 v_i t_{2,i}}\right)^{x_i\theta_1 b+h_{1,i}} \\ & \quad \cdot \left((g^a)^{-v_i t_{1,i}/2c_x} g^{\theta_1\phi_i-\lambda_2 v_i t_{1,i}}\right)^{x_i\theta_2 b+h_{2,i}} \\ & = (g^{ab})^{v_i x_i \Omega/2c_x} (g^a)^{v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})/2c_x} g^{\phi_i(\theta_1 h_{2,i}-\theta_2 h_{1,i})+\lambda_2 v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})}. \end{aligned}$$

As a result, the $\prod_{i=1}^n K_{1,i}^{-\widetilde{f}_{1,i}} K_{2,i}^{-\widetilde{f}_{2,i}} K_{3,i}^{-\widetilde{f}_{1,i}} K_{4,i}^{-\widetilde{f}_{2,i}}$ element can be of the following form

$$\begin{aligned} & \prod_{i=1}^n K_{1,i}^{-\widetilde{f}_{1,i}} K_{2,i}^{-\widetilde{f}_{2,i}} K_{3,i}^{-\widetilde{f}_{1,i}} K_{4,i}^{-\widetilde{f}_{2,i}} \\ & \quad = \prod_{i=1}^n (g^{ab})^{v_i x_i \Omega/2c_x + v_i x_i \Omega/2c_x} \cdot \prod_{i=1}^n (g^a)^{-v_i \Upsilon_i/2c_x + v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})/2c_x} \\ & \quad \cdot \prod_{i=1}^n g^{r_i\Psi_i - \lambda_1 v_i \Upsilon_i + \phi_i(\theta_1 h_{2,i}-\theta_2 h_{1,i})+\lambda_2 v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})} \\ & \quad = (g^{ab})^{(\langle \overrightarrow{v}, \overrightarrow{x}\rangle + \langle \overrightarrow{v}, \overrightarrow{x}\rangle)\Omega/2c_x} \cdot \prod_{i=1}^n (g^a)^{-v_i \Upsilon_i/2c_x + v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})/2c_x} \cdots \\ & \quad = (g^{ab})^{\Omega} \cdot \prod_{i=1}^n (g^a)^{-v_i \Upsilon_i/2c_x + v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})/2c_x} \cdots , \end{aligned}$$

where $\Upsilon_i = w_{1,i} f_{2,i} - w_{2,i} f_{1,i}$ and $\Psi_i = \delta_1 f_{2,i} - \delta_2 f_{1,i}$. Since $g_2 = g^{-\Omega ab} g^{\gamma}$, $\mathcal{B}$ can then compute the K_A element as

$$\begin{aligned} K_A = g^{\gamma} & \cdot \prod_{i=1}^n (g^a)^{-v_i \Upsilon_i/2c_x + v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})/2c_x} \\ & \cdot \prod_{i=1}^n g^{r_i\Psi_i - \lambda_1 v_i \Upsilon_i + \phi_i(\theta_1 h_{2,i}-\theta_2 h_{1,i})+\lambda_2 v_i(t_{2,i}h_{1,i}-t_{1,i}h_{2,i})}. \end{aligned}$$

$\mathcal{B}$ gives $\mathcal{A}$ the private key $\mathsf{SK}_{\overrightarrow{v}} = (K_A, K_B, \{K_{1,i}, K_{2,i}\}_{i=1}^n, \{K_{3,i}, K_{4,i}\}_{i=1}^n)$ for the queried vector $\overrightarrow{v}$.

The Challenge Ciphertext: To generate a challenge ciphertext, $\mathcal{B}$ picks random s_1, s_3, $s_4 \in \mathbb{Z}_p$. In generating the ciphertext, $\mathcal{B}$ implicitly will be setting:

$$\widetilde{s}_1 = s_1, \quad \widetilde{s}_2 = c, \quad \widetilde{s}_3 = -bc + s_3, \quad \widetilde{s}_4 = -bc + s_4.$$

$\mathcal{B}$ computes (A, B) elements as $A = g^c = g^{s_2}$ and $B = g^{\Omega s_1} = (g^{\Omega})^{s_1} = g_1^{s_1}$. Next, for $i = 1, \ldots, n$, $\mathcal{B}$ computes $(C_{1,i}, C_{2,i})$ elements as:

$$\begin{aligned} C_{1,i} & = g^{w_{1,i}s_1} \cdot (g^c)^{f_{1,i}} \cdot g^{\delta_1 x_i s_3} \\ & = (g^{w_{1,i}})^{s_1} \cdot \left((g^b)^{x_i\delta_1} g^{f_{1,i}}\right)^c \cdot (g^{\delta_1})^{x_i(-bc+s_3)} = W_{1,i}^{\widetilde{s}_1} \cdot F_{1,i}^{\widetilde{s}_2} \cdot U_1^{x_i\widetilde{s}_3}, \\ C_{2,i} & = g^{w_{2,i}s_1} \cdot (g^c)^{f_{2,i}} \cdot g^{\delta_2 x_i s_3} \\ & = (g^{w_{2,i}})^{s_1} \cdot \left((g^b)^{x_i\delta_2} g^{f_{2,i}}\right)^c \cdot (g^{\delta_2})^{x_i(-bc+s_3)} = W_{2,i}^{\widetilde{s}_1} \cdot F_{2,i}^{\widetilde{s}_2} \cdot U_2^{x_i\widetilde{s}_3}. \end{aligned}$$

Similarly, $\mathcal{B}$ computes $(C_{3,i}, C_{4,i})$ elements as

$$\begin{aligned} C_{3,i} &= g^{t_{1,i}s_1} \cdot (g^c)^{h_{1,i}} \cdot g^{\theta_1 x_i s_4} \\ &= (g^{t_{1,i}})^{s_1} \cdot \left((g^b)^{x_i\theta_1} g^{h_{1,i}}\right)^c \cdot (g^{\theta_1})^{x_i(-bc+s_4)} = T_{1,i}^{\widetilde{s}_1} \cdot H_{1,i}^{\widetilde{s}_2} \cdot V_1^{x_i\widetilde{s}_4}, \\ C_{4,i} &= g^{t_{2,i}s_1} \cdot (g^c)^{h_{2,i}} \cdot g^{\theta_2 x_i s_4} \\ &= (g^{t_{2,i}})^{s_1} \cdot \left((g^b)^{x_i\theta_2} g^{h_{2,i}}\right)^c \cdot (g^{\theta_2})^{x_i(-bc+s_4)} = T_{2,i}^{\widetilde{s}_1} \cdot H_{2,i}^{\widetilde{s}_2} \cdot V_2^{x_i\widetilde{s}_4}, \end{aligned}$$

for $i = 1, \ldots, n$. Next, $\mathcal{B}$ computes the D element as $D = Z^{-\Omega} \cdot e(g, g^c)^\gamma \cdot M_0$.

Analysis: If $Z = e(g, g)^{abc}$ the challenge ciphertext is distributed exactly as in Game_0, whereas if Z is randomly chosen in $\mathbb{G}_T$ the challenge ciphertext is distributed exactly as in Game_1. It follows that under the Decision BDH assumption, these two games are indistinguishable. □

Acknowledgements The author thanks the reviewers for their helpful comments and suggestions for this paper. The author was supported by a grant from the College of Applied Science, Kyung Hee University reserach professor fellowship.

References

1. Abdalla M., Bellare M., Catalano D., Kiltz E., Kohno T., Lange T., Malone-Lee J., Neven G., Paillier P., Shi H.: Searchable encryption revisited: consistency properties, relation to anonymous IBE, and extentions. In: CRYPTO'05, vol. 3621, pp. 205–222. (2005).
2. Bethancourt J., Sahai A., Waters B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334. (2007).
3. Boneh D., Boyen X.: Efficient selective-ID secure identity based encryption without random oracles. In: EUROCRYPT'04, vol. 3027, pp. 223–238. (2004).
4. Boneh D., Boyen X., Shacham H.: Short group signatures, In: CRYPTO'04, vol. 3152, pp. 41–55 (2004).
5. Boneh D., Crescenzo G.D., Octrovsky R., Persiano G., Public key encryption with keyword search. In: EUROCRYPT'04. vol. 3027, pp. 506–522. (2004).
6. Boneh D., Franklin M., Identity-based encryption from the Weil pairing. In: CRYPTO'01, vol. 2139, pp. 213–229. (2001).
7. Boneh D., Gentry C., Hamburg M.: Space-efficient identity based encryption without pairings. In: FOCS'07, pp. 647–657. (2007).
8. Boneh D., Waters B.: Conjunctive, subset, and range queries on encrypted data. In: TCC'07, vol. 4392, pp. 535–554. (2007).
9. Boyen D., Waters B.: Anonymous hierarchical identity-based encryption (without random oracles). In: CRYPTO'06, vol. 4117, pp. 290–307. (2006).
10. Chen L., Cheng Z.: Security proof of Sakai-Kasahara's idenitty based encryption scheme. In: IMA'05, vol. 3796, pp. 442–459. (2005).
11. Cocks C.: An identity based encryption scheme based on quadratic residues. In: IMA'01. (2001).
12. Gentry C.: Practical identity-based encryption without random oracles. In: EUROCRYPT'06, vol. 4004, pp. 445–464. (2006).
13. Goyal V., Pandey O., Sahai A., Waters B.: Attribute-based encryption for fine-grained access control of encrypted data. In: ACM-CCS'06, pp. 89–98. (2006).
14. Iovino V., Persiano G.: Hidden-vector encryption with groups of prime order. In: Pairing'08, vol. 5209, pp. 75–88. (2008).
15. Katz J., Sahai A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. In: EUROCRYPT'08, vol. 4965, pp. 146–162 (2008).
16. Lewko A., Okamoto T., Sahai A., Takashima K., Waters B.: Fully secure functional encryption: attribute-based encryption and (hierarchical) inner-product encryption. To appear in EUROCRYPT'10 (2010).
17. Okamoto T., Takashima K.: Hierarchical predicate encryption for inner-products. In: ASIACRYPT'09, vol. 5912, pp. 214–231 (2009).

18. Ostrovsky R., Sahai A., Waters B.: Attribute-based encryption with non-monotonic access structures. In: ACM-CCS‘07, pp. 195–203. (2007).
19. Sahai A., Waters B.: Fuzzy identity-based encryption. In: EUROCRYPT‘05, vol. 3494, pp. 457–473 (2005).
20. Seo J.H., Kobayashi T., Oukubo M., Suzuki K.: Anonymous hierarchical identity-based enryption with constant size ciphertexts. In: PKC‘09, vol. 5443, pp. 215–234. (2009).
21. Waters B.: Efficient identity-based encryption without random oracles. In: EUROCRYPT‘05, vol. 3494, pp. 114–127. (2005).
22. Waters B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. In: CRYPTO‘09, vol. 5677, pp. 619–636. (2009).