# Functional Encryption for Inner Product with Full Function Privacy

Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay

Department of Mathematics
Indian Institute of Technology Kharagpur
Kharagpur-721302, India
{pratishdatta,ratna,sourav}@maths.iitkgp.ernet.in

**Abstract.** *Functional encryption* (FE) supports constrained decryption keys that allow decrypters to learn specific functions of encrypted messages. In numerous practical applications of FE, confidentiality must be assured not only for the encrypted data but also for the functions for which functional keys are provided. This paper presents a *non-generic simple* private key FE scheme for the *inner product* functionality, also known as *inner product encryption* (IPE). In contrast to the existing similar schemes, our construction achieves the *strongest* indistinguishability-based notion of *function privacy* in the *private key* setting without employing any computationally expensive cryptographic tool or non-standard complexity assumption. Our construction is built in the *asymmetric bilinear pairing group* setting of *prime* order. The security of our scheme is based on the well-studied *Symmetric External Diffie-Hellman* (SXDH) assumption.

**Keywords:** functional encryption, inner product, function privacy, asymmetric bilinear group.

## 1 Introduction

The recent advancement in cloud technology has triggered an emerging trend among individuals and organizations to outsource potentially sensitive private informations to external untrustworthy servers and remotely carry out various computations on the outsourced data at some later point in time by querying the server. *Functional encryption* (FE) is an ambitious vision of modern cryptography that attempts to preserve confidentiality of externally stored data while allowing entities to delegate computations on the outsourced data in such cloud computing platforms. FE supports "restricted" decryption keys, also known as "functional keys", that enable decrypters to learn specific functions of the encrypted data and nothing else. More precisely, in an FE scheme for certain function family $\mathcal{F}$, it is possible to derive functional keys $\mathrm{SK}_f$ for any function $f \in \mathcal{F}$ from a master secret key. Any party given such a functional key $\mathrm{SK}_f$ and a ciphertext $\mathrm{CT}_z$ encrypting some message $z$, should be able to learn $f(z)$ and nothing beyond that about $z$.

A principle focus of research on FE has been to identify what class of functions $\mathcal{F}$ can be supported and what notion of security can be achieved. In terms of

functionality, starting with the seminal notions of *identity-based encryption* (IBE) and *attribute-based encryption* (ABE), FE has progressively evolved through a series of distinguished works to support more and more expressive function families culminating into the recent state of the art schemes which are now able to realize computation of *arbitrary polynomial-size circuits* [6], [12], [11], [10], [7]. Regarding security, the vast majority of research on FE so far has concentrated on protecting privacy of the encrypted contents [6], [15].

### 1.1 Function Privacy in Functional Encryption

A wide range of practical applications, however, demands not only privacy of the encrypted messages but also privacy of the functions for which functional keys are provided. This is especially desirable whenever the function embedded in the functional key itself contains sensitive informations.

Consider the following motivating scenario: Assume that a health organization subscribes to a cloud service provider to store medical records of its patients. To ensure confidentiality of informations, the organization encrypts those records locally using an FE scheme prior to uploading them to the cloud server. Now, using the inherent feature of FE, later on the organization can request the cloud server to perform some analysis on the encrypted records by providing the server the functional key for the respective function. However, if the FE scheme in use does not guarantee any hiding for the functions, which may include sensitive contents, embedded in the functional keys, then the functional keys might reveal the functions completely to the cloud, thereby leaking sensitive informations.

**Private key vs public key setup**: Countless real-life applications have driven the research on function privacy in the context of FE, using the private key setting first by Shen et al. [16] followed by the works of [2], [8], while in the public key setting by Boneh et al. [4], [5]. Intuitively, function privacy requires that functional keys reveal no unnecessary information on their functionality. However, the extent to which function privacy can be satisfied differs dramatically between the private key and public key regimes. Specifically, in the public key domain, where anyone can encrypt messages, only a limited form of function privacy can be attained. To formulate a meaningful security definition, a framework must assume that the functions come from a distribution having sufficient entropy [4], [5]. On the contrary, in the private key setting, function privacy has been shown to have tremendously greater potential compared to the public key domain, both as a stand-alone feature and as a very useful building block.

**Full-hiding security model for private key FE**: For private key FE schemes, the *strongest* (indistinguishability-based) notion of function privacy, also known as *full-hiding* security, formulated in [2], [8] considers both privacy of functional keys and privacy of encrypted data in a perfectly symmetric manner. More precisely, full-hiding security considers adversaries that interact with

I) a left-or-right functional key generation oracle and
II) a left-or-right encryption oracle,

where both oracles operate using the same bit $c \in \{0, 1\}$. The adversaries submit a pair of functions $(f^{(j,0)}, f^{(j,1)})$ to the functional key generation oracle in order to make the $j$-th functional key query while they submit a pair of messages

$(z^{(\ell,0)}, z^{(\ell,1)})$ to the encryption oracle for making the $\ell$-th ciphertext query. Depending on the bit $c$, the functional key generation oracle returns the functional key $\mathrm{SK}_{f^{(j,c)}}$ whereas the encryption oracle sends back the ciphertext $\mathrm{CT}_{z^{(\ell,c)}}$. The adversaries are allowed to interact with these oracles for any polynomial number of queries and the adversaries' goal is to distinguish the cases $c = 0$ and $c = 1$. The constraint on the adversaries is that for all $(f^{(j,0)}, f^{(j,1)})$ and $(z^{(\ell,0)}, z^{(\ell,1)})$ with which they query the functional key generation and encryption oracles respectively, it should hold that $f^{(j,0)}(z^{(\ell,0)}) = f^{(j,1)}(z^{(\ell,1)})$. This is clearly the minimum necessary restriction as otherwise the adversaries can trivially determine the bit $c$ used by the oracles.

Regarding the construction of function private FE schemes in the private key setting, recently Brakerski and Segev [8] have presented a generic transformation from any private key (possibly non-function-private) FE scheme for general polynomial-size circuits into one that achieves function privacy in the strongest model discussed above. Then by combining [8] with the works of [12], [11], or [10], one can obtain private key function-private FE scheme supporting general circuits with strong security guarantee. However, the most significant drawback of the resulting constructions is that they would employ computationally intensive tools for secure computation such as fully homomorphic encryption or program obfuscation and their security would rely on strong assumptions such as indistinguishability obfuscation, extractability obfuscation, or polynomial hardness of simple assumptions on multilinear maps. Consequently, these solutions are far from being practical.

## 1.2 Inner Product Encryption and Function Privacy

A current motivation of cryptographic research community is to design direct and efficient FE schemes for functionalities of practical interest which are still expressive enough for real-life applications. As a first attempt, researchers have focused on the *inner product* functionality which is an extremely useful functionality in the context of descriptive statistics, for example, to compute the weighted mean of a collection of informations. Further, the inner product enables computation of conjunctions, disjunctions, polynomial evaluations, and exact thresholds.

An inner product function family $\mathcal{IP}_p$ is parameterized by a prime integer $p$. A function $\mathrm{IP}_{\vec{y}} \in \mathcal{IP}_p$ is associated with a vector $\vec{y} \in \mathbb{Z}_p^n$ of length $n$ over the finite field $\mathbb{Z}_p$. On a message $\vec{x} \in \mathbb{Z}_p^n$, $\mathrm{IP}_{\vec{y}}(\vec{x})$ is defined to be the inner product $\langle \vec{x}, \vec{y} \rangle$ modulo $p$ of the vectors $\vec{x}$ and $\vec{y}$. We stress that this formulation of inner-product FE, also referred to as *inner product encryption* (IPE) is distinct from [16], [13], [14], [2] which study inner product in the context of predicate encryption (PE). In inner product PE, a message $M$ is encrypted along with a tag $\vec{x} \in \mathbb{Z}_p^n$ and decryption with a key corresponding to a vector $\vec{y} \in \mathbb{Z}_p^n$ yields $M$ if and only if $\langle \vec{x}, \vec{y} \rangle = 0$. In contrast, the objective in the IPE formulation is to learn the actual inner product value in $\mathbb{Z}_p$ itself.

The first construction of IPE was presented by Abdalla et al. [1] who developed a selectively secure construction in traditional discrete log groups. However, this construction is built in public key domain and do not support any form of function privacy. Very recently, Bishop et al. [3] have taken a first step forward

towards exploring the possibility of attaining function privacy in the context of IPE utilizing efficient and well-studied primitives. In fact, they have constructed a function-private IPE scheme in private key domain that withstands any polynomial number of ciphertext and functional key queries. Their construction makes use of asymmetric bilinear pairing groups and derives its security from the well-studied Symmetric External Diffie-Hellman (SXDH) assumption albeit in a rather weak and unrealistic security model.

### 1.3 Our Contribution

The current state of the art leaves open the problem of constructing a private key IPE scheme achieving the strongest practical notion of full-hiding security under standard assumptions without employing any heavy-duty cryptographic tool. In this paper we provide a positive answer to this challenging problem. In particular, we develop a *simple* and *efficient* private key IPE scheme achieving the *strongest* notion of *function privacy* based on well-studied complexity assumption. As in [3], our construction utilizes asymmetric bilinear pairing groups of prime order and we are able to establish the stronger form of security under the SXDH assumption. In order to ensure correctness of our construction, like [1], [3], we assume that the target inner products will be contained within a range of polynomial-size. As pointed out in [1], [3], this assumption is quite reasonable for statistical applications, where, for instance, the average of some bounded quantity over a polynomial-size database will naturally be included in a polynomial range.

Although our construction has some resemblance to that of [3], we highlight several differences below:

- We innovate new technical ideas in order to realize the strongest notion of full-hiding security while maintaining the simplicity of the scheme. For all $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ and $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ with which the adversaries query the functional key generation and encryption oracles respectively, the security framework of [3] assumes that
$$\langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,0)} \rangle = \langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,1)} \rangle = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,0)} \rangle = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,1)} \rangle \quad (1)$$
whereas according to the full-hiding security framework of [2], [8], the only constraint should be
$$\langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,0)} \rangle = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,1)} \rangle. \quad (2)$$
The additional restriction in the security model of [3] has not only weakened the security of their construction significantly but also it has rendered the security model itself rather unrealistic. Our security framework is free from any such restriction beyond that specified in Eq. (2), therefore, much more practical compared to that of [3].

- As in [3], we make use of the concept of *dual pairing vector spaces* (DPVS) introduced in [13], [14] to obtain the features of hidden subspaces in prime order bilinear group setting. However, our two DPVS have dimensions $4n+2$ and 6 respectively while those of [3] have dimensions $2n$ and 2 respectively. Here $n$ is the dimension of vectors for functional keys and ciphertexts. This results in some loss in efficiency. However, this seems rather unavoidable for

strengthening the security both from theoretical and practical point of view.

– Analogous to [3], we consider two pairs of *dual orthonormal bases*, one for each of the two dimensions considered. But instead of including the complete bases like [3], we put certain portions of them in the master secret key while preserve the remaining dimensions for the security reduction. Specifically, we employ $3n$ and 3 hidden dimensions of the pairs of bases of dimensions $4n+2$ and 6 respectively to move things forward in our hybrid security argument.

– At a technical level, [3] used each component of the vectors *twice* while encoding the vectors in ciphertexts and functional keys by coupling them with the basis vectors included in the master secret key. On the contrary, in our construction, we utilize the components of these vectors only *once* in the process of encoding with the basis vectors of the master secret key.

– Although similar to [3], we treat ciphertexts and functional keys in a symmetric fashion in our construction, our hybrid security proof does not maintain any such symmetry. Specifically, the approach of [3] first established the privacy of encrypted messages in the multiple ciphertext framework and then leveraged the symmetry between the structures of ciphertexts and functional keys to flip the same reasoning to argue for function privacy. In doing so, they relied on an information theoretic step that required the additional constraint as in Eq. (1) on the queries of the adversaries. In order to remove the extra restriction, we face several challenges. For our security analysis, we design our hybrid argument differently using a different information theoretic property of DPVS proven by [13] in a non-trivial way. We begin our hybrid game transition by changing the form of the queried ciphertexts and instead of finishing it off completely, at some appropriate point, we initiate change in the queried functional keys. Since then the transformations of functional keys and ciphertexts proceed hand in hand.

# 2 Preliminaries

Throughout this paper we will follow notations presented in Figure 1.

## 2.1 The Notion of Private Key Function-Private IPE

We adopt the general notion of function-private functional encryption in the private key setting, introduced in [2], [8], to the particular functionality of computing inner products of $n$-length vectors over $\mathbb{Z}_p$ for some prime integer $p$ and some positive integer $n$. We will consider only non-zero vectors. Note that this is a reasonable consideration for all practical applications of inner products.

■ **Syntax**: A private key function-private IPE (PKFP-IPE) scheme consists of the following probabilistic polynomial-time algorithms:

PKFP-IPE.Setup$(1^\lambda, n)$: The data owner takes as input the security parameter $1^\lambda$ and a positive integer $n$ (polynomial in $\lambda$) specifying the desired length of

| Symbol | Explanation |
|---|---|
| $\aleph \xleftarrow{\$} A$ | $\aleph$ is randomly selected from $A$ according to $A$'s distribution, when $A$ is a random variable, and $\aleph$ is uniformly selected from $A$, when $A$ is a set. |
| $\vec{v}$ | a vector $(v_1, \ldots, v_n) \in \mathbb{Z}_p^n$ of length $n$ for some positive integers $p$ and $n$. |
| $\langle \vec{v}, \vec{w} \rangle$ | the inner product of vectors $\vec{v}$ and $\vec{w} \in \mathbb{Z}_p^n$, i.e., $\sum_{i=1}^{n} v_i w_i \mod p$. |
| $g^{\vec{v}}$ | an $n$-tuple of group elements, $(g^{v_1}, \ldots, g^{v_n}) \in \mathbb{G}^n$ for some cyclic group $\mathbb{G}$ of order $p$, where $\vec{v} \in \mathbb{Z}_p^n$ and $g \in \mathbb{G}$. |
| $g^{a\vec{v}}$ | $(g^{av_1}, \ldots, g^{av_n}) \in \mathbb{G}^n$, where $a \in \mathbb{Z}_p$, $\vec{v} \in \mathbb{Z}_p^n$, and $g \in \mathbb{G}$. |
| $g^{\vec{v}+\vec{w}}$ | $(g^{v_1+w_1}, \ldots, g^{v_n+w_n}) \in \mathbb{G}^n$, where $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$ and $g \in \mathbb{G}$. |
| $\mathsf{GL}(n, \mathbb{Z}_p)$ | the group of all $n \times n$ invertible matrices over $\mathbb{Z}_p$. |

Fig. 1: Notations

vectors for the functional keys and ciphertexts. It generates a master secret key MSK for itself while publishes public parameters PP. (Note that we are not dealing with a public key scheme, so PP are not sufficient to encrypt – those are just parameters that need not be kept secret.)

PKFP-IPE.Encrypt(MSK, PP, $\vec{x}$): On input the master secret key MSK, the public parameters PP, and a vector $\vec{x} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$, where $\vec{0}$ denotes the all zero vector in $\mathbb{Z}_p^n$, the data owner produces a ciphertext $\mathrm{CT}_{\vec{x}}$.

PKFP-IPE.KeyGen(MSK, PP, $\vec{y}$): Taking as input the master secret key MSK, the public parameters PP, and a vector $\vec{y} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$, the data owner provides a functional key $\mathrm{SK}_{\vec{y}}$ to a legitimate decrypter.

PKFP-IPE.Decrypt(PP, $\mathrm{CT}_{\vec{x}}$, $\mathrm{SK}_{\vec{y}}$): A decrypter takes as input the public parameters PP, a ciphertext $\mathrm{CT}_{\vec{x}}$ encrypting some vector $\vec{x}$, and a functional key $\mathrm{SK}_{\vec{y}}$ corresponding to some vector $\vec{y}$. It outputs either a value $m \in \mathbb{Z}_p$ or the distinguished symbol $\perp$.

■ **Correctness**: The correctness of an PKFP-IPE scheme requires the following: For all $\vec{x}, \vec{y} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$,

$$\mathsf{Pr}\big[(\mathrm{MSK}, \mathrm{PP}) \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.Setup}(1^\lambda, n); \mathrm{CT}_{\vec{x}} \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.Encrypt}(\mathrm{MSK}, \mathrm{PP}, \vec{x});$$
$$\mathrm{SK}_{\vec{y}} \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.KeyGen}(\mathrm{MSK}, \mathrm{PP}, \vec{y}) :$$
$$\mathsf{PKFP\text{-}IPE.Decrypt}(\mathrm{PP}, \mathrm{CT}_{\vec{x}}, \mathrm{SK}_{\vec{y}}) = \langle \vec{x}, \vec{y} \rangle\big] > 1 - \epsilon(\lambda)$$

for some negligible function $\epsilon$. As in [1], [3], in our construction as well we would only require that the above holds when $\langle \vec{x}, \vec{y} \rangle$ is from a fixed polynomial range of values inside $\mathbb{Z}_p$.

■ **Security**: The indistinguishability-based full hiding security notion for a PKFP-IPE scheme is defined by the following game between a probabilistic adversary $\mathcal{A}$ and a probabilistic challenger $\mathcal{C}$:

**Setup**: $\mathcal{C}$ generates $(\mathrm{MSK}, \mathrm{PP}) \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.Setup}(1^\lambda, n)$. It gives PP to $\mathcal{A}$. It also selects $c \xleftarrow{\$} \{0, 1\}$.

**Query Phase**: Throughout the game, $\mathcal{A}$ may adaptively make any polynomial number of queries of the following two types:

- *Functional key query*: To make the $j$-th functional key query, $\mathcal{A}$ submits a pair of vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)}) \in \left(\mathbb{Z}_p^n \backslash \{\vec{0}\}\right)^2$ to $\mathcal{C}$. $\mathcal{C}$ creates a functional key $\text{SK}^{(j)} \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.KeyGen}(\text{MSK}, \text{PP}, \vec{y}^{(j,c)})$ and hands $\text{SK}^{(j)}$ to $\mathcal{A}$.
- *Ciphertext query*: To make the $\ell$-th ciphertext query, $\mathcal{A}$ sends a pair of vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)}) \in \left(\mathbb{Z}_p^n \backslash \{\vec{0}\}\right)^2$ to $\mathcal{C}$. $\mathcal{C}$ forms $\text{CT}^{(\ell)} \xleftarrow{\$} \mathsf{PKFP\text{-}IPE.Encrypt}(\text{MSK}, \text{PP}, \vec{x}^{(\ell,c)})$ and returns $\text{CT}^{(\ell)}$ to $\mathcal{A}$.

Suppose that $\mathcal{A}$ makes $q_1$ number of functional key queries and $q_2$ number of ciphertext queries during the game. The restriction on the queries is that for all $j = 1, \ldots, q_1$ and for all $\ell = 1, \ldots, q_2$, $\langle \vec{x}^{(\ell,0)}, \vec{y}^{(j,0)} \rangle = \langle \vec{x}^{(\ell,1)}, \vec{y}^{(j,1)} \rangle$.
**Guess**: $\mathcal{A}$ eventually outputs a bit $c' \in \{0, 1\}$.

Let $\mathsf{View}_{\mathcal{A}}(c)$ denotes the view of $\mathcal{A}$ in the above game when the $c \in \{0, 1\}$ is the random bit selected by $\mathcal{C}$ in the setup phase.

**Definition 1.** *A* PKFP-IPE *is said to achieve (full) indistinguishability-based full hiding security if for any probabilistic polynomial-time adversary $\mathcal{A}$, for any security parameter $\lambda$, the advantage of $\mathcal{A}$ in the above game, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PKFP\text{-}IPE}}(\lambda) = \left|\mathsf{Pr}\left[\mathcal{A}(\mathsf{View}_{\mathcal{A}}(0)) = 1\right] - \mathsf{Pr}\left[\mathcal{A}(\mathsf{View}_{\mathcal{A}}(1)) = 1\right]\right| < \epsilon(\lambda)$ for some negligible function $\epsilon$.*

## 2.2 Asymmetric Bilinear Group and SXDH Assumption

**Definition 2 (Asymmetric Bilinear Pairing Group).** *An asymmetric bilinear pairing group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ is a tuple of a prime integer $p$; cyclic multiplicative groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order $p$ each with polynomial-time computable group operations; generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$; and a polynomial-time computable non-degenerate bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, i.e., $e$ satisfies*

- *(bilinearity) $e(g_1^s, g_2^{\check{s}}) = e(g_1, g_2)^{s\check{s}}$ for all $s, \check{s} \in \mathbb{Z}_p$ and*
- *(non-degeneracy) $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the identity element of the group $\mathbb{G}_T$.*

*Let $\mathcal{G}_{\mathsf{ABPG}}$ be an algorithm that on input the security parameter $1^\lambda$, outputs a description $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of an asymmetric bilinear pairing group.*

**Assumption 1 (Symmetric External Diffie-Hellman: SXDH).** *The* SXDH *problem is to distinguish between the distributions $\varrho_\beta = \big((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^\mu, g_1^\nu, \Re_\beta, \big)$ for $\beta \in \{0, 1\}$ such that $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda)$, $\mu, \nu \xleftarrow{\$} \mathbb{Z}_p$, and $\Re_\beta = g_1^{\mu\nu+r}$ where $r = 0$ or $r \xleftarrow{\$} \mathbb{Z}_p$ according as $\beta = 0$ or 1 respectively.*

*The* SXDH *assumption states that for any probabilistic polynomial-time algorithm $\mathcal{C}$, for any security parameter $\lambda$, $\mathsf{Adv}_{\mathcal{C}}^{\mathsf{SXDH}}(\lambda) = \left|\mathsf{Pr}\left[\mathcal{C}(\varrho_0) = 1\right] - \mathsf{Pr}\left[\mathcal{C}(\varrho_1) = 1\right]\right| < \epsilon(\lambda)$ for some negligible function $\epsilon$. It also states that the same is true for*

*the analogous distributions obtained from switching the roles of* $\mathbb{G}_1$ *and* $\mathbb{G}_2$*, i.e.,* $\breve{\varrho}_\beta = \big((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_2^{\breve{\mu}}, g_2^{\breve{\nu}}, \breve{\Re}_\beta\big)$ *for* $\beta \in \{0,1\}$ *such that* $\breve{\mu}, \breve{\nu} \xleftarrow{\$} \mathbb{Z}_p$*, and* $\breve{\Re}_\beta = g_2^{\breve{\mu}\breve{\nu}+\breve{r}}$ *where* $\breve{r} = 0$ *or* $\breve{r} \xleftarrow{\$} \mathbb{Z}_p$ *according as* $\beta = 0$ *or 1 respectively.*

## 2.3 Dual Pairing Vector Spaces

**Definition 3 (Dual Pairing Vector Spaces (DPVS)).** *A dual pairing vector space* (DPVS) $(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E)$ *by a direct product of asymmetric pairing groups* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ *is a tuple of a prime integer* $p$*; n-dimensional vector space* $\mathbb{V}_h = \mathbb{G}_h^n$ *over* $\mathbb{Z}_p$ *under vector addition* $\oplus$ *and scalar multiplication* $\otimes$ *defined respectively as* $g_h^{\vec{v}} \oplus g_h^{\vec{w}} = g_h^{\vec{v}+\vec{w}}$ *and* $a \otimes g_h^{\vec{v}} = g_h^{a\vec{v}}$*, for* $h = 1, 2$*, where* $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$*, and* $a \in \mathbb{Z}_p$*; canonical bases* $\mathbb{A}_h = \{g_h^{\vec{e}_i}\}_{i=1,\ldots,n}$ *of* $\mathbb{V}_h$*, for* $h = 1, 2$*, where* $\vec{e}_i = (\overbrace{0, \ldots, 0}^{i-1}, 1, \overbrace{0, \ldots, 0}^{n-i}) \in \mathbb{Z}_p^n$*; and a pairing* $E : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{G}_T$*. The pairing* $E$ *is defined by* $E(g_1^{\vec{v}}, g_2^{\vec{w}}) = \prod_{i=1}^{n} e(g_1^{v_i}, g_2^{w_i}) = e(g_1, g_2)^{\langle \vec{v}, \vec{w} \rangle} \in \mathbb{G}_T$*, where* $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$*. Observe that the map* $E$ *is non-degenerate bilinear, i.e.,* $E$ *satisfies*

- *(bilinearity)* $E(s \otimes g_1^{\vec{v}}, \breve{s} \otimes g_2^{\vec{w}}) = E(g_1^{s\vec{v}}, g_2^{\breve{s}\vec{w}}) = E(g_1^{\vec{v}}, g_2^{\vec{w}})^{s\breve{s}}$ *for* $s, \breve{s} \in \mathbb{Z}_p$, $\vec{v}, \vec{w} \in \mathbb{Z}_p^n$ *and*
- *(non-degeneracy) if* $E(g_1^{\vec{v}}, g_2^{\vec{w}}) = 1_{\mathbb{G}_T}$ *for all* $\vec{w} \in \mathbb{Z}_p^n$*, then* $\vec{v} = \vec{0}$*.*

*When clear from the context, we will often omit the symbols* $\oplus$ *and* $\otimes$ *for vector addition and scalar multiplication respectively in* DPVS*'s. The* DPVS *generation algorithm* $\mathcal{G}_{\mathsf{DPVS}}$ *takes input a positive integer* $n$ *together with* $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda)$ *and outputs a description* $(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E)$ *of* DPVS *with* $n$*-dimensional vector spaces* $\mathbb{V}_h$ *for* $h = 1, 2$*.*

In Figure 2 we describe random *dual orthonormal* basis generator $\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^n)$ for some prime integer $p$ and positive integer $n$. This algorithm would be utilized as a subroutine in our PKFP-IPE construction.

$\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^n)$: This algorithm performs the following operations:

1. Choose $\boldsymbol{B} = (b_{i,j})_{i,j=1,\ldots,n} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$.
2. Compute $\boldsymbol{B}^* = (b^*_{i,j})_{i,j=1,\ldots,n} = (\boldsymbol{B}^\intercal)^{-1}$, where $\boldsymbol{B}^\intercal$ denotes transpose of the matrix $\boldsymbol{B}$. Let, $\vec{b}_i$ and $\vec{b}^*_i$ represent the $i$-th rows of $\boldsymbol{B}$ and $\boldsymbol{B}^*$ respectively, for $i = 1, \ldots, n$. Set $\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_n\}$ and $\mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_n\}$. Note that $(\mathbb{B}, \mathbb{B}^*)$ are dual orthonormal in the sense that for $i, i' = 1, \ldots, n$,
$$\langle \vec{b}_i, \vec{b}^*_{i'} \rangle = \begin{cases} 1, \text{ if } i = i' \\ 0, \text{ otherwise} \end{cases}$$
3. Return $(\mathbb{B}, \mathbb{B}^*)$.

Fig. 2: Dual orthonormal basis generator $\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^n)$

# 3 Our PKFP-IPE Scheme

■ **Construction**

PKFP-IPE.Setup$(1^\lambda, n)$: The data owner takes as input the security parameter $1^\lambda$ and a positive integer $n$ specifying the desired length of vectors for the keys and ciphertexts. It proceeds as follows:

1. It first generates an asymmetric bilinear group
$$(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\$} \mathcal{G}_{\mathsf{ABPG}}(1^\lambda).$$
2. Then it forms
$(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E) \xleftarrow{\$} \mathcal{G}_{\mathsf{DPVS}}\big(4n+2, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$ and
$(p, \mathbb{V}'_1, \mathbb{V}'_2, \mathbb{G}_T, \mathbb{A}'_1, \mathbb{A}'_2, E') \xleftarrow{\$} \mathcal{G}_{\mathsf{DPVS}}\big(6, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$.
3. Next, it samples dual orthonormal bases
$\big(\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_{4n+2}\}, \mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_{4n+2}\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ and
$\big(\mathbb{D} = \{\vec{d}_1, \ldots, \vec{d}_6\}, \mathbb{D}^* = \{\vec{d}^*_1, \ldots, \vec{d}^*_6\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$.
It defines $\widehat{\mathbb{B}} = \{\vec{b}_1, \ldots, \vec{b}_n, \vec{b}_{4n+2}\}, \widehat{\mathbb{B}}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_n, \vec{b}^*_{4n+1}\}, \widehat{\mathbb{D}} = \{\vec{d}_1, \vec{d}_6\}$, and $\widehat{\mathbb{D}}^* = \{\vec{d}^*_1, \vec{d}^*_5\}$.
4. It keeps the master secret key $\text{MSK} = (\widehat{\mathbb{B}}, \widehat{\mathbb{B}}^*, \widehat{\mathbb{D}}, \widehat{\mathbb{D}}^*)$ to itself while publishes the public parameters $\text{PP} = \big(p, \{\mathbb{V}_h, \mathbb{V}'_h\}_{h=1,2}, \mathbb{G}_T, \{\mathbb{A}_h, \mathbb{A}'_h\}_{h=1,2}, E, E'\big)$.

PKFP-IPE.Encrypt$(\text{MSK}, \text{PP}, \vec{x})$: Taking as input the master secret key MSK, the public parameters PP, and a vector $\vec{x} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$, the data owner prepares the ciphertext as follows:

1. It selects $\alpha, \xi, \xi_0 \xleftarrow{\$} \mathbb{Z}_p$ and computes
$$\boldsymbol{c}_1 = g_1^{\alpha \sum_{i=1}^n x_i \vec{b}_i + \xi \vec{b}_{4n+2}} = g_1^{\alpha \sum_i x_i \vec{b}_i + \xi \vec{b}_{4n+2}}, \boldsymbol{c}_2 = g_1^{\alpha \vec{d}_1 + \xi_0 \vec{d}_6} \quad (3)$$
utilizing $\widehat{\mathbb{B}}$ and $\widehat{\mathbb{D}}$ respectively from MSK, where a sum over index $i$ ranges from $i = 1$ to $i = n$ unless explicitly specified otherwise. We will follow the same convention in the sequel as well.
2. It outputs the ciphertext $\text{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$.

PKFP-IPE.KeyGen$(\text{MSK}, \text{PP}, \vec{y})$: On input the master secret key MSK, the public parameters PP, and a vector $\vec{y} \in \mathbb{Z}_p^n \backslash \{\vec{0}\}$, the data owner performs the following:

1. It picks $\gamma, \eta, \eta_0 \xleftarrow{\$} \mathbb{Z}_p$ and computes
$$\boldsymbol{k}_1^* = g_2^{\gamma \sum_i y_i \vec{b}^*_i + \eta \vec{b}^*_{4n+1}}, \boldsymbol{k}_2^* = g_2^{\gamma \vec{d}^*_1 + \eta_0 \vec{d}^*_5} \quad (4)$$
utilizing $\widehat{\mathbb{B}}^*$ and $\widehat{\mathbb{D}}^*$ respectively from MSK.
2. It provides the functional key $\text{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)$ to a legitimate decrypter.

PKFP-IPE.Decrypt$(\text{PP}, \text{CT}_{\vec{x}}, \text{SK}_{\vec{y}})$: A decrypter takes as input the public parameters PP, a ciphertext $\text{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$, and a functional key $\text{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)$. It proceeds as follows:

1. It computes $T_1 = E(\boldsymbol{c}_1, \boldsymbol{k}_1^*), T_2 = E'(\boldsymbol{c}_2, \boldsymbol{k}_2^*)$.
2. It then attempts to determine a value $m \in \mathbb{Z}_p$ such that $T_2^m = T_1$ as elements of $\mathbb{G}_T$ by checking a specified polynomial-size range of possible values. If it is successful, then it outputs $m$. Otherwise it outputs $\perp$.

We stress that the polynomial running time of our decryption algorithm is ensured by restricting the output to lie within a fixed polynomial-size range.

■ **Correctness**

The correctness of the above PKFP-IPE construction can be verified as follows: Observe that for any ciphertext $\text{CT}_{\vec{x}} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$ encrypting some vector $\vec{x}$ and any functional key $\text{SK}_{\vec{y}} = (\boldsymbol{k}_1^*, \boldsymbol{k}_2^*)$ corresponding to some vector $\vec{y}$, we have

$$T_1 = E(\boldsymbol{c}_1, \boldsymbol{k}_1^*) = e(g_1, g_2)^{\alpha\gamma\langle\vec{x},\vec{y}\rangle}, T_2 = E'(\boldsymbol{c}_2, \boldsymbol{k}_2^*) = e(g_1, g_2)^{\alpha\gamma}.$$

This follows from the expressions of $\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{k}_1^*, \boldsymbol{k}_2^*$ together with the fact that $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are dual orthonormal bases. Thus if $\langle\vec{x}, \vec{y}\rangle$ is contained in the specified polynomial-size range of possible values that the decryption algorithm checks, it would output $\langle\vec{x}, \vec{y}\rangle$ as desired.

■ **Discussion**

In our PKFP-IPE construction, we begin with the intuition of [3] to use an asymmetric bilinear group setting $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$, visualizing $\mathbb{G}_1$ as the ciphertext space whereas $\mathbb{G}_2$ as the functional key space. The plaintext vectors are encrypted in the exponent of $g_1$ while the functional key vectors are encapsulated in the exponent of $g_2$, so that the bilinearity of the pairing $e$ can be employed to compute the inner product of the plaintext and functional key vectors in the exponent without the explicit knowledge of the vectors.

As discussed earlier in this paper, the only PKFP-IPE scheme available in the literature so far [3] achieves a rather limited and unrealistic form of function privacy. In particular, for the sake of managing the hybrid security proof of their construction, they put further restrictions on the queries of the adversaries, as shown in Eq. (1), beyond those specified in the strongest framework of full-hiding security described in §2.1. This additional constraint not only leads to a weak security but it is also not conformal with the intuitive spirit of function privacy. With the motivation to remove such an undesirable restriction we recourse to an information theoretic step that uses a nice property of DPVS introduced in [13] that enables to hide a pair of ciphertext and functional key vectors perfectly among all vectors having the same inner product.

To generate space for our hybrid proof, we consider two pairs of dual orthonormal bases, namely, $(\mathbb{B}, \mathbb{B}^*)$ of dimension $4n + 2$ and $(\mathbb{D}, \mathbb{D}^*)$ of dimension 6, where $n$ is the length of vectors for ciphertexts and functional keys. The $n+2$ dimensions of the first pair of bases and 3 of the second pair are used in the actual scheme while the remaining dimensions are preserved to move things forward in the security proof. As displayed in Eq. (3), to encode a vector $\vec{x}$ in the ciphertext, we construct a linear combination of the first $n$ vectors together with the $(4n+2)$-th vector of $\mathbb{B}$, where the $n$ components of $\vec{x}$ masked with a random scalar $\alpha$ are used as coefficients of the first $n$ vectors of $\mathbb{B}$. The resulting vector is then placed in the exponent of $g_1 \in \mathbb{G}_1$. After that, the randomness $\alpha$ is encoded by forming another linear combination of the first and sixth members of $\mathbb{D}$ in the exponent of $g_1$ using the masking factor $\alpha$ as coefficient of the first vector of $\mathbb{D}$. The $(4n + 2)$-th dimension of $\mathbb{B}$ and the sixth dimension of $\mathbb{D}$ are utilized to supply additional randomization for strengthening the security of our ciphertexts. The encoding of a vector for the functional key is performed in a directly symmetric fashion utilizing bases $\mathbb{B}^*, \mathbb{D}^*$, and $g_2 \in \mathbb{G}_2$ in place of $\mathbb{B}, \mathbb{D}$, and $g_1$

respectively, as can be seen from Eq. (4), where the additional randomization is provided by the $(4n + 1)$-th dimension of $\mathbb{B}^*$ and the fifth dimension of $\mathbb{D}^*$.

In contrast, the construction of [3] considers two pairs of dual orthonormal bases, one of dimension $2n$ and the other of dimension 2. Moreover, they make use of the complete bases in their construction itself and employ each component of a vector as coefficient twice during formation of the linear combinations in the process of encoding the vector for ciphertext or functional key, once for basis vectors in the range 1 to $n$ and again for the basis vectors ranging from $n + 1$ to $2n$. Further, [3] rely on the orthogonality of all the queried functional key vectors (respectively all queried ciphertext vectors) to the difference of a pair of queried ciphertext vectors (respectively a pair of queried functional key vectors) to simulate a hidden dimension in the bases in the security proof that they employ to switch from one vector of the pair to the other. However, it is precisely this approach which necessitates the additional constraint imposed by them on the adversaries' queries as in Eq. (1). Furthermore, increasing the dimensions of the DPVS's in use seems rather unavoidable for managing the security reduction without requiring the extra restriction. In fact the $3n$ and 3 hidden dimensions of our two pairs of bases respectively that we keep aside for the security argument play a vital role to elegantly isolate a pair of ciphertext and functional key vectors in an $n$-dimensional hidden subspace in order to apply our information theoretic argument.

In summery, although our construction has some kind of resemblance to that of [3], our proof idea is widely apart. The most significant contribution of our work lies in a rigorous proof of full-hiding security of a fairly simple construction. The detail security reduction is presented in the next section.

In terms of communication cum storage complexity, observe that both the ciphertexts and functional keys of our PKFP-IPE construction consist of $4n + 8$ group elements while our master secret key contains $8n^2 + 12n + 28$ members of the finite field $\mathbb{Z}_p$. In contrast, the ciphertexts and functional keys in the construction of [3] are comprised of $2n + 2$ group elements each whereas the master secret key is composed of $8n^2 + 8$ $\mathbb{Z}_p$ components.

Regarding computation complexity, note that both our encryption and functional key generation algorithms require $4n + 8$ exponentiations while the decryption algorithm involves $4n + 8$ pairing operations followed by an exhaustive search over a polynomial range of values in order to solve a discrete log. On the contrary, the encryption and functional key generation algorithms of [3] amount to $2n + 2$ exponentiations each. Other than a similar exhaustive search step, their decryption algorithm incurs $2n + 2$ pairings.

It is evident that our scheme loses a constant factor of 2 compared to that of [3] in both communication cum storage and computation efficiency. However, the additional cost is compensated with stronger and realistic data as well as function privacy guarantees provided by our construction as opposed to a rather limited form of security achieved by [3]. Given the rapid advancements in computing technology and the growing security breaches, high security is often desirable even at the expense of an admissible increase in complexity.

The ciphertexts and master public key of the only known IPE scheme in public key setup [1] involve $n+1$ and $n$ elements respectively in a discrete log group of prime order $p$ while the master secret key and functional keys are comprised of $n$ and 1 $\mathbb{Z}_p$ components respectively. The encryption and decryption algorithms of [1] respectively incur $2n+1$ exponentiations and $n+1$ exponentiations followed by an analogous exhaustive search step towards determining a discrete log. However, the scheme of [1] offers no function privacy and, moreover, provides only selective data privacy.

# 4 Security Analysis

**Theorem 1.** *The* PKFP-IPE *scheme described in §3 is secure as per the security model of §2.1 under the* SXDH *assumption.*

*Proof.* The proof of Theorem 1 is structured as a hybrid argument over a series of games which differ in the construction of the functional keys and ciphertexts queried by the adversary $\mathcal{A}$ in the security game described in §2.1. In the first game, the queried functional keys and ciphertexts are constructed as those in the security game of §2.1 where the bit used by the challenger is $c = 0$. We then progressively change the functional keys and ciphertexts in multiple hybrid games to those in the security game of §2.1 where the bit used by the challenger is $c = 1$. We prove that each game is indistinguishable from the previous one, thus proving our PKFP-IPE construction to be secure in the security model of §2.1. Let $q_1$ be the number of $\mathcal{A}$'s functional key queries and $q_2$ the number of $\mathcal{A}$'s ciphertext queries. The hybrid game transition is described below. In these games, a portion of an exponent framed by a white box indicates those terms which were added or modified in a transition from the previous game, unless explicitly specified otherwise, while a part of an exponent which was deleted in the transformation from the earlier game is highlighted in the text.

■ **Sequence of Hybrid Games**

**I⟩ Game 0** : This game corresponds to the real security game of §2.1 where the bit used by the challenger to generate queried functional keys and ciphertexts is $c = 0$. More precisely, for $j = 1, \ldots, q_1$, the response to the $j$-th functional key query for vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ is created as $\mathrm{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ such that

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(j)} &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \eta_j \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{d}^*_1 + \eta_{j,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{5}$$

where $\gamma_j, \eta_j, \eta_{j,0} \xleftarrow{\$} \mathbb{Z}_p$. On the other hand, for $\ell = 1, \ldots, q_2$, the reply to the $\ell$-th ciphertext query of $\mathcal{A}$ for vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ is generated as $\mathrm{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ such that

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \xi_\ell \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}\right\} \tag{6}$$

where $\alpha_\ell, \xi_\ell, \xi_{\ell,0} \xleftarrow{\$} \mathbb{Z}_p$.

**II⟩ Game 1 Sequence [Game 1-$\kappa$-1, ..., Game 1-$\kappa$-4 ($\kappa = 1, \ldots, q_2$)]**

**Game 1-$\kappa$-1**: Game 1-0-4 coincides with Game 0. Game 1-$\kappa$-1 is the same as Game 1-$(\kappa-1)$-4 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \boxed{\alpha''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{2n+i}} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \boxed{\alpha''_\kappa \vec{d}_3} + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{7}$$

where $\alpha_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 1-$(\kappa-1)$-4.

**Game 1-$\kappa$-2**: This game is identical to Game 1-$\kappa$-1 with the only exception that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha''_\kappa \sum_i \boxed{x_i^{(\kappa,1)}} \vec{b}_{2n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha''_\kappa \vec{d}_3 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{8}$$

where all the variables are generated as in Game 1-$\kappa$-1.

**Game 1-$\kappa$-3**: This game is analogous to Game 1-$\kappa$-2 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{2n+i} + \boxed{\alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i}} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha''_\kappa \vec{d}_3 + \boxed{\alpha'''_\kappa \vec{d}_4} + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{9}$$

where $\alpha'''_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 1-$\kappa$-2.

**Game 1-$\kappa$-4**: This game is the same as Game 1-$\kappa$-3 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{10}$$

where all the variables are generated as in Game 1-$\kappa$-3, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are modified from those in the last game by dropping the terms involving $\alpha''_\kappa$ in the exponent of $g_1$.

**III⟩ Game 2 Sequence [Game 2-$\omega$-1, ..., Game 2-$\omega$-6 ($\omega = 1, \ldots, q_1$)]**

**Game 2-$\omega$-1**: Game 2-0-6 coincides with Game 1-$q_2$-4. Game 2-$\omega$-1 is the similar to Game 2-$(\omega-1)$-6 except that the components of the $\omega$-th queried functional

key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \boxed{\gamma'_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{n+i} + \gamma''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i}} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \boxed{\gamma'_\omega \vec{d}^*_2 + \gamma''_\omega \vec{d}^*_3} + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \quad (11)$$

where $\gamma'_\omega, \gamma''_\omega \xleftarrow{\$} \mathbb{Z}_p$, and all the other variables are generated as in Game 2-$(\omega-1)$-6.

**Sequence of Subgames of Game 2-$\omega$-2 [Game 2-$\omega$-2-$\kappa$-1, ..., Game 2-$\omega$-2-$\kappa$-5 ($\kappa = 1, \ldots, q_2$)]**

**Game 2-$\omega$-2-$\kappa$-1**: Game 2-$\omega$-2-0-5 coincides with Game 2-$\omega$-1. Game 2-$\omega$-2-$\kappa$-1 is analogous to Game 2-$\omega$-2-$(\kappa-1)$-5 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma'_\omega \sum_i \boxed{y_i^{(\omega,0)}} \vec{b}^*_{n+i} + \gamma''_\omega \sum_i \boxed{y_i^{(\omega,1)}} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma'_\omega \vec{d}^*_2 + \gamma''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \quad (12)$$

where all the variables are generated as in Game 2-$\omega$-2-$(\kappa-1)$-5. Here a part of the exponent framed by a white box (respectively light gray box) indicates those terms which were changed in the transition from the previous game when $\kappa \geq 2$ (respectively $\kappa = 1$). More specifically, when $\kappa = 1$, $\boldsymbol{k}_1^{*(\omega)}$ in Eq. (12) is transformed from that in Eq. (11), which is the form of $\boldsymbol{k}_1^{*(\omega)}$ in Game 2-$\omega$-2-0-5, by changing the portion of the exponent framed by a light gray box. On the other hand, when $\kappa \geq 2$, $\boldsymbol{k}_1^{*(\omega)}$ in Eq. (12) is obtained from that in Eq. (14), which is the form of $\boldsymbol{k}_1^{*(\omega)}$ in Game 2-$\omega$-2-$(\kappa-1)$-5, by applying modification in the portion of the exponent framed by a white box.

**Game 2-$\omega$-2-$\kappa$-2**: This game is identical to Game 2-$\omega$-2-$\kappa$-1 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \boxed{\alpha'_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{n+i}} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \boxed{\alpha'_\kappa \vec{d}_2} + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \quad (13)$$

where $\alpha'_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 2-$\omega$-2-$\kappa$-1.

**Game 2-$\omega$-2-$\kappa$-3**: This game is similar to Game 2-$\omega$-2-$\kappa$-2 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma'_\omega \sum_i \boxed{y_i^{(\omega,1)}} \vec{b}^*_{n+i} + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma'_\omega \vec{d}^*_2 + \gamma''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \quad (14)$$

while the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha'_\kappa \sum_i \boxed{x_i^{(\kappa,1)}} \vec{b}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'_\kappa \vec{d}_2 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{15}$$

where all the variables are generated as in Game 2-$\omega$-2-$\kappa$-2.

**Game 2-$\omega$-2-$\kappa$-4**: This game is the same as Game 2-$\omega$-2-$\kappa$-3 except that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\sum_i (\alpha_\kappa x_i^{(\kappa,0)} \vec{b}_i + \alpha'_\kappa x_i^{(\kappa,1)} \vec{b}_{n+i} + \boxed{\breve{\alpha}''_\kappa x_i^{(\kappa,1)} \vec{b}_{2n+i}} + \alpha'''_\kappa x_i^{(\kappa,1)} \vec{b}_{3n+i}) + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'_\kappa \vec{d}_2 + \boxed{\breve{\alpha}''_\kappa \vec{d}_3} + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{16}$$

where $\breve{\alpha}''_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 2-$\omega$-2-$\kappa$-3.

**Game 2-$\omega$-2-$\kappa$-5**: This game is analogous to Game 2-$\omega$-2-$\kappa$-4 with the only exception that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \breve{\alpha}''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{2n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \breve{\alpha}''_\kappa \vec{d}_3 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{17}$$

where all the variables are generated as in Game 2-$\omega$-2-$\kappa$-4, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are transformed from those in the earlier game by removing the terms involving $\alpha'_\kappa$ in the exponent of $g_1$.

**Game 2-$\omega$-3**: This game is identical to Game 2-$\omega$-2-$q_2$-5 with the only exception that the components of the $\omega$-th queried functional key for vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{18}$$

where all the variables are generated as in Game 2-$\omega$-2-$q_2$-5, i.e., in this game $\boldsymbol{k}_1^{*(\omega)}$ and $\boldsymbol{k}_2^{*(\omega)}$ are changed from those in the last game by deleting the terms involving $\gamma'_\omega$ in the exponent of $g_2$.

**Game 2-$\omega$-4**: This game is the same as Game 2-$\omega$-3 except that the components of the $\omega$-th queried functional key for vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{2n+i} + \boxed{\gamma'''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{3n+i}} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma''_\omega \vec{d}^*_3 + \boxed{\gamma'''_\omega \vec{d}^*_4} + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{19}$$

where $\gamma'''_\omega \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 2-$\omega$-3.

**Game 2-$\omega$-5**: This game is similar to Game 2-$\omega$-4 with the only exception that for $\ell = 1, \ldots, q_2$, the components of the $\ell$-th queried ciphertext for vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \alpha'''_\ell \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \alpha'''_\ell \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}\right\} \tag{20}$$

where all the variables are generated as in Game 2-$\omega$-4, i.e., Eq. (20) resets $\boldsymbol{c}_1^{(\ell)}$ and $\boldsymbol{c}_2^{(\ell)}$, for $\ell = 1, \ldots, q_2$, as those in Eq. (10) by dropping the terms involving $\breve{\alpha}''_\ell$ in the exponent of $g_1$.

**Game 2-$\omega$-6**: This game is the same as Game 2-$\omega$-5 except that the components of the $\omega$-th queried functional key for vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma'''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{3n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma'''_\omega \vec{d}^*_4 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{21}$$

where all the variables are generated as in Game 2-$\omega$-5, i.e., in this game $\boldsymbol{k}_1^{*(\omega)}$ and $\boldsymbol{k}_2^{*(\omega)}$ are changed from those in the earlier game by deleting the terms involving $\gamma''_\omega$ in the exponent of $g_2$.

**IV⟩ Game 3**: This game is analogous to Game 2-$q_1$-6 except that for $j = 1, \ldots, q_1$, the components of the $j$-th queried functional key corresponding to vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(j)} &= g_2^{\gamma_j \sum_i \boxed{y_i^{(j,1)}} \vec{b}^*_i + \gamma'''_j \sum_i \boxed{y_i^{(j,0)}} \vec{b}^*_{3n+i} + \eta_j \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{d}^*_1 + \gamma'''_j \vec{d}^*_4 + \eta_{j,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{22}$$

while for $\ell = 1, \ldots, q_2$, the components of the $\ell$-th queried ciphertext for vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i \boxed{x_i^{(\ell,1)}} \vec{b}_i + \alpha'''_\ell \sum_i \boxed{x_i^{(\ell,0)}} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \alpha'''_\ell \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}\right\} \tag{23}$$

where all the variables are generated as in Game 2-$q_1$-6.

**V⟩ Game 4 Sequence [Game 4-$\omega$-1, ..., Game 4-$\omega$-6 ($\omega = 1, \ldots, q_1$)]**

**Game 4-$\omega$-1**: Game 4-0-6 coincides with Game 3. Game 4-$\omega$-1 is the same as Game 4-$(\omega-1)$-6 except that the components of the $\omega$-th queried functional key for vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \boxed{\breve{\gamma}''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i}} + \gamma'''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{3n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \boxed{\breve{\gamma}''_\omega \vec{d}^*_3} + \gamma'''_\omega \vec{d}^*_4 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{24}$$

where $\breve{\gamma}''_\omega \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 4-$(\omega-1)$-6.

**Game 4-$\omega$-2**: This game is identical to Game 4-$\omega$-1 with the only exception that for $\ell = 1, \ldots, q_2$, the components of the $\ell$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,1)} \vec{b}_i + \boxed{\breve{\alpha}''_\ell \sum_i x_i^{(\ell,0)} \vec{b}_{2n+i}} + \alpha'''_\ell \sum_i x_i^{(\ell,0)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \boxed{\breve{\alpha}''_\ell \vec{d}_3} + \alpha'''_\ell \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}\right\} \tag{25}$$

where $\breve{\alpha}''_\ell \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 4-$\omega$-1.

**Game 4-$\omega$-3**: This game is the same as Game 4-$\omega$-2 with the only exception that the components of the $\omega$-th queried functional key for vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \breve{\gamma}''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \breve{\gamma}''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{26}$$

where all the variables are generated as in Game 4-$\omega$-2, i.e., in this game $\boldsymbol{k}_1^{*(\omega)}$ and $\boldsymbol{k}_2^{*(\omega)}$ are transformed from those in the previous game by dropping the terms involving $\gamma'''_\omega$ in the exponent of $g_2$.

**Game 4-$\omega$-4**: This game is analogous to Game 4-$\omega$-3 except that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \boxed{\breve{\gamma}'_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{n+i}} + \breve{\gamma}''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \boxed{\breve{\gamma}'_\omega \vec{d}^*_2} + \breve{\gamma}''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{27}$$

where $\breve{\gamma}'_\omega \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 4-$\omega$-3.

**Sequence of Subgames of Game 4-$\omega$-5 [Game 4-$\omega$-5-$\kappa$-1, ..., Game 4-$\omega$-5-$\kappa$-5 ($\kappa = 1, \ldots, q_2$)]**

**Game 4-$\omega$-5-$\kappa$-1**: Game 4-$\omega$-5-0-5 coincides with Game 4-$\omega$-4. Game 4-$\omega$-5-$\kappa$-1 is identical to Game 4-$\omega$-5-$(\kappa-1)$-5 except that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\sum_i (\alpha_\kappa x_i^{(\kappa,1)} \vec{b}_i + \boxed{\breve{\alpha}'_\kappa x_i^{(\kappa,0)} \vec{b}_{n+i}} + \breve{\alpha}''_\kappa x_i^{(\kappa,0)} \vec{b}_{2n+i} + \alpha'''_\kappa x_i^{(\kappa,0)} \vec{b}_{3n+i}) + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \boxed{\breve{\alpha}'_\kappa \vec{d}_2} + \breve{\alpha}''_\kappa \vec{d}_3 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{28}$$

where $\breve{\alpha}'_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 4-$\omega$-5-$(\kappa-1)$-5.

**Game 4-$\omega$-5-$\kappa$-2**: This game is the same as Game 4-$\omega$-5-$\kappa$-1 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \breve{\alpha}'_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \breve{\alpha}'_\kappa \vec{d}_2 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{29}$$

where all the variables are generated as in Game 4-$\omega$-5-$\kappa$-1, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are changed from those in the last game by deleting the terms involving $\breve{\alpha}''_\kappa$ in the exponent of $g_1$.

**Game 4-$\omega$-5-$\kappa$-3**: This game is similar to Game 4-$\omega$-5-$\kappa$-2 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \breve{\gamma}'_\omega \sum_i \boxed{y_i^{(\omega,1)}} \vec{b}^*_{n+i} + \breve{\gamma}''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \breve{\gamma}'_\omega \vec{d}^*_2 + \breve{\gamma}''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{30}$$

while the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \breve{\alpha}'_\kappa \sum_i \boxed{x_i^{(\kappa,1)}} \vec{b}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \breve{\alpha}'_\kappa \vec{d}_2 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{31}$$

where all the variables are generated as in Game 4-$\omega$-5-$\kappa$-2.

**Game 4-$\omega$-5-$\kappa$-4**: This game is the same as Game 4-$\omega$-5-$\kappa$-3 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \alpha'''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{32}$$

where all the variables are generated as in Game 4-$\omega$-5-$\kappa$-3, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are transformed from those in the earlier game by removing the terms involving $\breve{\alpha}'_\kappa$ in the exponent of $g_1$.

**Game 4-$\omega$-5-$\kappa$-5**: This game is analogous to Game 4-$\omega$-5-$\kappa$-4 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\boldsymbol{k}_1^{*(\omega)} = \begin{cases} g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \breve{\gamma}'_\omega \sum_i \boxed{y_i^{(\omega,0)}} \vec{b}^*_{n+i} + \breve{\gamma}''_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, & \\ \qquad \text{if } \kappa \le q_2 - 1 & \text{(33a)} \\ g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \breve{\gamma}'_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{n+i} + \breve{\gamma}''_\omega \sum_i \boxed{y_i^{(\omega,1)}} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, & \\ \qquad \text{if } \kappa = q_2 & \text{(33b)} \end{cases}$$

$$\boldsymbol{k}_2^{*(\omega)} = g_2^{\gamma_\omega \vec{d}^*_1 + \breve{\gamma}'_\omega \vec{d}^*_2 + \breve{\gamma}''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5} \tag{33c}$$

where all the variables are generated as in Game 4-$\omega$-5-$\kappa$-4. Here a part of the exponent framed by a white box (respectively light gray box) indicates those terms which were changed from the previous game when $\kappa \leq q_2 - 1$ (respectively $\kappa = q_2$). More precisely, for $\kappa \leq q_2 - 1$, Eq. (33a) resets $\boldsymbol{k}_1^{*(\omega)}$ as in Eq. (27) by changing the portion of the exponent framed by a white box before executing the sequence of subgames Game 4-$\omega$-5-$\kappa$-1 – Game 4-$\omega$-5-$\kappa$-5 for the next value of $\kappa$. Eq. (33b) modifies $\boldsymbol{k}_1^{*(\omega)}$ only once for $\kappa = q_2$ by applying change in the portion of the exponent framed by a light gray box and comes out of the sequence of subgames of Game 4-$\omega$-5.

**Game 4-$\omega$-6**: This game is the same as Game 4-$\omega$-5-$q_2$-5 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_i + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{34}$$

where all the variables are generated as in Game 4-$\omega$-5-$q_2$-5, i.e., in this game $\boldsymbol{k}_1^{*(\omega)}$ and $\boldsymbol{k}_2^{*(\omega)}$ are changed from those in the previous game by deleting the terms involving $\breve{\gamma}'_\omega$ and $\breve{\gamma}''_\omega$ in the exponent of $g_2$.

## VI⟩ Game 5 Sequence [Game 5-$\kappa$-1, ..., Game 5-$\kappa$-4 ($\kappa = 1, \ldots, q_2$)]

**Game 5-$\kappa$-1**: Game 5-0-4 coincides with Game 4-$q_1$-6. Game 5-$\kappa$-1 is similar to Game 5-$(\kappa-1)$-4 except that the components of the $\kappa$-th queried ciphertext for vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \boxed{\grave{\alpha}''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{2n+i}} + \alpha'''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \boxed{\grave{\alpha}''_\kappa \vec{d}_3} + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{35}$$

where $\grave{\alpha}''_\kappa \xleftarrow{\$} \mathbb{Z}_p$ and all the other variables are generated as in Game 5-$(\kappa-1)$-4.

**Game 5-$\kappa$-2**: This game is analogous to Game 5-$\kappa$-1 with the only exception that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \grave{\alpha}''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{2n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \grave{\alpha}''_\kappa \vec{d}_3 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{36}$$

where all the variables are generated as in Game 5-$\kappa$-1, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are modified from those in the last game by dropping the terms involving $\alpha'''_\kappa$ in the exponent of $g_1$.

**Game 5-$\kappa$-3**: This game is identical to Game 5-$\kappa$-2 except that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are

computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \grave{\alpha}''_\kappa \sum_i \boxed{x_i^{(\kappa,1)}} \vec{b}_{2n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \grave{\alpha}''_\kappa \vec{d}_3 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{37}$$

where all the variables are generated as in Game 5-$\kappa$-2.

**Game 5-$\kappa$-4**: This game is similar to Game 5-$\kappa$-3 with the only exception that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are computed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_i + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{38}$$

where all the variables are generated as in Game 5-$\kappa$-3, i.e., in this game $\boldsymbol{c}_1^{(\kappa)}$ and $\boldsymbol{c}_2^{(\kappa)}$ are changed from those in the earlier game by deleting the terms involving $\grave{\alpha}''_\kappa$ in the exponent of $g_1$. Note that in the final game, i.e., Game 5-$q_2$-4, all the queried functional keys $\text{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$, for $j = 1, \ldots, q_1$, and all the queried ciphertexts $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$, for $\ell = 1, \ldots, q_2$, corresponds to functional keys and ciphertexts in the real security game of §2.1 where the bit used by the challenger is $c = 1$.

### ■ Advantages of Adversary in Hybrid Games

Denote $\mathsf{View}_{\mathcal{A}}^{(0)}$; $\mathsf{View}_{\mathcal{A}}^{(1\text{-}\kappa\text{-}h)}$, for $h = 1, \ldots, 4$; $\mathsf{View}_{\mathcal{A}}^{(2\text{-}\omega\text{-}h)}$, for $h = 1, 3, \ldots, 6$; $\mathsf{View}_{\mathcal{A}}^{(2\text{-}\omega\text{-}2\text{-}\kappa\text{-}h)}$, for $h = 1, \ldots, 5$; $\mathsf{View}_{\mathcal{A}}^{(3)}$; $\mathsf{View}_{\mathcal{A}}^{(4\text{-}\omega\text{-}h)}$, for $h = 1, \ldots, 4, 6$; $\mathsf{View}_{\mathcal{A}}^{(4\text{-}\omega\text{-}5\text{-}\kappa\text{-}h)}$, for $h = 1, \ldots, 5$; and $\mathsf{View}_{\mathcal{A}}^{(5\text{-}\kappa\text{-}h)}$, for $h = 1, \ldots, 4$ to be the views of the adversary $\mathcal{A}$ in Game 0; Game 1-$\kappa$-$h$, for $h = 1, \ldots, 4$; Game 2-$\omega$-$h$, for $h = 1, 3, \ldots, 6$; Game 2-$\omega$-2-$\kappa$-$h$, for $h = 1, \ldots, 5$; Game 3; Game 4-$\omega$-$h$, for $h = 1, \ldots, 4, 6$; Game 4-$\omega$-5-$\kappa$-$h$, for $h = 1, \ldots, 5$; and Game 5-$\kappa$-$h$, for $h = 1, \ldots, 4$ respectively. We define the advantage of $\mathcal{A}$ in Game $\iota$ as

$$\mathsf{Adv}_{\mathcal{A}}^{(\iota)}(\lambda) = \mathsf{Pr}\big[\mathcal{A}(\mathsf{View}_{\mathcal{A}}^{(\iota)}) = 1\big],$$

for $\iota \in \{0, 1\text{-}\kappa\text{-}h \ (h = 1, \ldots, 4), 2\text{-}\omega\text{-}h \ (h = 1, 3, \ldots, 6), 2\text{-}\omega\text{-}2\text{-}\kappa\text{-}h \ (h = 1, \ldots, 5), 3, 4\text{-}\omega\text{-}h \ (h = 1, \ldots, 4, 6), 4\text{-}\omega\text{-}5\text{-}\kappa\text{-}h \ (h = 1, \ldots, 5), 5\text{-}\kappa\text{-}h \ (h = 1, \ldots, 4)\}$.

Lemmas 1 – 16, some of which are provided in the next subsection and the rest in the full version [9], will show that the gap in the advantage of the adversary $\mathcal{A}$ between the neighboring games from Game 0 to Game 3 is at most negligible. Further, observe that the transition from Game 3 to Game 5-$q_2$-4 is actually the reverse of the transformation from Game 0 to Game 2-$q_1$-6 with the roles of $(\vec{x}_\ell^{(0)}, \vec{y}_j^{(0)})$ exchanged with that of $(\vec{x}_\ell^{(1)}, \vec{y}_j^{(1)})$, for $j = 1, \ldots, q_1$; $\ell = 1, \ldots, q_2$. Thus, it follows that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PKFP\text{-}IPE}}(\lambda) = \big|\mathsf{Adv}_{\mathcal{A}}^{(0)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(5\text{-}q_2\text{-}4)}(\lambda)\big|$$

is negligible under the SXDH assumption.

Hence the theorem. □

■ **Technically Distinguished Lemmas for Proof of Theorem 1**

Here, we present the statements along with the proofs of those lemmas among Lemmas 1 – 16, namely, Lemmas 1, 2, 5, 8, and 16, which are technically apart from one another. The proofs of these lemmas will demonstrate in detail our main techniques. The remaining lemmas, proofs of which follow methods analogous to some of the lemmas presented in this section, are given in the full version [9].

**Lemma 1.** *For any probabilistic adversary $\mathcal{A}$, there exists a probabilistic algorithm $\mathcal{C}_{1\text{-}1}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter $\lambda$, $\left|\mathsf{Adv}_{\mathcal{A}}^{(1\text{-}(\kappa-1)\text{-}4)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(1\text{-}\kappa\text{-}1)}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{C}_{1\text{-}\kappa\text{-}1}}^{\mathsf{SXDH}}(\lambda)$, where $\mathcal{C}_{1\text{-}\kappa\text{-}1}(\cdot) = \mathcal{C}_{1\text{-}1}(\kappa, \cdot)$.*

*Proof.* Suppose that there is a probabilistic adversary $\mathcal{A}$ that achieves a non-negligible difference in advantage between $\mathsf{Game}$ 1-$(\kappa-1)$-4 and $\mathsf{Game}$ 1-$\kappa$-1. We construct a probabilistic algorithm $\mathcal{C}_{1\text{-}1}$ that attempts to decide the $\mathsf{SXDH}$ problem using $\mathcal{A}$ as a subroutine. $\mathcal{C}_{1\text{-}1}$ is given a positive integer $\kappa$ and an instance of the $\mathsf{SXDH}$ problem $\varrho_\beta = \big((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_1^{\mu}, g_1^{\nu}, \Re_\beta = g_1^{\mu\nu+r}\big)$, where $\mu, \nu \xleftarrow{\$} \mathbb{Z}_p$, and $r = 0$ or $r \xleftarrow{\$} \mathbb{Z}_p$ according as $\beta = 0$ or 1. $\mathcal{C}_{1\text{-}1}$ plays the role of the challenger in the security game of §2.1 and interacts with $\mathcal{A}$ as follows:

• $\mathcal{C}_{1\text{-}1}$ forms $(p, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, E) \xleftarrow{\$} \mathcal{G}_{\mathsf{DPVS}}\big(4n+2, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$ and $(p, \mathbb{V}'_1, \mathbb{V}'_2, \mathbb{G}_T, \mathbb{A}'_1, \mathbb{A}'_2, E') \xleftarrow{\$} \mathcal{G}_{\mathsf{DPVS}}\big(6, (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)\big)$. Next, it samples dual orthonormal bases $\big(\mathbb{F} = \{\vec{f}_1, \ldots, \vec{f}_{4n+2}\}, \mathbb{F}^* = \{\vec{f}^*_1, \ldots, \vec{f}^*_{4n+2}\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ and $\big(\mathbb{H} = \{\vec{h}_1, \ldots, \vec{h}_6\}, \mathbb{H}^* = \{\vec{h}^*_1, \ldots, \vec{h}^*_6\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$. It *implicitly* defines

$$\begin{array}{ll}
\vec{b}_i = \vec{f}_i + \mu\vec{f}_{2n+i}\,(i = 1, \ldots, n), & \vec{b}_i = \vec{f}_i\,(i = n+1, \ldots, 4n+2), \\
\vec{b}^*_{2n+i} = \vec{f}^*_{2n+i} - \mu\vec{f}^*_i\,(i = 1, \ldots, n), & \vec{b}^*_i = \vec{f}^*_i\,(i = 1, \ldots, 2n, 3n+1, \ldots, 4n+2), \\
\vec{d}_1 = \vec{h}_1 + \mu\vec{h}_3, & \vec{d}_i = \vec{h}_i\,(i = 2, \ldots, 6), \\
\vec{d}^*_3 = \vec{h}^*_3 - \mu\vec{h}^*_1, & \vec{d}^*_i = \vec{h}^*_i\,(i = 1, 2, 4, \ldots, 6).
\end{array}$$

It *implicitly* sets $\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_{4n+2}\}, \mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_{4n+2}\}, \mathbb{D} = \{\vec{d}_1, \ldots, \vec{d}_6\}$, and $\mathbb{D}^* = \{\vec{d}^*_1, \ldots, \vec{d}^*_6\}$. Note that $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are dual orthonormal bases since those are obtained by applying an invertible linear transformation to the output of $\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ and $\mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$ respectively. For instance, observe that for $i = 1, \ldots, n$,

$\langle \vec{b}_i, \vec{b}^*_{2n+i} \rangle = \langle \vec{f}_i, \overset{0}{\vec{f}}{}^*_{2n+i} \rangle - \mu\,\langle \vec{f}_i, \overset{1}{\vec{f}}{}^*_i \rangle + \mu\,\langle \overset{1}{\vec{f}}{}_{2n+i}, \overset{1}{\vec{f}}{}^*_{2n+i} \rangle - \mu^2\,\langle \overset{0}{\vec{f}}{}_{2n+i}, \vec{f}^*_i \rangle = 0,$

$\langle \vec{b}_i, \vec{b}^*_i \rangle = \langle \vec{f}_i, \overset{1}{\vec{f}}{}^*_i \rangle + \mu\,\langle \overset{0}{\vec{f}}{}_{2n+i}, \vec{f}^*_i \rangle = 1,$ etc.

It hands the public parameters $\mathrm{PP} = \big(p, \{\mathbb{V}_h, \mathbb{V}'_h\}_{h=1,2}, \mathbb{G}_T, \{\mathbb{A}_h, \mathbb{A}'_h\}_{h=1,2}, E, E'\big)$ to $\mathcal{A}$.

• In response to the $j$-th functional key query of $\mathcal{A}$ corresponding to vectors

$(\overrightarrow{y}^{(j,0)}, \overrightarrow{y}^{(j,1)})$, for $j = 1, \ldots, q_1$, $\mathcal{C}_{1\text{-}1}$ chooses $\gamma_j, \eta_j, \eta_{j,0} \xleftarrow{\$} \mathbb{Z}_p$, computes

$$\boldsymbol{k}_1^{*(j)} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \overrightarrow{f}^*_i + \eta_j \overrightarrow{f}^*_{4n+1}} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \overrightarrow{b}^*_i + \eta_j \overrightarrow{b}^*_{4n+1}},$$
$$\boldsymbol{k}_2^{*(j)} = g_2^{\gamma_j \overrightarrow{h}^*_1 + \eta_{j,0} \overrightarrow{h}^*_5} = g_2^{\gamma_j \overrightarrow{d}^*_1 + \eta_{j,0} \overrightarrow{d}^*_5},$$

and gives the functional key $\text{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ to $\mathcal{A}$.

• In reply to $\mathcal{A}$'s $\ell$-th ciphertext query corresponding to vectors $(\overrightarrow{x}^{(\ell,0)}, \overrightarrow{x}^{(\ell,1)})$, $\mathcal{C}_{1\text{-}1}$ proceeds as follows:

a) $(\ell < \kappa)$ $\mathcal{C}_{1\text{-}1}$ picks $\alpha_\ell, \alpha_\ell''', \xi_\ell, \xi_{\ell,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\boldsymbol{c}_1^{(\ell)} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{f}_i + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \overrightarrow{f}_{3n+i} + \xi_\ell \overrightarrow{f}_{4n+2}} (g_1^\mu)^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{f}_{2n+i}}$$
$$= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{b}_i + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \overrightarrow{b}_{3n+i} + \xi_\ell \overrightarrow{b}_{4n+2}},$$
$$\boldsymbol{c}_2^{(\ell)} = g_1^{\alpha_\ell \overrightarrow{h}_1 + \alpha_\ell''' \overrightarrow{h}_4 + \xi_{\ell,0} \overrightarrow{h}_6} (g_1^\mu)^{\alpha_\ell \overrightarrow{h}_3} = g_1^{\alpha_\ell \overrightarrow{d}_1 + \alpha_\ell''' \overrightarrow{d}_4 + \xi_{\ell,0} \overrightarrow{d}_6}.$$

b) $(\ell = \kappa)$ $\mathcal{C}_{1\text{-}1}$ selects $\xi_\kappa, \xi_{\kappa,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\boldsymbol{c}_1^{(\kappa)} = (g_1^\nu)^{\sum_i x_i^{(\kappa,0)} \overrightarrow{f}_i} (\Re_\beta)^{\sum_i x_i^{(\kappa,0)} \overrightarrow{f}_{2n+i}} g_1^{\xi_\kappa \overrightarrow{f}_{4n+2}}$$
$$= g_1^{\nu \sum_i x_i^{(\kappa,0)} (\overrightarrow{f}_i + \mu \overrightarrow{f}_{2n+i}) + r \sum_i x_i^{(\kappa,0)} \overrightarrow{f}_{2n+i} + \xi_\kappa \overrightarrow{f}_{4n+2}}$$
$$= g_1^{\nu \sum_i x_i^{(\kappa,0)} \overrightarrow{b}_i + r \sum_i x_i^{(\kappa,0)} \overrightarrow{b}_{2n+i} + \xi_\kappa \overrightarrow{b}_{4n+2}},$$
$$\boldsymbol{c}_2^{(\kappa)} = (g_1^\nu)^{\overrightarrow{h}_1} (\Re_\beta)^{\overrightarrow{h}_3} g_1^{\xi_{\kappa,0} \overrightarrow{h}_6} = g_1^{\nu(\overrightarrow{h}_1 + \mu \overrightarrow{h}_3) + r \overrightarrow{h}_3 + \xi_{\kappa,0} \overrightarrow{h}_6} = g_1^{\nu \overrightarrow{d}_1 + r \overrightarrow{d}_3 + \xi_{\kappa,0} \overrightarrow{d}_6}.$$

c) $(\ell > \kappa)$ $\mathcal{C}_{1\text{-}1}$ chooses $\alpha_\ell, \xi_\ell, \xi_{\ell,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\boldsymbol{c}_1^{(\ell)} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{f}_i + \xi_\ell \overrightarrow{f}_{4n+2}} (g_1^\mu)^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{f}_{2n+i}} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \overrightarrow{b}_i + \xi_\ell \overrightarrow{b}_{4n+2}},$$
$$\boldsymbol{c}_2^{(\ell)} = g_1^{\alpha_\ell \overrightarrow{h}_1 + \xi_{\ell,0} \overrightarrow{h}_6} (g_1^\mu)^{\alpha_\ell \overrightarrow{h}_3} = g_1^{\alpha_\ell \overrightarrow{d}_1 + \xi_{\ell,0} \overrightarrow{d}_6}.$$

$\mathcal{C}_{1\text{-}1}$ provides the ciphertext $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ to $\mathcal{A}$.

• Finally, $\mathcal{A}$ outputs a bit $c'$. $\mathcal{C}_{1\text{-}1}$ outputs $\beta' = c'$.

Observe that if $\beta = 0$, i.e., $r = 0$, the $\kappa$-th answered ciphertext is of the form (6), as in Game 1-$(\kappa - 1)$-4, where $\alpha_\kappa = \nu$. On the other hand, if $\beta = 1$, i.e., $r \xleftarrow{\$} \mathbb{Z}_p$, the $\kappa$-th answered ciphertext is of the form (7), as in Game 1-$\kappa$-1, where $\alpha_\kappa = \nu$ and $\alpha''_\kappa = r$. Further, for $\ell < \kappa$, the $\ell$-th answered ciphertext is of the form (10) corresponding to Game 1-$\ell$-4, which is its proper form in both Game 1-$(\kappa-1)$-4 and Game 1-$\kappa$-1 since the full sequence of transformations Game 1-$\ell$-1 – Game 1-$\ell$-4 has already been executed, whereas for $\ell > \kappa$, the $\ell$-th answered ciphertext is of the form (6) corresponding to Game 0, which is its proper form since the sequence of transitions Game 1-$\ell$-1 – Game 1-$\ell$-4 has not yet been taken place. Additionally, for $j = 1, \ldots, q_1$, the $j$-th answered functional key is of the form (5) corresponding to Game 0, which is its proper form since in the game transition so far no change is made in the form of the queried functional keys. Thus the view of $\mathcal{A}$ simulated by $\mathcal{C}_{1\text{-}1}$ is distributed as in Game 1-$(\kappa - 1)$-4 or Game 1-$\kappa$-1 according as $\beta = 0$ or 1. This completes the proof of Lemma 1. □

**Lemma 2.** *For any probabilistic adversary $\mathcal{A}$, for any security parameter $\lambda$, $\mathsf{Adv}_{\mathcal{A}}^{(1\text{-}\kappa\text{-}1)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(1\text{-}\kappa\text{-}2)}(\lambda)$.*

*Proof.* In order to prove Lemma 2, we define an intermediate game, namely, Game 1-$\kappa$-1' as follows and show the equivalence of the distributions of the views of the adversary $\mathcal{A}$ in Game 1-$\kappa$-1 and that in Game 1-$\kappa$-1' (Claim 1) as well as those in Game 1-$\kappa$-2 and in Game 1-$\kappa$-1' (Claim 2).

**Game 1-$\kappa$-1' ($\kappa = 1, \ldots, q_2$)**: This game is identical to Game 1-$\kappa$-1 with the only exception that the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha''_\kappa \sum_i \theta_i^{(\kappa)} \vec{b}_{2n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha''_\kappa \vec{d}_3 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{39}$$

where $\vec{\theta}^{(\kappa)} \xleftarrow{\$} \mathbb{Z}_p^n \backslash \{\vec{0}\}$ and all the other variables are generated as in Game 1-$\kappa$-1.

**Claim 1** *The distribution of the view of the adversary $\mathcal{A}$ in* Game 1-$\kappa$-1 *and that in* Game 1-$\kappa$-1*' are equivalent.*

*Proof.* Consider the distribution of the view of $\mathcal{A}$ in Game 1-$\kappa$-1. We define new dual orthonormal bases $(\mathbb{U}, \mathbb{U}^*)$ of $\mathbb{Z}_p^{4n+2}$ using $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ below. We generate $\boldsymbol{M} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$ and define

$$\left.\begin{aligned} \begin{pmatrix} \vec{u}_{2n+1} \\ \vdots \\ \vec{u}_{3n} \end{pmatrix} &= \boldsymbol{M}^{-1} \cdot \begin{pmatrix} \vec{b}_{2n+1} \\ \vdots \\ \vec{b}_{3n} \end{pmatrix}, \quad \begin{pmatrix} \vec{u}^*_{2n+1} \\ \vdots \\ \vec{u}^*_{3n} \end{pmatrix} = \boldsymbol{M}^{\mathsf{T}} \cdot \begin{pmatrix} \vec{b}^*_{2n+1} \\ \vdots \\ \vec{b}^*_{3n} \end{pmatrix}, \\ \vec{u}_i &= \vec{b}_i, \qquad \vec{u}^*_i = \vec{b}^*_i, \\ & (i = 1, \ldots, 2n, 3n+1, \ldots, 4n+2). \end{aligned}\right\} \tag{40}$$

We set $\mathbb{U} = \{\vec{u}_1, \ldots, \vec{u}_{4n+2}\}, \mathbb{U}^* = \{\vec{u}^*_1, \ldots, \vec{u}^*_{4n+2}\}$. Note that $(\mathbb{U}, \mathbb{U}^*)$ are indeed dual orthonormal bases since those are obtained from the dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ by applying an invertible linear transformation. The components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are expressed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha''_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{2n+i} + \xi_\kappa \vec{b}_{4n+2}} \\ &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{u}_i + \alpha''_\kappa \sum_i \theta_i^{(\kappa)} \vec{u}_{2n+i} + \xi_\kappa \vec{u}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha''_\kappa \vec{d}_3 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{41}$$

where $\alpha_\kappa, \alpha''_\kappa, \xi_\kappa, \xi_{\kappa,0} \xleftarrow{\$} \mathbb{Z}_p$, and $\vec{\theta}^{(\kappa)} = \vec{x}^{(\kappa,0)} \cdot \boldsymbol{M}$.

Since $\vec{x}^{(\kappa,0)} \neq \vec{0}$ and $\boldsymbol{M}$ is uniformly selected from $\mathsf{GL}(n, \mathbb{Z}_p)$, $\vec{\theta}^{(\kappa)}$ is uniformly distributed in $\mathbb{Z}_p^n \backslash \{\vec{0}\}$ and it is independent from all the other variables. The components of any other $\ell$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ are expressed as

a) $(\ell < \kappa)$

$$\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \alpha'''_\ell \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}} \\ &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{u}_i + \alpha'''_\ell \sum_i x_i^{(\ell,1)} \vec{u}_{3n+i} + \xi_\ell \vec{u}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \alpha'''_\ell \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}$$

b) $(\ell > \kappa)$

$$\boldsymbol{c}_1^{(\ell)} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \xi_\ell \vec{b}_{4n+2}} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{u}_i + \xi_\ell \vec{u}_{4n+2}}, \boldsymbol{c}_2^{(\ell)} = g_1^{\alpha_\ell \vec{d}_1 + \xi_{\ell,0} \vec{d}_6},$$

and for all $j = 1, \ldots, q_1$, the components of the $j$-th queried functional key for vectors $(\vec{\boldsymbol{y}}^{(j,0)}, \vec{\boldsymbol{y}}^{(j,1)})$ are expressed as

$$\boldsymbol{k}_1^{*(j)} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \eta_j \vec{b}^*_{4n+1}} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{u}^*_i + \eta_j \vec{u}^*_{4n+1}}, \boldsymbol{k}_2^{*(j)} = g_2^{\gamma_j \vec{d}^*_1 + \eta_{j,0} \vec{d}^*_5},$$

where all the variables are generated as in Game 1-$\kappa$-1.

Observe that in the light of the adversary $\mathcal{A}$'s view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{U}, \mathbb{U}^*)$ are consistent with respect to PP. Also, this transformation of bases maintains the form (5) of the $j$-th answered functional key $\text{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ corresponding to Game 0, for $j = 1, \ldots, q_1$. Additionally, for $\ell < \kappa$, the $\ell$-th answered ciphertext $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ preserves its form as in Eq. (10) corresponding to Game 1-$\ell$-4 while for $\ell > \kappa$, $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ remains the same as in Eq. (6) corresponding to Game 0 under the basis transformation. Moreover, since the RHS of Eq. (41) and that of Eq. (39) are of the same form, the answered ciphertext $\text{CT}^{(\kappa)} = (\boldsymbol{c}_1^{(\kappa)}, \boldsymbol{c}_2^{(\kappa)})$ is Game 1-$\kappa$-1 can be conceptually changed to that in Game 1-$\kappa$-1'. □

**Claim 2** *The distribution of the view of adversary $\mathcal{A}$ in* Game 1-$\kappa$-2 *and that in* Game 1-$\kappa$-1*' are equivalent.*

*Proof.* Claim 2 is proven in a similar manner to Claim 1, using new dual orthonormal bases $(\mathbb{U}, \mathbb{U}^*)$ as in Eq. (40). □

From Claims 1 and 2, it follows that adversary $\mathcal{A}$'s view in Game 1-$\kappa$-1 can be conceptually changed to that in Game 1-$\kappa$-2. This completes the proof of Lemma 2. □

**Lemma 5.** *For any probabilistic adversary $\mathcal{A}$, there exists a probabilistic algorithm $\mathcal{C}_{2\text{-}1}$, whose running time is essentially the same as that of $\mathcal{A}$, such that for any security parameter $\lambda$, $\left|\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}(\omega-1)\text{-}6)}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}\omega\text{-}1)}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{C}_{2\text{-}\omega\text{-}1}}^{\mathsf{SXDH}}(\lambda)$, where $\mathcal{C}_{2\text{-}\omega\text{-}1}(\cdot) = \mathcal{C}_{2\text{-}1}(\omega, \cdot)$.*

*Proof.* Suppose that there is a probabilistic adversary $\mathcal{A}$ that achieves a non-negligible difference in advantage between Game 2-$(\omega - 1)$-6 and Game 2-$\omega$-1. We construct a probabilistic algorithm $\mathcal{C}_{2\text{-}1}$ that attempts to decide the SXDH problem using $\mathcal{A}$ as a subroutine. $\mathcal{C}_{2\text{-}1}$ is given a positive integer $\omega$ and an instance of the SXDH problem $\breve{\varrho}_\beta = \big((p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), g_2^{\breve{\mu}}, g_2^{\breve{\nu}}, \breve{\Re}_\beta = g_2^{\breve{\mu}\breve{\nu} + \breve{r}}\big)$, where $\breve{\mu}, \breve{\nu} \xleftarrow{\$} \mathbb{Z}_p$, and $\breve{r} = 0$ or $\breve{r} \xleftarrow{\$} \mathbb{Z}_p$ according as $\beta = 0$ or 1. $\mathcal{C}_{2\text{-}1}$ plays the

role of the challenger in the security game of §2.1 and interacts with $\mathcal{A}$ as follows:

• The setup phase is executed by $\mathcal{C}_{2\text{-}1}$ in an analogous fashion as that performed by $\mathcal{C}_{1\text{-}1}$ in the proof of Lemma 1 except that $\mathcal{C}_{2\text{-}1}$ sets the dual orthonormal bases $\big(\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_{4n+2}\}, \mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_{4n+2}\}\big)$ and $\big(\mathbb{D} = \{\vec{d}_1, \ldots, \vec{d}_6\}, \mathbb{D}^* = \{\vec{d}^*_1, \ldots, \vec{d}^*_6\}\big)$ *implicitly* from $\big(\mathbb{F} = \{\vec{f}_1, \ldots, \vec{f}_{4n+2}\}, \mathbb{F}^* = \{\vec{f}^*_1, \ldots, \vec{f}^*_{4n+2}\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ and $\big(\mathbb{H} = \{\vec{h}_1, \ldots, \vec{h}_6\}, \mathbb{H}^* = \{\vec{h}^*_1, \ldots, \vec{h}^*_6\}\big) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$ respectively by selecting $\delta, \sigma \xleftarrow{\$} \mathbb{Z}_p$ and *implicitly* defining the following:

$\vec{b}_{n+i} = \vec{f}_{n+i} - \delta\breve{\mu}\vec{f}_i\,(i = 1, \ldots, n)$, $\vec{b}_{2n+i} = \vec{f}_{2n+i} - \sigma\breve{\mu}\vec{f}_i\,(i = 1, \ldots, n)$,
$\vec{b}_i = \vec{f}_i\,(i = 1, \ldots, n, 3n+1, \ldots, 4n+2)$,
$\vec{b}^*_i = \vec{f}^*_i + \delta\breve{\mu}\vec{f}^*_{n+i} + \sigma\breve{\mu}\vec{f}^*_{2n+i}\,(i = 1, \ldots, n)$, $\vec{b}^*_i = \vec{f}^*_i\,(i = n+1, \ldots, 4n+2)$,
$\vec{d}_2 = \vec{h}_2 - \delta\breve{\mu}\vec{h}_1$, $\vec{d}_3 = \vec{h}_3 - \sigma\breve{\mu}\vec{h}_1$, $\vec{d}_i = \vec{h}_i\,(i = 1, 4, \ldots, 6)$,
$\vec{d}^*_1 = \vec{h}^*_1 + \delta\breve{\mu}\vec{h}^*_2 + \sigma\breve{\mu}\vec{h}^*_3$, $\vec{d}^*_i = \vec{h}^*_i\,(i = 2, \ldots, 6)$.

• In response to the $j$-th functional key query of $\mathcal{A}$ corresponding to vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$, $\mathcal{C}_{2\text{-}1}$ proceeds as follows:

a) $(j < \omega)$ $\mathcal{C}_{2\text{-}1}$ picks $\gamma_j, \gamma'''_j, \eta_j, \eta_{j,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{aligned}\boldsymbol{k}_1^{*(j)} &= g_2^{\sum_i (\gamma_j y_i^{(j,0)} \vec{f}^*_i + \gamma'''_j y_i^{(j,1)} \vec{f}^*_{3n+i}) + \eta_j \vec{f}^*_{4n+1}} \oplus \\ &\qquad\qquad (g_2^{\breve{\mu}})^{\sum_i (\delta\gamma_j y_i^{(j,0)} \vec{f}^*_{n+i} + \sigma\gamma_j y_i^{(j,0)} \vec{f}^*_{2n+i})} \\ &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \gamma'''_j \sum_i y_i^{(j,1)} \vec{b}^*_{3n+i} + \eta_j \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{h}^*_1 + \gamma'''_j \vec{h}^*_4 + \eta_{j,0} \vec{h}^*_5} (g_2^{\breve{\mu}})^{\delta\gamma_j \vec{h}^*_2 + \sigma\gamma_j \vec{h}^*_3} = g_2^{\gamma_j \vec{d}^*_1 + \gamma'''_j \vec{d}^*_4 + \eta_{j,0} \vec{d}^*_5}.\end{aligned}$$

b) $(j = \omega)$ $\mathcal{C}_{2\text{-}1}$ chooses $\eta_\omega, \eta_{\omega,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{aligned}\boldsymbol{k}_1^{*(\omega)} &= (g_2^{\breve{\nu}})^{\sum_i y_i^{(\omega,0)} \vec{f}^*_i} (\breve{\Re}_\beta)^{\sum_i (\delta y_i^{(\omega,0)} \vec{f}^*_{n+i} + \sigma y_i^{(\omega,0)} \vec{f}^*_{2n+i})} g_2^{\eta_\omega \vec{f}^*_{4n+1}} \\ &= g_2^{\sum_i \left(\breve{\nu} y_i^{(\omega,0)} (\vec{f}^*_i + \delta\breve{\mu}\vec{f}^*_{n+i} + \sigma\breve{\mu}\vec{f}^*_{2n+i}) + \delta\breve{r} y_i^{(\omega,0)} \vec{f}^*_{n+i} + \sigma\breve{r} y_i^{(\omega,0)} \vec{f}^*_{2n+i}\right) + \eta_\omega \vec{f}^*_{4n+1}} \\ &= g_2^{\breve{\nu} \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \delta\breve{r} \sum_i y_i^{(\omega,0)} \vec{b}^*_{n+i} + \sigma\breve{r} \sum_i y_i^{(\omega,0)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= (g_2^{\breve{\nu}})^{\vec{h}^*_1} (\breve{\Re}_\beta)^{\delta\vec{h}^*_2 + \sigma\vec{h}^*_3} g_2^{\eta_{\omega,0}\vec{h}^*_5} = g_2^{\breve{\nu}(\vec{h}^*_1 + \delta\breve{\mu}\vec{h}^*_2 + \sigma\breve{\mu}\vec{h}^*_3) + \delta\breve{r}\vec{h}^*_2 + \sigma\breve{r}\vec{h}^*_3 + \eta_{\omega,0}\vec{h}^*_5} \\ &= g_2^{\breve{\nu}\vec{d}^*_1 + \delta\breve{r}\vec{d}^*_2 + \sigma\breve{r}\vec{d}^*_3 + \eta_{\omega,0}\vec{d}^*_5}.\end{aligned}$$

c) $(j > \omega)$ $\mathcal{C}_{2\text{-}1}$ selects $\gamma_j, \eta_j, \eta_{j,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{aligned}\boldsymbol{k}_1^{*(j)} &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{f}^*_i + \eta_j \vec{f}^*_{4n+1}} (g_2^{\breve{\mu}})^{\delta\gamma_j \sum_i y_i^{(j,0)} \vec{f}^*_{n+i} + \sigma\gamma_j \sum_i y_i^{(j,0)} \vec{f}^*_{2n+i}} \\ &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \eta_j \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{h}^*_1 + \eta_{j,0}\vec{h}^*_5} (g_2^{\breve{\mu}})^{\delta\gamma_j \vec{h}^*_2 + \sigma\gamma_j \vec{h}^*_3} = g_2^{\gamma_j \vec{d}^*_1 + \eta_{j,0}\vec{d}^*_5}.\end{aligned}$$

$\mathcal{C}_{2\text{-}1}$ hands the functional key $\mathrm{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ to $\mathcal{A}$.

• In reply to the $\ell$-th ciphertext query of $\mathcal{A}$ for vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$, for

$\ell = 1, \ldots, q_2$, $\mathcal{C}_{2\text{-}1}$ selects $\alpha_\ell, \alpha'''_\ell, \xi_\ell, \xi_{\ell,0} \xleftarrow{\$} \mathbb{Z}_p$ and computes

$$\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{f}_i + \alpha'''_\ell \sum_i x_i^{(\ell,1)} \vec{f}_{3n+i} + \xi_\ell \vec{f}_{4n+2}} \\ &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \alpha'''_\ell \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{h}_1 + \alpha'''_\ell \vec{h}_4 + \xi_{\ell,0} \vec{h}_6} = g_1^{\alpha_\ell \vec{d}_1 + \alpha'''_\ell \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}$$

and provides the ciphertext $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ to $\mathcal{A}$.

• Finally, $\mathcal{A}$ outputs a bit $c'$. $\mathcal{C}_{2\text{-}1}$ outputs $\beta' = c'$.

Observe that if $\beta = 0$, i.e., $\breve{r} = 0$, the $\omega$-th answered functional key is of the form (5), as in Game 2-$(\omega-1)$-6, where $\gamma_\omega = \breve{\nu}$. On the other hand, if $\beta = 1$, i.e., $\breve{r} \xleftarrow{\$} \mathbb{Z}_p$, the $\omega$-th answered functional key is of the form (11), as in Game 2-$\omega$-1, where $\gamma_\omega = \breve{\nu}$, $\gamma'_\omega = \delta\breve{r}$, and $\gamma''_\omega = \sigma\breve{r}$. Further, for $j < \omega$, the $j$-th answered functional key is of the form (21) as in Game 2-$j$-6, which is its proper form in both Game 2-$(\omega-1)$-6 and Game 2-$\omega$-1 since the sequence of transitions Game 2-$j$-1 – Game 2-$j$-6 has already been completed, whereas for $j > \omega$, the $j$-th answered functional key is of the form (5) corresponding to Game 0, which is its proper form since during Game 1 sequence of transformations no change was made to the queried functional keys and the sequence of hybrids Game 2-$j$-1 – Game 2-$j$-6 has not yet been executed. Additionally, for $\ell = 1, \ldots, q_2$, the $\ell$-th answered ciphertext is of the form (10) as in Game 1-$q_2$-4, which is the proper form since for $\omega = 1$, no more alteration in the form of these ciphertexts has occurred after Game 1-$q_2$-4 and for $\omega \geq 2$, these ciphertexts have been reset to this form by Eq. (20) in Game 2-$(\omega-1)$-5. Thus the view of $\mathcal{A}$ simulated by $\mathcal{C}_{2\text{-}1}$ is distributed as in Game 2-$(\omega-1)$-6 or Game 2-$\omega$-1 according as $\beta = 0$ or 1. This completes the proof of Lemma 5. □

**Lemma 8.** *For any probabilistic adversary $\mathcal{A}$, for any security parameter $\lambda$, $\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}\omega\text{-}2\text{-}\kappa\text{-}2)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(2\text{-}\omega\text{-}2\text{-}\kappa\text{-}3)}(\lambda)$.*

*Proof.* The proof of Lemma 8 utilizes the following result:

**Lemma 9 (Lemma 3 in [13]).** *For $\tau \in \mathbb{Z}_p$, let $\mathbb{S}_\tau = \{(\vec{\chi}, \vec{\vartheta}) \mid \langle \vec{\chi}, \vec{\vartheta} \rangle = \tau\} \subset \mathbb{Z}_p^n \times \mathbb{Z}_p^n$, where $p$ is a prime integer and $n$ is some positive integer. For all $(\vec{\chi}, \vec{\vartheta}) \in \mathbb{S}_\tau$, for all $(\vec{\zeta}, \vec{v}) \in \mathbb{S}_\tau$,*

$$\mathsf{Pr}\big[\vec{\chi} \cdot \boldsymbol{F} = \vec{\zeta} \bigwedge \vec{\vartheta} \cdot \boldsymbol{F}^* = \vec{v}\big] = \mathsf{Pr}\big[\vec{\chi} \cdot \boldsymbol{F}^* = \vec{\zeta} \bigwedge \vec{\vartheta} \cdot \boldsymbol{F} = \vec{v}\big] = 1/\sharp\mathbb{S}_\tau,$$

*where $\boldsymbol{F} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$, $\boldsymbol{F}^* = (\boldsymbol{F}^\intercal)^{-1}$, and for any set $A$, $\sharp A$ denotes the cardinality of the set $A$.*

In order to prove Lemma 8, we define an intermediate game, namely, Game 2-$\omega$-2-$\kappa$-2' and show the equivalence of the distribution of the view of the adversary $\mathcal{A}$ in Game 2-$\omega$-2-$\kappa$-2 and that in Game 2-$\omega$-2-$\kappa$-2' (Claim 3) as well as those in Game 2-$\omega$-2-$\kappa$-3 and in Game 2-$\omega$-2-$\kappa$-2' (Claim 4).

**Game 2-$\omega$-2-$\kappa$-2' ($\omega = 1, \ldots, q_1$; $\kappa = 1, \ldots, q_2$)**: This game is similar to

Game 2-$\omega$-2-$\kappa$-2 with the only exception that the components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are formed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma'_\omega \sum_i \vartheta_i^{(\omega)} \vec{b}^*_{n+i} + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma'_\omega \vec{d}^*_2 + \gamma''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{42}$$

while the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are created as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha'_\kappa \sum_i \chi_i^{(\kappa)} \vec{b}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'_\kappa \vec{d}_2 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{43}$$

such that $(\vec{\chi}^{(\kappa)}, \vec{\vartheta}^{(\omega)}) \xleftarrow{\$} \mathbb{S}_{\tau_{\omega,\kappa}} = \{(\vec{\chi}, \vec{\vartheta}) \mid \langle \vec{\chi}, \vec{\vartheta} \rangle = \tau_{\omega,\kappa}\} \subset \mathbb{Z}_p^n \times \mathbb{Z}_p^n$, where $\tau_{\omega,\kappa} = \langle \vec{x}^{(\kappa,0)}, \vec{y}^{(\omega,0)} \rangle \big(= \langle \vec{x}^{(\kappa,1)}, \vec{y}^{(\omega,1)} \rangle$ according to the restriction of the security game$\big)$, and all the other variables are generated as in Game 2-$\omega$-2-$\kappa$-2.

**Claim 3** *The distribution of the view of adversary $\mathcal{A}$ in* Game 2-$\omega$-2-$\kappa$-2 *and that in* Game 2-$\omega$-2-$\kappa$-2*' are equivalent.*

*Proof.* Consider the distribution of the view of $\mathcal{A}$ in Game 2-$\omega$-2-$\kappa$-2. We define new dual orthonormal bases $(\mathbb{U}, \mathbb{U}^*)$ of $\mathbb{Z}_p^{4n+2}$ using $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ below. We generate $\boldsymbol{W} \xleftarrow{\$} \mathsf{GL}(n, \mathbb{Z}_p)$ and set

$$\left.\begin{aligned} \begin{pmatrix} \vec{u}_{n+1} \\ \vdots \\ \vec{u}_{2n} \end{pmatrix} &= \boldsymbol{W}^{-1} \cdot \begin{pmatrix} \vec{b}_{n+1} \\ \vdots \\ \vec{b}_{2n} \end{pmatrix}, \quad \begin{pmatrix} \vec{u}^*_{n+1} \\ \vdots \\ \vec{u}^*_{2n} \end{pmatrix} = \boldsymbol{W}^\intercal \cdot \begin{pmatrix} \vec{b}^*_{n+1} \\ \vdots \\ \vec{b}^*_{2n} \end{pmatrix}, \\ \vec{u}_i &= \vec{b}_i, \qquad \vec{u}^*_i = \vec{b}^*_i, \\ & (i = 1, \ldots, n, 2n+1, \ldots, 4n+2). \end{aligned}\right\} \tag{44}$$

We define $\mathbb{U} = \{\vec{u}_1, \ldots, \vec{u}_{4n+2}\}, \mathbb{U}^* = \{\vec{u}^*_1, \ldots, \vec{u}^*_{4n+2}\}$. Note that $(\mathbb{U}, \mathbb{U}^*)$ are indeed dual orthonormal bases since those are obtained from the dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ by applying an invertible linear transformation. The components of the $\omega$-th queried functional key corresponding to vectors $(\vec{y}^{(\omega,0)}, \vec{y}^{(\omega,1)})$ are expressed as

$$\left.\begin{aligned} \boldsymbol{k}_1^{*(\omega)} &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_i + \gamma'_\omega \sum_i y_i^{(\omega,0)} \vec{b}^*_{n+i} + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{b}^*_{2n+i} + \eta_\omega \vec{b}^*_{4n+1}} \\ &= g_2^{\gamma_\omega \sum_i y_i^{(\omega,0)} \vec{u}^*_i + \gamma'_\omega \sum_i \vartheta_i^{(\omega)} \vec{u}^*_{n+i} + \gamma''_\omega \sum_i y_i^{(\omega,1)} \vec{u}^*_{2n+i} + \eta_\omega \vec{u}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(\omega)} &= g_2^{\gamma_\omega \vec{d}^*_1 + \gamma'_\omega \vec{d}^*_2 + \gamma''_\omega \vec{d}^*_3 + \eta_{\omega,0} \vec{d}^*_5}, \end{aligned}\right\} \tag{45}$$

while the components of the $\kappa$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\kappa,0)}, \vec{x}^{(\kappa,1)})$ are expressed as

$$\left.\begin{aligned} \boldsymbol{c}_1^{(\kappa)} &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_i + \alpha'_\kappa \sum_i x_i^{(\kappa,0)} \vec{b}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{b}_{3n+i} + \xi_\kappa \vec{b}_{4n+2}} \\ &= g_1^{\alpha_\kappa \sum_i x_i^{(\kappa,0)} \vec{u}_i + \alpha'_\kappa \sum_i \chi_i^{(\kappa)} \vec{u}_{n+i} + \alpha'''_\kappa \sum_i x_i^{(\kappa,1)} \vec{u}_{3n+i} + \xi_\kappa \vec{u}_{4n+2}}, \\ \boldsymbol{c}_2^{(\kappa)} &= g_1^{\alpha_\kappa \vec{d}_1 + \alpha'_\kappa \vec{d}_2 + \alpha'''_\kappa \vec{d}_4 + \xi_{\kappa,0} \vec{d}_6}, \end{aligned}\right\} \tag{46}$$

where $\gamma_\omega, \gamma'_\omega, \gamma''_\omega, \eta_\omega, \eta_{\omega,0}, \alpha_\kappa, \alpha'_\kappa, \alpha'''_\kappa, \xi_\kappa, \xi_{\kappa,0} \xleftarrow{\$} \mathbb{Z}_p$, and $\vec{\vartheta}_\omega = \vec{y}^{(\omega,0)} \cdot (\boldsymbol{W}^\intercal)^{-1}$, $\vec{\chi}^{(\kappa)} = \vec{x}^{(\kappa,0)} \cdot \boldsymbol{W}$.

From Lemma 9 it follows that $(\vec{\chi}^{(\kappa)}, \vec{\vartheta}^{(\omega)})$ are uniformly distributed in $\mathbb{S}_{\tau_{\omega,\kappa}}$, where $\langle \vec{x}^{(\kappa,0)}, \vec{y}^{(\omega,0)} \rangle = \tau_{\omega,\kappa}$, and are independent from all the other variables.

The components of any other $j$-th queried functional key corresponding to vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ are expressed as follows

a) $(j < \omega)$

$$\begin{aligned} \boldsymbol{k}_1^{*(j)} &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \gamma_j''' \sum_i y_i^{(j,1)} \vec{b}^*_{3n+i} + \eta_j \vec{b}^*_{4n+1}} \\ &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{u}^*_i + \gamma_j''' \sum_i y_i^{(j,1)} \vec{u}^*_{3n+i} + \eta_j \vec{u}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{d}^*_1 + \gamma_j''' \vec{d}^*_4 + \eta_{j,0} \vec{d}^*_5}, \end{aligned}$$

b) $(j > \omega)$

$$\begin{aligned} \boldsymbol{k}_1^{*(j)} &= g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \eta_j \vec{b}^*_{4n+1}} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{u}^*_i + \eta_j \vec{u}^*_{4n+1}}, \\ \boldsymbol{k}_2^{*(j)} &= g_2^{\gamma_j \vec{d}^*_1 + \eta_{j,0} \vec{d}^*_5}, \end{aligned}$$

while the components of any other $\ell$-th queried ciphertext corresponding to vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ are expressed as

a) $(\ell < \kappa)$

$$\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \breve{\alpha}_\ell'' \sum_i x_i^{(\ell,1)} \vec{b}_{2n+i} + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}} \\ &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{u}_i + \breve{\alpha}_\ell'' \sum_i x_i^{(\ell,1)} \vec{u}_{2n+i} + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \vec{u}_{3n+i} + \xi_\ell \vec{u}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \breve{\alpha}_\ell'' \vec{d}_3 + \alpha_\ell''' \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}$$

b) $(\ell > \kappa)$

$$\begin{aligned} \boldsymbol{c}_1^{(\ell)} &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}} \\ &= g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{u}_i + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \vec{u}_{3n+i} + \xi_\ell \vec{u}_{4n+2}}, \\ \boldsymbol{c}_2^{(\ell)} &= g_1^{\alpha_\ell \vec{d}_1 + \alpha_\ell''' \vec{d}_4 + \xi_{\ell,0} \vec{d}_6}, \end{aligned}$$

where all the variables are generated as in Game 2-$\omega$-2-$\kappa$-2.

Observe that in the light of the adversary $\mathcal{A}$'s view, both $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{U}, \mathbb{U}^*)$ are consistent with respect to PP. Also, for $j < \omega$, the $j$-th answered functional key $\text{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ preserves its form as in Eq. (21) corresponding to Game 2-$j$-6 while for $j > \omega$, $\text{SK}^{(j)} = (\boldsymbol{k}_1^{*(j)}, \boldsymbol{k}_2^{*(j)})$ remains the same as in Eq. (5) corresponding to Game 0, and for $\ell < \kappa$, the $\ell$-th answered ciphertext $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ preserves its form as in Eq. (17) corresponding to Game 2-$\omega$-2-$\ell$-5 while for $\ell > \kappa$, $\text{CT}^{(\ell)} = (\boldsymbol{c}_1^{(\ell)}, \boldsymbol{c}_2^{(\ell)})$ remains the same as in Eq. (10) corresponding to Game 1-$q_2$-4 (or equivalently of the form (20) as in Game 2-$(\omega-1)$-5, for $\omega \geq 2$) under the basis transformation. Moreover, since the RHS of Eq. (45) (respectively Eq. (46)) and that of Eq. (42) (respectively Eq. (43)) are of the same form, the $\omega$-th queried functional key $\text{SK}^{(\omega)} = (\boldsymbol{k}_1^{*(\omega)}, \boldsymbol{k}_2^{*(\omega)})$ and the $\kappa$-th queried ciphertext $\text{CT}^{(\kappa)} = (\boldsymbol{c}_1^{(\kappa)}, \boldsymbol{c}_2^{(\kappa)})$ in Game 2-$\omega$-2-$\kappa$-2 can be conceptually changed to those in Game 2-$\omega$-2-$\kappa$-2'. □

**Claim 4** *The distribution of the view of the adversary $\mathcal{A}$ in* Game 2-$\omega$-2-$\kappa$-3 *and that in* Game 2-$\omega$-2-$\kappa$-2*' are equivalent.*

*Proof.* Claim 4 is proven in an analogous manner to Claim 3 using new dual orthonormal bases $(\mathbb{U}, \mathbb{U}^*)$ as in Eq. (44). □

From Claims 3 and 4 it follows that adversary $\mathcal{A}$'s view in Game 2-$\omega$-2-$\kappa$-2 can be conceptually changed to that in Game 2-$\omega$-2-$\kappa$-3. This completes the proof of Lemma 8. □

**Lemma 16.** *For any probabilistic adversary $\mathcal{A}$, for any security parameter $\lambda$,* $\mathsf{Adv}_{\mathcal{A}}^{(2\text{-}q_1\text{-}6)}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{(3)}(\lambda)$.

*Proof.* In Game 2-$q_1$-6, for $j = 1, \ldots, q_1$, the components of the $j$-th queried functional key corresponding to vectors $(\vec{y}^{(j,0)}, \vec{y}^{(j,1)})$ have the form

$$\boldsymbol{k}_1^{*(j)} = g_2^{\gamma_j \sum_i y_i^{(j,0)} \vec{b}^*_i + \gamma_j''' \sum_i y_i^{(j,1)} \vec{b}^*_{3n+i} + \eta_j \vec{b}^*_{4n+1}}, \boldsymbol{k}_2^{*(j)} = g_2^{\gamma_j \vec{d}^*_1 + \gamma_j''' \vec{d}^*_4 + \eta_{j,0} \vec{d}^*_5},$$

as in Eq. (21), where $\gamma_j, \gamma_j''', \eta_j, \eta_{j,0} \xleftarrow{\$} \mathbb{Z}_p$, while for $\ell = 1, \ldots, q_2$, the components of the $\ell$-th queried ciphertext for vectors $(\vec{x}^{(\ell,0)}, \vec{x}^{(\ell,1)})$ of the form

$$\boldsymbol{c}_1^{(\ell)} = g_1^{\alpha_\ell \sum_i x_i^{(\ell,0)} \vec{b}_i + \alpha_\ell''' \sum_i x_i^{(\ell,1)} \vec{b}_{3n+i} + \xi_\ell \vec{b}_{4n+2}}, \boldsymbol{c}_2^{(\ell)} = g_1^{\alpha_\ell \vec{d}_1 + \alpha_\ell''' \vec{d}_4 + \xi_{\ell,0} \vec{d}_6},$$

as in Eq. (20), where $\alpha_\ell, \alpha_\ell''', \xi_\ell, \xi_{\ell,0} \xleftarrow{\$} \mathbb{Z}_p$.

Therefore, by swapping the components of the dual orthonormal bases $\big(\mathbb{B} = \{\vec{b}_1, \ldots, \vec{b}_{4n+2}\}, \mathbb{B}^* = \{\vec{b}^*_1, \ldots, \vec{b}^*_{4n+2}\}\big)$ $\big($respectively $\big(\mathbb{D} = \{\vec{d}_1, \ldots, \vec{d}_6\}, \mathbb{D}^* = \{\vec{d}^*_1, \ldots, \vec{d}^*_6\}\big)\big)$ in the first block, i.e., in the range $i = 1, \ldots, n$ (respectively $i = 1$) and in the fourth block, i.e., in the range $i = 3n+1, \ldots, 4n$ (respectively $i = 4$), we obtain the distribution in Game 3. More precisely, we define new dual orthonormal bases $(\mathbb{U}, \mathbb{U}^*)$ of $\mathbb{Z}_p^{4n+2}$ and $(\mathbb{W}, \mathbb{W}^*)$ of $\mathbb{Z}_p^6$ using $(\mathbb{B}, \mathbb{B}^*) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^{4n+2})$ and $(\mathbb{D}, \mathbb{D}^*) \xleftarrow{\$} \mathcal{G}_{\mathsf{OB}}(\mathbb{Z}_p^6)$ as follows: We set

$$\begin{array}{llll}
\vec{u}_{3n+i} = \vec{b}_i, & \vec{u}^*_{3n+i} = \vec{b}^*_i \ (i = 1, \ldots, n), & & \\
\vec{u}_i = \vec{b}_{3n+i}, & \vec{u}^*_i = \vec{b}^*_{3n+i} \ (i = 1, \ldots n), & & \\
\vec{u}_i = \vec{b}_i, & \vec{u}^*_i = \vec{b}^*_i \ (i = n+1, \ldots, 3n, 4n+1, 4n+2), & & \\
\vec{w}_4 = \vec{d}_1, & \vec{w}^*_4 = \vec{d}^*_1, & \vec{w}_1 = \vec{d}_4, & \vec{w}^*_1 = \vec{d}^*_4, \\
\vec{w}_i = \vec{d}_i, & \vec{w}^*_i = \vec{d}^*_i \ (i = 2, 3, 5, 6). & &
\end{array}$$

We define $\mathbb{U} = \{\vec{u}_1, \ldots, \vec{u}_{4n+2}\}, \mathbb{U}^* = \{\vec{u}^*_1, \ldots, \vec{u}^*_{4n+2}\}, \mathbb{W} = \{\vec{w}_1, \ldots, \vec{w}_6\}$, $\mathbb{W}^* = \{\vec{w}^*_1, \ldots, \vec{w}^*_6\}$. It is clear that $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{W}, \mathbb{W}^*)$ are indeed dual orthonormal bases since those are obtained from the dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ respectively by means of invertible linear transformations.

Observe that in light of the adversary $\mathcal{A}$'s view, both $(\mathbb{B}, \mathbb{B}^*)$ (respectively $(\mathbb{D}, \mathbb{D}^*)$) and $(\mathbb{U}, \mathbb{U}^*)$ (respectively $(\mathbb{W}, \mathbb{W}^*)$) are consistent with respect to PP. Moreover, it readily follows that the components of the queried functional keys and ciphertexts in Game 2-$q_1$-6 over bases $(\mathbb{B}, \mathbb{B}^*)$ and $(\mathbb{D}, \mathbb{D}^*)$ are expressed as those in Eq. (22) and Eq. (23) of Game 3 over bases $(\mathbb{U}, \mathbb{U}^*)$ and $(\mathbb{W}, \mathbb{W}^*)$. This completes the proof of Lemma 16. □

# 5 Conclusion

In this paper, we have presented the *first non-generic* private key FE scheme for the inner product functionality achieving the *strongest indistinguishability-based* notion of function privacy, namely, the full-hiding security [2], [8]. Our

construction has utilized the standard asymmetric bilinear pairing group of prime order and has derived its security from the SXDH assumption. A significant future direction of research in this area would be to explore *simulation-based* notion of function privacy [2] in the context of IPE in the private key setting.

# References


1. Abdalla, M., Bourse, F., De Caro, A., Pointcheval, D.: Simple functional encryption schemes for inner products. In: Public-Key Cryptography–PKC 2015, pp. 733–751. Springer (2015)
2. Agrawal, S., Agrawal, S., Badrinarayanan, S., Kumarasubramanian, A., Prabhakaran, M., Sahai, A.: Function private functional encryption and property preserving encryption: New definitions and positive results. Tech. rep., Cryptology ePrint Archive, Report 2013/744 (2013)
3. Bishop, A., Jain, A., Kowalczyk, L.: Function-hiding inner product encryption. Tech. rep., Cryptology ePrint Archive, Report 2015/672 (2015)
4. Boneh, D., Raghunathan, A., Segev, G.: Function-private identity-based encryption: Hiding the function in functional encryption. In: Advances in Cryptology–CRYPTO 2013, pp. 461–478. Springer (2013)
5. Boneh, D., Raghunathan, A., Segev, G.: Function-private subspace-membership encryption and its applications. In: Advances in Cryptology-ASIACRYPT 2013, pp. 255–275. Springer (2013)
6. Boneh, D., Sahai, A., Waters, B.: Functional encryption: Definitions and challenges. In: Theory of Cryptography, pp. 253–273. Springer (2011)
7. Boyle, E., Chung, K.M., Pass, R.: On extractability obfuscation. In: Theory of Cryptography, pp. 52–73. Springer (2014)
8. Brakerski, Z., Segev, G.: Function-private functional encryption in the private-key setting. In: Theory of Cryptography, pp. 306–324. Springer (2015)
9. Datta, P., Dutta, R., Mukhopadhyay, S.: Functional encryption for inner product with full function privacy. Tech. rep., Cryptology ePrint Archive, Report 2015/1255 (2015)
10. Garg, S., Gentry, C., Halevi, S., Zhandry, M.: Fully secure functional encryption without obfuscation. Tech. rep., Cryptology ePrint Archive, Report 2014/666 (2014)
11. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate indistinguishability obfuscation and functional encryption for all circuits. In: Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on. pp. 40–49. IEEE (2013)
12. Goldwasser, S., Kalai, Y., Popa, R.A., Vaikuntanathan, V., Zeldovich, N.: Reusable garbled circuits and succinct functional encryption. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing. pp. 555–564. ACM (2013)
13. Okamoto, T., Takashima, K.: Fully secure functional encryption with general relations from the decisional linear assumption. In: Advances in Cryptology–CRYPTO 2010, pp. 191–208. Springer (2010)
14. Okamoto, T., Takashima, K.: Fully secure unbounded inner-product and attribute-based encryption. In: Advances in Cryptology–ASIACRYPT 2012, pp. 349–366. Springer (2012)
15. O'Neill, A.: Definitional issues in functional encryption. Tech. rep., Cryptology ePrint Archive, Report 2010/556 (2010)
16. Shen, E., Shi, E., Waters, B.: Predicate privacy in encryption systems. In: Theory of Cryptography, pp. 457–473. Springer (2009)