Unbounded Predicate Inner Product Functional Encryption from Pairings[⋆]

Uddipana Dowerah[1], Subhranil Dutta[2], Aikaterini Mitrokotsa[3], Sayantan Mukherjee[3], Tapas Pal[4]

[1]Chalmers University of Technology, Sweden uddipana@chalmers.se
[2]Indian Institute of Technology Kharagpur, India subhranildutta@iitkgp.ac.in
[3]University of St Gallen, Switzerland katerina.mitrokotsa@unisg.ch
[3]University of St Gallen, Switzerland csayantan.mukherjee@gmail.com
[4]NTT Social Informatics Laboratories, Japan tapas.pal.wh@hco.ntt.co.jp

Abstract.

Predicate inner product functional encryption (P-IPFE) is essentially attribute-based IPFE (AB-IPFE) which additionally hides attributes associated to ciphertexts. In a P-IPFE, a message $\mathbf{x}$ is encrypted under an attribute $\mathbf{w}$ and a secret key is generated for a pair $(\mathbf{y}, \mathbf{v})$ such that recovery of $\langle \mathbf{x}, \mathbf{y} \rangle$ requires the vectors $\mathbf{w}, \mathbf{v}$ to satisfy a linear relation. We call a P-IPFE *unbounded* if it can encrypt unbounded length attributes and message vectors.

- *zero predicate IPFE.* We construct the *first* unbounded zero predicate IPFE (UZP-IPFE) which recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ if $\langle \mathbf{w}, \mathbf{v} \rangle = 0$. This construction is inspired by the unbounded IPFE of Tomida and Takashima (ASIACRYPT 2018) and the unbounded zero inner product encryption of Okamoto and Takashima (ASIACRYPT 2012). The UZP-IPFE stands secure against general attackers capable of decrypting the challenge ciphertext. Concretely, it provides full attribute-hiding security in the indistinguishability-based semi-adaptive model under the standard symmetric external Diffie-Hellman assumption.
- *non-zero predicate IPFE.* We present the *first* unbounded non-zero predicate IPFE (UNP-IPFE) that successfully recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ if $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$. We generically transform an unbounded quadratic FE (UQFE) scheme to weak attribute-hiding UNP-IPFE in both public and secret key settings. Interestingly, our secret key *simulation* secure UNP-IPFE has *succinct* secret keys and is constructed from a novel *succinct* UQFE that we build in the random oracle model. We leave the problem of constructing a succinct public key UNP-IPFE or UQFE in the standard model as an important open problem.

[⋆] This paper will appear in the Topical Collection on Computing on Encrypted Data of the Journal of Cryptology.

Table of Contents

1 Introduction

Functional encryption (FE) is an advanced cryptographic primitive that enables elegant access control over encrypted data. The motivation behind introducing FE is to deviate from the classical "all-or-nothing" type encryption schemes that entirely unveil the plaintext to the secret key owner. On the contrary, specific functions are embedded into the secret keys of an FE scheme which reveal no information about the plaintext but only its functional values. More formally, an FE scheme that supports a function class $\mathcal{F}$, allows the authority to issue secret keys SK_f corresponding to any function $f \in \mathcal{F}$. Using the public parameters of the scheme, a message m is encrypted to the ciphertext CT_m which reveals $f(m)$ on decrypting it with the secret key SK_f. The security of an FE scheme ensures that no information about the message m can be extracted from the pair $(\mathsf{SK}_f, \mathsf{CT}_m)$ apart from the functional value $f(m)$.

A significant amount of effort [GKP+13, GVW15] has been put forth in realizing FE schemes supporting the class of all polynomial-size functions. Although such powerful FE schemes are being developed through a long sequence of works [JLS21, BV18, APM20] based on standard assumptions, these are relatively complex to understand and far away from practical deployment. On the positive side, FE schemes for specific function classes, e.g., linear or quadratic functions [ABCP15, BJK15, ABDCP16, ALS16, DDM16, AGRW17, BBL17, BCFG17, ACF+18, CLT18, DOT18, ABKW19, LT19, ACGU20, Wee20, Gay20, AGT21b, AGT21a], can be constructed with much more efficient parameters. This work is devoted to realizing practical FE schemes primarily for linear functions in a setting suitable for various applications that deal with variable lengthed data.

FE for Attribute-Based Linear Functions. Inner Product Functional Encryption (IPFE) refers to a practical class of FE, introduced by Abdalla et al. [ABCP15], that supports inner product functionality. The secret keys and ciphertexts are computed for vectors $\mathbf{y}$, $\mathbf{x} \in \mathbb{Z}_p^n$ respectively and the decryption obtains an inner product value $\langle \mathbf{x}, \mathbf{y} \rangle$. Due to its simple and linear functionality, IPFE possesses an inherent security issue. More precisely, releasing a set of n secret keys corresponding to a basis of $\mathbb{Z}_p^n$ entirely breaks the security of the IPFE system. This necessitates the key generation algorithm of stateful IPFE to prevent the risk of releasing n secret keys of particular nature. To make the IPFE system resilient from such information leakage even when many secret keys are issued, the scheme must allow to embed access policies, while also being able to compute the weighted sums on the data. This can be achieved for instance by combining attribute-based encryption (ABE) [GPSW06] with IPFE. Abdalla et al. [ACGU20] addressed this problem by proposing a primitive called attribute-based IPFE (AB-IPFE) that provides the access control functionality of ABE along with the inner product functionality of IPFE. More precisely, the secret keys of IPFE are now additionally associated with some policies P and message vectors are encrypted under some attributes att. At the time of the decryption, computing the inner product value $\langle \mathbf{x}, \mathbf{y} \rangle$ requires the secret key to satisfy an extra condition $P(\mathsf{att}) = 1$, *i.e.,* the attribute att must satisfy the policy P.

Various constructions of AB-IPFE were proposed in previous works [ACGU20, PD21, LLW21] depending on group-based and lattice-based assumptions. These AB-IPFEs focus on hiding the message vectors, not the associated attributes. However, hiding attributes in ABE [OT12b, OT12a, OT15] has its own popularity in applications where attributes contain user-specific sensitive information. Another drawback is that existing AB-IPFEs can only process attributes or messages of bounded length. As a result, the size of system parameters depends on the upper bounds set for the lengths of attributes/messages at the beginning. Furthermore, the ciphertexts always scale with the upper bounds even when the original lengths of the corresponding attributes and messages are much shorter. In the literature, the primitives ABE [LW11, OT12b, BV16, Att19, AMY19, DPD21] and IPFE [TT18, DSP19] are individually constructed to handle unbounded length attributes and messages respectively. However, no AB-IPFE scheme is designed to process data of arbitrary length. Additionally, these salient features, namely *attribute-hiding* and *unboundedness* of an AB-IPFE scheme, make the parameters cost-effective and amplify the importance and wide applicability of AB-IPFE. Specific application scenarios include weighted sum of body temperature or blood pressure of patients in a hospital, average salary of a minority group in a private/government office or even counting votes of political leaders in a presidential election. In all these examples, the size of the data set may vary from time-to-time, for instance, the number of patients in the hospital or employees in the minority group. Concurrently, the associated attributes contain sensitive information such as the patients' social security numbers or the employee codes of the minority group members. This motivates us to ask the following question:

Is it possible to design an AB-IPFE scheme that can embed unbounded size (policy, key vector) to a secret key and unbounded length (attribute, message vector) to a ciphertext, so that only authorized persons can recover the inner product between the key and message vectors, without revealing any information about the attributes apart from whether they are satisfied or not by the embedded policy?

1.1 Our Contributions

This work proposes a solution to the above open problem for inner-product. We define the notion of *attribute-hiding unbounded* AB-IPFE where access to the inner product values is controlled via linear predicates. More fundamentally, we explore the primitive AB-IPFE from the lens of FE. This means the entities associated with a ciphertext CT of AB-IPFE, i.e., the attribute att and the message vector $\mathbf{x}$ are both hidden during the decryption. A secret key associated with a tuple $(P, \mathbf{y})$ reveals at most the information about $P(\mathsf{att})$ and the inner product $\langle \mathbf{x}, \mathbf{y} \rangle$ from CT. We propose the name *predicate inner-product functional encryption* (P-IPFE) to separate this primitive from usual AB-IPFE of [ACGU20, AGT21b, PD21, LLW21] and *predicated inner product functional encryption* of [AGT21a]. Before we note down the difference between *predicated inner product functional encryption* of [AGT21a] and our primitive, we state that the name is inspired by the attribute-hiding feature that differentiates *predicate encryption* [KSW08] from the *attribute-based encryption* [GPSW06]. The definition of *predicated inner product functional encryption* of [AGT21a], although captured inner-product computation conditioned on a linear predicate, did not capture the essence of *predicate encryption* [KSW08] thoroughly. Indeed, the *partially function-hiding* security in [AGT21a] did not consider the attribute-hiding feature. On the other hand, our definition captures both the message vector hiding and the full attribute-hiding [OT12b]. Therefore, we propose a new name for the primitive we consider.

We further enhance the primitive P-IPFE by adding the property of *unboundedness* that makes it more efficient in terms of system keys and ciphertext sizes. This means the master keys of the P-IPFE only depend on the security parameter and hence there is no bound on the sizes of $P, \mathsf{att}, \mathbf{x}$ and $\mathbf{y}$. This work deals with *unbounded inner product predicates* [OT12b, DPD21] where $P = \mathbf{v}$ is a linear function of $\mathsf{att} = \mathbf{w}$ having unbounded lengths. In particular, we construct *unbounded inner product predicate* IPFE (UP-IPFE) schemes which recover $\langle \mathbf{x}, \mathbf{y} \rangle$ if a linear relation $R(\mathbf{w}, \mathbf{v})$ holds. We emphasize that our UP-IPFE is the first primitive to simultaneously capture unbounded inner product predicate encryption scheme [OT12b] and unbounded inner product functional encryption schemes [TT18, DSP19].

UP-IPFE with *zero* relation. First, we consider UP-IPFE with *zero* relation (UZP-IPFE) meaning that the decryption recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ if the inner product between the predicate and attribute vectors is *zero*. We present a construction of semi-adaptively secure UZP-IPFE in the standard model under the *symmetric external Diffie-Hellman* (SXDH) assumption in an asymmetric pairing group, where the unbounded length vectors (both $\mathbf{x}, \mathbf{y}$ and $\mathbf{w}, \mathbf{v}$) satisfy a *permissive* relation. A pair of unbounded length vectors is said to satisfy the *permissive* relation if the index set of one is contained in the index set of the other [TT18, DSP19]. The ciphertexts and secret keys of our UZP-IPFE grow linearly with the lengths of the associated vectors.

We achieve *full* attribute-hiding indistinguishability-based security with semi-adaptive attributes. In our security model, only the challenge attributes are submitted before an adversary asks for a secret key whereas the challenge message vectors are adaptively chosen after observing a set of secret key queries. Note that the notion of full attribute-hiding approves secret key queries capable of decrypting the challenge ciphertext. Hence, it provides more power to the adversary than the usual payload hiding model where it is prohibited to query a secret key that decodes the challenge ciphertext.

Technically, we combine the full attribute-hiding unbounded zero-predicate encryption (UZIPE) of Okamoto and Takashima [OT12b] with the UIPFE of Tomida and Takashima [TT18] to achieve our result. We show that it may not be possible to generically construct UZP-IPFE from UZIPE and UIPFE with our desired security notion. The previous works have also noted this [ACGU20, AGT21b, PD21, LLW21] in the context of AB-IPFE. Our main technical insight is that it is possible to semi-generically combine the existing UZIPE [OT12b] and UIPFE [TT18] by implicitly employing a *joint* secret sharing protocol. This enables us to design a framework for UZP-IPFE which hides the arbitrary length attributes into the ciphertexts. We believe our technique could be useful to combine primitives such as ABE/PE with linear/quadratic FE in a semi-generic manner for achieving more expressive classes of functional encryption.

UP-IPFE with *non-zero* relation. Next, we consider UP-IPFE with *non-zero* relation (UNP-IPFE) meaning that the decryption recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ if the inner product between the predicate and attribute vectors is *non-zero*. We present a generic construction of *weak*

Table 1: Summary of our results

Scheme	$\lvert\mathsf{MPK}\rvert$	$\lvert\mathsf{CT}\rvert$	$\lvert\mathsf{SK}\rvert$	Assumption
UZP-IPFE	$56\,\lvert\mathbb{G}_1\rvert$	$7(m_1+m_2)\,\lvert\mathbb{G}_1\rvert$	$7(n_1+n_2)\,\lvert\mathbb{G}_2\rvert$	SXDH
UNP-IPFE	$10\,\lvert\mathbb{G}_1\rvert$	$(2m_1+4m_2+6)\,\lvert\mathbb{G}_1\rvert+2m_2\,\lvert\mathbb{G}_2\rvert$	$4\,\lvert\mathbb{G}_2\rvert$	bi-2-Lin

– m_1, m_2: the lengths of the vectors associated with the ciphertext.
– n_1, n_2: the lengths of the vectors associated with the secret key.
– $\lvert\mathsf{MPK}\rvert$: the size of the master public key.
– $\lvert\mathsf{CT}\rvert, \lvert\mathsf{SK}\rvert$: the size of the ciphertext and the secret key, respectively.
– SXDH, bi-k-Lin: symmetric external Diffie-Hellman (or 1-Lin), bilateral k-Lin.
The computations for UNP-IPFE are for the specific case of $(k'=1)$-Lin (SXDH) and $k=2$ (bi-2-Lin).

attribute-hiding UNP-IPFE where the unbounded length vectors (both $\mathbf{x}, \mathbf{y}$ and $\mathbf{w}, \mathbf{v}$) satisfy either the permissive relation or a *strict* relation. We say a pair of unbounded length vectors satisfies the *strict* relation if the index sets of the vectors are identical. We instantiate our generic construction in the public key setting with permissive relation and in the secret key setting with strict relation, enjoying variable efficiency parameters and security levels.

- The public key UNP-IPFE achieves the permissive relation for vectors and is indistinguishability-based secure in the standard model. The sizes of secret keys and ciphertexts scale linearly with the associated vectors. We obtain the public key UNP-IPFE by plugging in the existing unbounded linear and quadratic FE schemes [TT18, Tom22] to our generic construction.
- The secret key UNP-IPFE is instantiated with strict relation for vectors and it is simulation-secure in the random oracle model (ROM). The ciphertext size is linear in the length of associated vectors as in the case of our UZP-IPFE and public key UNP-IPFE. Moreover, the secret key achieves succinctness, meaning that the secret keys' size are independent of the length of the predicate and key vectors. To instantiate the secret key UNP-IPFE, we construct a succinct secret key unbounded quadratic FE (UQFE) scheme, which is simulation secure under the bilateral k-Lin assumption in the ROM. In literature, such a succinct UQFE scheme does not exist to the best of our knowledge. The only existing (public key) UQFE scheme by Tomida [Tom22] is semi-adaptively indistinguishability-based secure in the ROM. It generates secret keys that grow linearly with the size of key vectors; hence they are *not* succinct. As illustrated in Table 1, our secret key UNP-IPFE delivers significant efficiency improvements in all departments compared to the other UP-IPFE schemes.

Lastly, both of these UNP-IPFEs are semi-adaptively secure with respect to the attribute and message vectors. The adversary submits challenge attributes and message vectors before it receives any secret key. In Table 2, we provide a comparison of existing (partially/weak/full) attribute-hiding FE schemes with our proposed UP-IPFEs with respect to the functionality and security model.

Application Scenarios: Similar to IPFE, the primitive AB-IPFE finds application on various fronts. We believe AB-IPFE can be useful in Hamming distance-based biometric au-

Table 2: Comparison of our results with existing attribute-based FE schemes

Scheme	Functionality	$(\lvert\mathsf{att}\rvert, \lvert\mathsf{msg}\rvert)$	Attribute-hiding	Security	Assumption
[TT18]	$\phi_{\mathbf{y}\in\mathbb{Z}_p^{\lvert I_{\mathbf{y}}\rvert}} : \mathbb{Z}_p^{\lvert I_{\mathbf{x}}\rvert} \to \mathbb{Z}_p, \phi_{\mathbf{y}}(\mathbf{x}) = \mathbf{x}^\top\mathbf{y}$	(×, unbd)	×	AD-IND	SXDH
[DSP19]	$\phi_{\mathbf{y}\in\mathbb{Z}_p^{\lvert I_{\mathbf{y}}\rvert}} : \mathbb{Z}_p^{\lvert I_{\mathbf{x}}\rvert} \to \mathbb{Z}_p, \phi_{\mathbf{y}}(\mathbf{x}) = \mathbf{x}^\top\mathbf{y}$	(×, unbd)	×	Sel-IND	DBDH
[ACGU20]	$\phi_{(f\in(\mathsf{NC}^1)^{(n)},\mathbf{y}\in\mathbb{Z}_p^{n'})} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_{f,\mathbf{y}}(\mathbf{x},\mathbf{w}) = (f(\mathbf{w}) \overset{?}{=} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(bnd, bnd)	×	AD-IND	SXDH
[AGW20]	$\phi_{f\in\mathsf{ABP}^{(n',n)}} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_f(\mathbf{x},\mathbf{w}) = f(\mathbf{w})^\top\mathbf{x}$	(bnd, bnd)	partially	SA-SIM	k-Lin
[Wee20]	$\phi_{f\in\mathsf{ABP}^{(n_1{}'n_2{}',n)}} : \mathbb{Z}_p^{n_1'+n_2'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_f((\mathbf{x}_1,\mathbf{x}_2),\mathbf{w}) = (\mathbf{x}_1\otimes\mathbf{x}_2)f(\mathbf{w})^\top$	(bnd, bnd)	partially	SA-SIM	bi-k-Lin
[LLW21]	$\phi_{(f\in(\mathsf{GC})^{(n)},\mathbf{y}\in\mathbb{Z}_p^{n'})} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_{f,\mathbf{y}}(\mathbf{x},\mathbf{w}) = (f(\mathbf{w}) \overset{?}{=} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(bnd, bnd)	×	SA-IND	LWE
[AGT21a]	$\phi_{(\mathbf{y}\in\mathbb{Z}_p^{m_1},\mathbf{v}\in\mathbb{Z}_p^{m_2})} : \mathbb{Z}_p^{m_1} \times \mathbb{Z}_p^{m_2} \to \mathbb{Z}_p, \phi_{(\mathbf{y},\mathbf{v})}(\mathbf{x},\mathbf{w}) = (\mathbf{w}^\top\mathbf{v} \overset{?}{=} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(bnd, bnd)	×	Sel-IND	MDDH
[AGT21b]	$\phi_{(f\in(\mathsf{NC}^1)^{n},\mathbf{y}\in\mathbb{Z}_p^{n'})} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_{f,\mathbf{y}}(\mathbf{x},\mathbf{w}) = (f(\mathbf{w}) \overset{?}{=} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(bnd, bnd)	weak	Sel-IND	k-Lin
[DP21]	$\phi_{f\in\mathsf{ABP}^{(n',n)}} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_f(\mathbf{x},\mathbf{w}) = f(\mathbf{w})^\top\mathbf{x}$	(bnd, bnd)	partially	AD-SIM	k-Lin
[Tom22]	$\phi_{f\in\mathsf{ABP}^{(n'\cdot n',n)}} : \mathbb{Z}_p^{n'} \times \mathbb{Z}_p^{n} \to \mathbb{Z}_p, \phi_f(\mathbf{x},\mathbf{w}) = (\mathbf{x}\otimes\mathbf{x})f(\mathbf{w})^\top$	(bnd, unbd)	partially	SA-IND	MDDH
This work	$\phi_{(\mathbf{y}\in\mathbb{Z}_p^{\lvert I_{\mathbf{y}}\rvert},\mathbf{v}\in\mathbb{Z}_p^{\lvert I_{\mathbf{v}}\rvert})} : \mathbb{Z}_p^{\lvert I_{\mathbf{x}}\rvert} \times \mathbb{Z}_p^{\lvert I_{\mathbf{w}}\rvert} \to \mathbb{Z}_p, \phi_{(\mathbf{y},\mathbf{v})}(\mathbf{x},\mathbf{w}) = (\mathbf{w}^\top\mathbf{v} \overset{?}{=} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(unbd, unbd)	full	SA-IND	SXDH
This work	$\phi_{(\mathbf{y}\in\mathbb{Z}_p^{\lvert I_{\mathbf{y}}\rvert},\mathbf{v}\in\mathbb{Z}_p^{\lvert I_{\mathbf{v}}\rvert})} : \mathbb{Z}_p^{\lvert I_{\mathbf{x}}\rvert} \times \mathbb{Z}_p^{\lvert I_{\mathbf{w}}\rvert} \to \mathbb{Z}_p, \phi_{(\mathbf{y},\mathbf{v})}(\mathbf{x},\mathbf{w}) = (\mathbf{w}^\top\mathbf{v} \overset{?}{\neq} 0)\cdot\mathbf{x}^\top\mathbf{y}$	(unbd, unbd)	weak	SA-SIM	bi-k-Lin

- ABP, GC: arithmetic branching programs, general circuits respectively.
- AD, SA, Sel: adaptive, semi-adaptive and selective security respectively.
- IND, SIM: indistinguishability and simulation based security.
- $\lvert\mathsf{att}\rvert, \lvert\mathsf{msg}\rvert$: lengths of attribute and message respectively.
- $\lvert I_{\mathbf{x}}\rvert$: size of the index set of $\mathbf{x}$.
- bnd, unbd: bounded, unbounded respectively.
- DBDH, LWE,MDDH: decisional bilinear Diffie-Hellman, learning with errors, matrix decisional Diffie-Hellman, respectively.

thentication [LKK+18], cloud-assisted computing etc. while providing strong privacy guarantees. Nevertheless, we discuss a concrete and simple application scenario relevant to the modern-day use cases of cloud computing in medical science. In Fig. 1, we illustrate one such application scenario that our UP-IPFE schemes can efficiently realize. The data owners are hospitals that encrypt patients' health records under their attributes and upload the ciphertexts into a cloud server. At the same time, the data users are distinguished scientists from various research centers who study health records. Suppose the Ministry of Healthcare (MoH) department wants to perform statistical analysis over the encrypted data of patients to determine the state of some emerging diseases such as lung infections, influenza, dengue etc. in a certain region of the country so that necessary steps could be taken in advance to prevent escalation of the diseases. The MoH provisions certain research centers for this purpose and provides secret keys generated for specific policies and functions to the centers. The secret keys enable a specific group of scientists in the research centers to compute patient data functions if they are authorized or satisfy the embedded policies. For example, the hospitals are requested to encrypt the patients' dataset $\mathbf{x}$ including body temperature, heart rate, blood pressure etc. under attributes $\mathbf{w}$ including social security number (SSN), race, age, sex, past significant diseases, radiological accession number and possible designations of scientists that are allowed to study such data. The MoH employs an AB-IPFE scheme to compute the average body temperature or blood pressure of the patients recently treated for influenza. The MoH provides a secret key $\mathsf{SK}_{P,\mathbf{y}}$ to the research center, where P is the policy defined by $5000 < \text{SSN} < 8000$, a specified range of social security numbers (of patients) and Infectious Disease Specialist (scientists), and $\mathbf{y}$ is a chosen weight vector.

In this example, data users search patients' health records with respect to some specific attributes and then perform statistical computations on the encrypted data. Since inner product predicates enable evaluation of disjunctions, polynomials, and CNF/DNF formulae [KSW08], we can consider any such predicates with IPFE for computing the average. Therefore, P-IPFE, a particular case of AB-IPFE, serves the purpose of the MoH. However, if the MoH employs an existing AB-IPFE [ACGU20, PD21, LLW21] that supports only bounded size data/attribute sets, it faces two major problems. Firstly, it is almost impossible to guess the size of data/attribute sets (or the number of patients/characteristics) at the time of the system setup. Eventually, the MoH is forced to choose an upper bound on the size of those sets; hence, the size of system parameters (especially the master public key) grows with the upper bound. Moreover, the ciphertexts that are ever generated by the hospitals scale with the upper bound although the associated message/attribute size is much smaller than the bound. Secondly, existing AB-IPFEs [ACGU20, PD21, LLW21] completely disclose the attributes associated with ciphertexts. This leads to the leakage of patients' personal information (age, social security number etc.) to the data users, which could be against the privacy policy of the hospitals. On the other hand, if the MoH employs our attribute-hiding UP-IPFE schemes, then it circumvents these two essential drawbacks. More specifically, the ciphertext hides the associated attributes and grows linearly with the data size and attribute sets available at the time of the encryption.

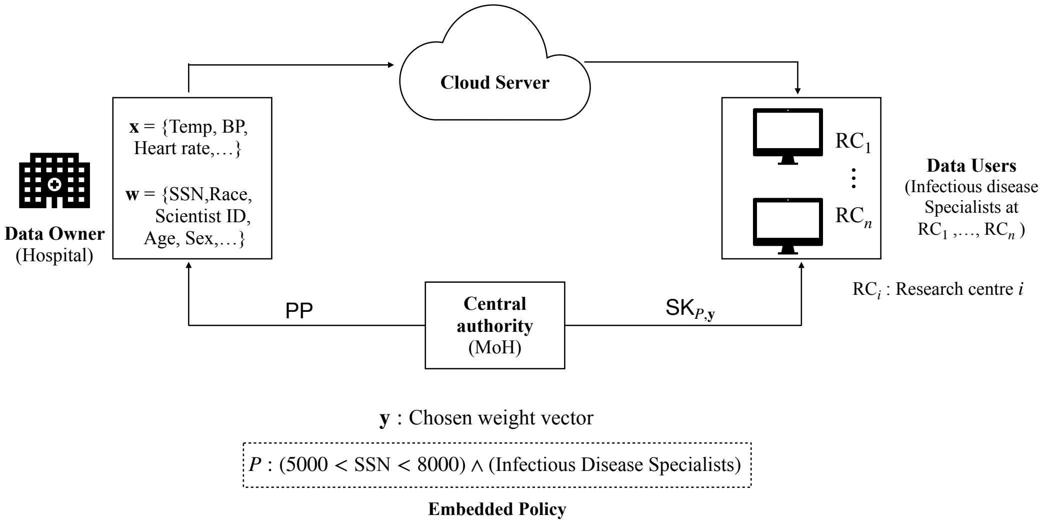
Cloud Server
x = {Temp, BP, Heart rate,…}
w = {SSN,Race, Scientist ID, Age, Sex,…}
Data Owner
(Hospital)
RC$_1$
RC$_n$
Data Users
(Infectious disease Specialists at RC$_1$,…, RC$_n$)
RC$_i$: Research centre i
PP
Central authority
(MoH)
SK$_{P,\mathbf{y}}$
y : Chosen weight vector
P : (5000 < SSN < 8000) ∧ (Infectious Disease Specialists)
Embedded Policy

Fig. 1: An application scenario of UP-IPFE

1.2 Related work

The first unbounded IPFE schemes were concurrently and independently proposed by Tomida and Takashima [TT18] and Dufour-Sans and Pointcheval [DSP19]. In [TT18], Tomida and Takashima presented two constructions for unbounded IPFE (UIPFE): a private key UIPFE with full function-hiding and a public key UIPFE with adaptive indistinguishability security based on the standard SXDH assumption. Concurrently, in [DSP19], Dufour-Sans and Pointcheval presented public key UIPFE constructions with succinct public key, master secret key as well as succinct functional decryption keys. They also considered identity-based access control in their constructions. However, their constructions achieve only selective security in the random oracle model. Abdalla et al. combined the access control properties of ABE with IPFE in [ACGU20] and presented the first constructions of *attribute-based* IPFE (AB-IPFE) using state-of-the-art ABE schemes from prime order pairing groups. Agrawal et al. extended the construction of [ACGU20] to the multi-authority setting in [AGT21b]. However, the constructions of [ACGU20] do not achieve the attribute-hiding property, whereas the multi-authority construction in [AGT21b] only achieves weak attribute-hiding property. Further, in [AGW20], Abdalla et. al proposed an FE scheme for a new functionality called *attribute-weighted sums* with semi-adaptive security and subsequently, Datta and Pal [DP21] presented the first adaptively secure FE schemes for attribute-weighted sums. However, these constructions are only partially attribute-hiding and not to mention that all of these attribute-based FE schemes [ACGU20, AGT21b, AGW20, DP21] are in the bounded setting. Recently, the first unbounded FE scheme for quadratic functions has been proposed by Tomida in [Tom22]. The scheme achieves semi-adaptive indistinguishability-based security under the MDDH assumption in the random oracle model. The same work provides attribute-based access control over the UQFE via arithmetic branching programs (ABP) [AGW20]. Since ABPs are a type of a non-uniform model of computation, the length of attributes in [Tom22] is essentially bounded. Nevertheless, the attributes associated with the ABPs are public, yielding a partially attribute-hiding FE scheme.

Organization of the paper: In Section 2, we briefly overview our techniques. In Section 3, we define some standard notations and recall the definition of bilinear groups, our complexity assumptions, DPVS and the syntax with the security definitions of UP-IPFE scheme. In Section 4, we propose the construction of UZP-IPFE along with a security proof. Section 5 presents the definition of UQFE and a candidate construction along with simulation-based security proof. Section 6 describes our generic construction of UNP-IPFE along with simulation-based security analysis. Further, we give instantiations of our UNP-IPFE scheme in both public key and private key settings. Finally, in Appendix A, we provide the IND-based security analysis of our proposed UNP-IPFE construction.

2 Technical Overview

This section gives an overview of how to achieve UP-IPFE schemes with semi-generic and generic approaches. Before going into the technical details, we discuss the notion of UP-IPFE in a bit more detail.

UP-IPFE and its Variants. The setup algorithm generates a pair of master public-private keys. A secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$ is generated using the master secret key where $\mathbf{y} = (y_i)_{i\in I_\mathbf{y}}, \mathbf{v} = (v_j)_{j\in I_\mathbf{v}}$ are denoted as key and predicate vectors respectively. A ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$ is computed using the master public key where $\mathbf{x} = (x_i)_{i\in[m_1]}$ and $\mathbf{w} = (w_j)_{j\in[m_2]}$ represents the message and attribute vectors respectively. The decryption recovers $\langle \mathbf{x}, \mathbf{y}\rangle$ depending upon the value of $R(\mathbf{w}, \mathbf{v})$ and a relation between the index sets. Note that, the inner product computation is defined based on the relation between the index sets of the vectors involved:

- *permissive relation* $\mathcal{R}_p$: $(\mathbf{a} = (a_i)_{i\in I_\mathbf{a}}, \mathbf{b} = (b_j)_{j\in I_\mathbf{b}}) \in \mathcal{R}_p$ if and only if $I_\mathbf{b} \subseteq I_\mathbf{a}$. In this case, we define $\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i\in I_\mathbf{b}} a_i b_i$.
- *strict relation* $\mathcal{R}_s$: $(\mathbf{a} = (a_i)_{i\in I_\mathbf{a}}, \mathbf{b} = (b_j)_{j\in I_\mathbf{b}}) \in \mathcal{R}_s$ if and only if $I_\mathbf{a} = I_\mathbf{b} = I$. In this case, we define $\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i\in I} a_i b_i$.

An UP-IPFE scheme is permissive if $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$. On the other hand, if $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_s$ then the UP-IPFE is said to be strict. Next, the permissive/strict UP-IPFE is further classified according to $R(\mathbf{w}, \mathbf{v})$:

- *zero predicate* or UZP-IPFE: $R(\mathbf{w}, \mathbf{v}) = 1$ if and only if $\langle \mathbf{w}, \mathbf{v}\rangle = 0$.
- *non-zero predicate* or UNP-IPFE: $R(\mathbf{w}, \mathbf{v}) = 1$ if and only if $\langle \mathbf{w}, \mathbf{v}\rangle \neq 0$.

We call a secret key *accepting* (resp. *non-accepting*) if it can decrypt (resp. fails to decrypt) a given ciphertext. The goal of *full attribute-hiding* security is to restrict any adversary from extracting information other than $\langle \mathbf{x}, \mathbf{y}\rangle$ when $R(\mathbf{w}, \mathbf{v}) = 1$ even given many accepting and non-accepting secret keys with respect to the challenge ciphertext. In contrast, the *weak attribute-hiding* notion allows an adversary to query any polynomial number of non-accepting and accepting keys with certain *restriction* on the predicate vectors associated with the accepting keys. The adversary is allowed to learn a set of the inner product values between the predicate and attribute vectors, so it is impossible to recover the challenge attribute vector from the set. Such restriction on the predicate vectors has also been considered in the weak attribute-hiding (bounded) P-IPFE of Agrawal et al. [AGT21b].

Our first construction is a permissive UZP-IPFE scheme that achieves semi-adaptive full attribute-hiding indistinguishability-based ($\mathsf{SA\text{-}FAH\text{-}IND}$) security in the standard model under the SXDH assumption. Our second contribution is a strict UNP-IPFE scheme in the secret key setting, i.e., the encryption is performed in the presence of the master secret key. The strict UNP-IPFE achieves semi-adaptive weak attribute-hiding simulation-based ($\mathsf{SA\text{-}WAH\text{-}SIM}$) security under the standard bilateral k-Lin assumption. Our UZP-IPFE is more technical and semi-generic, whereas the UNP-IPFE is simple and generic, as discussed next.

2.1 Public Key UP-IPFE: UZP-IPFE

Our first contribution is a full attribute-hiding UZP-IPFE that on a high level utilizes pairing-based dual system encryption techniques [Wat09]. The starting point of the construction is the public key UIPFE scheme of Tomida and Takashima [TT18], hereafter denoted by TT18. Since UZP-IPFE is a particular class of AB-IPFE in the sense that one gets the inner product value if the attribute is satisfying, while the adversary is allowed to query both accepting and non-accepting keys. Consequently, such an adversary is more powerful than the UIPFE [TT18, DSP19] or UZIPE [OT12b, DPD21] adversary. Existing works [ACGU20, AGT21b, PD21, LLW21] already have noted this fact with a conclusion that it is highly unlikely to obtain AB-IPFE, even in the bounded setting, by combining an IPFE with an ABE generically. Therefore, the possible path of building the full attribute-hiding UZP-IPFE from TT18 and the UZIPE of Okamoto and Takashima [OT12b] is uncertain and might be unrealizable.

To achieve the *unbounded* property with permissive relation, TT18 indeed employed the index encoding technique of Okamoto and Takashima [OT12b]. The purpose of encoding indices into the secret keys and ciphertexts is to generate additional entropy which prevents an adversary to learn extra information about the message vector via a key vector that does not belong to the permissive relation. In this work, we extend such an encoding technique in the context of UZP-IPFE and devise a novel procedure to combine TT18 and [OT12b] in a semi-generic way to achieve our goal.

Main Intuition. We start by discussing our core idea for the construction of UZP-IPFE. Recall that, a ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$ encodes two vectors $\mathbf{x} = (x_i)_{i\in[m_1]}, \mathbf{w} = (w_i)_{i\in[m_2]}$ and a secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$ encodes two vectors $\mathbf{y} = (y_i)_{i\in I_{\mathbf{y}}}, \mathbf{v} = (v_i)_{i\in I_{\mathbf{v}}}$ such that the scheme outputs $\langle \mathbf{x}, \mathbf{y}\rangle$ if $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ and $R(\mathbf{w}, \mathbf{v}) = 1$, i.e., $\langle \mathbf{w}, \mathbf{v}\rangle = 0$. As a starting point, we set concatenated vectors $(\mathbf{x}, \mathbf{w})$ and $(\mathbf{y}, \mathbf{v})$ as the message and key vector into the UIPFE of TT18. Observe that, this naturally satisfy the required functionality, i.e., by the correctness of UIPFE one obtains $\langle \mathbf{w}, \mathbf{v} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle$. Thus, the sum leads to $\langle \mathbf{x}, \mathbf{y} \rangle$ if $\langle \mathbf{w}, \mathbf{v} \rangle = 0$. However, in the case of $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$, the sum is easily distinguishable to an adversary from a random entity. In the next step, we avoid such a trivial attack by randomizing the predicate and attribute vectors. In particular, $\mathbf{w}$ and $\mathbf{v}$ are replaced with $\delta \mathbf{w}$ and $\omega \mathbf{v}$ respectively for uniformly random δ, ω. Now, the sum becomes $\delta\omega\langle \mathbf{w}, \mathbf{v} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle$ and we might hope to hide $\langle \mathbf{x}, \mathbf{y} \rangle$ whenever $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$. Our construction is based on this basic intuition, although many challenges await to be overcome.

At a first glance, the basic scheme described above follows the correctness of a UZP-IPFE. However, it is easy to see that the scheme already fails to satisfy the desired permissiveness since $I_{\mathbf{y}} \cup I_{\mathbf{v}} \subseteq [m_1 + m_2]$ does not guarantee that $I_{\mathbf{y}} \subseteq [m_1]$ and $I_{\mathbf{v}} \subseteq [m_2]$. Another concern arises regarding the full attribute-hiding security. In particular, the $\mathsf{SA\text{-}FAH\text{-}IND}$ security enables an adversary to make both accepting and non-accepting queries, meaning that for the challenge message pair $(\mathbf{w}^{(0)}, \mathbf{x}^{(0)}), (\mathbf{w}^{(1)}, \mathbf{x}^{(1)})$, the adversary can query secret keys with $(\mathbf{v}, \mathbf{y}, I_{\mathbf{v}}, I_{\mathbf{y}})$ where either $\langle \mathbf{w}^{(0)}, \mathbf{v} \rangle \neq 0, \langle \mathbf{w}^{(1)}, \mathbf{v} \rangle \neq 0$ or $\langle \mathbf{w}^{(0)}, \mathbf{v} \rangle = \langle \mathbf{w}^{(1)}, \mathbf{v} \rangle = 0$ and $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$. Simply applying the proof technique of TT18 does not work for us in simulating the non-accepting keys as the equality $\delta\omega\langle \mathbf{w}^{(0)}, \mathbf{v} \rangle + \langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \delta\omega\langle \mathbf{w}^{(1)}, \mathbf{v} \rangle + \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$ would not hold for such keys with high probability. Hence, encrypting the vectors $\mathbf{x}, \mathbf{w}$ together using TT18 seems problematic in realizing UZP-IPFE.

We therefore take a different route here. Our next approach is to encrypt $\mathbf{w}$ and $\mathbf{x}$ using two encryption calls of TT18 and the secret key consists of two TT18 keys corresponding to $(\mathbf{v}, I_{\mathbf{v}})$ and $(\mathbf{y}, I_{\mathbf{y}})$. This allows us to achieve the desired correctness property (i.e., permissiveness) and put security restrictions separately on $(\mathbf{w}, (\mathbf{v}, I_{\mathbf{v}}))$ and $(\mathbf{x}, (\mathbf{y}, I_{\mathbf{y}}))$. However, two independent TT18 keys in the modified construction actually opens door to a mix-n-match attack. In particular, given secret keys $\mathsf{SK}_{\mathbf{v},\mathbf{y}} = (\mathsf{sk}_{\mathbf{v}}, \mathsf{sk}_{\mathbf{y}})$ for $(\mathbf{v}, I_{\mathbf{v}})$, $(\mathbf{y}, I_{\mathbf{y}})$ and $\mathsf{SK}_{\mathbf{v}',\mathbf{y}'} = (\mathsf{sk}_{\mathbf{v}'}, \mathsf{sk}_{\mathbf{y}'})$ for $(\mathbf{v}', I_{\mathbf{v}'})$, $(\mathbf{y}', I_{\mathbf{y}'})$, one can create a new legitimate secret key $\mathsf{SK}_{\mathbf{v},\mathbf{y}'} = (\mathsf{sk}_{\mathbf{v}}, \mathsf{sk}_{\mathbf{y}'})$ which may lead to an attack to the UZP-IPFE.

A Middle Route. From the above discussion, it is evident that neither the idea of encrypting the concatenated vector nor the independent encryption method serves our purpose. Instead, we consider a middle route, a hybrid of these two ideas. The UZP-IPFE secret key or ciphertext is computed using two parallel TT18 key generations or encryptions, but these are not completely independent of each other. As per the construction of TT18, the secret key and ciphertext for the vectors $\mathbf{y}, \mathbf{x}$ are encoded by the components[1]

$$\mathsf{sk}_{\mathbf{y}} : [\![\mathbf{k}_i = (\rho_i(-i, 1), y_i, \gamma_i)\mathbf{B}^*]\!]_2 \quad \text{s.t.} \quad \textstyle\sum_i \gamma_i = 0; \qquad \mathsf{ct}_{\mathbf{x}} : [\![\mathbf{c}_i = (\pi_i(1, i), x_i, z)\mathbf{B}]\!]_1$$

where the bases $\mathbf{B}, \mathbf{B}^*$ are sampled from $\mathsf{GL}_4(\mathbb{Z}_p)$ according to a dual pairing vector space (DPVS) structure [OT10] and $[\![\cdot]\!]_\iota$ represents encoding vectors or matrices in the group $\mathbb{G}_\iota$. The first two entries of $\mathbf{k}_i$ or $\mathbf{c}_i$ encode the indices, the third entry encodes the vector and the randomness placed in the last entry ensures that no partial information is leaked. While calling the TT18 key generation or encryption twice for the UZP-IPFE, our idea is to *jointly* sample the randomnesses residing in the last entry. More precisely, for the pair of vectors

[1] We exclude the additional subspaces that are only necessary for security analysis.

$(\mathbf{y}, \mathbf{v})$, we employ a *joint* secret sharing protocol. A set $\mathcal{S} = \{\gamma_i, \widetilde{\gamma}_j\}_{i,j}$ of joint secret shares of zero binds the secret key parts $\mathsf{sk}_{\mathbf{y}}$ and $\mathsf{sk}_{\mathbf{v}}$ which prevents the aforementioned mix-n-match attack. On the other hand, the ciphertext parts $\mathsf{ct}_{\mathbf{x}}$ and $\mathsf{ct}_{\mathbf{w}}$ share a *common* randomness z to ensure that a secret key holder successfully combines the secret shares from $\mathcal{S}$ at the time of the decryption. Applying these ideas we now present a simplified UZP-IPFE scheme as follows.

$$\begin{array}{ll} \mathsf{SK}_{\mathbf{y},\mathbf{v}}: & [\![\mathbf{k}_i = (\rho_i(-i, 1), y_i, \gamma_i)\mathbf{B}^*]\!]_2 \\ & [\![\mathbf{k}_j = (\widetilde{\rho}_j(-j, 1), \omega v_j, \widetilde{\gamma}_j)\mathbf{B}^*]\!]_2 \quad \text{s.t.} \quad \sum_i \gamma_i + \sum_j \widetilde{\gamma}_j = 0; \end{array}$$

$$\begin{array}{ll} \mathsf{CT}_{\mathbf{x},\mathbf{w}}: & [\![\mathbf{c}_i = (\pi_i(1, i), x_i, z)\mathbf{B}]\!]_1 \\ & [\![\mathbf{c}_j = (\widetilde{\pi}_j(1, j), \delta w_j, z)\mathbf{B}]\!]_1. \end{array}$$

The hybrid approach makes sure that the UZP-IPFE satisfies the desired permissive property individually for the pair of vectors $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{w}, \mathbf{v})$. At the same time, it restricts an adversary to combine different secret keys and eventually mount an attack to the system. However, the scheme allows an adversary to perform a different kind of mix-n-match attack. Suppose the index sets corresponding to the vectors satisfy the condition $I_{\mathbf{y}} \subseteq I_{\mathbf{w}}$ and $I_{\mathbf{v}} \subseteq I_{\mathbf{x}}$ then it is possible to pair $\mathbf{k}_i$ with $\mathbf{c}_j$ and $\mathbf{k}_j$ with $\mathbf{c}_i$ and obtain the sum $\delta\langle \mathbf{w}, \mathbf{y}\rangle + \omega\langle \mathbf{x}, \mathbf{v}\rangle$. Now, if the vectors are chosen such that $\langle \mathbf{w}, \mathbf{y}\rangle = 0$ and $\langle \mathbf{x}, \mathbf{v}\rangle$ comes from a polynomial range then it is possible to extract unwanted information about the message vector $\mathbf{x}$. To prevent such an attack by cross pairing, we use different pair of bases $(\mathbf{B}, \mathbf{B}^*)$ for encoding $\mathbf{x}, \mathbf{y}$ and $(\widetilde{\mathbf{B}}, \widetilde{\mathbf{B}}^*)$ for encoding $\mathbf{w}, \mathbf{v}$. Next, we briefly describe the security of our UZP-IPFE.

Remaining Challenges. It remains to discuss the full attribute-hiding security of the scheme. Although our secret keys and ciphertexts are closely distributed to the TT18 framework, several technical challenges remain to be addressed due to the strong security requirement. As discussed earlier, an adversary of UZP-IPFE is more powerful than the UIPFE or TT18 in the sense that we need to additionally restrict the adversary to gain any information about the message/attribute vector from a non-accepting key that satisfies the permissive relation $\mathcal{R}_p$, but the zero-predicate relation R does not hold. On the other hand, no security can be guaranteed for the encrypted message if an adversary of UZIPE [OT12b] gets to see an accepting key. In contrast, our UZP-IPFE must ensure security for the message and attribute vectors against an adversary that holds the power of UIPFE and (full attribute-hiding) UZIPE. We acquire such a strong notion of security by extending the framework of TT18 from UIPFE to UZP-IPFE, i.e., from *unbounded length message hiding* to *unbounded length message-attribute hiding* in the context of FE.

We now briefly discuss the IND security outline of the UZP-IPFE scheme. Suppose $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)})$ and $(\mathbf{x}^{(1)}, \mathbf{w}^{(1)})$ are the challenge message-attribute vector pairs. The adversary can ask mainly the following three types of secret keys for the key-predicate pair $(\mathbf{y}, \mathbf{v})$:

1. $(\mathbf{x}^{(0)}, \mathbf{y}) \notin \mathcal{R}_p$ or $(\mathbf{w}^{(0)}, \mathbf{v}) \notin \mathcal{R}_p$.
2. $(\mathbf{x}^{(0)}, \mathbf{y}), (\mathbf{w}^{(0)}, \mathbf{v}) \in \mathcal{R}_p$, but $R(\mathbf{w}^{(0)}, \mathbf{v}) \neq 1$ and $R(\mathbf{w}^{(1)}, \mathbf{v}) \neq 1$.

3. $(\mathbf{x}^{(0)}, \mathbf{y}), (\mathbf{w}^{(0)}, \mathbf{v}) \in \mathcal{R}_p$ and $R(\mathbf{w}^{(0)}, \mathbf{v}) = R(\mathbf{w}^{(1)}, \mathbf{v}) = 1$ and $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$.

To handle the secret key queries of type 1, we use techniques from previous works [OT12b, TT18]. In particular, we add one additional subspace to the encoded secret key vectors and fill it with one copy of $\mathcal{S}$, say $\mathcal{S}^{\mathsf{copy}} = \{\gamma_i^{\mathsf{copy}}, \tilde{\gamma}_j^{\mathsf{copy}}\}_{i,j}$ and we use z^{copy} into the corresponding entry of the encoded ciphertext vectors. Next, we replace $\mathcal{S}^{\mathsf{copy}}$ with uniform shares $\mathcal{S}^{\mathsf{rand}} = \{\gamma_i^{\mathsf{rand}}, \tilde{\gamma}_j^{\mathsf{rand}}\}_{i,j}$ using the amplified entropy generated from the encoded indices for non-permissive keys. This prevents decryption by type 1 secret keys. We apply a similar strategy for simulating the type 2 keys. However, we fail to replace $\mathcal{S}^{\mathsf{copy}}$ by $\mathcal{S}^{\mathsf{rand}}$ using the entropy amplification technique used by [TT18] as the vectors satisfy the permissive relation. One hope is to procreate the required entropy using the condition that $\langle \mathbf{w}^{(b)}, \mathbf{v} \rangle \neq 0$, which is exactly the direction we follow. To execute this step, the simulator requires the information of $R(\mathbf{w}^{(b)}, \mathbf{v})$. Thus, the pair of attributes $(\mathbf{w}^{(0)}, \mathbf{w}^{(1)})$ should be available while simulating the type 2 secret keys. Hence, the simulator needs to know the challenge attributes before replying to the adversary's key queries. Finally, we are left with the accepting or type 3 key queries. In this case, we utilize two linear transformations using the facts $\langle \mathbf{w}^{(0)} - \mathbf{w}^{(1)}, \mathbf{v} \rangle = 0$ and $\langle \mathbf{x}^{(0)} - \mathbf{x}^{(1)}, \mathbf{y} \rangle = 0$ to ensure that the adversary gains no information about the challenge bit b using the type 3 secret keys. Although the core technical idea discussed above provides a very high level intuition on how we achieve the full attribute-hiding security of UZP-IPFE, there are several subtle challenges faced while adapting the framework of TT18 into our setting. We present a complete and formal security analysis in Section 4.2.

2.2 Secret Key UP-IPFE: UNP-IPFE

We construct an UP-IPFE with non-zero inner product predicate having succinct secret keys and compact ciphertexts. In particular, we provide a generic construction of a weak attribute-hiding simulation secure UNP-IPFE. Although our UNP-IPFE is built in the secret key setting, it has the nice advantage over the proposed UZP-IPFE of having constant size secret keys, that is the secret key size does not depend on the (unbounded) length of predicate or key vectors. The ciphertext must depend on the length of the message as well as the attribute vectors since we aim to achieve attribute-hiding security. However, a compact ciphertext should only grow linearly with those lengths. Recall that, the secret key and ciphertext both grow linearly with the length of vectors in case of our IND-based secure UZP-IPFE. Further, we provide security of UNP-IPFE in the SIM-based model which is known to be stronger than the IND-based model [BSW11]. To the best of our knowledge, no unbounded AB-IPFE features properties such as attribute-hiding in the simulation setting and succinctness of secret keys.

Main Idea. The starting point is the generic transformation of a NIPE scheme from an IPFE in the bounded-vector setting by [KY19, PD21]. The generic construction encrypts two vectors independently: the attribute vector $\mathbf{w}$ and payload multiplied with the attribute vector $M\mathbf{w}$ using the IPFE scheme. If a IPFE secret key $\mathsf{sk}_{\mathbf{v}}$ is given then we first recover $\langle \mathbf{w}, \mathbf{v} \rangle$, $M\langle \mathbf{w}, \mathbf{v} \rangle$ and ultimately the payload M if $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$. We need to recover an inner product value instead of a payload in our setting. Our idea is to replace the IPFE with an existing UIPFE scheme and encrypt the vector $(\mathbf{x} \otimes \mathbf{w})$. This yields a UNP-IPFE scheme as follows. Suppose $\mathsf{UIPFE} = (\mathsf{iSetup}, \mathsf{iKeyGen}, \mathsf{iEnc}, \mathsf{iDec})$ be a pairing-based UIPFE scheme [DSP19, TT18].

$$\mathsf{SK}_{\mathbf{y},\mathbf{v}} : \quad \begin{array}{l} \mathsf{isk}_{\mathbf{y}\otimes\mathbf{v}} \leftarrow \mathsf{iKeyGen}(\mathbf{y} \otimes \mathbf{v}) \\ \mathsf{isk}_{\mathbf{v}} \leftarrow \mathsf{iKeyGen}(\mathbf{v}) \end{array} \qquad \mathsf{CT}_{\mathbf{x},\mathbf{w}} : \quad \begin{array}{l} \mathsf{ict}_{\mathbf{x}\otimes\mathbf{w}} \leftarrow \mathsf{iEnc}(\mathbf{x} \otimes \mathbf{w}) \\ \mathsf{ict}_{\mathbf{w}} \leftarrow \mathsf{iEnc}(\mathbf{w}). \end{array}$$

At the time of the decryption, we recover[2] $\langle \mathbf{x}, \mathbf{y} \rangle$ from the outcomes $\langle \mathbf{x} \otimes \mathbf{w}, \mathbf{y} \otimes \mathbf{v} \rangle = \langle \mathbf{x}, \mathbf{y} \rangle \cdot \langle \mathbf{w}, \mathbf{v} \rangle$ and $\langle \mathbf{w}, \mathbf{v} \rangle$ of iDec, if $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$. We seem to be on the verge of the desired solution, but the ciphertext size is unacceptable since it swallows a quadratic factor with the lengths of $\mathbf{x}$ and $\mathbf{w}$. Our next idea is to employ a UQFE scheme [Tom22] to compute the quadratic term $\langle \mathbf{x} \otimes \mathbf{w}, \mathbf{y} \otimes \mathbf{v} \rangle$. A UQFE scheme generates secret keys for unbounded length vectors $\mathbf{f}$ and encrypts two message vectors $\mathbf{z}_1, \mathbf{z}_2$ of arbitrary length such that the decryption only recovers $\langle \mathbf{z}_1 \otimes \mathbf{z}_2, \mathbf{f} \rangle$ if the index sets satisfy a given relation. We say that the UQFE has compact ciphertexts if the size of the ciphertexts scales linearly with the lengths of $\mathbf{z}_1$ and $\mathbf{z}_2$. This readily yields an UNP-IPFE that enjoys compact ciphertexts given that the UQFE has linear size ciphertexts. More precisely, let us consider a $\mathsf{UQFE} = (\mathsf{qSetup}, \mathsf{qKeyGen}, \mathsf{qEnc}, \mathsf{qDec})$ scheme. Then, our UNP-IPFE works as follows.

$$\mathsf{SK}_{\mathbf{y},\mathbf{v}} : \quad \begin{array}{l} \mathsf{qsk}_{\mathbf{y}\otimes\mathbf{v}} \leftarrow \mathsf{qKeyGen}(\mathbf{y} \otimes \mathbf{v}) \\ \mathsf{isk}_{\mathbf{v}} \leftarrow \mathsf{iKeyGen}(\mathbf{v}) \end{array} \qquad \mathsf{CT}_{\mathbf{x},\mathbf{w}} : \quad \begin{array}{l} \mathsf{qct}_{\mathbf{x}\otimes\mathbf{w}} \leftarrow \mathsf{qEnc}(\mathbf{x}, \mathbf{w}) \\ \mathsf{ict}_{\mathbf{w}} \leftarrow \mathsf{iEnc}(\mathbf{w}). \end{array}$$

Observe that the correctness follows similarly as discussed above. The succinctness of the UNP-IPFE depends on the succinctness of the UQFE and UIPFE. It is not difficult to prove the weak attribute-hiding (semi-adaptive) simulation security of the UNP-IPFE. In the ideal world, the functional values of the challenge message vectors are used while generating secret keys and the challenge ciphertext is computed using the simulated encryption algorithms of UIPFE and UQFE.

The UIPFE of Dufour-Sans and Pointcheval [DSP19] has succinct keys, but it is IND-based secure in the ROM. Moreover, no simulation secure succinct QFE/IPFE in the unbounded setting exists. The only UQFE scheme, proposed very recently by Tomida [Tom22], is secure in the IND-based model and both the secret key and ciphertext sizes grow linearly with the vector lengths. Further, the UQFE has much larger ciphertext than existing (bounded) QFE schemes [BCFG17, Wee20, Gay20, AGT21b]. Hence, our next target is to design a simulation-secure UQFE scheme that has constant size secret keys and compact ciphertexts.

UQFE from Pairing. We start with the recent QFE scheme by Hoeteck Wee [Wee20]. The QFE utilizes the techniques of linear function evaluations [Gay20, Lin17] to compute quadratic terms. An important property of the QFE is that the secret keys are succinct which is what we require for our UNP-IPFE. We exploit properties of the tensor product to transform the QFE of [Wee20] into UQFE that preserves the succinctness. We first revisit

[2] The inner product values are first recovered in the exponent of the target group then we extract the value $\langle \mathbf{x}, \mathbf{y} \rangle$ which comes from a polynomial range, if $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$.

the QFE of [Wee20]. Let us consider the class of quadratic functions over $\mathbb{Z}_p^n \times \mathbb{Z}_p^n$ given by $(\mathbf{z}_1, \mathbf{z}_2) \mapsto (\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top$ where $\mathbf{f} \in \mathbb{Z}_p^{n^2}$.

$$\begin{array}{lll}
\mathsf{qSetup}' : & \begin{array}{l} \mathbf{A}_1 \leftarrow \mathbb{Z}_p^{k\times n}, \mathbf{A}_2 \leftarrow \mathbb{Z}_p^{k'\times n} \\ \mathsf{qpp}' = \begin{pmatrix} \llbracket \mathbf{A}_0, \mathbf{A}_0\mathbf{W}, \mathbf{A}_1 \rrbracket_1, \\ \llbracket \mathbf{A}_1, \mathbf{A}_2 \rrbracket_2 \end{pmatrix} \end{array} & \begin{array}{l} \mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k'\times(k'+1)}, \\ \mathbf{W} \leftarrow \mathbb{Z}_p^{(k'+1)\times(k+k')n} \\ \mathsf{qmsk}' = \mathbf{W} \end{array} \\
\mathsf{qsk}'_{\mathbf{f}} : & \llbracket \mathsf{sk} = \mathbf{W}\widetilde{\mathbf{f}} \rrbracket_2, & \widetilde{\mathbf{f}} = \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_n)\mathbf{f}^\top \\ (\mathbf{I}_n \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix} \\
\mathsf{qct}'_{\mathbf{z}_1,\mathbf{z}_2} : & \begin{array}{l} \llbracket \mathbf{c}_0 = \mathbf{s}_0\mathbf{A_0} \rrbracket_1, \llbracket \mathbf{c}_1 = \mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1 \rrbracket_1, \\ \llbracket \mathbf{c}_2 = \mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2 \rrbracket_2, \end{array} & \begin{array}{l} \mathbf{s}_1 \leftarrow \mathbb{Z}_p^k, , \mathbf{s}_0, \mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k'} \\ \llbracket \mathbf{c}_3 = \mathbf{s}_0\mathbf{A}_0\mathbf{W} + (\mathbf{s_1} \otimes \mathbf{z_2} \parallel \mathbf{c_1} \otimes \mathbf{s_2}) \rrbracket_1. \end{array}
\end{array}$$

The decryption algorithm extracts $\llbracket (\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top \rrbracket_T$ from the product $\llbracket (\mathbf{c}_1 \otimes \mathbf{c}_2)\mathbf{f}^\top \rrbracket_T$ by getting rid of the extra term with the help of $\llbracket \mathsf{sk} \rrbracket_2, \llbracket \mathbf{c_0} \rrbracket_1$ and $\llbracket \mathbf{c}_3 \rrbracket_1$. To upgrade the scheme into UQFE, we need to run the setup independent of the vector lengths. If we allow using hash functions (to be modeled as ROM in the security proof), then one would have generated the matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ on the fly depending on the indices of the vectors. However, it is not so trivial to compute $\mathbf{W}$ on the fly by hashing the indices directly. This is because $\mathbf{W}$ depends on the indices of both $\mathbf{z}_1$ and $\mathbf{z}_2$ as well as it must scale with the row-numbers of $\mathbf{A}_1$ and $\mathbf{A}_2$. Our idea is to split $\mathbf{W}$ using the properties of tensor product. In particular, we write it as

$$\mathbf{W} = \left[\underbrace{(\mathbf{W}_1 \otimes \mathbf{w}_1)}_{(k'+1)\times kn} \parallel \underbrace{(\mathbf{W}_2 \otimes \mathbf{w}_2)}_{(k'+1)\times k'n} \right]$$

where $\mathbf{W}_1 \in \mathbb{Z}_p^{(k'+1)\times k}$ and $\mathbf{W}_2 \in \mathbb{Z}_p^{(k'+1)\times k'}$ are chosen at the system setup and the vectors $\mathbf{w}_1, \mathbf{w}_2 \in \mathbb{Z}_p^n$ are generated using a hash function. The reader might wonder whether we are done with constructing UQFE (in the public key setting), but $\mathbf{W}$ is the master secret key and hence the security of the system is at stake if we make some parts of $\mathbf{W}$ publicly computable. We surpass the vulnerability by replacing the hash function with a pseudorandom function, which eventually leads to a secret key UQFE with the desired properties. More precisely, our UQFE works as follows:

$$\begin{array}{lll}
\mathsf{qSetup} : & \begin{array}{l} K \leftarrow \mathcal{K} \\ \mathsf{qpp} = \begin{pmatrix} \llbracket \mathbf{A}_0\mathbf{W}_1, \mathbf{A}_0\mathbf{W}_2 \rrbracket_1, \\ \llbracket \mathbf{A}_0 \rrbracket_1 \end{pmatrix} \end{array} & \begin{array}{l} \mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k'\times(k'+1)}, \\ \mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1)\times k'}, \mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1)\times k} \\ \mathsf{qmsk} = \mathbf{W}_1, \mathbf{W}_2, K \end{array} \\
\mathsf{qsk}_{\mathbf{f}} : & \begin{array}{l} \widetilde{\mathbf{W}}_1 = \mathbf{W}_1 \otimes \mathrm{PRF}(K, I_{\mathbf{f}_1}), \mathsf{H}_1(I_{\mathbf{f}_1}) = (\llbracket \mathbf{A}_1 \rrbracket_1, \llbracket \mathbf{A}_1 \rrbracket_2) \\ \widetilde{\mathbf{W}}_2 = \mathbf{W}_2 \otimes \mathrm{PRF}(K, I_{\mathbf{f}_2}), \mathsf{H}_2(I_{\mathbf{f}_2}) = \llbracket \mathbf{A}_2 \rrbracket_2 \\ \llbracket \mathsf{sk} = \mathbf{W}_{\mathbf{f}_1,\mathbf{f}_2}\widetilde{\mathbf{f}} \rrbracket_2 \end{array} & \begin{array}{l} \mathbf{W}_{\mathbf{f}_1,\mathbf{f}_2} = (\widetilde{\mathbf{W}}_1 \parallel \widetilde{\mathbf{W}}_2) \\ \widetilde{\mathbf{f}} = \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_n)\mathbf{f}^\top \\ (\mathbf{I}_n \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix} \end{array} \\
\mathsf{qct}_{\mathbf{z}_1,\mathbf{z}_2} : & \begin{array}{l} \llbracket \mathbf{c}_0 = \mathbf{s}_0\mathbf{A_0} \rrbracket_1, \llbracket \mathbf{c}_1 = \mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1 \rrbracket_1, \\ \llbracket \mathbf{c}_2 = \mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2 \rrbracket_2, \end{array} & \begin{array}{l} \mathbf{s}_1 \leftarrow \mathbb{Z}_p^k, , \mathbf{s}_0, \mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k'} \\ \llbracket \mathbf{c}_3 = \mathbf{s}_0\mathbf{A}_0\mathbf{W}_{\mathbf{z}_1,\mathbf{z}_2} + (\mathbf{s_1} \otimes \mathbf{z_2} \parallel \mathbf{c_1} \otimes \mathbf{s_2}) \rrbracket_1 \end{array}
\end{array}$$

where we assume that the secret key vector $\mathbf{f}$ is associated with the index set of the form $I_{\mathbf{f}} = I_{\mathbf{f}_1} \otimes I_{\mathbf{f}_2}$ such that $I_{\mathbf{f}_1}$ and $I_{\mathbf{f}_2}$ corresponds to the weights of $\mathbf{z}_1$ and $\mathbf{z}_2$ respectively. Note that, the correctness of the scheme follows similarly as in the above QFE if $\mathbf{W}_{\mathbf{z}_1,\mathbf{z}_2} = \mathbf{W}_{\mathbf{f}_1,\mathbf{f}_2} = \mathbf{W}$, i.e., the decryption recovers $(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top$ if $(\mathbf{f}_1, \mathbf{z}_1), (\mathbf{f}_2, \mathbf{z}_2) \in \mathcal{R}_s$ where $\mathbf{f} = \mathbf{f}_1 \otimes \mathbf{f}_2$ (according to $I_{\mathbf{f}_1}$, $I_{\mathbf{f}_2}$). Thus, we are able to upgrade Wee's QFE to a *strict* UQFE scheme in the secret key setting based on the ROM. On the positive side, our UQFE achieves efficiency identical to [Wee20] regarding the secret key and ciphertext sizes, that is the UQFE preserves succinctness of the secret keys and compactness of the ciphertexts. Although the UQFE of Tomida [Tom22] is built in the public key setting and satisfy permissiveness based on the ROM, the scheme does not satisfy succinctness and is proven secure in the IND-based model. Moreover, our UQFE is simple to understand whereas the UQFE of [Tom22] is much more complicated and requires a newly tailored building block, namely partially hiding unbounded slot IPFE [Tom22]. Lastly, we note that UIPFE is a particular case of UQFE and hence we achieve a strict (secret key) UNP-IPFE by plugging our strict UQFE into the generic transformation described above. Moreover, a permissive (public key) UNP-IPFE scheme can be obtained by plugging the permissive UQFE of [Tom22] and the UIPFE of [TT18] into our generic UNP-IPFE construction that achieves IND-based security in the ROM.

3 Preliminaries

Notations. For $a, b \in \mathbb{N}$ where $a < b$, we denote by $[a, b]$ the set $\{a, \ldots, b\}$ and $[a] = [1, a] = \{1, \ldots, a\}$. For some prime p, $\mathbb{Z}_p$ denotes a finite field of order p. For some $n \in \mathbb{N}$, $\mathsf{GL}_n(\mathbb{Z}_p)$ denotes the set of all $n \times n$ invertible matrices with entries from $\mathbb{Z}_p$. We indicate by $a \leftarrow S$ the process of random sampling of an element a from the finite set S. For a distribution $\mathcal{X}$, we write $x \leftarrow \mathcal{X}$ to denote that x is sampled at random according to distribution $\mathcal{X}$. We consider a bold uppercase letter to represent a matrix, e.g., $\mathbf{A}$, a bold lowercase letter to indicate a vector, e.g., $\mathbf{x}$ and $I_{\mathbf{x}}$ denotes the index set of the vector $\mathbf{x}$. For example, if $\mathbf{x} = (x_1, x_3, x_8)$ then we write $I_{\mathbf{x}} = \{1, 3, 8\}$. We denote by $\mathbf{A} \otimes \mathbf{B}$ the tensor product between the matrices $\mathbf{A}$ and $\mathbf{B}$. Consider g_ι is a generator of the cyclic group $\mathbb{G}_\iota$. If $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is an n-tuple vector then $[\![\mathbf{x}]\!]_\iota = (g_\iota^{x_1}, g_\iota^{x_2}, \ldots, g_\iota^{x_n})$. For a matrix $\mathbf{A} = (a_{ij}) \in \mathsf{GL}_n(\mathbb{Z}_p)$, we define $[\![\mathbf{A}]\!]_\iota$ as

$$[\![\mathbf{A}]\!]_\iota = \begin{bmatrix} g_\iota^{a_{11}} & g_\iota^{a_{12}} & \cdots & g_\iota^{a_{1n}} \\ g_\iota^{a_{21}} & g_\iota^{a_{22}} & \cdots & g_\iota^{a_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ g_\iota^{a_{n1}} & g_\iota^{a_{n2}} & \cdots & g_\iota^{a_{nn}} \end{bmatrix}.$$

Let $\mathbf{I}_n$ denote an $n \times n$ identity matrix and $\mathbf{A}^\top$ signifies the transpose of the matrix $\mathbf{A}$. We use '$\approx_s$' to denote two distributions being statistically indistinguishable, '$\approx_c$' to denote two distributions being computationally indistinguishable, and '$\equiv$' to denote two distributions being identically distributed. Concatenation between two matrices or vectors is denoted by the symbol '$\|$'. For $\mathbb{R}_{[0,1]} = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$, a function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}_{[0,1]}$ is said to be *negligible* if for every $c \in \mathbb{N}$ there exists a $\lambda_c \in \mathbb{N}$ such that $\mathsf{negl}(\lambda) \leq \frac{1}{\lambda^c}$ for all $\lambda > \lambda_c$.

3.1 Bilinear Groups

A bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ consists of a prime p, two multiplicative source groups $\mathbb{G}_1, \mathbb{G}_2$ and a target group $\mathbb{G}_T$ with the order $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$ where g_1, g_2 are the generators of the group $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. Let us consider a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. It satisfies the following:

- *bilinearity:* $e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}$ for all $g_1 \in \mathbb{G}_1, g_2 \in \mathbb{G}_2,\ a, b \in \mathbb{Z}_p$ and
- *non-degeneracy:* $e(g_1, g_2)$ is a generator of $\mathbb{G}_T$.

A bilinear group generator $\mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ takes the security parameter λ and outputs a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ with a λ-bit prime integer p.

3.2 Complexity Assumptions

Assumption 1 (Symmetric External Diffie-Hellman (SXDH)). *For $\iota \in \{1, 2\}$, we define the distribution $(D, \llbracket t_\beta \rrbracket_\iota)$ on a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ as*

$$D = (\mathsf{G}, \llbracket a \rrbracket_\iota, \llbracket u \rrbracket_\iota) \text{ for } a, u \leftarrow \mathbb{Z}_p$$
$$\llbracket t_\beta \rrbracket_\iota = \llbracket au + \beta f \rrbracket_\iota \text{ for } \beta \in \{0, 1\} \text{ and } f \leftarrow \mathbb{Z}_p.$$

We say that the SXDH *assumption holds in* G *if for all PPT adversaries $\mathcal{A}$, if there exists a negligible function* $\mathsf{negl}(\cdot)$ *satisfying the following:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{SXDH}}(\lambda) := |\Pr[\mathcal{A}(D, \llbracket t_0 \rrbracket_\iota) = 1] - \Pr[\mathcal{A}(D, \llbracket t_1 \rrbracket_\iota) = 1]| \leq \mathsf{negl}(\lambda).$$

Assumption 2 (Matrix Decisional Diffie-Hellman ($\mathsf{MDDH}_{k,\ell}^d$)). *Consider a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ with $k, \ell, d \in \mathbb{N}$. We say that the $\mathsf{MDDH}_{k,\ell}^d$ assumption holds in G if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ satisfying the following*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{MDDH}_{k,\ell}^d}(\lambda) := |\Pr[\mathcal{A}(\mathsf{G}, \llbracket \mathbf{A} \rrbracket_1, \llbracket \mathbf{AB} \rrbracket_1) = 1] - \Pr[\mathcal{A}(\mathsf{G}, \llbracket \mathbf{A} \rrbracket_1, \llbracket \mathbf{R} \rrbracket_1) = 1]| \leq \mathsf{negl}(\lambda)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{\ell \times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{k \times d}$ with $\mathbf{R} \leftarrow \mathbb{Z}_p^{\ell \times d}$.

Remark 1. *The* MDDH *assumption on $\mathbb{G}_2$ can be defined in an analogous way. Escala et al. [EHK+17] showed that*

$$k\text{-}\mathsf{Lin} \implies \mathsf{MDDH}_{k,k+1}^1 \implies \mathsf{MDDH}_{k,\ell}^d\ \forall k, d \geq 1, \ell > k$$

with a tight security reduction. For $\ell \leq k$, the $\mathsf{MDDH}_{k,\ell}^d$ assumption also holds unconditionally.

Assumption 3 (Bilateral Matrix Decisional Diffie-Hellman (*bi*-$\mathsf{MDDH}_{k,\ell}^d$)). *Consider a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$ with $k, \ell, d \in \mathbb{N}$. We say that the bi-$\mathsf{MDDH}_{k,\ell}^d$ assumption holds in G if for all PPT adversaries $\mathcal{A}$, there exists a negligible function $\mathsf{negl}(\cdot)$ satisfying the following*

$$\mathsf{Adv}_{\mathcal{A}}^{bi\text{-}\mathsf{MDDH}_{k,\ell}^d}(\lambda) := |\Pr[\mathcal{A}(\mathsf{G}, \llbracket \mathbf{A} \rrbracket_1, \llbracket \mathbf{A} \rrbracket_2, \llbracket \mathbf{AB} \rrbracket_1, \llbracket \mathbf{AB} \rrbracket_2) = 1] - \Pr[\mathcal{A}(\mathsf{G}, \llbracket \mathbf{A} \rrbracket_1, \llbracket \mathbf{A} \rrbracket_2, \llbracket \mathbf{R} \rrbracket_1, \llbracket \mathbf{R} \rrbracket_2) = 1]| \leq \mathsf{negl}(\lambda)$$

where $\mathbf{A} \leftarrow \mathbb{Z}_p^{\ell \times k}, \mathbf{B} \leftarrow \mathbb{Z}_p^{k \times d}$ with $\mathbf{R} \leftarrow \mathbb{Z}_p^{\ell \times d}$.

3.3 Dual Pairing Vector Space

Let $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T,\ g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$. For a natural number $n \in \mathbb{N}$, we generate a random dual orthonormal bases $(\mathbf{B}, \mathbf{B}^*) \leftarrow \mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$ and a dual pairing vector space (DPVS) [OT12b] as $\mathsf{params}_V = (p, V, V^*, \mathbb{G}_T, A_1, A_2, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(n, \mathsf{G})$ where $\mathbf{B} \leftarrow \mathsf{GL}_n(\mathbb{Z}_p)$ and $\mathbf{B}^* = (\mathbf{B}^{-1})^\top$ are dual orthonormal bases of the vector spaces $V = \mathbb{G}_1^n$ and $V^* = \mathbb{G}_2^n$ respectively. Let $A_\kappa = (g_\kappa^{\mathbf{e}_1}, g_\kappa^{\mathbf{e}_2} \ldots, g_\kappa^{\mathbf{e}_n})$ for $\kappa = 1, 2$ where $\mathbf{e}_i = (\overbrace{0, ..., 0}^{i-1}, 1, \overbrace{0, ..., 0}^{n-i})$. Then A_1 and A_2 are the canonical basis of V and V^* respectively. Let us extend the bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ to a mapping $E : V \times V^* \to \mathbb{G}_T$ as $E([\![\mathbf{x}\mathbf{B}]\!]_1, [\![\mathbf{y}\mathbf{B}^*]\!]_2) = e(g_1, g_2)^{\langle \mathbf{x}, \mathbf{y} \rangle}$ for any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_p^n$. Then for arbitrary vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_\ell \in \mathbb{Z}_p^n$, and any matrix $\mathbf{M} \in \mathsf{GL}_n(\mathbb{Z}_p)$, the distributions $\left(\{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[\ell]}\right)$ and $\left(\{\mathbf{x}_i\mathbf{M}\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{B}^*\}_{i\in[\ell]}\right)$ are identically distributed where $\mathbf{M}^* = (\mathbf{M}^{-1})^\top$ is the orthonormal dual corresponding to the matrix $\mathbf{M}$. More generally, for any set $S \subseteq [n]$ such that $\forall i \in S, \mathbf{d}_i = \mathbf{M}^{-1}\mathbf{b}_i$, the distributions $(\{\mathbf{b}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{b}_i\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{b}_i^*\}_{i\in[\ell]})$ and $(\{\mathbf{d}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{M}\mathbf{d}_i\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{d}_i^*\}_{i\in[\ell]})$ are also identical. Therefore, $(\mathbf{D}, \mathbf{D}^*) = (\mathbf{M}^{-1}\mathbf{B}, \mathbf{M}^T\mathbf{B}^*)$ are also random dual orthonormal bases such that

$$(\{\mathbf{b}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{B}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{B}^*\}_{i\in[\ell]}) \equiv (\{\mathbf{d}_i\}_{i\in S}, \{\mathbf{x}_i\mathbf{M}\mathbf{D}\}_{i\in[k]}, \{\mathbf{y}_i\mathbf{M}^*\mathbf{D}^*\}_{i\in[\ell]}).$$

In Fig. 2, we describe a random dual orthonormal basis generator $\mathcal{G}_{\mathsf{OB.Gen}}(\mathbb{Z}_p^n)$ for some prime p and positive integer n.

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

- Chooses $\mathbb{B} \leftarrow \mathsf{GL}_n(\mathbb{Z}_p)$.
- Computes $\mathbb{B}^* = (\mathbb{B}^{-1})^\top$. Let $\mathbf{b}_i$ and $\mathbf{b}_i^*$ represent the i-th row of $\mathbb{B}$ and $\mathbb{B}^*$ respectively.
- Sets $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n)$ and $\mathbf{B}^* = (\mathbf{b}_1^*, \mathbf{b}_2^*, \ldots, \mathbf{b}_n^*)$. Note that, $(\mathbf{B}, \mathbf{B}^*)$ are dual orthonormal bases satisfying for $i, i' = 1, 2, \ldots, n$
$$\langle \mathbf{b}_i, \mathbf{b}_{i'}^* \rangle = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{elsewhere.} \end{cases}$$
- Returns $(\mathbf{B}, \mathbf{B}^*)$.

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

3.4 Pseudorandom Function

Definition 1. *A pseudo-random function (PRF) family $\mathcal{F} = \{F_K\}_{K\in\mathcal{K}_\lambda}$ with a keyspace $\mathcal{K}_\lambda$, a domain $\mathcal{X}_\lambda$ and a range $\mathcal{Y}_\lambda$ is a function family that consists of functions $F_K : \mathcal{X}_\lambda \to \mathcal{Y}_\lambda$. Let Rand_λ be the set of random functions with domain $\mathcal{X}_\lambda$ and co-domain $\mathcal{Y}_\lambda$. Then for all PPT adversaries $\mathcal{A}$, the following holds:*

$$\mathsf{Adv}_{\mathcal{A}}^{PRF}(1^\lambda) := \left|\Pr[\mathcal{A}^{F_K(\cdot)}(\lambda) = 1] - \Pr[\mathcal{A}^{\mathsf{Rand}(\cdot)}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda)$$

with $K \leftarrow \mathcal{K}_\lambda$ and $\mathsf{Rand}(\cdot) \leftarrow \mathsf{Rand}_\lambda$.

3.5 Unbounded Predicate Inner Product Functional Encryption

In the following, we define the notion of *unbounded predicate inner product functional encryption* (UP-IPFE) for the message space $\{\mathcal{X}_\lambda\}_\lambda$, an attribute space $\{\mathcal{W}_\lambda\}_\lambda$, a predicate class $\{\mathcal{P}_\lambda\}_\lambda$ and a key space $\{\mathcal{Y}_\lambda\}_\lambda$ for any $\lambda \in \mathbb{N}$ where λ denotes the security parameter. For any two vectors $\mathbf{a} = (a_i)_{i\in I_\mathbf{a}}, \mathbf{b} = (b_i)_{i\in I_\mathbf{b}}$ associated with the index sets $I_\mathbf{a}$ and $I_\mathbf{b}$, we define a *permissive* relation $\mathcal{R}_p$ such that $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_p$ if and only if $I_\mathbf{b} \subseteq I_\mathbf{a}$ and the inner product is defined as $\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i\in I_\mathbf{b}} a_i b_i$. Similarly, a *strict* relation $\mathcal{R}_s$ between the vectors $\mathbf{a}, \mathbf{b}$ is defined as $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_s$ if and only if $I_\mathbf{b} = I_\mathbf{a} = I$(say) and the inner product is given by $\langle \mathbf{a}, \mathbf{b}\rangle = \sum_{i\in I} a_i b_i$. It can be observed that if $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_p$ then $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}_s$. Now, we describe the UP-IPFE scheme with the permissive relation. Our $\mathsf{UP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ for a predicate relation $R : \mathcal{P}_\lambda \times \mathcal{W}_\lambda \rightarrow \{0, 1\}$ consists of four PPT algorithms satisfying the following requirements.

$\mathbf{Setup(1^\lambda) \rightarrow (MPK, MSK)}$: The setup algorithm takes as input the security parameter 1^λ, and outputs a master public key and master secret key pair $(\mathsf{MPK}, \mathsf{MSK})$.

$\mathbf{Enc(MPK, x, w) \rightarrow CT_{x,w}}$: The encryption algorithm takes as input the master public key MPK, a message vector $\mathbf{x} \in \mathcal{X}_\lambda$ and an attribute $\mathbf{w} \in \mathcal{W}_\lambda$ with the associated index sets $I_\mathbf{x}$, $I_\mathbf{w}$ respectively, and outputs a ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$.

$\mathbf{KeyGen(MPK, MSK, y, v) \rightarrow SK_{y,v}}$: The key generation algorithm takes as input the master public key MPK, the master secret key MSK, a key vector $\mathbf{y} \in \mathcal{Y}_\lambda$ and a predicate vector $\mathbf{v} \in \mathcal{V}_\lambda$ with the associated index sets $I_\mathbf{y}$ and $I_\mathbf{v}$ respectively, and outputs a secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$.

$\mathbf{Dec(MPK, SK_{y,v}, CT_{x,w}) \rightarrow \boldsymbol{d}/ \perp}$: The decryption algorithm takes as input the master public key MPK, the ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$, the secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$, and outputs either a decrypted value d or the special symbol $\perp$ indicating failure.

Correctness: For any $\lambda \in \mathbb{N}$, any pair of message-attribute vectors $(\mathbf{x}, \mathbf{w})$ with associated index sets $I_\mathbf{x}, I_\mathbf{w}$, any pair of key-predicate vectors $(\mathbf{y}, \mathbf{v})$ with associated index sets $I_\mathbf{y}, I_\mathbf{v}$, if $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ with $R(\mathbf{v}, \mathbf{w}) = 1$ holds, then we have

$$\Pr\left[\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathbf{y},\mathbf{v}}, \mathsf{CT}_{\mathbf{x},\mathbf{w}}) = \langle \mathbf{x}, \mathbf{y}\rangle : \begin{array}{c}(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)\\ \mathsf{CT}_{\mathbf{x},\mathbf{w}} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathbf{x}, \mathbf{w})\\ \mathsf{SK}_{\mathbf{y},\mathbf{v}} \leftarrow \mathsf{KeyGen}(\mathsf{MSK}, \mathbf{y}, \mathbf{v})\end{array}\right] = 1.$$

Depending on the inner product value $\langle \mathbf{w}, \mathbf{v}\rangle$, we classify UP-IPFE as follows:

- *unbounded zero predicate IPFE* (UZP-IPFE): decryption recovers $\langle \mathbf{x}, \mathbf{y}\rangle$ whenever $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ (or $\mathcal{R}_s$) and $R(\mathbf{w}, \mathbf{v}) = 1$ holds if and only if $\langle \mathbf{w}, \mathbf{v}\rangle = 0$.
- *unbounded non-zero predicate IPFE* (UNP-IPFE): decryption recovers $\langle \mathbf{x}, \mathbf{y}\rangle$ whenever $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ (or $\mathcal{R}_s$) and $R(\mathbf{w}, \mathbf{v}) = 1$ holds if and only if $\langle \mathbf{w}, \mathbf{v}\rangle \neq 0$.

Definition 2 (Semi-Adaptive Full Attribute-Hiding Indistinguishability). *The* $\mathsf{UP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is said to be semi-adaptive full attribute-hiding indistinguishability* ($\mathsf{SA\text{-}FAH\text{-}IND}$) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that the following holds*

$$\mathsf{Adv}^{\mathsf{UP\text{-}IPFE}}_{\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{0,\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{1,\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda)$ *is defined for* $\beta \in \{0, 1\}$ *as follows:*

```
$\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda)$
  $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$.
  $(\mathbf{w}^{(0)}, \mathbf{w}^{(1)}) \leftarrow \mathcal{A}(1^\lambda, \mathsf{MPK})$ where $|I_{\mathbf{w}^{(0)}}| = |I_{\mathbf{w}^{(1)}}|$.
  $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot,\cdot)}(\mathsf{MPK})$ where $|I_{\mathbf{x}^{(0)}}| = |I_{\mathbf{x}^{(1)}}|$.
  $\mathsf{CT}^{(\beta)}_{\mathbf{x},\mathbf{w}} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathbf{x}^{(\beta)}, \mathbf{w}^{(\beta)})$.
  $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot,\cdot)}(\mathsf{MPK}, \mathsf{CT}^{(\beta)}_{\mathbf{x},\mathbf{w}})$.
  Outputs: $\beta'$.
```

In this experiment, $\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \cdot, \cdot)$ *is an oracle that takes as input the key-predicate vector pair* $(\mathbf{y}, \mathbf{v})$ *associated with the index sets* $I_{\mathbf{y}}, I_{\mathbf{v}}$ *and outputs the secret key* $\mathsf{SK}_{\mathbf{y},\mathbf{v}} \leftarrow \mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v})$*. If* $(\mathbf{x}^{(b)}, \mathbf{y}), (\mathbf{w}^{(b)}, \mathbf{v}) \in \mathcal{R}_p$ *for all* $b \in \{0, 1\}$ *then either* $R(\mathbf{w}^{(0)}, \mathbf{v}) = R(\mathbf{w}^{(1)}, \mathbf{v}) = 0$*, or* $R(\mathbf{w}^{(0)}, \mathbf{v}) = R(\mathbf{w}^{(1)}, \mathbf{v}) = 1$ *and* $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$*.*

In this work, we consider a weaker security notion for UP-IPFE in the simulation-based model with *strict* relation between the unbounded length vectors. We *emphasize* that our weak attribute-hiding security notion also allows the adversary to query secret keys that are capable of decrypting the challenge ciphertext, however, there is a restriction on such queries.

Definition 3 (Semi-Adaptive Weak Attribute-Hiding Simulation security). *The* $\mathsf{UP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is said to be semi-adaptive weak attribute-hiding simulation* ($\mathsf{SA\text{-}WAH\text{-}SIM}$) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a PPT simulator* $\mathcal{S} := (\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$ *such that the following holds*

$$\mathsf{Adv}^{\mathsf{UP\text{-}IPFE}}_{\mathcal{A},\mathsf{SA\text{-}WAH\text{-}SIM}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UP\text{-}IPFE},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UP\text{-}IPFE},\mathcal{A},\mathcal{S}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda)$$

where the experiments $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UP\text{-}IPFE},\mathcal{A}}(\lambda)$ *and* $\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UP\text{-}IPFE},\mathcal{A},\mathcal{S}}(\lambda)$ *are defined as follows:*

```
$\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UP\text{-}IPFE},\mathcal{A}}(\lambda)$
  $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$
  $(\mathbf{x}^*, \mathbf{w}^*) \leftarrow \mathcal{A}(\mathsf{MPK})$
  $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathbf{x}^*, \mathbf{w}^*)$
  $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot,\cdot)}(\mathsf{CT}^*)$
```

```
$\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UP\text{-}IPFE},\mathcal{A},\mathcal{S}}(\lambda)$
  $(\mathsf{MPK}^*, \mathsf{MSK}^*) \leftarrow \mathsf{Setup}^*(1^\lambda)$
  $(\mathbf{x}^*, \mathbf{w}^*) \leftarrow \mathcal{A}(\mathsf{MPK}^*)$
  $\mathsf{CT}^* \leftarrow \mathsf{Enc}^*(\mathsf{MPK}^*, I_{\mathbf{x}^*}, I_{\mathbf{w}^*})$
  $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}^*(\mathsf{MPK}^*,\mathsf{MSK}^*,\cdot,\cdot,\cdot)}(\mathsf{CT}^*)$
```

In the Real *security experiment,* $\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \cdot, \cdot)$ *is an oracle that takes input the key-predicate vector pair* $(\mathbf{y}, \mathbf{v})$ *with associated index sets* $I_{\mathbf{y}}, I_{\mathbf{v}}$ *and outputs* $\mathsf{SK}_{\mathbf{y},\mathbf{v}} \leftarrow \mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v})$*. In the* Ideal *security experiment,* $\mathsf{KeyGen}^*(\mathsf{MPK}^*, \mathsf{MSK}^*, \cdot, \cdot, \cdot)$ *oracle returns the simulated secret key* $\mathsf{SK}^*_{\mathbf{y},\mathbf{v}}$ *on input a pair of key-predicate vectors* $\mathbf{y}, \mathbf{v}$ *with the associated index sets* $I_{\mathbf{y}}, I_{\mathbf{v}}$ *and a pair of values* (σ, μ) *where*

$$(\sigma, \mu) = \begin{cases} (\langle \mathbf{w}^*, \mathbf{v} \rangle, \langle \mathbf{x}^*, \mathbf{y} \rangle), & \textit{if } (\mathbf{x}^*, \mathbf{y}), (\mathbf{w}^*, \mathbf{v}) \in \mathcal{R}_s, R(\mathbf{w}^*, \mathbf{v}) = 1 \\ (\perp, \perp), & \textit{elsewhere.} \end{cases}$$

Additionally, the secret key queries must satisfy the condition that $\dim\{\mathbf{v} : (\mathbf{w}^*, \mathbf{v}) \in \mathcal{R}_s\} \leq |I_{\mathbf{w}^*}| - 1$*.*

4 Our Full Attribute-hiding UZP-IPFE

In this section, we construct a public key UZP-IPFE scheme in the permissive setting. Our scheme is based on the DPVS framework introduced by Okamoto and Takashima in [OT10].

4.1 Construction

Our $\mathsf{UZP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ scheme can be described in terms of the following algorithms. As all pairing based IPFE in the literature, our required inner product values come from a polynomial range so that at the end of the decryption phase, we can efficiently perform an exhaustive search to obtain the value.

$\mathsf{\mathbf{Setup}}(1^\lambda) \rightarrow (\mathsf{\mathbf{MPK}}, \mathsf{\mathbf{MSK}})$: The setup algorithm takes as input the security parameter λ and executes the following steps:

1. Sample a bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$.
2. Set $g_T = e(g_1, g_2)$.
3. Generate a DPVS as $\mathsf{params}_V = (p, V, V^*, \mathbb{G}_T, A_1, A_1, E) \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(7, \mathsf{G})$.
4. Sample $\mathbf{B}, \widetilde{\mathbf{B}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$.
5. Set $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$.
6. Output $\mathsf{MPK} = \left(\mathsf{PP}, \{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i \in \{1,2,\ldots,4\}}\right), \mathsf{MSK} = (\{\mathbf{b}^*_i, \widetilde{\mathbf{b}}^*_i\}_{i \in \{1,2,\ldots,4\}})$.

$\mathsf{\mathbf{Enc}}(\mathsf{\mathbf{MPK}}, \mathbf{x}, \mathbf{w}) \rightarrow \mathsf{\mathbf{CT}}_{\mathbf{x},\mathbf{w}}$: The encryption algorithm takes as input the master public key MPK, a message vector $\mathbf{x} = (x_i)_{i \in [m_1]} \in \mathbb{Z}^{m_1}$, an attribute vector $\mathbf{w} = (w_i)_{i \in [m_2]} \in \mathbb{Z}^{m_2}$ and executes the following steps:

1. Parse $\mathsf{MPK} = \left(\mathsf{PP}, \{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i \in \{1,2,\ldots,4\}}\right)$ where $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$.
2. Sample $\delta, \alpha \leftarrow \mathbb{Z}_p$ and $\pi_i, \widetilde{\pi}_j \leftarrow \mathbb{Z}_p$ for all $i \in [m_1], j \in [m_2]$.
3. Compute
$$[\![\mathbf{c}^1_i]\!]_1 = [\![(\pi_i(1, i), x_i, \alpha, 0, 0, 0)\mathbf{B}]\!]_1 \quad \forall i \in [m_1].$$
$$[\![\mathbf{c}^2_j]\!]_1 = [\![(\widetilde{\pi}_j(1, j), \delta w_j, \alpha, 0, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \quad \forall j \in [m_2].$$
4. Output $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\{[\![\boldsymbol{c}^1_i]\!]_1\}_{i \in [m_1]}, \{[\![\boldsymbol{c}^2_j]\!]_1\}_{j \in [m_2]})$.

$\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v}) \rightarrow \mathsf{SK}_{\mathbf{y},\mathbf{v}}$: The key generation algorithm takes as input the master public key MPK, the master secret key MSK, the key vector $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}} \in \mathbb{Z}^{|I_{\mathbf{y}}|}$ and the predicate vector $\mathbf{v} = (v_i)_{i \in I_{\mathbf{v}}} \in \mathbb{Z}^{|I_{\mathbf{v}}|}$ associated with the index sets $I_{\mathbf{y}}$, $I_{\mathbf{v}}$ respectively. It performs the following steps:

1. Parse $\mathsf{MPK} = \Big(\mathsf{PP}, \{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i \in \{1,2,\ldots,4\}}\Big)$ where $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$.
2. Parse $\mathsf{MSK} = (\{\mathbf{b}_i^*, \widetilde{\mathbf{b}}_i^*\}_{i \in \{1,2,\ldots,4\}})$.
3. Sample $\omega \leftarrow \mathbb{Z}_p$, $\rho_i, \widetilde{\rho}_j \leftarrow \mathbb{Z}_p$ and $\gamma_i, \widetilde{\gamma}_j \leftarrow \mathbb{Z}_p$ for all $i \in I_{\mathbf{y}}, j \in I_{\mathbf{v}}$ such that $\sum_{i \in I_{\mathbf{y}}} \gamma_i + \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j = 0$.
4. Compute
$$\mathbf{k}_i^1 = (\rho_i(-i, 1), y_i, \gamma_i, 0, 0, 0)\mathbf{B}^* \in \mathbb{Z}_p^7 \ \ \forall i \in I_{\mathbf{y}}.$$
$$\mathbf{k}_j^2 = (\widetilde{\rho}_j(-j, 1), \omega v_j, \widetilde{\gamma}_j, 0, 0, 0)\widetilde{\mathbf{B}}^* \in \mathbb{Z}_p^7 \ \ \forall j \in I_{\mathbf{v}}.$$
5. Output $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}}}, I_{\mathbf{y}}, I_{\mathbf{v}})$.

$\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathbf{y},\mathbf{v}}, \mathsf{CT}_{\mathbf{x},\mathbf{w}}) \rightarrow \boldsymbol{d}/\perp$: The decryptor takes as input the master public key MPK, a ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$ associated with the message, an attribute vector pair $\mathbf{x}, \mathbf{w}$ of length m_1, m_2 respectively and a secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$ corresponding to the key, predicate vector pair $\mathbf{y}, \mathbf{v}$ with the index sets $I_{\mathbf{y}}$, $I_{\mathbf{v}}$. Then, the decryption algorithm works as follows:

1. Parse $\mathsf{MPK} = \Big(\mathsf{PP}, \{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i \in \{1,2,\ldots,4\}}\Big)$ where $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$.
2. Parse $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}}}, I_{\mathbf{y}}, I_{\mathbf{v}})$ and $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\{[\![\boldsymbol{c}_i^1]\!]_1\}_{i \in [m_1]}, \{[\![\boldsymbol{c}_j^2]\!]_1\}_{j \in [m_2]})$
3. If $(\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_p$ or $(\mathbf{w}, \mathbf{v}) \notin \mathcal{R}_p$, output $\perp$.
4. Else, compute
$$h = \prod_{i \in I_{\mathbf{y}}} \prod_{j \in I_{\mathbf{v}}} E\Big([\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{k}_i^1]\!]_2\Big) \cdot E\Big([\![\mathbf{c}_j^2]\!]_1, [\![\mathbf{k}_j^2]\!]_2\Big).$$
5. Output $\log_{g_T} h$.

Correctness: For our above $\mathsf{UZP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ scheme, let the master public key, and the master secret key pair be $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{UZP\text{-}IPFE.Setup}(1^\lambda)$, the ciphertext be $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\{[\![\boldsymbol{c}_i^1]\!]_1\}_{i \in [m_1]}, \{[\![\boldsymbol{c}_j^2]\!]_1\}_{j \in [m_2]}) \leftarrow \mathsf{UZP\text{-}IPFE.Enc}(\mathsf{MPK}, \mathbf{x}, \mathbf{w})$ for a pair of vectors $\mathbf{x} = (x_i)_{i \in [m_1]} \in \mathbb{Z}^{m_1}, \mathbf{w} = (w_j)_{j \in [m_2]} \in \mathbb{Z}^{m_2}$ and the secret key be $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}}}, I_{\mathbf{y}}, I_{\mathbf{v}}) \leftarrow \mathsf{UZP\text{-}IPFE.KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v})$ corresponding to a pair of vectors $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}} \in \mathbb{Z}^{|I_{\mathbf{y}}|}, \mathbf{v} = (v_j)_{j \in I_{\mathbf{v}}} \in \mathbb{Z}^{|I_{\mathbf{v}}|}$. Since $\sum_{i \in I_{\mathbf{y}}} \gamma_i + \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j = 0$, the decryption succeeds if $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_p$ and $\langle \mathbf{w}, \mathbf{v} \rangle = 0$ as shown below

$$A = \prod_{i \in I_{\mathbf{y}}} E\Big([\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{k}_i^1]\!]_2\Big) = e(g_1, g_2)^{\sum_{i \in I_{\mathbf{y}}} x_i y_i + \alpha \sum_{i \in I_{\mathbf{y}}} \gamma_i} = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \alpha \sum_{i \in I_{\mathbf{y}}} \gamma_i]\!]_T.$$
$$B = \prod_{j \in I_{\mathbf{v}}} E\Big([\![\mathbf{c}_j^2]\!]_1, [\![\mathbf{k}_j^2]\!]_2\Big) = e(g_1, g_2)^{\sum_{j \in I_{\mathbf{v}}} \omega \delta v_j w_j + \alpha \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j} = [\![\omega \delta (\langle \mathbf{w}, \mathbf{v} \rangle) + \alpha \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j]\!]_T.$$
$$h = A \cdot B = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \omega \delta \langle \mathbf{w}, \mathbf{v} \rangle + \alpha (\sum_{i \in I_{\mathbf{y}}} \gamma_i + \sum_{j \in I_{\mathbf{v}}} \widetilde{\gamma}_j)]\!]_T = [\![\langle \mathbf{x}, \mathbf{y} \rangle + \omega \delta \langle \mathbf{w}, \mathbf{v} \rangle]\!]_T. \quad (1)$$

Using $\langle \mathbf{w}, \mathbf{v} \rangle = 0$, it can be seen that the correctness follows from Eq. (1).

4.2 Security Analysis

Theorem 1. *Assuming the* SXDH *assumption holds in the pairing groups, our* $\mathsf{UZP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *scheme is* $\mathsf{SA\text{-}FAH\text{-}IND}$ *secure as per the security model described in Def. 2. More precisely, if there exists a PPT adversary* $\mathcal{A}$ *that breaks the* $\mathsf{SA\text{-}FAH\text{-}IND}$ *security of our UZP-IPFE scheme then we can construct a PPT machine* $\mathcal{B}$ *against the* SXDH *assumption such that for any security parameter* λ*, the advantage*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{UZP\text{-}IPFE}}_{\mathcal{A},\mathsf{SA\text{-}FAH\text{-}IND}}(\lambda) \leq m_{1,\mathsf{max}}[16(m_{1,\mathsf{max}} + m_{2,\mathsf{max}}) +& \\ 8m_{2,\mathit{max}}(t_{\mathsf{max}} - 1) + 8(s_{\mathsf{max}} - 1) + 5]\mathsf{Adv}^{\mathit{SXDH}}_{\mathcal{B}}(\lambda) + 2^{-\Omega(\lambda)}&\end{aligned}$$

where $m_{1,\mathsf{max}}, m_{2,\mathsf{max}}$ *be the maximum length of the challenge message and attribute vectors (i.e.,* $\mathbf{x}$ *and* $\mathbf{w}$*) respectively and* $s_{\mathsf{max}}, t_{\mathsf{max}}$ *be the maximum indices of key and predicate vectors (i.e.,* $\mathbf{y}$ *and* $\mathbf{v}$*) respectively with which* $\mathcal{A}$ *queries the key generation oracle.*

Proof. To prove the above Theorem 1, we use the following lemmas.

Lemma 1. *[TT18] Let* $m = m(\lambda), n = n(\lambda)$ *be two integers. The problem 1-*SXDH $(\mathsf{P1\text{-}SXDH})$ *is to guess the bit* β*, given the following distributions:*

$$\mathsf{G} \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda), \mathsf{params}_V \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(7, \mathsf{G}), \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p).$$

$$\mathbf{u}_i = (\pi'_i(1, i), 0, 0, 0, \alpha', 0, 0)\mathbf{B} \quad \forall i \in [m] \textit{ with } \alpha', \{\pi'_i\}_{i\in[m]} \leftarrow \mathbb{Z}_p.$$

$$\mathcal{D} = (G, \mathsf{params}_V, \llbracket \mathbf{b}_1 \rrbracket_1, \llbracket \mathbf{b}_2 \rrbracket_1, \ldots, \llbracket \mathbf{b}_4 \rrbracket_1, \llbracket \mathbf{b}^*_1 \rrbracket_2, \llbracket \mathbf{b}^*_2 \rrbracket_2, \ldots, \llbracket \mathbf{b}^*_5 \rrbracket_2, \{\llbracket \mathbf{u}_i \rrbracket_1\}_{i\in[m]}).$$

$$\textit{Choose} \quad \rho'_{m+1}, \rho'_{m+2}, \ldots, \rho'_n, r'_{m+1}, r'_{m+2}, \ldots, r'_n \leftarrow \mathbb{Z}_p.$$

$$\mathbf{u}^*_{i,\beta} = (\rho'_i(-i, 1), 0, 0, \beta r'_i, 0, 0)\mathbf{B}^* \quad \forall i \in [m + 1, n].$$

$$\mathcal{U}_\beta = \{\llbracket \widehat{\mathbf{u}}^*_{i,\beta} \rrbracket_2\}_{i\in[m+1,n]}.$$

For any PPT adversary $\mathcal{A}$*,* $\exists$ *a PPT adversary* $\mathcal{B}_1$ *for the* SXDH *assumption such that*

$$\begin{aligned}\mathsf{Adv}^{\mathsf{P1\text{-}SXDH}}_{\mathcal{A}}(\lambda) &= |\Pr[\mathcal{A}(\mathcal{D}, \mathcal{U}_0) \rightarrow 1] - \Pr[\mathcal{A}(\mathcal{D}, \mathcal{U}_1) \rightarrow 1]| \\ &\leq 4(n - m)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_1}(\lambda) + 2^{-\Omega(\lambda)}.\end{aligned}$$

We refer to [TT18, Section 4] for a detailed proof of Lemma 1.

Lemma 2. *[TT18] Let* $m = m(\lambda), n = n(\lambda)$ *be two integers. Problem 2-*SXDH $(\mathsf{P2\text{-}SXDH})$ *is to guess the bit* β*, given the following distributions:*

$$\mathsf{G} \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda), \mathsf{params}_V \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(7, \mathsf{G}), \mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p).$$

$$\mathbf{v}^*_i = (\rho'_i(-i, 1), 1, 0, 0, 0, 0)\mathbf{B}^* \quad \forall i \in [m + 1, n] \textit{ with } \{\rho'_i\}_{i\in[m+1,n]} \leftarrow \mathbb{Z}_p.$$

$$\mathcal{D} = (G, \mathsf{params}_V, \llbracket \mathbf{b}_1 \rrbracket_1, \llbracket \mathbf{b}_2 \rrbracket_1, \ldots, \llbracket \mathbf{b}_4 \rrbracket_1, \llbracket \mathbf{b}^*_1 \rrbracket_2, \llbracket \mathbf{b}^*_2 \rrbracket_2, \llbracket \mathbf{b}_4 \rrbracket^*_2, \llbracket \mathbf{b}^*_5 \rrbracket_2, \{\llbracket \mathbf{v}^*_i \rrbracket_2\}_{i\in[m+1,n]}).$$

$$\textit{Choose} \quad \{\pi'_i, \xi_i, \rho'_i\}_{i\in[m]} \leftarrow \mathbb{Z}_p.$$

$$\mathbf{u}_{i,\beta} = (\pi'_i(1, i), \beta\xi_i, 0, 1, 0, 0)\mathbf{B} \quad \forall i \in [m].$$

$$\mathbf{u}^*_{i,\beta} = (\rho'_i(-i, 1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* \quad \forall i \in [m].$$

$$\mathcal{U} = \{\llbracket \mathbf{u}_{i,\beta} \rrbracket_1, \llbracket \mathbf{u}^*_{i,\beta} \rrbracket_2\}_{i\in[m]}.$$

For any PPT adversary $\mathcal{A}$, $\exists$ a PPT adversary $\mathcal{B}_3$ for the SXDH *assumption such that*

$$\mathsf{Adv}^{\mathsf{P2\text{-}SXDH}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{D},\mathcal{U}_0) \to 1] - \Pr[\mathcal{A}(\mathcal{D},\mathcal{U}_1) \to 1]| \leq 8m\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_3}(\lambda) + 2^{-\Omega(\lambda)}.$$

We refer to [TT18, Section 4] for a detailed proof of Lemma 2.

Lemma 3. *Let $m_1 = m_1(\lambda), m_2 = m_2(\lambda), n = n(\lambda)$ be three integers. Problem 3-*SXDH ($\mathsf{P3\text{-}SXDH}$) *is to guess the bit β, given the following distributions:*

$$\begin{aligned}
&\mathsf{G} \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^{\lambda}), \mathsf{params}_V \leftarrow \mathcal{G}_{\mathsf{DPVS.Gen}}(7,\mathsf{G}), \mathbf{B}, \widetilde{\mathbf{B}} \leftarrow GL_7(\mathbb{Z}_p).\\
&\mathcal{D} = (\mathsf{G}, \mathsf{params}_V, \mathcal{D}_{\mathbf{B}}, \mathcal{D}_{\widetilde{\mathbf{B}}})\\
&\textit{where } \mathcal{D}_{\mathbf{B}} = (\llbracket \mathbf{b}_1 \rrbracket_1, \llbracket \mathbf{b}_2 \rrbracket_1, \ldots, \llbracket \mathbf{b}_4 \rrbracket_1, \llbracket \mathbf{b}^*_1 \rrbracket_2, \llbracket \mathbf{b}^*_2 \rrbracket_2, \llbracket \mathbf{b}_4 \rrbracket^*_2, \llbracket \mathbf{b}^*_5 \rrbracket_2, \{\llbracket \mathbf{v}^*_i \rrbracket_2\}_{i\in[m_1+1,n]}).\\
&\mathbf{v}^*_i = (\rho'_i(-i,1),1,0,0,0,0)\mathbf{B}^* \quad \forall i \in [m_1+1,n] \textit{ with } \{\rho'_i\}_{i\in[m_1+1,n]} \leftarrow \mathbb{Z}_p\\
&\textit{and } \mathcal{D}_{\widetilde{\mathbf{B}}} = (\llbracket \widetilde{\mathbf{b}}_1 \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_2 \rrbracket_1, \ldots, \llbracket \widetilde{\mathbf{b}}_4 \rrbracket_1, \llbracket \widetilde{\mathbf{b}}^*_1 \rrbracket_2, \llbracket \widetilde{\mathbf{b}}^*_2 \rrbracket_2, \llbracket \widetilde{\mathbf{b}}^*_4 \rrbracket_2, \llbracket \widetilde{\mathbf{b}}^*_5 \rrbracket_2, \{\llbracket \widetilde{\mathbf{v}}^*_i \rrbracket_2\}_{i\in[m_2+1,n]}).\\
&\widetilde{\mathbf{v}}^*_j = (\widetilde{\rho}'_j(-j,1),1,0,0,0,0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2+1,n] \textit{ with } \{\widetilde{\rho}'_j\}_{j\in[m_2+1,n]} \leftarrow \mathbb{Z}_p.\\
&\mathbf{u}_{i,\beta} = (\pi'_i(1,i),\beta\xi_i,0,1,0,0)\mathbf{B} \quad \forall i \in [m_1] \textit{ with } \{\pi'_i,\xi_i\}_{i\in[m_1]} \leftarrow \mathbb{Z}_p.\\
&\widetilde{\mathbf{u}}_{j,\beta} = (\widetilde{\pi}'_j(1,j),\beta\widetilde{\xi}_j,0,1,0,0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2] \textit{ with } \{\widetilde{\pi}'_j,\widetilde{\xi}_j\}_{j\in[m_2]} \leftarrow \mathbb{Z}_p.\\
&\mathbf{u}^*_{i,\beta} = (\rho'_i(-i,1),1,0,-\beta\xi_i,0,0)\mathbf{B}^* \quad \forall i \in [m_1] \textit{ with } \{\xi_i,\rho'_i\}_{i\in[m_1]} \leftarrow \mathbb{Z}_p.\\
&\widetilde{\mathbf{u}}^*_{j,\beta} = (\widetilde{\rho}'_j(-j,1),1,0,-\beta\widetilde{\xi}_j,0,0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m^*_2] \textit{ with } \{\widetilde{\xi}_j,\widetilde{\rho}'_j\}_{j\in[m_2]} \leftarrow \mathbb{Z}_p.\\
&\mathcal{U}_\beta = \{\llbracket \mathbf{u}_{i,\beta} \rrbracket_1, \llbracket \mathbf{u}^*_{i,\beta} \rrbracket_2\}_{i\in[m_1]}.\\
&\mathcal{V}_\beta = \{\llbracket \widetilde{\mathbf{u}}_{j,\beta} \rrbracket_1, \llbracket \widetilde{\mathbf{u}}^*_{j,\beta} \rrbracket_2\}_{j\in[m_2]}.\\
&\mathcal{W}_\beta = \{\mathcal{U}_\beta, \mathcal{V}_\beta\}.
\end{aligned}$$

For any PPT adversary $\mathcal{A}$, $\exists$ a PPT adversary $\mathcal{B}_4$ for the SXDH *assumption such that*

$$\mathsf{Adv}^{\mathsf{P3\text{-}SXDH}}_{\mathcal{A}}(\lambda) = |\Pr[\mathcal{A}(\mathcal{D},\mathcal{W}_0) \to 1] - \Pr[\mathcal{A}(\mathcal{D},\mathcal{W}_1) \to 1]| \leq 8(m_1+m_2)\mathsf{Adv}^{\mathsf{SXDH}}_{\mathcal{B}_3}(\lambda) + 2^{-\Omega(\lambda)}.$$

Proof of Lemma 3. Let us consider the following Games to prove Lemma 3. For each game transition, we show that the difference of probabilities that $\mathcal{A}$ outputs 1 in both games is negligible.

Game 0: This game is the same as for the case $\beta = 0$ i.e., $\mathcal{A}$ is given an instance $(\mathcal{D}, \mathcal{W}_0)$.

$$\begin{aligned}
&\mathbf{v}^*_i = (\rho'_i(-i,1),1,0,0,0,0)\mathbf{B}^* \quad \forall i \in [m_1+1,n]\\
&\mathbf{u}_{i,0} = (\pi'_i(1,i),0,0,1,0,0)\mathbf{B} \quad \forall i \in [m_1]\\
&\mathbf{u}^*_{i,0} = (\rho'_i(-i,1),1,0,0,0,0)\mathbf{B}^* \quad \forall i \in [m_1]\\
&\widetilde{\mathbf{v}}^*_j = (\widetilde{\rho}'_j(-j,1),1,0,0,0,0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2+1,n]\\
&\widetilde{\mathbf{u}}_{j,0} = (\widetilde{\pi}'_j(1,j),0,0,1,0,0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2]\\
&\widetilde{\mathbf{u}}^*_{j,0} = (\widetilde{\rho}'_j(-j,1),1,0,0,0,0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2]
\end{aligned}$$

where $\pi'_i, \xi_i \leftarrow \mathbb{Z}_p$ for all $i \in [m_1]$; $\widetilde{\pi}'_j, \widetilde{\xi}_j \leftarrow \mathbb{Z}_p$ for all $j \in [m_2]$ and $\rho'_i, \widetilde{\rho}'_j \leftarrow \mathbb{Z}_p$ for all $i, j \in [n]$.

Game 1: This game is the same as Game 0 except of the following changes:

$$\begin{aligned}
\mathbf{v}_i^* &= (\rho'_i(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* \quad \forall i \in [m_1+1, n] \\
\mathbf{u}_{i,1} &= (\pi'_i(1,i), \xi_i, 0, 1, 0, 0)\mathbf{B} \quad \forall i \in [m_1] \\
\mathbf{u}_{i,1}^* &= (\rho'_i(-i,1), 1, 0, -\xi_i, 0, 0)\mathbf{B}^* \quad \forall i \in [m_1]
\end{aligned}$$

where $\xi_i \leftarrow \mathbb{Z}_p$ for all $i \in [m_1]$.

Game 2: This game is the same as Game 1 except of the following changes:

$$\begin{aligned}
\widetilde{\mathbf{v}}_j^* &= (\widetilde{\rho}'_j(-j,1), 1, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2+1, n] \\
\widetilde{\mathbf{u}}_{j,1} &= (\widetilde{\pi}'_j(1,j), \widetilde{\xi}_j, 0, 1, 0, 0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2] \\
\widetilde{\mathbf{u}}'_{j,1} &= (\widetilde{\rho}'_j(-j,1), 1, 0, -\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2]
\end{aligned}$$

where $\widetilde{\xi}_j \leftarrow \mathbb{Z}_p$ for all $j \in [m_2]$. Observe that Game 2 is the same as the case of $\beta = 1$, i.e., $\mathcal{A}$ is given an instance $(\mathcal{D}, \mathcal{W}_1)$. In the following, we denote the event that $\mathcal{A}$ outputs 1 in Game ι by E'_ι.

Claim 1. $|\mathsf{Pr}(\mathsf{E}'_0) - \mathsf{Pr}(\mathsf{E}'_1)| \leq 8m_1 \cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. Let us consider a PPT adversary $\mathcal{A}$ against the P3-SXDH assumption. We use $\mathcal{A}$ as a subroutine to construct an adversary $\mathcal{B}$ against the underlying P2-SXDH scheme. In particular, we show that if $\mathcal{A}$ can break the P3-SXDH assumption, then there exists a PPT adversary $\mathcal{B}$ that can break the P2-SXDH assumption. The adversary $\mathcal{B}(1^\lambda)$ simulates $\mathcal{A}$ as follows.

Let $\mathcal{B}$ gets the challenge instances $(\mathsf{G}, \mathsf{params}_V, \mathcal{D}_{\mathbf{B}}, \mathcal{U}_\beta)$ from $\mathcal{A}$. Then, $\mathcal{B}$ chooses a matrix $\widetilde{\mathbf{B}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. Using the matrix $\widetilde{\mathbf{B}}$, $\mathcal{B}$ samples

$$\widetilde{\mathbf{v}}_i^* = (\widetilde{\rho}'_i(-i,1), 1, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall i \in [m_2+1, n]$$

where $\widetilde{\rho}'_j \leftarrow \mathbb{Z}_p$. Now $\mathcal{B}$ samples $\pi'_j \leftarrow \mathbb{Z}_p$ for $j \in [m_2]$ as

$$\begin{aligned}
\widetilde{\mathbf{u}}_{j,0} &= (\widetilde{\pi}'_j(1,j), 0, 0, 1, 0, 0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2], \\
\widetilde{\mathbf{u}}_{j,0}^* &= (\widetilde{\rho}'_j(-j,1), 1, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2].
\end{aligned}$$

Therefore, $\mathcal{B}$ generates the instances $(\mathcal{D}_{\widetilde{\mathbf{B}}}, \mathcal{V}_0 = \{[\![\widetilde{\mathbf{u}}_{j,0}]\!]_1, [\![\widetilde{\mathbf{u}}_{j,0}^*]\!]_2\}_{j\in[m_2]})$ using the basis $\widetilde{\mathbf{B}}$ where $\mathcal{D}_{\widetilde{\mathbf{B}}} = (\mathsf{G}, \mathsf{params}_V, \{[\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i=1}^4, \{[\![\widetilde{\mathbf{b}}_i^*]\!]_2\}_{i=1}^5, \{[\![\widetilde{\mathbf{v}}_j]\!]_2\}_{j=1}^{m_2})$. According to P2-SXDH, $\mathcal{B}$ can interpolate between Game 1 and Game 0 with the advantage $8m_1 \cdot \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{SXDH}}(\lambda)$. Therefore, $\mathcal{A}$'s view is the same as Game 0 for $\beta = 0$ and for $\beta = 1$ the adversarial view is identical with Game 1. □

Claim 2. $|\mathsf{Pr}(\mathsf{E}'_1) - \mathsf{Pr}(\mathsf{E}'_2)| \leq 8m_2 \cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. Let us consider a PPT adversary $\mathcal{A}$ against the P3-SXDH assumption. We use $\mathcal{A}$ as a subroutine to construct an adversary $\mathcal{B}$ against the underlying P2-SXDH scheme. In particular, we show that if $\mathcal{A}$ can break the P3-SXDH assumption, then there is a PPT adversary $\mathcal{B}$ which breaks the P2-SXDH assumption. The adversary $\mathcal{B}(1^\lambda)$ simulates $\mathcal{A}$ as follows.

Let $\mathcal{B}$ gets the challenge instances $(\mathsf{G}, \mathsf{params}_V, \mathcal{D}_{\widetilde{\mathbf{B}}}, \mathcal{V}_\beta)$ from $\mathcal{A}$. Then $\mathcal{B}$ chooses a matrix $\mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. Using the matrix $\mathbf{B}$, $\mathcal{B}$ samples

$$\mathbf{v}_i^* = (\rho_i'(-i, 1), 1, 0, 0, 0, 0)\, \mathbf{B}^* \quad \forall i \in [m_1 + 1, n]$$

where $\rho_i' \leftarrow \mathbb{Z}_p$. Now $\mathcal{B}$ samples $\pi_i', \xi_i \leftarrow \mathbb{Z}_p$ for $i \in [m_1]$ as

$$\begin{aligned} \mathbf{u}_{i,1} &= (\pi_i'(1, i), \xi_i, 0, 1, 0, 0)\mathbf{B} \quad \forall i \in [m_1], \\ \mathbf{u}_{i,1}^* &= (\rho_i'(-i, 1), 1, 0, \xi_i, 0, 0)\mathbf{B}^* \ \ \forall i \in [m_1]. \end{aligned}$$

Therefore, $\mathcal{B}$ generates the instances $(\mathcal{D}_{\mathbf{B}}, \mathcal{U}_1 = \{[\![\widetilde{\mathbf{u}}_{j,1}]\!]_1, [\![\widetilde{\mathbf{u}}_{j,1}^*]\!]_2\}_{j \in [m_1]})$ using the basis $\mathbf{B}$ where $\mathcal{D}_{\mathbf{B}} = (\mathsf{G}, \mathsf{params}_V, \{[\![\mathbf{b}_i]\!]_1\}_{i=1}^4, \{[\![\mathbf{b}_i^*]\!]_2\}_{i=1}^5, \{[\![\mathbf{v}_j]\!]_2\}_{j=1}^{m_1})$. According the P2-$\mathsf{SXDH}$, $\mathcal{B}$ can interpolate between Game 1 and Game 2 with the advantage $8m_2 \cdot \mathsf{Adv}_{\mathcal{B}'}^{\mathsf{SXDH}}(\lambda)$. Therefore, $\mathcal{A}$'s view is the same as Game 1 for $\beta = 0$ and for $\beta = 1$ the adversarial view is identical with Game 2. □

□

Proof of Theorem 1. Suppose $\mathcal{A}$ be a PPT adversary against the semi-adaptive full attribute-hiding indistinguishability ($\mathsf{SA\text{-}FAH\text{-}IND}$) security of our UZP-IPFE scheme. We construct an algorithm $\mathcal{B}$ for breaking the SXDH assumption that uses $\mathcal{A}$ as a subroutine. We prove Theorem 1 by a series of games. For each game transition, we calculate the difference of probabilities that $\mathcal{A}$ outputs 1 in the corresponding games. In every game, the challenger chooses a random element $m_1' \leftarrow [m_{1,\mathsf{max}}]$, as a guess of m_1^* at the beginning of the games. As we consider the semi-adaptive model here, we set $m_2^* = m_{2,\mathsf{max}}$. We represent E_ι as the event that $\mathcal{A}$ outputs 1 in Game ι.

Game 0: This game is the same as the real security game where the challenge ciphertext is the encryption of $\mathbf{x}^{(0)}$ as described in Def. 2 i.e., the challenge ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}^{(0)} = (\{[\![\boldsymbol{c}_i^1]\!]_1\}_{i \in [m_1^*]}, \{[\![\boldsymbol{c}_j^2]\!]_1\}_{j \in [m_2^*]})$ for a pair of vectors $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}), (\mathbf{w}^{(0)}, \mathbf{w}^{(1)})$ is replied as

$$\begin{aligned} [\![\mathbf{c}_i^1]\!]_1 &= [\![\Big(\pi_i(1, i), x_i^{(0)}, \alpha, 0, 0, 0\Big)\mathbf{B}]\!]_1 \ \ \forall i \in [m_1^*] \\ [\![\mathbf{c}_j^2]\!]_1 &= [\![\Big(\widetilde{\pi}_j(1, j), \delta w_j^{(0)}, \alpha, 0, 0, 0\Big)\widetilde{\mathbf{B}}]\!]_1 \ \ \forall j \in [m_2^*] \end{aligned}$$

with $\pi_i, \widetilde{\pi}_j \leftarrow \mathbb{Z}_p$ for all $i \in [m_1^*], j \in [m_2^*]$ and $\delta, \alpha \leftarrow \mathbb{Z}_p$. Here $\mathbf{B}, \widetilde{\mathbf{B}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$ and $\mathbf{b}_i, \widetilde{\mathbf{b}}_i$ are their i-th row respectively. The ℓ-th secret keys $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ for the vectors $\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}$ are replied as

$$\begin{aligned} \mathbf{k}_i^1 &= (\rho_i^{(\ell)}(-i, 1), y_i^{(\ell)}, \gamma_i^{(\ell)}, 0, 0, 0)\mathbf{B}^* \in \mathbb{Z}_p^7 \ \ \forall i \in I_{\mathbf{y}^{(\ell)}} \\ \mathbf{k}_j^2 &= (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, 0, 0, 0)\widetilde{\mathbf{B}}^* \in \mathbb{Z}_p^7 \ \ \forall j \in I_{\mathbf{v}^{(\ell)}} \end{aligned}$$

with $\rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \omega^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i\in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j\in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

Game 1: This game is similar to Game 0 except that $[\![\mathbf{c}_i^1]\!]_1$ in the challenge ciphertext set by $\mathcal{B}$ is $\mathsf{CT}^{(0)}_{\mathbf{x},\mathbf{w}} = (\{[\![\boldsymbol{c}_i^1]\!]_1\}_{i\in[m_1^*]}, \{[\![\boldsymbol{c}_j^2]\!]_1\}_{j\in[m_2^*]})$ where

$$[\![\mathbf{c}_i^1]\!]_1 = [\![\Big(\pi_i(1,i), x_i^{(0)}, \alpha, \boxed{\sigma}, 0, 0\Big)\mathbf{B}]\!]_1 \quad \forall i \in [m_1^*]$$
$$[\![\mathbf{c}_j^2]\!]_1 = [\![\Big(\widetilde{\pi}_j(1,j), \delta w_j^{(0)}, \alpha, \boxed{\sigma}, 0, 0\Big)\widetilde{\mathbf{B}}]\!]_1 \quad \forall j \in [m_2^*]$$

where $\sigma \leftarrow \mathbb{Z}_p$. Others variables π_i, α and $\widetilde{\pi}_j, \delta$ are generated similarly by $\mathcal{B}$ as in Game 0.

Game 2: For $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$, Game 2 is equivalent to Game 1 except that the reply to $\mathcal{B}$ for the ℓ-th secret key query for associated pair of vectors $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i\in I_{\mathbf{y}^{(\ell)}}}, \mathbf{v}^{(\ell)} = (v_j^{(\ell)})_{j\in I_{\mathbf{v}^{(\ell)}}}$ is $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ where

$$\mathbf{k}_i^1 = (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \boxed{s_i^{(\ell)}}, 0, 0)\mathbf{B}^* \quad \forall i \in I_{\mathbf{y}^{(\ell)}}$$
$$\mathbf{k}_j^2 = (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \boxed{t_j^{(\ell)}}, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in I_{\mathbf{v}^{(\ell)}}$$

with $s_i^{(\ell)}, t_j^{(\ell)} \leftarrow \mathbb{Z}_p$ and $\sum_{i\in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j\in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0$. All other variables $\omega^{(\ell)}, \rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}$ are generated exactly as in Game 1.

Game 3. Game 3 is identical to Game 2 except that the ℓ-th secret key component $[\![\mathbf{k}_j]\!]_2$ satisfying the condition $\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^* \wedge \mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*$ is $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ where

$$\mathbf{k}_j^2 = (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \boxed{\widehat{t}_j^{(\ell)}}, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in I_{\mathbf{v}^{(\ell)}}$$

with $\widehat{t}_j^{(\ell)} \leftarrow \mathbb{Z}_p$. Other components are generated similarly by $\mathcal{B}$ as in Game 2.

Game 4: This game is the same as Game 3 except that the challenger aborts the game immediately if $m_1' \neq m_1^*$, i.e., the vector length associated with the challenge ciphertext is not equal to the guess m_1'. The adversary $\mathcal{A}$ will output $\perp$ if the game aborts.

Game 5: Game 5 is equivalent to Game 4 except that the reply to $\mathcal{B}$ of the ℓ-th secret key query for the pair of vectors $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i\in I_{\mathbf{y}^{(\ell)}}}, \mathbf{v}^{(\ell)} = (v_j^{(\ell)})_{j\in I_{\mathbf{v}^{(\ell)}}}$ satisfying the condition $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1' \wedge \mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \leq m_1'$ is $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}}, \{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_{\mathbf{v}^{(\ell)}}})$ where

$$\mathbf{k}_i^1 = (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \boxed{\widehat{s}_i^{(\ell)}}, 0, 0)\mathbf{B}^* \quad \forall i \in I_{\mathbf{y}^{(\ell)}}$$

with $\widehat{s}_i^{(\ell)} \leftarrow \mathbb{Z}_p$ for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$. All other variables are similarly generated by $\mathcal{B}$ as in Game 4.

Game 6: This game is similar to Game 5 except that the challenge ciphertext $\mathsf{CT}^{(0)}_{\mathbf{x},\mathbf{w}} = (\{\llbracket \boldsymbol{c}^1_i \rrbracket_1\}_{i\in[m_1^*]},\ \{\llbracket \boldsymbol{c}^2_j \rrbracket_1\}_{j\in[m_2^*]})$ is generated as

$$\llbracket \mathbf{c}^1_i \rrbracket_1 = \llbracket \Big(\pi_i(1,i), \boxed{x^{(0)}_i + \xi_i\sigma}, \alpha, \sigma, 0, 0\Big)\mathbf{B} \rrbracket_1 \ \ \forall i \in [m_1^*]$$
$$\llbracket \mathbf{c}^2_j \rrbracket_1 = \llbracket \Big(\widetilde{\pi}_j(1,j), \boxed{\delta w^{(0)}_j + \widetilde{\xi}_j\sigma}, \alpha, \sigma, 0, 0\Big)\widetilde{\mathbf{B}} \rrbracket_1 \ \ \forall j \in [m_2^*]$$

where $\xi_i \leftarrow \mathbb{Z}_p$ for all $i \in [m_1']$ and the ℓ-th secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (\{\llbracket \mathbf{k}^1_i \rrbracket_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{\llbracket \mathbf{k}^2_j \rrbracket_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ corresponding to the pair of vectors $\mathbf{y}^{(\ell)} = (y^{(\ell)}_i)_{i\in I_{\mathbf{y}^{(\ell)}}}, \mathbf{v}^{(\ell)} = (v^{(\ell)}_j)_{j\in I_{\mathbf{v}^{(\ell)}}}$ for all $\ell \in [\mathsf{Q_{SK}}]$ such that $\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \leq m_1'$ and $\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*$ are generated as follows

$$\mathbf{k}^1_i = \begin{cases} (\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \boxed{s^{(\ell)}_i - \xi_i y^{(\ell)}_i}, 0, 0)\mathbf{B}^* & \text{if } \mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1' \\ (\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \boxed{\widehat{s}^{(\ell)}_i - \xi_i y^{(\ell)}_i}, 0, 0)\mathbf{B}^* & \text{if } \mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1' \end{cases}$$

$$\mathbf{k}^2_j = \begin{cases} (\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \boxed{t^{(\ell)}_j - \widetilde{\xi}_j \omega^{(\ell)} v^{(\ell)}_j}, 0, 0)\widetilde{\mathbf{B}}^* & \text{if } \mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^* \\ (\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \boxed{\widehat{t}^{(\ell)}_j - \widetilde{\xi}_j \omega^{(\ell)} v^{(\ell)}_j}, 0, 0)\widetilde{\mathbf{B}}^* & \text{if } \mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^* \end{cases}$$

where $\xi_i, \widetilde{\xi}_j, s^{(\ell)}_i, t^{(\ell)}_j, \widehat{s}^{(\ell)}_i, \widehat{t}^{(\ell)}_j \leftarrow \mathbb{Z}_p$ such that $\sum_{i\in I_{\mathbf{y}^{(\ell)}}} s^{(\ell)}_i + \sum_{i\in I_{\mathbf{v}^{(\ell)}}} t^{(\ell)}_j = 0$. All other random values are similarly generated as Game 5.

Game 7: Game 7 is the same as Game 6 except that the ℓ-th secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (\{\llbracket \mathbf{k}^1_i \rrbracket_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{\llbracket \mathbf{k}^2_j \rrbracket_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ corresponding to the vectors $\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}$ are generated as follows. If $\langle \mathbf{w}^{(0)}, \mathbf{v}^{(\ell)} \rangle \neq 0, \langle \mathbf{w}^{(1)}, \mathbf{v}^{(\ell)} \rangle \neq 0$, with $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1'$, $\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*$,

$$\mathbf{k}^2_j = \Big(\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \boxed{\widetilde{\mathfrak{r}}^{(\ell)}_j}, 0, 0\Big)\, \widetilde{\mathbf{B}}^*$$

where $\widetilde{\mathfrak{r}}^{(\ell)}_j \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}^{(\ell)}}$.

Game 8: Game 8 is exactly identical to Game 7 except that $\llbracket \mathbf{c}^1_i \rrbracket_1, \llbracket \mathbf{c}^2_j \rrbracket_1$ in the challenge ciphertext are generated as follows

$$\llbracket \mathbf{c}^1_i \rrbracket_1 = \llbracket \Big(\pi_i(1,i), \boxed{x^{(1)}_i + \xi_i\sigma}, \alpha, \sigma, 0, 0\Big)\mathbf{B} \rrbracket_1 \ \ \forall i \in [m_1^*]$$
$$\llbracket \mathbf{c}^2_j \rrbracket_1 = \llbracket \Big(\widetilde{\pi}_j(1,j), \boxed{\delta w^{(1)}_j + \widetilde{\xi}_j\sigma}, \alpha, \sigma, 0, 0\Big)\widetilde{\mathbf{B}} \rrbracket_1 \ \ \forall j \in [m_2^*].$$

The remaining values are generated identically by $\mathcal{B}$ as in Game 8.

Game 9: Game 9 is the same as Game 8 except that the challenge ciphertext components $[\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{c}_j^2]\!]_1$ and $\forall \ell \in [\mathsf{Q}_{\mathsf{SK}}]$, the ℓ-th secret key components $\mathbf{k}_i^1, \mathbf{k}_j^2$ are set as

$$\begin{aligned}
[\![\mathbf{c}_i^1]\!]_1 &= [\![\Big(\pi_i(1,i), \boxed{x_i^{(1)}}, \alpha, \sigma, 0, 0\Big)\mathbf{B}]\!]_1 \ \ \forall i \in [m_1^*] \\
[\![\mathbf{c}_j^2]\!]_1 &= [\![\Big(\widetilde{\pi}_j(1,j), \boxed{\delta w_j^{(1)}}, \alpha, \sigma, 0, 0\Big)\widetilde{\mathbf{B}}]\!]_1 \ \ \forall j \in [m_2^*] \\
\mathbf{k}_i^1 &= (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \boxed{s_i^{(\ell)}}, 0, 0)\mathbf{B}^* \ \ \forall i \in I_{\mathbf{y}^{(\ell)}} \\
\mathbf{k}_j^2 &= \Big(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \boxed{t_j^{(\ell)}}, 0, 0\Big)\,\widetilde{\mathbf{B}}^* \quad \forall j \in I_{\mathbf{v}^{(\ell)}}
\end{aligned}$$

where $\sigma, s_i^{(\ell)}, t_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{y}^{(\ell)}}} t_j^{(\ell)} = 0$. All other variables are similarly generated by $\mathcal{B}$ as in Game 8.

Game 10: This game is similar to Game 9 except that the abort condition defined in Game 4 is removed.

Game 11: Game 11 is similar to Game 10 except that the ℓ-th secret key components $\mathbf{k}_i^1$ and $\mathbf{k}_j^2$ of $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ are generated as follows:

$$\begin{aligned}
\mathbf{k}_i^1 &= (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \boxed{0}, 0, 0)\mathbf{B}^* \ \ \forall i \in I_{\mathbf{y}^{(\ell)}} \\
\mathbf{k}_j^2 &= (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \boxed{0}, 0, 0)\widetilde{\mathbf{B}}^* \ \ \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

All random values $\pi_i, \widetilde{\pi}_j, \rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \alpha, \delta, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}$ are chosen from $\mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

Game 12. Game 12 is identical to Game 11 except that the challenge ciphertext components $[\![\mathbf{c}_i^1]\!]_1$, and $[\![\mathbf{c}_j^2]\!]_1$ of $\mathsf{CT}_{\mathbf{x},\mathbf{w}}^{(1)} = (\{[\![\boldsymbol{c}_i^1]\!]_1\}_{i \in [m_1^*]},\ \{[\![\boldsymbol{c}_j^2]\!]_1\}_{j \in [m_2^*]})$ are generated as follows:

$$\begin{aligned}
[\![\mathbf{c}_i^1]\!]_1 &= [\![\Big(\pi_i(1,i), x_i^{(1)}, \alpha, \boxed{0}, 0, 0\Big)\mathbf{B}]\!]_1 \ \ \forall i \in [m_1^*] \\
[\![\mathbf{c}_j^2]\!]_1 &= [\![\Big(\widetilde{\pi}_j(1,j), \delta w_j^{(1)}, \alpha, \boxed{0}, 0, 0\Big)\widetilde{\mathbf{B}}]\!]_1 \ \ \forall j \in [m_2^*].
\end{aligned}$$

We now prove the indistinguishability of the above games by the following claims. Combining the following claims, the above Theorem follows.

Claim 3. $|\Pr(\mathsf{E}_1) - \Pr(\mathsf{E}_0)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. We will show that the challenger $\mathcal{B}$ can solve the SXDH assumption using $\mathcal{A}$ as a subroutine. Let $\mathcal{B}$ obtain an instance $(\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T,\ g_1, g_2, e), [\![a]\!]_1 = g_1^a, [\![u]\!]_1 = g_1^u, [\![t_\beta]\!]_1 = [\![au + \beta f]\!]_1 = g_1^{au+\beta f})$ of SXDH assumption for $\iota = 1$ where $a, u, f \leftarrow \mathbb{Z}_p, \beta \leftarrow$

$\{0,1\}$ and sets $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$ as in Game 0. Now, $\mathcal{B}$ uses the SXDH instances to interpolate between Game 0 and Game 1. The algorithm $\mathcal{B}$ implicitly defines random orthonormal dual $(\mathbf{B}, \mathbf{B}^*)$ by choosing $\mathbf{D} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$ and setting

$$\mathbf{B} = \begin{bmatrix} I_3 & & & \\ & 1 & -a & \\ & 0 & 1 & \\ & & & I_2 \end{bmatrix} \mathbf{D}, \mathbf{B}^* = \begin{bmatrix} I_3 & & & \\ & 1 & 0 & \\ & a & 1 & \\ & & & I_2 \end{bmatrix} \mathbf{D}^*; \quad \widetilde{\mathbf{B}} = \begin{bmatrix} I_3 & & & \\ & 1 & -a & \\ & 0 & 1 & \\ & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}, \widetilde{\mathbf{B}}^* = \begin{bmatrix} I_3 & & & \\ & 1 & 0 & \\ & a & 1 & \\ & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}^*$$

where $\mathbf{D}^* = (\mathbf{D}^{-1})^\top$ and a is implicitly set from the SXDH instance. Note that, $[\![a]\!]_1 = g_1^a$ and the algorithm $\mathcal{B}$ can compute $[\![\mathbf{B}]\!]_1$ using the given SXDH instances. Now, $\mathcal{B}$ simulates the ℓ-th secret key queries for the key vector $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i \in I_{\mathbf{y}^{(\ell)}}}$ along with the predicate vector $\mathbf{v}^{(\ell)} = (v_i^{(\ell)})_{i \in I_{\mathbf{v}^{(\ell)}}}$ by responding with $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ where

$$\begin{aligned} [\![\mathbf{k}_i^1]\!]_2 &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, 0, 0, 0)\mathbf{B}^*]\!]_2 \;\; \forall i \in I_{\mathbf{y}^{(\ell)}} \\ [\![\mathbf{k}_j^2]\!]_2 &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, 0, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \; \forall j \in I_{\mathbf{v}^{(\ell)}} \end{aligned}$$

with $\rho_i^{(\ell)}, \omega^{(\ell)}, \gamma_i^{(\ell)}, \tilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{i \in I_{\mathbf{y}^{(\ell)}}} \tilde{\gamma}_j^{(\ell)} = 0$ and $[\![\mathbf{k}_j^2]\!]_2$ is generated similarly as in Game 0. Now for the challenge ciphertext, $\mathsf{CT}_{\mathbf{x},\mathbf{w}}^{(0)} = (\{[\![\mathbf{c}_i^1]\!]_1\}_{i \in [m_1^*]}, \{[\![\mathbf{c}_j^2]\!]_1\}_{j \in [m_2^*]})$, $\mathcal{B}$ sets $[\![\mathbf{c}_i^1]\!]_1, [\![\mathbf{c}_j^1]\!]_1$ for $i \in [m_1^*], j \in [m_2^*]$ as

$$\begin{aligned} [\![\mathbf{c}_i^1]\!]_1 &= [\![(\pi_i(1,i), x_i^{(0)}, \alpha', 0, 0, 0)\mathbf{B} + (0, 0, 0, -u, t_\beta, 0, 0)\mathbf{D}]\!]_1 \\ &= [\![(\pi_i(1,i), x_i^{(0)}, \alpha' - u, \beta f, 0, 0)\mathbf{B}]\!]_1 \;\; \forall i \in [m_1^*], \\ \text{and } [\![\mathbf{c}_j^2]\!]_1 &= [\![(\tilde{\pi}_j(1,j), \delta w_j^{(0)}, \alpha', 0, 0, 0)\widetilde{\mathbf{B}} + (0, 0, 0, -u, t_\beta, 0, 0)\widetilde{\mathbf{D}}]\!]_1 \\ &= [\![(\tilde{\pi}_j(1,j), \delta w_j^{(0)}, \alpha' - u, \beta f, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \;\; \forall j \in [m_2^*] \end{aligned}$$

where $\mathbf{x}^{(0)} = (x_i^{(0)})_{i \in [m_1^*]}$ and $\alpha', \zeta \leftarrow \mathbb{Z}_p$. Here the knowledge of $\{[\![\mathbf{b}_i]\!]_1\}_{i \in \{1,2,\ldots,4\}}$ are sufficient to compute $[\![\mathbf{c}_i^1]\!]_1$ and $[\![\mathbf{c}_j^2]\!]_1$. As $\mathcal{B}$ has no information about $[\![a]\!]_2$, $\mathcal{B}$ cannot compute $[\![\mathbf{b}_5^*]\!]_2$ as $\mathbf{b}_5^*$ contains the unknown a. However, the above simulation does not require any knowledge of $[\![a]\!]_2 = g_2^a$ as the 5-th, 6-th and 7-th components of $\mathbf{k}_i^1$ is set as 0 in both Game 0 and Game 1. Then the secret key simulated by $\mathcal{B}$ has the same distribution as in Game 0 and Game 1. Let us implicitly set $\alpha = \alpha' - u$. Then $\mathcal{A}$'s view simulated by $\mathcal{B}$ is the same as in Game 0 if $\beta = 0$ since the sixth component of $[\![\mathbf{c}_i^1]\!]_1$ is 0 and the challenge ciphertext has the same distribution as in Game 0. On the other hand, $\mathcal{A}$'s view simulated by $\mathcal{B}$ is identical as in Game 1 if $\beta = 1$ since the sixth components of $\mathbf{c}_i$ is $-\beta f = \sigma$ unless $f = 0$ and the distribution of the challenge ciphertext in Game 0 is identical with the distribution in Game 1. So, $\mathcal{B}$ interpolates between Game 0 and Game 1. Thus the claim follows. □

Claim 4. $|\Pr(\mathsf{E}_2) - \Pr(\mathsf{E}_1)| \le \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. Let $\mathcal{B}$ obtain an instance of $(\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), \llbracket a \rrbracket_2 = g_2^a, \llbracket u \rrbracket_2 = g_2^u, \llbracket t_\beta \rrbracket_2 = \llbracket au + \beta f \rrbracket_2 = g_2^{au+\beta f})$ of the SXDH problem for $\iota = 2$ where $a, u, f \leftarrow \mathbb{Z}_p$, $\beta \leftarrow \{0, 1\}$ and sets $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$. We will show that $\mathcal{B}$ can utilize the instances of the SXDH assumption to interpolate between Game 1 and Game 2 using $\mathcal{A}$ as a subroutine. The algorithm $\mathcal{B}$ implicitly defines two orthonormal dual bases $(\mathbf{B}, \mathbf{B}^*)$ and $(\widetilde{\mathbf{B}}, \widetilde{\mathbf{B}}^*)$ by choosing $\mathbf{D}, \widetilde{\mathbf{D}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$ and setting

$$\mathbf{B} = \begin{bmatrix} I_3 & & & \\ & 0 & 1 & \\ & -1 & -a & \\ & & & I_2 \end{bmatrix} \mathbf{D}, \mathbf{B}^* = \begin{bmatrix} I_3 & & & \\ & -a & 1 & \\ & -1 & 0 & \\ & & & I_2 \end{bmatrix} \mathbf{D}^*; \quad \widetilde{\mathbf{B}} = \begin{bmatrix} I_3 & & & \\ & 0 & 1 & \\ & -1 & -a & \\ & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}, \widetilde{\mathbf{B}}^* = \begin{bmatrix} I_3 & & & \\ & -a & 1 & \\ & -1 & 0 & \\ & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}^*$$

where $\mathbf{D}^* = (\mathbf{D}^{-1})^\top$ and $\widetilde{\mathbf{D}}^* = (\widetilde{\mathbf{D}}^{-1})^\top$ and a is implicitly set from the SXDH instance. The algorithm $\mathcal{B}$ simulates the ℓ-th secret key query for the vector $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i \in I_{\mathbf{y}^{(\ell)}}}$ and the predicate $\mathbf{v}^{(\ell)} = (v_j^{(\ell)})_{j \in I_{\mathbf{v}^{(\ell)}}}$ by responding with the secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{\llbracket \mathbf{k}_i^1 \rrbracket_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{\llbracket \mathbf{k}_j^2 \rrbracket_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ where

$$\begin{aligned} \llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket (\rho_i^{(\ell)}(-i, 1), y_i^{(\ell)}, \gamma_i^{(\ell)}, 0, 0, 0)\mathbf{B}^* + s_i^{(\ell)}(0, 0, 0, t_\beta, -u, 0, 0)\mathbf{D}^* \rrbracket_2 \\ &= \llbracket (\rho_i^{(\ell)}(-i, 1), y_i^{(\ell)}, \gamma_i^{(\ell)} - us_i^{(\ell)}, -\beta f s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \ \ \forall i \in I_{\mathbf{y}^{(\ell)}} \\ \llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, 0, 0, 0)\widetilde{\mathbf{B}}^* + t_j^{(\ell)}(0, 0, 0, t_\beta, -u, 0, 0)\widetilde{\mathbf{D}}^* \rrbracket_2 \\ &= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} - ut_j^{(\ell)}, -\beta f t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \ \ \forall j \in I_{\mathbf{v}^{(\ell)}} \end{aligned}$$

with $\omega^{(\ell)}, \rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, s_i^{(\ell)}, t_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0, \sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$. Now the challenge ciphertext $\mathsf{CT}_{\mathbf{x}, \mathbf{w}}^{(0)} = (\{\llbracket \mathbf{c}_i^1 \rrbracket_1\}_{i \in [m_1^*]}, \{\llbracket \mathbf{c}_j^2 \rrbracket_1\}_{j \in [m_2^*]})$ is generated by $\mathcal{B}$ by setting

$$\begin{aligned} \llbracket \mathbf{c}_i^1 \rrbracket_1 &= \llbracket (\pi_i(1, i), x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B} + (0, 0, 0, \sigma, 0, 0, 0)\mathbf{D} \rrbracket_1 \\ &= \llbracket (\pi_i(1, i), x_i^{(0)}, \alpha - a\sigma, -\sigma, 0, 0)\mathbf{B} \rrbracket_1 \quad \forall i \in [m_1^*] \\ \llbracket \mathbf{c}_j^2 \rrbracket_1 &= \llbracket (\widetilde{\pi}_j(1, j), \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}} + (0, 0, 0, \sigma, 0, 0, 0)\widetilde{\mathbf{D}} \rrbracket_1 \\ &= \llbracket (\widetilde{\pi}_j(1, j), \delta w_j^{(0)}, \alpha - a\sigma, -\sigma, 0, 0)\widetilde{\mathbf{B}} \rrbracket_1 \quad \forall j \in [m_2^*] \end{aligned}$$

where $\pi_i, \widetilde{\pi}_j, \delta, \alpha \leftarrow \mathbb{Z}_p$. Note that $\{\llbracket \mathbf{b}_i \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_i \rrbracket_1\}_{i \in \{1, 2, \ldots, 4\}}$ are sufficient to compute $\llbracket (\pi_i(1, i), x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B} \rrbracket_1$ and $\llbracket (\widetilde{\pi}_j(1, j), \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}} \rrbracket_1$ respectively. Without knowledge of $\llbracket a \rrbracket_1$ here $\mathcal{B}$ cannot compute $\llbracket \mathbf{b}_5 \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_5 \rrbracket_1$ as the rows $\mathbf{b}_5, \widetilde{\mathbf{b}}_5$ consist of the element a and $\mathcal{B}$ has no information about $\llbracket a \rrbracket_1$. Let us implicitly set $\alpha' = \alpha - a\sigma$. Then $\mathcal{A}$'s view simulated by $\mathcal{B}$ is the same as in Game 1 if $\beta = 0$ since the fifth component of $\llbracket \mathbf{k}_i^1 \rrbracket_2, \llbracket \mathbf{k}_j^2 \rrbracket_2$ are zero, so the secret keys have the same distribution as in Game 1. On the other hand, $\mathcal{A}$'s view simulated by $\mathcal{B}$ is identical to that in Game 2 for $\beta = 1$ since $-\sum_{i \in I_{\mathbf{y}^{(\ell)}}} f s_i^{(\ell)} - \sum_{j \in I_{\mathbf{v}^{(\ell)}}} f t_j^{(\ell)} = 0$

and thus the distribution of secret keys in Game 2 is identical with the distribution of Game 1. Hence, $\mathcal{B}$ interpolates between Game 1 and Game 2 and the claim follows. □

Claim 5. $|\Pr(\mathsf{E}_3) - \Pr(\mathsf{E}_2)| \leq 4m_{2,\mathsf{max}} \cdot (t_{\mathsf{max}} - 1)\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. We can make a reduction algorithm $\mathcal{B}_1$ that distinguishes the instances $(\mathcal{D}, \mathcal{U}_\beta)$ where $\widetilde{\mathbf{B}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. We consider the following distributions of Lemma 1.

$$\begin{aligned}
&\mathcal{D} = (\mathsf{G}, \mathsf{params}_V, \llbracket \widetilde{\mathbf{b}}_1 \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_2 \rrbracket_1, \ldots, \llbracket \widetilde{\mathbf{b}}_4 \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_1^* \rrbracket_2, \llbracket \widetilde{\mathbf{b}}_2^* \rrbracket_2, \ldots, \llbracket \widetilde{\mathbf{b}}_5^* \rrbracket_2, \{\llbracket \widetilde{\mathbf{u}}_j \rrbracket\}_{j \in [m_2^*]}),\\
&\widetilde{\mathbf{u}}_j = (\widetilde{\pi}_j'(1, j), 0, 0, \theta', 0, 0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2^*], \theta', \{\widetilde{\pi}_j'\}_{j \in [m_2^*]} \leftarrow \mathbb{Z}_p,\\
&\widetilde{\mathbf{u}}_{i,\beta}^* = (\widetilde{\rho}_j'(-j, 1), 0, 0, \beta \bar{s}_j', 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2^* + 1, n], \{\widetilde{\rho}_j', \bar{s}_j'\}_{j \in [m_2^*+1,n]} \leftarrow \mathbb{Z}_p,\\
&\mathcal{U}_\beta = \{\llbracket \widetilde{\mathbf{u}}_{j,\beta}^* \rrbracket_2\}_{j \in [m_2^*+1,n]}.
\end{aligned}$$

The algorithm $\mathcal{B}_1$ obtains the instances of Lemma 1 where $n = t_{\mathsf{max}}$, $m = m_2^*$ and sets $\mathsf{MPK} = (\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E), \{\llbracket \mathbf{b}_i \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_i \rrbracket_1\}_{i \in \{1,2,\ldots,4\}})$ where $\mathbf{b}_i$ is the i-th row of uniformly chosen matrix $\mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. Recall that, t_{max} is the maximum index of input vector $\mathbf{v}^{(\ell)}$ for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ with which $\mathcal{A}$ queries to the key generation oracle. Then, the challenge ciphertext $\llbracket \mathbf{c}_j^2 \rrbracket_1$ is generated by $\mathcal{B}_1$ using the instances $(\mathcal{D}, \mathcal{U}_\beta)$ as below:

$$\begin{aligned}
\llbracket \mathbf{c}_j^2 \rrbracket_1 &= \llbracket \chi \cdot \widetilde{\mathbf{u}}_j + \delta w_j^{(0)} \cdot \widetilde{\mathbf{b}}_3 + \alpha \cdot \widetilde{\mathbf{b}}_4 \rrbracket_1 \quad \text{for } \eta, \delta, \chi, \alpha \leftarrow \mathbb{Z}_p\\
&= \llbracket \chi \widetilde{\pi}_j' \cdot \widetilde{\mathbf{b}}_1 + j\chi\widetilde{\pi}_j' \cdot \widetilde{\mathbf{b}}_2 + \delta w_j^{(0)} \cdot \widetilde{\mathbf{b}}_3 + \alpha \cdot \widetilde{\mathbf{b}}_4 + \chi\theta' \cdot \widetilde{\mathbf{b}}_5 \rrbracket_1\\
&= \llbracket (\chi\widetilde{\pi}_j'(1, j), \delta w_j^{(0)}, \alpha, \chi\theta', 0, 0)\widetilde{\mathbf{B}} \rrbracket_1 \quad \forall j \in [m_2^*]
\end{aligned}$$

and $\llbracket \mathbf{c}_i^1 \rrbracket_1$ is set by choosing a matrix $\mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$. Now for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$, $\mathcal{B}_1$ generates the secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{\llbracket \mathbf{k}_i^1 \rrbracket_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{\llbracket \mathbf{k}_j^2 \rrbracket_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ component $\llbracket \mathbf{k}_j^2 \rrbracket_2$ for two cases

Case 1: $\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^* \vee \mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \geq m_2^*$

$\rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, s_i{}^{(\ell)}, t_j{}^{(\ell)} \leftarrow \mathbb{Z}_p$ for all $i \in I_{\mathbf{y}^{(\ell)}}, j \in I_{\mathbf{v}^{(\ell)}}$ and $\omega^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0$, $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

$$\begin{aligned}
&\llbracket \mathbf{k}_i^1 \rrbracket_2 = \llbracket (\rho_i^{(\ell)}(-i, 1), y_j^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \;\; \forall j \in I_{\mathbf{y}^{(\ell)}}.\\
&\llbracket \mathbf{k}_j^2 \rrbracket_2 = \llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \;\; \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

By using $\llbracket \widetilde{\mathbf{b}}_1^* \rrbracket_2, \llbracket \widetilde{\mathbf{b}}_2^* \rrbracket_2, \llbracket \widetilde{\mathbf{b}}_3^* \rrbracket_2, \llbracket \widetilde{\mathbf{b}}_4^* \rrbracket_2, \llbracket \widetilde{\mathbf{b}}_5^* \rrbracket_2$ from the instances of Lemma 1, it is sufficient to compute $\llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2$ and $\llbracket \mathbf{k}_i^1 \rrbracket_2$ is set as in Game 2.

Case 2: $(\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*)$

$$\text{For } j \leq m_2^*, \quad \llbracket \mathbf{k}_j^2 \rrbracket_2 = \llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2$$

where $\widetilde{\rho}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ for $j \leq m_2^*$.

$$\begin{aligned}
\text{For } j > m_2^*, \quad [\![\mathbf{k}_j^2]\!]_2 &= [\![\widetilde{\mu}_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widehat{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\widetilde{\mu}_j^{(\ell)}(\widetilde{\rho}_j'(-j,1), 0, 0, \beta \bar{s}_j', 0, 0)\widetilde{\mathbf{B}}^* + (\widehat{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\mu}_j^{(\ell)} \widetilde{\rho_j}' + \widehat{\rho}_j^{(\ell)})(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} + \beta \widetilde{\mu}_j^{(\ell)} \bar{s}_j', 0, 0)]\!]_2
\end{aligned}$$

where $\widetilde{\mu}_j^{(\ell)}, \widehat{\rho}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ for $j > m_2^*$. We implicitly set $\widetilde{\rho}_j^{(\ell)} = \widetilde{\mu}_j^{(\ell)} \widetilde{\rho_j}' + \widehat{\rho}_j^{(\ell)}$ with $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0, \sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$. Here the key component $[\![\mathbf{k}_i^1]\!]_2$ is generated as previous Case 1. We define

$$\widehat{t}_j^{(\ell)} = \begin{cases} t_j^{(\ell)} & \text{if } j \leq m_2^* \\ t_j^{(\ell)} + \widetilde{\mu}_j^{(\ell)} \bar{s}_j' & \text{if } j > m_2^* \;\; (\textit{i.e.}, \text{ setting } \beta = 1) \end{cases}$$

unless $\bar{s}_j' = 0$. Since $\{t_j^{(\ell)}\}_{j \in [m_2^*]}$ and $\{\widetilde{\mu}_j^{(\ell)}\}_{j \in I_{\mathbf{v}^{(\ell)}}, j > m_2^*}$ both are independently chosen from $\mathbb{Z}_p$. So $\widehat{t}_j^{(\ell)}$'s are uniformly random for all $j \in I_{\mathbf{v}^{(\ell)}}$. Therefore, the adversary's view is the same as in Game 2 for $\beta = 0$, and if $\beta = 1$ it turns to Game 3. Now we have

$$\begin{aligned}
|\Pr(\mathsf{E}_3) - \Pr(\mathsf{E}_2)| &\leq \sum_{i \in [m_{2,\mathsf{max}}]} \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{P1\text{-}SXDH}}(\lambda) \\
&\leq 4 \cdot m_{2,\mathsf{max}} \cdot (t_{\mathsf{max}} - m_2^*) \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)} \\
&\leq 4 \cdot m_{2,\mathsf{max}} \cdot (t_{\mathsf{max}} - 1) \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}.
\end{aligned}$$

Thus the claim follows. □

Claim 6. $\Pr(\mathsf{E}_4) = \frac{1}{m_{1,\mathsf{max}}} \cdot \Pr(\mathsf{E}_3)$.

Proof. Let $m_{1,\mathsf{max}}, m_{2,\mathsf{max}}$ be the maximum length of the challenge vector and challenge attribute vector respectively. Note that Game 4 is similar to Game 3 except that $\mathcal{A}$'s output is $\perp$ if $m_1' \neq m_1^*$ where m_1' is the length guess of the challenge message vectors $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}$. We have

$$\begin{aligned}
\Pr(\mathsf{E}_4) &= \sum_{i \in [m_{1,\mathsf{max}}]} \Pr[m_1' = i] \cdot \Pr[m_1^* = i \wedge \mathsf{E}_3 | m_1' = i] \\
&= \frac{1}{m_{1,\mathsf{max}}} \cdot \sum_{i \in [m_{1,\mathsf{max}}]} \Pr[m_1^* = i \wedge \mathsf{E}_3 | m_1' = i] \\
&= \frac{1}{m_{1,max}} \cdot \sum_{i \in [m_{1,max}]} \frac{\Pr[m_1' = i \wedge m_1^* = i \wedge \mathsf{E}_3]}{\Pr[m_1' = i \wedge m_2^* = i]} \\
&= \frac{1}{m_{1,\mathsf{max}}} \cdot \Pr(\mathsf{E}_3).
\end{aligned}$$

Claim 7. $|\Pr(\mathsf{E}_5) - \Pr(\mathsf{E}_4)| \leq 4(s_{\mathsf{max}} - 1)\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. We use the following instances of Lemma 1 to prove this claim. We can make a reduction algorithm $\mathcal{B}_1$ that distinguishes the instances $(\mathcal{D}, \mathcal{U}_\beta)$ where $\mathbf{B} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$,

$$\begin{aligned}
\mathcal{D} &= (\mathsf{G}, \mathsf{params}_V, \llbracket \mathbf{b}_1 \rrbracket_1, \llbracket \mathbf{b}_2 \rrbracket_1, \ldots, \llbracket \mathbf{b}_4 \rrbracket_1, \llbracket \mathbf{b}_1^* \rrbracket_2, \llbracket \mathbf{b}_2^* \rrbracket_2, \ldots, \llbracket \mathbf{b}_5^* \rrbracket_2, \{\llbracket \mathbf{u}_j \rrbracket\}_{j \in [m]}) \\
\mathbf{u}_i &= (\pi_i'(1, i), 0, 0, \phi', 0, 0)\mathbf{B} \quad \forall i \in [m_1'] \text{ with } \phi', \{\pi_i'\}_{i \in [m_1']} \leftarrow \mathbb{Z}_p \\
\text{and } \mathbf{u}_{i,\beta}^* &= (\rho_i'(-i, 1), 0, 0, \beta s_i', 0, 0)\mathbf{B}^* \quad \forall i \in [m_1' + 1, n], \{\rho_i', s_i'\}_{i \in [m_1'+1, n]} \leftarrow \mathbb{Z}_p \\
\mathcal{U}_\beta &= \{\llbracket \mathbf{u}_{i,\beta}^* \rrbracket_2\}_{i \in [m_1'+1, n]}.
\end{aligned}$$

Using $\mathcal{A}$ as a subroutine, we construct the reduction algorithm $\mathcal{B}_1$ that interpolates between Game 4 and Game 5. Before proceeding further, $\mathcal{B}_1$ chooses $m_1' \leftarrow [m_{1,\mathsf{max}}]$ which is a guess of the length of the challenge vector $\mathbf{x}^{(0)} = (x_i^{(0)})_{i \in [m_1^*]}$. If the guess is incorrect i.e., $m_1' \neq m_1^*$, then the algorithm $\mathcal{B}_1$ will output 0. Otherwise $\mathcal{B}_1$ outputs $\mathcal{A}$'s outputs as it is. Now, $\mathcal{B}_1$ obtains an instance of Lemma 1 with $n = s_{\mathsf{max}}$, $m = m_1'$ and set $\mathsf{MPK} = (\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E), \{\llbracket \mathbf{b}_i \rrbracket_1, \llbracket \widetilde{\mathbf{b}}_i \rrbracket_1\}_{i \in \{1,2,\ldots,4\}})$ where $\mathbf{b}_i, \widetilde{\mathbf{b}}_i$ are i-th rows of uniformly chosen matrices $\mathbf{B}, \widetilde{\mathbf{B}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$ respectively. Recall that, s_{max} is the maximum index of input vector $\mathbf{y}^{(\ell)}$ for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ with which $\mathcal{A}$ queries to the key generation oracle. Now, the challenger $\mathcal{B}_1$ simulates the component $\llbracket \mathbf{c}_i^1 \rrbracket_1$ for all $i \in [m_1^*]$ of the challenge ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}^{(0)} = (\{\llbracket \mathbf{c}_i^1 \rrbracket_1\}_{i \in [m_1^*]}, \{\llbracket \mathbf{c}_j^2 \rrbracket_1\}_{j \in [m_2^*]})$ as

$$\begin{aligned}
\llbracket \mathbf{c}_i^1 \rrbracket_1 &= \llbracket \xi \cdot \mathbf{u}_i + x_i^{(0)} \cdot \mathbf{b}_3 + \alpha \cdot \mathbf{b}_4 \rrbracket_1 \quad \text{for } \xi, \alpha \leftarrow \mathbb{Z}_p \\
&= \llbracket \xi \pi_i' \cdot \mathbf{b}_1 + i \xi \pi_i' \cdot \mathbf{b}_2 + x_i^{(0)} \cdot \mathbf{b}_3 + \alpha \cdot \mathbf{b}_4 + \xi \phi' \cdot \mathbf{b}_5 \rrbracket_1 \\
&= \llbracket (\xi \pi_i'(1, i), x_i^{(0)}, \alpha, \xi \phi', 0, 0)\mathbf{B} \rrbracket_1
\end{aligned}$$

and $\llbracket \mathbf{c}_j^2 \rrbracket_1$ is set as in Game 4 using $\widetilde{\mathbf{B}}$. Observe that, we implicitly set $\pi_i = \xi \pi_i', \sigma = \xi \phi'$ unless $\xi = 0$. By using the instances of Lemma 1, $\mathcal{B}_1$ simulates the ℓ-th secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} = (\{\llbracket \mathbf{k}_i^1 \rrbracket_2\}_{i \in I_{\mathbf{y}^{(\ell)}}}, \{\llbracket \mathbf{k}_j^2 \rrbracket_2\}_{j \in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$ in two cases:

Case 1: $((\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1') \vee (\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \geq m_1'))$

$$\begin{aligned}
&\rho_i^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, {s_i}^{(\ell)}, {t_j}^{(\ell)} \leftarrow \mathbb{Z}_p \text{ for all } i \in I_{\mathbf{y}^{(\ell)}}, j \in I_{\mathbf{v}^{(\ell)}} \\
&\quad \text{such that } \sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0, \sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0. \\
&\llbracket \mathbf{k}_i^1 \rrbracket_2 = \llbracket (\rho_i^{(\ell)}(-i, 1), y_j^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \;\; \forall j \in I_{\mathbf{y}^{(\ell)}}. \\
&\llbracket \mathbf{k}_j^2 \rrbracket_2 = \llbracket (\widetilde{\rho}_j^{(\ell)}(-j, 1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \;\; \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

From the instances of $\mathsf{P1\text{-}SXDH}$ problem $\{\llbracket \mathbf{b}_i^* \rrbracket_2\}_{i \in 1,2,\ldots,5}$ are sufficient to compute $\llbracket (\rho_i^{(\ell)}(-i, 1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)},\ 0, 0)\mathbf{B}^* \rrbracket_2$. Other key components $\llbracket \mathbf{k}_j^2 \rrbracket_2$ are generated as in Game 4.

Case 2: $((\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \leq m_1') \wedge (\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1'))$

$$\text{If } i \leq m_1', \quad \llbracket \mathbf{k}_i^1 \rrbracket_2 = \llbracket (\rho_i^{(\ell)}(-i, 1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2$$

where $\rho_i^{(\ell)} \leftarrow \mathbb{Z}_p$ for $i \leq m_1'$.

$$\begin{aligned} \text{If } i > m_1', \quad [\![\mathbf{k}_i^1]\!]_2 &= [\![\mu_i^{(\ell)}\mathbf{u}^*_{i,\beta} + (\rho_i^{'(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\ &= [\![\mu_i^{(\ell)}(\rho_i'(-i,1), 0, 0, \beta s_i', 0, 0)\mathbf{B}^* + (\rho_i^{'(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\ &= [\![((\rho_i'\mu_i^{(\ell)} + \rho_i^{'(\ell)})(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} + \beta\mu_i^{(\ell)}s_i', 0, 0)\mathbf{B}^*]\!]_2 \end{aligned}$$

where $\mu_i^{(\ell)}, \rho_i^{'(\ell)} \leftarrow \mathbb{Z}_p, \rho_i^{(\ell)} = \rho_i'\mu_i^{(\ell)} + \rho_i^{'(\ell)}$ for $i > m_1'$ and the secret key component $[\![\mathbf{k}_j^2]\!]_2$ is generated similarly as Case 1 with the restriction that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0, \sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$. We can also set

$$\widehat{s}_i^{(\ell)} = \begin{cases} s_i^{(\ell)} & \text{if } i \leq m_1' \\ s_i^{(\ell)} + \mu_i^{(\ell)}s_i' & \text{if } i > m_1' \ \ (\textit{i.e.}, \text{ setting } \beta = 1) \end{cases}$$

unless $s_i' = 0$. Since the information of $\mu_i^{(\ell)}$ is hidden in $\rho_i^{(\ell)}$ using $\rho_i^{'(\ell)}$, both collections $\{s_i^{(\ell)}\}_{i \leq m_1'}$, $\{\mu_i^{(\ell)}s_i'\}_{i \in I_{\mathbf{y}^{(\ell)}}, i > m_1'}$ are randomly chosen from $\mathbb{Z}_p$. Therefore, $\{\widehat{s}_i^{(\ell)}\}_{i \in I_{\mathbf{y}^{(\ell)}}}$ are independently random elements in $\mathbb{Z}_p$. Therefore, $\mathcal{A}$'s view is the same as in Game 4 for $\beta = 0$ and in Game 5 if $\beta = 1$. Now we have

$$\begin{aligned} |\Pr(\mathsf{E}_5) - \Pr(\mathsf{E}_4)| &= \left| \sum_{i \in [m_{1,\mathsf{max}}]} \Pr(m_1' = i)\Pr(\mathsf{G}_5 | m_1' = i) - \Pr(m_1' = i)\Pr(\mathsf{G}_4 | m_1' = i) \right| \\ &= \frac{1}{m_{1,\mathsf{max}}} \left| \sum_{i \in [m_{1,\mathsf{max}}]} \Pr(\mathsf{G}_5 | m_1' = i) - \Pr(\mathsf{G}_4 | m_1' = i) \right| \\ &\leq \frac{1}{m_{1,\mathsf{max}}} \sum_{i \in [m_{1,\mathsf{max}}]} \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{P1\text{-}SXDH}}(\lambda) \\ &\leq \frac{1}{m_{1,\mathsf{max}}} \cdot \sum_{i \in [m_{1,\mathsf{max}}]} 4(s_{\mathsf{max}} - 1)\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)} \\ &= 4(s_{\mathsf{max}} - 1)\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}. \end{aligned}$$

Thus, the claim follows. □

Claim 8. $|\Pr(\mathsf{E}_6) - \Pr(\mathsf{E}_5)| \leq 8(m_{2,\mathsf{max}} + m_{1,\mathsf{max}})\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$

Proof. To prove the above claim, we construct a reduction algorithm $\mathcal{B}'$ that uses $\mathcal{A}$ as a subroutine to distinguish P3-SXDH instances. Let the reduction algorithm $\mathcal{B}'$ distinguish between the instances $(\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2), \mathcal{W}_\beta = (\mathcal{U}_\beta, \mathcal{V}_\beta))$.

$$\mathcal{D}_1 = (\mathsf{G}, \mathsf{params}_V, [\![\mathbf{b}_1]\!]_1, [\![\mathbf{b}_2]\!]_1, \ldots, [\![\mathbf{b}_4]\!]_1, [\![\mathbf{b}_1^*]\!]_2, [\![\mathbf{b}_2^*]\!]_2, [\![\mathbf{b}_4^*]\!]_2, [\![\mathbf{b}_5^*]\!]_2, \{[\![\mathbf{v}_i^*]\!]_2\}_{i\in[m_1'+1,n]})$$
$$\mathbf{v}_i^* = (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* \quad \forall i \in [m_1'+1, n] \text{ with } \{\rho_i'\}_{i\in[m_1'+1,n]} \leftarrow \mathbb{Z}_p$$
$$\mathbf{u}_{i,\beta} = (\pi_i'(1,i), \beta\xi_i, 0, 1, 0, 0)\mathbf{B} \quad \forall i \in [m_1'] \text{ with } \{\pi_i', \xi_i\}_{i\in[m_1']} \leftarrow \mathbb{Z}_p$$
$$\mathbf{u}_{i,\beta}^* = (\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* \ \ \forall i \in [m_1'] \text{ with } \{\xi_i, \rho_i'\}_{i\in[m_1']} \leftarrow \mathbb{Z}_p$$
$$\mathcal{U}_\beta = \{[\![\mathbf{u}_{i,\beta}]\!]_1, [\![\mathbf{u}_{i,\beta}^*]\!]_2\}_{i\in[m_1']}$$
$$\mathcal{D}_2 = (\mathsf{G}, \mathsf{params}_V, [\![\widetilde{\mathbf{b}}_1]\!]_1, [\![\widetilde{\mathbf{b}}_2]\!]_1, \ldots, [\![\widetilde{\mathbf{b}}_4]\!]_1, [\![\widetilde{\mathbf{b}}_1^*]\!]_1, , [\![\widetilde{\mathbf{b}}_2^*]\!]_1, , [\![\widetilde{\mathbf{b}}_4^*]\!]_1, [\![\widetilde{\mathbf{b}}_5^*]\!]_1, \{[\![\widetilde{\mathbf{v}}_j^*]\!]_2\}_{j\in[m_2^*+1,n]})$$
$$\widetilde{\mathbf{v}}_j^* = (\widetilde{\rho}_j'(-j,1), 1, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* \quad \forall j \in [m_2^*+1, n] \text{ with } \{\widetilde{\rho}_j'\}_{j\in[m_2^*+1,n]} \leftarrow \mathbb{Z}_p$$
$$\widetilde{\mathbf{u}}_{j,\beta} = (\widetilde{\pi}_j'(1,j), \beta\widetilde{\xi}_j, 0, 1, 0, 0)\widetilde{\mathbf{B}} \quad \forall j \in [m_2^*] \text{ with } \{\widetilde{\pi}_j', \widetilde{\xi}_j\}_{j\in[m_2^*]} \leftarrow \mathbb{Z}_p$$
$$\widetilde{\mathbf{u}}_{j,\beta}^* = (\widetilde{\rho}_j'(-j,1), 1, 0, -\beta\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* \ \ \forall j \in [m_2^*] \text{ with } \{\widetilde{\xi}_j, \widetilde{\rho}_j'\}_{j\in[m_2^*]} \leftarrow \mathbb{Z}_p$$
$$\mathcal{V}_\beta = \{[\![\widetilde{\mathbf{u}}_{j,\beta}]\!]_1, [\![\widetilde{\mathbf{u}}_{j,\beta}^*]\!]_2\}_{j\in[m_2^*]}$$

Before starting the game, $\mathcal{B}'$ first chooses $m_1' \leftarrow m_{1,\mathsf{max}}$ as a guess of the length m_1^* of the challenge vector $\mathbf{x}^{(0)} = (x_i^{(0)})_{i\in[m_1^*]}$. For the incorrect guess i.e., $m_1^* \neq m_1'$, $\mathcal{B}'$ outputs 0. Otherwise, $\mathcal{B}'$ outputs $\mathcal{A}$'s output as it is. On receiving the instance $(\mathcal{D}, \mathcal{W}_\beta)$ of P3-SXDH problem as described in Lemma 3 with $n = s_{\mathsf{max}}$, $m = m_1'$ and $n = t_{\mathsf{max}}$, $m = m_2^*$, the challenger $\mathcal{B}'$ sets $\mathsf{MPK} = (\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E), \{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i\in\{1,2,\ldots,4\}})$. Now, $\mathcal{B}'$ simulates the challenge ciphertext components $[\![\mathbf{c}_i^1]\!]_1$ and $[\![\mathbf{c}_j^2]\!]_1$ as follows:

$$\begin{aligned}
[\![\mathbf{c}_i^1]\!]_1 &= [\![\sigma\mathbf{u}_{i,\beta} + (0, 0, x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B}]\!]_1 \\
&= [\![\sigma(\pi_i'(1,i), \beta\xi_i, 0, 1, 0, 0)\mathbf{B} + (0, 0, x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B}]\!]_1 \\
&= [\![(\sigma\pi_i'(1,i), \beta\sigma\xi_i, 0, \sigma, 0, 0)\mathbf{B} + (0, 0, x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B}]\!]_1 \\
&= [\![(\sigma\pi_i'(1,i), \beta\sigma\xi_i + x_i^{(0)}, \alpha, \sigma, 0, 0)\mathbf{B}]\!]_1 \ \ \forall i \in [m_1']
\end{aligned}$$

$$\begin{aligned}
[\![\mathbf{c}_j^2]\!]_1 &= [\![\sigma\widetilde{\mathbf{u}}_{j,\beta} + (0, 0, \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \\
&= [\![\sigma(\widetilde{\pi}_j'(1,j), \beta\widetilde{\xi}_j, 0, 1, 0, 0)\widetilde{\mathbf{B}} + (0, 0, \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \\
&= [\![(\sigma\widetilde{\pi}_j'(1,j), \beta\sigma\widetilde{\xi}_j, 0, \sigma, 0, 0)\widetilde{\mathbf{B}} + (0, 0, \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \\
&= [\![(\sigma\widetilde{\pi}_j'(1,j), \beta\sigma\widetilde{\xi}_j + \delta w_j^{(0)}, \alpha, \sigma, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \ \ \forall j \in [m_2^*]
\end{aligned}$$

where $\sigma, \delta, \alpha \leftarrow \mathbb{Z}_p$. We set $\pi_i = \sigma\pi_i', \widetilde{\pi}_j = \sigma\widetilde{\pi}_j'$ unless $\sigma = 0$. Now for all $\ell \in [\mathsf{Q}_{\mathsf{SK}}]$, $\mathcal{B}'$ replies to $\mathcal{A}$, the ℓ-th secret key query $\mathsf{SK}_{\mathbf{y}^{(\ell)},\mathbf{v}^{(\ell)}} = (\{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ for the vector $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i\in I_{\mathbf{y}^{(\ell)}}}$ which is categorized into the following cases:
Case 1: $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1' \wedge \mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*$

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket y_i^{(\ell)} \mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_1 \\
&= \llbracket (y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket ((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* \rrbracket_2 \;\; \forall i \in I_{\mathbf{y}^{(\ell)}}.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket \omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket \omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi_j}, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_1 \\
&= \llbracket (\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi_j}, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket ((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)}\widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi_j}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \;\; \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

Here $\rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, s_i^{(\ell)}, t_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_i^{(\ell)} = 0$ with $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

Case 2: $((\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \leq m_1')) \wedge ((\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*) \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*))$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, \widehat{s}_i^{(\ell)}, \widehat{t}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\displaystyle\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For $i \leq m_1'$,

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket y_i^{(\ell)} \mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_1 \\
&= \llbracket (y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket ((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* \rrbracket_2.
\end{aligned}$$

For $i > m_1'$,

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket y_i^{(\ell)} \mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_1 \\
&= \llbracket (y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket ((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2.
\end{aligned}$$

For $j \leq m_2^*$,

$$\begin{aligned}
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket \omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket \omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi_j}, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_1 \\
&= \llbracket (\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi_j}, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket ((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)}\widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi_j}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2.
\end{aligned}$$

For $j > m_2^*$,

$$\begin{aligned}
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket \omega^{(\ell)} v_j^{(\ell)} \tilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket \omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_1 \\
&= \llbracket (\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket ((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2.
\end{aligned}$$

We implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)} \rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'$.

Case 3: $(\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) > m_2^*)$

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket y_i^{(\ell)} \mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_1 \\
&= \llbracket (y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket ((\rho_i''^{(\ell)} + y_i^{(\ell)} \rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket \omega^{(\ell)} v_j^{(\ell)} \tilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket \omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_1 \\
&= \llbracket (\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket ((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2.
\end{aligned}$$

Here $\omega^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, s_i^{(\ell)}, t_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0, \sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0$. Let $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)} \rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'$. Then $\rho_i^{(\ell)}$ and $\widetilde{\rho}_j^{(\ell)}$'s are uniformly random since $\rho_i''^{(\ell)}, \rho_i', \widetilde{\rho}_j''^{(\ell)}$ and $\widetilde{\rho}_j'$ all are uniformly random in $\mathbb{Z}_p$.

Case 4: $(\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1') \wedge ((\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*) \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*))$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, s_i^{(\ell)}, \widehat{t}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\displaystyle\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket y_i^{(\ell)} \mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_1 \\
&= \llbracket (y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket ((\rho_i''^{(\ell)} + y_i^{(\ell)} \rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* \rrbracket_2.
\end{aligned}$$

For $j \leq m_2^*$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi}_j, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

For $j > m_2^*$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'$.

Case 5: $((\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \leq m_1')) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*)$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, \widehat{s}_i^{(\ell)}, t_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\displaystyle\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For $i \leq m_1'$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)} \mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For $i > m_1'$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)} \mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi}_j, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}'$.

Case 6: $((\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) \le m_1')) \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) > m_2^*)$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, \widehat{s}_i^{(\ell)}, t_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For $i \le m_1'$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)}\mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)}(\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)}\rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For $i > m_1'$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)}\mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)}(\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)}\rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)}v_j^{(\ell)}\widetilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)}v_j^{(\ell)}(\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}'(-j,1), \omega^{(\ell)}v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}')(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}'$.

Case 7: $(\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge ((\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*) \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) \le m_2^*))$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, s_i^{(\ell)}, \widehat{t}_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)}\mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)}(\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)}\rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For $j \leq m_2^*$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi}_j, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

For $j > m_2^*$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'$.

Case 8: $(\mathsf{min}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*)$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, s_i^{(\ell)}, t_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)} \mathbf{v}_i^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)} (\rho_i'(-i,1), 1, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)} \rho_i'(-i,1), y_i^{(\ell)}, 0, 0, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)} \rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)} v_j^{(\ell)} \widetilde{\mathbf{u}}_{j,\beta}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)} v_j^{(\ell)} (\widetilde{\rho_j}'(-j,1), 1, 0, -\beta\widetilde{\xi}_j, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'(-j,1), \omega^{(\ell)} v_j^{(\ell)}, 0, -\beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}')(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \beta\omega^{(\ell)} v_j^{(\ell)}\widetilde{\xi}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)} v_j^{(\ell)} \widetilde{\rho_j}'$.

Case 9: $(\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1') \wedge (\mathsf{min}(I_{\mathbf{v}^{(\ell)}}) > m_2^*)$

Choose $\omega^{(\ell)}, \rho_i''^{(\ell)}, \widetilde{\rho}_j''^{(\ell)}, s_i^{(\ell)}, t_j^{(\ell)}, \gamma_i^{(\ell)}, \widetilde{\gamma}_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$.

For all $i \in I_{\mathbf{y}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![y_i^{(\ell)}\mathbf{u}_{i,\beta}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![y_i^{(\ell)}(\rho_i'(-i,1), 1, 0, -\beta\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_1 \\
&= [\![(y_i^{(\ell)}\rho_i'(-i,1), y_i^{(\ell)}, 0, -\beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^* + (\rho_i''^{(\ell)}(-i,1), 0, \gamma_i^{(\ell)}, s_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![((\rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i')(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \beta y_i^{(\ell)}\xi_i, 0, 0)\mathbf{B}^*]\!]_2.
\end{aligned}$$

For all $j \in I_{\mathbf{v}^{(\ell)}}$,

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![\omega^{(\ell)}v_j^{(\ell)}\widetilde{\mathbf{v}}_j^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![\omega^{(\ell)}v_j^{(\ell)}(\widetilde{\rho_j}'(-j,1), 1, 0, 0, 0, 0)\widehat{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_1 \\
&= [\![(\omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}'(-j,1), \omega^{(\ell)}v_j^{(\ell)}, 0, 0, 0, 0)\widetilde{\mathbf{B}}^* + (\widetilde{\rho_j}''^{(\ell)}(-j,1), 0, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![((\widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}')(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.
\end{aligned}$$

We have implicitly set $\rho_i^{(\ell)} = \rho_i''^{(\ell)} + y_i^{(\ell)}\rho_i'$ and $\widetilde{\rho}_j^{(\ell)} = \widetilde{\rho_j}''^{(\ell)} + \omega^{(\ell)}v_j^{(\ell)}\widetilde{\rho_j}'$.

Thus $\mathcal{A}$'s view is the same as in Game 5 if $\beta = 0$ and in Game 6 if $\beta = 1$. Therefore, we have

$$|\Pr(\mathsf{E}_5) - \Pr(\mathsf{E}_6)| \leq 8(m_{2,\mathsf{max}} + m_{1,\mathsf{max}}) \cdot \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$$

due to Lemma 3. □

Claim 9. $|\Pr(\mathsf{E}_7) - \Pr(\mathsf{E}_6)| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

Proof. Let $\mathcal{B}$ obtain an instance of $(\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e), [\![a]\!]_2 = g_2^a, [\![u]\!]_2 = g_2^u, [\![t_\beta]\!]_2 = [\![au + \beta f]\!]_2 = g_2^{au+\beta f})$ of the SXDH assumption for $\iota = 2$ where $a, u, f \leftarrow \mathbb{Z}_p$, $\beta \leftarrow \{0,1\}$ and sets $\mathsf{PP} = (p, g_1, g_2, g_T, V, V^*, E)$. We will show that $\mathcal{B}$ can utilize the instances of the SXDH assumption to interpolate between Game 6 and Game 7 using $\mathcal{A}$ as a subroutine. The algorithm $\mathcal{B}$ implicitly define two orthonormal dual bases $(\widetilde{\mathbf{B}}, \widetilde{\mathbf{B}}^*)$ by choosing $\mathbf{D}, \widetilde{\mathbf{D}} \leftarrow \mathsf{GL}_7(\mathbb{Z}_p)$ and setting

$$\widehat{\mathbf{B}} = \begin{bmatrix} I_2 & & & & \\ & 0 & 0 & -1 & \\ & 0 & 1 & 0 & \\ & 1 & 0 & a & \\ & & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}, \quad \widetilde{\mathbf{B}}^* = \begin{bmatrix} I_2 & & & & \\ & a & 0 & -1 & \\ & 0 & 1 & 0 & \\ & 1 & 0 & 0 & \\ & & & & I_2 \end{bmatrix} \widetilde{\mathbf{D}}^*$$

where $\widetilde{\mathbf{D}}^* = (\widetilde{\mathbf{D}}^{-1})^\top$ and a is implicitly provided through the SXDH instance. Note that, by using $[\![a]\!]_2 = g_2^a$, the algorithm $\mathcal{B}$ can compute the first four rows $\{[\![\mathbf{b}_i]\!]_1^*, [\![\widetilde{\mathbf{b}}_i^*]\!]_1\}_{i \in \{1,2,\ldots,4\}}$ of $\widetilde{\mathbf{B}}^*$. Note that, $(0, 0, t_\beta, 0, -u, 0, 0)\widetilde{\mathbf{D}}^* = (0, 0, u, 0, \beta f, 0, 0)\widetilde{\mathbf{B}}^*$. For $\langle \mathbf{w}^{(0)}, \mathbf{v}^{(\ell)} \rangle \neq 0, \langle \mathbf{w}^{(1)}, \mathbf{v}^{(\ell)} \rangle \neq 0$, the algorithm $\mathcal{B}$ simulates the ℓ-th secret key $\mathsf{SK}_{\mathbf{y}^{(\ell)}, \mathbf{v}^{(\ell)}} =$

$(\{[\![\mathbf{k}_i^1]\!]_2\}_{i\in I_{\mathbf{y}^{(\ell)}}}, \{[\![\mathbf{k}_j^2]\!]_2\}_{j\in I_{\mathbf{v}^{(\ell)}}}, I_{\mathbf{y}^{(\ell)}}, I_{\mathbf{v}^{(\ell)}})$ corresponding to the vectors $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i\in I_{\mathbf{y}^{(\ell)}}}, \mathbf{v}^{(\ell)} = (v_j^{(\ell)})_{j\in I_{\mathbf{v}^{(\ell)}}}$ as follows:

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \widetilde{\xi}_j(\omega^{(\ell)} + u\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle)v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*\\
&\quad + v_j^{(\ell)}\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle(0,0,t_\beta,0,-u,0,0)\widetilde{\mathbf{D}}^*]\!]_2\\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)}v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \widetilde{\xi}_j(\omega^{(\ell)} + u\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle)v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*\\
&\quad + v_j^{(\ell)}\langle\mathbf{w}, \mathbf{v}^{(\ell)}\rangle(0,0,u,0,\beta f,0,0)\widetilde{\mathbf{B}}^*]\!]_2\\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), (\omega^{(\ell)} + u\langle\mathbf{w}, \mathbf{v}^{(\ell)}\rangle)v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \widetilde{\xi}_j(\omega^{(\ell)} + u\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle)v_j^{(\ell)}+\\
&\quad \beta f v_j^{(\ell)}\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2
\end{aligned}$$

with $\widetilde{\rho}_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i\in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j\in I_{\mathbf{v}^{(\ell)}}} \widetilde{\gamma}_j^{(\ell)} = 0$ and $\sum_{i\in I_{\mathbf{y}^{(\ell)}}} s_j^{(\ell)} + \sum_{j\in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0$ where both $\gamma_i^{(\ell)}, s_i^{(\ell)}$ are uniformly chosen from $\mathbb{Z}_p$. As $\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle \neq 0$, we can implicitly set $\omega^{(\ell)\prime} = \omega^{(\ell)} + u\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle, \mathfrak{r}_j^{(\ell)} = t_j^{(\ell)} - \widetilde{\xi}_j(\omega^{(\ell)} + u\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle)v_j^{(\ell)} + fv_j^{(\ell)}\langle\mathbf{w}^{(0)}, \mathbf{v}^{(\ell)}\rangle$ which are random elements in $\mathbb{Z}_p$ for $f \neq 0$. Therefore, the fifth component of $[\![\mathbf{k}_j^2]\!]_2$ is random element for $\beta = 1$. Here, we use a fact that $\widetilde{\mathfrak{r}}_j^{(\ell)} + s_i^{(\ell)} + \xi_i y_i^{(\ell)} \neq 0$ with high probability. Hence, the adversarial view is the same as in Game 7 for $\beta = 1$, otherwise, the view is similar as in Game 6 if $\beta = 0$. Let choose $\sigma \leftarrow \mathbb{Z}_p$ and computes $(0,0,\sigma,0,0,0,0)\widetilde{\mathbf{D}} = (0,0,a\sigma,0,\sigma,0,0)\widetilde{\mathbf{B}}$ and $(0,0,\sigma,0,0,0,0)\mathbf{D} = (0,0,a\sigma,0,\sigma,0,0)\mathbf{B}$. Now, the challenge ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}^{(0)} = (\{[\![\mathbf{c}_i^1]\!]_1\}_{i\in[m_1^*]}, \{[\![\mathbf{c}_j^2]\!]_1\}_{j\in[m_2^*]})$ components $[\![\mathbf{c}_j^2]\!]_1, [\![\mathbf{c}_i^1]\!]_1$ are generated by $\mathcal{B}$ as follows:

$$\begin{aligned}
[\![\mathbf{c}_j^2]\!]_1 &= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)} + \sigma\widetilde{\xi}_j', \alpha, 0, 0, 0)\widetilde{\mathbf{B}} + (0,0,\sigma,0,0,0,0)\widetilde{\mathbf{D}}]\!]_1.\\
&= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)} + \sigma\widetilde{\xi}_j', \alpha, 0, 0, 0)\widetilde{\mathbf{B}} + (0,0,a\sigma,0,\sigma,0,0)\widetilde{\mathbf{B}}]\!]_1\\
&= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)} + \sigma\widetilde{\xi}_j, \alpha, \sigma, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \quad \forall j \in [m_2^*]
\end{aligned}$$

where $\widetilde{\pi}_j, \delta, \widetilde{\xi}_j', \alpha \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}^{(\ell)}}$. Note that, $\{[\![\mathbf{b}_i]\!]_1, [\![\widetilde{\mathbf{b}}_i]\!]_1\}_{i\in\{1,2,\ldots,5\}}$ are sufficient to compute $[\![(\pi_i(1,i), x_i^{(0)}, \alpha, 0, 0, 0)\mathbf{B}]\!]_1$ and $[\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)}, \alpha, 0, 0, 0)\widetilde{\mathbf{B}}]\!]_1$ respectively. Without knowledge of $[\![a]\!]_1$ here $\mathcal{B}$ cannot compute $[\![\mathbf{b}_5]\!]_1, [\![\widetilde{\mathbf{b}}_5]\!]_1$ as the rows $\mathbf{b}_5, \widetilde{\mathbf{b}}_5$ consist of the element a and $\mathcal{B}$ has no information about $[\![a]\!]_1$. Thus the distribution of the challenge ciphertext components in Game 6 is identical with the distribution of Game 7. Hence, $\mathcal{B}$ interpolates between Game 7 and Game 6 and the claim follows. □

Claim 10. $|\Pr(\mathsf{E}_8) - \Pr(\mathsf{E}_7)| \leq 2^{-\Omega(\lambda)}$.

Proof. Let $\widetilde{\mathsf{E}}_\iota$ be the event that denotes $m_1' = m_1^*$ in Game ι where m_1' is the guess of the length m_1^* of message vector. Since $\mathcal{A}$'s view are equivalent for all previous ciphertext query, we have $\Pr(\widetilde{\mathsf{E}}_7) = \Pr(\widetilde{\mathsf{E}}_8)$. Let us define for all $i \in [m_1']$, $j \in [m_2^*]$ as follows:

$$\xi_i' = \xi_i - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}, \quad \widetilde{\xi}_j' = \widetilde{\xi}_j - \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}$$

where $\sigma, \delta \leftarrow \mathbb{Z}_p$ and $(\mathbf{x}^{(0)}, \mathbf{w}^{(0)}), (\mathbf{x}^{(1)}, \mathbf{w}^{(1)})$ are challenge message and attribute pairs. Note that, $\xi'_i, \widetilde{\xi}'_j$ are independently random elements in $\mathbb{Z}_p$ unless $\sigma = 0$. Then the challenge ciphertext components $[\![\mathbf{c}_i^1]\!]_1$ and $[\![\mathbf{c}_j^2]\!]_1$ are indistinguishable in Game 7 and Game 8 as shown below,

$$\begin{aligned}
[\![\mathbf{c}_i^1]\!]_1 &= [\![(\pi_i(1,i), x_i^{(0)} + \xi_i\sigma, \alpha, \sigma, 0, 0)\mathbf{B}]\!]_1 \\
&= [\![(\pi_i(1,i), x_i^{(0)} + \sigma(\xi'_i + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), \alpha, \sigma, 0, 0)\mathbf{B}]\!]_1 \\
&= [\![(\pi_i(1,i), x_i^{(1)} + \sigma\xi'_i, \alpha, \sigma, 0, 0)\mathbf{B}]\!]_1 \quad \forall i \in [m'_1]
\end{aligned}$$

$$\begin{aligned}
[\![\mathbf{c}_j^2]\!]_1 &= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)} + \widetilde{\xi}_j\sigma, \alpha, \sigma, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \\
&= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(0)} + \sigma(\widetilde{\xi}'_j + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), \alpha, \sigma, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \\
&= [\![(\widetilde{\pi}_j(1,j), \delta w_j^{(1)} + \sigma\widetilde{\xi}'_j, \alpha, \sigma, 0, 0)\widetilde{\mathbf{B}}]\!]_1 \quad \forall j \in [m_2^*]
\end{aligned}$$

where $\pi_i, \sigma, \alpha \leftarrow \mathbb{Z}_p$ and $\widetilde{\pi}_j \leftarrow \mathbb{Z}_p$. For all $\ell \in [\mathsf{Q_{SK}}]$ we categorise adversary's queries to the ℓ-th oracle secret key on $\mathbf{y}^{(\ell)} = (y_i^{(\ell)})_{i \in I_{\mathbf{y}^{(\ell)}}}, \mathbf{v}^{(\ell)} = (v_j^{(\ell)})_{j \in I_{\mathbf{v}^{(\ell)}}}$ and show that in each cases the ℓ-th secret key components $[\![\mathbf{k}_i^1]\!]_2, [\![\mathbf{k}_j^2]\!]_2$ are indistinguishable in Game 7 and Game 8.

Case I when $\langle \mathbf{w}^{(0)}, \mathbf{v}^{(\ell)} \rangle \neq 0, \langle \mathbf{w}^{(1)}, \mathbf{v}^{(\ell)} \rangle \neq 0$.

(i) If $(\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m'_1) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*)$, then

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![\left(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \mathfrak{r}_i^{(\ell)}, 0, 0\right)\mathbf{B}^*]\!]_2 \\
[\![\mathbf{k}_j^2]\!]_2 &= [\![\left(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widetilde{\mathfrak{r}}_j^{(\ell)}, 0, 0\right)\widetilde{\mathbf{B}}^*]\!]_2
\end{aligned}$$

where $\widetilde{\mathfrak{r}}_j^{(\ell)}, \mathfrak{r}_i^{(\ell)} \leftarrow \mathbb{Z}_p$ for all $j \in I_{\mathbf{v}^{(\ell)}}, i \in I_{\mathbf{y}^{(\ell)}}$. Since $\mathbf{k}_i^1$ and $\mathbf{k}_j^2$ does not contain the value ξ_i and $\widetilde{\xi}_j$, so there is no need to use the transformations as mentioned above. So the distributions for the ℓ-th secret key components $\mathbf{k}_i^1, \mathbf{k}_j^2$ remain unaltered as Game 8.

(ii) If $(\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m'_1) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*)$, then

For $i \leq m'_1$,

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi_i y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - y_i^{(\ell)}(\xi'_i + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi'_i y_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}^{(\ell)}}.
\end{aligned}$$

For $i > m'_1$;

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\widetilde{\xi}'_j + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \widetilde{\xi}'_j - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \quad \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

Hence, we set $\widehat{\mathfrak{s}}_i^{(\ell)} = \widehat{s}_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$ for $i \leq m'_1$ which are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(x_i^{(0)} - x_i^{(1)})y_i^{(\ell)}$ and $\widehat{s}_i^{(\ell)}$ are independently random elements in $\mathbb{Z}_p$. Also, $\widehat{\mathfrak{s}}_i^{(\ell)}$ are random elements from $i > m'_1$, so fifth component of $\mathbf{k}_i^1$ is uniform element from $\mathbb{Z}_p$ for all $i \in I_{\mathbf{y}^{(\ell)}}$. Similarly set, $\mathfrak{t}_j^{(\ell)} = t_j^{(\ell)} - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}$ which are uniformly random in $\mathbb{Z}_p$.

(iii) If $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m'_1) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*$, then

$$\begin{aligned}
\llbracket \mathbf{k}_i^1 \rrbracket_2 &= \llbracket (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \xi_i y_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - y_i^{(\ell)}(\xi'_i + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), 0, 0)\mathbf{B}^* \rrbracket_2 \\
&= \llbracket (\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \xi'_i y_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}, 0, 0)\mathbf{B}^* \rrbracket_2 \quad \forall i \in I_{\mathbf{y}^{(\ell)}}.
\end{aligned}$$

For $j \leq m_2^*$;

$$\begin{aligned}
\llbracket \mathbf{k}_j^2 \rrbracket_2 &= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\widetilde{\xi}'_j + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \\
&= \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \widetilde{\xi}'_j - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2 \quad \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

For $j > m_2^*$;

$$\llbracket \mathbf{k}_j^2 \rrbracket_2 = \llbracket (\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^* \rrbracket_2.$$

Hence, we set $\mathfrak{s}_i^{(\ell)} = s_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$ for $i \in I_{\mathbf{y}^{(\ell)}}$ which are independently random elements from $\mathbb{Z}_p$ as there are no condition on $(x_i^{(0)} - x_i^{(1)})y_i^{(\ell)}$. Similarly take $\widehat{\mathfrak{t}}_j^{(\ell)} = \widehat{t}_j^{(\ell)} - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}$ for $j \leq m_2^*$, which are uniformly random element in $\mathbb{Z}_p$ and also for $j > m_2^*$, $\widehat{\mathfrak{t}}_j^{(\ell)}$ are uniform elements in $\mathbb{Z}_p$. So the fifth component of $\mathbf{k}_j^2$ are independently random elements in $\mathbb{Z}_p$.

(iv) If $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m'_1) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m^*_2$, then
For $i \leq m'_1$;

$$\begin{aligned}
[\![\mathbf{k}^1_i]\!]_2 &= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \widehat{s}^{(\ell)}_i - \xi_i y^{(\ell)}_i, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \widehat{s}^{(\ell)}_i - y^{(\ell)}_i(\xi'_i + \frac{x^{(1)}_i - x^{(0)}_i}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \widehat{s}^{(\ell)}_i - \xi'_i y^{(\ell)}_i - \frac{x^{(1)}_i - x^{(0)}_i}{\sigma} y^{(\ell)}_i, 0, 0)\mathbf{B}^*]\!]_2
\end{aligned}$$

and for $i > m'_1$ and $i \in I_{\mathbf{y}^{(\ell)}}$, we set

$$[\![\mathbf{k}^1_i]\!]_2 = [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, \widehat{s}^{(\ell)}_i, 0, 0)\mathbf{B}^*]\!]_2$$

where $\gamma^{(\ell)}_i, \widehat{r}^{(\ell)}_i, \rho^{(\ell)}_i \leftarrow \mathbb{Z}_p$. As there are no condition on $\sum_{i \in I_{\mathbf{y}^{(\ell)}}}(x^{(1)}_i - x^{(0)}_i)y^{(\ell)}_i$ i.e., $\sum_{i \in I_{\mathbf{y}^{(\ell)}}}(x^{(1)}_i - x^{(0)}_i)y^{(\ell)}_i \neq 0$ or not, let us define $\widehat{\mathfrak{s}}^{(\ell)}_i = \widehat{s}^{(\ell)}_i - \frac{x^{(1)}_i - x^{(0)}_i}{\sigma} y^{(\ell)}_i$ which is uniformly random in $\mathbb{Z}_p$ for $i \leq m'_1$ as $\widehat{s}^{(\ell)}_i$ is uniformly random over $\mathbb{Z}_p$.

For $j \leq m^*_2$, we set

$$\begin{aligned}
[\![\mathbf{k}^2_j]\!]_2 &= [\![(\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \widehat{t}^{(\ell)}_j - \widetilde{\xi}_j \omega^{(\ell)} v^{(\ell)}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \widehat{t}^{(\ell)}_j - \omega^{(\ell)} v^{(\ell)}_j(\widetilde{\xi}'_j + \frac{\delta(w^{(1)}_j - w^{(0)}_j)}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \widehat{t}^{(\ell)}_j - \omega^{(\ell)} v^{(\ell)}_j \widetilde{\xi}'_j - \frac{\omega^{(\ell)}\delta(w^{(1)}_j - w^{(0)}_j)}{\sigma} v^{(\ell)}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2
\end{aligned}$$

and for $j > m^*_2$, we set

$$[\![\mathbf{k}^2_j]\!]_2 = [\![(\widetilde{\rho}^{(\ell)}_j(-j,1), \omega^{(\ell)} v^{(\ell)}_j, \widetilde{\gamma}^{(\ell)}_j, \widehat{t}^{(\ell)}_j, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \quad \forall j \in I_{\mathbf{v}^{(\ell)}}$$

with $\widetilde{\gamma}^{(\ell)}_j, \widehat{t}^{(\ell)}_j, \widetilde{\rho}^{(\ell)}_j, \omega^{(\ell)} \leftarrow \mathbb{Z}_p$. As $\sum_{j \in I_{\mathbf{v}^{(\ell)}}}(w^{(1)}_j - w^{(0)}_j)v^{(\ell)}_i \neq 0$, $\widehat{\mathfrak{t}}_j = \widehat{t}^{(\ell)}_j - \frac{\omega^{(\ell)}\delta(w^{(1)}_j - w^{(0)}_j)}{\sigma} v^{(\ell)}_j$ for $j \leq m^*_2$ are independently random elements from $\mathbb{Z}_p$ and for $j > m^*_2$, the fifth component of $[\![\mathbf{k}^2_j]\!]_2$ is also random.

Case II when $\langle \mathbf{w}^{(0)}, \mathbf{v}^{(\ell)} \rangle = \langle \mathbf{w}^{(1)}, \mathbf{v}^{(\ell)} \rangle = 0$.

(v) If $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m'_1) \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m^*_2$, then

$$\begin{aligned}
[\![\mathbf{k}^1_i]\!]_2 &= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, s^{(\ell)}_i - \xi_i y^{(\ell)}_i, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, s^{(\ell)}_i - y^{(\ell)}_i(\xi'_i + \frac{x^{(1)}_i - x^{(0)}_i}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho^{(\ell)}_i(-i,1), y^{(\ell)}_i, \gamma^{(\ell)}_i, s^{(\ell)}_i - \xi'_i y^{(\ell)}_i - \frac{x^{(1)}_i - x^{(0)}_i}{\sigma} y^{(\ell)}_i, 0, 0)\mathbf{B}^*]\!]_2 \;\; \forall i \in I_{\mathbf{y}^{(\ell)}},
\end{aligned}$$

$$\begin{aligned}[\![\mathbf{k}_j^2]\!]_2 &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \tilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\ &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\tilde{\xi}_j' + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\ &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \tilde{\xi}_j' - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}^{(\ell)}}\end{aligned}$$

where $\gamma_i^{(\ell)}, s_i^{(\ell)}, \rho_i^{(\ell)} \leftarrow \mathbb{Z}_p$ and $\tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)}, \tilde{\rho}_j^{(\ell)}, \omega^{(\ell)} \leftarrow \mathbb{Z}_p$ such that $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j^{(\ell)} = 0$ and $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} \gamma_i^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} \tilde{\gamma}_j^{(\ell)} = 0$. Since $\langle \mathbf{x}^{(0)}, \mathbf{y}^{(\ell)} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}^{(\ell)} \rangle$ in challenge query phase as per Def. 2, we get $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} y_i^{(\ell)}(x_i^{(0)} - x_i^{(1)}) = 0$ when $\langle \mathbf{w}^{(0)}, \mathbf{v}^{(\ell)} \rangle = \langle \mathbf{w}^{(1)}, \mathbf{v}^{(\ell)} \rangle = 0$ which yields $\sum_{j \in I_{\mathbf{v}^{(\ell)}}} v_j^{(\ell)}(w_j^{(0)} - w_j^{(1)}) = 0$. We set $s_i'^{(\ell)} = s_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$ and $t_j'^{(\ell)} = t_j^{(\ell)} - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}$ which are uniformly random over $\mathbb{Z}_p$ for all $i \in I_{\mathbf{y}^{(\ell)}}, j \in I_{\mathbf{v}^{(\ell)}}$ respectively and these satisfy $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} s_i'^{(\ell)} + \sum_{j \in I_{\mathbf{v}^{(\ell)}}} t_j'^{(\ell)} = 0$ as in Game 7.

(vi) If $(\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) \leq m_2^*$, then

For $i \leq m_1'$,

$$\begin{aligned}[\![\mathbf{k}_i^1]\!]_2 &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi_i y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\ &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - y_i^{(\ell)}(\xi_i' + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\ &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi_i' y_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \ \ \forall i \in I_{\mathbf{y}^{(\ell)}}.\end{aligned}$$

For $i > m_1'$;

$$[\![\mathbf{k}_i^1]\!]_2 = [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2$$

$$\begin{aligned}[\![\mathbf{k}_j^2]\!]_2 &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \tilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\ &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\tilde{\xi}_j' + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\ &= [\![(\tilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \tilde{\gamma}_j^{(\ell)}, t_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \tilde{\xi}_j' - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \ \ \forall j \in I_{\mathbf{v}^{(\ell)}}.\end{aligned}$$

Hence, we set $\widehat{\mathfrak{s}}_i^{(\ell)} = \widehat{s}_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$ for $i \leq m_1'$ which are independently random elements from $\mathbb{Z}_p$ as we have no condition on $(x_i^{(0)} - x_i^{(1)}) y_i^{(\ell)}$, also $\widehat{\mathfrak{s}}_i^{(\ell)}$ are independently random elements from $i > m_1'$, so fifth component of $\mathbf{k}_i^1$ is uniform in $\mathbb{Z}_p$ for all $i \in I_{\mathbf{y}^{(\ell)}}$. Also set, $\mathfrak{t}_j^{(\ell)} = t_j^{(\ell)}$ (as $\sum_{j \in I_{\mathbf{v}^{(\ell)}}} (w_j^{(0)} - w_j^{(1)}) v_j^{(\ell)} = 0$) which are uniformly random in $\mathbb{Z}_p$ since the corresponding fifth element of $\mathbf{k}_j^2$ is set as random.

(vii) If $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) \leq m_1') \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*$, then

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \xi_i y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - y_i^{(\ell)}(\xi_i' + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, s_i^{(\ell)} - \xi_i' y_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \;\; \forall i \in I_{\mathbf{y}^{(\ell)}}.
\end{aligned}$$

For $j \leq m_2^*$;

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\widetilde{\xi}_j' + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \widetilde{\xi}_j' - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \;\; \forall j \in I_{\mathbf{v}^{(\ell)}}.
\end{aligned}$$

For $j > m_2^*$;

$$[\![\mathbf{k}_j^2]\!]_2 = [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2.$$

Hence, we set $\mathfrak{s}_i^{(\ell)} = s_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$ for $i \in I_{\mathbf{y}^{(\ell)}}$ which are independently random elements from $\mathbb{Z}_p$ as we have no condition on $(x_i^{(0)} - x_i^{(1)}) y_i^{(\ell)}$, Also set, $\widehat{\mathfrak{t}}_j^{(\ell)} = \widehat{t}_j^{(\ell)} - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}$ for $j \leq m_2^*$, which are uniformly random elements from $\mathbb{Z}_p$ and for $j > m_2^*$, $\widehat{\mathfrak{t}}_j^{(\ell)}$'s are uniform in $\mathbb{Z}_p$. So the fifth component of $\mathbf{k}_j^2$ are independently random elements from $\mathbb{Z}_p$.

(viii) If $\mathsf{max}(I_{\mathbf{y}^{(\ell)}}) > m_1') \wedge (\mathsf{max}(I_{\mathbf{v}^{(\ell)}}) > m_2^*$, then:

For $i \leq m_1'$;

$$\begin{aligned}
[\![\mathbf{k}_i^1]\!]_2 &= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi_i y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - y_i^{(\ell)}(\xi_i' + \frac{x_i^{(1)} - x_i^{(0)}}{\sigma}), 0, 0)\mathbf{B}^*]\!]_2 \\
&= [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)} - \xi_i' y_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2
\end{aligned}$$

also for $i > m_1'$ and $i \in I_{\mathbf{y}^{(\ell)}}$, we set

$$[\![\mathbf{k}_i^1]\!]_2 = [\![(\rho_i^{(\ell)}(-i,1), y_i^{(\ell)}, \gamma_i^{(\ell)}, \widehat{s}_i^{(\ell)}, 0, 0)\mathbf{B}^*]\!]_2$$

where $\gamma_i^{(\ell)}, \widehat{r}_i^{(\ell)}, \rho_i^{(\ell)} \leftarrow \mathbb{Z}_p$. Since there are no condition on $\sum_{i \in I_{\mathbf{y}^{(\ell)}}} (x_i^{(1)} - x_i^{(0)}) y_i^{(\ell)}$. Let us define $\widehat{\mathfrak{s}}_i^{(\ell)} = \widehat{s}_i^{(\ell)} - \frac{x_i^{(1)} - x_i^{(0)}}{\sigma} y_i^{(\ell)}$, which is uniformly random in $\mathbb{Z}_p$ for $i \leq m_1'$ as $\widehat{s}_i^{(\ell)}$ is uniformly random in $\mathbb{Z}_p$.

For $j \leq m_2^*$, we set

$$\begin{aligned}
[\![\mathbf{k}_j^2]\!]_2 &= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \widetilde{\xi}_j \omega^{(\ell)} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)}(\widetilde{\xi}'_j + \frac{\delta(w_j^{(1)} - w_j^{(0)})}{\sigma}), 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \\
&= [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)} - \omega^{(\ell)} v_j^{(\ell)} \widetilde{\xi}'_j - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2
\end{aligned}$$

and for $j > m_2^*$, we set

$$[\![\mathbf{k}_j^2]\!]_2 = [\![(\widetilde{\rho}_j^{(\ell)}(-j,1), \omega^{(\ell)} v_j^{(\ell)}, \widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, 0, 0)\widetilde{\mathbf{B}}^*]\!]_2 \quad \forall j \in I_{\mathbf{v}^{(\ell)}}$$

with $\widetilde{\gamma}_j^{(\ell)}, \widehat{t}_j^{(\ell)}, \widetilde{\rho}_j^{(\ell)}, \omega^{(\ell)} \leftarrow \mathbb{Z}_p$. Since, $\widehat{\mathfrak{t}}_j^{(\ell)} = \widehat{t}_j^{(\ell)} - \frac{\omega^{(\ell)}\delta(w_j^{(1)} - w_j^{(0)})}{\sigma} v_j^{(\ell)}$ for $j \leq m_2^*$ are independently random elements from $\mathbb{Z}_p$ and for $j > m_2^*$ the fifth component of $\mathbf{k}_j^2$ are also random.

Therefore, Game 7 and Game 8 are indistinguishable except a negligible probability i.e.,

$$|\Pr(\mathsf{E}_8) - \Pr(\mathsf{E}_7)| = \left|\Pr(\widetilde{\mathsf{E}}_8) \cdot \Pr(\mathsf{E}_8|\widetilde{\mathsf{E}}_8) - \Pr(\widetilde{\mathsf{E}}_7) \cdot \Pr(\mathsf{E}_7|\widetilde{\mathsf{E}}_7)\right| \leq 2^{-\Omega(\lambda)}.$$

This establishes the claim. □

Claim 11. $|\Pr(\mathsf{E}_9) - \Pr(\mathsf{E}_8)| \leq \Big[8(m_{1,\mathsf{max}} + m_{2,\mathsf{max}}) + 4(s_{\mathsf{max}} - 1) + 4m_{2,\mathsf{max}}(t_{\mathsf{max}} - 1)\Big]\mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

The proof of this claim can be achieved utilizing the proofs of claims 9, 7 and 5.

Claim 12. $\Pr(\mathsf{E}_9) \leq \frac{1}{m_{1,\mathsf{max}}} \cdot \Pr(\mathsf{G}_{10})$.

This proof is exactly the same as that of claim 6.

Claim 13. $|\Pr(\mathsf{E}_{11}) - \Pr(\mathsf{E}_{10})| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

This proof is exactly the same as that of claim 4.

Claim 14. $|\Pr(\mathsf{E}_{12}) - \Pr(\mathsf{E}_{11})| \leq \mathsf{Adv}_{\mathcal{B}}^{\mathsf{SXDH}}(\lambda) + 2^{-\Omega(\lambda)}$.

This proof is exactly the same as that of claim 3.

□

5 Our Succinct UQFE

In the following, we define unbounded quadratic functional encryption (UQFE) for the message spaces $\{\mathcal{X}_\lambda\}_\lambda^2$, a key space $\{\mathcal{Y}_\lambda\}_\lambda$ for any $\lambda \in \mathbb{N}$ which is a security parameter. In our definition, the message vectors $\mathbf{z}_1 \in \mathbb{Z}_p^{n_1}, \mathbf{z}_2 \in \mathbb{Z}_p^{n_2}$ are associated with index sets $I_{\mathbf{z}_1}, I_{\mathbf{z}_2}$ respectively and the key vector $\mathbf{f} \in \mathbb{Z}_p^{n_1 n_2}$ is associated with the index set $I_{\mathbf{f}}$. We assume that the index set $I_{\mathbf{f}}$ is a cartesian product between two index sets $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$. In the permissive case of UQFE scheme, it recovers $(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top$ if and only if $I_{\mathbf{f}_1} \subseteq I_{\mathbf{z}_1}$ and $I_{\mathbf{f}_2} \subseteq I_{\mathbf{z}_2}$ and in the strict case, it outputs $(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top$ if and only if $I_{\mathbf{z}_1} = I_{\mathbf{f}_1}$ and $I_{\mathbf{z}_2} = I_{\mathbf{f}_2}$. Clearly, it can be observed that if the UQFE scheme is permissive then it also satisfies the condition of strict relation. So for simplicity here we define the UQFE scheme in permissive setting. Our $\mathsf{UQFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ consists of four PPT algorithms satisfying the following requirements.

$\mathsf{Setup}(1^\lambda) \rightarrow (\mathsf{PP}, \mathsf{MSK})$: The setup algorithm takes as input the security parameter 1^λ, and outputs a public parameter and a master secret key pair $(\mathsf{PP}, \mathsf{MSK})$.

$\mathsf{Enc}(\mathsf{PP}, \mathbf{z}_1, \mathbf{z}_2) \rightarrow \mathsf{CT}$: The encryption algorithm takes as input the public parameter PP and a pair of message vectors $(\mathbf{z}_1, \mathbf{z}_2) \in \mathcal{X}_\lambda \times \mathcal{X}_\lambda$ with associated index sets $I_{\mathbf{z}_1}, I_{\mathbf{z}_2}$ respectively, and outputs a ciphertext CT.

$\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbf{f}) \rightarrow \mathsf{SK}_{\mathbf{f}}$: The key generation algorithm takes as input the public parameter PP, the master secret key MSK and a function $\mathbf{f} \in \mathcal{Y}_\lambda$ with an associated index set $I_{\mathbf{f}}$, and outputs a secret key $\mathsf{SK}_{\mathbf{f}}$.

$\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathbf{f}}, \mathsf{CT}) \rightarrow d/\perp$: The decryption algorithm takes as input the secret key $\mathsf{SK}_{\mathbf{f}}$, a ciphertext CT and the vector $\mathbf{f}$, and outputs a value d or the symbol $\perp$.

Correctness: An UQFE scheme is said to be correct if for any $\lambda \in \mathbb{N}$, any pair of message vectors $(\mathbf{z}_1, \mathbf{z}_2)$ with associated index sets $I_{\mathbf{z}_1}, I_{\mathbf{z}_2}$, any secret key vector $\mathbf{f}$ with associated index set $I_{\mathbf{f}} = I_{\mathbf{f}_1} \times I_{\mathbf{f}_2}$ satisfying $I_{\mathbf{f}_1} \subseteq I_{\mathbf{z}_1}$ and $I_{\mathbf{f}_2} \subseteq I_{\mathbf{z}_2}$, it holds that

$$\Pr\left[\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathbf{f}}, \mathsf{CT}) = (\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top : \begin{array}{l} (\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda) \\ \mathsf{CT} \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \mathbf{z}_1, \mathbf{z}_2) \\ \mathsf{SK}_{\mathbf{f}} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbf{f}) \end{array}\right] = 1.$$

Succinctness and Compactness: An UQFE is said to be succinct if the secret key size is independent of the size of the function $\mathbf{f}$, i.e. $|\mathsf{SK}_{\mathbf{f}}| = O(1)$, and the ciphertext size is linear in the size of $\mathbf{z}_1$ and $\mathbf{z}_2$, i.e. $|\mathsf{CT}| = O(|\mathbf{z}_1|) + O(|\mathbf{z}_2|)$.

Concurrently, Tomida [Tom22] studied UQFE in the public key setting and presented a construction with IND-based security model. We use UQFE of [Tom22] to instantiate our public key UNP-IPFE. The secret key UNP-IPFE is, however, proved in the SIM-based model and depends on a secret key UQFE which encrypts message vectors in the presence of MSK. Thus, we present the SIM-based security notion of UQFE in the secret key setting below and for completeness the IND-based security model is given in Appendix A.

Definition 4 (SA-SIM Security for UQFE). *The* $\mathsf{UQFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is said to be semi-adaptive simulation* (SA-SIM) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a PPT simulator* $\mathcal{S} := (\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$ *such that the following holds*

$$\mathsf{Adv}^{\mathsf{UQFE}}_{\mathcal{A},\mathsf{SA\text{-}SIM}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UQFE},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UQFE},\mathcal{A},\mathcal{S}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda)$$

where the experiments $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UQFE},\mathcal{A}}(\lambda)$ *and* $\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UQFE},\mathcal{A},\mathcal{S}}(\lambda)$ *are defined as follows:*

```
$\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UQFE},\mathcal{A}}(\lambda)$
1: $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$
2: $(\mathbf{z}_1^*, \mathbf{z}_2^*) \leftarrow \mathcal{A}(\mathsf{PP})$
3: $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \mathbf{z}_1^*, \mathbf{z}_2^*)$
4: $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{PP},\mathsf{MSK},\cdot)}(\mathsf{CT}^*)$.
```

```
$\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UQFE},\mathcal{A},\mathcal{S}}(\lambda)$
1: $(\mathsf{PP}^*, \mathsf{MSK}^*) \leftarrow \mathsf{Setup}^*(1^\lambda)$
2: $(\mathbf{z}_1^*, \mathbf{z}_2^*) \leftarrow \mathcal{A}(\mathsf{PP}^*)$
3: $\mathsf{CT}^* \leftarrow \mathsf{Enc}^*(\mathsf{PP}^*, \mathsf{MSK}^*, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*})$
4: $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}^*(\mathsf{PP}^*,\mathsf{MSK}^*,\cdot,\cdot)}(\mathsf{CT}^*)$.
```

In the Real *security experiment,* $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \cdot)$ *is an oracle that takes input the secret key vector* $\mathbf{f}$ *with associated the index set* $I_{\mathbf{f}}$ *and outputs* $\mathsf{SK}_{\mathbf{f}} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbf{f})$*. In the* Ideal *security experiment,* $\mathsf{KeyGen}^*(\mathsf{PP}^*, \mathsf{MSK}^*, \cdot, \cdot)$ *oracle returns a simulated secret key* $\mathsf{SK}^*_{\mathbf{f}}$ *on input a key vector* $\mathbf{f}$ *with index set* $I_{\mathbf{f}}$ *and* μ *where the value of* μ *is* $(\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top$ *whenever the conditions* $I_{\mathbf{f}_1} \subseteq I_{\mathbf{z}_1^*}$ *and* $I_{\mathbf{f}_2} \subseteq I_{\mathbf{z}_2^*}$ *hold, else* $\mu = \perp$*.*

5.1 Construction

In this section, we construct a secret key UQFE scheme with strict relation. Let us consider two hash functions $\mathsf{H}_1, \mathsf{H}_2$ and two PRF families $\mathcal{F}_1 = \{F_{K_1}\}_{K_1 \in \mathcal{K}_\lambda}$, $\mathcal{F}_2 = \{F_{K_2}\}_{K_2 \in \mathcal{K}_\lambda}$ with the key space $\mathcal{K}_\lambda$ defined as follows:

- $\mathsf{H}_1 : \mathbb{Z} \to \mathbb{G}_1^{k+1} \times \mathbb{G}_2^{k+1}$ s.t. $\mathsf{H}_1(i) = ([\![\mathbf{a}_i]\!]_1, [\![\mathbf{a}_i]\!]_2) \in \mathbb{G}_1^{k+1} \times \mathbb{G}_2^{k+1}$.
- $\mathsf{H}_2 : \mathbb{Z} \to \mathbb{G}_2^{k'+1}$ s.t. $\mathsf{H}_2(i) = [\![\mathbf{b}_i]\!]_2 \in \mathbb{G}_2^{k'+1}$.
- $\mathcal{F}_1 = \{F_{K_1} | F_{K_1} : \mathbb{Z} \to \mathbb{Z}_p, K_1 \in \mathcal{K}_\lambda\}$ s.t. $F_{K_1}(i) = w_i \in \mathbb{Z}_p$.
- $\mathcal{F}_2 = \{F_{K_2} | F_{K_2} : \mathbb{Z} \to \mathbb{Z}_p, K_2 \in \mathcal{K}_\lambda\}$ s.t. $F_{K_2}(j) = w_j \in \mathbb{Z}_p$.

Our $\mathsf{UQFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ scheme is described below. As all prior works on FEs from DDH and bilinear groups, the required functional value comes from a polynomial range so that at the end of the decryption phase, we can efficiently perform an exhaustive search to obtain the value $(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top$.

$\mathbf{Setup(1^\lambda) \to (PP, MSK)}$: The setup algorithm takes as input security parameter 1^λ and proceeds the following steps:

1. Sample bilinear group $\mathsf{G} = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \leftarrow \mathcal{G}_{\mathsf{BG.Gen}}(1^\lambda)$.
2. Sample $\mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k' \times (k'+1)}$, $\mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1) \times k'}$, $\mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1) \times k}$.
3. Chooses PRF keys $K_1, K_2 \leftarrow \mathcal{K}_\lambda$.
4. Output $\mathsf{PP} = (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{A}_0\mathbf{W}_1]\!]_1, [\![\mathbf{A}_0\mathbf{W}_2]\!]_1)$ and $\mathsf{MSK} = (K_1, K_2, \mathbf{W}_1, \mathbf{W}_2)$.

$\mathsf{\mathbf{Enc}}(\mathsf{\mathbf{PP}}, \mathsf{\mathbf{MSK}}, \mathbf{z_1}, \mathbf{z_2}) \rightarrow \mathsf{\mathbf{CT}}$: The encryption algorithm takes as input the public parameter PP, the master secret key MSK, a message $(\mathbf{z}_1, \mathbf{z}_2) \in \mathbb{Z}_p^{n_1} \times \mathbb{Z}_p^{n_2}$ with its associated index sets $I_{\mathbf{z}_1}, I_{\mathbf{z}_2}$ and executes the following steps:

1. Parse $\mathsf{PP} = (\mathsf{G}, \llbracket \mathbf{A}_0 \rrbracket_1, \llbracket \mathbf{A}_0\mathbf{W}_1 \rrbracket_1, \llbracket \mathbf{A}_0\mathbf{W}_2 \rrbracket_1)$.
2. Parse $I_{\mathbf{z}_1} := \{i_1, \ldots, i_{n_1}\}$, $I_{\mathbf{z}_2} := \{j_1, \ldots, j_{n_2}\}$ for some $n_1, n_2 \geq 1$.
3. Compute the following vectors using the hash functions as

$$\mathsf{H}_1(i_\ell) = \left(\llbracket \mathbf{a}_{i_\ell}^{(1)} \rrbracket_1, \llbracket \mathbf{a}_{i_\ell}^{(1)} \rrbracket_2\right) \in \mathbb{G}_1^{k\times 1} \times \mathbb{G}_2^{k\times 1}, \qquad \forall \ell \in [n_1] \tag{2}$$

$$\mathsf{H}_2(j_\ell) = \llbracket \mathbf{a}_{j_\ell}^{(2)} \rrbracket_2 \in \mathbb{G}_2^{k'\times 1}, \qquad \forall \ell \in [n_2]. \tag{3}$$

4. Set the vectors $\mathbf{w}_1 = (F_{K_1}(i_t))_{t\in[n_1]} \in \mathbb{Z}_p^{n_1}$ and $\mathbf{w}_2 = (F_{K_2}(j_t))_{t\in[n_2]} \in \mathbb{Z}_p^{n_2}$.
5. Set the matrices

$$\llbracket \mathbf{A}_1 \rrbracket_1 = \llbracket \mathbf{a}_{i_1}^{(1)} \ \| \ \ldots \ \| \ \mathbf{a}_{i_{n_1}}^{(1)} \rrbracket_1 \in \mathbb{G}_1^{k\times n_1}, \tag{4}$$

$$\llbracket \mathbf{A}_2 \rrbracket_2 = \llbracket \mathbf{a}_{j_1}^{(2)} \ \| \ \ldots \ \| \ \mathbf{a}_{j_{n_2}}^{(2)} \rrbracket_2 \in \mathbb{G}_2^{k'\times n_2} \tag{5}$$

6. Compute and set $\llbracket \mathbf{A}_0\mathbf{W} \rrbracket_1 := \llbracket \mathbf{A}_0\widetilde{\mathbf{W}}_1 \ \| \ \mathbf{A}_0\widetilde{\mathbf{W}}_2 \rrbracket_1$ where $\widetilde{\mathbf{W}}_1 = \mathbf{W}_1 \otimes \mathbf{w}_1 \in \mathbb{Z}_p^{(k'+1)\times k'n_1}$ and $\widetilde{\mathbf{W}}_2 = \mathbf{W}_2 \otimes \mathbf{w}_2 \in \mathbb{Z}_p^{(k'+1)\times kn_2}$.
7. Sample $\mathbf{s}_1 \leftarrow \mathbb{Z}_p^k$ and $\mathbf{s}_0, \mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k'}$
8. Output the ciphertext

$$\mathsf{CT} = \left(\underbrace{\llbracket \mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1 \rrbracket_1}_{\mathbf{y}_1}, \underbrace{\llbracket \mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2 \rrbracket_2}_{\mathbf{y}_2}, \underbrace{\llbracket \mathbf{s}_0\mathbf{A}_0 \rrbracket_1}_{\mathbf{c}_0}, \underbrace{\llbracket \mathbf{s}_0\mathbf{A}_0\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_2 \ \| \ \mathbf{y}_1 \otimes \mathbf{s}_2) \rrbracket_1}_{\mathbf{y}_0}, I_{\mathbf{z}_1}, I_{\mathbf{z}_2}\right).$$

$\mathsf{\mathbf{KeyGen}}(\mathsf{\mathbf{PP}}, \mathsf{\mathbf{MSK}}, \mathbf{f}) \rightarrow \mathsf{\mathbf{SK}}_{\mathbf{f}}$: The key generation algorithm takes as input the public paramter PP, the master secret keys MSK and a function $\mathbf{f} \in \mathbb{Z}_p^{n_1' n_2'}$ which is associated with an index set $I_{\mathbf{f}}$. It performs as follows:

1. Parse $\mathsf{MSK} = (\mathbf{W}_1, \mathbf{W}_2)$.
2. Parse $I_{\mathbf{f}} = I_{\mathbf{f}_1} \otimes I_{\mathbf{f}_2}$ where $I_{\mathbf{f}_1} := \{i_1', \ldots, i_{n_1'}'\}$, $I_{\mathbf{f}_2} := \{j_1', \ldots, j_{n_2'}'\}$.
3. Use H_1 and H_2 (as in Eq. 2, 3) for the index sets $I_{\mathbf{f}_1}$ and $I_{\mathbf{f}_2}$ to generate the matrices $\llbracket \mathbf{A}_1' \rrbracket_2 \in \mathbb{G}_2^{k\times n_1'}$ and $\llbracket \mathbf{A}_2' \rrbracket_2 \in \mathbb{G}_2^{k'\times n_2'}$ similar to Eq. 4 and 5 respectively.
4. Set the vectors $\mathbf{w}_1' = (F_{K_1}(i_t'))_{t\in[n_1']} \in \mathbb{Z}_p^{n_1'}$ and $\mathbf{w}_2' = (F_{K_2}(j_t'))_{t\in[n_2']} \in \mathbb{Z}_p^{n_2'}$.
5. Define $\mathbf{W}' := (\mathbf{W}_1 \otimes \mathbf{w}_1' \ \| \ \mathbf{W}_2 \otimes \mathbf{w}_2')$.
6. Output the secret key

$$\mathsf{SK}_{\mathbf{f}} = \left(\llbracket \mathbf{W}' \cdot \begin{pmatrix} (\mathbf{A}_1' \otimes \mathbf{I}_{n_2'})\mathbf{f}^\top \\ (\mathbf{I}_{n_1'} \otimes \mathbf{A}_2')\mathbf{f}^\top \end{pmatrix} \rrbracket_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right).$$

$\mathsf{Dec}(\mathsf{PP}, \mathsf{SK}_{\mathbf{f}}, \mathsf{CT}) \to d$: The decryption algorithm takes as input the public parameter PP, the secret key $\mathsf{SK}_{\mathbf{f}}$ of a function $\mathbf{f}$ and a ciphertext CT. It works as follows:

1. Parse $\mathsf{SK}_{\mathbf{f}} = ([\![\mathbf{k}_1^\top]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ and $\mathsf{CT} = ([\![\mathbf{y}_1]\!]_1, [\![\mathbf{y}_2]\!]_2, [\![\mathbf{c}_0]\!]_1, [\![\mathbf{y}_0]\!]_1, I_{\mathbf{z}_1}, I_{\mathbf{z}_2})$.
2. If $I_{\mathbf{z}_1} \neq I_{\mathbf{f}_1}$ or $I_{\mathbf{z}_2} \neq I_{\mathbf{f}_2}$, then output $\perp$.
3. Else compute $\mathbf{k}_2^\top = [\![\begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_2$ where $[\![\mathbf{A}_1]\!]_2, [\![\mathbf{A}_2]\!]_2$ are generated as Eq. 4 and Eq. 5 over the index sets $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$ respectively and output $\log_{g_T} d$ where

$$[\![d]\!]_T = [\![(\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top]\!]_T \cdot e\left([\![\mathbf{c}_0]\!]_1, [\![\mathbf{k}_1^\top]\!]_2\right) \cdot e\left([\![\mathbf{y}_0]\!]_1, [\![\mathbf{k}_2^\top]\!]_2\right)^{-1}. \tag{6}$$

Correctness: If $I_{\mathbf{z}_1} = I_{\mathbf{f}_1}$ and $I_{\mathbf{z}_2} = I_{\mathbf{f}_2}$ then we have $\mathbf{A}_1' = \mathbf{A}_1, \mathbf{A}_2' = \mathbf{A}_2, \mathbf{W}' = \mathbf{W}$. The terms in the decryption equation can be simplified as follows:

$$\begin{aligned}
[\![(\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^T]\!]_T &= [\![(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top + (\mathbf{y}_1 \otimes \mathbf{s}_2\mathbf{A}_2)\mathbf{f}^\top + (\mathbf{s}_1\mathbf{A}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top]\!]_T \\
&= [\![(\mathbf{z}_1 \otimes \mathbf{z}_2)\mathbf{f}^\top]\!]_T \cdot [\![(\mathbf{s}_1 \otimes \mathbf{z}_2 \parallel \mathbf{y}_1 \otimes \mathbf{s}_2) \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_T. \\
e\left([\![\mathbf{c}_0]\!]_1, [\![\mathbf{k}_1^\top]\!]_2\right) &= e\left([\![\mathbf{s}_0\mathbf{A}_0]\!]_1, [\![\mathbf{W} \cdot \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_2\right) \\
&= [\![\mathbf{s}_0\mathbf{A}_0\mathbf{W} \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_T. \\
e\left([\![\mathbf{y}_0]\!]_1, [\![\mathbf{k}_2^\top]\!]_2\right) &= e\left([\![\mathbf{s}_0\mathbf{A}_0\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_2 \parallel \mathbf{y}_1 \otimes \mathbf{s}_2)]\!]_1, [\![\begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_2\right) \\
&= [\![\mathbf{s}_0\mathbf{A}_0\mathbf{W} \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix} + (\mathbf{s}_1 \otimes \mathbf{z}_2 \parallel \mathbf{y}_1 \otimes \mathbf{s}_2) \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}]\!]_T.
\end{aligned}$$

Putting everything together, it can be seen that correctness follows from Eq. 6.

Succinctness and Compactness: A salient feature of our UQFE is the *succinctness* of secret keys. A secret key $\mathsf{SK}_{\mathbf{f}}$ consists of only $(k'+1)$ elements of $\mathbb{G}_2$[3], no matter how long is the vector $\mathbf{f}$. Further, the ciphertext size is *compact*. It consists of $(k'+1)(n_1+1)$ elements from $\mathbb{G}_1$ and kn_2 elements from $\mathbb{G}_2$. Concretely, the size of the secret key could be as small as $2|\mathbb{G}_2|$ and the ciphertext is $2(n_1+1)|\mathbb{G}_1| + 2n_2|\mathbb{G}_2|$ where $|\mathbb{G}|$ represents the size of a single element of the group $\mathbb{G}$. The public key UQFE of [Tom22] is designed for $n_1 = n_2 = n$. For message vectors of lengths n and key vectors of length $(n')^2$, the size of the the ciphertext is $(26n+21)|\mathbb{G}_1| + 12n|\mathbb{G}_2|$ and that of the secret key is at least $(14n'+9)|\mathbb{G}_2|$.

[3] We do not include the function $\mathbf{f}$ while measuring the actual size of $\mathsf{SK}_{\mathbf{f}}$ since a secret key holder always has the corresponding to the key.

5.2 Simulator

We now describe the simulator of our UQFE before going to the formal security analysis.

$\textsf{Setup}^*(1^\lambda)$: Sample $\mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k'\times(k'+1)}, \mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1)\times k'}, \mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1)\times k}, \mathbf{W} \leftarrow \mathbb{Z}_p^{(k'+1)\times(kn_2+k'n_1)}$, $\mathbf{a}_0^\perp \leftarrow \mathbb{Z}_p^{k'+1}, \mathbf{u} \leftarrow \mathbb{Z}_p^{1\times(k'+1)}$ such that $\mathbf{A}_0 \cdot \mathbf{a}_0^\perp = 0; \mathbf{u} \cdot \mathbf{a}_0^\perp = 1$. Choose $K_1, K_2 \leftarrow \mathcal{K}_\lambda$. Then it generates

$$\mathsf{PP}^* = (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{A}_0\mathbf{W}_1]\!]_1, [\![\mathbf{A}_0\mathbf{W}_2]\!]_1), \quad \mathsf{MSK}^* = (K_1, K_2, \mathbf{W}_1, \mathbf{W}_2, \mathbf{W}, \mathbf{u}, \mathbf{a}_0^\perp).$$

$\textsf{Enc}^*(\mathsf{PP}^*, \mathsf{MSK}^*, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*})$: Outputs

$$\mathsf{CT}^* = \Big(\mathbf{y}_1, \mathbf{y}_2, \mathbf{c}_0 = [\![\mathbf{u}]\!]_1, \mathbf{y}_0 = [\![\mathbf{u}\mathbf{W}]\!]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\Big)$$

where $\mathbf{y}_1 \leftarrow \mathbb{G}_1^{|I_{\mathbf{z}_1^*}|}, \mathbf{y}_2 \leftarrow \mathbb{G}_2^{|I_{\mathbf{z}_2^*}|}$.

$\textsf{KeyGen}^*(\mathsf{PP}^*, \mathsf{MSK}^*, \boldsymbol{\mu}, \mathbf{f})$: Parse $I_\mathbf{f} = I_{\mathbf{f}_1} \otimes I_{\mathbf{f}_2}$. If $I_{\mathbf{f}_1} \neq I_{\mathbf{z}_1^*}$ or $I_{\mathbf{f}_2} \neq I_{\mathbf{z}_2^*}$ then the secret key is computed as in the real key generation algorithm, i.e., using $\mathbf{W}_1, \mathbf{W}_2$, it outputs

$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = [\![\mathbf{W}' \begin{pmatrix} (\mathbf{A}_1' \otimes \mathbf{I}_{n_2'})\mathbf{f}^\top \\ (\mathbf{I}_{n_1'} \otimes \mathbf{A}_2')\mathbf{f}^\top \end{pmatrix}]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$

where $\mathbf{W}' := (\mathbf{W}_1 \otimes \mathbf{w}_1' \parallel \mathbf{W}_2 \otimes \mathbf{w}_2')$ with $(\mathbf{w}_1', \mathbf{w}_2')$ and $(\mathbf{A}_1', \mathbf{A}_2')$ are generated via the PRF functions F_{K_1}, F_{K_2} and hash functions $\mathsf{H}_1, \mathsf{H}_2$ respectively (similar ato Eq. 4 and Eq. 5). Otherwise, if $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}, I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$, it outputs

$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = [\![\mathbf{W}\widetilde{\mathbf{f}}^\top - \mu'\mathbf{a}_0^\perp]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$

where $[\![\mathbf{A}_1]\!]_2 \leftarrow \mathbb{G}_2^{k\times|I_{\mathbf{f}_1}|}, [\![\mathbf{A}_2]\!]_2 \leftarrow \mathbb{G}_2^{k'\times|I_{\mathbf{f}_2}|}$, $\widetilde{\mathbf{f}}^\top = \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}$, $\mu' = (\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top - \mu$ with $\mu = (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top$ and $\mathbf{y}_1 \leftarrow \mathbb{G}_1^{|I_{\mathbf{f}_1}|}, \mathbf{y}_2 \leftarrow \mathbb{G}_2^{|I_{\mathbf{f}_2}|}$.

5.3 Security Analysis

Theorem 2. *Assuming the hardness of the bilateral k-*Lin *and k'-*Lin *assumptions, our* $\mathsf{UQFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *scheme is* $\mathsf{SA\text{-}SIM}$ *secure in the random oracle model as per Def. 4. More precisely, if there exists a PPT adversary $\mathcal{A}$ that breaks the* $\mathsf{SA\text{-}SIM}$ *security of our UQFE then we construct PPT machines $\mathcal{B}_1, \mathcal{B}_2$ and $\mathcal{B}_3$ such that for any security parameter λ, the advantage*

$$\mathsf{Adv}^{\mathsf{UQFE}}_{\mathcal{A},\mathsf{SA\text{-}SIM}}(\lambda) \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{MDDH}^{k'+1}_{k',1}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_2}^{bi\text{-}\mathsf{MDDH}^{n_1}_{k,1}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{MDDH}^{n_2}_{k',1}}(\lambda)$$

where (n_1, n_2) is the lengths of challenge message vectors.

Proof. We consider a sequence of games to prove the theorem. Suppose $\mathcal{A}$ be a PPT adversary against $\mathsf{SA\text{-}SIM}$ experiment of our UQFE scheme. The games are described below. In the description of these games, a part framed by a box indicates the elements that are altered in a transition from its previous game.

Game 0: This game corresponds to the experiment $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UQFE},\mathcal{A}}(1^\lambda)$ as defined in Def. 4 where the ciphertext CT^* associated with the vectors pair $(\mathbf{z}_1^*, \mathbf{z}_2^*)$ is generated as

$$\mathsf{CT}^* = \left(\underbrace{[\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1}_{\mathbf{y}_1}, \underbrace{[\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2}_{\mathbf{y}_2}, \underbrace{[\![\mathbf{s}_0\mathbf{A}_0]\!]_1}_{\mathbf{c}_0}, \underbrace{[\![\mathbf{s}_0\mathbf{A}_0\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_2^* \parallel \mathbf{y}_1 \otimes \mathbf{s}_2)]\!]_1}_{\mathbf{y}_0}, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\right).$$

Here, $\mathbf{s}_1 \leftarrow \mathbb{Z}_q^k, \mathbf{s}_2 \leftarrow \mathbb{Z}_q^{k'}$ and $[\![\mathbf{A}_1]\!]_1, [\![\mathbf{A}_2]\!]_2$ are generated similarly as Eq. 4 5. Also, $[\![\mathbf{A}_0\mathbf{W}]\!]_1 := [\![\mathbf{A}_0(\mathbf{W}_1 \otimes \mathbf{w}_1) \parallel \mathbf{A}_0(\mathbf{W}_2 \otimes \mathbf{w}_2)]\!]_1$ with $\mathbf{w}_1 = (F_{K_1}(i_t))_{t\in[n_1]} \in \mathbb{Z}_p^{n_1}$ and $\mathbf{w}_2 = (F_{K_2}(j_t))_{t\in[n_2]} \in \mathbb{Z}_p^{n_2}$. The secret key queried by $\mathcal{A}$ corresponding to the function $\mathbf{f}$ with the index sets $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$ is formed as $\mathsf{SK}_{\mathbf{f}} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ such that

$$\mathsf{SK}_{\mathbf{f}} = \left(\mathbf{k}_1^\top = [\![\mathbf{W}\begin{pmatrix}(\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top\end{pmatrix}]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$

whenever $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}$, $I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$.

Game 1: This game is identical with the Game 0 except that the challenge ciphertext component $\mathbf{y}_0$ is set as

$$\mathsf{CT}^* = \left([\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1, [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2, [\![\mathbf{s}_0\mathbf{A}_0]\!]_1, [\![\mathbf{s}_0\mathbf{A}_0\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_2^* || \mathbf{y}_1 \otimes \mathbf{s}_2)]\!]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\right)$$

where $[\![\mathbf{A}_0\mathbf{W}]\!]_1 = [\![\mathbf{A}_0(\mathbf{W}_1 \otimes \mathbf{w}_1) || \mathbf{A}_0(\mathbf{W}_2 \otimes \mathbf{w}_2)]\!]_1$ such that $\boxed{\mathbf{w}_1 \leftarrow \mathbb{Z}_p^{1\times n_1}, \mathbf{w}_2 \leftarrow \mathbb{Z}_p^{1\times n_2}}$. In this Game, we replace the PRFs $F_{K_1}(\cdot), F_{K_2}(\cdot)$ with the random functions $\mathsf{Rand}_1(\cdot) \leftarrow \mathsf{Rand}_{1,\lambda}, \mathsf{Rand}_2(\cdot) \leftarrow \mathsf{Rand}_{2,\lambda}$ where $\mathsf{Rand}_{1,\lambda}, \mathsf{Rand}_{2,\lambda}$ are the set of functions that have the same domain and range space as F_{K_1} and F_{K_2} respectively. Therefore, from the security of PRF, the Game 0 and Game 1 are computationally indistinguishable.

Game 2: Game 2 is the same as Game 1 except that the challenge ciphertext component $\mathbf{c}_0$ are generated as follows:

$$\mathsf{CT}^* = \left([\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1, [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2, \boxed{[\![\mathbf{u}]\!]_1}, [\![\mathbf{u}\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_1^* || \mathbf{y}_1 \otimes \mathbf{s}_2)]\!]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\right)$$

where $\mathbf{u} \leftarrow \mathbb{Z}_p^{k'+1}$. All others components are generated similarly by $\mathcal{B}$ as Game 1.

We prove the indistinguishability between Game 1 and Game 2 in the Lemma 4.

Game 3: This is exactly the same as Game 2 except the secret key $\mathsf{SK}_{\mathbf{f}}$ for $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}$ and $I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$, and the challenge ciphertext text CT^* are computed as

$$\mathsf{CT}^* = \Big([\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1, [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2, [\![\mathbf{u}]\!]_1, \boxed{[\![\mathbf{u}\widetilde{\mathbf{W}}]\!]_1}, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\Big)$$

$$\mathsf{SK}_{\mathbf{f}} = \Big(\boxed{[\![\widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^\perp(\langle \mathbf{z}^*, \widetilde{\mathbf{f}}\rangle)]\!]_2}, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\Big)$$

where $\widetilde{\mathbf{W}}$ is chosen uniformly from $\mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$. We justify the transition between Game 2 and Game 3 in the Lemma 5.

Game 4: Game 4 is the same as Game 3 except that the secret key component $\mathbf{k}_1^\top$ associated with the function $\mathbf{f} \in \mathbb{Z}_p^{n_1 n_2}$ is generated as

$$\mathsf{SK}_{\mathbf{f}} = \Big(\mathbf{k}_1^\top = [\![\widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^\perp \boxed{\mu'}]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\Big)$$

where $\mu' = (\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top - (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top$ and $\widetilde{\mathbf{f}}^\top = \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}$.

We justify the game transition between Game 3 and Game 4 in the Lemma 6.

Game 5: This game is the same as Game 4 except that we program $[\![\mathbf{A}_1]\!]_1 = [\![\mathbf{a}_{i_1}^{(1)} \parallel \dots \parallel \mathbf{a}_{i_{n_1}}^{(1)}]\!]_1 \in \mathbb{G}_1^{k\times|I_{\mathbf{z}_1^*}|}$ by $\boxed{[\![\mathbf{a}_{i_\ell}^{(1)}]\!]_1 \leftarrow \mathbb{G}_1^{k\times 1}}$ for $I_{\mathbf{z}_1^*} = \{i_1, \dots, i_{n_1}\}$ and the challenge ciphertext as

$$\mathsf{CT}^* = \Big(\boxed{\mathbf{y}_1}, \mathbf{y}_2 = [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2, \mathbf{c}_0 = [\![\mathbf{u}]\!]_1, \mathbf{y}_0 = [\![\mathbf{u}\widetilde{\mathbf{W}}]\!]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\Big)$$

where $\mathbf{y}_1 \leftarrow \mathbb{G}_1^{|I_{\mathbf{z}_1^*}|}$.
We prove the indistinguishability between Game 4 and Game 5 in Lemma 7.

Game 6: This game is the same as Game 5 except that we program $[\![\mathbf{A}_2]\!]_2 = [\![\mathbf{a}_{j_1}^{(2)} \parallel \dots \parallel \mathbf{a}_{j_{n_2}}^{(2)}]\!]_2 \in \mathbb{G}_2^{k'\times|I_{\mathbf{z}_2^*}|}$ by $\boxed{[\![\mathbf{a}_{j_\ell}^{(2)}]\!]_2 \leftarrow \mathbb{G}_2^{k'\times 1}}$ for $I_{\mathbf{z}_2^*} = \{j_1, \dots, j_{n_2}\}$ and the challenge ciphertext is computed as

$$\mathsf{CT}^* = \Big(\mathbf{y}_1, \boxed{\mathbf{y}_2}, \mathbf{c}_0 = [\![\mathbf{u}]\!]_1, \mathbf{y}_0 = [\![\mathbf{u}\widetilde{\mathbf{W}}]\!]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\Big)$$

where $\mathbf{y}_2 \leftarrow \mathbb{G}_2^{|I_{\mathbf{z}_2^*}|}$.

We justify the transition from Game 5 to Game 6 in Lemma 8.

Finally, note that Game 6 is exactly the output of the simulator. We represent E_ι as the event that $\mathcal{A}$ outputs 1 in Game ι. We prove the following lemmas by showing the indistinguishability of adjacent games listed above.

Lemma 4. *For all adversary $\mathcal{A}$, there exist $\mathcal{B}_1$ such that*

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{MDDH}_{k',1}^{k'+1}}(\lambda).$$

Proof. Let us assume that the challenger obtains an instance $(\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{u}_b]\!]_1)$ of $\mathsf{MDDH}_{k',1}^{k'+1}$ assumption where

$$\mathbf{u}_b = \begin{cases} \mathbf{s}_0\mathbf{A}_0 & \text{if } b = 0, \\ \mathbf{u} \leftarrow \mathbb{Z}_p^{k'+1} & \text{if } b = 1. \end{cases}$$

The challenger uses the $\mathsf{MDDH}_{k',1}^{k'+1}$ instance to traverse from Game 2 to Game 1.

Public key simulation. The reduction samples $\mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1)\times k'}$, $\mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1)\times k}$ and sets $\mathsf{PP} = (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{A}_0\mathbf{W}_1]\!]_1, [\![\mathbf{A}_0\mathbf{W}_2]\!]_1)$.

Ciphertext simulation. The challenge ciphertext $\mathsf{CT}^* = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{c}_0, \mathbf{y}_0, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*})$ corresponding to the challenge message $(\mathbf{z}_1^*, \mathbf{z}_2^*)$ is generated as follows:

- Sample $\mathbf{s}_1 \leftarrow \mathbb{Z}_p^k$ and $\mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k'}$ and generate $\mathbf{y}_1 = [\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1, \mathbf{y}_2 = [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2$, $\mathbf{c}_0 = [\![\mathbf{u}_b]\!]_1$ where

$$\begin{aligned} [\![\mathbf{A}_1]\!]_1 =& [\![\mathbf{a}_{i_1}^{(1)} \parallel \dots \parallel \mathbf{a}_{i_{n_1}}^{(1)}]\!]_1 \in \mathbb{G}_1^{k\times n_1}, \\ [\![\mathbf{A}_2]\!]_2 =& [\![\mathbf{a}_{j_1}^{(2)} \parallel \dots \parallel \mathbf{a}_{j_{n_2}}^{(2)}]\!]_2 \in \mathbb{G}_2^{k'\times n_2} \end{aligned}$$

  and $\mathsf{H}_1(i_\ell) = \left([\![\mathbf{a}_{i_\ell}^{(1)}]\!]_1, [\![\mathbf{a}_{i_\ell}^{(1)}]\!]_2\right) \in \mathbb{G}_1^{k\times 1} \times \mathbb{G}_2^{k\times 1}$ for all $\ell \in [n_1]$ and $\mathsf{H}_2(j_\ell) = [\![\mathbf{a}_{j_\ell}^{(2)}]\!]_2 \in \mathbb{G}_2^{k'\times 1}$ for all $\ell \in [n_2]$.
- Compute $\mathbf{y}_0 = [\![\mathbf{u}_b\mathbf{W} + (\mathbf{s}_1 \otimes \mathbf{z}_2^* \parallel \mathbf{y}_1 \otimes \mathbf{s}_2)]\!]_1$ where $[\![\mathbf{u}_b\mathbf{W}]\!]_1 \coloneqq [\![\mathbf{u}_b\widetilde{\mathbf{W}}_1 \parallel \mathbf{u}_b\widetilde{\mathbf{W}}_2]\!]_1$ and $\widetilde{\mathbf{W}}_1 = \mathbf{W}_1 \otimes \mathbf{w}_1 \in \mathbb{Z}_p^{(k'+1)\times k'n_1}$, $\widetilde{\mathbf{W}}_2 = \mathbf{W}_2 \otimes \mathbf{w}_2 \in \mathbb{Z}_p^{(k'+1)\times kn_2}$ with $\mathbf{w}_1 \leftarrow \mathbb{Z}_p^{n_1}$, $\mathbf{w}_2 \leftarrow \mathbb{Z}_p^{n_2}$.

Secret key simulation. In the following, we describe how challenger simulates the secret key $\mathsf{SK}_\mathbf{f} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ associated with the function $\mathbf{f} \in \mathbb{Z}_p^{n_1n_2}$.

- Using the hash functions $\mathsf{H}_1, \mathsf{H}_2$ over the index sets $I_{\mathbf{f}_1}$, $I_{\mathbf{f}_2}$, generate the matrices $[\![\mathbf{A}_1]\!]_2 \in \mathbb{G}_2^{k\times n_1}$, $[\![\mathbf{A}_2]\!]_2 \in \mathbb{G}_2^{k'\times n_2}$ as described in ciphertext simulation phase.
- Generate the secret key component $\mathbf{k}_1^\top$ corresponding to the function $\mathbf{f} \in Z_p^{n_1n_2}$ as

$$\mathbf{k}_1^\top = [\![\mathbf{W}\begin{pmatrix}(\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top\end{pmatrix}]\!]_2$$

  where $\mathbf{W} \coloneqq (\mathbf{W}_1 \otimes \mathbf{w}_1 \parallel \mathbf{W}_2 \otimes \mathbf{w}_2)$ such that $\mathbf{w}_1 \leftarrow \mathbb{Z}_p^{n_1}$, $\mathbf{w}_2 \leftarrow \mathbb{Z}_p^{n_2}$.

Analysis. According to the $\mathsf{MDDH}_{k',1}^{k'+1}$ assumption, we have

$$(\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{s}_0\mathbf{A}_0]\!]_1) \approx_c (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![u]\!]_1).$$

If $b = 0$, $\mathbf{u}_b = [\![\mathbf{s}_0\mathbf{A}_0]\!]_1$, then the adversarial view is the same as Game 1; otherwise for $b = 1$, $\mathbf{u}_b$ is randomly chosen from the group $\mathbb{G}_1^{k'+1}$ and hence the adversarial view is similar to Game 2. Thus, we have Game 1 $\approx_c$ Game 2 via the $\mathsf{MDDH}_{k',1}^{k'+1}$ assumption. □

Lemma 5. *For all adversary $\mathcal{A}$, we have $\mathsf{E}_3 \approx_s \mathsf{E}_2$.*

Proof. The change from Game 2 to Game 3 follows from the following change of variables which embeds the selective challenge $\mathbf{z}^*$ into $\mathbf{W}$:

$$\widetilde{\mathbf{W}} = \mathbf{W} + \mathbf{a}_0^{\perp}\mathbf{z}^*$$

where $\mathbf{a}_0^{\perp} \in \mathbb{Z}_p^{(k'+1)}$ such that $\mathbf{A}_0 \cdot \mathbf{a}_0^{\perp} = 0, \mathbf{u} \cdot \mathbf{a}_0^{\perp} = 1$. Since $\mathbf{W}$ is chosen uniformly from $\mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$, then the matrix $\widetilde{\mathbf{W}}$ is also uniform over $\mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$.

Let us denote $(\mathbf{z}^*)^\top = (\mathbf{s}_1 \otimes \mathbf{z}_1^* || \mathbf{y}_1 \otimes \mathbf{s}_2) \in \mathbb{Z}_p^{k'n_1+kn_2}$. We have

$$\widetilde{\mathbf{W}}_1 = \mathbf{W}_1 \otimes \mathbf{w}_1^*;\ \widetilde{\mathbf{W}}_2 = \mathbf{W}_2 \otimes \mathbf{w}_2^*;\ \mathbf{W} = (\widetilde{\mathbf{W}}_1 || \widetilde{\mathbf{W}}_2) \in \mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$$

which in particular implies that

$$\begin{aligned}
\mathbf{A}_0\widetilde{\mathbf{W}} &= \mathbf{A}_0\mathbf{W} + \mathbf{A}_0\mathbf{a}_0^{\perp}\mathbf{z} = \mathbf{A}_0\mathbf{W} = (\mathbf{A}_0\widetilde{\mathbf{W}}_1 || \mathbf{A}_0\widetilde{\mathbf{W}}_2) && (7)\\
\mathbf{u}\widetilde{\mathbf{W}} &= \mathbf{u}\mathbf{W} + \mathbf{u}(\mathbf{a}_0^{\perp}\mathbf{z}^*) = \mathbf{u}\mathbf{W} + \mathbf{z}^* && (8)\\
\widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top &= \mathbf{W}\widetilde{\mathbf{f}}^\top + \mathbf{a}_0^{\perp}(\mathbf{z}^*\widetilde{\mathbf{f}}^\top) = \mathbf{W}\widetilde{\mathbf{f}}^\top + \mathbf{a}_0^{\perp}(\langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle)\\
\mathbf{W}\widetilde{\mathbf{f}}^\top &= \widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^{\perp}(\langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle) = \mathbf{k}_1^\top && (9)\\
\text{where } \widetilde{\mathbf{f}}^\top &= \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}
\end{aligned}$$

Formally to justify the change of variables, observe that for all $\mathbf{A}_0, \mathbf{z}^*$, we have

$$\left(\mathbf{A}_0\mathbf{W}, \mathbf{W} + \mathbf{a}_0^{\perp}\mathbf{z}^*\right) \equiv \left(\mathbf{A}_0\widetilde{\mathbf{W}}, \widetilde{\mathbf{W}}\right) \tag{10}$$

where the distributions are taken over the random choice of $\mathbf{W}$. Then, whenever $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}$ and $I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$ we have

$$\begin{aligned}
\mathsf{CT}^* &= \left([\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]_1, [\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]_2, [\mathbf{u}]_1, [\mathbf{u}\widetilde{\mathbf{W}}]_1, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*}\right),\\
\mathsf{SK}_{\mathbf{f}} &= \left(\llbracket \widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^{\perp}(\langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle) \rrbracket_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right).
\end{aligned}$$

If $I_{\mathbf{f}_1} \neq I_{\mathbf{z}_1^*}$ or $I_{\mathbf{f}_2} \neq I_{\mathbf{z}_2^*}$ then the secret key corresponding to the vector $\mathbf{f}$ is generated as

$$\mathsf{SK}_{\mathbf{f}} = \left(\mathbf{k}_1^\top = \llbracket \mathbf{W}' \begin{pmatrix} (\mathbf{A}_1' \otimes \mathbf{I}_{n_2'})\mathbf{f}^\top \\ (\mathbf{I}_{n_1'} \otimes \mathbf{A}_2')\mathbf{f}^\top \end{pmatrix} \rrbracket_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$

where $\llbracket \mathbf{A}_1' \rrbracket_2 \in \mathbb{G}_2^{k\times n_1'}$, $\llbracket \mathbf{A}_2' \rrbracket_2 \in \mathbb{G}_2^{k'\times n_2'}$ are generated by using the hash functions $\mathsf{H}_1, \mathsf{H}_2$ over the index sets $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$ and $\mathbf{W} := (\mathbf{W}_1 \otimes \mathbf{w}_1 \parallel \mathbf{W}_2 \otimes \mathbf{w}_2)$ with $\mathbf{w}_1 = (F_{K_1}(i_t))_{t\in[n_1']} \in \mathbb{Z}_p^{n_1'}, \mathbf{w}_2 = (F_{K_2}(j_t))_{t\in[n_2']} \in \mathbb{Z}_p^{n_2'}$.

So from the above, we can conclude that the distribution of Eq. 10 are identically distributed even if $\mathbf{z}^*$ is selectively chosen. Therefore, Game 2 and Game 3 are statistically indistinguishable.

Lemma 6. *For all adversary $\mathcal{A}$, we have* $\Pr[\mathsf{E}_4] = \Pr[\mathsf{E}_3]$.

Proof. The secret key $\mathsf{SK}_{\mathbf{f}} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ for $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}$ and $I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$ is given by the component

$$\mathbf{k}_1^\top = \widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^\perp \langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle.$$

To see that the Game 3 and Game 4 follow the same distribution, we use the following identity

$$\begin{aligned}(\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top &= (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top + (\mathbf{s}_1 \otimes \mathbf{z}_1^* || \mathbf{y}_1 \otimes \mathbf{s}_2) \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix} \\ &= (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top + \langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle \\ \implies \langle \mathbf{z}^*, \widetilde{\mathbf{f}} \rangle &= (\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top - (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top = \mu'.\end{aligned}$$

It is now clear that the distribution of secret keys in Game 3 and Game 4 are identical. Hence, Game 3 and Game 4 are identically in the adversary's view. □

Lemma 7. *For all adversary $\mathcal{A}$, there exist $\mathcal{B}_2$ in the random oracle model such that*

$$|\Pr[\mathsf{E}_5] - \Pr[\mathsf{E}_4]| \le \mathsf{Adv}_{\mathcal{B}_2}^{bi\text{-}\mathsf{MDDH}_{k,1}^{n_1}}(\lambda).$$

Proof. We will now show that the challenger $\mathcal{B}_2$ can break the bi-$\mathsf{MDDH}_{k,1}^{n_1}$ assumption using $\mathcal{A}$ as a subroutine. The adversary $\mathcal{B}_2$ obtains the instances $(\mathsf{G}, [\![\mathbf{A}_1]\!]_1, [\![\mathbf{A}_1]\!]_2, [\![\mathbf{t}_b]\!]_1, [\![\mathbf{t}_b]\!]_2)$ of bi-$\mathsf{MDDH}_{k,1}^{n_1}$ assumption where

$$\mathbf{t}_b = \begin{cases} \mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^* & \text{if } b = 0, \\ \mathbf{t} \leftarrow \mathbb{Z}_p^{n_1} & \text{if } b = 1. \end{cases}$$

We have to show that $\mathcal{B}_2$ can interpolate between Game 4 and Game 5 by using the bi-$\mathsf{MDDH}_{k,1}^{n_1}$ instances.

Public parameter simulation. Samples $\mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1)\times k'}$, $\mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1)\times k}$ and $\mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k'\times(k'+1)}$ generate $\mathsf{PP} = (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{A}_0\mathbf{W}_1]\!]_1, [\![\mathbf{A}_0\mathbf{W}_2]\!]_1)$.

Random oracle simulation. When the adversary $\mathcal{A}$ gives out $I_{\mathbf{z}_1^*}$ and $I_{\mathbf{z}_2^*}$, we initiate the random oracle $\mathsf{H}_1(i) = ([\![\mathbf{a}_i^{(1)}]\!]_1, [\![\mathbf{a}_i^{(1)}]\!]_2)$ for all $i \in I_{\mathbf{z}_1^*}$ where $\mathbf{a}_i^{(1)}$ is the i^{th} column of $\mathbf{A}_1$ and $\mathsf{H}_2 = \phi$. When the adversary $\mathcal{A}$ makes random oracle query on j,

- on H_1: if $\mathsf{H}_1(j)$ is empty, we sample $\mathbf{u}_j \leftarrow \mathbb{Z}_p^k$ and assign $\mathsf{H}_1(j) = ([\![\mathbf{u}_j^{(1)}]\!]_1, [\![\mathbf{u}_j^{(1)}]\!]_2)$ before sending $\mathsf{H}_1(j)$ back. Otherwise, we just send $\mathsf{H}_1(j)$ back.
- on H_2: if $\mathsf{H}_2(j)$ is empty, we sample $\mathbf{a}_j \leftarrow \mathbb{Z}_p^{k'}$ and assign $\mathsf{H}_2(j) = ([\![\mathbf{a}_j^{(2)}]\!]_2)$ before sending $\mathsf{H}_2(j)$ back. Otherwise, we just send $\mathsf{H}_2(j)$ back.

Ciphertext simulation. The challenger simulates the challenge ciphertext $\mathsf{CT}^* = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{c}_0, \mathbf{y}_0, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*})$ as follows:

- Sample $\mathbf{s}_2 \leftarrow \mathbb{Z}_p^{k'}, \mathbf{u} \leftarrow \mathbb{Z}_p^{k'}$ and set $\mathbf{y}_1 = [\![\mathbf{t}_b]\!]_1, \mathbf{y}_2 = [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2$, $\mathbf{c}_0 = [\![\mathbf{u}]\!]_1$, $\mathbf{y}_0 = [\![\mathbf{u}\widetilde{\mathbf{W}}]\!]_1$ where
$$[\![\mathbf{A}_2]\!]_2 =[\![\mathbf{a}_{j_1}^{(2)} \parallel \ \dots \ \parallel \mathbf{a}_{j_{n_2}}^{(2)}]\!]_2 \in \mathbb{G}_2^{k' \times n_2}$$
such that $[\![\mathbf{a}_{j_\ell}^{(2)}]\!]_2 \leftarrow \mathbb{G}_2^{k' \times 1}$ for all $\ell \in [n_2]$, and $\widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$.

Secret key simulation. In the following, we describe how the challenger simulates the secret key $\mathsf{SK}_\mathbf{f}$ using the given instance whenever $I_{\mathbf{f}_1} = I_{\mathbf{z}_1^*}, I_{\mathbf{f}_2} = I_{\mathbf{z}_2^*}$.

- Generate the secret key $\mathsf{SK}_\mathbf{f} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ corresponding to the secret key vector $\mathbf{f} \in \mathbb{Z}_p^{n_1 n_2}$ as
$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = [\widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^\perp \mu']_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$
where $\mu' = (\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top - (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top$ and $\widetilde{\mathbf{f}}^\top = \begin{pmatrix} (\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top \end{pmatrix}$.

If $I_{\mathbf{f}_1} \neq I_{\mathbf{z}_1^*}$ or $I_{\mathbf{f}_2} \neq I_{\mathbf{z}_2^*}$ then the secret key $\mathsf{SK}_\mathbf{f} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ corresponding to the secret key vector $\mathbf{f}$ is generated as
$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = \left[\mathbf{W}' \begin{pmatrix} (\mathbf{A}_1' \otimes \mathbf{I}_{n_2'})\mathbf{f}^\top \\ (\mathbf{I}_{n_1'} \otimes \mathbf{A}_2')\mathbf{f}^\top \end{pmatrix}\right]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$
where $[\![\mathbf{A}_1']\!]_2 \in \mathbb{G}_2^{k \times n_1'}$ and $[\![\mathbf{A}_2']\!]_2 \in \mathbb{G}_2^{k' \times n_2'}$ are computed honestly by using the hash functions H_1 and H_2 over the index set $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$ and $\mathbf{W} := (\mathbf{W}_1 \otimes \mathbf{w}_1 \parallel \mathbf{W}_2 \otimes \mathbf{w}_2)$ with $\mathbf{w}_1 = (F_{K_1}(i_t))_{t \in [n_1']} \in \mathbb{Z}_p^{n_1'}, \mathbf{w}_2 = (F_{K_2}(j_t))_{t \in [n_2']} \in \mathbb{Z}_p^{n_2'}$.

Analysis. According to the security of bilateral k-Lin assumption, we have
$$([\![\mathbf{A}_1]\!]_1, [\![\mathbf{A}_1]\!]_2, [\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_1, [\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]_2) \approx_c ([\![\mathbf{A}_1]\!]_1, [\![\mathbf{A}_1]\!]_2, [\![\mathbf{t}]\!]_1, [\![\mathbf{t}]\!]_2) \, .$$
In that case, for $b = 0$, i.e., $\mathbf{t}_b = [\![\mathbf{s}_1\mathbf{A}_1 + \mathbf{z}_1^*]\!]$, then $\mathcal{B}_3$ simulates the Game 4; otherwise for $b = 1$, $\mathbf{t}_b = \mathbf{t}$ is uniformly chosen from the group $\mathbb{Z}_p^{n_1}$ and hence $\mathcal{B}$ simulates the Game 5. Therefore, we can conclude that Game 4 $\approx_c$ Game 5, i.e., Game 4 and Game 5 are computationally indistinguishable. □

Lemma 8. *For all adversary $\mathcal{A}$, there exist $\mathcal{B}_3$ in the random oracle model such that*
$$|\Pr[\mathsf{E}_6] - \Pr[\mathsf{E}_5]| \leq \mathsf{Adv}_{\mathcal{B}_3}^{\mathsf{MDDH}_{k',1}^{n_2}}(\lambda).$$

Proof. We will now show that the challenger $\mathcal{B}_3$ can break the $\mathsf{MDDH}_{k',1}^{n_2}$ problem using $\mathcal{A}$ as a subroutine. The adversary $\mathcal{B}_3$ obtains the instance $(\mathsf{G}, [\![\mathbf{A}_2]\!]_2, [\![\mathbf{t'}_b]\!]_2)$ of $\mathsf{MDDH}_{k',1}^{n_2}$ problem where
$$\mathbf{t'}_b = \begin{cases} \mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^* & \text{if } b = 0, \\ \mathbf{t'} \leftarrow \mathbb{Z}_p^{n_2} & \text{if } b = 1 \end{cases}$$

by using the instance of $\mathsf{MDDH}^{n_2}_{k',1}$, the challenger $\mathcal{B}_3$ can interpolate between Game 5 and Game 6.

Public parameter simulation. Sample $\mathbf{W}_1 \leftarrow \mathbb{Z}_p^{(k'+1)\times k'}$, $\mathbf{W}_2 \leftarrow \mathbb{Z}_p^{(k'+1)\times k}$ and $\mathbf{A}_0 \leftarrow \mathbb{Z}_p^{k'\times(k'+1)}$ to generate the public parameter $\mathsf{PP} = (\mathsf{G}, [\![\mathbf{A}_0]\!]_1, [\![\mathbf{A}_0\mathbf{W}_1]\!]_1, [\![\mathbf{A}_0\mathbf{W}_2]\!]_1)$.

Random oracle simulation. When the adversary $\mathcal{A}$ gives out $I_{\mathbf{z}_1^*}$ and $I_{\mathbf{z}_2^*}$, we initiate the random oracles $\mathsf{H}_1 = \phi$ and $\mathsf{H}_2(i) = ([\![\mathbf{a}_i^{(2)}]\!]_2)$ for all $i \in I_{\mathbf{z}_2^*}$ where $\mathbf{a}_i^{(2)}$ is the i^{th} column of $\mathbf{A}_2$. When the adversary $\mathcal{A}$ makes random oracle query on j,

- on H_1: if $\mathsf{H}_1(j)$ is empty, we sample $\mathbf{a}_j^{(1)} \leftarrow \mathbb{Z}_p^k$ and assign $\mathsf{H}_1(j) = ([\![\mathbf{a}_j^{(1)}]\!]_1, [\![\mathbf{a}_j^{(1)}]\!]_2)$ before sending $\mathsf{H}_1(j)$ back. Otherwise, we just send $\mathsf{H}_1(j)$ back.
- on H_2: if $\mathsf{H}_2(j)$ is empty, we sample $\mathbf{d}_j \leftarrow \mathbb{Z}_p^{k'}$ and assign $\mathsf{H}_2(j) = ([\![\mathbf{d}_j^{(1)}]\!]_2)$ before sending $\mathsf{H}_2(j)$ back. Otherwise, we just send $\mathsf{H}_2(j)$ back.

Ciphertext simulation. Now $\mathcal{B}_3$ simulates the challenge ciphertext $\mathsf{CT}^* = (\mathbf{y}_1, \mathbf{y}_2, \mathbf{c}_0, \mathbf{y}_0, I_{\mathbf{z}_1^*}, I_{\mathbf{z}_2^*})$ as follows:

- Sample $\mathbf{u} \leftarrow \mathbb{Z}_p^{k'}$, $\mathbf{y}_1 \leftarrow \mathbb{G}^{n_1}$ and sets $\mathbf{y}_2 = [\![\mathbf{t'}_b]\!]_2$, $\mathbf{c}_0 = [\![\mathbf{u}]\!]_1$, $\mathbf{y}_0 = [\![\mathbf{u}\widetilde{\mathbf{W}}]\!]_1$ where $\widetilde{\mathbf{W}} \leftarrow \mathbb{Z}_p^{(k'+1)\times(k'n_1+kn_2)}$.

Secret key simulation. In the following, we describe how $\mathcal{B}_3$ simulates the secret key $\mathsf{SK}_\mathbf{f}$ using the given instance whenever $I_{\mathbf{z}_1^*} = I_{\mathbf{f}_1}$ and $I_{\mathbf{z}_2^*} = I_{\mathbf{f}_2}$:

- Generate the secret key $\mathsf{SK}_\mathbf{f} = (\mathbf{k}_1^\top, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2})$ corresponding to the secret key vector $\mathbf{f} \in \mathbb{Z}_p^{n_1n_2}$ as
$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = [\![\widetilde{\mathbf{W}}\widetilde{\mathbf{f}}^\top - \mathbf{a}_0^\perp \mu']\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$
where $\mu' = (\mathbf{y}_1 \otimes \mathbf{y}_2)\mathbf{f}^\top - (\mathbf{z}_1^* \otimes \mathbf{z}_2^*)\mathbf{f}^\top$, $\widetilde{\mathbf{f}}^\top = \begin{pmatrix}(\mathbf{A}_1 \otimes \mathbf{I}_{n_2})\mathbf{f}^\top \\ (\mathbf{I}_{n_1} \otimes \mathbf{A}_2)\mathbf{f}^\top\end{pmatrix}$ and $[\![\mathbf{A}_1]\!]_2 \leftarrow \mathbb{G}_2^{k\times|I_{\mathbf{f}_1}|}$.

If $I_{\mathbf{f}_1} \neq I_{\mathbf{z}_1^*}$ or $I_{\mathbf{f}_2} \neq I_{\mathbf{z}_2^*}$ then the secret key corresponding to the vector $\mathbf{f}$ is generated as

$$\mathsf{SK}_\mathbf{f} = \left(\mathbf{k}_1^\top = [\![\mathbf{W}' \begin{pmatrix}(\mathbf{A}'_1 \otimes \mathbf{I}_{n'_2})\mathbf{f}^\top \\ (\mathbf{I}_{n'_1} \otimes \mathbf{A}'_2)\mathbf{f}^\top\end{pmatrix}]\!]_2, \mathbf{f}, I_{\mathbf{f}_1}, I_{\mathbf{f}_2}\right)$$

where $[\![\mathbf{A}'_1]\!]_2 = [\![\mathbf{a}_{i_1}^{(1)} \parallel \ldots \parallel \mathbf{a}_{i_{n_1}}^{(1)}]\!]_2 \in \mathbb{G}_2^{k\times n'_1}$ and $[\![\mathbf{A}'_2]\!]_2 = [\![\mathbf{a}_{j_1}^{(2)} \parallel \ldots \parallel \mathbf{a}_{j_{n_2}}^{(2)}]\!]_2 \in \mathbb{G}_2^{k'\times n_2}$ such that $\mathsf{H}_1(i_\ell) = \left([\![\mathbf{a}_{i_\ell}^{(1)}]\!]_1, [\![\mathbf{a}_{i_\ell}^{(1)}]\!]_2\right) \in \mathbb{G}_1^{k\times 1} \times \mathbb{G}_2^{k\times 1}$ for all $\ell \in [n'_1]$ and $\mathsf{H}_2(j_\ell) = [\![\mathbf{a}_{j_\ell}^{(2)}]\!]_2 \in \mathbb{G}_2^{k'\times 1}$ for all $\ell \in [n'_2]$ and $\mathbf{W}' := (\mathbf{W}_1 \otimes \mathbf{w}'_1 \parallel \mathbf{W}_2 \otimes \mathbf{w}'_2)$ with $\mathbf{w}'_1 = (F_{K_1}(i_t))_{t\in[n'_1]} \in \mathbb{Z}_p^{n'_1}, \mathbf{w}'_2 = (F_{K_2}(j_t))_{t\in[n'_2]} \in \mathbb{Z}_p^{n'_2}$.

Analysis. According to the security of k'-Lin assumption, we have

$$([\![\mathbf{A}_2]\!]_2, [\![\mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*]\!]_2) \approx_c ([\![\mathbf{A}_2]\!]_2, [\![\mathbf{y}_2]\!]_2)\,.$$

In that case, if $b = 0$, then $\mathbf{t'}_b = \mathbf{s}_2\mathbf{A}_2 + \mathbf{z}_2^*$, then $\mathcal{B}_3$ simulates the Game 5; otherwise for $b = 1$, $\mathbf{t'}_b = \mathbf{t'}$ is uniformly chosen from $\mathbb{Z}_p^{n_2}$ and hence $\mathcal{B}_3$ simulates the Game 6. Therefore, we can conclude that Game 5 $\approx_c$ Game 6, i.e., Game 5 and Game 6 are computationally indistinguishable. □

6 Our Weak Attribute-Hiding UNP-IPFE

We consider a $\mathsf{UQFE} = (\mathsf{UQFE.Setup}, \mathsf{UQFE.Enc}, \mathsf{UQFE.KeyGen}, \mathsf{UQFE.Dec})$ and a $\mathsf{UIPFE} = (\mathsf{UIPFE.Setup}, \mathsf{UIPFE.Enc}, \mathsf{UIPFE.KeyGen}, \mathsf{UIPFE.Dec})$ scheme to construct an $\mathsf{UNP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ scheme. Recall that a UNP-IPFE computes ciphertexts for vectors $(\mathbf{x}, \mathbf{w})$ and generate secret keys for vectors $(\mathbf{y}, \mathbf{v})$ such that decryption algorithm recovers $\langle \mathbf{x}, \mathbf{y} \rangle$ if $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$ and the vectors satisfy a strict/permissive relation, i.e., $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_s$ or $\mathcal{R}_p$. Depending on the underlying UQFE and UIPFE, our generic construction yields a permissive or strict UNP-IPFE. Here, we present the strict case where the correctness holds only when $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_s$.

We slightly modify the Dec algorithms of UIPFE and UQFE schemes, which now return the inner product and quadratic values in the exponent of the underlying target group (that is, before solving the discrete logarithm problem). As with all pairing-based IPFE in the literature, our required inner product value comes from a polynomial range so that we can efficiently perform an exhaustive search to obtain the value at the end of the decryption phase.

6.1 Construction

Our $\mathsf{UNP\text{-}IPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ scheme works as follows:

$\mathsf{\mathbf{Setup}}(1^\lambda) \rightarrow (\mathsf{\mathbf{MPK}}, \mathsf{\mathbf{MSK}})$: The setup algorithm takes input the security parameter λ and performs the following steps:

1. Generate

$$(\mathsf{UQFE.MPK},\ \mathsf{UQFE.MSK}) \leftarrow \mathsf{UQFE.Setup}(1^\lambda)$$
$$(\mathsf{UIPFE.MPK},\ \mathsf{UIPFE.MSK}) \leftarrow \mathsf{UIPFE.Setup}(1^\lambda).$$

2. Set $\mathsf{MSK} = (\mathsf{UQFE.MSK}, \mathsf{UIPFE.MSK}\ )$ and $\mathsf{MPK} = (\mathsf{UQFE.MPK},\ \mathsf{UIPFE.MPK})$

$\mathsf{\mathbf{Enc}}(\mathsf{\mathbf{MPK}}, \mathbf{x}, \mathbf{w}) \rightarrow \mathsf{\mathbf{CT}}_{\mathbf{x},\mathbf{w}}$: The encryption algorithm takes input the master key MPK, message-attribute vector pair $(\mathbf{x}, \mathbf{w}) \in \mathbb{Z}^{|I_\mathbf{x}|} \times \mathbb{Z}^{|I_\mathbf{w}|}$ with the associated index sets $I_\mathbf{x}, I_\mathbf{w}$ and executes the following steps:

1. Parse $\mathsf{MPK} = (\mathsf{UQFE.MPK}, \mathsf{UIPFE.MPK}\ )$.
2. Compute

$$\mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}} \leftarrow \mathsf{UQFE.Enc}(\mathsf{UQFE.MPK}, \mathbf{x}, \mathbf{w})$$
$$\mathsf{UIPFE.CT}_{\mathbf{w}} \leftarrow \mathsf{UIPFE.Enc}(\mathsf{UIPFE.MPK}, \mathbf{w}).$$

3. Output $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}}, \mathsf{UIPFE.CT}_{\mathbf{w}})$.

$\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v}) \rightarrow \mathsf{SK}_{\mathbf{y},\mathbf{v}}$: The key generation algorithm takes input the master public key MPK, the master secret key MSK, key-predicate vector pair $(\mathbf{y}, \mathbf{v}) \in \mathbb{Z}^{|I_{\mathbf{y}}|} \times \mathbb{Z}^{|I_{\mathbf{v}}|}$ with the associated index sets $I_{\mathbf{y}}, I_{\mathbf{v}}$ and performs the following steps:

1. Parse $\mathsf{MSK} = (\mathsf{UQFE.MSK}, \mathsf{UIPFE.MSK})$ and $\mathsf{MPK} = (\mathsf{UQFE.MPK}, \mathsf{UIPFE.MPK})$.
2. Compute

$$\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}} \leftarrow \mathsf{UQFE.KeyGen}(\mathsf{UQFE.MPK}, \mathsf{UQFE.MSK}, \mathbf{y} \otimes \mathbf{v})$$
$$\mathsf{UIPFE.SK}_{\mathbf{v}} \leftarrow \mathsf{UIPFE.KeyGen}(\mathsf{UIPFE.MPK}, \mathsf{UIPFE.MSK}, \mathbf{v}).$$

3. Output $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}_{\mathbf{v}})$.

$\mathsf{Dec}(\mathsf{MPK}, \mathsf{SK}_{\mathbf{y},\mathbf{v}}, \mathsf{CT}_{\mathbf{x},\mathbf{w}}) \rightarrow d/\perp$: The decryptor takes as input the master public key MPK, a ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}}$ for the associated vectors $\mathbf{x}, \mathbf{w}$ with the index sets $I_{\mathbf{x}}, I_{\mathbf{w}}$ and a secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}}$ corresponding to the vectors $\mathbf{y}, \mathbf{v}$ with index sets $I_{\mathbf{y}}$, $I_{\mathbf{v}}$ respectively. Then the decryption algorithm runs the following steps:

1. Parse $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}_{\mathbf{v}})$.
2. Parse $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}}, \mathsf{UIPFE.CT}_{\mathbf{w}})$.
3. If $(\mathbf{x}, \mathbf{y}) \notin \mathcal{R}_s$ or $(\mathbf{w}, \mathbf{v}) \notin \mathcal{R}_s$, return $\perp$.
4. Else, compute

$$\zeta \leftarrow \mathsf{UQFE.Dec}(\mathsf{UQFE.MPK}, \mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}})$$
$$\eta \leftarrow \mathsf{UIPFE.Dec}(\mathsf{UIPFE.MPK}, \mathsf{UIPFE.SK}_{\mathbf{v}}, \mathsf{UIPFE.CT}_{\mathbf{w}}).$$

5. Output $\log_\eta \zeta$.

Correctness. Let the ciphertext $\mathsf{CT}_{\mathbf{x},\mathbf{w}} = (\mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}}, \mathsf{UIPFE.CT}_{\mathbf{w}})$ be computed for a pair of vectors $\mathbf{x} = (x_i)_{i\in I_{\mathbf{x}}} \in \mathbb{Z}^{|I_{\mathbf{x}}|}, \mathbf{w} = (w_j)_{j\in I_{\mathbf{w}}} \in \mathbb{Z}^{|I_{\mathbf{w}}|}$ and the secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}_{\mathbf{v}})$ be generated for a pair of vectors $\mathbf{y} = (y_i)_{i\in I_{\mathbf{y}}} \in \mathbb{Z}^{|I_{\mathbf{y}}|}, \mathbf{v} = (v_j)_{j\in I_{\mathbf{v}}} \in \mathbb{Z}^{|I_v|}$. If $R(\mathbf{w}, \mathbf{v}) = 1$, i.e., $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$ with $(\mathbf{x}, \mathbf{y}), (\mathbf{w}, \mathbf{v}) \in \mathcal{R}_s$, then we have

$$\mathsf{UQFE.Dec}(\mathsf{UQFE.MPK}, \mathsf{UQFE.SK}_{\mathbf{f}}, \mathsf{UQFE.CT}_{\mathbf{x},\mathbf{w}}) = [\![\langle \mathbf{x}, \mathbf{y} \rangle \langle \mathbf{w}, \mathbf{v} \rangle]\!]_T = \zeta$$
$$\mathsf{UIPFE.Dec}(\mathsf{UIPFE.MPK}, \mathsf{UIPFE.SK}_{\mathbf{v}}, \mathsf{UIPFE.CT}_{\mathbf{w}}) = [\![\langle \mathbf{w}, \mathbf{v} \rangle]\!]_T = \eta.$$

Since $\langle \mathbf{w}, \mathbf{v} \rangle \neq 0$, the correctness follows as one can compute $\log_\eta \zeta = \langle \mathbf{x}, \mathbf{y} \rangle$ by performing an exhaustive search over a polynomial range where $\langle \mathbf{x}, \mathbf{y} \rangle$ belongs.

Remark 2. *In this paper, we consider* IND*-based security for our UNP-IPFE in the public key setting and* SIM*-based security in the secret key setting. We present two instantations accordingly and compare the efficiency of the concrete schemes. For now, we prove the* SIM*-based security of our UNP-IPFE in the secret key setting. The* IND*-based security can be proved similarly, however, for completeness we provide the security analysis of the public key version in Appendix A.*

6.2 Simulator

We present the PPT simulator of our secret key UNP-IPFE scheme in the $\mathsf{SA\text{-}WAH\text{-}SIM}$ security model. Let $\mathcal{S} := (\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$ be a PPT simulator for our UNP-IPFE scheme and also let $\mathcal{S}_1 := \mathsf{UQFE}.(\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$, $\mathcal{S}_2 := \mathsf{UIPFE}.(\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$ be the PPT simulators for the $\mathsf{SA\text{-}SIM}$ simulation secure UQFE and UIPFE respectively.

$\mathsf{\textbf{Setup}}^*(\mathbf{1}^{\boldsymbol{\lambda}})$: Run

$$(\mathsf{UQFE.PP}^*, \mathsf{UQFE.MSK}^*) \leftarrow \mathsf{UQFE.Setup}^*(1^\lambda),$$
$$(\mathsf{UIPFE.PP}^*, \mathsf{UIPFE.MSK}^*) \leftarrow \mathsf{UIPFE.Setup}^*(1^\lambda)$$

and output $\mathsf{PP}^* = (\mathsf{UQFE.PP}^*, \mathsf{UIPFE.PP}^*)$; $\mathsf{MSK}^* = (\mathsf{UQFE.MSK}^*, \mathsf{UIPFE.MSK}^*)$.

$\mathsf{\textbf{Enc}}^*(\mathsf{\textbf{PP}}^*, \mathsf{\textbf{MSK}}^*, \boldsymbol{I}_{\mathbf{x}^*}, \boldsymbol{I}_{\mathbf{w}^*})$: Compute

$$\mathsf{UQFE.CT}^* \leftarrow \mathsf{UQFE.Enc}^*(\mathsf{UQFE.PP}^*, \mathsf{UQFE.MSK}^*, I_{\mathbf{x}^*}, I_{\mathbf{w}^*}),$$
$$\mathsf{UIPFE.CT}^* \leftarrow \mathsf{UIPFE.Enc}^*(\mathsf{UIPFE.PP}^*, \mathsf{UIPFE.MSK}^*, I_{\mathbf{w}^*})$$

and output $\mathsf{CT}^* = (\mathsf{UQFE.CT}^*,\ \mathsf{UIPFE.CT}^*)$.

$\mathsf{\textbf{KeyGen}}^*(\mathsf{\textbf{PP}}^*, \mathsf{\textbf{MSK}}^*, \mathbf{y}, \mathbf{v}, (\boldsymbol{\sigma}, \boldsymbol{\mu}))$: Compute

$$\mathsf{UQFE.SK}^*_{\mathbf{y}\otimes\mathbf{v}} \leftarrow \mathsf{UQFE.KeyGen}^*(\mathsf{UQFE}, \mathsf{PP}^*, \mathsf{UQFE.MSK}^*, \mu\sigma, \mathbf{y}\otimes\mathbf{v}),$$
$$\mathsf{UIPFE.SK}^*_{\mathbf{v}} \leftarrow \mathsf{UIPFE.KeyGen}^*(\mathsf{UIPFE.PP}^*, \mathsf{UIPFE.MSK}^*, \sigma, \mathbf{v}).$$

Output $\mathsf{SK}^*_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}^*_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}^*_{\mathbf{v}})$.

Note that, the pair (σ, μ) is set as in Def. 3, however, it can be seen from the correctness of our UNP-IPFE that the decryption of both UIPFE and UQFE output $[\![0]\!]_T$ when $R(\mathbf{w}^*, \mathbf{v}) \neq 1$, i.e. $\langle \mathbf{w}^*, \mathbf{v}\rangle = 0$. Thus, the simulator reassigns $(\sigma, \mu) = (0, 0)$ if $\langle \mathbf{w}^*, \mathbf{v}\rangle = 0$ while simulating a secret key for $(\mathbf{y}, \mathbf{v})$ such that $(\mathbf{y}, \mathbf{v}) \in \mathcal{R}_s$ and $\langle \mathbf{w}^*, \mathbf{v}\rangle = 0$.

6.3 Security Analysis

Theorem 3. *Assuming the underlying UQFE and UIPFE schemes are* $\mathsf{SA\text{-}SIM}$ *secure, our proposed UNP-IPFE scheme is* $\mathsf{SA\text{-}WAH\text{-}SIM}$ *secure as per Def. 3.*

Proof. We consider a sequence of games to prove the above theorem. Let $\mathcal{A}$ be a PPT adversary of the $\mathsf{SA\text{-}WAH\text{-}SIM}$ security experiment. For $\iota \in \{0, 1, 2\}$, We represent E_ι as the event that $\mathcal{A}$ outputs 1 in Game ι. We show that the games are computationally indistinguishable $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UNP\text{-}IPFE},\mathcal{A}}(\lambda) \equiv$ Game 0 $\approx$ Game 1 $\approx$ Game 2 $\equiv \mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UNP\text{-}IPFE},\mathcal{A},\mathcal{S}}(\lambda)$.

Game 0: This game corresponds to the experiment $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UNP\text{-}IPFE},\mathcal{A}}(\lambda)$ as defined in Def. 3. Therefore, it can be written as

$$\Pr[\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UNP\text{-}IPFE},\mathcal{A}}(\lambda) = 1] = \Pr[\mathsf{E}_0]$$

In this experiment, the ciphertext $\mathsf{CT}^* = (\mathsf{UQFE.CT}^*, \mathsf{UIPFE.CT}^*)$ corresponding to the vector pair $\mathbf{x}^* = (x_i^*)_{i \in I_{\mathbf{x}^*}} \in \mathbb{Z}^{|I_{\mathbf{x}^*}|}$, $\mathbf{w}^* = (w_i^*)_{i \in I_{\mathbf{w}^*}} \in \mathbb{Z}^{|I_{\mathbf{w}^*}|}$ is generated as

$$\mathsf{UQFE.CT}^* = \mathsf{UQFE.Enc}(\mathsf{UQFE.PP}, \mathsf{UQFE.MSK}, \mathbf{x}^*, \mathbf{w}^*)$$
$$\mathsf{UIPFE.CT}^* = \mathsf{UIPFE.Enc}(\mathsf{UIPFE.PP}, \mathsf{UIPFE.MSK}, \mathbf{w}^*)$$

A secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}_{\mathbf{v}})$ queried by the adversary $\mathcal{A}$ corresponding to vectors $\mathbf{y} = (y_i)_{i \in I_{\mathbf{y}}} \in \mathbb{Z}^{|I_{\mathbf{y}}|}, \mathbf{v} = (v_i)_{i \in I_{\mathbf{v}}} \in \mathbb{Z}^{|I_{\mathbf{v}}|}$ is computed as

$$\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}} = \mathsf{UQFE.KeyGen}(\mathsf{UQFE.PP}, \mathsf{UQFE.MSK}, \mathbf{y} \otimes \mathbf{v})$$
$$\mathsf{UIPFE.SK}_{\mathbf{v}} = \mathsf{UIPFE.KeyGen}(\mathsf{UIPFE.PP}, \mathsf{UIPFE.MSK}, \mathbf{v})$$

Game 1: It proceeds exactly the same as Game 0 except the honest algorithms of UIPFE are replaced by their simulated versions. In particular, the challenger replaces $\mathsf{UIPFE.(Setup, Enc, KeyGen)}$ by $\mathsf{UIPFE.(Setup^*,\ Enc^*, KeyGen^*)}$. Therefore, the challenge ciphertext and secret key components generated using the UIPFE are given by

$$\mathsf{UIPFE.CT}^* = \mathsf{UIPFE.Enc}^*(\mathsf{UIPFE.PP}^*, \mathsf{UIPFE.MSK}^*, I_{\mathbf{w}^*})$$
$$\mathsf{UIPFE.SK}_{\mathbf{v}} = \mathsf{UIPFE.KeyGen}^*(\mathsf{UIPFE.PP}^*, \mathsf{UIPFE.MSK}^*, \sigma, \mathbf{v})$$

where $\sigma = \langle \mathbf{w}^*, \mathbf{v} \rangle$ if $(\mathbf{w}^*, \mathbf{v}) \in \mathcal{R}_s$, else $\sigma = \perp$.

Analysis. First, we note that all the secret key queries for the pair of vectors $(\mathbf{v}, \mathbf{y})$ satisfy the condition that $dim\{\mathbf{v} : (\mathbf{w}^*, \mathbf{v}) \in \mathcal{R}_s\} \leq |I_{\mathbf{w}^*}| - 1$. Thus, the $\mathsf{SA\text{-}SIM}$ security of the underlying UIPFE guarantees that for any PPT adversary $\mathcal{B}_1$, we have

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_0]| \leq \mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{B}_1,\mathsf{SA\text{-}SIM}}(\lambda).$$

Game 2: It proceeds exactly the same as Game 1 except the honest algorithms of UQFE are replaced by their simulated versions. In particular, the challenger replaces $\mathsf{UQFE.(Setup, Enc, KeyGen)}$ by $\mathsf{UQFE.(Setup^*,\ Enc^*, KeyGen^*)}$. Therefore, the challenge ciphertext and secret key components generated using the UQFE are given by

$$\mathsf{UQFE.CT}^* = \mathsf{UQFE.Enc}^*(\mathsf{UQFE.PP}^*, \mathsf{UQFE.MSK}^*, I_{\mathbf{x}^*}, I_{\mathbf{w}^*})$$
$$\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}} = \mathsf{UQFE.KeyGen}^*(\mathsf{UQFE.PP}^*, \mathsf{UQFE.MSK}^*, \mu', \mathbf{y} \otimes \mathbf{v})$$

where $\mu' = \mu\sigma = \langle \mathbf{x}^*, \mathbf{y} \rangle \sigma$ if $(\mathbf{x}^*, \mathbf{y}), (\mathbf{w}^*, \mathbf{v}) \in \mathcal{R}_s$; else $\mu' = \perp$. Here, we use the fact that $(\mathbf{x}^* \otimes \mathbf{w}^*)(\mathbf{y} \otimes \mathbf{v})^\top = \langle \mathbf{x}^*, \mathbf{y} \rangle \langle \mathbf{w}^*, \mathbf{v} \rangle$.

Analysis. From the $\mathsf{SA\text{-}SIM}$ security of the underlying UQFE scheme, it holds that for any PPT adversary $\mathcal{B}_2$,

$$|\Pr[\mathsf{E}_2] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}^{\mathsf{UQFE}}_{\mathcal{B}_2,\mathsf{SA\text{-}SIM}}(\lambda).$$

Observe that, Game 2 coincides with the experiment $\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UNP\text{-}IPFE},\mathcal{A},\mathcal{S}}(\lambda)$ of the simulator $\mathcal{S}$ as described above. This concludes the proof. □

6.4 Instantiations

We instantiate our generic UNP-IPFE construction both in the public key setting as well as the secret key setting. We obtain our pubic key UNP-IPFE by plugging the existing UIPFE and UQFE schemes of [TT18, Tom22] to our generic construction. To obtain the secret key construction, we use the UQFE scheme proposed in this paper. In Table 3, we present concrete efficiency matrices of the two instantiations with respect to 128-bit and 256-bit security levels.

Table 3: Sizes of our UNP-IPFE parameters in terms of kilo-bits.

Scheme	(mes, att) vec. length	(key, pred) vec. length	128-bit AES			256-bit AES		
	$m_1 = m_2 = m$	$n_1 = n_2 = n$	MPK	CT	SK	MPK	CT	SK
Secret-key	100	100	0.32	32.19	0.256	0.8	112.48	1.28
UNP-IPFE	200	200	0.32	64.19	0.256	0.8	224.48	1.28
Public-key	100	100	1.632	183.072	134.976	4.08	649.68	674.8
UNP-IPFE	200	200	1.632	365.47	269.37	4.08	1297.68	1346.88

– (mes, att) vec.: (message , attribute) vectors; (key, pred) vec.: (key , predicate) vectors.

– Group sizes of asymmetric pairing follows from 2007 NIST recommendations of [BBB$^+$06]. Descriptions of an elliptic curves are in [FST10]. We consider a 256-bit Barreto-Naehrig curve [BN05] with embedding degree 12 for 128 bit security and a 640-bit Brezing-Weng curve [BW05] with embedding degree 24 for 256-bit security.

Public key UNP-IPFE: In [TT18], Tomida and Takashima proposed two UIPFE schemes where one is a public key scheme. Their public key UIPFE scheme is permissive and achieves adaptive IND-based security under the SXDH assumption in the standard model. Recently in [Tom22], Tomida proposed the first UQFE scheme in the public key setting. Their UQFE construction is in a symmetric vector setting, i.e., the inputs $\mathbf{z}_1 = \mathbf{z}_2 = \mathbf{z}$ and the UQFE scheme recovers $(\mathbf{z} \otimes \mathbf{z})\mathbf{f}^\top$ where $\mathbf{f}$ denotes the key vector. Their scheme achieves semi-adaptive IND-based security under the MDDH assumption in the ROM. In both these schemes, the secret key and ciphertext sizes grow linearly with the length of the vectors. By plugging these schemes in our generic UNP-IPFE construction, we obtain a public key UNP-IPFE scheme in the permissive setting. Since the underlying UIPFE and UQFE schemes are IND-based secure, our public key UNP-IPFE scheme achieves semi-adaptive IND-based security in the ROM. Further, the secret key and ciphertext sizes grow linearly with the length of the associated vectors. Concretely, for a message-attribute pair of length m each, the ciphertext requires $(33m + 21)$ group elements in $\mathbb{G}_1$ and $12m$ group elements in $\mathbb{G}_2$. The secret key requires $(21n + 9)$ group elements in $\mathbb{G}_2$ where n is the length of both key and predicate vectors.

Secret key UNP-IPFE: We use the secret key UQFE scheme proposed in this work to obtain our secret key UNP-IPFE scheme. Our UQFE is an upgrade of the QFE scheme of

Wee [Wee20] in the strict setting with succinct secret keys and compact ciphertexts. The proposed UQFE scheme achieves semi-adaptive SIM-based security in the ROM under the bilateral k-Lin assumption. Since, UIPFE is a special case of UQFE, we instantiate our secret key UNP-IPFE scheme in the strict setting by plugging our strict UQFE into the generic construction. Since the underlying UQFE scheme is semi-adaptive SIM-secure in the ROM, so is our UNP-IPFE scheme. Unlike our public key UNP-IPFE, our secret key UNP-IPFE achieves succinct secret keys due to the succinctness of the underlying UQFE scheme. The size of the ciphertext grows linearly with the length of the vectors. More specifically, for a message-attribute pair with lengths m_1 and m_2, the ciphertext requires $(2m_1 + 4m_2 + 6)$ group elements in $\mathbb{G}_1$ and $2m_2$ group elements in $\mathbb{G}_2$. The secret key only requires 4 group elements in $\mathbb{G}_2$.

References

ABCP15. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Simple functional encryption schemes for inner products. In J. Katz, editor, *Public-Key Cryptography – PKC 2015, Lecture Notes in Computer Science, vol 9020*, pages 733–751. Springer, 2015. 3

ABDCP16. Michel Abdalla, Florian Bourse, Angelo De Caro, and David Pointcheval. Better security for functional encryption for inner product evaluations. *Cryptology ePrint Archive*, 2016. 3

ABKW19. Michel Abdalla, Fabrice Benhamouda, Markulf Kohlweiss, and Hendrik Waldner. Decentralizing inner-product functional encryption. In D. Lin and K. Sako, editors, *Public-Key Cryptography – PKC 2019, Lecture Notes in Computer Science, vol 11443*, pages 128–157. Springer, 2019. 3

ACF$^+$18. Michel Abdalla, Dario Catalano, Dario Fiore, Romain Gay, and Bogdan Ursu. Multi-input functional encryption for inner products: function-hiding realizations and constructions without pairings. In H. Shacham and A. Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018, Lecture Notes in Computer Science, vol 10991*, pages 597–627. Springer, 2018. 3

ACGU20. Michel Abdalla, Dario Catalano, Romain Gay, and Bogdan Ursu. Inner-product functional encryption with fine-grained access control. In S. Moriai and H. Wang, editors, *Advances in Cryptology – ASIACRYPT 2020, Lecture Notes in Computer Science, vol 12493*, pages 467–497. Springer, 2020. 3, 4, 5, 7, 8, 9, 11

AGRW17. Michel Abdalla, Romain Gay, Mariana Raykova, and Hoeteck Wee. Multi-input inner-product functional encryption from pairings. In JS. Coron and J. Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Lecture Notes in Computer Science, vol 10210*, pages 601–626. Springer, 2017. 3

AGT21a. Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-input quadratic functional encryption from pairings. In T. Malkin and C. Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Lecture Notes in Computer Science, vol 12828*, pages 208–238. Springer, 2021. 3, 4, 7

AGT21b. Shweta Agrawal, Rishab Goyal, and Junichi Tomida. Multi-party functional encryption. In K. Nissim and B. Waters, editors, *Theory of Cryptography Conference – TCC 2021, Lecture Notes in Computer Science, vol 13043*, pages 224–255. Springer, 2021. 3, 4, 5, 7, 9, 11, 15

AGW20. Michel Abdalla, Junqing Gong, and Hoeteck Wee. Functional encryption for attribute-weighted sums from k-lin. In Ristenpart T. Micciancio D., editor, *Advances in Cryptology – CRYPTO 2020, Lecture Notes in Computer Science, vol 12170*, pages 685–716. Springer, 2020. 7, 9, 10

ALS16. Shweta Agrawal, Benoît Libert, and Damien Stehlé. Fully secure functional encryption for inner products, from standard assumptions. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016, Lecture Notes in Computer Science, vol 9816*, pages 333–362. Springer, 2016. 3

AMY19. Shweta Agrawal, Monosij Maitra, and Shota Yamada. Attribute based encryption (and more) for nondeterministic finite automata from LWE. In A. Boldyreva and D. Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Lecture Notes in Computer Science, vol 11693*, pages 765–797. Springer, 2019. 4

APM20. Shweta Agrawal and Alice Pellet-Mary. Indistinguishability obfuscation without maps: attacks and fixes for noisy linear fe. In A. Canteaut and Y. Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Lecture Notes in Computer Science, vol 12105*, pages 110–140. Springer, 2020. 3

Att19. Nuttapong Attrapadung. Unbounded dynamic predicate compositions in attribute-based encryption. In Y. Ishai and V. Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019, Lecture Notes in Computer Science, vol 11476*, pages 34–67. Springer, 2019. 4

BBB$^+$06. Elaine Barker, Elaine Barker, William Burr, William Polk, Miles Smid, et al. *Recommendation for key management: Part 1: General*. National Institute of Standards and Technology, Technology Administration . . . , 2006. 67

BBL17. Fabrice Benhamouda, Florian Bourse, and Helger Lipmaa. CCA-secure inner-product functional encryption from projective hash functions. In S. Fehr, editor, *Public-Key Cryptography – PKC 2017, Lecture Notes in Computer Science, vol 10175.*, pages 36–66. Springer, 2017. 3

BCFG17. Carmen Elisabetta Zaira Baltico, Dario Catalano, Dario Fiore, and Romain Gay. Practical functional encryption for quadratic functions with applications to predicate encryption. In J. Katz and H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Lecture Notes in Computer Science, vol 10401*, pages 67–98. Springer, 2017. 3, 15

BJK15. Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In T. Iwata and J. Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Lecture Notes in Computer Science, vol 9452*, pages 470–491. Springer, 2015. 3

BN05. Paulo SLM Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In B. Preneel and S. Tavares, editors, *International workshop on selected areas in cryptography – SAC 2005, Lecture Notes in Computer Science, vol 3897*, pages 319–331. Springer, 2005. 67

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Y. Ishai, editor, *Theory of Cryptography Conference – TCC 2011, Lecture Notes in Computer Science, vol 6597*, pages 253–273. Springer, 2011. 14

BV16. Zvika Brakerski and Vinod Vaikuntanathan. Circuit-ABE from LWE: unbounded attributes and semi-adaptive security. In M. Robshaw and J. Katz, editors, *Advances in Cryptology – CRYPTO 2016, Lecture Notes in Computer Science, vol 9816*, pages 363–384. Springer, 2016. 4

BV18. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. *Journal of the ACM (JACM)*, 65(6):1–37, 2018. 3

BW05. Friederike Brezing and Annegret Weng. Elliptic curves suitable for pairing based cryptography. *Designs, Codes and Cryptography*, 37(1):133–141, 2005. 67

CLT18. Guilhem Castagnos, Fabien Laguillaumie, and Ida Tucker. Practical fully secure unrestricted inner product functional encryption modulo p. In T. Peyrin and S. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Lecture Notes in Computer Science, vol 11273*, pages 733–764. Springer, 2018. 3

DDM16. Pratish Datta, Ratna Dutta, and Sourav Mukhopadhyay. Functional encryption for inner product with full function privacy. In CM. Cheng, KM. Chung, G. Persiano, and BY. Yang, editors, *Public-Key Cryptography–PKC 2016, Lecture Notes in Computer Science, vol 9614*, pages 164–195. Springer, 2016. 3

DOT18. Pratish Datta, Tatsuaki Okamoto, and Junichi Tomida. Full-hiding (unbounded) multi-input inner product functional encryption from the k-Linear assumption. In M. Abdalla and R. Dahab, editors, *Public-Key Cryptography – PKC 2018, Lecture Notes in Computer Science, vol 10770*, pages 245–277. Springer, 2018. 3

DP21. Pratish Datta and Tapas Pal. (Compact) adaptively secure FE for attribute-weighted sums from k-lin. In *Advances in Cryptology – ASIACRYPT 2021, Lecture Notes in Computer Science, vol 13093*, pages 434–467. Springer, 2021. 7, 9

DPD21. Subhranil Dutta, Tapas Pal, and Ratna Dutta. Fully secure unbounded zero inner product encryption with short ciphertexts and keys. In Q. Huang and Y. Yu, editors, *International Conference on Provable Security, Lecture Notes in Computer Science, vol 13059*, pages 241–258. Springer, 2021. 4, 5, 11

DSP19. Edouard Dufour-Sans and David Pointcheval. Unbounded inner-product functional encryption with succinct keys. In R. Deng, V. Gauthier-Umaña, M. Ochoa, and M. Yung, editors, *Applied Cryptography and Network Security – ACNS 2019, Lecture Notes in Computer Science, vol 11464*, pages 426–441. Springer, 2019. 4, 5, 7, 9, 11, 15

EHK$^+$17. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for diffie–hellman assumptions. *Journal of cryptology*, 30(1):242–288, 2017. 18

FST10. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of cryptology*, 23(2):224–280, 2010. 67

Gay20. Romain Gay. A new paradigm for public-key functional encryption for degree-2 polynomials. In *IACR International Conference on Public-Key Cryptography – PKC 2020, Lecture Notes in Computer Science, vol 12110*, pages 95–120. Springer, 2020. 3, 15

GKP$^+$13. Shafi Goldwasser, Yael Kalai, Raluca Ada Popa, Vinod Vaikuntanathan, and Nickolai Zeldovich. Reusable garbled circuits and succinct functional encryption. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 555–564, 2013. 3

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and Communications security*, pages 89–98, 2006. 3, 4

GVW15. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *Journal of the ACM (JACM)*, 62(6):1–33, 2015. 3

JLS21. Aayush Jain, Huijia Lin, and Amit Sahai. Indistinguishability obfuscation from well-founded assumptions. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 60–73, 2021. 3

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In N. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008, Lecture Notes in Computer Science, vol 4965*, pages 146–162. Springer, 2008. 4, 8

KY19. Shuichi Katsumata and Shota Yamada. Non-zero inner product encryption schemes from various assumptions: LWE, DDH and DCR. In D. Lin and K. Sako, editors, *Public-Key Cryptography – PKC 2019, Lecture Notes in Computer Science, vol 11443*, pages 158–188. Springer, 2019. 14

Lin17. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In J. Katz and H. Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Lecture Notes in Computer Science, vol 10401*, pages 599–629. Springer, 2017. 15

LKK$^+$18. Joohee Lee, Dongwoo Kim, Duhyeong Kim, Yongsoo Song, Junbum Shin, and Jung Hee Cheon. Instant privacy-preserving biometric authentication for hamming distance. Cryptology ePrint Archive, Paper 2018/1214, 2018. https://eprint.iacr.org/2018/1214. 8

LLW21. Qiqi Lai, Feng-Hao Liu, and Zhedong Wang. New lattice two-stage sampling technique and its applications to functional encryption–stronger security and smaller ciphertexts. In A. Canteaut and FX. Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Lecture Notes in Computer Science, vol 12696*, pages 498–527. Springer, 2021. 3, 4, 5, 7, 8, 11

LT19. Benoit Libert and Radu Titiu. Multi-client functional encryption for linear functions in the standard model from LWE. In S. Galbraith and S. Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Lecture Notes in Computer Science, vol 11923*, pages 520–551. Springer, 2019. 3

LW11. Allison Lewko and Brent Waters. Unbounded HIBE and attribute-based encryption. In K.G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011, Lecture Notes in Computer Science, vol 6632*, pages 547–567. Springer, 2011. 4

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In T. Rabin, editor, *Advances in Cryptology – CRYPTO 2010, Lecture Notes in Computer Science, vol 6223*, pages 191–208. Springer, 2010. 12, 22

OT12a. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012, Lecture Notes in Computer Science, vol 7237*, pages 591–608. Springer, 2012. 4

OT12b. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In X. Wang and K. Sako, editors, *Advances in Cryptology – ASIACRYPT 2012, Lecture Notes in Computer Science, vol 7658*, pages 349–366. Springer, 2012. 4, 5, 11, 13, 14, 19

OT15. Tatsuaki Okamoto and Katsuyuki Takashima. Achieving short ciphertexts or short secret-keys for adaptively secure general inner-product encryption. *Designs, Codes and Cryptography*, 77(2):725–771, 2015. 4

PD21. Tapas Pal and Ratna Dutta. CCA secure attribute-hiding inner product encryption from minimal assumption. In *Information Security and Privacy: 26th Australasian Conference, ACISP 2021, Virtual Event, December 1–3, 2021, Proceedings*, page 254–274, Berlin, Heidelberg, 2021. Springer-Verlag. 3, 4, 5, 8, 11, 14

Tom22. Junichi Tomida. Unbounded quadratic functional encryption and more from pairings. Cryptology ePrint Archive, Paper 2022/1124, 2022. 6, 7, 10, 15, 17, 51, 54, 67

TT18. Junichi Tomida and Katsuyuki Takashima. Unbounded inner product functional encryption from bilinear maps. In T. Peyrin and S. Galbraith, editors, *Advances in Cryptology – ASIACRYPT 2018, Lecture Notes in Computer Science, vol 11273*, pages 609–639. Springer, 2018. 4, 5, 6, 7, 9, 11, 14, 15, 17, 24, 25, 67

Wat09. Brent Waters. Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. In S. Halevi, editor, *Advances in Cryptology - CRYPTO 2009, Lecture Notes in Computer Science, vol 5677*, pages 619–636. Springer, 2009. 11

Wee20. Hoeteck Wee. Functional encryption for quadratic functions from k-lin, revisited. In R. Pass and K. Pietrzak, editors, *Theory of Cryptography Conference – TCC 2020, Lecture Notes in Computer Science, vol 12550*, pages 210–228. Springer, 2020. 3, 7, 15, 16, 17, 68

A Appendix

We provide the SIM-based security definition of UIPFE with permissive relations. As we mentioned earlier that the permissive relation implies the strict relation, so we use the following security model in the security analysis of our secret key UNP-IPFE in Section 6.3 with strict relation.

Definition 5 (SA-SIM Security for UIPFE). *The* $\mathsf{UIPFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is said to be semi-adaptive simulation* (SA-SIM) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a PPT simulator* $\mathcal{S} := (\mathsf{Setup}^*, \mathsf{Enc}^*, \mathsf{KeyGen}^*)$ *such that the following holds*

$$\mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{A},\mathsf{SA\text{-}SIM}}(\lambda) := \left|\Pr[\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UIPFE},\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UIPFE},\mathcal{A},\mathcal{S}}(\lambda) = 1]\right| \leq \mathsf{negl}(\lambda)$$

where the experiments $\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UIPFE},\mathcal{A}}(\lambda)$ *and* $\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UIPFE},\mathcal{A},\mathcal{S}}(\lambda)$ *are defined as follows:*

```
$\underline{\mathsf{Exp}^{\mathsf{Real}}_{\mathsf{UIPFE},\mathcal{A}}(\lambda)}$
1: $(\mathsf{PP}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$
2: $\mathbf{x}^* \leftarrow \mathcal{A}(\mathsf{PP})$
3: $\mathsf{CT}^* \leftarrow \mathsf{Enc}(\mathsf{PP}, \mathsf{MSK}, \mathbf{x}^*)$
4: $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{PP},\mathsf{MSK},\cdot)}(\mathsf{CT}^*)$
```

```
$\underline{\mathsf{Exp}^{\mathsf{Ideal}}_{\mathsf{UIPFE},\mathcal{A},\mathcal{S}}(\lambda)}$
1: $(\mathsf{PP}^*, \mathsf{MSK}^*) \leftarrow \mathsf{Setup}^*(1^\lambda)$
2: $\mathbf{x}^* \leftarrow \mathcal{A}(\mathsf{PP}^*)$
3: $\mathsf{CT}^* \leftarrow \mathsf{Enc}^*(\mathsf{PP}^*, \mathsf{MSK}^*, I_{\mathbf{x}^*})$
4: $b \leftarrow \mathcal{A}^{\mathsf{KeyGen}^*(\mathsf{PP}^*,\mathsf{MSK}^*,\cdot,\cdot)}(\mathsf{CT}^*)$
```

In the Real *security experiment,* $\mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \cdot)$ *is an oracle that takes input the secret key vector* $\mathbf{y}$ *with associated the index set* $I_{\mathbf{y}}$ *and outputs* $\mathsf{SK}_{\mathbf{y}} \leftarrow \mathsf{KeyGen}(\mathsf{PP}, \mathsf{MSK}, \mathbf{y})$*. In the* Ideal *security experiment,* $\mathsf{KeyGen}^*(\mathsf{PP}^*, \mathsf{MSK}^*, \cdot, \cdot)$ *oracle returns a simulated secret key* $\mathsf{SK}^*_{\mathbf{y}}$ *on input a key vector* $\mathbf{y}$ *with index set* $I_{\mathbf{y}}$ *and* μ *where the value of* μ *is* $\langle \mathbf{x}^*, \mathbf{y} \rangle$ *whenever the condition* $(\mathbf{x}^*, \mathbf{y}) \in \mathcal{R}_p$ *holds, else* $\mu = \perp$*.*

Now, we present the security definitions of UQFE and UP-IPFE with the permissive relation in the IND-based model, which are needed for the security analysis of our UNP-IPFE in Appendix A.1.

Definition 6 (Semi-Adaptive Indistinguishability). *The* $\mathsf{UQFE} = (\mathsf{Setup}, \mathsf{Enc}, \mathsf{KeyGen}, \mathsf{Dec})$ *is said to be semi-adaptive indistinguishability* (SA-IND) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that the following holds*

$$\mathit{Adv}^{\mathsf{UQFE}}_{\mathcal{A},\mathsf{SA\text{-}IND}}(\lambda) := \left|\Pr\left[\mathsf{Expt}^{\mathsf{UQFE}}_{0,\mathcal{A},\mathsf{SA\text{-}IND}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{UQFE}}_{1,\mathcal{A},\mathsf{SA\text{-}IND}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{UQFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}IND}}(\lambda)$ *is defined for* $\beta \in \{0, 1\}$ *as follows:*

```
$\underline{\mathsf{Expt}^{\mathsf{UQFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}IND}}(\lambda)}$
  $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$
  $((\mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}), (\mathbf{z}_1^{(1)}, \mathbf{z}_2^{(1)})) \leftarrow \mathcal{A}(1^\lambda, \mathsf{MPK})$ where $|I_{\mathbf{z}_1^{(0)}}| = |I_{\mathbf{z}_1^{(1)}}|$ and $|I_{\mathbf{z}_2^{(0)}}| = |I_{\mathbf{z}_2^{(1)}}|$
  $\mathsf{CT}^{(\beta)} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathbf{z}_1^{(\beta)}, \mathbf{z}_2^{(\beta)})$
  $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot)}(\mathsf{MPK}, \mathsf{CT}^{(\beta)})$
  Outputs: $\beta'$
```

In this experiment, $\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \cdot)$ *is an oracle that takes input the secret key vector* $\mathbf{f}$ *with the associated index set* $I_{\mathbf{f}}$ *(a cartesian product between two index sets* $I_{\mathbf{f}_1}, I_{\mathbf{f}_2}$*) and outputs the secret key* $\mathsf{SK}_{\mathbf{f}} \leftarrow \mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{f})$ *satisfying* $(\mathbf{z}_1^{(0)} \otimes \mathbf{z}_2^{(0)})\mathbf{f}^\top = (\mathbf{z}_1^{(1)} \otimes \mathbf{z}_2^{(1)})\mathbf{f}^\top$ *whenever* $I_{\mathbf{f}_1} \subseteq I_{\mathbf{z}_1^{(0)}}, I_{\mathbf{f}_2} \subseteq I_{\mathbf{z}_2^{(0)}}$ *and* $I_{\mathbf{f}_1} \subseteq I_{\mathbf{z}_1^{(1)}}, I_{\mathbf{f}_2} \subseteq I_{\mathbf{z}_2^{(1)}}$*. Here* $(I_{\mathbf{z}_1^{(0)}}, I_{\mathbf{z}_2^{(0)}})$ *and* $(I_{\mathbf{z}_1^{(1)}}, I_{\mathbf{z}_2^{(1)}})$ *represents the index sets of the challenge message vectors* $(\mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)})$ *and* $(\mathbf{z}_1^{(1)}, \mathbf{z}_2^{(1)})$ *respectively.*

Definition 7 (Semi-Adaptive Weak Attribute-hiding Indistinguishability). *The* UP-IPFE = (Setup, Enc, KeyGen, Dec) *is said to be semi-adaptive weak attribute-hiding indistinguishability* (SA-WAH-IND) *secure if for any security parameter* λ*, any PPT adversary* $\mathcal{A}$*, there exists a negligible function* negl *such that the following holds*

$$Adv^{\mathsf{UP\text{-}IPFE}}_{\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda) := \left| \Pr\left[\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{0,\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda) = 1\right] - \Pr\left[\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{1,\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda) = 1\right]\right| \leq \mathsf{negl}(\lambda)$$

where the experiment $\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda)$ *is defined for* $\beta \in \{0, 1\}$ *as follows:*

```
$\underline{\mathsf{Expt}^{\mathsf{UP\text{-}IPFE}}_{\beta,\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda)}$
  $(\mathsf{MPK}, \mathsf{MSK}) \leftarrow \mathsf{Setup}(1^\lambda)$
  $(\mathbf{w}^{(0)}, \mathbf{w}^{(1)}) \leftarrow \mathcal{A}(1^\lambda, \mathsf{MPK})$ where $|I_{\mathbf{w}^{(0)}}| = |I_{\mathbf{w}^{(1)}}|$
  $(\mathbf{x}^{(0)}, \mathbf{x}^{(1)}) \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot,\cdot)}(\mathsf{MPK})$ where $|I_{\mathbf{x}^{(0)}}| = |I_{\mathbf{x}^{(1)}}|$.
  $\mathsf{CT}^{(\beta)}_{\mathbf{x},\mathbf{w}} \leftarrow \mathsf{Enc}(\mathsf{MPK}, \mathbf{x}^{(\beta)}, \mathbf{w}^{(\beta)})$
  $\beta' \leftarrow \mathcal{A}^{\mathsf{KeyGen}(\mathsf{MPK},\mathsf{MSK},\cdot,\cdot)}(\mathsf{MPK}, \mathsf{CT}^{(\beta)}_{\mathbf{x},\mathbf{w}})$
  Outputs: $\beta'$
```

In this experiment, $\mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \cdot, \cdot)$ *is an oracle that takes input the key-predicate vector pair* $(\mathbf{y}, \mathbf{v})$ *associated with the index sets* $I_{\mathbf{y}}, I_{\mathbf{v}}$ *and outputs the secret key* $\mathsf{SK}_{\mathbf{y},\mathbf{v}} \leftarrow \mathsf{KeyGen}(\mathsf{MPK}, \mathsf{MSK}, \mathbf{y}, \mathbf{v})$*. The secret key queries satisfy the following conditions:*

- *if* $(\mathbf{w}^{(b)}, \mathbf{v}) \in \mathcal{R}_p$ *for* $b = 1, 2$ *then* $\langle \mathbf{w}^{(0)}, \mathbf{v}\rangle = \langle \mathbf{w}^{(1)}, \mathbf{v}\rangle$,
- *if* $R(\mathbf{w}^{(0)}, \mathbf{v}) = R(\mathbf{w}^{(1)}, \mathbf{v}) = 1$ *and* $(\mathbf{x}^{(b)}, \mathbf{y}), (\mathbf{w}^{(b)}, \mathbf{v}) \in \mathcal{R}_p$ *then* $\langle \mathbf{x}^{(0)}, \mathbf{y}\rangle = \langle \mathbf{x}^{(1)}, \mathbf{y}\rangle$.

A.1 IND-based Security Analysis of UNP-IPFE of Section 6

Theorem 4. *Assuming the underlying UQFE and UIPFE schemes are* SA-IND*-based secure in the public key setting, then UNP-IPFE scheme as described in Sec. 6 is a* SA-WAH-IND *secure as per Def. 7.*

Proof. We consider a PPT adversary $\mathcal{A}$ against $\mathsf{SA\text{-}WAH\text{-}IND}$ security of the UNP-IPFE scheme. Let us choose an adversary $\mathcal{B}_1$ against $\mathsf{SA\text{-}IND}$ security of the underlying UQFE scheme and an adversary $\mathcal{B}_2$ against $\mathsf{SA\text{-}IND}$ security of the underlying UIPFE scheme. In particular, we show that if $\mathcal{A}$ can break the $\mathsf{SA\text{-}WAH\text{-}IND}$ security of the UNP-IPFE scheme, then there exist PPT adversaries $\mathcal{B}_1$, $\mathcal{B}_2$ which will break $\mathsf{SA\text{-}IND}$ security of the UQFE and $\mathsf{SA\text{-}IND}$ security of the UIPFE scheme.

To prove this theorem, consider the following games. We start with Game 0 which is the real $\mathsf{SA\text{-}WAH\text{-}IND}$ security experiment as mentioned in Def. 7 where the challenger chooses the random bit as $\beta = 0$. Then we modify this game in Game 1 and finally end up in Game 2 where the random bit (chosen by the challenger) is converted to $\beta = 1$. We proof the indistinguishability between corresponding games using the security of UQFE and UIPFE. Let E_ι denotes the event that $\mathcal{A}$ outputs 1 in Game ι. Now, we formally describe the games as follows:

Game 0: Game 0 is the same as real security experiment $\mathsf{Expt}^{\mathsf{UNP\text{-}IPFE}}_{0,\mathcal{A},\mathsf{SA\text{-}WAH\text{-}IND}}(\lambda)$ of Def. 7. All the secret key queries corresponding to the key-predicate vector pair $(\mathbf{y}, \mathbf{v})$ associated with the index sets $I_{\mathbf{y}}, I_{\mathbf{v}}$ must satisfy the restrictions as given in Def. 7. The challenge ciphertext $\mathsf{CT}^{(0)}_{\mathbf{x},\mathbf{w}} = (\mathsf{UQFE.CT}^{(0)}, \mathsf{UIPFE.CT}^{(0)})$ corresponding to the pair of vectors $(\mathbf{x}^{(0)}$, $\mathbf{w}^{(0)})$ is generated as

$$\begin{aligned}\mathsf{UQFE.CT}^{(0)} &= \mathsf{UQFE.Enc}(\mathsf{UQFE.MPK}, \mathbf{x}^{(0)}, \mathbf{w}^{(0)}),\\ \mathsf{UIPFE.CT}^{(0)} &= \mathsf{UIPFE.Enc}(\mathsf{UIPFE.MPK}, \mathbf{w}^{(0)}).\end{aligned}$$

The secret key $\mathsf{SK}_{\mathbf{y},\mathbf{v}} = (\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}}, \mathsf{UIPFE.SK}_{\mathbf{v}})$ associated with the pair of vectors $(\mathbf{y}, \mathbf{v})$ are generated as

$$\begin{aligned}\mathsf{UQFE.SK}_{\mathbf{y}\otimes\mathbf{v}} &= \mathsf{UQFE.KeyGen}(\mathsf{UQFE.MSK}, \mathbf{y} \otimes \mathbf{v}),\\ \mathsf{UIPFE.SK}_{\mathbf{v}} &= \mathsf{UIPFE.KeyGen}(\mathsf{UIPFE.MSK}, \mathbf{v}).\end{aligned}$$

Game 1: Game 1 is identical with Game 0 except the second component of the challenge ciphertext is now replaced with

$$\boxed{\mathsf{UIPFE.CT}^{(1)} = \mathsf{UIPFE.Enc}(\mathsf{UIPFE.MPK}, \mathbf{w}^{(1)})}$$

Therefore, the challenge ciphertext can be represented as $(\mathsf{UQFE.CT}^{(0)}$, $\mathsf{UIPFE.CT}^{(1)})$. Consider, $\mathcal{B}_2$ is an admissible adversary for the $\mathsf{SA\text{-}IND}$ security game of UIPFE. From the admissible condition of $\mathsf{SA\text{-}WAH\text{-}IND}$)(as per Def. 7), it holds that $\langle \mathbf{w}^{(0)}, \mathbf{v}\rangle = \langle \mathbf{w}^{(1)}, \mathbf{v}\rangle$ for all secret key queries corresponding to the key, predicate vectors $\mathbf{y}, \mathbf{v}$ satisfying $(\mathbf{w}^{(b)}, \mathbf{v}) \in \mathcal{R}_p$. Therefore, the advantage of $\mathcal{A}$ in distinguishing between Game 1 and Game 2 is exactly the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{UIPFE}}_{0,\mathcal{B}_2,\mathsf{SA\text{-}IND}}(\lambda)$ and $\mathsf{Expt}^{\mathsf{UIPFE}}_{1,\mathcal{B}_2,\mathsf{SA\text{-}IND}}(\lambda)$. Thus, we have

$$|\Pr[\mathsf{E}_0] - \Pr[\mathsf{E}_1]| \leq \mathsf{Adv}^{\mathsf{UIPFE}}_{\mathcal{B}_2,\mathsf{SA\text{-}IND}}(\lambda).$$

Game 2: Game 2 is the same as Game 1 except the second component of the challenger ciphertext is now replaced by

$$\boxed{\mathsf{UQFE.CT}^{(1)} = \mathsf{UQFE.Enc}(\mathsf{UQFE.MPK}, \mathbf{x}^{(1)}, \mathbf{w}^{(1)})}$$

Observe that, for $\langle \mathbf{w}^{(0)}, \mathbf{v} \rangle = \langle \mathbf{w}^{(1)}, \mathbf{v} \rangle \neq 0$, i.e., when the decryption succeeds, it holds that $\langle \mathbf{x}^{(0)}, \mathbf{y} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle$ whenever $(\mathbf{w}^{(b)}, \mathbf{v}), (\mathbf{x}^{(b)}, \mathbf{y}) \in \mathcal{R}_p$. Therefore, $\mathcal{B}_1$ is an admissible adversary for the $\mathsf{SA\text{-}IND}$ security game of UQFE since

$$(\mathbf{x}^{(0)} \otimes \mathbf{w}^{(0)})(\mathbf{y} \otimes \mathbf{v})^\top = \langle \mathbf{x}^{(0)}, \mathbf{y} \rangle \langle \mathbf{w}^{(0)}, \mathbf{v} \rangle = \langle \mathbf{x}^{(1)}, \mathbf{y} \rangle \langle \mathbf{w}^{(1)}, \mathbf{v} \rangle = (\mathbf{x}^{(1)} \otimes \mathbf{w}^{(1)})(\mathbf{y} \otimes \mathbf{v})^\top$$

holds for all key queries made by $\mathcal{B}_1$ satisfying $(\mathbf{w}^{(b)}, \mathbf{v}), (\mathbf{x}^{(b)}, \mathbf{y}) \in \mathcal{R}_p$. Thus, the advantage of $\mathcal{A}$ in distinguishing between Game 1 and Game 2 is exactly the same as the advantage in distinguishing between the experiments $\mathsf{Expt}^{\mathsf{UQFE}}_{0,\mathcal{B}_1,\mathsf{SA\text{-}IND}}(\lambda)$ and $\mathsf{Expt}^{\mathsf{UQFE}}_{1,\mathcal{B}_1,\mathsf{SA\text{-}IND}}(\lambda)$, and we have

$$|\Pr[\mathsf{E}_1] - \Pr[\mathsf{E}_2]| \leq \mathsf{Adv}^{\mathsf{UQFE}}_{\mathcal{B}_1,\mathsf{SA\text{-}IND}}(\lambda).$$

This completes the security proof. □