Adaptively Simulation-Secure Attribute-Hiding Predicate Encryption

Pratish Datta[1], Tatsuaki Okamoto[1], and Katsuyuki Takashima[2]

[1] NTT Secure Platform Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180-8585 Japan
`pratish.datta.yg@hco.ntt.co.jp, tatsuaki.okamoto@gmail.com`
[2] Mitsubishi Electric
5-1-1 Ofuna, Kamakura, Kanagawa, 247-8501 Japan
`Takashima.Katsuyuki@aj.MitsubishiElectric.co.jp`

November 12, 2018

Abstract

This paper demonstrates how to achieve *simulation-based strong attribute hiding* against *adaptive* adversaries for *predicate encryption* (PE) schemes supporting *expressive* predicate families under *standard* computational assumptions in bilinear groups. Our main result is a *simulation-based adaptively strongly partially-hiding* PE (PHPE) scheme for predicates computing *arithmetic branching programs* (ABP) on *public* attributes, followed by an *inner-product predicate* on *private* attributes. This simultaneously generalizes attribute-based encryption (ABE) for boolean formulas and ABP's as well as strongly attribute-hiding PE schemes for inner products. The proposed scheme is proven secure for *any a priori bounded* number of ciphertexts and an *unbounded* (polynomial) number of decryption keys, which is the *best possible* in the simulation-based adaptive security framework. This directly implies that our construction also achieves *indistinguishability-based strongly partially-hiding* security against adversaries requesting an *unbounded* (polynomial) number of ciphertexts and decryption keys. The security of the proposed scheme is derived under (asymmetric version of) the *well-studied decisional linear* (DLIN) assumption. Our work resolves an open problem posed by Wee in TCC 2017, where his result was limited to the *semi-adaptive* setting. Moreover, our result advances the current state of the art in both the fields of simulation-based and indistinguishability-based strongly attribute-hiding PE schemes. Our main technical contribution lies in extending the strong attribute hiding methodology of Okamoto and Takashima [EUROCRYPT 2012, ASIACRYPT 2012] to the framework of simulation-based security and beyond inner products.

Keywords: predicate encryption, partially-hiding, simulation-based adaptive security, arithmetic branching programs, inner products

1 Introduction

Functional encryption (FE) is a new vision of modern cryptography that aims to overcome the potential limitation of the traditional encryption schemes, namely, the all or nothing control over decryption capabilities. FE supports restricted decryption keys which enable decrypters to learn specific functions of encrypted messages, and nothing else. More precisely, a (public-key) FE scheme for a function family $\mathcal{F}$ involves a setup authority which holds a master secret key and publishes public system parameters. An encrypter uses the public parameters to encrypt its message M belonging to some supported message space $\mathbb{M}$, creating a ciphertext CT. A decrypter may obtain a private decryption key $\text{SK}(F)$ for some function $F \in \mathcal{F}$ from the setup authority, provided the authority deems that the decrypter is entitled for that key. Such a decryption key $\text{SK}(F)$ can be used to decrypt CT to recover $F(M)$, but nothing more about M.

The most intuitive security requirement for an FE scheme is *collusion resistance*, i.e., a group of decrypters cannot jointly retrieve any more information about an encrypted message beyond the union of what each of them is allowed to learn individually. This intuitive notion has been formalized by Boneh et al. [BSW11] and O'Neill [O'N10] in two distinct frameworks, namely, (a) *indistinguishability-based security* and (b) *simulation-based security.* The former stipulates that

[*] This is the full version of an extended abstract that will appear in the proceedings of ASIACRYPT 2018.

distinguishing encryptions of any two messages is infeasible for a group of colluders which do not have a decryption key that decrypts the ciphertext to distinct values. The latter, on the other hand, stipulates the existence of a polynomial-time simulator that given $F_1(M), \ldots, F_{q_{\text{KEY}}}(M)$ for any message $M \in \mathbb{M}$ and functions $F_1, \ldots, F_{q_{\text{KEY}}} \in \mathcal{F}$, outputs the view of the colluders which are given an encryption of M together with decryption keys for $F_1, \ldots, F_{q_{\text{KEY}}}$. Both of the above notions can be further refined, depending on how the queries of the adversary to the decryption key generation and encryption oracles depend on one another as well as on the public parameters of the system, as *adaptive* vs *semi-adaptive* vs *selective*. Boneh et al. [BSW11] and O'Neill [O'N10] showed that in general, simulation-based security provides a stronger guarantee than indistinguishability-based security, i.e., simulation-based security of some kind, e.g., adaptive, semi-adaptive, or selective, implies indistinguishability-based security of the same kind; but the converse does not hold in general. In fact, Boneh et al. pointed out that indistinguishability-based security is vacuous for certain circuit families, which indicates that we should opt for simulation-based security whenever possible. On the other hand, it is known that while security for single and multiple ciphertexts are equivalent in the indistinguishability-based setting [BSW11], this is not the case in the simulation-based setting [BSW11, BO13, AGVW13, DCIJ$^+$13]. In particular, it has been demonstrated by Boneh et al. [BSW11] that in the adaptive or semi-adaptive simulation-based setting, where the adversary is allowed to make decryption key queries even after receiving the queried ciphertexts, achieving security for an unbounded number of ciphertexts is impossible.

An important subclass of FE is *predicate encryption* (PE). In recent years, with the rapid advancement of Internet communication and cloud technology, there has been an emerging trend among individuals and organizations to outsource potentially sensitive private data to external untrusted servers, and to perform selective computations on the outsourced data by remotely querying the server at some later point in time, or to share specific portions of the outsourced data to other parties of choice. PE is an indispensable tool for performing such operations on outsourced sensitive data without compromising the confidentiality of the data.

Consider a predicate family $R = \{R(Y, \cdot) : \mathcal{X} \to \{0,1\} \mid Y \in \mathcal{Y}\}$, where $\mathcal{X}$ and $\mathcal{Y}$ are two collections of indices or attributes. In a PE scheme for some predicate family R, the associated message space $\mathbb{M}$ is of the form $\mathcal{X} \times \mathcal{M}$, where $\mathcal{M}$ contains the actual payloads. The functionality F_{R_Y} associated with a predicate $R(Y, \cdot) \in R$ is defined as $F_{R_Y}(X, \mathsf{msg}) = \mathsf{msg}$, if $R(Y, X) = 1$, or in other words, Y is authorized for X, and $F_{R_Y}(X, \mathsf{msg}) = \perp$ (a special empty string), if $R(Y, X) = 0$, or in other words, Y is not authorized for X for all $(X, \mathsf{msg}) \in \mathbb{M} = \mathcal{X} \times \mathcal{M}$.

The standard security notion for FE described above, when adopted in the context of PE, stipulates that recovering the payload from a ciphertext generated with respect to some attribute $X \in \mathcal{X}$ should be infeasible for a group of colluders none of which possesses a decryption key corresponding to an attribute authorized for X, also referred to as an authorized decryption key; and moreover, the ciphertext should conceal X from any group of colluders, even those in possession of authorized decryption keys. In the context of PE, this security notion is referred to as *strongly attribute-hiding* security. A weakening of the above notion, called *weakly attribute-hiding* security requires that X should only remain hidden to colluders in possession of unauthorized keys. An even weaker notion, which only demands the payload to remain hidden to colluders with unauthorized keys, is known as *payload-hiding* security, and a payload-hiding PE scheme is often referred to as an *attribute-based encryption* (ABE) scheme in the literature.

Over the last decade, a long sequence of works have developed extremely powerful techniques for realizing indistinguishability-based ABE and weakly attribute-hiding PE schemes supporting more and more expressive predicate families under well-studied computational assumptions in bilinear groups and lattices, culminating into schemes that can now support general polynomial-size circuits [GPSW06, LOS$^+$10, OT10, OT12b, IW14, LW12, Wee14, Att14, CGW15, GVW15a, BGG$^+$14, GVW15b]. However, very little is known for strongly attribute-hiding PE schemes, even in the indistinguishability-based setting. The situation is even worse when security against an unbounded (polynomial) number of authorized-key-possessing colluders under standard computational assumption is considered. In fact, until very recently, the known candidates were restricted to only inner products or even simpler predicates [BW07, KSW08, OT12a, OT12b, OT13, DCIJ+13], out of which the schemes designed in the more efficient and secure prime order bilinear groups being only the works of Okamoto and Takashima [OT12a, OT12b, OT13]. One big reason for this state of the art is that unlike payload-hiding or weakly attribute-hiding, for proving strongly attribute-hiding security, one must argue about an adversary that gets hold of authorized decryption keys, something cryptographers do not have a good understanding of so far. Moreover, there are indeed reasons to believe that constructing strongly attribute-hiding PE schemes for sufficiently expressive predicate classes such as NC^1 under standard computational assumptions could be very difficult. In fact, it is known that a strongly attribute-hiding PE scheme for NC^1 predicates, even in the weakest selective setting, can lead all the way to indistinguishability obfuscation (IO) for general circuits, the new holy grail of modern cryptography [AJ15, BV15, AJS15]. In view of this state of affairs, it is natural to ask the following important question:

Can we realize "the best of both worlds", i.e., can we design PE *scheme for some sufficiently expressive predicate family (e.g.,* NC^1*) that is secure against an unbounded (polynomial) number of colluders under standard computational assumptions (without* IO*), such that the strongly attribute-hiding guarantee holds for a limited segment (e.g., one belonging to some subclass of* NC^1*) of each predicate in the predicate family?*

Towards answering this question, in TCC 2017, Wee [Wee17] put forward a new PE scheme for an NC^1 predicate family in bilinear groups of prime order that is secure against an unbounded (polynomial) number of colluders under the well-studied *k-linear* (k-LIN) assumption, where the strongly attribute-hiding property is achieved only for an inner product evaluating segment of each predicate of the predicate class. More precisely, in his proposed PE system, the ciphertext attribute set $\mathcal{X}$ is given by $\mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ for some finite field $\mathbb{F}_q$ and $n', n \in \mathbb{N}$, while the decryption key attribute set $\mathcal{Y}$ is given by the function family $\mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$. Any function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ operates on a pair $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ by first computing n arithmetic branching programs (ABP) $f_1, \ldots, f_n : \mathbb{F}_q^{n'} \to \mathbb{F}_q$ on $\vec{x}$ to obtain a vector $(f_1(\vec{x}), \ldots, f_n(\vec{x})) \in \mathbb{F}_q^n$, and then evaluating the inner product of the computed vector and $\vec{z}$. The predicate family $R^{\text{ABP}\circ\text{IP}}$ associated with the PE scheme is defined as $R^{\text{ABP}\circ\text{IP}} = \{R^{\text{ABP}\circ\text{IP}}(f, (\cdot, \cdot)) : \mathbb{F}_q^{n'} \times \mathbb{F}_q^n \to \{0, 1\} \mid f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}\}$, where $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 1$, if $f(\vec{x}, \vec{z}) = 0$, and 0, if $f(\vec{x}, \vec{z}) \neq 0$ for any $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ and $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$. The security property of Wee's PE scheme guarantees that other than hiding the payload, a ciphertext generated for some attribute pair $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ also conceals the attribute $\vec{z}$ (but not the attribute $\vec{x}$). Moreover, the concealment of the attribute $\vec{z}$ is strong, i.e., even against colluders possessing authorized keys. Wee termed this security notion as *strongly partially-hiding* security, while the attributes $\vec{x} \in \mathbb{F}_q^{n'}$ and $\vec{z} \in \mathbb{F}_q^n$ as the *public* and *private* attributes respectively.

This PE scheme simultaneously generalizes ABE for boolean formulas and ABP's, as well as strongly attribute-hiding inner-product PE (IPE). For instance, unlike standard IPE schemes, where an inner-product predicate is evaluated between the (private) attribute vector $\vec{z}$ associated with a ciphertext and the attribute vector $\vec{y}$ hardwired within a decryption key, this PE scheme evaluates inner-product predicate between $\vec{z}$ and $\vec{y}$ obtained as the result of complicated ABP computations on a public attribute string $\vec{x}$, which is now associated in addition to the private attribute vector $\vec{z}$ with the ciphertext. This in turn means that this PE scheme can be deployed in richer variants of the applications captured by IPE schemes. For example, it is well-known that inner-product predicates can be used to evaluate conjunctive comparison predicates of the form $R^{\text{COMP}}((c_1, \ldots, c_n), (z_1, \ldots, z_n)) = \bigwedge_{j \in [n]} [z_j \geq c_j]$, where c_j's and z_j's lie in polynomial-size domains [BW07]. In case of standard IPE schemes, $c_1, \ldots, c_n$ are fixed constants which are specified within the decryption key. On the contrary, in case of a PE scheme for $R^{\text{ABP}\circ\text{IP}}$, we can carry out more complex computation, where instead of being fixed constants, $c_1, \ldots, c_n$ can be derived as the outputs of ABP evaluations on public ciphertext attributes. Of course, fixed $c_1, \ldots, c_n$ is a special case of this more expressive computation, since one can have ABP's that ignore the public ciphertext attributes, and simply output hardwired constants. Similarly, standard IPE schemes can be employed for evaluating polynomials with constant coefficients, where the coefficients are specified within the decryption keys [KSW08]. In contrast, in case of a PE scheme for $R^{\text{ABP}\circ\text{IP}}$, the polynomial coefficients can be generated as outputs of ABP computations on public ciphertext attributes.

Partially-hiding PE (PHPE) schemes for similar type of predicate families were considered in [GVW15b, Agr17] in the lattice setting, and those PHPE schemes are in fact capable of evaluating general polynomial-size circuits, as opposed to ABP's in Wee's construction, over public ciphertext attributes prior to evaluating inner-product predicates over private ciphertext attributes. However, those constructions are either only weakly partially-hiding, i.e., the security of the private attributes of the ciphertexts are only guaranteed against unauthorized colluders [GVW15b], or strongly partially-hiding against a priori bounded number of authorized colluders [Agr17]. In contrast, Wee's PHPE scheme is strongly partially-hiding against an unbounded (polynomial) number of authorized colluders. Another strong aspect of the PHPE construction of Wee is that its security is proven in the (unbounded) simulation-based framework [AGVW13], while except [DCIJ$^+$13], all prior PE constructions with strongly attribute-hiding security against an unbounded (polynomial) number of authorized colluders were proven in the weaker indistinguishability-based framework.

However, the PHPE scheme proposed by Wee [Wee17] only achieves semi-adaptive security [CW14], i.e., against an adversary that is restricted to submit its ciphertext queries immediately after viewing the public parameters, and can make decryption key queries only after that. While semi-adaptive security seems somewhat stronger, it has recently been shown by Goyal et al. [GKW16] that it is essentially equivalent to the selective security, the weakest notion of security in which the adversary is bound to declare its ciphertext queries even before the system is setup. Their result also indicates that the gap between semi-adaptive and adaptive security, the strongest and most reasonable notion in which the adversary is allowed to make ciphertext and decryption key queries at any point during the security experiment, is in fact much wider than was previously thought. While Ananth et al. [ABSV15] have demonstrated how to generically transform an FE scheme that supports arbitrary polynomial-size circuits from selective security to one that achieves adaptive security, their conversion does not work for ABE or PE schemes which fall below this threshold in functionality. In view of this state of affairs, it is interesting to explore *whether it is possible to construct an efficient adaptively simulation-secure strongly partially-hiding* PE *scheme for the predicate family* $R^{\text{ABP}\circ\text{IP}}$ *that is secure against an unbounded* (*polynomial*) *number of colluders under well-studied computational assumption.* Note that while several impossibility results exist against the achievability of simulation-based security in certain settings [BSW11, BO13, AGVW13, DCIJ$^+$13], those results do not overrule the existence of such a construction, provided of course we bound the number of allowed ciphertext queries by the adversary. In fact, Wee has posed the realization of such a PHPE construction as an open problem in his paper [Wee17].

Our Contributions

In this paper, we resolve the above open problem. Specifically, our main result is a PE scheme for the predicate family $R^{\text{ABP}\circ\text{IP}}$ that achieves *simulation-based adaptively strongly partially hiding* security against adversaries making *any a priori bounded* number of ciphertext queries while requesting an *unbounded* (polynomial) number of decryption keys both before and after the ciphertext queries, which is the *best* one could hope for in the simulation-based framework when the adversary is allowed to make decryption key queries even after making the ciphertext queries [BSW11]. From the relation between simulation-based and indistinguishability-based security as well as that between single and multiple ciphertext security in the indistinguishability-based setting as mentioned above, it is immediate that the proposed scheme is also *adaptively strongly partially-hiding* in the *indistinguishability-based* framework against adversaries making an *unbounded* number of queries to both the encryption and the decryption key generation oracles. Thus, our work advances the state of the art in both the fields of simulation-based and indistinguishability-based strongly attribute-hiding PE schemes. Our construction is built in *asymmetric* bilinear groups of *prime* order. The security of our PHPE scheme is derived under the *simultaneous external decisional linear* (SXDLIN) assumption [ACD+12, TAO16], which is a natural extension of the *well-studied decisional linear* (DLIN) assumption in asymmetric bilinear group setting, and as noted in [ACD+12], the two assumptions are in fact *equivalent* in the generic bilinear group model. Nevertheless, our scheme can be readily generalized to one that is secure under the k-LIN assumption.

Similar to [Wee17], we only consider security against a single ciphertext query for the construction presented in this paper to keep the exposition simple. However, we explain in Remark 3.1 how our techniques can be readily extended to design a PHPE scheme that is secure for any a priori bounded number of ciphertexts. Following [DCIJ+13], we present our main construction in the attribute-only mode (i.e., without any actual payload). However, in Section 4 we also provide a key-encapsulation mechanism (KEM) version (i.e., one that uses a symmetric session key as the payload) of our scheme similar to [Wee17]. For the attribute-only version, we design a simulator that runs in polynomial time, and thus this version of our scheme is secure in the standard simulation-based security framework. On the other hand, for the KEM version, similar to Wee [Wee17], our simulator needs to perform a brute force discrete log computation, and thus requires super-polynomial (e.g., sub-exponential) computational power. Nonetheless, this is still stronger than the indistinguishability-based framework [AGVW13, Wee17].

In terms of efficiency, our PHPE scheme is fairly practical. The length of ciphertexts and decryption keys of our scheme grow linearly with the total length of the associated attribute strings and the ABP-size of the associated functions respectively. This is the same as that of [Wee17] except for a constant blow-up, which is common in the literature for semi-adaptive vs adaptive security. Moreover, asymmetric bilinear groups of prime order, which are used for implementing our scheme, are now considered to be both faster and more secure in the cryptographic community following the recent progress in analysing bilinear groups of composite order [Fre10, Gui13] and symmetric bilinear groups instantiated with elliptic curves of small characteristics [BGJT14, GGMZ13, Jou13a, Jou13b].

As a byproduct of our main result, we also obtain the *first simulation-based* adaptively strongly attribute-hiding IPE scheme in asymmetric bilinear groups of prime order under the SXDLIN assumption. The only prior simulation-based strongly attribute-hiding IPE scheme, also due to Wee [Wee17], only achieves semi-adaptive security.

On the technical side, our approach is completely different from that of Wee [Wee17]. More precisely, Wee's technique consists of two steps, namely, first building a private-key scheme, and then bootstrapping it to a public-key one by applying a private-key to public-key compiler similar to [Wee14, CGW15], built on Water's dual system encryption methodology [Wat09]. In contrast, we *directly* construct our scheme in the public-key setting by extending the technique of Okamoto and Takashima [OT12a, OT12b, OT13], a more sophisticated methodology than the dual system encryption originally developed for designing adaptively strongly attribute-hiding IPE schemes in the indistinguishability-based setting, to the scenario of simulation-based adaptively strongly attribute-hiding security for a much expressive predicate class. Also, in order to incorporate the information of the session keys within the ciphertexts in the KEM version of our scheme, we adopt an idea along the lines of the works of Okamoto and Takashima [OT12a, OT12b, OT13], that deviates from that of Wee [Wee17]. Thus, our work further demonstrates the power of the technique introduced by Okamoto and Takashima [OT12a,OT12b,OT13] in achieving very strong security for highly expressive predicate families. We also believe that our work will shed further light on one of the longstanding questions of modern cryptography:

What is the most expressive function or predicate family for which it is possible to construct FE *or strongly attribute-hiding* PE *schemes with adaptive security against adversaries making an unbounded* (*polynomial*) *number of decryption key queries under standard computational assumptions?*

Table 1.1: Current State of the Art in Attribute-Hiding PE

Schemes	Supported Predicates	IND	SIM	Attribute Hiding	Computational Assumptions
[OT10]	IP∘SP	(poly, poly, poly)-AD	$\times$	Weak (IP-part)	DLIN
[OT12a]	IP	(poly, poly, poly)-AD	$\times$	Strong	DLIN
[Agr17]	GC∘IP	(–, poly, bdd)-S-AD	(–, 1, bdd)-S-AD	Strong (IP-part)	LWE
[Wee17]	ABP∘IP	(–, poly, poly)-S-AD	(–, 1, poly)-S-AD	Strong (IP-part)	k-LIN
Ours	ABP∘IP	(poly, poly, poly)-AD	(poly, bdd, poly)-AD	Strong (IP-part)	SXDLIN

The notations used in this table have the following meanings:
- SP: Boolean span programs
- IP: Inner products
- ABP: Arithmetic branching programs
- GC: General polynomial-size circuits
- IND: Indistinguishability-based security
- SIM: Simulation-based security
- AD: Adaptive security
- S-AD: Semi-adaptive security
- poly: Arbitrary polynomial in the security parameter
- bdd: A priori bounded by the public parameters

In this table, (A, B, C) signifies that the adversary is allowed to make B number of ciphertext queries in the relevant security experiment, while A and C number of decryption key queries in the pre- and post-ciphertext phases respectively.

Overview of Our Techniques

We now proceed to explain the key technical ideas underlying our construction. For simplicity, here we will only deal with the IPE scheme, which is a special case of our PHPE construction for $R^{\text{ABP}\circ\text{IP}}$. The proposed PHPE scheme for $R^{\text{ABP}\circ\text{IP}}$ is obtained via a more sophisticated application of the techniques described in this section, and is formally presented in full details in the sequel.

In this overview, we will consider IPE in the attribute-only mode. For IPE, the ciphertext attribute set $\mathcal{X} = \mathbb{F}_q^n$, the decryption key attribute set $\mathcal{Y} = \mathbb{F}_q^n$ for some finite field $\mathbb{F}_q$ and $n \in \mathbb{N}$, and the predicate family is given by $R^{\text{IP}} = \{R^{\text{IP}}(\vec{y}, \cdot) : \mathbb{F}_q^n \to \{0, 1\} \mid \vec{y} \in \mathbb{F}_q^n\}$, where $R^{\text{IP}}(\vec{y}, \vec{z}) = 1$, if $\vec{z} \cdot \vec{y} = 0$, and 0, if $\vec{z} \cdot \vec{y} \neq 0$ for any $\vec{z}, \vec{y} \in \mathbb{F}_q^n$. Observe that the predicate family R^{IP} is subclass of the predicate family $R^{\text{ABP}\circ\text{IP}}$, where we set $n' = 0$, and the component ABP's $f_1, \ldots, f_n$ of a function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ to simply output hardwired constants. In the attribute-only mode, a ciphertext is associated with only a vector $\vec{z} \in \mathbb{F}_q^n$ but no payload, and decryption with a key for some vector $\vec{y} \in \mathbb{F}_q^n$ only reveals the predicate, i.e., whether $\vec{z} \cdot \vec{y} = 0$ or not, but not the exact value of $\vec{z} \cdot \vec{y}$.

Just like [OT12a, OT12b, OT13], we make use of the machinery of the *dual pairing vector spaces* (DPVS) [OT09, OT10]. A highly powerful feature of DPVS is that one can completely or partially hide a linear subspace of the whole vector space by concealing the basis of that subspace or the basis of its dual from the public parameters respectively. In DPVS-based constructions, a pair of mutually dual vector spaces $\mathbb{V}_1$ and $\mathbb{V}_2$, along with a bilinear pairing $e : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{G}_T$

constructed from a standard bilinear group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of prime order q is used. Typically a pair of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$ of the vector spaces $(\mathbb{V}_1, \mathbb{V}_2)$ are generated during setup, using random linear transformations, and a portion of $\mathbb{B}$, say $\widehat{\mathbb{B}}$, is used as the public parameters. Thus, the corresponding segment of $\mathbb{B}^*$, say $\widehat{\mathbb{B}}^*$ remains partially hidden (its dual subspace is disclosed), while the part $\mathbb{B}\backslash\widehat{\mathbb{B}}$ of the basis $\mathbb{B}$ and the corresponding portion $\mathbb{B}^*\backslash\widehat{\mathbb{B}}^*$ of the basis $\mathbb{B}^*$ remain completely hidden to an adversary that is given the public parameters, ciphertexts, and decryption keys. This provides a strong framework for various kinds of information-theoretic tricks in the public-key setting by exploiting various nice properties of linear transformations.

In the proposed IPE scheme, we consider a $(4n+1)$-dimensional DPVS. During setup, we generate a random pair of dual orthonormal bases $(\mathbb{B}, \mathbb{B}^*)$, and use as the public parameters the subset $\widehat{\mathbb{B}}$ consisting of the first n and the last $n+1$ vectors of the basis $\mathbb{B}$, while as the master secret key the corresponding portion of the dual basis $\mathbb{B}^*$. Thus, the linear subspaces spanned by the remaining $2n$ vectors of the bases $\mathbb{B}$ and $\mathbb{B}^*$ are kept completely hidden. Intuitively, we will use the first n-dimensional subspaces of these $2n$-dimensional subspaces for simulating the post-ciphertext decryption key queries, while the latter n-dimensional subspaces for simulating the pre-ciphertext decryption key queries in the ideal experiment. A ciphertext for some vector $\vec{z} \in \mathbb{F}_q^n$ in the proposed scheme has the form $\text{CT} = \boldsymbol{c}$ such that

$$\boldsymbol{c} = (\omega\vec{z}, \vec{0}^n, \vec{0}^n, \vec{0}^n, \varphi)_{\mathbb{B}},$$

where $\omega, \varphi \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and $(\vec{v})_{\mathbb{W}}$ represents the linear combination of the elements of $\mathbb{W}$ with the entries of $\vec{v}$ as coefficients for any $\vec{v} \in \mathbb{F}_q^n$ and any basis $\mathbb{W}$ of DPVS. Similarly, a decryption key corresponding to some vector $\vec{y} \in \mathbb{F}_q^n$ is given by $\text{SK}(\vec{y}) = (\vec{y}, \boldsymbol{k})$ such that

$$\boldsymbol{k} = (\zeta\vec{y}, \vec{0}^n, \vec{0}^n, \vec{\kappa}, 0)_{\mathbb{B}^*},$$

where $\zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$ and $\vec{\kappa} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$. Decryption computes $e(\boldsymbol{c}, \boldsymbol{k})$ to obtain $g_T^{\omega\zeta(\vec{z}\cdot\vec{y})} \in \mathbb{G}_T$, which equals to the identity element of the group $\mathbb{G}_T$, if $\vec{z} \cdot \vec{y} = 0$, and a uniformly random element of $\mathbb{G}_T$, if $\vec{z} \cdot \vec{y} \neq 0$. Observe that this IPE construction is essentially the same as that presented by Okamoto and Takashima in [OT12a]. However, they only proved the strongly attribute-hiding security of this construction in the indistinguishability-based framework, while we prove this construction to be strongly attribute-hiding in the simulation-based framework, by extending their techniques. Let us start with describing our simulation strategy.

In the semi-adaptive case, as considered in [Wee17], the simulation strategy is relatively simple. In fact, for designing an IPE in the semi-adaptive setting, only a $(3n+1)$-dimensional DPVS with n-dimensional hidden subspace would suffice. Note that in the semi-adaptive setting, the adversary is restricted to make the ciphertext query immediately after seeing the public parameters, and there is no pre-ciphertext decryption key query. So, in the semi-adaptive setting, when the adversary makes a ciphertext query, the simulator has no constraint arising from the pre-ciphertext queries of the adversary, and can simply simulate the ciphertext as $\text{CT} = \boldsymbol{c}$ such that

$$\boldsymbol{c} = (\vec{0}^n, (\vec{0}^{n-1}, \tau), \vec{0}^n, \varphi)_{\mathbb{B}},$$

where $\tau, \varphi \xleftarrow{\mathsf{U}} \mathbb{F}_q$, i.e., the simulator puts nothing in the subspace spanned by the public segment of the basis $\mathbb{B}$, and merely puts a random value in a one-dimensional subspace spanned by the hidden segment of the basis. Later, when the adversary queries a decryption key for some vector $\vec{y} \in \mathbb{F}_q^n$, the simulator gets $\vec{y}$ along with the inner product relation of $\vec{y}$ with $\vec{z}$, and the simulator can simply hardwire this information in the corresponding hidden subspace of the decryption key. More precisely, it can simply generate the decryption key as $\text{SK}(\vec{y}) = (\vec{y}, \boldsymbol{k})$ such that

$$\boldsymbol{k} = (\zeta\vec{y}, (\vec{\eta}, \nu), \vec{\kappa}, 0)_{\mathbb{B}^*},$$

where $\zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\eta} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{n-1}$, $\vec{\kappa} \xleftarrow{\mathsf{U}} \mathbb{F}_q^n$, and $\nu = 0$, if $\vec{z} \cdot \vec{y} = 0$, and $\nu \xleftarrow{\mathsf{U}} \mathbb{F}_q$, if $\vec{z} \cdot \vec{y} \neq 0$. Observe that when the simulated ciphertext is decrypted using this simulated decryption key, one obtains the identity element of $\mathbb{G}_T$, or a random element of $\mathbb{G}_T$ according as the inner product relation is satisfied or not, i.e., decryption correctness clearly holds. At this point, please note that the simulator cannot put anything in the subspace of the ciphertext corresponding to the public segment of $\mathbb{B}$, since it must put the actual attribute vectors in the corresponding dual subspace of the decryption keys to ensure correct decryption with other honestly generated ciphertexts.

In the adaptive setting, the situation is much more complex, and we need a $(4n+1)$-dimensional DPVS with $2n$-dimensional hidden subspace. Now, the simulator should also correctly simulate the pre-ciphertext decryption key queries of the adversary. The difference between the pre-ciphertext and post-ciphertext decryption key queries is that unlike the post-ciphertext ones, the information about whether the inner product relation between the associated attribute vector and the attribute vector $\vec{z}$ corresponding to the ciphertext query of the adversary is not supplied when the decryption key is queried. In fact, $\vec{z}$ is not even declared at that time. On the contrary, the information about predicate satisfaction for all the pre-ciphertext decryption key vectors become available to the simulator when the ciphertext query is made by the adversary. The main hurdle for the simulator is to compactly embed this huge amount of information (note that we are considering an unbounded number of pre-ciphertext decryption key queries) in the simulated ciphertext, so that when the simulated ciphertext is decrypted using any pre-ciphertext decryption key, one should get the proper information about predicate satisfaction.

Towards overcoming this difficulty, we observe that it has already been demonstrated by O'Neill [O'N10] that the inner-product predicate family is *pre-image samplable*, i.e., given a set of vectors and their inner-product relation with another fixed vector (but not the vector itself), one can efficiently sample a vector that satisfies all those inner-product relations with high probability. To simulate the ciphertext queried by the adversary, our simulator does exactly this, i.e., it samples a vector $\vec{s} \in \mathbb{F}_q^n$ that has the same inner-product relations as the original queried ciphertext attribute vector $\vec{z}$ with all the attribute vectors corresponding to the pre-ciphertext decryption key queries of the adversary. However, $\vec{s}$ may not have the same inner-product relation as $\vec{z}$ with the attribute vectors corresponding to the post-ciphertext decryption key queries. Therefore, it cannot be embedded in the hidden subspace of the ciphertext devoted for handling the post-ciphertext decryption key queries. Therefore, the simulator needs another n-dimensional subspace to embed $\vec{s}$. Thus, the simulator simulates the queried ciphertext as $\text{CT} = \boldsymbol{c}$ such that

$$\boldsymbol{c} = (\vec{0}^n, (\vec{0}^{n-1}, \tau), \theta\vec{s}, \vec{0}^n, \varphi)_{\mathbb{B}},$$

where $\theta \xleftarrow{\mathsf{U}} \mathbb{F}_q$. On the other hand, it simulates a decryption key corresponding to some vector $\vec{y} \in \mathbb{F}_q^n$ as $\text{SK}(\vec{y}) = (\vec{y}, \boldsymbol{k})$ such that

$$\boldsymbol{k} = \begin{cases} (\zeta\vec{y}, \vec{0}^n, \widehat{\zeta}\vec{y}, \vec{\kappa}, 0)_{\mathbb{B}^*} & \text{(pre-ciphertext)}, \\ (\zeta\vec{y}, (\vec{\eta}, \nu), \vec{0}^n, \vec{\kappa}, 0)_{\mathbb{B}^*} & \text{(post-ciphertext)}, \end{cases}$$

where $\zeta, \widehat{\zeta} \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

Here, we would like to emphasize that while we make use of the pre-image samplability property introduced by O'Neill [O'N10] to design our simulator, our result is not a mere special case of the result that O'Neill obtained using that property. Specifically, O'Neill showed that indistinguishability-based and simulation-based security notions are equivalent in case of FE schemes for function families which are pre-image samplable, provided the adversary is constrained from making any decryption key query after making a ciphertext query. His result does not apply if the adversary is allowed to make decryption key queries even after making ciphertext queries, as is the case in this paper. Moreover, note that there is no known PE scheme for the predicate family $R^{\text{ABP}\circ\text{IP}}$, the actual focus of this paper, even with indistinguishability-based strongly partially-hiding security against adversaries that are allowed to make decryption key queries prior to making ciphertext queries.

Let us continue with the technical overview. It remains to argue that the above simulated forms of ciphertexts and decryption keys are indistinguishable from their real forms. In order to accomplish these changes, we design elaborate hybrid transitions over different forms of ciphertext and decryption keys. In fact, the $2n$-dimensional hidden subspace not only allows us to simulate the pre-ciphertext and post-ciphertext queries differently, but are also crucially leveraged to realize the various forms of ciphertext and decryption keys throughout our hybrid transitions. The hybrid transitions are alternatively computational and information-theoretic. Also, note that not only our simulation strategy for pre-ciphertext and post-ciphertext decryption key queries are different, rather in our hybrid transitions, we handle the pre-ciphertext and post-ciphertext decryption key queries differently, and thereby achieve a security loss that is only proportional to the number of pre-ciphertext decryption key queries.

We start by changing the pre-ciphertext decryption keys to their simulated form. For making these changes, we use the first n-dimensional subspace of the $2n$-dimensional hidden subspace as the *working space*, where we generate the simulated components, and the next n-dimensional subspace as the *storing space*, where we transfer and store the simulated components once they are generated. Note that in the simulated pre-ciphertext decryption keys, the additional simulated components are placed in the second n-dimensions subspace of the $2n$-dimensional hidden subspace. For the hybrid transitions of this part, we make use of the first two of the three types of information-theoretic tricks, namely, Type I, Type II, and Type III introduced in [OT12a, OT12b, OT13], in conjunction with the three types of computational tricks based on the SXDLIN assumption also used in those works. The Type I trick is to apply a linear transformation inside a hidden subspace on the ciphertext side, while the more complex Type II trick is to apply a linear transformation inside a hidden subspace on the ciphertext side preserving the predicate relation with the entries in the corresponding dual subspace of a specific decryption key.

After the transformation of the pre-ciphertext queries is completed, we turn our attention to vanish the component of the ciphertext in the subspace spanned by the public portion of the basis $\mathbb{B}$. For doing this, we apply one of the three computational tricks followed by a Type III information-theoretic trick, which amounts to applying a linear transformation across a hidden and a partially public subspace on both the ciphertext and decryption key sides. While, this enables us to achieve our target for the ciphertext, the forms of the pre-ciphertext decryption keys get distorted. To bring the pre-ciphertext decryption keys to their correct simulated form, we then apply an extension of one of the computational tricks mentioned above.

Once the component in the public subspace of the ciphertext is vanished and pre-ciphertext decryption keys are brought back to their correct simulated form, we turn our attention to the post-cipertext decryption keys. Note that the Type III trick applied for the ciphertext, has already altered the forms of the post-ciphertext queries to something else. Starting with these modified forms, we apply a more carefully crafted variant of the Type II information-theoretic trick, followed by another computational trick based on the SXDLIN assumption to alter the post-ciphertext decryption keys to their simulated forms. This step is reminiscent of the *one-dimensional localization of the inner-product values* used in [OT13]. This step also alters the ciphertext to its simulated form. At this point we arrive at the simulated experiment, and our security analysis gets complete.

2 Preliminaries

In this section we present the backgrounds required for the rest of this paper.

2.1 Notations

Let $\lambda \in \mathbb{N}$ denotes the security parameter and 1^λ be its unary encoding. Let $\mathbb{F}_q$ for any prime $q \in \mathbb{N}$, denotes the finite field of integers modulo q. For $d \in \mathbb{N}$ and $c \in \mathbb{N} \cup \{0\}$ (with $c < d$), we let $[d] = \{1, \ldots, d\}$ and $[c, d] = \{c, \ldots, d\}$. For any set Z, $z \xleftarrow{\mathsf{U}} Z$ represents the process of uniformly sampling an element z from the set Z, and $\sharp Z$ signifies the size or cardinality of Z. For a probabilistic algorithm $\mathcal{U}$, we denote by $\Pi = \mathcal{U}(\Theta; \Phi)$ the output of $\mathcal{U}$ on input Θ with the content of the random tape being Φ, while by $\Pi \xleftarrow{\mathsf{R}} \mathcal{U}(\Theta)$ the process of sampling Π from the output distribution of $\mathcal{U}$ with a uniform random tape on input Θ. Similarly, for any deterministic algorithm $\mathcal{V}$, we write $\Pi = \mathcal{V}(\Theta)$ to denote the output of $\mathcal{V}$ on input Θ. We use the abbreviation PPT to mean probabilistic polynomial-time. We assume that all the algorithms are given the unary representation 1^λ of the security parameter λ as input and will not write 1^λ explicitly as input of the algorithms when it is clear from the context. For any finite field $\mathbb{F}_q$ and $d \in \mathbb{N}$, let $\vec{v}$ denotes the (row) vector $(v_1, \ldots, v_d) \in \mathbb{F}_q^d$, where $v_i \in \mathbb{F}_q$ for all $i \in [d]$. The all zero vectors in $\mathbb{F}_q^d$ will be denoted by $\vec{0}^d$. For any two vectors $\vec{v}, \vec{w} \in \mathbb{F}_q^d$, $\vec{v} \cdot \vec{w}$ stands for the inner product of the vectors $\vec{v}$ and $\vec{w}$, i.e., $\vec{v} \cdot \vec{w} = \sum\limits_{i \in [d]} v_i w_i \in \mathbb{F}_q$. For any multiplicative cyclic group $\mathbb{G}$ of order q and any generator $g \in \mathbb{G}$, let $\boldsymbol{v}$ represents a d-dimensional (row) vector of group elements, i.e., $\boldsymbol{v} = (g^{v_1}, \ldots, g^{v_d}) \in \mathbb{G}^d$ for some $d \in \mathbb{N}$, where $\vec{v} = (v_1, \ldots, v_d) \in \mathbb{F}_q^d$. We use $\boldsymbol{M} = (m_{i,k})$ to represent a matrix with entries $m_{i,k} \in \mathbb{F}_q$. By $\boldsymbol{M}^\intercal$ we will signify the transpose of the matrix $\boldsymbol{M}$. The determinant of a matrix $\boldsymbol{M}$ is denoted by $\det(\boldsymbol{M})$. Let $\mathsf{GL}(d, \mathbb{F}_q)$ denotes the set of all $d \times d$ invertible matrices over $\mathbb{F}_q$. A function $\mathsf{negl} : \mathbb{N} \to \mathbb{R}^+$ is said to be *negligible* if for every $c \in \mathbb{N}$, there exists $T \in \mathbb{N}$ such that for all $\lambda \in \mathbb{N}$ with $\lambda > T$, $|\mathsf{negl}(\lambda)| < 1/\lambda^c$.

2.2 Arithmetic Branching Programs

A *branching program* (BP) Γ is defined by a 5-tuple $\Gamma = (V, E, v_0, v_1, \phi)$, where (V, E) is a directed acyclic graph, $v_0, v_1 \in V$ are two special vertices called the source and the sink respectively, and ϕ is a labeling function for the edges in E. An *arithmetic branching program* (ABP) Γ over a finite field $\mathbb{F}_q$ computes a function $f : \mathbb{F}_q^d \to \mathbb{F}_q$ for some $d \in \mathbb{N}$. In this case, the labeling function ϕ assigns to each edge in E either a degree one polynomial function in one of the input variables with coefficients in $\mathbb{F}_q$ or a constant in $\mathbb{F}_q$. Let $\wp$ be the set of all v_0-v_1 paths in Γ. The output of the function f computed by the ABP Γ on some input $\vec{w} = (w_1, \ldots, w_d) \in \mathbb{F}_q^d$ is defined as $f(\vec{w}) = \sum\limits_{P \in \wp} \left[\prod\limits_{e \in P} \phi(e)|_{\vec{w}} \right]$, where for any $e \in E$, $\phi(e)|_{\vec{w}}$ represents the evaluation of the function $\phi(e)$ at $\vec{w}$. We refer to $\sharp V + \sharp E$ as the size of the ABP Γ. Ishai and Kushilevitz [IK97,IK02] showed how to relate the computation performed by an ABP to the computation of the determinant of a matrix.

Lemma 2.1 ([IK02]): *Given an* ABP $\Gamma = (V, E, v_0, v_1, \phi)$ *computing a function* $f : \mathbb{F}_q^d \to \mathbb{F}_q$*, we can efficiently and deterministically compute a function* $\boldsymbol{L}$ *mapping an input* $\vec{w} \in \mathbb{F}_q^d$ *to a* $(\sharp V - 1) \times (\sharp V - 1)$ *matrix* $\boldsymbol{L}(\vec{w})$ *over* $\mathbb{F}_q$ *such that the following holds:*

– $\det(\boldsymbol{L}(\vec{w})) = f(\vec{w})$.
– *Each entry of* $\boldsymbol{L}(\vec{w})$ *is either a degree one polynomial in a single input variable* w_i $(i \in [d])$ *with coefficients in* $\mathbb{F}_q$ *or a constant in* $\mathbb{F}_q$.
– $\boldsymbol{L}(\vec{w})$ *contains only* -1*'s in the second diagonal, i.e., the diagonal just below the main diagonal, and 0's below the second diagonal.*

Specifically, $\boldsymbol{L}$ *is obtained by removing the column corresponding to* v_0 *and the row corresponding to* v_1 *in the matrix* $\boldsymbol{A}_\Gamma - \boldsymbol{I}$*, where* $\boldsymbol{A}_\Gamma$ *is the adjacency matrix for* Γ *and* $\boldsymbol{I}$ *is the identity matrix.*

Note that there is a linear-time algorithm that converts any Boolean formula, Boolean branching program, or arithmetic formula to an ABP with a constant blow-up in the representation size.

Thus, ABP's can be viewed as a stronger computational model than all the others mentioned above.

2.3 The Function Family $\mathcal{F}_{\mathsf{ABP}\circ\mathsf{IP}}^{(q,n',n)}$ and the Algorithm PGB

Here, we formally describe the function family $\mathcal{F}_{\mathrm{ABP}\circ\mathrm{IP}}^{(q,n',n)}$ which our PHPE scheme supports, and an algorithm PGB for this function class that will be used as a sub-routine in our PHPE construction. Parts of this section is taken verbatim from [IW14, Wee17].

▪ The Function Family $\mathcal{F}_{\mathsf{ABP}\circ\mathsf{IP}}^{(q,n',n)}$

The function class $\mathcal{F}_{\mathrm{ABP}\circ\mathrm{IP}}^{(q,n',n)}$, parameterized by a prime q and $n', n \in \mathbb{N}$, contains functions of the form $f : \mathbb{F}_q^{n'} \times \mathbb{F}_q^n \to \mathbb{F}_q$ defined by $f(\vec{x}, \vec{z}) = \sum_{j\in[n]} f_j(\vec{x}) z_j$ for all $\vec{x} = (x_1, \ldots, x_{n'}) \in \mathbb{F}_q^{n'}$ and $\vec{z} = (z_1, \ldots, z_n) \in \mathbb{F}_q^n$, where $f_1, \ldots, f_n : \mathbb{F}_q^{n'} \to \mathbb{F}_q$ are functions computed by some ABP's $\Gamma_1, \ldots, \Gamma_n$ respectively. We will view the input $\vec{x} = (x_1, \ldots, x_{n'})$ as the *public attribute string*, while $\vec{z} = (z_1, \ldots, z_n)$ as the *private attribute string*. Please refer to [Wee17] for some illustrative examples. A simple but crucial property of the function f is that for any $\zeta \in \mathbb{F}_q$ and any $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, we have $f(\vec{x}, \zeta\vec{z}) = \zeta f(\vec{x}, \vec{z})$.

Observe that the function f can itself be realized by an ABP Γ constructed as follows: First, marge the source vertices of all the component ABP's $\{\Gamma_j\}_{j\in[n]}$ together to form a single vertex, and designate it as the source vertex of the ABP Γ. Next, generate a new sink vertex for Γ, and for each $j \in [n]$, connect the sink vertex of the component ABP Γ_j to that newly formed sink vertex with a directed edge labeled with z_j. For ease of notations, we will denote the size of the ABP Γ computing the function f as $m + n + 1$, where 1 corresponds to the sink vertex of Γ, n accounts for the number of edges directed to that sink vertex, and m accounts for the number of other vertices and edges in Γ. Also, note that the ABP Γ can be further modified to another ABP Γ' in which each vertex has at most one outgoing edge having a label of degree one, by replacing each edge e in Γ with a pair of edges labeled 1 and $\phi(e)$ respectively, where ϕ is the labeling function of the ABP Γ. It is clear that the number of vertices in this modified ABP Γ' is $m+n+1$, since Γ' is obtained by adding a fresh vertex for each edge in Γ as a result of replacing each edge in Γ with a pair of edges. Throughout this paper, whenever we will talk about the ABP computing the function f, we will refer to the ABP Γ' just described, unless otherwise specified.

▪ The Algorithm PGB

► **Syntax and Properties**:

PGB$(f; \vec{r})$: PGB is a PPT algorithm takes as input a function $f \in \mathcal{F}_{\mathrm{ABP}\circ\mathrm{IP}}^{(q,n',n)}$, uses randomness $\vec{r} \in \mathbb{F}_q^{m+n-1}$, and outputs a collection of constants $(\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]}) \in \mathbb{F}_q^n \times (\mathbb{F}_q^2)^m$ along with a function $\rho : [m] \to [n']$. Together with some $\vec{x} \in \mathbb{F}_q^{n'}$ and $\vec{z} \in \mathbb{F}_q^n$, this specifies a collection of $n + m$ *shares*

$$(\{z_j + \sigma_j\}_{j\in[n]}, \{\alpha_{j'} x_{\rho(j')} + \gamma_{j'}\}_{j'\in[m]}). \tag{2.1}$$

Here, $m + n + 1$ is the number of vertices in the ABP computing f and ρ is deterministically derived from f.

The algorithm PGB satisfies the following properties:

- **Linearity**: For a fixed $f \in \mathcal{F}_{\mathrm{ABP}\circ\mathrm{IP}}^{(q,n',n)}$, PGB$(f; \cdot)$ computes a linear function of its randomness over $\mathbb{F}_q$.

- **Reconstruction**: There exists a deterministic polynomial-time algorithm REC that on input any $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ and any $\vec{x} \in \mathbb{F}_q^{n'}$, outputs a collection of coefficients $(\{\Omega_j\}_{j\in[n]}, \{\Omega'_{j'}\}_{j'\in[m]}) \in \mathbb{F}_q^n \times \mathbb{F}_q^{n'}$. These coefficients can be used in combination with any set of shares of the form as in Eq. (2.1), computed by combining the output of $\mathsf{PGB}(f)$ with $\vec{x}$ and any $\vec{z} \in \mathbb{F}_q^n$, to recover $f(\vec{x}, \vec{z})$. Moreover, the recovery procedure is linear in the shares used.
- **Privacy**: There exists a PPT simulator SIM such that for all $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}, \vec{x} \in \mathbb{F}_q^{n'}, \vec{z} \in \mathbb{F}_q^n$, the output of SIM on input f, $\vec{x}$, and $f(\vec{x}, \vec{z})$ is identically distributed to the shares obtained by combining the output of $\mathsf{PGB}(f; \vec{r})$ for uniformly random $\vec{r}$, with $\vec{x}$ and $\vec{z}$ as in Eq. (2.1).

► **Instantiation of the Algorithm**: We now sketch an instantiation of the algorithm PGB following [IW14, Wee17]. This instantiation will be utilized in our PHPE construction.

$\mathsf{PGB}(f)$: The algorithm takes as input a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, and proceeds as follows:

1. Let Γ' denotes the ABP computing f as described above. Recall that in the ABP Γ', there are $m+n+1$ vertices, the variables z_j's only appear on edges leading into the sink vertex, and any vertex has at most one outgoing edge with a label of degree one. It first computes the matrix representation $\boldsymbol{L} \in \mathbb{F}_q^{(m+n)\times(m+n)}$ of the ABP Γ' using the efficient algorithm of Lemma 2.1. Then as per Lemma 2.1, the matrix $\boldsymbol{L}$ satisfies the following properties:
   - $\det(\boldsymbol{L}(\vec{x}, \vec{z})) = f(\vec{x}, \vec{z})$ for all $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$.
   - For $j' \in [m]$, each entry in the j'^{th} row of $\boldsymbol{L}$ is either a degree one polynomial function in one (and the same) input variable $x_{\iota'}$ ($\iota' \in [n']$), with coefficients in $\mathbb{F}_q$ or a constant in $\mathbb{F}_q$.
   - $\boldsymbol{L}$ contains only -1's in the second diagonal, and 0's below the second diagonal.
   - The last column of $\boldsymbol{L}$ is $(0, \ldots, 0, z_1, \ldots, z_n)^\intercal$.
   - $\boldsymbol{L}$ has 0's everywhere else in the last n rows.

   It defines the function $\rho : [m] \to [n']$ as $\rho(j') = \iota'$ if the entries of the j'^{th} row of $\boldsymbol{L}$ involves the variable $x_{\iota'}$ for $j' \in [m]$.
2. Next, it chooses $\vec{r} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m+n-1}$, and computes
$$\boldsymbol{L}\begin{pmatrix}\vec{r}^\intercal\\ 1\end{pmatrix} = (\alpha_1 x_{\rho(1)} + \gamma_1, \ldots, \alpha_m x_{\rho(m)} + \gamma_m, z_1 + \sigma_1, \ldots, z_n + \sigma_n)^\intercal.$$
3. It outputs $\big((\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]}), \rho : [m] \to [n']\big)$.

It is straightforward to verify that each of $\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]}$ are indeed linear functions of the randomness $\vec{r}$.

$\mathsf{REC}(f, \vec{x})$: This algorithm takes as input a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ and a vector $\vec{x} \in \mathbb{F}_q^{n'}$. It proceeds as follows:

1. It first executes Step 1 of the algorithm PGB described above to generate the matrix representation $\boldsymbol{L}$ of f.
2. After that, it computes the cofactors of each entry in the last column of $\boldsymbol{L}$. Let $(\{\Omega'_{j'}\}_{j'\in[m]}, \{\Omega_j\}_{j\in[n]}) \in \mathbb{F}_q^{m+n}$ be the collection of all the cofactors in the order of the entries. Note that the first $m+n-1$ columns of $\boldsymbol{L}$ involve only the variables $\{x_{\iota'}\}_{\iota'\in[n']}$. Hence, it can compute all the cofactors using the input $\vec{x}$.
3. It outputs $(\{\Omega_j\}_{j\in[n]}, \{\Omega'_{j'}\}_{j'\in[m]})$.

The output of $\mathsf{REC}(f, \vec{x})$ can be used in conjunction with a collection of shares $(\{z_j + \sigma_j\}_{j\in[n]}, \{\alpha_{j'} x_{\rho(j')} + \gamma_{j'}\}_{j'\in[m]})$ for any $\vec{z} \in \mathbb{F}_q^n$, to compute $f(\vec{x}, \vec{z})$ as
$$f(\vec{x}, \vec{z}) = \sum_{j'\in[m]} \Omega'_{j'}(\alpha_{j'} x_{\rho(j')} + \gamma_{j'}) + \sum_{j\in[n]} \Omega_j(z_j + \sigma_j). \tag{2.2}$$

Observe that the RHS of Eq. (2.2) corresponds to computing $\det(\boldsymbol{L}'(\vec{x}, \vec{z}))$, where the matrix $\boldsymbol{L}'$ is obtained by replacing the last column of the matrix $\boldsymbol{L}$ with the column $(\alpha_1 x_{\rho(1)} +$

$\gamma_1, \ldots, \alpha_m x_{\rho(m)} + \gamma_m, z_1 + \sigma_1, \ldots, z_n + \sigma_n)^\intercal$, where $\boldsymbol{L}$ is the matrix representation of the ABP Γ' computing the function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, obtained by applying the algorithm of Lemma 2.1. Hence, the correctness of Eq. (2.2) follows from the fact that

$$\det(\boldsymbol{L}'(\vec{x}, \vec{z})) = \det(\boldsymbol{L}(\vec{x}, \vec{z})) \begin{vmatrix} 1 & & & r_1 \\ & \ddots & & \vdots \\ & & 1 & r_{m+n-1} \\ & & & 1 \end{vmatrix} = \det(\boldsymbol{L}(\vec{x}, \vec{z})) \cdot 1 = f(\vec{x}, \vec{z}).$$

Here, $\vec{r} = (r_1, \ldots, r_{m+n-1}) \in \mathbb{F}_q^{m+n-1}$ is the randomness used by PGB while generating the constants $(\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]})$. In fact, an augmented version of Eq. (2.2) also holds. More precisely, for any $\Upsilon, \widetilde{\Upsilon} \in \mathbb{F}_q$, we have

$$\Upsilon f(\vec{x}, \vec{z}) = \sum_{j'\in[m]} \Omega'_{j'} \widetilde{\Upsilon}(\alpha_{j'} x_{\rho(j')} + \gamma_{j'}) + \sum_{j\in[n]} \Omega_j (\Upsilon z_j + \widetilde{\Upsilon}\sigma_j). \tag{2.3}$$

This follows by observing that

$$\det(\boldsymbol{L}(\vec{x}, \vec{z})) = \sum_{j\in[n]} \Omega_j z_j, \text{(since the first } m \text{ entries in the last column is 0)}$$

and hence, the RHS of Eq. (2.3) can be written as

$$\begin{aligned} &\widetilde{\Upsilon}\Big[\sum_{j'\in[m]} \Omega'_{j'}(\alpha_{j'} x_{\rho(j')} + \gamma_{j'}) + \sum_{j\in[n]} \Omega_j(z_j + \sigma_j)\Big] + (\Upsilon - \widetilde{\Upsilon}) \sum_{j\in[n]} \Omega_j z_j \\ &= \widetilde{\Upsilon} \det(\boldsymbol{L}'(\vec{x}, \vec{z})) + (\Upsilon - \widetilde{\Upsilon}) \det(\boldsymbol{L}(\vec{x}, \vec{z})) = \Upsilon \det(\boldsymbol{L}(\vec{x}, \vec{z})), \end{aligned}$$

as $\det(\boldsymbol{L}'(\vec{x}, \vec{z})) = \det(\boldsymbol{L}(\vec{x}, \vec{z}))$. This fact will be used to justify the correctness of our PHPE construction.

$\mathsf{SIM}(f, \vec{x}, \epsilon)$: The simulator takes as input a function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, a vector $\vec{x} \in \mathbb{F}_q^{n'}$, and a value $\epsilon \in \mathbb{F}_q$. It proceeds as follows:

1. At first, it executes Step 1 of the algorithm PGB described above to obtain the matrix representation $\boldsymbol{L}$ of f together with the function $\rho : [m] \to [n']$.
2. Next, it constructs a matrix $\widehat{\boldsymbol{L}}$ from the matrix $\boldsymbol{L}$ by replacing its last column with $(\epsilon, 0, \ldots, 0)^\intercal$.
3. Next, it samples $\vec{r} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m+n-1}$, and computes
$$\widehat{\boldsymbol{L}} \begin{pmatrix} \vec{r}^\intercal \\ 1 \end{pmatrix} = (\mu_1, \ldots, \mu_m, \nu_1, \ldots, \nu_n)^\intercal.$$
4. It outputs $\big((\{\nu_j\}_{j\in[n]}, \{\mu_{j'}\}_{j'\in[m]}), \rho : [m] \to [n']\big)$.

It readily follows from Theorem 3, Corollary 1 of [IW14] that for all $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, $\vec{x} \in \mathbb{F}_q^{n'}$, and $\vec{z} \in \mathbb{F}_q^n$, the output of $\mathsf{SIM}(f, \vec{x}, f(\vec{x}, \vec{z}))$ is identically distributed to the shares obtained by combining with $(\vec{x}, \vec{z})$ the output of $\mathsf{PGB}(f)$ with uniform randomness, thereby establishing the privacy property of the algorithm PGB described above. We omit the details here. Clearly the determinant value of the matrix $\widehat{\boldsymbol{L}}(\vec{x}, \vec{z})$ generated by SIM on input any $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, $\vec{x} \in \mathbb{F}_q^{n'}$, and $f(\vec{x}, \vec{z})$ for any $\vec{z} \in \mathbb{F}_q^n$ is $f(\vec{x}, \vec{z})$.

2.4 Bilinear Groups and Dual Pairing Vector Spaces

In this section, we will provide the necessary backgrounds on bilinear groups and dual pairing vector spaces, which are the primary building blocks of our PHPE construction.

Definition 2.1 (Bilinear Group): A bilinear group $\mathsf{params}_{\mathbb{G}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ is a tuple of a prime integer $q \in \mathbb{N}$; cyclic multiplicative groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order q each with polynomial-time computable group operations; generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$; and a polynomial-time computable non-degenerate bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, i.e., e satisfies the following two properties:

- *Bilinearity*: $e(g_1^{\delta}, g_2^{\hat{\delta}}) = e(g_1, g_2)^{\delta\hat{\delta}}$ for all $\delta, \hat{\delta} \in \mathbb{F}_q$.
- *Non-degeneracy*: $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ denotes the identity element of the group $\mathbb{G}_T$.

A bilinear group is said to be asymmetric if no efficiently computable isomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$. Let $\mathcal{G}_{\text{BPG}}$ be an algorithm that on input the unary encoding 1^{λ} of the security parameter λ, outputs a description $\mathsf{params}_{\mathbb{G}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ of a bilinear group.

Definition 2.2 (Dual Pairing Vector Spaces: DPVS [OT09, OT10]): A dual pairing vector space (DPVS) $\mathsf{params}_{\mathbb{V}} = (q, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, e)$ formed by the direct product of a bilinear group $\mathsf{params}_{\mathbb{G}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e)$ is a tuple of a prime integer q; d-dimensional vector spaces $\mathbb{V}_t = \mathbb{G}_t^d$ over $\mathbb{F}_q$ for $t \in [2]$ under vector addition and scalar multiplication defined componentwise in the usual manner; canonical bases $\mathbb{A}_t = \{\boldsymbol{a}^{(t,\ell)} = (\overbrace{1_{\mathbb{G}_t}, \ldots, 1_{\mathbb{G}_t}}^{\ell-1}, g_t, \overbrace{1_{\mathbb{G}_t}, \ldots, 1_{\mathbb{G}_t}}^{d-\ell})\}_{\ell \in [d]}$ of $\mathbb{V}_t$ for $t \in [2]$, where $1_{\mathbb{G}_t}$ is the identity element of the group $\mathbb{G}_t$ for $t \in [2]$; and a pairing $e : \mathbb{V}_1 \times \mathbb{V}_2 \to \mathbb{G}_T$ defined by $e(\boldsymbol{v}, \boldsymbol{w}) = \prod_{\ell \in [d]} e(g_1^{v_\ell}, g_2^{w_\ell}) \in \mathbb{G}_T$ for all $\boldsymbol{v} = (g_1^{v_1}, \ldots, g_1^{v_d}) \in \mathbb{V}_1$, $\boldsymbol{w} = (g_2^{w_1}, \ldots, g_2^{w_d}) \in \mathbb{V}_2$. Observe that the newly defined map e is also non-degenerate bilinear, i.e., e also satisfies the following two properties:

- *Bilinearity*: $e(\delta\boldsymbol{v}, \widehat{\delta}\boldsymbol{w}) = e(\boldsymbol{v}, \boldsymbol{w})^{\delta\widehat{\delta}}$ for all $\delta, \widehat{\delta} \in \mathbb{F}_q$, $\boldsymbol{v} \in \mathbb{V}_1$, and $\boldsymbol{w} \in \mathbb{V}_2$.
- *Non-degeneracy*: If $e(\boldsymbol{v}, \boldsymbol{w}) = 1_{\mathbb{G}_T}$ for all $\boldsymbol{w} \in \mathbb{V}_2$, then $\boldsymbol{v} = (\overbrace{1_{\mathbb{G}_1}, \ldots, 1_{\mathbb{G}_1}}^{d})$. Similar statement also holds with the vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ interchanged.

For any ordered basis $\mathbb{W} = \{\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(d)}\}$ of $\mathbb{V}_t$ for $t \in [2]$, and any vector $\vec{v} \in \mathbb{F}_q^d$, let $(\vec{v})_{\mathbb{W}}$ represents the vector in $\mathbb{V}_t$ formed by the linear combination of the members of $\mathbb{W}$ with the components of $\vec{v}$ as the coefficients, i.e., $(\vec{v})_{\mathbb{W}} = \sum_{\ell \in [d]} v_\ell \boldsymbol{w}^{(\ell)} \in \mathbb{V}_t$. The DPVS generation algorithm $\mathcal{G}_{\text{DPVS}}$ takes as input the unary encoded security parameter 1^{λ}, a dimension value $d \in \mathbb{N}$, along with a bilinear group $\mathsf{params}_{\mathbb{G}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{BPG}}()$, and outputs a description $\mathsf{params}_{\mathbb{V}} = (q, \mathbb{V}_1, \mathbb{V}_2, \mathbb{G}_T, \mathbb{A}_1, \mathbb{A}_2, e)$ of DPVS with d-dimensional $\mathbb{V}_1$ and $\mathbb{V}_2$.

We now describe random *dual orthonormal basis* generator $\mathcal{G}_{\text{OB}}$ [OT09, OT10] in Fig. 2.1. This algorithm will be utilized as a sub-routine in our PHPE construction.

2.5 Complexity Assumption

For realizing our PHPE construction in asymmetric bilinear groups, we rely on the natural extension of the well-studied decisional linear (DLIN) assumption to the asymmetric bilinear group setting, called the external decisional linear (XDLIN) assumption.

Assumption (External Decisional Linear: XDLIN [ACD+12, TAO16]): For $t \in [2]$, the XDLIN_t problem is to guess the bit $\widehat{\beta} \xleftarrow{\mathsf{U}} \{0,1\}$ given $\varrho_{\widehat{\beta}}^{\mathsf{XDLIN}_t} = (\mathsf{params}_{\mathbb{G}}, g_1^{\varpi}, g_1^{\xi}, g_1^{\varkappa\varpi}, g_1^{\varsigma\xi}, g_2^{\varpi}, g_2^{\xi},$

$\mathcal{G}_{\text{OB}}(N,(d_0,\ldots,d_N))$: This algorithm takes as input the unary encoded security parameter 1^λ, a number $N \in \mathbb{N}$, and the respective dimensions $d_0,\ldots,d_N \in \mathbb{N}$ of the $N+1$ pairs of bases to be generated. It executes the following operations:

1. It first generates $\mathsf{params}_{\mathbb{G}} = (q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,g_1,g_2,e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{BPG}}()$.
2. Next, it samples $\psi \xleftarrow{\mathsf{U}} \mathbb{F}_q\backslash\{0\}$ and computes $g_T = e(g_1,g_2)^\psi$.
3. Then, for $\imath \in [0,N]$, it performs the following:
   (a) It constructs $\mathsf{params}_{\mathbb{V}_\imath} = (q,\mathbb{V}_{\imath,1},\mathbb{V}_{\imath,2},\mathbb{G}_T,\mathbb{A}_{\imath,1},\mathbb{A}_{\imath,2},e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{DPVS}}(d_\imath,\mathsf{params}_{\mathbb{G}})$.
   (b) It samples $\boldsymbol{B}^{(\imath)} = (b^{(\imath)}_{\ell,k}) \xleftarrow{\mathsf{U}} \mathsf{GL}(d_\imath,\mathbb{F}_q)$.
   (c) It computes $\boldsymbol{B}^{*(\imath)} = (b^{*(\imath)}_{\ell,k}) = \psi((\boldsymbol{B}^{(\imath)})^{-1})^\intercal$.
   (d) For all $\ell \in [d_\imath]$, let $\vec{b}^{(\imath,\ell)}$ and $\vec{b}^{*(\imath,\ell)}$ represent the ℓ^{th} rows of $\boldsymbol{B}^{(\imath)}$ and $\boldsymbol{B}^{*(\imath)}$ respectively. It computes $\boldsymbol{b}^{(\imath,\ell)} = (\vec{b}^{(\imath,\ell)})_{\mathbb{A}_{\imath,1}}, \boldsymbol{b}^{*(\imath,\ell)} = (\vec{b}^{*(\imath,\ell)})_{\mathbb{A}_{\imath,2}}$ for $\ell \in [d_\imath]$, and sets
   $$\mathbb{B}_\imath = \{\boldsymbol{b}^{(\imath,1)},\ldots,\boldsymbol{b}^{(\imath,d_\imath)}\}, \mathbb{B}^*_\imath = \{\boldsymbol{b}^{*(\imath,1)},\ldots,\boldsymbol{b}^{*(\imath,d_\imath)}\}.$$
   Clearly $\mathbb{B}_\imath$ and $\mathbb{B}^*_\imath$ form bases of the vector spaces $\mathbb{V}_{\imath,1}$ and $\mathbb{V}_{\imath,2}$ respectively. Also, note that $\mathbb{B}_\imath$ and $\mathbb{B}^*_\imath$ are dual orthonormal in the sense that for all $\ell,\ell' \in [d_\imath]$,
   $$e(\boldsymbol{b}^{(\imath,\ell)},\boldsymbol{b}^{*(\imath,\ell')}) = \begin{cases} g_T, & \text{if } \ell = \ell', \\ 1_{\mathbb{G}_T}, & \text{otherwise.} \end{cases}$$
4. Next, it sets $\mathsf{params} = (\{\mathsf{params}_{\mathbb{V}_\imath}\}_{\imath\in[0,N]}, g_T)$.
5. It returns $(\mathsf{params}, \{\mathbb{B}_\imath,\mathbb{B}^*_\imath\}_{\imath\in[0,N]})$.

Fig. 2.1: Dual Orthonormal Basis Generator $\mathcal{G}_{\text{OB}}$

$g_2^{\varkappa\varpi}, g_2^{\varsigma\xi}, \Re_{t,\widehat{\beta}})$, where

$$\mathsf{params}_{\mathbb{G}} = (q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,g_1,g_2,e) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{BPG}}();$$
$$\varpi,\xi,\varkappa,\varsigma,\varepsilon \xleftarrow{\mathsf{U}} \mathbb{F}_q;$$
$$\Re_{t,0} = g_t^{(\varkappa+\varsigma)}, \Re_{t,1} = g_t^{(\varkappa+\varsigma)+\varepsilon}.$$

The XDLIN_t assumption states that for any PPT algorithm $\mathcal{E}$, for any security parameter λ, the advantage of $\mathcal{E}$ in deciding the XDLIN_t problem, defined as

$$\mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLIN}_t}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{E}(\varrho_0^{\mathsf{XDLIN}_t})] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{E}(\varrho_1^{\mathsf{XDLIN}_t})]|,$$

is negligible in λ, i.e., $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{XDLIN}_t}(\lambda) \le \mathsf{negl}(\lambda)$, where negl is some negligible function. The simultaneous XDLIN (SXDLIN) assumption states that both XDLIN_1 and XDLIN_2 assumptions hold at the same time. For any security parameter λ, we denote the advantage of any probabilistic algorithm $\mathcal{E}$ against SXDLIN as $\mathsf{Adv}_{\mathcal{E}}^{\mathsf{SXDLIN}}(\lambda)$.

2.6 The Notion of Partially-Hiding Predicate Encryption

Here, we formally present the syntax and simulation-based security notion of a partially-hiding predicate encryption (PHPE) scheme for the function family $\mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ for some prime q and $n',n \in \mathbb{N}$. Following [Wee17], we define the $\mathsf{ABP}\circ\mathsf{IP}$ predicate family $R^{\text{ABP}\circ\text{IP}}$ as $R^{\text{ABP}\circ\text{IP}} = \{R^{\text{ABP}\circ\text{IP}}(f,(\cdot,\cdot)) : \mathbb{F}_q^{n'} \times \mathbb{F}_q^n \to \{0,1\} \mid f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}\}$, where $R^{\text{ABP}\circ\text{IP}}(f,(\vec{x},\vec{z})) = 1$, if $f(\vec{x},\vec{z}) = 0$, and $R^{\text{ABP}\circ\text{IP}}(f,(\vec{x},\vec{z})) = 0$, if $f(\vec{x},\vec{z}) \neq 0$ for all $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ and $(\vec{x},\vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$.

► **Syntax**: An *attribute-only/key-encapsulation mechanism* (KEM) *partially-hiding predicate encryption* (PHPE) scheme for the function family $\mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ consists of the following polynomial-time algorithms:

$\mathsf{PHPE.Setup}(1^{n'}, 1^n)$: The setup algorithm takes as input the security parameter λ along with the public and private attribute lengths n' and n respectively (all encoded in unary). It outputs the public parameters $\textsc{mpk}$ and the master secret key $\textsc{msk}$.

$\mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}, \vec{z}))$: The encryption algorithm takes as input the public parameters $\textsc{mpk}$, a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$. It outputs a ciphertext $\textsc{ct}$. In the KEM mode, it additionally outputs a session key $\textsc{kem}$.

$\mathsf{PHPE.KeyGen}(\textsc{mpk}, \textsc{msk}, f)$: On input the public parameters $\textsc{mpk}$, the master secret key $\textsc{msk}$, along with a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, the key generation algorithm outputs a decryption key $\textsc{sk}(f)$.

$\mathsf{PHPE.Decrypt}(\textsc{mpk}, (f, \textsc{sk}(f)), (\vec{x}, \textsc{ct}))$: The decryption algorithm takes as input the public parameters $\textsc{mpk}$, a pair of a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ and a decryption key $\textsc{sk}(f)$ for f, along with a pair of a public attribute $\vec{x} \in \mathbb{F}_q^{n'}$ and a ciphertext $\textsc{ct}$ associated with $\vec{x}$ and some private attribute string. In the attribute-only mode, it outputs either 1 or 0, while in the KEM mode, it outputs a session key $\widetilde{\textsc{kem}}$. For notational convenience, we will think of f and $\vec{x}$ as parts of $\textsc{sk}(f)$ and $\textsc{ct}$ respectively, and will not write them explicitly in the argument of $\mathsf{PHPE.Decrypt}$.

The algorithm $\mathsf{PHPE.Decrypt}$ is deterministic, while all the others are probabilistic.

► **Correctness**: A PHPE scheme for the function family $\mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is said to be correct if for any security parameter λ, any $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, any $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, any $(\textsc{mpk}, \textsc{msk}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Setup}(1^{n'}, 1^n)$, and any $\textsc{sk}(f) \xleftarrow{\mathsf{R}} \mathsf{PHPE.KeyGen}(\textsc{mpk}, \textsc{msk}, f)$, the following holds:

– (*Authorized*) If $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 1$, then

$$\begin{aligned}&\mathsf{Pr}[\mathsf{PHPE.Decrypt}(\textsc{mpk}, \textsc{sk}(f), \textsc{ct}) = 1 : \textsc{ct} \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}, \vec{z}))]\\ &\geq 1 - \mathsf{negl}(\lambda) \text{ (attribute-only mode)},\\ &\mathsf{Pr}[\mathsf{PHPE.Decrypt}(\textsc{mpk}, \textsc{sk}(f), \textsc{ct}) = \textsc{kem} : (\textsc{ct}, \textsc{kem}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}, \vec{z}))]\\ &\geq 1 - \mathsf{negl}(\lambda) \text{ (}\mathsf{KEM}\text{ mode)}.\end{aligned}$$

– (*Unauthorized*) If $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 0$, then

$$\begin{aligned}&\mathsf{Pr}[\mathsf{PHPE.Decrypt}(\textsc{mpk}, \textsc{sk}(f), \textsc{ct}) = 0 : \textsc{ct} \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}, \vec{z}))]\\ &\geq 1 - \mathsf{negl}(\lambda) \text{ (attribute-only mode)},\\ &\mathsf{Pr}[\mathsf{PHPE.Decrypt}(\textsc{mpk}, \textsc{sk}(f), \textsc{ct}) \neq \textsc{kem} : (\textsc{ct}, \textsc{kem}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}, \vec{z}))]\\ &\geq 1 - \mathsf{negl}(\lambda) \text{ (}\mathsf{KEM}\text{ mode)}.\end{aligned}$$

Here, negl is some negligible function, and the probabilities are taken over the random coins of $\mathsf{PHPE.Encrypt}$.

► **Simulation-Based Security**: The simulation-based adaptively strongly partially-hiding security notion for a PHPE scheme is formulated by considering the following two experiments involving a stateful probabilistic adversary $\mathcal{A}$ and a stateful probabilistic simulator $\mathcal{S}$:

$\mathsf{Exp}^{\mathsf{PHPE,REAL}}_{\mathcal{A}}(\lambda)$:

1. $(\textsc{mpk}, \textsc{msk}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Setup}(1^{n'}, 1^n)$.
2. $\{(\vec{x}^{(i)}, \vec{z}^{(i)})\}_{i \in [q_{\textsc{ct}}]} \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathsf{PHPE.KeyGen}(\textsc{msk}, \cdot)}(\textsc{mpk})$.
3. (a) (attribute-only case) $\textsc{ct}^{(i)} \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}^{(i)}, \vec{z}^{(i)}))$ for $i \in [q_{\textsc{ct}}]$.

(b) (KEM case) $(\textsc{ct}^{(i)}, \textsc{kem}^{(i)}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\textsc{mpk}, (\vec{x}^{(i)}, \vec{z}^{(i)}))$ for $i \in [q_{\textsc{ct}}]$.

4. (a) (attribute-only case) $\Im \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathsf{PHPE.KeyGen}(\textsc{msk},\cdot)}(\textsc{mpk}, \{\textsc{ct}^{(i)}\}_{i\in[q_{\textsc{ct}}]})$.

   (b) (KEM case) $\Im \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathsf{PHPE.KeyGen}(\textsc{msk},\cdot)}(\textsc{mpk}, \{(\textsc{ct}^{(i)}, \textsc{kem}^{(i)})\}_{i\in[q_{\textsc{ct}}]})$.

5. Output $\varrho_{\mathcal{A}}^{\mathsf{PHPE,REAL}} = \big(\textsc{mpk}, \{(\vec{x}^{(i)}, \vec{z}^{(i)})\}_{i\in[q_{\textsc{ct}}]}, \Im\big)$.

$\mathsf{Exp}_{\mathcal{A},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)$:

1. $\textsc{mpk} \xleftarrow{\mathsf{R}} \mathcal{S}(1^{n'}, 1^{n})$.
2. $\{(\vec{x}^{(i)}, \vec{z}^{(i)})\}_{i\in[q_{\textsc{ct}}]} \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathcal{S}(\cdot)}(\textsc{mpk})$.
3. (a) (attribute-only case) $\{\textsc{ct}^{(i)}\}_{i\in[q_{\textsc{ct}}]} \xleftarrow{\mathsf{R}} \mathcal{S}(q_{\textsc{ct}}, \{(\vec{x}^{(i)}, R^{\textsc{abp}\circ\textsc{ip}}(f_h, (\vec{x}^{(i)}, \vec{z}^{(i)})))\}_{i\in[q_{\textsc{ct}}],h\in[q_{\textsc{key-pre}}]})$.

   (b) (KEM case) $\{\textsc{kem}^{(i)}\}_{i\in[q_{\textsc{ct}}]} \xleftarrow{\mathsf{U}} \mathbb{K}$, where $\mathbb{K}$ = session key space $\{\textsc{ct}^{(i)}\}_{i\in[q_{\textsc{ct}}]} \xleftarrow{\mathsf{R}} \mathcal{S}(q_{\textsc{ct}}, \{(\vec{x}^{(i)}, \overline{\textsc{kem}}^{(i,h)})\}_{i\in[q_{\textsc{ct}}],h\in[q_{\textsc{key-pre}}]})$, where for all $i \in [q_{\textsc{ct}}], h \in [q_{\textsc{key-pre}}]$, $\overline{\textsc{kem}}^{(i,h)} = \textsc{kem}^{(i)}$, if $R^{\textsc{abp}\circ\textsc{ip}}(f_h, (\vec{x}^{(i)}, \vec{z}^{(i)})) = 1$, and $\perp$, if $R^{\textsc{abp}\circ\textsc{ip}}(f_h, (\vec{x}^{(i)}, \vec{z}^{(i)})) = 0$.

4. (a) (attribute-only case) $\Im \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathcal{S}^{\mathcal{O}_{R^{\textsc{abp}\circ\textsc{ip}}}(\{(\vec{x}^{(i)},\vec{z}^{(i)})\}_{i\in[q_{\textsc{ct}}]},\cdot)}(\cdot)}(\textsc{mpk}, \{\textsc{ct}^{(i)}\}_{i\in[q_{\textsc{ct}}]})$.

   (b) (KEM case) $\Im \xleftarrow{\mathsf{R}} \mathcal{A}^{\mathcal{S}^{\mathcal{O}_{R^{\textsc{abp}\circ\textsc{ip}}}(\{((\vec{x}^{(i)},\vec{z}^{(i)}),\textsc{kem}^{(i)})\}_{i\in[q_{\textsc{ct}}]},\cdot)}(\cdot)}(\textsc{mpk}, \{(\textsc{ct}^{(i)}, \textsc{kem}^{(i)})\}_{i\in[q_{\textsc{ct}}]})$.

5. Output $\varrho_{\mathcal{A},\mathcal{S}}^{\mathsf{PHPE,IDEAL}} = \big(\textsc{mpk}, \{(\vec{x}^{(i)}, \vec{z}^{(i)})\}_{i\in[q_{\textsc{ct}}]}, \Im\big)$.

Here, the simulator $\mathcal{S}$ accepts as input a function $f \in \mathcal{F}_{\textsc{abp}\circ\textsc{ip}}^{(q,n',n)}$ when it acts as an oracle to $\mathcal{A}$. Also, $q_{\textsc{ct}}$ and $q_{\textsc{key-pre}}$ respectively denotes the number of ciphertext queries made by $\mathcal{A}$ and number of decryption key queries made by $\mathcal{A}$ prior to submitting the ciphertext queries. Further, in the attribute-only case, the oracle $\mathcal{O}_{R^{\textsc{abp}\circ\textsc{ip}}}$ receives as its second argument a function $f \in \mathcal{F}_{\textsc{abp}\circ\textsc{ip}}^{(q,n',n)}$, and outputs $\{R^{\textsc{abp}\circ\textsc{ip}}(f, (\vec{x}^{(i)}, \vec{z}^{(i)}))\}_{i\in[q_{\textsc{ct}}]}$. On the other hand, in the KEM case, the oracle $\mathcal{O}_{R^{\textsc{abp}\circ\textsc{ip}}}$ takes as its second argument a function $f \in \mathcal{F}_{\textsc{abp}\circ\textsc{ip}}^{(q,n',n)}$, and outputs $\textsc{kem}^{(i)}$, if $R^{\textsc{abp}\circ\textsc{ip}}(f, (\vec{x}^{(i)}, \vec{z}^{(i)})) = 1$, and $\perp$, if $R^{\textsc{abp}\circ\textsc{ip}}(f, (\vec{x}^{(i)}, \vec{z}^{(i)})) = 0$ for $i \in [q_{\textsc{ct}}]$. A simulator $\mathcal{S}$ is said to be admissible if on each decryption key query $f \in \mathcal{F}_{\textsc{abp}\circ\textsc{ip}}^{(q,n',n)}$ of $\mathcal{A}$ in the post-ciphertext query phase, $\mathcal{S}$ makes just a single query to the oracle $\mathcal{O}_{R^{\textsc{abp}\circ\textsc{ip}}}$ on f itself. Let the number of decryption key queries made by $\mathcal{A}$ after receiving the queried ciphertexts be $q_{\textsc{key-post}}$.

For any security parameter λ, for any probabilistic distinguisher $\mathcal{D}$, the advantage of $\mathcal{D}$ in distinguishing the above two experiments is defined as

$$\mathsf{Adv}_{\mathcal{D}}^{\mathsf{PHPE,SIM\text{-}AH}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{D}(\varrho_{\mathcal{A}}^{\mathsf{PHPE,REAL}})] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{D}(\varrho_{\mathcal{A},\mathcal{S}}^{\mathsf{PHPE,IDEAL}})]|.$$

Definition 2.3: A PHPE scheme is called $(q_{\textsc{key-pre}}, q_{\textsc{ct}}, q_{\textsc{key-post}})$-simulation-based adaptively strongly partially hiding if there exists an admissible stateful PPT simulator $\mathcal{S}$ such that for any stateful PPT adversary $\mathcal{A}$ making at most $q_{\textsc{ct}}$ ciphertext queries, $q_{\textsc{key-pre}}$ decryption key queries in the pre-ciphertext query phase, while $q_{\textsc{key-post}}$ decryption key queries in the post-ciphertext query phase, any PPT distinguisher $\mathcal{D}$, and any security parameter λ, $\mathsf{Adv}_{\mathcal{D}}^{\mathsf{PHPE,SIM\text{-}AH}}(\lambda) \leq \mathsf{negl}(\lambda)$, where negl is some negligible function. Also, a PHPE scheme is said to be $(\mathsf{poly}, q_{\textsc{ct}}, \mathsf{poly})$-simulation-based adaptively strongly partially hiding if it is $(q_{\textsc{key-pre}}, q_{\textsc{ct}}, q_{\textsc{key-post}})$-simulation-based adaptively strongly partially hiding as well as $q_{\textsc{key-pre}}$ and $q_{\textsc{key-post}}$ are unbounded polynomials in the security parameter λ.

Remark 2.1: Consider an adversary $\mathcal{H}$ that first invokes $\mathcal{A}$ and then invokes $\mathcal{D}$ once the transcript ($\varrho_{\mathcal{A}}^{\mathsf{PHPE,REAL}}$ or $\varrho_{\mathcal{A},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}$) of the experiment is obtained. Consider the experiments $\mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE,REAL}}(\lambda)$ and $\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)$ which are obtained from the experiments $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{PHPE,REAL}}(\lambda)$ and $\mathsf{Exp}_{\mathcal{A},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)$ respectively by applying the corresponding augmentations. Let us define the outputs of the augmented experiments as the output of $\mathcal{H}$ in those experiments, and the advantage of $\mathcal{H}$ as

$$\mathsf{Adv}_{\mathcal{H}}^{\mathsf{PHPE,SIM\text{-}AH}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE,REAL}}(\lambda)] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)]|.$$

Then, clearly $\mathsf{Adv}_{\mathcal{H}}^{\mathsf{PHPE,SIM\text{-}AH}}(\lambda) = \mathsf{Adv}_{\mathcal{D}}^{\mathsf{PHPE,SIM\text{-}AH}}(\lambda)$. We make use of this combined adversary $\mathcal{H}$ as well as the associated augmented experiments $\mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE,REAL}}(\lambda)$ and $\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)$ in the security proof of our PHPE construction, both the attribute-only and KEM versions.

► **Indistinguishability-Based Security**: The indistinguishability-based adaptively strongly partially hiding security of a PHPE scheme is formalized through the experiment $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{PHPE,IND\text{-}AH}}(\lambda)$ between a stateful probabilistic adversary $\mathcal{A}$ and a stateful probabilistic challenger $\mathcal{B}$ described below:

1. $\mathcal{B}$ generates $(\text{MPK}, \text{MSK}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Setup}(1^{n'}, 1^{n})$, and provides MPK to $\mathcal{A}$.
2. $\mathcal{A}$ is allowed to adaptively request any polynomial number of decryption keys to $\mathcal{B}$. In response to a decryption key query of $\mathcal{A}$ for some function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, $\mathcal{B}$ forms corresponding decryption key $\text{SK}(f) \xleftarrow{\mathsf{R}} \mathsf{PHPE.KeyGen}(\text{MPK}, \text{MSK}, f)$ and hands it to $\mathcal{A}$.
3. $\mathcal{A}$ submits to $\mathcal{B}$ a public challenge attribute $\vec{x} \in \mathbb{F}_q^{n'}$ and a pair of private challenge attributes $(\vec{z}^{(0)}, \vec{z}^{(1)}) \in (\mathbb{F}_q^{n})^2$ of its choice, subject to the restriction that $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(0)})) = R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(1)}))$ for all the functions $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ for which $\mathcal{A}$ has queried decryption keys so far, i.e., in other words, for all the functions $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ for which $\mathcal{A}$ has queried decryption keys so far, one of the following holds:
   - $f(\vec{x}, \vec{z}^{(0)}) \neq 0$ and $f(\vec{x}, \vec{z}^{(1)}) \neq 0$
   - $f(\vec{x}, \vec{z}^{(0)}) = 0$ and $f(\vec{x}, \vec{z}^{(1)}) = 0$

   In the attribute-only case, $\mathcal{B}$ selects a random bit $\beta \xleftarrow{\mathsf{U}} \{0,1\}$, and gives $\mathcal{A}$ the ciphertext $\text{CT} \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\text{MPK}, (\vec{x}, \vec{z}^{(\beta)}))$.

   On the other hand, in the KEM case, $\mathcal{A}$ additionally submits an indicator bit $\breve{\beta} \in \{0,1\}$. If $\breve{\beta} = 0$, and $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(0)})) = R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(1)})) = 1$ for the function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ corresponding to one of the decryption key queries that $\mathcal{A}$ has made so far, $\mathcal{B}$ aborts, and in that case the output of the experiment is defined to be 0. Otherwise, if $\breve{\beta} = 0$ and $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(0)})) = R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(1)})) = 0$ for the functions $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ corresponding to all the decryption key queries of $\mathcal{A}$ so far, then $\mathcal{B}$ samples a random bit $\beta \xleftarrow{\mathsf{U}} \{0,1\}$, generates $(\text{CT}, \text{KEM}) \xleftarrow{\mathsf{R}} \mathsf{PHPE.Encrypt}(\text{MPK}, (\vec{x}, \vec{z}^{(\beta)}))$, and also samples another random session key $\widehat{\text{KEM}}$ from the session key space. If $\beta = 0$, then $\mathcal{B}$ hands (CT, KEM) to $\mathcal{A}$, while if $\beta = 1$, $\mathcal{B}$ gives $(\text{CT}, \widehat{\text{KEM}})$ to $\mathcal{A}$. Else, if $\breve{\beta} = 1$, then $\mathcal{B}$ selects a random bit $\beta \xleftarrow{\mathsf{U}} \{0,1\}$, and provides $\mathcal{A}$ with $(\text{CT}, \text{KEM}) \xleftarrow{\mathsf{U}} \mathsf{PHPE.Encrypt}(\text{MPK}, (\vec{x}, \vec{z}^{(\beta)})$.
4. $\mathcal{A}$ may continue adaptively to request any polynomial number of additional decryption keys for functions in $\mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ subject to the same restriction as above. In the attribute-only case, $\mathcal{B}$ simply gives the corresponding decryption keys to $\mathcal{A}$. On the other hand, in the KEM case, if the indicator bit $\breve{\beta} \in \{0,1\}$ submitted by $\mathcal{A}$ in Step 3 above is $\breve{\beta} = 0$, and $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(0)})) = R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z}^{(1)})) = 1$ for the function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ that $\mathcal{A}$ queries decryption key for, then $\mathcal{B}$ aborts, and the output of the experiment is defined to be 0 in such situation. Otherwise, $\mathcal{B}$ provides $\mathcal{A}$ with the corresponding decryption key.
5. Eventually, $\mathcal{A}$ outputs a guess bit β'. The output of the experiment is defined to be 1, if $\beta' = \beta$, and 0, otherwise.

For any security parameter λ, the advantage of any probabilistic adversary $\mathcal{A}$ in $\mathsf{Exp}_{\mathcal{A}}^{\mathsf{PHPE,IND\text{-}AH}}(\lambda)$ is defined by

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PHPE,IND\text{-}AH}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathsf{Exp}_{\mathcal{A}}^{\mathsf{PHPE,IND\text{-}AH}}(\lambda)] - 1/2|.$$

Definition 2.4: A PHPE scheme is said to be $(\mathsf{poly}, 1, \mathsf{poly})$-indistinguishability-based adaptively strongly partially hiding if for any stateful PPT adversary $\mathcal{A}$, for any security parameter λ, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PHPE,IND\text{-}AH}}(\lambda) \leq \mathsf{negl}(\lambda)$, where negl is some negligible function, and $(\mathsf{poly}, 1, \mathsf{poly})$ refers to the fact that the adversary $\mathcal{A}$ can make a single ciphertext query, but is allowed to make any polynomial number of decryption key queries both before and after the ciphertext query during the above experiment. If the adversary is allowed to make any polynomial number of ciphertext queries in the above experiment, the resulting security would be called $(\mathsf{poly}, \mathsf{poly}, \mathsf{poly})$-indistinguishability-based adaptively strongly partially-hiding security.

Remark 2.2: It follows from existing results on indistinguishability-based security for FE, e.g., [BSW11], that $(\mathsf{poly}, 1, \mathsf{poly})$ and $(\mathsf{poly}, \mathsf{poly}, \mathsf{poly})$ security notions are in fact equivalent in the indistinguishability-based setting.

3 The Proposed PHPE Scheme

3.1 Construction

In this section, we will present our PHPE scheme for the function family $\mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$. This construction is presented in the attribute-only mode, i.e., without any actual payload. A key-encapsulation mechanism (KEM) version of this construction is presented in Section 4. In the proposed scheme, we assume that the function ρ outputted by $\mathsf{PGB}(f)$ for any $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is injective. This restriction can be readily overcome using standard techniques along the lines of [LOS+10, OT12b].

$\mathsf{PHPE.Setup}(1^{n'}, 1^{n})$: The setup algorithm takes as input the security parameter λ together with the lengths n' and n of the public and private attribute strings respectively. It proceeds as follows:

1. It first generates $(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (0, \overbrace{9, \ldots, 9}^{n'+n}))$.
2. For $\imath \in [n'+n]$, it sets $\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}, \widehat{\mathbb{B}}_\imath^* = \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,7)}, \boldsymbol{b}^{*(\imath,8)}\}$.
3. It outputs the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath\}_{\imath\in[n'+n]})$ and the master secret key $\text{MSK} = \{\widehat{\mathbb{B}}_\imath^*\}_{\imath\in[n'+n]}$.

$\mathsf{PHPE.Encrypt}(\text{MPK}, (\vec{x}, \vec{z}))$: The encryption algorithm takes as input the public parameters MPK and a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$. It executes the following:

1. First, it samples $\omega \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
2. Next, for $\iota' \in [n']$, it samples $\varphi'_{\iota'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}'^{(\iota')} = (\omega(1, x_{\iota'}), \vec{0}^4, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}}.$$
3. Then, for $\iota \in [n]$, it samples $\varphi_\iota \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}^{(\iota)} = (\omega(1, z_\iota), \vec{0}^4, \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}}.$$
4. It outputs the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$.

$\mathsf{PHPE.KeyGen}(\text{MPK}, \text{MSK}, f)$: The key generation algorithm takes as input the public parameters MPK, the master secret key MSK, along with a function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$. It operates as follows:

1. It first generates $\big((\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]}), \rho : [m] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f)$.
2. Next, it samples $\zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$.

3. Then, for $j' \in [m]$, it samples $\vec{\kappa}'^{(j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes

$$\boldsymbol{k}'^{(j')} = ((\gamma_{j'}, \alpha_{j'}), \vec{0}^4, \vec{\kappa}'^{(j')}, 0)_{\mathbb{B}^*_{\rho(j')}}.$$

4. Then, for $j \in [n]$, it samples $\vec{\kappa}^{(j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes

$$\boldsymbol{k}^{(j)} = ((\sigma_j, \zeta), \vec{0}^4, \vec{\kappa}^{(j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$

5. It outputs the decryption key $\text{SK}(f) = (f, \{\boldsymbol{k}'^{(j')}\}_{j' \in [m]}, \{\boldsymbol{k}^{(j)}\}_{j \in [n]})$.

$\mathsf{PHPE.Decrypt}(\text{MPK}, \text{SK}(f), \text{CT})$: The decryption algorithm takes in the public parameters MPK, a decryption key $\text{SK}(f) = (f, \{\boldsymbol{k}'^{(j')}\}_{j' \in [m]}, \{\boldsymbol{k}^{(j)}\}_{j \in [n]})$, and a ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$. It proceeds as follows:

1. It first computes $\Lambda'_{j'} = e(\boldsymbol{c}'^{(\rho(j'))}, \boldsymbol{k}'^{(j')})$ for $j' \in [m]$, and $\Lambda_j = e(\boldsymbol{c}^{(j)}, \boldsymbol{k}^{(j)})$ for $j \in [n]$.
2. Next, it determines the coefficients $(\{\Omega_j\}_{j \in [n]}, \{\Omega'_{j'}\}_{j' \in [m]}) = \mathsf{REC}(f, \vec{x})$.
3. Then, it computes $\Lambda = \left(\prod\limits_{j' \in [m]} \Lambda'^{\Omega'_{j'}}_{j'} \right) \left(\prod\limits_{j \in [n]} \Lambda_j^{\Omega_j} \right)$.
4. It outputs 1, if $\Lambda = 1_{\mathbb{G}_T}$, and 0, otherwise, where $1_{\mathbb{G}_T}$ is the identity element in $\mathbb{G}_T$.

► **Correctness**: For any decryption key $\text{SK}(f) = (f, \{\boldsymbol{k}'^{(j')}\}_{j' \in [m]}, \{\boldsymbol{k}^{(j)}\}_{j \in [n]})$ for a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, and any ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ encrypting a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, we have

$$\Lambda'_{j'} = g_T^{\omega(\alpha_{j'} x_{\rho(j')} + \gamma_{j'})} \text{ for } j' \in [m],$$
$$\Lambda_j = g_T^{\omega(\zeta z_j + \sigma_j)} \text{ for } j \in [n].$$

The above follows from the expressions of $\{\boldsymbol{k}'^{(j')}\}_{j' \in [m]}, \{\boldsymbol{k}^{(j)}\}_{j \in [n]}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$, and the dual orthonormality property of $\{\mathbb{B}_\iota, \mathbb{B}^*_\iota\}_{\iota \in [n'+n]}$. Hence, from Eq. (2.3) it follows that

$$\Lambda = g_T^{\omega\zeta f(\vec{x}, \vec{z})}.$$

Therefore, if $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 1$, i.e., $f(\vec{x}, \vec{z}) = 0$, then $\Lambda = 1_{\mathbb{G}_T}$, while if $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 0$, i.e., $f(\vec{x}, \vec{z}) \neq 0$, then $\Lambda \neq 1_{\mathbb{G}_T}$ with all but negligible probability $2/q$, i.e., except when $\omega = 0$ or $\zeta = 0$.

Remark 3.1 (On Multi-Ciphertext Scheme): The PHPE scheme described above is only secure against adversaries that are allowed to make a single ciphertext query. However, we can readily extend the above scheme to one that is secure for any a priori bounded number of ciphertext queries of the adversary. The extension is as follows: Suppose we want to design a scheme that is secure for q_{CT} number of ciphertext queries. Then, we would introduce a $4q_{\text{CT}}$-dimensional hidden subspace on each of the ciphertext and the decryption key sides, where each 4-dimensional hidden subspace on the ciphertext side and its corresponding 4-dimensional dual subspace on the decryption key side will be used to handle each ciphertext query in the security reduction. Clearly the size of ciphertexts, decryption keys, and public parameters would scale linearly with q_{CT}.

3.2 Security

We now present our main theorem:

Theorem 3.1: *The proposed* PHPE *scheme is* $(\mathsf{poly}, 1, \mathsf{poly})$*-simulation-based adaptively strongly partially hiding* (*as per the security model described in Section 2.6*) *under the* SXDLIN *assumption.*

The following corollary is immediate from the relation between indistinguishability-based and simulation-based security for FE, as mentioned in the Introduction as well as the equivalence of the single- and multi-ciphertext security in the indistinguishability-based setting for FE:

Corollary 3.1: *The proposed* PHPE *scheme is* $(\mathsf{poly}, \mathsf{poly}, \mathsf{poly})$*-indistinguishability-based adaptively strongly partially hiding* (*as per the security model described in Section 2.6*) *under the* SXDLIN *assumption.*

■ Proof Outline of Theorem 3.1

In order to prove Theorem 3.1, we apply an extended and more sophisticated version of the strong attribute hiding technique introduced by Okamoto and Takashima [OT12a, OT12b, OT13]. We consider a sequence of hybrid experiments which differ from one another in the construction of the ciphertext and/or the decryption keys queried by the augmented adversary $\mathcal{H}$ (described in Remark 2.1). The first hybrid corresponds to the experiment $\mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE},\mathsf{REAL}}(\lambda)$ (described in Section 2.6), while the last one corresponds to the experiment $\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE},\mathsf{IDEAL}}(\lambda)$ (also described in Section 2.6) with the simulator $\mathcal{S}$ described below. We argue that $\mathcal{H}$'s probability of outputting 1 changes only by a negligible amount in each successive hybrid experiment, thereby establishing Theorem 3.1. Note that we are considering only one ciphertext query made by the adversary $\mathcal{H}$. Let, $q_{\text{KEY-PRE}}, q_{\text{KEY-POST}}$ be respectively the number of decryption key queries made by $\mathcal{H}$ before and after making the ciphertext query, and $q_{\text{KEY}} = q_{\text{KEY-PRE}} + q_{\text{KEY-POST}}$. Note that we consider $q_{\text{KEY-PRE}}$ and $q_{\text{KEY-POST}}$ to be arbitrary polynomials in the security parameter λ.

Our simulator $\mathcal{S}$ adopts a different strategy for simulating the pre-ciphertext and post-ciphertext decryption keys. Accordingly, we design a different hybrid sequence for the pre-ciphertext and post-ciphertext decryption keys. Our approach for handling pre-ciphertext and post-ciphertext decryption keys is somewhat similar in spirit to that used in [LW12, Att14, OT13]. As a result of this approach, we are also able to achieve a security loss that is proportional to only the number of pre-ciphertext decryption key queries. In our hybrid transitions, we consider 5 different forms for the ciphertext queried by $\mathcal{H}$, namely, Normal, Temporal 0, Temporal 1, Temporal 2, and Simulated, while 4 different forms for the decryption keys queried by $\mathcal{H}$, namely, Normal, Temporal 1, Temporal 2, and Simulated. However, the Temporal 2 and Simulated forms of the pre-ciphertext and post-ciphertext decryption keys are different. In $\mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE},\mathsf{REAL}}(\lambda)$, the Normal forms of ciphertext and decryption keys are used, whereas other forms of ciphertext and decryption keys are used in various intermediate hybrid experiments as well as in the final hybrid experiment, which corresponds to $\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE},\mathsf{IDEAL}}(\lambda)$ and uses the Simulated forms for both the ciphertext and the decryption keys.

The overall structure of our security reduction is demonstrated in Fig. 3.1. We employ the sequence of hybrid experiments Hyb_0 through Hyb_8, which are formally described below after the description of our simulator. We start with Hyb_0, which is the real experiment. Hence, in this hybrid, the ciphertext queried by $\mathcal{H}$ is in its Normal form (Eq. (3.1)) and all the decryption keys queried by $\mathcal{H}$ are also in their Normal forms (Eq. (3.2)). First, in Hyb_1, we change the ciphertext queried by $\mathcal{H}$ to the Temporal 0 form (Eq. (3.3)). From $\mathsf{Hyb}_1 \equiv \mathsf{Hyb}_{2\text{-}0\text{-}4}$, we perform a sequence of $4q_{\text{KEY-PRE}}$ hybrid transitions, namely, $\{\mathsf{Hyb}_{2\text{-}\chi\text{-}1}, \ldots, \mathsf{Hyb}_{2\text{-}\chi\text{-}4}\}_{\chi \in [q_{\text{KEY-PRE}}]}$, where for each $\chi \in [q_{\text{KEY-PRE}}]$, the sequence of 4 hybrid transitions, namely, $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}, \ldots, \mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ are devoted for changing the χ^{th} pre-ciphertext decryption key queried by $\mathcal{H}$ to its Simulated form. In $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$, the ciphertext is changed to its Temporal 1 form (Eq. (3.4)), while for $h \in [q_{\text{KEY}}]$, the h^{th} decryption key is in its Simulated form (Eq. (3.7)), if $h < \chi$, and in its Normal form (Eq. (3.2)), if $h \geq \chi$. Next, in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$, the χ^{th} decryption key is altered to its Temporal 1 form (Eq. (3.5)). After that, in the ciphertext is modified to its Temporal 2 form (Eq. (3.6)) in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$. Then, in $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$, we change the χ^{th} decryption key to its Simulated form (Eq. (3.7)). This process is repeated for all $\chi \in [q_{\text{KEY-PRE}}]$. In the last hybrid $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$ of this sequence, we get the ciphertext in its Temporal 2 form (Eq. (3.6)), all the pre-ciphertext decryption keys in their Simulated forms (Eq. (3.7)), while all the post-ciphertext decryption keys in their Normal forms (Eq. (3.2)). Upto this point, we follow a similar top level hybrid transition strategy as that employed in [OT12a, OT12b, OT13], although our analysis of the hybrid transitions is much more sophisticated compared to those works.

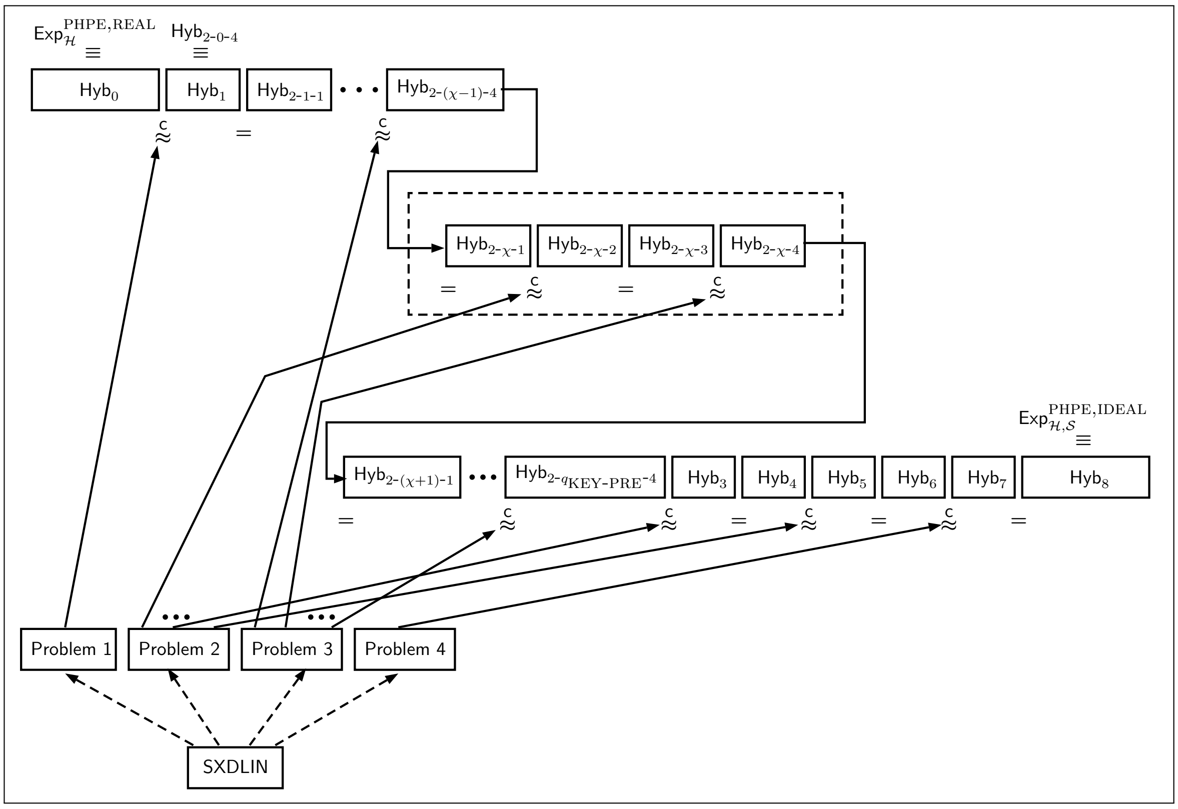
$\mathsf{Exp}_{\mathcal{H}}^{\mathrm{PHPE,REAL}}$
$\mathsf{Hyb}_{2\text{-}0\text{-}4}$
Hyb_0
Hyb_1
$\mathsf{Hyb}_{2\text{-}1\text{-}1}$
$\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$
$\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$
$\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$
$\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$
$\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$
$\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathrm{PHPE,IDEAL}}$
$\mathsf{Hyb}_{2\text{-}(\chi+1)\text{-}1}$
$\mathsf{Hyb}_{2\text{-}q_{\mathrm{KEY\text{-}PRE}}\text{-}4}$
Hyb_3
Hyb_4
Hyb_5
Hyb_6
Hyb_7
Hyb_8
Problem 1
Problem 2
Problem 3
Problem 4
SXDLIN

Fig. 3.1: Structure of the Hybrid Reduction for the Proof of Theorem 3.1

After this point, we perform a transition to Hyb_3, where we alter the form of the ciphertext again to its Temporal 1 form (Eq. (3.4)), the pre-ciphertext decryption keys to their Temporal 2 forms (Eq. (3.8), while the post-ciphertext decryption keys to their Temporal 1 forms (Eq. (3.8)). Next, in Hyb_4, the ciphertext is modified to its Temporal 3 form (Eq. (3.9)). After that the pre-ciphertext decryption keys are brought back to their Simulated forms (Eq. (3.7)) by executing a transition to Hyb_5. At this point the alteration of pre-ciphertext decryption keys becomes complete.

After that, we change the ciphertext to its Simulated form (Eq. (3.10)), while the post-ciphertext decryption keys to their Temporal 2 forms (Eq. (3.11)) in Hyb_6. At this point the modification of the ciphertext is finished. Finally, using the transition to Hyb_7 followed by the transition to Hyb_8, we alter the post-ciphertext decryption keys to their Simulated forms (Eq. (3.13)). Now, we arrive at the ideal experiment, and our hybrid transition gets complete.

For the rest of this section, by saying 'some quantity A is bounded by another quantity B', we mean that $A \leq B + \mathsf{negl}(\lambda)$, where negl is some negligible function. We prove that the advantage gap of $\mathcal{H}$ between Hyb_0 and Hyb_1 is bounded by the advantage of any algorithm in solving Problem 1 (Definition A.1). For this, in the usual way, we construct an algorithm that given a Problem 1 instance simulates the view of $\mathcal{H}$ in Hyb_0 or that in Hyb_1 depending on whether the challenge bit $\widehat{\beta} \in \{0,1\}$ of the given Problem 1 instance is $\widehat{\beta} = 0$ or 1 (Lemma B.1). The advantage in solving Problem 1 is already known to be bounded by that in solving the SXDLIN problem (Lemma A.1).

We show that $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ can be information-theoretically changed to $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ for all $\chi \in [q_{\mathrm{KEY\text{-}PRE}}]$ (Lemma B.2) with the help of a Type I information-theoretic trick using the fact that for all $\imath \in [n'+n]$, the portion $\{\boldsymbol{b}^{(\imath,3)}, \ldots, \boldsymbol{b}^{(\imath,6)}\}$ of the basis $\mathbb{B}_\imath$ and the corresponding segment $\{\boldsymbol{b}^{*(\imath,3)}, \ldots, \boldsymbol{b}^{*(\imath,6)}\}$ of the basis $\mathbb{B}^*_\imath$ are completely unknown to $\mathcal{H}$. In particular, when $\chi = 1$, it means that Hyb_1 can be conceptually changed to $\mathsf{Hyb}_{2\text{-}1\text{-}1}$.

The advantage gap of $\mathcal{H}$ between $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ is shown to be bounded by the advantage of any algorithm in solving Problem 2 (Definition A.2) for all $\chi \in [q_{\text{KEY-PRE}}]$ (Lemma B.3), i.e., the advantage of any algorithm in solving the SXDLIN problem (Lemma A.2). Here, we crucially rely on the linearity property of PGB, as described in Section 2.3.

We demonstrate that $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ can be information-theoretically changed to $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ for all $\chi \in [q_{\text{KEY-PRE}}]$ (Lemma B.4), with the help of a Type II conceptual trick that uses the fact that for all $\imath \in [n'+n]$, the part $\{\boldsymbol{b}^{(\imath,3)}, \boldsymbol{b}^{(\imath,4)}\}$ of the basis $\mathbb{B}_\imath$ and the corresponding portion $\{\boldsymbol{b}^{*(\imath,3)}, \boldsymbol{b}^{*(\imath,4)}\}$ of the basis $\mathbb{B}^*_\imath$ is unknown to $\mathcal{H}$. Here, we crucially employ the statistical indistinguishability lemma due to Okamoto and Takashima [OT10] (Lemma B.5). This information-theoretic step is central towards changing the pre-ciphertext decryption keys to their Simulated forms. We achieve this step by a sophisticated analysis of the various nice linear algebraic properties of the matrix representation $\boldsymbol{L}^{(\chi)}$ (obtained by applying the algorithm of Lemma 2.1) of the ABP computing the function $f_\chi \in \mathcal{F}^{(q,n',n)}_{\text{ABPOIP}}$ associated with the χth decryption key queried by $\mathcal{H}$. We perform a case analysis depending on whether f_χ satisfies the R^{ABPOIP} relation with the public-private attribute pair $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$ associated with the ciphertext query of $\mathcal{H}$. Our analysis of this step may be useful in other similar applications of the matrix representation of specific forms of ABP's.

We argue that the advantage difference of $\mathcal{H}$ between $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ is bounded by the advantage of any algorithm in solving Problem 3 (Definition A.3), i.e., that of any algorithm in solving the SXDLIN problem (Lemma A.3), for all $\chi \in [q_{\text{KEY-PRE}}]$ (Lemma B.6). Here again we make use of the linearity property of PGB.

We achieve the transition from $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$ to Hyb_3 by applying a Type I information-theoretic trick similar to that used in the transition from $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$, followed by a computational change with the help of Problem 2 similar to that used for the transition from $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ (Lemma B.7).

We demonstrate that Hyb_3 can be information-theoretically changed to Hyb_4 (Lemma B.8) with the help of a Type III information-theoretic change that uses the fact that for all $\imath \in [n'+n]$, the segment $\{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}\}$ of the basis $\mathbb{B}^*_\imath$ is partially hidden from $\mathcal{H}$ (note that its dual basis elements are parts of the public parameters), as well as the part $\{\boldsymbol{b}^{(\imath,3)}, \boldsymbol{b}^{(\imath,4)}\}$ of the basis $\mathbb{B}_\imath$ and the corresponding portion $\{\boldsymbol{b}^{*(\imath,3)}, \boldsymbol{b}^{*(\imath,4)}\}$ of the basis $\mathbb{B}^*_\imath$ are completely hidden from $\mathcal{H}$.

The transition from Hyb_4 to Hyb_5 is shown to be achievable by a more sophisticated computational trick using Problem 2 (Lemma B.9), and thus the advantage gap of $\mathcal{H}$ between these two hybrids is bounded by the advantage of any algorithm in solving the SXDLIN problem. Here, we crucially make use of the pre-image samplability property [O'N10], which is satisfied by the predicate family R^{ABPOIP}, as argued below in the description of our simulator. We rely on the linearity property of PGB as well to achieve this transition.

The transformation from Hyb_5 to Hyb_6 is again accomplished applying a Type II information-theoretic trick (Lemma B.10), where we again use the fact that for all $\imath \in [n'+n]$, the part $\{\boldsymbol{b}^{(\imath,3)}, \boldsymbol{b}^{(\imath,4)}\}$ of the basis $\mathbb{B}_\imath$ and the corresponding segment $\{\boldsymbol{b}^{*(\imath,3)}, \boldsymbol{b}^{*(\imath,4)}\}$ of the basis $\mathbb{B}^*_\imath$ are completely unknown to $\mathcal{H}$. The linear transformations used for realizing the Type II information-theoretic trick are carefully selected keeping in view the specific forms of ciphertext and post-ciphertext decryption keys in Hyb_6, rather than being random ones as usually used in application of the Type II trick.

The difference in the advantage of $\mathcal{H}$ between Hyb_6 and Hyb_7 is proven (Lemma B.11) to be bounded by the advantage of any algorithm in solving Problem 4 (Definition A.4), i.e., that of any algorithm in solving the SXDLIN problem (Lemma A.4).

Finally, we prove that the view of $\mathcal{H}$ in Hyb_7 and that in Hyb_8 are identically distributed (Lemma B.12) using the privacy property of the algorithm PGB, as described in Section 2.3.

▪ Description of the Simulator

The simulator $\mathcal{S}$ is described below.

- In order to generate the public parameters, $\mathcal{S}$ proceeds as follows:
  1. It first generates $(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathrm{OB}}(n'+n, (0, \overbrace{9, \ldots, 9}^{n'+n}))$.
  2. For $\imath \in [n'+n]$, it sets $\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}$.
  3. It outputs the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath\}_{\imath\in[n'+n]})$.
- For $h \in [q_{\text{KEY-PRE}}]$, $\mathcal{S}$ simulates the h^{th} decryption key queried by $\mathcal{H}$ corresponding to some function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ as follows:
  1. At first, it generates $\big((\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big), \big((\{\widehat{\sigma}_{h,j}\}_{j\in[n]}, \{\widehat{\alpha}_{h,j'}, \widehat{\gamma}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$.
  2. Next, it samples $\zeta_h, \widehat{\zeta}_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
  3. Then, for $j' \in [m_h]$, it samples $\vec{\kappa}'^{(h,j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes
  $$\boldsymbol{k}'^{(h,j')} = ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}.$$
  4. Then, for $j \in [n]$, it samples $\vec{\kappa}^{(h,j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes
  $$\boldsymbol{k}^{(h,j)} = ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$
  5. It outputs $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$.
- When $\mathcal{H}$ queries a ciphertext for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{S}$ receives $\vec{x}$ and $\{R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z}))\}_{h\in[q_{\text{KEY-PRE}}]}$. It simulates the ciphertext as follows:
  1. At first, it samples $\vec{s} \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n \mid R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{s})) = R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z}))\forall h \in [q_{\text{KEY-PRE}}]\}$. Note that the set S is exactly identical to the set $\widetilde{S} = \{\vec{s} \in \mathbb{F}_q^n \mid R^{\text{IP}}((f_{h,1}(\vec{x}), \ldots, f_{h,n}(\vec{x})), \vec{s}) = R^{\text{IP}}((f_{h,1}(\vec{x}), \ldots, f_{h,n}(\vec{x})), \vec{z})\forall h \in [q_{\text{KEY-PRE}}]\}$, where R^{IP} represents the inner-product predicate family defined as $R^{\text{IP}} = \{R^{\text{IP}}(\vec{w}, \cdot) : \mathbb{F}_q^n \to \{0,1\} \mid \vec{w} \in \mathbb{F}_q^n\}$ such that $R^{\text{IP}}(\vec{w}, \vec{v}) = 1$, if $\vec{v} \cdot \vec{w} = 0$, and 0, if $\vec{v} \cdot \vec{w} \neq 0$ for $\vec{v}, \vec{w} \in \mathbb{F}_q^n$, and $f_{h,j}$ is the j^{th} component ABP of f_h for $h \in [q_{\text{KEY-PRE}}], j \in [n]$. It has already been demonstrated by O'Neill [O'N10] that the inner-product predicate family R^{IP} is *pre-image-samplable*, which essentially means that we can efficiently sample from the set $\widetilde{S}$. In fact, he provided an explicit algorithm for doing this. Thus, given $\{f_h\}_{h\in[q_{\text{KEY-PRE}}]}$ and $\vec{x}$, $\mathcal{S}$ can efficiently sample from S by first determining the vectors $\{(f_{h,1}(\vec{x}), \ldots, f_{h,n}(\vec{x}))\}_{h\in[q_{\text{KEY-PRE}}]}$ and then sampling from the set $\widetilde{S}$ using the algorithm described in [O'N10].
  2. Then, it samples $\tau, \theta, \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
  3. Next, for $\iota' \in [n']$, it samples $\varphi'_{\iota'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
  $$\boldsymbol{c}'^{(\iota')} = (\vec{0}^3, \tau, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}}.$$
  4. Then, for $\iota \in [n]$, it samples $\varphi_\iota \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
  $$\boldsymbol{c}^{(\iota)} = (\vec{0}^3, \tau, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}}.$$
  5. It outputs the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$.
- For $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, in response to the h^{th} decryption key query of $\mathcal{H}$ corresponding to some function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, $\mathcal{S}$ executes the following steps:

1. It first generates $\big((\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$.
2. Next, it samples $\zeta_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
3. Then, it queries its oracle $\mathcal{O}_{R^{\text{ABP}\circ\text{IP}}}((\vec{x}, \vec{z}), \cdot)$ with the function f_h, and receives back $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z}))$. If $R^{\text{ABP}\circ\text{IP}}(f_h(\vec{x}, \vec{z})) = 1$, i.e., $f_h(\vec{x}, \vec{z}) = 0$, it forms $\big((\{\nu_{h,j}\}_{j\in[n]}, \{\mu_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, 0)$. Otherwise, if $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$, i.e., $f_h(\vec{x}, \vec{z}) \neq 0$, then it samples $\check{\zeta}_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and generates $\big((\{\nu_{h,j}\}_{j\in[n]}, \{\mu_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, \check{\zeta}_h)$.
4. Next, for $j' \in [m_h]$, it samples $\eta'_{h,j'} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \vec{\kappa}'^{(h,j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes
$$\boldsymbol{k}'^{(h,j')} = ((\gamma_{h,j'}, \alpha_{h,j'}), (\eta'_{h,j'}, \mu_{h,j'}), \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}.$$
5. Then, for $j \in [n]$, it samples $\eta_{h,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \vec{\kappa}^{(h,j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes
$$\boldsymbol{k}^{(h,j)} = ((\sigma_{h,j}, \zeta_h), (\eta_{h,j}, \nu_{h,j}), \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$
6. It outputs $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$.

▪ Sequence of Hybrid Experiments

The hybrid experiments are described below. In the description of these hybrids, a part framed by a box indicates coefficients that are altered in a transition from its previous hybrid.

$\mathsf{Hyb_0}$: This experiment corresponds to the experiment $\mathsf{Exp}_{\mathcal{H}}^{\mathsf{PHPE,REAL}}(\lambda)$ defined in Section 2.6. Thus, in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that
$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \vec{0}^2, \vec{0}^2, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \vec{0}^2, \vec{0}^2, \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{3.1}$$
where $\omega, \{\varphi'_{\iota'}\}_{\iota'\in[n']}, \{\varphi_\iota\}_{\iota\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, while for $h \in [q_{\text{KEY}}]$, the hth decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is formed as $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that
$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.2}$$
where $\zeta_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\{\vec{\kappa}'^{(h,j')}\}_{j'\in[m_h]}, \{\vec{\kappa}^{(h,j)}\}_{j\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, m_h+n+1 is the number of vertices in the ABP Γ'_h computing the function f_h as described in Section 2.3, and $\big((\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$. Here, $\{\mathbb{B}_\iota, \mathbb{B}^*_\iota\}_{\iota\in[n'+n]}$ is the collection of dual orthonormal bases generated by executing $\mathcal{G}_{\text{OB}}(n' + n, (0, \overbrace{9, \ldots, 9}^{n'+n}))$ during setup.

$\mathsf{Hyb_1}$: This experiment is analogous to Hyb_0 except that in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that
$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), (\boxed{\vartheta}, 0), \vec{0}^2, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), (\boxed{\vartheta}, 0), \vec{0}^2, \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{3.3}$$
where $\vartheta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and all the other variables are generated as in Hyb_0.

$\mathsf{Hyb}_{\mathbf{2}\text{-}\boldsymbol{\chi}\text{-}\mathbf{1}}$ **(**$\boldsymbol{\chi} \in [\boldsymbol{q}_{\text{KEY-PRE}}]$**)**: The experiment $\mathsf{Hyb}_{2\text{-}0\text{-}4}$ coincides with Hyb_1. This experiment is analogous to $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ except that in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \boxed{\tau(1, x_{\iota'}), \theta(1, x_{\iota'})}, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \boxed{\tau(1, z_\iota), \theta(1, s_\iota)}, \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{3.4}$$

where $\tau, \theta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{s} \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n \mid R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{s})) = R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) \forall h \in [q_{\text{KEY-PRE}}]\}$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$.

$\mathsf{Hyb}_{\mathbf{2}\text{-}\boldsymbol{\chi}\text{-}\mathbf{2}}$ **(**$\boldsymbol{\chi} \in [\boldsymbol{q}_{\text{KEY-PRE}}]$**)**: This experiment is the same as $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ with the only exception that the χ^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_\chi \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is formed as $\text{SK}(f_\chi) = (f_\chi, \{\boldsymbol{k}'^{(\chi,j')}\}_{j' \in [m_\chi]}, \{\boldsymbol{k}^{(\chi,j)}\}_{j \in [n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \boxed{(\widetilde{\gamma}_{\chi,j'}, \widetilde{\alpha}_{\chi,j'})}, \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \boxed{(\widetilde{\sigma}_{\chi,j}, \widetilde{\zeta}_\chi)}, \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.5}$$

where $\widetilde{\zeta}_\chi \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\big((\{\widetilde{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]}), \rho_\chi : [m_\chi] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_\chi)$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$.

$\mathsf{Hyb}_{\mathbf{2}\text{-}\boldsymbol{\chi}\text{-}\mathbf{3}}$ **(**$\boldsymbol{\chi} \in [\boldsymbol{q}_{\text{KEY-PRE}}]$**)**: This experiment is analogous to $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ except that in this experiment, the ciphertext queried by $\mathcal{H}$ for the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is formed as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ such that $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}$ are given by Eq. (3.4) and

$$\boldsymbol{c}^{(\iota)} = (\omega(1, z_\iota), \tau(1, \boxed{s_\iota}), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \tag{3.6}$$

where all the variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$.

$\mathsf{Hyb}_{\mathbf{2}\text{-}\boldsymbol{\chi}\text{-}\mathbf{4}}$ **(**$\boldsymbol{\chi} \in [\boldsymbol{q}_{\text{KEY-PRE}}]$**)**: This experiment is identical to $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ except that the χ^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_\chi \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is generated as $\text{SK}(f_\chi) = (f_\chi, \{\boldsymbol{k}'^{(\chi,j')}\}_{j' \in [m_\chi]}, \{\boldsymbol{k}^{(\chi,j)}\}_{j \in [n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \boxed{\vec{0}^2, (\widehat{\gamma}_{\chi,j'}, \widehat{\alpha}_{\chi,j'})}, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \boxed{\vec{0}^2, (\widehat{\sigma}_{\chi,j}, \widehat{\zeta}_\chi)}, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.7}$$

where $\widehat{\zeta}_\chi \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\big((\{\widehat{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widehat{\alpha}_{\chi,j'}, \widehat{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]}), \rho_\chi : [m_\chi] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_\chi)$, and all the other variables are generated in the same manner as that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$.

$\mathsf{Hyb}_{\mathbf{3}}$: This experiment is analogous to $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$ except that in this experiment, the ciphertext queried by $\mathcal{H}$ for the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is formed as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$, where $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$ are given by Eq. (3.4), while for $h \in [q_{\text{KEY}}]$, the h^{th} decryption key queried by $\mathcal{H}$ for $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is generated as $\text{SK}(f_h) = (f_h,$

$\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that

$$\boldsymbol{k}'^{(h,j')} = \begin{cases} ((\gamma_{h,j'}, \alpha_{h,j'}), \boxed{(\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})}, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ \qquad \text{for } j' \in [m_h], \text{if } h \in [q_{\text{KEY-PRE}}], \\ ((\gamma_{h,j'}, \alpha_{h,j'}), \boxed{(\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})}, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ \qquad \text{for } j' \in [m_h], \text{if } h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], \end{cases}$$
$$\boldsymbol{k}^{(h,j)} = \begin{cases} ((\sigma_{h,j}, \zeta_h), \boxed{(\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)}, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ \qquad \text{for } j \in [n], \text{if } h \in [q_{\text{KEY-PRE}}], \\ ((\sigma_{h,j}, \zeta_h), \boxed{(\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)}, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ \qquad \text{for } j \in [n], \text{if } h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], \end{cases} \tag{3.8}$$

where $\{\widetilde{\zeta}_h\}_{h\in[q_{\text{KEY}}]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\big((\{\widetilde{\sigma}_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{h,j'}, \widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$ for $h \in [q_{\text{KEY}}]$, and all the other variables are formed as in $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$.

Hyb_4: This experiment is analogous to Hyb_3 except that in this experiment, the ciphertext queried by $\mathcal{H}$ for the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\boxed{\vec{0}^2}, \tau(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\boxed{\vec{0}^2}, \tau(1, z_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{3.9}$$

where all the variables are generated as in Hyb_3.

Hyb_5: This experiment is identical to Hyb_4 except that in this experiment, for $h \in [q_{\text{KEY-PRE}}]$, the h^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is generated as $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ are given by Eq. (3.7).

Hyb_6: This experiment is the same as Hyb_5 except that in this experiment, the ciphertext queried by $\mathcal{H}$ for the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\vec{0}^2, \boxed{(0, \tau)}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\vec{0}^2, \boxed{(0, \tau)}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{3.10}$$

while for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, the h^{th} decryption key queried by $\mathcal{H}$ for $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is generated as $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \boxed{(\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})\boldsymbol{U}'^{(\rho_h(j'))}}, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \boxed{(\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)\boldsymbol{U}^{(j)}}, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.11}$$

where $\boldsymbol{Z}'^{(\iota')} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) \mid (1, x_{\iota'})\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$, $\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^\intercal$ for $\iota' \in [n']$, $\boldsymbol{Z}^{(\iota)} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q \mid (1, z_\iota)\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^\intercal$ for $\iota \in [n]$, and all the other variables are generated as in Hyb_5.

$\mathsf{Hyb_7}$: This experiment is identical to Hyb_6 with the only exception that for $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the h^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is generated as $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \boxed{(\eta'_{h,j'}, \widetilde{\alpha}_{h,j'} x_{\rho_h(j')} + \widetilde{\gamma}_{h,j'})}, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \boxed{(\eta_{h,j}, \widetilde{\zeta}_h z_j + \widetilde{\sigma}_{h,j})}, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.12}$$

where $\{\eta'_{h,j'}\}_{h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], j' \in [m_h]}, \{\eta_{h,j}\}_{h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], j \in [n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and all the other variables are generated as in Hyb_6.

$\mathsf{Hyb_8}$: This experiment is analogous to Hyb_7 with the only exception that for $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the h^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is formed as $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\eta'_{h,j'}, \boxed{\mu_{h,j'}}), \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), (\eta_{h,j}, \boxed{\nu_{h,j}}), \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{3.13}$$

where $\{\widetilde{\zeta}_h\}_{h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $((\{\nu_{h,j}\}_{j \in [n]}, \{\mu_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, f_h(\vec{x}, \widetilde{\zeta}_h \vec{z}))$ for $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, and all the other variables are generated as in Hyb_7. Observe that for any $h \in [q_{\text{KEY-PRE}}] + 1, q_{\text{KEY}}]$, if $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 1$, i.e., $f_h(\vec{x}, \vec{z}) = 0$, then $f_h(\vec{x}, \widetilde{\zeta}_h \vec{z}) = \widetilde{\zeta}_h f_h(\vec{x}, \vec{z}) = 0$, while if $R^{\text{ABP}\circ\text{IP}}(f_h(\vec{x}, \vec{z})) = 0$, i.e., $f_h(\vec{x}, \vec{z}) \neq 0$, then due to the uniform and independent (of the other variables) choice of $\widetilde{\zeta}_h$, it follows that $f_h(\vec{x}, \widetilde{\zeta}_h \vec{z}) = \widetilde{\zeta}_h f_h(\vec{x}, \vec{z})$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$. Thus, this experiment coincides with the experiment $\mathsf{Exp}^{\mathsf{PHPE,IDEAL}}_{\mathcal{H},\mathcal{S}}(\lambda)$ with the simulator $\mathcal{S}$ as described above.

▪ **Analysis**

Let us now denote by $\mathsf{Adv}^{(\jmath)}_{\mathcal{H}}(\lambda)$ the probability that $\mathcal{H}$ outputs 1 in $\mathsf{Hyb}_\jmath$ for $\jmath \in \{0, 1, \{2\text{-}\chi\text{-}k\}_{\chi \in [q_{\text{KEY-PRE}}], k \in [4]}, 3, \ldots, 8\}$. By definition of the hybrids, we clearly have $\mathsf{Adv}^{\mathsf{PHPE,SIM\text{-}AH}}_{\mathcal{H}}(\lambda) = |\mathsf{Adv}^{(0)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(8)}_{\mathcal{H}}(\lambda)|$. Hence, we have

$$\begin{aligned} \mathsf{Adv}^{\mathsf{PHPE,SIM\text{-}AH}}_{\mathcal{H}}(\lambda) \leq{} & |\mathsf{Adv}^{(0)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{H}}(\lambda)| + \\ & \sum_{\chi \in [q_{\text{KEY-PRE}}]} \Big[|\mathsf{Adv}^{(2\text{-}(\chi-1)\text{-}4)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}1)}_{\mathcal{H}}(\lambda)| + \\ & \sum_{k \in [3]} |\mathsf{Adv}^{(2\text{-}\chi\text{-}k)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}(k+1))}_{\mathcal{H}}(\lambda)| \Big] + \\ & |\mathsf{Adv}^{(2\text{-}q_{\text{KEY-PRE}}\text{-}4)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{H}}(\lambda)| + \sum_{\jmath \in [3,7]} |\mathsf{Adv}^{(\jmath)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(\jmath+1)}_{\mathcal{H}}(\lambda)|. \end{aligned}$$

Then, Theorem 3.1 follows from Lemmas B.1–B.12 presented in Appendix B, in conjunction with Lemmas A.1–A.4 presented in Appendix A.

4 KEM Version of the Proposed PHPE Scheme

In this section, we will present the key-encapsulation mechanism (KEM) version of our PHPE scheme for the function family $\mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$. This scheme is an augmentation of our attribute-only scheme presented in Section 3. Similar to our attribute-only construction, we assume that the function ρ outputted by $\mathsf{PGB}(f)$ for any $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$ is injective.

$\mathsf{PHPE.Setup}(1^{n'}, 1^n)$: The setup algorithm takes as input the security parameter λ together with the lengths n' and n of the public and private attribute strings respectively. It proceeds as follows:

1. It first generates $(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n+1, (6, \overbrace{9,\ldots,9}^{n'+n}))$.
2. It sets
$$\widehat{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,4)}, \boldsymbol{b}^{(0,6)}\}, \widehat{\mathbb{B}}^*_0 = \{\boldsymbol{b}^{*(0,1)}, \boldsymbol{b}^{*(0,4)}, \boldsymbol{b}^{*(0,5)}\}.$$
3. For $\imath \in [n'+n]$, it sets $\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}, \widehat{\mathbb{B}}^*_\imath = \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,7)}, \boldsymbol{b}^{*(\imath,8)}\}$.
4. It outputs the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath\}_{\imath\in[0,n'+n]})$ and the master secret key $\text{MSK} = \{\widehat{\mathbb{B}}^*_\imath\}_{\imath\in[0,n'+n]}$.

$\mathsf{PHPE.Encrypt}(\text{MPK}, (\vec{x}, \vec{z}))$: The encryption algorithm takes as input the public parameters MPK and a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$. It executes the following:

1. First, it samples $\omega, \xi, \varphi'_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}'^{(0)} = (\omega, \vec{0}^2, \xi, 0, \varphi'_0)_{\mathbb{B}_0}, \text{KEM} = g_T^\xi.$$
2. Next, for $\iota' \in [n']$, it samples $\varphi'_{\iota'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}'^{(\iota')} = (\omega(1, x_{\iota'}), \vec{0}^4, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}}.$$
3. Then, for $\iota \in [n]$, it samples $\varphi_\iota \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}^{(\iota)} = (\omega(1, z_\iota), \vec{0}^4, \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}}.$$
4. It outputs the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$, and the session key KEM.

$\mathsf{PHPE.KeyGen}(\text{MPK}, \text{MSK}, f)$: The key generation algorithm takes as input the public parameters MPK, the master secret key MSK, along with a function $f \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$. It operates as follows:

1. It first generates $\big((\{\sigma_j\}_{j\in[n]}, \{\alpha_{j'}, \gamma_{j'}\}_{j'\in[m]}), \rho : [m] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f)$.
2. Then, it samples $r_0, \kappa'_0, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{k}'^{(0)} = (-r_0, \vec{0}^2, 1, \kappa'_0, 0)_{\mathbb{B}^*_0}.$$
3. Then, for $j' \in [m]$, it samples $b'_{j'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\kappa}'^{(j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\gamma^+_{j'} = \gamma_{j'} + b'_{j'} r_0$ for $j' \in [m]$, followed by
$$\boldsymbol{k}'^{(j')} = ((\gamma^+_{j'}, \alpha_{j'}), \vec{0}^4, \vec{\kappa}'^{(j')}, 0)_{\mathbb{B}^*_{\rho(j')}}.$$
4. Then, for $j \in [n]$, it samples $b_j \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\kappa}^{(j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\sigma^+_j = \sigma_j + b_j r_0$, followed by
$$\boldsymbol{k}^{(j)} = ((\sigma^+_j, \zeta), \vec{0}^4, \vec{\kappa}^{(j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$
5. It outputs decryption key $\text{SK}(f) = (f, \{b'_{j'}\}_{j'\in[m]}, \{b_j\}_{j\in[n]}, \{\boldsymbol{k}'^{(j')}\}_{j'\in[0,m]}, \{\boldsymbol{k}^{(j)}\}_{j\in[n]})$.

$\mathsf{PHPE.Decrypt}(\text{MPK}, \text{SK}(f), \text{CT})$: The decryption algorithm takes as input the public parameters MPK, a decryption key $\text{SK}(f) = (f, \{b'_{j'}\}_{j'\in[m]}, \{b_j\}_{j\in[n]}, \{\boldsymbol{k}'^{(j')}\}_{j'\in[0,m]}, \{\boldsymbol{k}^{(j)}\}_{j\in[n]})$, and a ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$. It proceeds as follows:

1. It first computes $\Lambda'_0 = e(\boldsymbol{c}'^{(0)}, \boldsymbol{k}'^{(0)})$, $\Lambda'_{j'} = e(\boldsymbol{c}'^{(\rho(j'))}, \boldsymbol{k}'^{(j')})$ for $j' \in [m]$, and $\Lambda_j = e(\boldsymbol{c}^{(j)}, \boldsymbol{k}^{(j)})$ for $j \in [n]$.
2. Next, it determines the coefficients $(\{\Omega_j\}_{j\in[n]}, \{\Omega'_{j'}\}_{j'\in[m]}) = \mathsf{REC}(f, \vec{x})$.
3. Then, it computes
$$\Lambda = \left[\left(\prod_{j'\in[m]} \Lambda'^{\Omega'_{j'}}_{j'}\right)\left(\prod_{j\in[n]} \Lambda_j^{\Omega_j}\right)\right]^{\frac{1}{\sum_{j'\in[m]} \Omega'_{j'} b'_{j'} + \sum_{j\in[n]} \Omega_j b_j}}.$$
4. It outputs $\widetilde{\text{KEM}} = \Lambda \Lambda'_0$.

► **Correctness**: Note that for any decryption key $\text{SK}(f) = (f, \{b'_{j'}\}_{j'\in[m]}, \{b_j\}_{j\in[n]}, \{\boldsymbol{k}'^{(j')}\}_{j'\in[0,m]}, \{\boldsymbol{k}^{(j)}\}_{j\in[n]})$ for a function $f \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, and any ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ associated with a pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, we have

$$\begin{aligned}
\Lambda'_0 &= g_T^{-\omega r_0+\xi},\\
\Lambda'_{j'} &= g_T^{\omega(\alpha_{j'}x_{\rho(j')}+\gamma_{j'}+b'_{j'}r_0)} \text{ for } j' \in [m],\\
\Lambda_j &= g_T^{\omega(\zeta z_j+\sigma_j+b_j r_0)} \text{ for } j \in [n].
\end{aligned}$$

This follows from the expressions of $\{\boldsymbol{k}'^{(j')}\}_{j'\in[0,m]}, \{\boldsymbol{k}^{(j)}\}_{j\in[n]}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}$, $\{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]}$, and the dual orthonormality property of $\{\mathbb{B}_\iota, \mathbb{B}^*_\iota\}_{\iota\in[0,n'+n]}$. Hence, from Eq. (2.3) it follows that

$$\Lambda = g_T^{\left[\omega\zeta f(\vec{x},\vec{z})+\omega r_0[\sum_{j'\in[m]}\Omega'_{j'}b'_{j'}+\sum_{j\in[n]}\Omega_j b_j]\right]/\left[\sum_{j'\in[m]}\Omega'_{j'}b'_{j'}+\sum_{j\in[n]}\Omega_j b_j\right]}.$$

Thus, if $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 1$, i.e., $f(\vec{x}, \vec{z}) = 0$, then $\Lambda = g_T^{\omega r_0}$, and hence, $\widetilde{\text{KEM}} = g_T^{\xi} = \text{KEM}$ with all but negligible probability $1/q$, i.e., except when $\sum\limits_{j'\in[m]} \Omega'_{j'}b'_{j'} + \sum\limits_{j\in[n]} \Omega_j b_j = 0$. On the other hand, if $R^{\text{ABP}\circ\text{IP}}(f, (\vec{x}, \vec{z})) = 0$, i.e., $f(\vec{x}, \vec{z}) \neq 0$, then $\Lambda \neq g_T^{\omega r_0}$, and hence, $\widetilde{\text{KEM}} \neq g_T^{\xi} = \text{KEM}$ with all but negligible probability $1/q + 1/q(1 - 1/q)$, i.e., except when $\omega = 0$ or $\zeta = 0$ but $\sum\limits_{j'\in[m]} \Omega'_{j'}b'_{j'} + \sum\limits_{j\in[n]} \Omega_j b_j \neq 0$.

► **Security**:

Theorem 4.1: *The proposed* KEM *version of our* PHPE *scheme is* $(\mathsf{poly}, 1, \mathsf{poly})$*-simulation-based adaptively strongly partially hiding* (*as per the security model described in Section 2.6*) *with respect to a simulator that runs in super-polynomial time under the* SXDLIN *assumption.*

The proof sketch of Theorem 4.1 is presented in Appendix C. The following corollary is also immediate from the relation between indistinguishability-based and simulation-based security with respect to super-polynomial simulators [AGVW13], as well as that between single- and multi-ciphertext security in the indistinguishability-based setting for FE.

Corollary 4.1: *The proposed* KEM *version of our* PHPE *scheme is* $(\mathsf{poly}, \mathsf{poly}, \mathsf{poly})$*-indistinguishability-based adaptively strongly partially hiding* (*as per the security model described in Section 2.6*) *under the* SXDLIN *assumption.*

References

ABSV15. Prabhanjan Ananth, Zvika Brakerski, Gil Segev, and Vinod Vaikuntanathan. From selective to adaptive security in functional encryption. In *CRYPTO 2015*, pages 657–677. Springer, 2015.

ACD$^+$12. Masayuki Abe, Melissa Chase, Bernardo David, Markulf Kohlweiss, Ryo Nishimaki, and Miyako Ohkubo. Constant-size structure-preserving signatures: Generic constructions and simple assumptions. In *ASIACRYPT 2012*, pages 4–24. Springer, 2012.

Agr17. Shweta Agrawal. Stronger security for reusable garbled circuits, general definitions and attacks. In *CRYPTO 2017*, pages 3–35. Springer, 2017.

AGVW13. Shweta Agrawal, Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Functional encryption: New perspectives and lower bounds. In *CRYPTO 2013*, pages 500–518. Springer, 2013.

AJ15. Prabhanjan Ananth and Abhishek Jain. Indistinguishability obfuscation from compact functional encryption. In *CRYPTO 2015*, pages 308–326. Springer, 2015.

AJS15. Prabhanjan Ananth, Abhishek Jain, and Amit Sahai. Indistinguishability obfuscation from functional encryption for simple functions. Cryptology ePrint Archive, Report 2015/730, 2015.

Att14. Nuttapong Attrapadung. Dual system encryption via doubly selective security : Framework, fully secure functional encryption for regular languages, and more. In *EUROCRYPT 2014*, pages 557–577. Springer, 2014.

BGG$^+$14. Dan Boneh, Craig Gentry, Sergey Gorbunov, Shai Halevi, Valeria Nikolaenko, Gil Segev, Vinod Vaikuntanathan, and Dhinakaran Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe and compact garbled circuits. In *EUROCRYPT 2014*, pages 533–556. Springer, 2014.

BGJT14. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *EUROCRYPT 2014*, pages 1–16. Springer, 2014.

BO13. Mihir Bellare and Adam O'Neill. Semantically-secure functional encryption: Possibility results, impossibility results and the quest for a general definition. In *CANS 2013*, pages 218–234. Springer, 2013.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In *TCC 2011*, pages 253–273. Springer, 2011.

BV15. Nir Bitansky and Vinod Vaikuntanathan. Indistinguishability obfuscation from functional encryption. In *FOCS 2015*, pages 171–190. IEEE, 2015.

BW07. Dan Boneh and Brent Waters. Conjunctive, subset, and range queries on encrypted data. In *TCC 2007*, pages 535–554. Springer, 2007.

CGW15. Jie Chen, Romain Gay, and Hoeteck Wee. Improved dual system abe in prime-order groups via predicate encodings. In *EUROCRYPT 2015*, pages 595–624. Springer, 2015.

CW14. Jie Chen and Hoeteck Wee. Semi-adaptive attribute-based encryption and improved delegation for boolean formula. In *SCN 2014*, pages 277–297. Springer, 2014.

DCIJ$^+$13. Angelo De Caro, Vincenzo Iovino, Abhishek Jain, Adam O'Neill, Omer Paneth, and Giuseppe Persiano. On the achievability of simulation-based security for functional encryption. In *CRYPTO 2013*, pages 519–535. Springer, 2013.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In *EUROCRYPT 2010*, pages 44–61. Springer, 2010.

GGMZ13. Faruk Göloğlu, Robert Granger, Gary McGuire, and Jens Zumbrägel. On the function field sieve and the impact of higher splitting probabilities. In *CRYPTO 2013*, pages 109–128. Springer, 2013.

GKW16. Rishab Goyal, Venkata Koppula, and Brent Waters. Semi-adaptive security and bundling functionalities made generic and easy. In *TCC 2016*, pages 361–388. Springer, 2016.

GPSW06. Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *CCS 2006*, pages 89–98. ACM, 2006.

Gui13. Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In *ACNS 2013*, pages 357–372. Springer, 2013.

GVW15a. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Attribute-based encryption for circuits. *Journal of the ACM (JACM)*, 62(6), 2015.

GVW15b. Sergey Gorbunov, Vinod Vaikuntanathan, and Hoeteck Wee. Predicate encryption for circuits from lwe. In *CRYPTO 2015*, pages 503–523. Springer, 2015.

IK97. Yuval Ishai and Eyal Kushilevitz. Private simultaneous messages protocols with applications. In *Theory of Computing and Systems, 1997, Proceedings of the Fifth Israeli Symposium on*, pages 174–184. IEEE, 1997.

IK02. Yuval Ishai and Eyal Kushilevitz. Perfect constant-round secure computation via perfect randomizing polynomials. In *ICALP 2002*, pages 244–256. Springer, 2002.

IW14. Yuval Ishai and Hoeteck Wee. Partial garbling schemes and their applications. In *ICALP 2014*, pages 650–662. Springer, 2014.

Jou13a. Antoine Joux. Faster index calculus for the medium prime case application to 1175-bit and 1425-bit finite fields. In *EUROCRYPT 2013*, pages 177–193. Springer, 2013.

Jou13b. Antoine Joux. A new index calculus algorithm with complexity l (1/4+ o (1)) in small characteristic. In *SAC 2013*, pages 355–379. Springer, 2013.

KSW08. Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT 2008*, pages 146–162. Springer, 2008.

LOS$^+$10. Allison Lewko, Tatsuaki Okamoto, Amit Sahai, Katsuyuki Takashima, and Brent Waters. Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *EUROCRYPT 2010*, pages 62–91. Springer, 2010.

LW12. Allison Lewko and Brent Waters. New proof methods for attribute-based encryption: Achieving full security through selective techniques. In *CRYPTO 2012*, pages 180–198. Springer, 2012.

O'N10. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010.

OT09. Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In *ASIACRYPT 2009*, pages 214–231. Springer, 2009.

OT10. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure functional encryption with general relations from the decisional linear assumption. In *CRYPTO 2010*, pages 191–208. Springer, 2010.

OT12a. Tatsuaki Okamoto and Katsuyuki Takashima. Adaptively attribute-hiding (hierarchical) inner product encryption. In *EUROCRYPT 2012*, pages 591–608. Springer, 2012.

OT12b. Tatsuaki Okamoto and Katsuyuki Takashima. Fully secure unbounded inner-product and attribute-based encryption. In *ASIACRYPT 2012*, pages 349–366. Springer, 2012.

OT13. Tatsuaki Okamoto and Katsuyuki Takashima. Efficient (hierarchical) inner-product encryption tightly reduced from the decisional linear assumption. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 96(1):42–52, 2013.

TAO16. Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In *ISC 2016*, pages 408–425. Springer, 2016.

Wat09. Brent Waters. Dual system encryption: Realizing fully secure ibe and hibe under simple assumptions. In *CRYPTO 2009*, pages 619–636. Springer, 2009.

Wee14. Hoeteck Wee. Dual system encryption via predicate encodings. In *TCC 2014*, pages 616–637. Springer, 2014.

Wee17. Hoeteck Wee. Attribute-hiding predicate encryption in bilinear groups, revisited. In *TCC 2017*, pages 206–233. Springer, 2017.

Appendix

A Some Necessary Computational Problems Derived from the SXDLIN Problem

In this section, we define some decisional problems. We rely on the hardness of these decisional problems for deriving security of our PHPE construction. The hardness of these decisional problems can be reduced to that of the SXDLIN problem, as shown in Lemmas A.1–A.4 below.

Definition A.1 (Problem 1): Problem 1 is to guess the bit $\widehat{\beta} \xleftarrow{\mathsf{U}} \{0,1\}$, given $\varrho_{\widehat{\beta}}^{\mathsf{P1}} = (\mathsf{params}, \{\mathbb{B}_\imath, \widetilde{\mathbb{B}}_\imath^*\}_{\imath\in[0,n'+n]}, \boldsymbol{f}^{(0,\widehat{\beta})}, \{\boldsymbol{f}^{(\imath,1,\widehat{\beta})}, \boldsymbol{f}^{(\imath,2)}\}_{\imath\in[n'+n]})$, where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathrm{OB}}(n'+n, (6, \overbrace{9, \dots, 9}^{n'+n}));\\
&\widetilde{\omega}, \widetilde{\varphi}, \widetilde{\vartheta} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0^* = \{\boldsymbol{b}^{*(0,1)}, \boldsymbol{b}^{*(0,3)}, \dots, \boldsymbol{b}^{*(0,6)};\\
&\boldsymbol{f}^{(0,0)} = (\widetilde{\omega}, 0, 0, 0, 0, \widetilde{\varphi})_{\mathbb{B}_0}, \boldsymbol{f}^{(0,1)} = (\widetilde{\omega}, \widetilde{\vartheta}, 0, 0, 0, \widetilde{\varphi})_{\mathbb{B}_0};\\
&\widetilde{\mathbb{B}}_\imath^* = \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,4)}, \dots, \boldsymbol{b}^{*(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1, 0) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
&\boldsymbol{f}^{(\imath,1,0)} = (\widetilde{\omega}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}\\
&\boldsymbol{f}^{(\imath,1,1)} = (\widetilde{\omega}\vec{e}^{(1)}, \widetilde{\vartheta}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}\\
&\boldsymbol{f}^{(\imath,2)} = \widetilde{\omega}\boldsymbol{b}^{(\imath,2)}
\end{aligned}\right\} \text{ for } \imath \in [n'+n].
\end{aligned}$$

For any security parameter λ, the advantage of any probabilistic algorithm $\mathcal{B}$ in deciding Problem 1 is defined as

$$\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho_0^{\mathsf{P1}})] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho_1^{\mathsf{P1}})]|.$$

Lemma A.1: *For any probabilistic algorithm $\mathcal{B}$, there exists a probabilistic algorithm $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}_{\mathcal{B}}^{\mathsf{P1}}(\lambda) \leq \mathsf{Adv}_{\mathcal{E}}^{\mathsf{SXDLIN}}(\lambda) + 6/q$.*

Proof: Problem 1 is essentially the same as the Basic Problem 1 of [OT10] with some specific choice of the number of bases and their dimension. Thus, Lemma A.1 can be proven in an analogous fashion to Lemmas 15 and 16 of [OT10], which reduce the hardness of the Basic Problem 1 to that of the DLIN problem in symmetric bilinear group setting. □

Definition A.2 (Problem 2): Problem 2 is to guess the bit $\widehat{\beta} \xleftarrow{\mathsf{U}} \{0,1\}$, given $\varrho^{\mathsf{P2}}_{\widehat{\beta}} = (\mathsf{params}, \{\widetilde{\mathbb{B}}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{y}^{(0,\widehat{\beta})}, \boldsymbol{f}^{(0)}, \{\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})}, \boldsymbol{f}^{(\imath,\ell)}\}_{\imath\in[n'+n],\ell\in[2]})$, where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\delta}, \widetilde{\kappa}, \widetilde{\pi}, \widetilde{\omega}, \widetilde{\tau} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,3)}, \ldots, \boldsymbol{b}^{(0,6)}\};\\
&\boldsymbol{y}^{(0,0)} = (\widetilde{\delta}, 0, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0}, \boldsymbol{y}^{(0,1)} = (\widetilde{\delta}, \widetilde{\pi}, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0};\\
&\boldsymbol{f}^{(0)} = (\widetilde{\omega}, \widetilde{\tau}, 0, 0, 0, 0)_{\mathbb{B}_0};\\
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,5)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0), \vec{e}^{(2)} = (0,1) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
\boldsymbol{y}^{(\imath,\ell,0)} &= (\widetilde{\delta}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{y}^{(\imath,\ell,1)} &= (\widetilde{\delta}\vec{e}^{(\ell)}, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{f}^{(\imath,\ell)} &= (\widetilde{\omega}\vec{e}^{(\ell)}, \widetilde{\tau}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, 0)_{\mathbb{B}_\imath}
\end{aligned}\right\} \text{ for } \imath \in [n'+n], \ell \in [2].
\end{aligned}$$

For any security parameter λ, the advantage of any probabilistic algorithm $\mathcal{B}$ in deciding Problem 2 is defined as

$$\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P2}}_0)] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P2}}_1)]|.$$

Lemma A.2: *For any probabilistic algorithm $\mathcal{B}$, there exists a probabilistic algorithm $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SXDLIN}}_{\mathcal{E}}(\lambda) + 8/q$.*

Proof: Problem 2 is essentially the same as the Basic Problem 2 of [OT10] with some specific choice of the number of bases and their dimension. Thus, Lemma A.2 can be proven in a similar manner to Lemmas 15 and 18 in [OT10], which reduce the hardness of the Basic Problem 2 to that of the DLIN problem in symmetric bilinear group setting. □

Definition A.3 (Problem 3): Problem 3 is to guess the bit $\widehat{\beta} \xleftarrow{\mathsf{U}} \{0,1\}$, given $\varrho^{\mathsf{P3}}_{\widehat{\beta}} = (\mathsf{params}, \{\widetilde{\mathbb{B}}_\imath, \widetilde{\mathbb{B}}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{y}^{(0,\widehat{\beta})}, \boldsymbol{f}^{(0)}, \{\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})}, \boldsymbol{f}^{(\imath,\ell)}\}_{\imath\in[n'+n],\ell\in[2]})$, where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\pi}, \widetilde{\pi}', \widetilde{\kappa}, \widetilde{\tau}, \widetilde{\theta} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,4)}, \ldots, \boldsymbol{b}^{(0,6)}\}, \widetilde{\mathbb{B}}^*_0 = \{\boldsymbol{b}^{*(0,1)}, \boldsymbol{b}^{*(0,3)}, \ldots, \boldsymbol{b}^{*(0,6)}\};\\
&\boldsymbol{y}^{(0,0)} = (0, \widetilde{\pi}, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0}, \boldsymbol{y}^{(0,1)} = (0, 0, \widetilde{\pi}', 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0};\\
&\boldsymbol{f}^{(0)} = (0, \widetilde{\tau}, \widetilde{\theta}, 0, 0, 0)_{\mathbb{B}_0};\\
&\left.\begin{aligned}
\widetilde{\mathbb{B}}_\imath &= \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,7)}, \ldots, \boldsymbol{b}^{(\imath,9)}\}\\
\widetilde{\mathbb{B}}^*_\imath &= \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,5)}, \ldots, \boldsymbol{b}^{*(\imath,9)}\}
\end{aligned}\right\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0), \vec{e}^{(2)} = (0,1) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
\boldsymbol{y}^{(\imath,\ell,0)} &= (\vec{0}^2, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{y}^{(\imath,\ell,1)} &= (\vec{0}^2, \vec{0}^2, \widetilde{\pi}'\vec{e}^{(\ell)}, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{f}^{(\imath,\ell)} &= (\vec{0}^2, \widetilde{\tau}\vec{e}^{(\ell)}, \widetilde{\theta}\vec{e}^{(\ell)}, \vec{0}^2, 0)_{\mathbb{B}_\imath}
\end{aligned}\right\} \text{ for } \imath \in [n'+n], \ell \in [2].
\end{aligned}$$

For any security parameter λ, the advantage of any probabilistic algorithm $\mathcal{B}$ in deciding Problem 3 is defined as

$$\mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P3}}_0)] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P3}}_1)]|.$$

Lemma A.3: *For any probabilistic algorithm $\mathcal{B}$, there exists a probabilistic algorithm $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{E}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SXDLIN}}_{\mathcal{E}}(\lambda) + 9/q$.*

Proof: Problem 3 is essentially the same as the multi-basis version of the Problem 3 of [OT12a] with some specific choice of the dimension of the bases. Thus, Lemma A.3 can be proven in an analogous fashion to that of Lemma 4 of [OT12a]. □

Definition A.4 (Problem 4): Problem 4 is to guess the bit $\widehat{\beta} \xleftarrow{\mathsf{U}} \{0,1\}$ given $\varrho^{\mathsf{P4}}_{\widehat{\beta}} = (\mathsf{params}, \mathbb{B}_0, \mathbb{B}^*_0, \{\widetilde{\mathbb{B}}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}, \{\boldsymbol{y}^{(\imath,1,\widehat{\beta})}\}_{\imath\in[n'+n]})$, where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathrm{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\pi}, \widetilde{\kappa}_1, \widetilde{\kappa}_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,4)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0) \in \mathbb{F}_q^2;\\
&\left.\begin{array}{l}\boldsymbol{y}^{(\imath,1,0)} = (\vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}^*_\imath}\\ \boldsymbol{y}^{(\imath,1,1)} = (\vec{0}^2, \widetilde{\pi}\vec{e}^{(1)}, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}^*_\imath}\end{array}\right\} \text{ for } \imath \in [n'+n].
\end{aligned}$$

For any security parameter λ, the advantage of any probabilistic algorithm $\mathcal{B}$ in deciding Problem 4 is defined as

$$\mathsf{Adv}^{\mathsf{P4}}_{\mathcal{B}}(\lambda) = |\mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P4}}_0)] - \mathsf{Pr}[1 \xleftarrow{\mathsf{R}} \mathcal{B}(\varrho^{\mathsf{P4}}_1)]|.$$

Lemma A.4: *For any probabilistic algorithm $\mathcal{B}$, there exists a probabilistic machine $\mathcal{E}$, whose running time is essentially the same as that of $\mathcal{B}$, such that for any security parameter λ, $\mathsf{Adv}^{\mathsf{P4}}_{\mathcal{B}}(\lambda) \leq \mathsf{Adv}^{\mathsf{SXDLIN}}_{\mathcal{E}}(\lambda) + 6/q$.*

Proof: Observe that Problem 4 is essentially the same as Problem 1 (Definition A.1) considered in $\mathbb{V}_2$ with some re-ordering of the vectors of the dual orthogonal bases. Thus, the proof of Lemma A.4 is the same as that of Lemma A.1 with certain appropriate changes that are easy to figure out. □

B Lemmas for the Proof of Theorem 3.1

Lemma B.1: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_1$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(0)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(1)}_{\mathcal{H}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P1}}_{\mathcal{B}_1}(\lambda)$.*

Proof: In order to prove Lemma B.1, we construct below a probabilistic algorithm $\mathcal{B}_1$ against Problem 1 using a stateful probabilistic adversary $\mathcal{H}$ for distinguishing between Hyb_0 and Hyb_1 as a black-box sub-routine. Suppose $\mathcal{B}_1$ is given an instance of Problem 1,

$$\varrho^{\mathsf{P1}}_{\widehat{\beta}} = (\mathsf{params}, \{\mathbb{B}_\imath, \widetilde{\mathbb{B}}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{f}^{(0,\widehat{\beta})}, \{\boldsymbol{f}^{(\imath,1,\widehat{\beta})}, \boldsymbol{f}^{(\imath,2)}\}_{\imath\in[n'+n]}),$$

where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\omega}, \widetilde{\varphi}, \widetilde{\vartheta} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0^* = \{\boldsymbol{b}^{*(0,1)}, \boldsymbol{b}^{*(0,3)}, \ldots, \boldsymbol{b}^{*(0,6)};\\
&\boldsymbol{f}^{(0,0)} = (\widetilde{\omega}, 0, 0, 0, 0, \widetilde{\varphi})_{\mathbb{B}_0}, \boldsymbol{f}^{(0,1)} = (\widetilde{\omega}, \widetilde{\vartheta}, 0, 0, 0, \widetilde{\varphi})_{\mathbb{B}_0};\\
&\widetilde{\mathbb{B}}_\imath^* = \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,4)}, \ldots, \boldsymbol{b}^{*(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
&\boldsymbol{f}^{(\imath,1,0)} = (\widetilde{\omega}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}\\
&\boldsymbol{f}^{(\imath,1,1)} = (\widetilde{\omega}\vec{e}^{(1)}, \widetilde{\vartheta}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}\\
&\boldsymbol{f}^{(\imath,2)} = \widetilde{\omega}\boldsymbol{b}^{(\imath,2)}
\end{aligned}\right\} \text{ for } \imath \in [n'+n].
\end{aligned}$$

$\mathcal{B}_1$ interacts with $\mathcal{H}$ as follows:

1. First, $\mathcal{B}_1$ gives $\mathcal{H}$ the public parameters MPK $= (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}\}_{\imath\in[n'+n]})$, all of which are taken from the given Problem 1 instance.
2. For all $h \in [q_{\text{KEY}}]$, in response to a decryption key query of $\mathcal{H}$ for some function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$, $\mathcal{B}_1$ provides $\mathcal{H}$ with a decryption key SK$(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, whose components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ are generated as in Eq. (3.2) using $\{\widetilde{\mathbb{B}}_\imath^*\}_{\imath\in[n'+n]}$ included within the given Problem 1 instance.
3. When $\mathcal{B}_1$ receives the ciphertext query from $\mathcal{H}$ for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{B}_1$ samples $\{\widehat{\varphi}'_{\iota'}\}_{\iota'\in[n']}, \{\widehat{\varphi}_\iota\}_{\iota\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes
$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= \boldsymbol{f}^{(\iota',1,\widehat{\beta})} + x_{\iota'}\boldsymbol{f}^{(\iota',2)} + \widehat{\varphi}'_{\iota'}\boldsymbol{b}^{(\iota',9)} \text{ for } \iota' \in [n'],\\
\boldsymbol{c}^{\iota)} &= \boldsymbol{f}^{(n'+\iota,1,\widehat{\beta})} + z_\iota\boldsymbol{f}^{(n'+\iota,2)} + \widehat{\varphi}_\iota\boldsymbol{b}^{(n'+\iota,9)} \text{ for } \iota \in [n],
\end{aligned}$$
and hands $\mathcal{H}$ the ciphertext CT $= (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$.
4. $\mathcal{H}$ eventually outputs a bit $\beta \in \{0,1\}$. $\mathcal{B}_1$ outputs $\widehat{\beta}' = \beta$ as its guess bit in its Problem 1 challenge.

Observe that when $\widehat{\beta} = 0$, i.e., $\boldsymbol{f}^{(\imath,1,\widehat{\beta})} = \boldsymbol{f}^{(\imath,1,0)} = (\widetilde{\omega}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}$ for $\imath \in [n'+n]$, the components of the ciphertext CT returned by $\mathcal{B}_1$ to $\mathcal{H}$ take the form

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= (\widetilde{\omega}(1, x_{\iota'}), \vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\varphi} + \widehat{\varphi}'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],\\
\boldsymbol{c}^{(\iota)} &= (\widetilde{\omega}(1, z_\iota), \vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\varphi} + \widehat{\varphi}_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],
\end{aligned}$$

which coincides with that in Hyb_0 (Eq. (3.1)), where we have $\omega = \widetilde{\omega}$, $\varphi'_{\iota'} = \widetilde{\varphi} + \widehat{\varphi}'_{\iota'}$ for $\iota' \in [n']$, and $\varphi_\iota = \widetilde{\varphi} + \widehat{\varphi}_\iota$ for $\iota \in [n]$. On the other hand, if $\widehat{\beta} = 1$, i.e., $\boldsymbol{f}^{(\imath,1,\widehat{\beta})} = \boldsymbol{f}^{(\imath,1,1)} = (\widetilde{\omega}\vec{e}^{(1)}, \widetilde{\vartheta}\vec{e}^{(1)}, \vec{0}^2, \vec{0}^2, \widetilde{\varphi})_{\mathbb{B}_\imath}$ for $\imath \in [n'+n]$, then the components of the ciphertext CT given by $\mathcal{B}_1$ to $\mathcal{H}$ take the form

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= (\widetilde{\omega}(1, x_{\iota'}), (\widetilde{\vartheta}, 0), \vec{0}^2, \vec{0}^2, \widetilde{\varphi} + \widehat{\varphi}'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],\\
\boldsymbol{c}^{(\iota)} &= (\widetilde{\omega}(1, z_\iota), (\widetilde{\vartheta}, 0), \vec{0}^2, \vec{0}^2, \widetilde{\varphi} + \widehat{\varphi}_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],
\end{aligned}$$

which is identical to the one in Hyb_1 (Eq. (3.3)), where we have $\omega = \widetilde{\omega}, \vartheta = \widetilde{\vartheta}, \varphi'_{\iota'} = \widetilde{\varphi} + \widehat{\varphi}'_{\iota'}$ for $\iota' \in [n']$, and $\varphi_\iota = \widetilde{\varphi} + \widehat{\varphi}_\iota$ for $\iota \in [n]$. Clearly, the variables $\omega, \vartheta, \{\varphi'_{\iota'}\}_{\iota'\in[n']}, \{\varphi_\iota\}_{\iota\in[n]}$ simulated by $\mathcal{B}_1$ are uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$. Moreover, for all $h \in [q_{\text{KEY}}]$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of the h^{th} decryption key SK(f_h)

returned by $\mathcal{B}_1$ are generated as in Eq. (3.2), which is their proper forms in both Hyb_0 and Hyb_1. Also, the public parameters MPK given to $\mathcal{H}$ by $\mathcal{B}_1$ have the same distribution as in both Hyb_0 and Hyb_1. Hence, it is clear that the distribution of the view of $\mathcal{H}$ simulated by $\mathcal{B}_1$ given a $\mathsf{Problem\ 1}$ instance $\varrho^{\mathsf{P1}}_{\widehat{\beta}}$ for $\widehat{\beta} \in \{0,1\}$, coincides with that in Hyb_0 or that in Hyb_1 according as $\widehat{\beta} = 0$ or 1. Hence the lemma follows. □

Lemma B.2: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}(\chi-1)\text{-}4)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}1)}_{\mathcal{H}}(\lambda)| \le 2/q$ for all $\chi \in [q_{\text{KEY-PRE}}]$.*

Proof: We will consider the following two cases separately:

▶ **Case (I) ($\chi = 1$, i.e., proof for $|\mathsf{Adv}^{(1)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}1\text{-}1)}_{\mathcal{H}}(\lambda)| \le 2/q$)**

In order to prove Lemma B.2 in this case, we first introduce the intermediate hybrid $\mathsf{Hyb}_{1'}$ described below. Next, we show the equivalence of the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ in Hyb_1 and that in $\mathsf{Hyb}_{1'}$ (Claim B.1), as well as those in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ and in $\mathsf{Hyb}_{1'}$ (Claim B.2).

$\mathsf{Hyb}_{1'}$: This experiment is the same as Hyb_1 except that in this experiment the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \boxed{\vec{a}'^{(\iota')}}, \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_{\iota}), \boxed{\vec{a}^{(\iota)}}, \vec{0}^2, \varphi_{\iota})_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{B.1}$$

where $(\{\vec{a}'^{(\iota')}\}_{\iota'\in[n']}, \{\vec{a}^{(\iota)}\}_{\iota\in[n]}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^4 \backslash \{\vec{0}^4\})^{n'} \times (\mathbb{F}_q^4 \backslash \{\vec{0}^4\})^{n}$, and all the other variables are generated as in Hyb_1.

Claim B.1: *The distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ in Hyb_1 and that in $\mathsf{Hyb}_{1'}$ are equivalent except with probability $1/q$.*

Proof: Consider the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ in Hyb_1. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\iota, \mathbb{D}^*_\iota\}_{\iota\in[n'+n]}$ from the original set of bases $\{\mathbb{B}_\iota, \mathbb{B}^*_\iota\}_{\iota\in[n'+n]}$ used in Hyb_1 as follows: Generate matrices $\{\boldsymbol{Z}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{Z}^{(\iota)}\}_{\iota\in[n]} \xleftarrow{\mathsf{U}} \mathsf{GL}(4, \mathbb{F}_q)$, compute the vectors

$$\begin{pmatrix} \boldsymbol{d}^{(\iota',3)} \\ \vdots \\ \boldsymbol{d}^{(\iota',6)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{-1} \begin{pmatrix} \boldsymbol{b}^{(\iota',3)} \\ \vdots \\ \boldsymbol{b}^{(\iota',6)} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{d}^{*(\iota',3)} \\ \vdots \\ \boldsymbol{d}^{*(\iota',6)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{\intercal} \begin{pmatrix} \boldsymbol{b}^{*(\iota',3)} \\ \vdots \\ \boldsymbol{b}^{*(\iota',6)} \end{pmatrix} \text{ for } \iota' \in [n'],$$

$$\begin{pmatrix} \boldsymbol{d}^{(n'+\iota,3)} \\ \vdots \\ \boldsymbol{d}^{(n'+\iota,6)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{-1} \begin{pmatrix} \boldsymbol{b}^{(n'+\iota,3)} \\ \vdots \\ \boldsymbol{b}^{(n'+\iota,6)} \end{pmatrix}, \begin{pmatrix} \boldsymbol{d}^{*(n'+\iota,3)} \\ \vdots \\ \boldsymbol{d}^{*(n'+\iota,6)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{\intercal} \begin{pmatrix} \boldsymbol{b}^{*(n'+\iota,3)} \\ \vdots \\ \boldsymbol{b}^{*(n'+\iota,6)} \end{pmatrix} \text{ for } \iota \in [n],$$

and set

$$\left.\begin{aligned} \mathbb{D}_{\iota'} &= \{\boldsymbol{b}^{(\iota',1)}, \boldsymbol{b}^{(\iota',2)}, \boldsymbol{d}^{(\iota',3)}, \ldots, \boldsymbol{d}^{(\iota',6)}, \boldsymbol{b}^{(\iota',7)}, \ldots, \boldsymbol{b}^{(\iota',9)}\} \\ \mathbb{D}^*_{\iota'} &= \{\boldsymbol{b}^{*(\iota',1)}, \boldsymbol{b}^{*(\iota',2)}, \boldsymbol{d}^{*(\iota',3)}, \ldots, \boldsymbol{d}^{*(\iota',6)}, \boldsymbol{b}^{*(\iota',7)}, \ldots, \boldsymbol{b}^{*(\iota',9)}\} \end{aligned}\right\} \text{ for } \iota' \in [n'],$$

$$\left.\begin{aligned} \mathbb{D}_{n'+\iota} &= \{\boldsymbol{b}^{(n'+\iota,1)}, \boldsymbol{b}^{(n'+\iota,2)}, \boldsymbol{d}^{(n'+\iota,3)}, \ldots, \boldsymbol{d}^{(n'+\iota,6)}, \boldsymbol{b}^{(n'+\iota,7)}, \ldots, \boldsymbol{b}^{(n'+\iota,9)}\} \\ \mathbb{D}^*_{n'+\iota} &= \{\boldsymbol{b}^{*(n'+\iota,1)}, \boldsymbol{b}^{*(n'+\iota,2)}, \boldsymbol{d}^{*(n'+\iota,3)}, \ldots, \boldsymbol{d}^{*(n'+\iota,6)}, \boldsymbol{b}^{*(n'+\iota,7)}, \ldots, \boldsymbol{b}^{*(n'+\iota,9)}\} \end{aligned}\right\} \text{ for } \iota \in [n]. \tag{B.2}$$

It can be readily observed that the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ are indeed dual orthonormal, and are distributed the same as the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[n'+n]}$.

Now, notice that the components of the ciphertext CT returned to $\mathcal{H}$ in Hyb_1 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ as

$$\begin{aligned}\boldsymbol{c}'^{(\iota')} &= (\omega(1,x_{\iota'}),(\vartheta,0),\vec{0}^2,\vec{0}^2,\varphi'_{\iota'})_{\mathbb{B}_{\iota'}}\\ &= (\omega(1,x_{\iota'}),\vec{a}'^{(\iota')},\vec{0}^2,\varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota'\in[n'],\\ \boldsymbol{c}^{(\iota)} &= (\omega(1,z_\iota),(\vartheta,0),\vec{0}^2,\vec{0}^2,\varphi_\iota)_{\mathbb{B}_{n'+\iota}}\\ &= (\omega(1,z_\iota),\vec{a}^{(\iota)},\vec{0}^2,\varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota\in[n],\end{aligned} \tag{B.3}$$

where $\vec{a}'^{(\iota')} = ((\vartheta,0),\vec{0}^2)\boldsymbol{Z}'^{(\iota')}$ for $\iota'\in[n']$, $\vec{a}^{(\iota)} = ((\vartheta,0),\vec{0}^2)\boldsymbol{Z}^{(\iota)}$ for $\iota\in[n]$, and all the other variables are generated as in Hyb_1. Clearly, the vector $((\vartheta,0),\vec{0}^2)\neq\vec{0}^4$ with all but negligible probability $1/q$, i.e., except when $\vartheta=0$. Then, $(\{\vec{a}'^{(\iota')}\}_{\iota'\in[n']},\{\vec{a}^{(\iota)}\}_{\iota\in[n]})$ is uniformly distributed in $(\mathbb{F}_q^4\backslash\{\vec{0}^4\})^{n'}\times(\mathbb{F}_q^4\backslash\{\vec{0}^4\})^n$ with all but negligible probability $1/q$, and are independent of all the other variables.

Also, for all $h\in[q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ provided to $\mathcal{H}$ in Hyb_1 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ as

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'},\alpha_{h,j'}),\vec{0}^2,\vec{0}^2,\vec{\kappa}'^{(h,j')},0)_{\mathbb{B}^*_{\rho_h(j')}}\\ &= ((\gamma_{h,j'},\alpha_{h,j'}),\vec{0}^2,\vec{0}^2,\vec{\kappa}'^{(h,j')},0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j'\in[m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j},\zeta_h),\vec{0}^2,\vec{0}^2,\vec{\kappa}^{(h,j)},0)_{\mathbb{B}^*_{n'+j}}\\ &= ((\sigma_{h,j},\zeta_h),\vec{0}^2,\vec{0}^2,\vec{\kappa}^{(h,j)},0)_{\mathbb{D}^*_{n'+j}} \text{ for } j\in[n],\end{aligned} \tag{B.4}$$

where all the variables are generated as in Hyb_1.

Now, note that in the view of $\mathcal{H}$, both the original sets of bases $\{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ are consistent with the public parameters MPK. Further, for all $h\in[q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their forms as in Eq. (3.2), which are their proper forms in $\mathsf{Hyb}_{1'}$, under the basis transformation. Finally, since the RHS of Eq. (B.3) and that in Eq. (B.1) have the same form except with probability $1/q$, it follows that the components of the ciphertext CT returned to $\mathcal{H}$ in Hyb_1 can be conceptually changed to those in $\mathsf{Hyb}_{1'}$ except with probability $1/q$. Hence the claim follows. □

Claim B.2: *The distribution* $(\text{MPK},\{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]},\text{CT})$ *in* $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ *and that in* $\mathsf{Hyb}_{1'}$ *are equivalent except with probability* $1/q^2$.

Proof: Consider the distribution $(\text{MPK},\{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]},\text{CT})$ in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ from the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[n'+n]}$ used in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ just as in Eq. (B.2) in the proof of Claim B.1. Observe that the components of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}_\imath^*\}_{\imath\in[n'+n]}$ as

$$\begin{aligned}\boldsymbol{c}'^{(\iota')} &= (\omega(1,x_{\iota'}),\tau(1,x_{\iota'}),\theta(1,x_{\iota'}),\vec{0}^2,\varphi'_{\iota'})_{\mathbb{B}_{\iota'}}\\ &= (\omega(1,x_{\iota'}),\vec{a}'^{(\iota')},\vec{0}^2,\varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota'\in[n'],\\ \boldsymbol{c}^{(\iota)} &= (\omega(1,z_\iota),\tau(1,z_\iota),\theta(1,s_\iota),\vec{0}^2,\varphi_\iota)_{\mathbb{B}_{n'+\iota}}\\ &= (\omega(1,z_\iota),\vec{a}^{(\iota)},\vec{0}^2,\varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota\in[n],\end{aligned} \tag{B.5}$$

where $\vec{a}'^{(\iota')} = (\tau(1,x_{\iota'}),\theta(1,x_{\iota'}))\boldsymbol{Z}'^{(\iota')}$ for $\iota'\in[n']$, $\vec{a}^{(\iota)} = (\tau(1,z_\iota),\theta(1,s_\iota))\boldsymbol{Z}^{(\iota)}$ for $\iota\in[n]$, and all the other variables are formed as in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$. Clearly, the vectors $(\tau(1,x_{\iota'}),\theta(1,x_{\iota'}))\neq\vec{0}^4$ for all

$\iota' \in [n']$ and the vectors $(\tau(1, z_\iota), \theta(1, s_\iota)) \neq \vec{0}^4$ for all $\iota \in [n]$ with all but negligible probability $1/q^2$, i.e., except when $\tau = 0$ and $\theta = 0$. Then, $(\{\vec{a}'^{(\iota')}\}_{\iota' \in [n']}, \{\vec{a}^{(\iota)}\}_{\iota \in [n]})$ is uniformly distributed in $(\mathbb{F}_q^4 \backslash \{\vec{0}^4\})^{n'} \times (\mathbb{F}_q^4 \backslash \{\vec{0}^4\})^n$ with all but negligible probability $1/q^2$, and are independent of all the other variables.

Also, for all $h \in [q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ provided to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ can be expressed in terms of the new set of bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ in the same fashion as in Eq. (B.4).

As in the proof of Claim B.1, in the view of $\mathcal{H}$, both the original sets of bases $\{\mathbb{B}_\iota, \mathbb{B}_\iota^*\}_{\iota \in [n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ are consistent with the public parameters MPK. Further, similar to the proof of Claim B.1, for all $h \in [q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their forms as in Eq. (3.2), which is their proper forms in $\mathsf{Hyb}_{1'}$, under the basis transformation. Finally, since the RHS of Eq. (B.5) and that in Eq. (B.1) have the same form, it follows that the components of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}1\text{-}1}$ can be conceptually changed to those in $\mathsf{Hyb}_{1'}$ except with probability $1/q^2$. Hence the claim follows. □

► Case (II) ($\chi \geq 2$, i.e., proof for $|\mathsf{Adv}_{\mathcal{H}}^{(2\text{-}(\chi-1)\text{-}4)}(\lambda) - \mathsf{Adv}_{\mathcal{H}}^{(2\text{-}\chi\text{-}1)}(\lambda)| \leq 2/q$ for $\chi \in [2, q_{\text{KEY-PRE}}]$)

In order to prove Lemma B.2 in this case, we first introduce the intermediate hybrid $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ below. Next, we show the equivalence of the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ and that in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ (Claim B.3), as well as those in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ (Claim B.4).

$\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ ($\chi \in [2, q_{\text{KEY-PRE}}]$): This experiment is the same as $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ except that in this experiment the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \boxed{\vec{a}'^{(\iota')}}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \boxed{\vec{a}^{(\iota)}}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{B.6}$$

where $(\{\vec{a}'^{(\iota')}\}_{\iota' \in [n']}, \{\vec{a}^{(\iota)}\}_{\iota \in [n]}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^2 \backslash \{\vec{0}^2\})^{n'} \times (\mathbb{F}_q^2 \backslash \{\vec{0}^2\})^n$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$.

Claim B.3: *The distribution* $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ *in* $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ *and that in* $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ *are equivalent except with probability* $1/q$.

Proof: Consider the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ from the original set of bases $\{\mathbb{B}_\iota, \mathbb{B}_\iota^*\}_{\iota \in [n'+n]}$ used in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ as follows: Generate matrices $\{\boldsymbol{Z}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{Z}^{(\iota)}\}_{\iota \in [n]} \xleftarrow{\mathsf{U}} \mathsf{GL}(2, \mathbb{F}_q)$, define

$\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^{\intercal}$ for $\iota' \in [n']$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^{\intercal}$ for $\iota \in [n]$, compute the vectors

$$\begin{pmatrix}\boldsymbol{d}^{(\iota',3)}\\ \boldsymbol{d}^{(\iota',4)}\end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{-1}\begin{pmatrix}\boldsymbol{b}^{(\iota',3)}\\ \boldsymbol{b}^{(\iota',4)}\end{pmatrix}, \quad \begin{pmatrix}\boldsymbol{d}^{*(\iota',3)}\\ \boldsymbol{d}^{*(\iota',4)}\end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{\intercal}\begin{pmatrix}\boldsymbol{b}^{*(\iota',3)}\\ \boldsymbol{b}^{*(\iota',4)}\end{pmatrix} \text{ for } \iota' \in [n'],$$

$$\begin{pmatrix}\boldsymbol{d}^{(n'+\iota,3)}\\ \boldsymbol{d}^{(n'+\iota,4)}\end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{-1}\begin{pmatrix}\boldsymbol{b}^{(n'+\iota,3)}\\ \boldsymbol{b}^{(n'+\iota,4)}\end{pmatrix}, \begin{pmatrix}\boldsymbol{d}^{*(n'+\iota,3)}\\ \boldsymbol{d}^{*(n'+\iota,4)}\end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{\intercal}\begin{pmatrix}\boldsymbol{b}^{*(n'+\iota,3)}\\ \boldsymbol{b}^{*(n'+\iota,4)}\end{pmatrix} \text{ for } \iota \in [n], \tag{B.7}$$

and set

$$\left.\begin{array}{l}\mathbb{D}_{\iota'} = \{\boldsymbol{b}^{(\iota',1)}, \boldsymbol{b}^{(\iota',2)}, \boldsymbol{d}^{(\iota',3)}, \boldsymbol{d}^{(\iota',4)}, \boldsymbol{b}^{(\iota',5)}, \ldots, \boldsymbol{b}^{(\iota',9)}\}\\ \mathbb{D}^*_{\iota'} = \{\boldsymbol{b}^{*(\iota',1)}, \boldsymbol{b}^{*(\iota',2)}, \boldsymbol{d}^{*(\iota',3)}, \boldsymbol{d}^{*(\iota',4)}, \boldsymbol{b}^{*(\iota',5)}, \ldots, \boldsymbol{b}^{*(\iota',9)}\}\end{array}\right\} \text{ for } \iota' \in [n'],$$

$$\left.\begin{array}{l}\mathbb{D}_{n'+\iota} = \{\boldsymbol{b}^{(n'+\iota,1)}, \boldsymbol{b}^{(n'+\iota,2)}, \boldsymbol{d}^{(n'+\iota,3)}, \boldsymbol{d}^{(n'+\iota,4)}, \boldsymbol{b}^{(n'+\iota,5)}, \ldots, \boldsymbol{b}^{(n'+\iota,9)}\}\\ \mathbb{D}^*_{n'+\iota} = \{\boldsymbol{b}^{*(n'+\iota,1)}, \boldsymbol{b}^{*(n'+\iota,2)}, \boldsymbol{d}^{*(n'+\iota,3)}, \boldsymbol{d}^{*(n'+\iota,4)}, \boldsymbol{b}^{*(n'+\iota,5)}, \ldots, \boldsymbol{b}^{*(n'+\iota,9)}\}\end{array}\right\} \text{ for } \iota \in [n].$$

It can be readily observed that the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ are indeed dual orthonormal, and are distributed the same as the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$.

Now, notice that the components of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ as

$$\begin{aligned}\boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \tau(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}}\\ &= (\omega(1, x_{\iota'}), \vec{a}'^{(\iota')}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota' \in [n'],\\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \tau(1, s_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}}\\ &= (\omega(1, z_\iota), \vec{a}^{(\iota)}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota \in [n],\end{aligned} \tag{B.8}$$

where $\vec{a}'^{(\iota')} = \tau(1, x_{\iota'})\boldsymbol{Z}'^{(\iota')}$ for $\iota' \in [n']$, $\vec{a}^{(\iota)} = \tau(1, s_\iota)\boldsymbol{Z}^{(\iota)}$ for $\iota \in [n]$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$. Clearly, the vectors $\tau(1, x_{\iota'}) \neq \vec{0}^2$ for all $\iota' \in [n']$ and $\tau(1, s_\iota) \neq \vec{0}^2$ for all $\iota \in [n]$ with all but negligible probability $1/q$, i.e., except when $\tau = 0$. Then, $(\{\vec{a}'^{(\iota')}\}_{\iota'\in[n']}, \{\vec{a}^{(\iota)}\}_{\iota\in[n]})$ is uniformly distributed in $(\mathbb{F}_q^2\backslash\{\vec{0}^2\})^{n'} \times (\mathbb{F}_q^2\backslash\{\vec{0}^2\})^n$ with all but negligible probability $1/q$, and are independent of all the other variables.

Also, for $h \in [q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ provided to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ as

(a) $(h < \chi)$

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}\\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}\\ &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n].\end{aligned} \tag{B.9}$$

(b) $(h \geq \chi)$

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}\\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}\\ &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],\end{aligned} \tag{B.10}$$

where all the variables are generated as in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$.

Now, note that in the view of $\mathcal{H}$, both the original sets of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ are consistent with the public parameters MPK. Further, for $h \in [q_{\text{KEY}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their forms as in Eq. (3.7), if $h < \chi$, and as in Eq. (3.2), if $h \geq \chi$, which are their proper forms in the respective cases in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$, under the basis transformation. Finally, since the RHS of Eq. (B.8) and that of Eq. (B.6) have the same form except with probability $1/q$, it follows that the components of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4}$ can be conceptually changed to those in $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ except with probability $1/q$. Hence the claim follows. □

Claim B.4: *The distribution* $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ *in* $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ *and that in* $\mathsf{Hyb}_{2\text{-}(\chi-1)\text{-}4'}$ *are equivalent except with probability* $1/q$.

Proof: The proof of Claim B.4 is similar to that of Claim B.3. We omit the details to avoid repetition. □

□

Lemma B.3: *For any stateful probabilistic adversary* $\mathcal{H}$*, there exists a probabilistic algorithm* $\mathcal{B}_{2\text{-}1}$*, whose running time is essentially the same as that of* $\mathcal{H}$*, such that for any security parameter* λ*,* $|\mathsf{Adv}^{(2\text{-}\chi\text{-}1)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}2)}_{\mathcal{H}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_{2\text{-}\chi\text{-}1}}(\lambda) + 1/q$ *for all* $\chi \in [q_{\text{KEY-PRE}}]$*, where* $\mathcal{B}_{2\text{-}\chi\text{-}1}(\cdot) = \mathcal{B}_{2\text{-}1}(\chi, \cdot)$ *for any* $\chi \in \mathbb{N}$.

Proof: In order to prove Lemma B.3, we construct below a probabilistic algorithm $\mathcal{B}_{2\text{-}1}$ against Problem 2 using a stateful probabilistic adversary $\mathcal{H}$ for distinguishing between $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ as a black-box sub-routine. Suppose $\mathcal{B}_{2\text{-}1}$ is given $\chi \in \mathbb{N}$ and an instance of Problem 2,

$$\varrho^{\mathsf{P2}}_{\widehat{\beta}} = (\mathsf{params}, \{\widetilde{\mathbb{B}}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{y}^{(0,\widehat{\beta})}, \boldsymbol{f}^{(0)}, \{\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})}, \boldsymbol{f}^{(\imath,\ell)}\}_{\imath\in[n'+n],\ell\in[2]}),$$

where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9, \ldots, 9}^{n'+n}));\\
&\widetilde{\delta}, \widetilde{\kappa}, \widetilde{\pi}, \widetilde{\omega}, \widetilde{\tau} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,3)}, \ldots, \boldsymbol{b}^{(0,6)}\};\\
&\boldsymbol{y}^{(0,0)} = (\widetilde{\delta}, 0, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0}, \boldsymbol{y}^{(0,1)} = (\widetilde{\delta}, \widetilde{\pi}, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}^*_0};\\
&\boldsymbol{f}^{(0)} = (\widetilde{\omega}, \widetilde{\tau}, 0, 0, 0, 0)_{\mathbb{B}_0};\\
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,5)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1, 0), \vec{e}^{(2)} = (0, 1) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
\boldsymbol{y}^{(\imath,\ell,0)} &= (\widetilde{\delta}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{y}^{(\imath,\ell,1)} &= (\widetilde{\delta}\vec{e}^{(\ell)}, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
\boldsymbol{f}^{(\imath,\ell)} &= (\widetilde{\omega}\vec{e}^{(\ell)}, \widetilde{\tau}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, 0)_{\mathbb{B}_\imath}
\end{aligned}\right\} \text{ for } \imath \in [n'+n], \ell \in [2].
\end{aligned}$$

$\mathcal{B}_{2\text{-}1}$ interacts with $\mathcal{H}$ as follows:

1. First, $\mathcal{B}_{2\text{-}1}$ provides the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}\}_{\imath\in[n'+n]})$, all of which are taken from the given Problem 2 instance.
2. For $h \in [q_{\text{KEY}}]$, in response to the h^{th} decryption key query of $\mathcal{H}$ for some function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, $\mathcal{B}_{2\text{-}1}$ proceeds as follows:
   (a) $(h < \chi)$ $\mathcal{B}_{2\text{-}1}$ gives $\mathcal{H}$ a decryption key $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of which are generated as in Eq. (3.7) using the bases $\{\mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ included within the given Problem 2 instance.

(b) $(h = \chi)$ $\mathcal{B}_{2\text{-}1}$ forms $\big((\{\breve{\sigma}_{\chi,j}\}_{j\in[n]}, \{\breve{\alpha}_{\chi,j'}, \breve{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]}), \rho_\chi : [m_\chi] \to [n']\big), \big((\{\bar{\sigma}_{\chi,j}\}_{j\in[n]}, \{\bar{\alpha}_{\chi,j'}, \bar{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]}), \rho_\chi : [m_\chi] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_\chi)$, samples $\breve{\zeta}_\chi, \bar{\zeta}_\chi, \{\widehat{\kappa}'_{\chi,j',1}, \widehat{\kappa}'_{\chi,j',2}\}_{j'\in[m_\chi]}, \{\widehat{\kappa}_{\chi,j,1}, \widehat{\kappa}_{\chi,j,2}\}_{j\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes

$$\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= \breve{\gamma}_{\chi,j'}\boldsymbol{y}^{(\rho_\chi(j'),1,\widehat{\beta})} + \breve{\alpha}_{\chi,j'}\boldsymbol{y}^{(\rho_\chi(j'),2,\widehat{\beta})} + \bar{\gamma}_{\chi,j'}\boldsymbol{b}^{*(\rho_\chi(j'),1)} + \\
&\quad \bar{\alpha}_{\chi,j'}\boldsymbol{b}^{*(\rho_\chi(j'),2)} + \widehat{\kappa}'_{\chi,j',1}\boldsymbol{b}^{*(\rho_\chi(j'),7)} + \widehat{\kappa}'_{\chi,j',2}\boldsymbol{b}^{(\rho_\chi(j'),8)} \text{ for } j' \in [m_\chi], \\
\boldsymbol{k}^{(\chi,j)} &= \breve{\sigma}_{\chi,j}\boldsymbol{y}^{(n'+j,1,\widehat{\beta})} + \breve{\zeta}_\chi\boldsymbol{y}^{(n'+j,2,\widehat{\beta})} + \bar{\sigma}_{\chi,j}\boldsymbol{b}^{*(n'+j,1)} + \bar{\zeta}_\chi\boldsymbol{b}^{*(n'+j,2)} + \\
&\quad \widehat{\kappa}_{\chi,j,1}\boldsymbol{b}^{*(n'+j,7)} + \widehat{\kappa}_{\chi,j,2}\boldsymbol{b}^{*(n'+j,8)} \text{ for } j \in [n],
\end{aligned}$$

and returns decryption key $\text{SK}(f_\chi) = (f_\chi, \{\boldsymbol{k}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\boldsymbol{k}^{(\chi,j)}\}_{j\in[n]})$ to $\mathcal{H}$.

(c) $(h > \chi)$ $\mathcal{B}_{2\text{-}1}$ gives $\mathcal{H}$ a decryption key $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of which are generated as in Eq. (3.2) using the bases $\{\mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ of the given $\mathsf{Problem\ 2}$ instance.

3. When $\mathcal{B}_{2\text{-}1}$ receives the ciphertext query from $\mathcal{H}$ for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{B}_{2\text{-}1}$ samples $\vec{s} \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n | R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{s})) = R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z}))\forall h \in [q_{\text{KEY-PRE}}]\}$, samples $\theta, \{\varphi'_{\iota'}\}_{\iota'\in[n']}, \{\varphi_\iota\}_{\iota\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= \boldsymbol{f}^{(\iota',1)} + x_{\iota'}\boldsymbol{f}^{(\iota',2)} + \theta\boldsymbol{b}^{(\iota',5)} + \theta x_{\iota'}\boldsymbol{b}^{(\iota',6)} + \varphi'_{\iota'}\boldsymbol{b}^{(\iota',9)} \text{ for } \iota' \in [n'], \\
\boldsymbol{c}^{(\iota)} &= \boldsymbol{f}^{(n'+\iota,1)} + z_\iota\boldsymbol{f}^{(n'+\iota,2)} + \theta\boldsymbol{b}^{(n'+\iota,5)} + \theta s_\iota\boldsymbol{b}^{(n'+\iota,6)} + \varphi_\iota\boldsymbol{b}^{(n'+\iota,9)} \text{ for } \iota \in [n],
\end{aligned}$$

and hands $\mathcal{H}$ the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$.

4. $\mathcal{H}$ eventually outputs a bit $\beta \in \{0,1\}$. $\mathcal{B}_{2\text{-}1}$ outputs $\widehat{\beta}' = \beta$ as its guess bit in its $\mathsf{Problem\ 2}$ challenge.

Observe that when $\widehat{\beta} = 0$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,0)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}$ for $\imath \in [n'+n], \ell \in [2]$, the components of the χth decryption key $\text{SK}(f_\chi)$ returned by $\mathcal{B}_{2\text{-}1}$ to $\mathcal{H}$ take the forms

$$\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= ((\widetilde{\delta}\breve{\gamma}_{\chi,j'} + \bar{\gamma}_{\chi,j'}, \widetilde{\delta}\breve{\alpha}_{\chi,j'} + \bar{\alpha}_{\chi,j'}), \vec{0}^2, \vec{0}^2, \\
&\quad (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2}), 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\
\boldsymbol{k}^{(\chi,j)} &= ((\widetilde{\delta}\breve{\sigma}_{\chi,j} + \bar{\sigma}_{\chi,j}, \widetilde{\delta}\breve{\zeta}_\chi + \bar{\zeta}_\chi), \vec{0}^2, \vec{0}^2, \\
&\quad (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\widetilde{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],
\end{aligned}$$

which coincides with those in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ (Eq. (3.2)), where we have $\alpha_{\chi,j'} = \widetilde{\delta}\breve{\alpha}_{\chi,j'} + \bar{\alpha}_{\chi,j'}, \gamma_{\chi,j'} = \widetilde{\delta}\breve{\gamma}_{\chi,j'} + \bar{\gamma}_{\chi,j'}$ for $j' \in [m_\chi]$, $\sigma_{\chi,j} = \widetilde{\delta}\breve{\sigma}_{\chi,j} + \bar{\sigma}_{\chi,j}$ for $j \in [n]$, $\zeta_\chi = \widetilde{\delta}\breve{\zeta}_\chi + \bar{\zeta}_\chi$, $\vec{\kappa}'^{(\chi,j')} = (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2})$ for $j' \in [m_\chi]$, and $\vec{\kappa}^{(\chi,j)} = (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2})$ for $j \in [n]$. On the other hand, in case $\widehat{\beta} = 1$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,1)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}$ for $\imath \in [n'+n], \ell \in [2]$, the components of the χth decryption key $\text{SK}(f_\chi)$ given by $\mathcal{B}_{2\text{-}1}$ to $\mathcal{H}$ take the forms

$$\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= ((\widetilde{\delta}\breve{\gamma}_{\chi,j'} + \bar{\gamma}_{\chi,j'}, \widetilde{\delta}\breve{\alpha}_{\chi,j'} + \bar{\alpha}_{\chi,j'}), (\widetilde{\pi}\breve{\gamma}_{\chi,j'}, \widetilde{\pi}\breve{\alpha}_{\chi,j'}), \vec{0}^2, \\
&\quad (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}), 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\
\boldsymbol{k}^{(\chi,j)} &= ((\widetilde{\delta}\breve{\sigma}_{\chi,j} + \bar{\sigma}_{\chi,j}, \widetilde{\delta}\breve{\zeta}_\chi + \bar{\zeta}_\chi), (\widetilde{\pi}\breve{\sigma}_{\chi,j}, \widetilde{\pi}\breve{\zeta}_\chi), \vec{0}^2, \\
&\quad (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],
\end{aligned}$$

which coincides with that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ (Eq. (3.5)), where we have $\alpha_{\chi,j'} = \widetilde{\delta}\breve{\alpha}_{\chi,j'} + \bar{\alpha}_{\chi,j'}, \gamma_{\chi,j'} = \widetilde{\delta}\breve{\gamma}_{\chi,j'} + \bar{\gamma}_{\chi,j'}$, $\widetilde{\alpha}_{\chi,j'} = \widetilde{\pi}\breve{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'} = \widetilde{\pi}\breve{\gamma}_{\chi,j'}$ for $j' \in [m_\chi]$, $\sigma_{\chi,j} = \widetilde{\delta}\breve{\sigma}_{\chi,j} + \bar{\sigma}_{\chi,j}$, $\widetilde{\sigma}_{\chi,j} = \widetilde{\pi}\breve{\sigma}_{\chi,j}$ for

$j \in [n]$, $\zeta_\chi = \widetilde{\delta}\breve{\zeta}_\chi + \bar{\zeta}_\chi$, $\widetilde{\zeta}_\chi = \widetilde{\pi}\breve{\zeta}_\chi$, $\vec{\kappa}'^{(\chi,j')} = (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2})$ for $j' \in [m_\chi]$, and $\vec{\kappa}^{(\chi,j)} = (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2})$ for $j \in [n]$.

Clearly, ζ_χ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$, and so is $\widetilde{\zeta}_\chi$ except when $\widetilde{\pi} = 0$. This is because $\bar{\zeta}_\chi$ and $\breve{\zeta}_\chi$ are sampled uniformly and independently (of the other variables) from $\mathbb{F}_q$. For similar reason, each of $\{\vec{\kappa}'^{(\chi,j')}\}_{j'\in[m_\chi]}$ and $\{\vec{\kappa}^{(\chi,j)}\}_{j\in[n]}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q^2$. In order to see that $(\{\sigma_{\chi,j}\}_{j\in[n]}, \{\alpha_{\chi,j'}, \gamma_{\chi,j'}\}_{j'\in[m_\chi]})$ and $(\{\widetilde{\sigma}_{\chi,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]})$ simulated by $\mathcal{B}_{2\text{-}1}$ have the desired distribution in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and $\mathrm{Hyb}_{2\text{-}\chi\text{-}2}$, i.e., their distribution coincides with the output distribution of $\mathsf{PGB}(f_\chi)$ with uniform and independent randomness, let $\vec{r}^{(\chi,1)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_\chi+n-1}$ and $\vec{r}^{(\chi,2)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_\chi+n-1}$ be the random vectors used while generating the constants $(\{\breve{\sigma}_{\chi,j}\}_{j\in[n]}, \{\breve{\alpha}_{\chi,j'}, \breve{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]})$ and $(\{\bar{\sigma}_{\chi,j}\}_{j\in[n]}, \{\bar{\alpha}_{\chi,j'}, \bar{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]})$ respectively. Then, by the linearity property of PGB (as described in Section 2.3), it follows that the constants $(\{\sigma_{\chi,j}\}_{j\in[n]}, \{\alpha_{\chi,j'}, \gamma_{\chi,j'}\}_{j'\in[m_\chi]})$ and $(\{\widetilde{\sigma}_{\chi,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]})$ respectively are the outputs of $\mathsf{PGB}(f_\chi)$ with randomness $\vec{r}^{(\chi)} = \widetilde{\delta}\vec{r}^{(\chi,1)} + \vec{r}^{(\chi,2)}$ and $\vec{r}'^{(\chi)} = \widetilde{\pi}\vec{r}^{(\chi,1)}$. Since, $\vec{r}^{(\chi,1)}$ and $\vec{r}^{(\chi,2)}$ are uniformly and independently sampled from $\mathbb{F}_q^{m_\chi+n-1}$, it follows that $\vec{r}^{(\chi)}$ is uniformly distributed in $\mathbb{F}_q^{m_\chi+n-1}$ and so is $\vec{r}'^{(\chi)}$ except when $\widetilde{\pi} = 0$, as well as they are independent of one another and of all the other variables. Thus, $(\{\sigma_{\chi,j}\}_{j\in[n]}, \{\alpha_{\chi,j'}, \gamma_{\chi,j'}\}_{j'\in[m_\chi]})$ and $(\{\widetilde{\sigma}_{\chi,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]})$ simulated by $\mathcal{B}_{2\text{-}1}$ have the desired distributions except when $\widetilde{\pi} = 0$.

Also, the components of the ciphertext CT returned to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}1}$ have the form

$$\boldsymbol{c}'^{(\iota')} = (\widetilde{\omega}(1, x_{\iota'}), \widetilde{\tau}(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],$$
$$\boldsymbol{c}^{(\iota)} = (\widetilde{\omega}(1, z_\iota), \widetilde{\tau}(1, z_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],$$

which coincides with those in Eq. (3.4) corresponding to both $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$, where $\omega = \widetilde{\omega}$ and $\tau = \widetilde{\tau}$ are uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$. Moreover, for all $h \in [q_{\text{KEY}}]\backslash\{\chi\}$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}_{h,j}\}_{j\in[n]}$ of the hth decryption key $\text{SK}(f_h)$ given to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}1}$ are generated the same as in Eq. (3.7), if $h < \chi$, and as in Eq. (3.2), if $h > \chi$, which are their proper forms in the respective cases both in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$. Finally, the public parameters MPK provided to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}1}$ are clearly distributed identically to those in both $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$ and $\mathrm{Hyb}_{2\text{-}\chi\text{-}2}$. Therefore, it follows that the view of $\mathcal{H}$ simulated by $\mathcal{B}_{2\text{-}1}$ given χ and a $\mathsf{Problem\ 2}$ instance $\varrho^{\mathsf{P2}}_{\widehat{\beta}}$ for $\widehat{\beta} \in \{0,1\}$, coincides with that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}1}$, if $\widehat{\beta} = 0$, while that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ except with probability $1/q$, if $\widehat{\beta} = 1$. Hence the lemma follows. □

Lemma B.4: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}\chi\text{-}2)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}3)}_{\mathcal{H}}(\lambda)| \leq 4/q$ for all $\chi \in [q_{\text{KEY-PRE}}]$.*

Proof: The proof of Lemma B.4 utilizes the following result:

Lemma B.5 (Lemma 3 in [OT10]): *For any $p \in \mathbb{F}_q$ and $d \in \mathbb{N}$, let $\mathbb{C}_p = \{(\vec{v}, \vec{w}) \in (\mathbb{F}_q^d \times \mathbb{F}_q^d)\backslash\{(\vec{0}^d, \vec{0}^d)\} | \vec{v} \cdot \vec{w} = p\}$. For all $(\vec{v}, \vec{w}) \in \mathbb{C}_p$, for all $(\vec{c}, \vec{k}) \in \mathbb{C}_p$, we have*

$$\mathsf{Pr}[\vec{v}\boldsymbol{F} = \vec{c} \bigwedge \vec{w}(\boldsymbol{F}^{-1})^{\intercal} = \vec{k}] = \mathsf{Pr}[\vec{v}(\boldsymbol{F}^{-1})^{\intercal} = \vec{c} \bigwedge \vec{w}\boldsymbol{F} = \vec{k}] = 1/\sharp\mathbb{C}_p,$$

where $\boldsymbol{F} \xleftarrow{U} GL(d, \mathbb{F}_q)$.

To prove Lemma B.4, we first define below the intermediate hybrid $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$. Next, we show the equivalence of the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ and that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ (Claim B.5), as well as those in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ and in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ (Claim B.6).

$\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ ($\chi \in [q_{\text{KEY-PRE}}]$): This experiment is the same as $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ except that in this experiment the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is generated as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ such that

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \boxed{\vec{a}'^{(\iota')}}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \boxed{\vec{a}^{(\iota)}}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{B.11}$$

while the χ^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is generated as $\text{SK}(f_\chi) = (f_\chi, \{\boldsymbol{k}'^{(\chi,j')}\}_{j' \in [m_\chi]}, \{\boldsymbol{k}^{(\chi,j)}\}_{j \in [n]})$ such that

$$\begin{aligned} \boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \boxed{\vec{u}'^{(\chi,j')}}, \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \boxed{\vec{u}^{(\chi,j)}}, \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{B.12}$$

where $(\{\vec{a}'^{(\iota')}\}_{\iota' \in [n']}, \{\vec{a}^{(\iota)}\}_{\iota \in [n]}, \{\vec{u}'^{(\chi,j')}\}_{j' \in [m_\chi]}, \{\vec{u}^{(\chi,j)}\}_{j \in [n]}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^2)^{n'} \times (\mathbb{F}_q^2)^n \times (\mathbb{F}_q^2)^{m_\chi} \times (\mathbb{F}_q^2)^n$, if $R^{\text{ABP}\circ\text{IP}}(f_\chi, (\vec{x}, \vec{z})) = 0$, whereas $(\{\vec{a}'^{(\iota')}\}_{\iota' \in [n']}, \{\vec{a}^{(\iota)}\}_{\iota \in [n]}, \{\vec{u}'^{(\chi,j')}\}_{j' \in [m_\chi]}, \{\vec{u}^{(\chi,j)}\}_{j \in [n]}) \xleftarrow{\mathsf{U}} (\mathbb{F}_q^2)^{n'} \times (\mathbb{F}_q^2)^n \times (\mathbb{F}_q^2)^{m_\chi} \times (\mathbb{F}_q^2)^n$ subject to the restriction that $\sum\limits_{j' \in [m_\chi]} \Omega'_{\chi,j'}(\vec{a}'^{(\rho_\chi(j'))} \cdot \vec{u}'^{(\chi,j')}) + \sum\limits_{j \in [n]} \Omega_{\chi,j}(\vec{a}^{(j)} \cdot \vec{u}^{(\chi,j)}) = 0$, if $R^{\text{ABP}\circ\text{IP}}(f_\chi, (\vec{x}, \vec{z})) = 1$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$. Here, $(\{\Omega_{\chi,j}\}_{j \in [n]}, \{\Omega'_{\chi,j'}\}_{j' \in [m_\chi]}) = \mathsf{REC}(f_\chi, \vec{x})$.

Claim B.5: *The distribution* $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ *in* $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ *and that in* $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ *are equivalent except with probability* $2/q$.

Proof: Consider the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ from the original set of bases $\{\mathbb{B}_\iota, \mathbb{B}_\iota^*\}_{\iota \in [n'+n]}$ used in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ as follows: Generate matrices $\{\boldsymbol{Z}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{Z}^{(\iota)}\}_{\iota \in [n]} \xleftarrow{\mathsf{U}} \mathsf{GL}(2, \mathbb{F}_q)$, define $\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^\intercal$ for $\iota' \in [n']$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^\intercal$ for $\iota \in [n]$, compute the vectors

$$\begin{pmatrix} \boldsymbol{d}^{(\iota',3)} \\ \boldsymbol{d}^{(\iota',4)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{-1} \begin{pmatrix} \boldsymbol{b}^{(\iota',3)} \\ \boldsymbol{b}^{(\iota',4)} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{d}^{*(\iota',3)} \\ \boldsymbol{d}^{*(\iota',4)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^\intercal \begin{pmatrix} \boldsymbol{b}^{*(\iota',3)} \\ \boldsymbol{b}^{*(\iota',4)} \end{pmatrix} \text{ for } \iota' \in [n'],$$

$$\begin{pmatrix} \boldsymbol{d}^{(n'+\iota,3)} \\ \boldsymbol{d}^{(n'+\iota,4)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{-1} \begin{pmatrix} \boldsymbol{b}^{(n'+\iota,3)} \\ \boldsymbol{b}^{(n'+\iota,4)} \end{pmatrix}, \begin{pmatrix} \boldsymbol{d}^{*(n'+\iota,3)} \\ \boldsymbol{d}^{*(n'+\iota,4)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^\intercal \begin{pmatrix} \boldsymbol{b}^{*(n'+\iota,3)} \\ \boldsymbol{b}^{*(n'+\iota,4)} \end{pmatrix} \text{ for } \iota \in [n],$$

and set

$$\left.\begin{aligned} \mathbb{D}_{\iota'} &= \{\boldsymbol{b}^{(\iota',1)}, \boldsymbol{b}^{(\iota',2)}, \boldsymbol{d}^{(\iota',3)}, \boldsymbol{d}^{(\iota',4)}, \boldsymbol{b}^{(\iota',5)}, \ldots, \boldsymbol{b}^{(\iota',9)}\} \\ \mathbb{D}^*_{\iota'} &= \{\boldsymbol{b}^{*(\iota',1)}, \boldsymbol{b}^{*(\iota',2)}, \boldsymbol{d}^{*(\iota',3)}, \boldsymbol{d}^{*(\iota',4)}, \boldsymbol{b}^{*(\iota',5)}, \ldots, \boldsymbol{b}^{*(\iota',9)}\} \end{aligned}\right\} \text{ for } \iota' \in [n'],$$
$$\left.\begin{aligned} \mathbb{D}_{n'+\iota} &= \{\boldsymbol{b}^{(n'+\iota,1)}, \boldsymbol{b}^{(n'+\iota,2)}, \boldsymbol{d}^{(n'+\iota,3)}, \boldsymbol{d}^{(n'+\iota,4)}, \boldsymbol{b}^{(n'+\iota,5)}, \ldots, \boldsymbol{b}^{(n'+\iota,9)}\} \\ \mathbb{D}^*_{n'+\iota} &= \{\boldsymbol{b}^{*(n'+\iota,1)}, \boldsymbol{b}^{*(n'+\iota,2)}, \boldsymbol{d}^{*(n'+\iota,3)}, \boldsymbol{d}^{*(n'+\iota,4)}, \boldsymbol{b}^{*(n'+\iota,5)}, \ldots, \boldsymbol{b}^{*(n'+\iota,9)}\} \end{aligned}\right\} \text{ for } \iota \in [n]. \tag{B.13}$$

It can be readily observed that the new set of bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ are indeed dual orthonormal, and are distributed the same as the original set of bases $\{\mathbb{B}_\iota, \mathbb{B}_\iota^*\}_{\iota \in [n'+n]}$.

Observe that the components of the χ^{th} decryption key $\text{SK}(f_\chi)$ returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ can be expressed in terms of the bases $\{\mathbb{D}_\iota, \mathbb{D}_\iota^*\}_{\iota \in [n'+n]}$ as

$$\begin{aligned} \boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), (\widetilde{\gamma}_{\chi,j'}, \widetilde{\alpha}_{\chi,j'}), \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \\ &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{u}'^{(\chi,j')}, \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{D}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), (\widetilde{\sigma}_{\chi,j}, \widetilde{\zeta}_\chi), \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ &= ((\sigma_{\chi,j}, \zeta_\chi), \vec{u}^{(\chi,j)}, \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{B.14}$$

where $\vec{u}'^{(\chi,j')} = (\widetilde{\gamma}_{\chi,j'}, \widetilde{\alpha}_{\chi,j'})\boldsymbol{U}'^{(\rho_\chi(j'))}$ for $j' \in [m_\chi]$, $\vec{u}^{(\chi,j)} = (\widetilde{\sigma}_{\chi,j}, \widetilde{\zeta}_\chi)\boldsymbol{U}^{(j)}$ for $j \in [n]$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$. Also, note that the components of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ can be expressed in terms of the bases $\{\mathbb{D}_\iota, \mathbb{D}^*_\iota\}_{\iota\in[n'+n]}$ as

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= (\omega(1,x_{\iota'}), \tau(1,x_{\iota'}), \theta(1,x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \\
&= (\omega(1,x_{\iota'}), \vec{a}'^{(\iota')}, \theta(1,x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota' \in [n'], \\
\boldsymbol{c}^{(\iota)} &= (\omega(1,z_\iota), \tau(1,z_\iota), \theta(1,s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \\
&= (\omega(1,z_\iota), \vec{a}^{(\iota)}, \theta(1,s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota \in [n],
\end{aligned} \tag{B.15}$$

where $\vec{a}'^{(\iota')} = \tau(1,x_{\iota'})\boldsymbol{Z}'^{(\iota')}$ for $\iota' \in [n']$, $\vec{a}^{(\iota)} = \tau(1,z_\iota)\boldsymbol{Z}^{(\iota)}$ for $\iota \in [n]$, and all the other variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$.

We argue that $(\{\vec{a}'^{(\iota')}\}_{\iota'\in[n']}, \{\vec{a}^{(\iota)}\}_{\iota\in[n]}, \{\vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\vec{u}^{(\chi,j)}\}_{j\in[n]}) \in (\mathbb{F}_q^2)^{n'} \times (\mathbb{F}_q^2)^n \times (\mathbb{F}_q^2)^{m_\chi} \times (\mathbb{F}_q^2)^n$ are jointly distributed as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ in the view of $\mathcal{H}$. Let us observe the following two facts:

(A) From Lemma B.5 it follows that for each $j' \in [m_\chi]$, the pair of vectors $(\vec{a}'^{(\rho_\chi(j'))}, \vec{u}'^{(\chi,j')})$ is uniformly distributed in $\mathbb{C}_{\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')}}$, and similarly, for each $j \in [n]$, the pair of vectors $(\vec{a}^{(j)}, \vec{u}^{(\chi,j)})$ is uniformly distributed in $\mathbb{C}_{\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)}}$.

(B) Observe that

$$\begin{pmatrix} \vec{a}'^{(\rho_\chi(1))} \cdot \vec{u}'^{(\chi,1)} \\ \vdots \\ \vec{a}'^{(\rho_\chi(m_\chi))} \cdot \vec{u}'^{(\chi,m_\chi)} \\ \vec{a}^{(1)} \cdot \vec{u}^{(\chi,1)} \\ \vdots \\ \vec{a}^{(n)} \cdot \vec{u}^{(\chi,n)} \end{pmatrix} = \begin{pmatrix} \tau(\widetilde{\alpha}_{\chi,1}x_{\rho_\chi(1)} + \widetilde{\gamma}_{\chi,1}) \\ \vdots \\ \tau(\widetilde{\alpha}_{\chi,m_\chi}x_{\rho_\chi(m_\chi)} + \widetilde{\gamma}_{\chi,m_\chi}) \\ \tau(\widetilde{\zeta}_\chi z_1 + \widetilde{\sigma}_{\chi,1}) \\ \vdots \\ \tau(\widetilde{\zeta}_\chi z_n + \widetilde{\sigma}_{\chi,n}) \end{pmatrix} = \tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})\begin{pmatrix} (\vec{r}'^{(\chi)})^\intercal \\ \widetilde{\zeta}_\chi \end{pmatrix}, \tag{B.16}$$

where $\boldsymbol{L}^{(\chi)} \in \mathbb{F}_q^{(m_\chi+n)\times(m_\chi+n)}$ is the matrix representation of the ABP Γ'_χ computing f_χ, as computed by PGB described in Section 2.3, and $\vec{r}'^{(\chi)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_\chi+n-1}$ is the random vector used by PGB when run on input f_χ to generate the constants $(\{\widetilde{\sigma}_{\chi,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]}) \in \mathbb{F}_q^n \times (\mathbb{F}_q^2)^{m_\chi}$. Now, we have the following two possibilities:

(I) $(R^{\text{ABP}\circ\text{IP}}(f_\chi, (\vec{x},\vec{z})) = 0$, i.e., $f_\chi(\vec{x},\vec{z}) \neq 0)$ From Lemma 2.1, we have $\det(\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) = f_\chi(\vec{x},\vec{z})$. Hence, $\det(\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) = \tau^{m_\chi+n}\det(\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) = \tau^{m_\chi+n}f_\chi(\vec{x},\vec{z})$. As $f_\chi(\vec{x},\vec{z}) \neq 0$ in this case, it follows that in this case $\det(\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) \neq 0$, or in other words, the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is invertible with all but negligible probability $1/q$, i.e., except when $\tau = 0$. Therefore, the image of the linear transformation defined by the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is $\mathbb{F}_q^{m_\chi+n}$.

Hence, it follows from Eq. (B.16) that since $(\vec{r}'^{(\chi)}, \widetilde{\zeta}_\chi)$ are sampled uniformly from $\mathbb{F}_q^{m_\chi+n-1} \times \mathbb{F}_q$, the variables $(\{\vec{a}'^{(\rho_\chi(j'))} \cdot \vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\vec{a}^{(j)} \cdot \vec{u}^{(\chi,j)}\}_{j\in[n]})$ are jointly distributed uniformly in $\mathbb{F}_q^{m_\chi} \times \mathbb{F}_q^n$ as well. Also, since fresh variables $(\vec{r}'^{(\chi)}, \widetilde{\zeta}_\chi)$ appear only in $(\{\vec{a}'^{\rho_\chi(j')} \cdot \vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\vec{a}^{(j)} \cdot \vec{u}^{(\chi,j)}\}_{j\in[n]})$, the variables $(\{\vec{a}'^{(\rho_\chi(j')} \cdot \vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\vec{a}^{(j)} \cdot \vec{u}^{(\chi,j)}\}_{j\in[n]})$ are also fresh, i.e., independent of all the other variables in the view of $\mathcal{H}$.

(II) $(R^{\text{ABP}\circ\text{IP}}(f_\chi, (\vec{x},\vec{z})) = 1$, i.e., $f_\chi(\vec{x},\vec{z}) = 0)$ Here, the rank of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is $m_\chi + n - 1$ with all but negligible probability $1/q$, i.e., except when $\tau = 0$. In order to see this, first notice that since $f_\chi(\vec{x},\vec{z}) = 0$ in this case, again from Lemma 2.1 it follows that $\det(\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) = \tau^{m_\chi+n}\det(\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})) = \tau^{m_\chi+n}f_\chi(\vec{x},\vec{z}) = 0$. Hence, the rank of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is at most $m_\chi + n - 1$. Also, note that according to Lemma 2.1, the matrix $\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ contains only -1's in the second diagonal, and 0's below the second diagonal, and hence, the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ possesses only $-\tau$'s in the second diagonal, and 0's below the second diagonal. Therefore, the $(m_\chi+n-1)\times(m_\chi+n-1)$ sub-matrix of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ obtained by removing its first row and last column clearly has full rank $m_\chi+n-1$ except when $\tau=0$, as this sub-matrix is in upper-triangular form with $-\tau$ as its diagonal entries. Since the rank of a matrix cannot be smaller than the rank of any of its sub-matrix, it follows that the rank of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ cannot be smaller than $m_\chi+n-1$. Consequently, it follows that the rank of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ must be $m_\chi+n-1$ in this case.

Also, in this case, it holds that $((\Omega'_{\chi,j'})_{j'\in[m_\chi]},(\Omega_{\chi,j})_{j\in[n]})\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})=\vec{0}^{m_\chi+n}$, where $(\{\Omega_{\chi,j}\}_{j\in[n]},\{\Omega'_{\chi,j'}\}_{j'\in[m_\chi]})=\mathsf{REC}(f_\chi,\vec{x})$. In order to see this, recall from the description of the algorithm REC presented in Section 2.3 that $((\Omega'_{\chi,j'})_{j'\in[m_\chi]},(\Omega_{\chi,j})_{j\in[n]})$ are the cofactors of the entries of the last column of the matrix $\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ in the order from top to bottom. Hence, if we add up the products of the entries in the last column of the matrix $\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ with the corresponding cofactors in $(\{\Omega'_{\chi,j'}\}_{j'\in[m_\chi]},\{\Omega_{\chi,j}\}_{j\in[n]})$, we obtain $\det(\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}))$ which, as already noted above, is 0 in this case. Also, it is a standard result in linear algebra that the sum of the products of the cofactors of any column of a matrix with the corresponding entries of another column of that matrix is 0. Therefore, it follows that $((\Omega'_{\chi,j'})_{j'\in[m_\chi]},(\Omega_{\chi,j})_{j\in[n]})\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})=\vec{0}^{m_\chi+n}$. Since, each entry of the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is merely τ times the corresponding entry of the matrix $\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$, the same holds for the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ as well.

From the above two observations, it follows that the image of the linear transformation defined by the matrix $\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$ is given by the $(m_\chi+n-1)$-dimensional subspace $\mathsf{Im}(\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}))=\{\vec{v}=(v_1,\ldots,v_{m_\chi+n})\in\mathbb{F}_q^{m_\chi+n}|((\Omega'_{\chi,j'})_{j'\in[m_\chi]},(\Omega_{\chi,j})_{j\in[n]})\cdot\vec{v}=\sum\limits_{j'\in[m_\chi]}\Omega'_{\chi,j'}v_{j'}+\sum\limits_{j\in[n]}\Omega_{\chi,j}v_{n'+j}=0\}\subset\mathbb{F}_q^{m_\chi+n}$.

Hence, it follows from Eq. (B.16) that since $(\vec{r}'^{(\chi)},\widetilde{\zeta}_\chi)$ are sampled uniformly from $\mathbb{F}_q^{m_\chi+n-1}\times\mathbb{F}_q$, the variables $(\{\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]},\{\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)}\}_{j\in[n]})$ are jointly distributed uniformly in $\mathsf{Im}(\tau\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}))$, i.e., those variables are jointly distributed uniformly in $\mathbb{F}_q^{m_\chi}\times\mathbb{F}_q^n$ subject to the relation $\sum\limits_{j'\in[m_\chi]}\Omega'_{\chi,j'}(\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')})+\sum\limits_{j\in[n]}\Omega_{\chi,j}(\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)})=0$. Also, just as in Case (I), since fresh variables $(\vec{r}'^{(\chi)},\widetilde{\zeta}_\chi)$ appear only in $(\{\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]},\{\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)}\}_{j\in[n]})$, the variables $(\{\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]},\{\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)}\}_{j\in[n]})$ are also fresh, i.e., independent of all the other variables in the view of $\mathcal{H}$ in this case as well.

From (A) and (B) above, it follows that the variables $(\{\vec{a}'^{(\iota')}\}_{\iota'\in[n']},\{\vec{a}^{(\iota)}\}_{\iota\in[n]},\{\vec{u}'^{(\chi,j')}\}_{j'\in[m_\chi]},\{\vec{u}^{(\chi,j)}\}_{j\in[n]})$ are jointly distributed uniformly in $(\mathbb{F}_q^2)^{n'}\times(\mathbb{F}_q^2)^n\times(\mathbb{F}_q^2)^{m_\chi}\times(\mathbb{F}_q^2)^n$, if $R^{\text{ABP}\circ\text{IP}}(f_\chi,(\vec{x},\vec{z}))=0$, where as these variables are jointly distributed uniformly in $(\mathbb{F}_q^2)^{n'}\times(\mathbb{F}_q^2)^n\times(\mathbb{F}_q^2)^{m_\chi}\times(\mathbb{F}_q^2)^n$ subject to the relation $\sum\limits_{j'\in[m_\chi]}\Omega'_{\chi,j'}(\vec{a}'^{(\rho_\chi(j'))}\cdot\vec{u}'^{(\chi,j')})+\sum\limits_{j\in[n]}\Omega_{\chi,j}(\vec{a}^{(j)}\cdot\vec{u}^{(\chi,j)})=0$, in case $R^{\text{ABP}\circ\text{IP}}(f_\chi,(\vec{x},\vec{z}))=1$, as well as these variables are independent of all the other variables in the view of $\mathcal{H}$.

Moreover, for $h\in[q_{\text{KEY}}]\backslash\{\chi\}$, the components of the hth decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ can be expressed in terms of $\{\mathbb{D}_\iota,\mathbb{D}_\iota^*\}_{\iota\in[n'+n]}$ as

(a) $(h < \chi)$

$$\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{0}^2, (\widehat{\gamma}_{\chi,j'}, \widehat{\alpha}_{\chi,j'}), \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \\
&= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{0}^2, (\widehat{\gamma}_{\chi,j'}, \widehat{\alpha}_{\chi,j'}), \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{D}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\
\boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \vec{0}^2, (\widehat{\sigma}_{\chi,j}, \widehat{\zeta}_\chi), \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\
&= ((\sigma_{\chi,j}, \zeta_\chi), \vec{0}^2, (\widehat{\sigma}_{\chi,j}, \widehat{\zeta}_\chi), \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],
\end{aligned} \tag{B.17}$$

(b) $(h > \chi)$

$$\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{0}^2, \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \\
&= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{0}^2, \vec{0}^2, \vec{\kappa}'^{(\chi,j')}, 0)_{\mathbb{D}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi], \\
\boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \vec{0}^2, \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\
&= ((\sigma_{\chi,j}, \zeta_\chi), \vec{0}^2, \vec{0}^2, \vec{\kappa}^{(\chi,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],
\end{aligned} \tag{B.18}$$

where all the variables are generated as in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$.

Now, note that in the view of $\mathcal{H}$, both the original sets of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ are consistent with the public parameters MPK. Moreover, for $h \in [q_{\text{KEY}}]\backslash\{\chi\}$, the components of the hth decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their forms as in Eq. (3.7), if $h < \chi$, and as in Eq. (3.2), if $h > \chi$, which are their proper forms in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$, under the basis transformation. Finally, since the RHS of Eq. (B.14) (respectively Eq. (B.15)) and that of Eq. (B.12) (Eq. (B.11)) have the same form except with probability $2/q$, the components of the χth decryption key $\text{SK}(f_\chi)$ and those of the ciphertext CT returned to $\mathcal{H}$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ can be conceptually changed to those in $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ except with probability $2/q$. Hence the claim follows. □

Claim B.6: *The distribution* $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ *in* $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ *and that in* $\mathsf{Hyb}_{2\text{-}\chi\text{-}2'}$ *are equivalent except with probability* $2/q$.

Proof: The proof of Claim B.6 is similar to that of Claim B.5. We omit the details to avoid repetition. □

□

Lemma B.6: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_{2\text{-}2}$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}\chi\text{-}3)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(2\text{-}\chi\text{-}4)}_{\mathcal{H}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}_{2\text{-}\chi\text{-}2}}(\lambda) + 2/q$ for all $\chi \in [q_{\text{KEY-PRE}}]$, where $\mathcal{B}_{2\text{-}\chi\text{-}2}(\cdot) = \mathcal{B}_{2\text{-}2}(\chi, \cdot)$ for any $\chi \in \mathbb{N}$.*

Proof: In order to prove Lemma B.5, we construct below a probabilistic algorithm $\mathcal{B}_{2\text{-}2}$ against Problem 3 using a stateful probabilistic adversary $\mathcal{H}$ for distinguishing between $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ as a black-box sub-routine. Suppose $\mathcal{B}_{2\text{-}2}$ is given $\chi \in \mathbb{N}$ and an instance of Problem 3,

$$\varrho^{\mathsf{P3}}_{\widehat{\beta}} = (\mathsf{params}, \{\widetilde{\mathbb{B}}_\imath, \widetilde{\mathbb{B}}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{y}^{(0,\widehat{\beta})}, \boldsymbol{f}^{(0)}, \{\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})}, \boldsymbol{f}^{(\imath,\ell)}\}_{\imath\in[n'+n],\ell\in[2]}),$$

where

$$
\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\rm OB}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\pi}, \widetilde{\pi}', \widetilde{\kappa}, \widetilde{\tau}, \widetilde{\theta} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,4)}, \ldots, \boldsymbol{b}^{(0,6)}\}, \widetilde{\mathbb{B}}_0^* = \{\boldsymbol{b}^{*(0,1)}, \boldsymbol{b}^{*(0,3)}, \ldots, \boldsymbol{b}^{*(0,6)}\};\\
&\boldsymbol{y}^{(0,0)} = (0, \widetilde{\pi}, 0, 0, \widetilde{\kappa}, 0)_{\mathbb{B}_0^*}, \boldsymbol{y}^{(0,1)} = (0, 0, \widetilde{\pi}', 0, \widetilde{\kappa}, 0)_{\mathbb{B}_0^*};\\
&\boldsymbol{f}^{(0)} = (0, \widetilde{\tau}, \widetilde{\theta}, 0, 0, 0)_{\mathbb{B}_0};\\
&\left.\begin{aligned}
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,7)}, \ldots, \boldsymbol{b}^{(\imath,9)}\}\\
&\widetilde{\mathbb{B}}_\imath^* = \{\boldsymbol{b}^{*(\imath,1)}, \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,5)}, \ldots, \boldsymbol{b}^{*(\imath,9)}\}
\end{aligned}\right\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0), \vec{e}^{(2)} = (0,1) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
&\boldsymbol{y}^{(\imath,\ell,0)} = (\vec{0}^2, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}\\
&\boldsymbol{y}^{(\imath,\ell,1)} = (\vec{0}^2, \vec{0}^2, \widetilde{\pi}'\vec{e}^{(\ell)}, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}\\
&\boldsymbol{f}^{(\imath,\ell)} = (\vec{0}^2, \widetilde{\tau}\vec{e}^{(\ell)}, \widetilde{\theta}\vec{e}^{(\ell)}, \vec{0}^2, 0)_{\mathbb{B}_\imath}
\end{aligned}\right\} \text{ for } \imath \in [n'+n], \ell \in [2].
\end{aligned}
$$

$\mathcal{B}_{2\text{-}2}$ interacts with $\mathcal{H}$ as follows:

1. First, $\mathcal{B}_{2\text{-}2}$ provides the public parameters MPK $= (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}\}_{\imath\in[n'+n]})$, all of which are taken from the given Problem 3 instance.
2. For $h \in [q_{\rm KEY}]$, in response to the $h^{\rm th}$ decryption key query of $\mathcal{H}$ for some function $f_h \in \mathcal{F}^{(q,n',n)}_{\rm ABP\circ IP}$, $\mathcal{B}_{2\text{-}2}$ proceeds as follows:
   (a) $(h < \chi)$ $\mathcal{B}_{2\text{-}2}$ gives $\mathcal{H}$ a decryption key SK$(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of which are generated as in Eq. (3.7) using the partial bases $\{\widetilde{\mathbb{B}}_\imath^*\}_{\imath\in[n'+n]}$ included within the given Problem 3 instance.
   (b) $(h = \chi)$ $\mathcal{B}_{2\text{-}2}$ forms $((\{\sigma_{\chi,j}\}_{j\in[n]}, \{\alpha_{\chi,j'}, \gamma_{\chi,j'}\}_{j'\in[m_\chi]}), \rho_\chi : [m_\chi] \to [n']), ((\{\breve{\sigma}_{\chi,j}\}_{j\in[n]}, \{\breve{\alpha}_{\chi,j'}, \breve{\gamma}_{\chi,j'}\}_{j'\in[m_\chi]}), \rho_\chi : [m_\chi] \to [n']) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_\chi)$, samples $\zeta_\chi, \breve{\zeta}_\chi, \{\widehat{\kappa}'_{\chi,j',1}, \widehat{\kappa}'_{\chi,j',2}\}_{j'\in[m_\chi]}, \{\widehat{\kappa}_{\chi,j,1}, \widehat{\kappa}_{\chi,j,2}\}_{j\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes
$$
\begin{aligned}
\boldsymbol{k}'^{(\chi,j')} &= \breve{\gamma}_{\chi,j'}\boldsymbol{y}^{(\rho_\chi(j'),1,\widehat{\beta})} + \breve{\alpha}_{\chi,j'}\boldsymbol{y}^{(\rho_\chi(j'),2,\widehat{\beta})} + \gamma_{\chi,j'}\boldsymbol{b}^{*(\rho_\chi(j'),1)} +\\
&\quad \alpha_{\chi,j'}\boldsymbol{b}^{*(\rho_\chi(j'),2)} + \widehat{\kappa}'_{\chi,j',1}\boldsymbol{b}^{*(\rho_\chi(j'),7)} + \widehat{\kappa}'_{\chi,j',2}\boldsymbol{b}^{(\rho_\chi(j'),8)} \text{ for } j' \in [m_\chi],\\
\boldsymbol{k}^{(\chi,j)} &= \breve{\sigma}_{\chi,j}\boldsymbol{y}^{(n'+j,1,\widehat{\beta})} + \breve{\zeta}_\chi\boldsymbol{y}^{(n'+j,2,\widehat{\beta})} + \sigma_{\chi,j}\boldsymbol{b}^{*(n'+j,1)} + \zeta_\chi\boldsymbol{b}^{*(n'+j,2)} +\\
&\quad \widehat{\kappa}_{\chi,j,1}\boldsymbol{b}^{*(n'+j,7)} + \widehat{\kappa}_{\chi,j,2}\boldsymbol{b}^{*(n'+j,8)} \text{ for } j \in [n],
\end{aligned}
$$
   and returns the decryption key SK$(f_\chi) = (f_\chi, \{\boldsymbol{k}'^{(\chi,j')}\}_{j'\in[m_\chi]}, \{\boldsymbol{k}^{(\chi,j)}\}_{j\in[n]})$ to $\mathcal{H}$.
   (c) $(h > \chi)$ $\mathcal{B}_{2\text{-}2}$ gives $\mathcal{H}$ a decryption key SK$(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of which are generated as in Eq. (3.2) using the partial bases $\{\widetilde{\mathbb{B}}_\imath^*\}_{\imath\in[n'+n]}$ of the given Problem 3 instance.
3. When $\mathcal{B}_{2\text{-}2}$ receives the ciphertext query from $\mathcal{H}$ for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{B}_{2\text{-}2}$ samples $\vec{s} \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n | R^{\rm ABP\circ IP}(f_h, (\vec{x}, \vec{s})) = R^{\rm ABP\circ IP}(f_h, (\vec{x}, \vec{z})) \forall h \in [q_{\rm KEY\text{-}PRE}]\}$, samples $\omega, \{\varphi'_{\iota'}\}_{\iota'\in[n']}, \{\varphi_\iota\}_{\iota\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes
$$
\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= \boldsymbol{f}^{(\iota',1)} + x_{\iota'}\boldsymbol{f}^{(\iota',2)} + \omega\boldsymbol{b}^{(\iota',1)} + \omega x_{\iota'}\boldsymbol{b}^{(\iota',2)} + \varphi'_{\iota'}\boldsymbol{b}^{(\iota',9)} \text{ for } \iota' \in [n'],\\
\boldsymbol{c}^{(\iota)} &= \boldsymbol{f}^{(n'+\iota,1)} + s_\iota\boldsymbol{f}^{(n'+\iota,2)} + \omega\boldsymbol{b}^{(n'+\iota,1)} + \omega z_\iota\boldsymbol{b}^{(n'+\iota,2)} + \varphi_\iota\boldsymbol{b}^{(n'+\iota,9)} \text{ for } \iota \in [n],
\end{aligned}
$$
   and hands $\mathcal{H}$ the ciphertext CT $= (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$.

4. $\mathcal{H}$ eventually outputs a bit $\beta \in \{0,1\}$. $\mathcal{B}_{2\text{-}2}$ outputs $\widehat{\beta}' = \beta$ as its guess bit in its $\mathsf{Problem\ 3}$ challenge.

Observe that when $\widehat{\beta} = 0$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,0)} = (\vec{0}^2, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n], \ell \in [2]$, the components of the χth decryption key $\text{SK}(f_\chi)$ returned by $\mathcal{B}_{2\text{-}2}$ to $\mathcal{H}$ take the form

$$\begin{aligned}\boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), (\widetilde{\pi}\breve{\gamma}_{\chi,j'}, \widetilde{\pi}\breve{\alpha}_{\chi,j'}), \vec{0}^2,\\ &\quad (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2}), 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi],\\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), (\widetilde{\pi}\breve{\sigma}_{\chi,j}, \widetilde{\pi}\breve{\zeta}_\chi), \vec{0}^2,\\ &\quad (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],\end{aligned}$$

which coincides with those in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ (Eq. (3.5)), where we have $\widetilde{\alpha}_{\chi,j'} = \widetilde{\pi}\breve{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'} = \widetilde{\pi}\breve{\gamma}_{\chi,j'}$ for $j' \in [m_\chi]$, $\widetilde{\sigma}_{\chi,j} = \widetilde{\pi}\breve{\sigma}_{\chi,j}$ for $j \in [n]$, $\widetilde{\zeta}_\chi = \widetilde{\pi}\breve{\zeta}_\chi$, $\vec{\kappa}'^{(\chi,j')} = (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2})$ for $j' \in [m_\chi]$, and $\vec{\kappa}^{(\chi,j)} = (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2})$ for $j \in [n]$. On the other hand, in case $\widehat{\beta} = 1$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,1)} = (\vec{0}^2, \vec{0}^2, \widetilde{\pi}'\vec{e}^{(\ell)}, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n], \ell \in [2]$, the components of the χth decryption key $\text{SK}(f_\chi)$ given by $\mathcal{B}_{2\text{-}2}$ to $\mathcal{H}$ take the form

$$\begin{aligned}\boldsymbol{k}'^{(\chi,j')} &= ((\gamma_{\chi,j'}, \alpha_{\chi,j'}), \vec{0}^2, (\widetilde{\pi}'\breve{\gamma}_{\chi,j'}, \widetilde{\pi}'\breve{\alpha}_{\chi,j'}),\\ &\quad (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2}), 0)_{\mathbb{B}^*_{\rho_\chi(j')}} \text{ for } j' \in [m_\chi],\\ \boldsymbol{k}^{(\chi,j)} &= ((\sigma_{\chi,j}, \zeta_\chi), \vec{0}^2, (\widetilde{\pi}'\breve{\sigma}_{\chi,j}, \widetilde{\pi}'\breve{\zeta}_\chi),\\ &\quad (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],\end{aligned}$$

which coincides with that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ (Eq. (3.7)), where we have $\widehat{\alpha}_{\chi,j'} = \widetilde{\pi}'\breve{\alpha}_{\chi,j'}$, $\widehat{\gamma}_{\chi,j'} = \widetilde{\pi}'\breve{\gamma}_{\chi,j'}$ for $j' \in [m_\chi]$, $\widehat{\sigma}_{\chi,j} = \widetilde{\pi}'\breve{\sigma}_{\chi,j}$ for $j \in [n]$, $\widehat{\zeta}_\chi = \widetilde{\pi}'\breve{\zeta}_\chi$, $\vec{\kappa}'^{(\chi,j')} = (\widetilde{\kappa}\breve{\gamma}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',1}, \widetilde{\kappa}\breve{\alpha}_{\chi,j'} + \widehat{\kappa}'_{\chi,j',2})$ for $j' \in [m_\chi]$, and $\vec{\kappa}^{(\chi,j)} = (\widetilde{\kappa}\breve{\sigma}_{\chi,j} + \widehat{\kappa}_{\chi,j,1}, \widetilde{\kappa}\breve{\zeta}_\chi + \widehat{\kappa}_{\chi,j,2})$ for $j \in [n]$.

Clearly, $\widetilde{\zeta}_\chi$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ except when $\widetilde{\pi} = 0$, and so is $\widehat{\zeta}_\chi$ except when $\widetilde{\pi}' = 0$. This is because $\breve{\zeta}_\chi$ is sampled uniformly and independently (of the other variables) from $\mathbb{F}_q$. For similar reasons, each of $\{\vec{\kappa}'^{(\chi,j')}\}_{j' \in [m_\chi]}$ and $\{\vec{\kappa}^{(\chi,j)}\}_{j \in [n]}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q^2$. In order to see that $(\{\widetilde{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ and $(\{\widehat{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widehat{\alpha}_{\chi,j'}, \widehat{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ simulated by $\mathcal{B}_{2\text{-}2}$ have the desired distribution in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and $\text{Hyb}_{2\text{-}\chi\text{-}4}$ respectively, i.e., their distributions coincide with the output distribution of $\mathsf{PGB}(f_\chi)$ with uniform and independent (of the other variables of those experiments) randomness, let $\vec{r}^{(\chi)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_\chi+n-1}$ be the random vector used while generating the constants $(\{\breve{\sigma}_{\chi,j}\}_{j \in [n]}, \{\breve{\alpha}_{\chi,j'}, \breve{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$. Then, by the linearity property of PGB (as described in Section 2.3), it follows that the constants $(\{\widetilde{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ and $(\{\widehat{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widehat{\alpha}_{\chi,j'}, \widehat{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ respectively are the outputs of $\mathsf{PGB}(f_\chi)$ with randomness $\vec{r}'^{(\chi)} = \widetilde{\pi}\vec{r}^{(\chi)}$ and $\vec{r}''^{(\chi)} = \widetilde{\pi}'\vec{r}^{(\chi)}$. Since, $\vec{r}^{(\chi)}$ is uniformly and independently (of the other variables) sampled from $\mathbb{F}_q^{m_\chi+n-1}$, it follows that $\vec{r}'^{(\chi)}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q^{m_\chi+n-1}$ except when $\widetilde{\pi} = 0$, and so is $\vec{r}''^{(\chi)}$ except when $\widetilde{\pi}' = 0$. Thus, $(\{\widetilde{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widetilde{\alpha}_{\chi,j'}, \widetilde{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ simulated by $\mathcal{B}_{2\text{-}2}$ have the desired distribution in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ except when $\widetilde{\pi} = 0$, and the same holds for $(\{\widehat{\sigma}_{\chi,j}\}_{j \in [n]}, \{\widehat{\alpha}_{\chi,j'}, \widehat{\gamma}_{\chi,j'}\}_{j' \in [m_\chi]})$ in $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ except when $\widetilde{\pi}' = 0$.

Also, the components of the ciphertext CT returned to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}2}$ have the form

$$\begin{aligned}\boldsymbol{c}'^{(\iota')} &= (\omega(1, x_{\iota'}), \widetilde{\tau}(1, x_{\iota'}), \widetilde{\theta}(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],\\ \boldsymbol{c}^{(\iota)} &= (\omega(1, z_\iota), \widetilde{\tau}(1, s_\iota), \widetilde{\theta}(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],\end{aligned}$$

which coincides with those in Eq. (3.6) corresponding to both $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$, where $\tau = \widetilde{\tau}$ and $\theta = \widetilde{\theta}$ are clearly distributed uniformly and independently (of the other variables) in $\mathbb{F}_q$. Moreover, for all $h \in [q_{\text{KEY}}]\backslash\{\chi\}$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\boldsymbol{k}_{h,j}\}_{j \in [n]}$ of the h^{th} decryption key $\text{SK}(f_h)$ given to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}2}$ are generated the same as in Eq. (3.7), if $h < \chi$, and as in Eq. (3.2), if $h > \chi$, which are their proper forms in the respective cases both in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and in $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$. Finally, the public parameters MPK provided to $\mathcal{H}$ by $\mathcal{B}_{2\text{-}2}$ are clearly distributed identically to those in both $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ and $\text{Hyb}_{2\text{-}\chi\text{-}4}$. Therefore, it follows that the view of $\mathcal{H}$ simulated by $\mathcal{B}_{2\text{-}2}$ given χ and a $\mathsf{Problem\ 3}$ instance $\varrho^{\mathsf{P3}}_{\widehat{\beta}}$ for $\widehat{\beta} \in \{0,1\}$, coincides with that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ except with probability $1/q$, if $\widehat{\beta} = 0$, while that in $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$ except with probability $1/q$, if $\widehat{\beta} = 1$. Hence the lemma follows. □

Lemma B.7: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_3$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(2\text{-}q_{\text{KEY-PRE}}\text{-}4)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(3)}_{\mathcal{H}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_3}(\lambda) + 3/q$.*

Proof: In order to achieve the transition from $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$ to Hyb_3, we first alter the forms of the components $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}$ and $\{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$ of the ciphertext CT returned to $\mathcal{H}$ from those in Eq. (3.6), which corresponds to $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$, to those in Eq. (3.4), which corresponds to Hyb_3, via an information-theoretic change as shown in Case (II) in the proof of Lemma B.2. Then, for all $h \in [q_{\text{KEY}}]$, we modify the forms of the components $\{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j \in [n]}$ of the h^{th} decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ from those in Eq. (3.7), if $h \in [q_{\text{KEY-PRE}}]$, and Eq. (3.2), if $h \in [q_{\text{KEY-PRE}}] + 1, q_{\text{KEY}}]$, which correspond to the respective cases in $\mathsf{Hyb}_{2\text{-}q_{\text{KEY-PRE}}\text{-}4}$, to those in Eq. (3.8), which correspond to Hyb_3, and we bound the difference in the advantage of $\mathcal{H}$ with that of an algorithm $\mathcal{B}_3$ against $\mathsf{Problem\ 2}$ in a manner similar to that in the proof of Lemma B.3. We omit the details to avoid repetition. □

Lemma B.8: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ, $|\mathsf{Adv}^{(3)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(4)}_{\mathcal{H}}(\lambda)| \leq 1/q$.*

Proof: In order to prove Lemma B.8, we show that the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ in Hyb_3 and that in Hyb_4 are equivalent. Consider the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h \in [q_{\text{KEY}}]}, \text{CT})$ in Hyb_4. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ using the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [n'+n]}$ use in Hyb_4 as follows: Sample $\aleph \xleftarrow{\mathsf{U}} \mathbb{F}_q$, compute the vectors

$$\begin{aligned}
&\boldsymbol{d}^{(\imath,2+\ell)} = \boldsymbol{b}^{(\imath,2+\ell)} - \aleph \boldsymbol{b}^{(\imath,\ell)}, \boldsymbol{d}^{*(\imath,\ell)} = \boldsymbol{b}^{*(\imath,\ell)} + \aleph \boldsymbol{b}^{*(\imath,2+\ell)} \quad \text{for } \imath \in [n'+n], \ell \in [2], \\
&\text{and set} \\
&\left.\begin{array}{l} \mathbb{D}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{d}^{(\imath,3)}, \boldsymbol{d}^{(\imath,4)}, \boldsymbol{b}^{(\imath,5)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \\ \mathbb{D}^*_\imath = \{\boldsymbol{d}^{*(\imath,1)}, \boldsymbol{d}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,3)}, \ldots, \boldsymbol{b}^{*(\imath,9)}\} \end{array}\right\} \text{ for } \imath \in [n'+n].
\end{aligned} \tag{B.19}$$

It can be readily observed that the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ are indeed dual orthonormal, and are distributed the same as the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [n'+n]}$.

Now, notice that the components of the ciphertext CT returned to $\mathcal{H}$ in Hyb_4 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ as

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')} &= (\vec{0}^2, \tau(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \\
&= (\omega(1, x_{\iota'}), \tau(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota' \in [n'], \\
\boldsymbol{c}^{(\iota)} &= (\vec{0}^2, \tau(1, z_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \\
&= (\omega(1, z_\iota), \tau(1, z_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota \in [n],
\end{aligned} \tag{B.20}$$

where $\omega = \aleph\tau$, and all the other variables are generated as in Hyb_4. Also, for $h \in [q_{\text{KEY}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ provided to $\mathcal{H}$ in Hyb_4 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ as

(a) $(h \in [q_{\text{KEY-PRE}}])$

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'}), (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}'_{h,j'}, \widetilde{\alpha}'_{h,j'}), (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h), (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}'_{h,j}, \widetilde{\zeta}'_h), (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n].\end{aligned} \tag{B.21}$$

(b) $(h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}])$

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'}), \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}'_{h,j'}, \widetilde{\alpha}'_{h,j'}), \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h), \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}'_{h,j}, \widetilde{\zeta}'_h), \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],\end{aligned} \tag{B.22}$$

where $\widetilde{\alpha}'_{h,j'} = \widetilde{\alpha}_{h,j'} - \aleph\alpha_{h,j'}$, $\widetilde{\gamma}'_{h,j'} = \widetilde{\gamma}_{h,j'} - \aleph\gamma_{h,j'}$ for all $h \in [q_{\text{KEY}}], j' \in [m_h]$, $\widetilde{\sigma}'_{h,j} = \widetilde{\sigma}_{h,j} - \aleph\sigma_{h,j}$ for all $h \in [q_{\text{KEY}}], j \in [n]$, $\widetilde{\zeta}'_h = \widetilde{\zeta}_h - \aleph\zeta_h$ for all $h \in [q_{\text{KEY}}]$, and all the other variables are generated as in Hyb_4.

Clearly, ω is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ except when $\tau = 0$, i.e., except with probability $1/q$, since $\aleph$ is sampled uniformly and independently (of the other variables) from $\mathbb{F}_q$. Similarly, $\{\widetilde{\zeta}'_h\}_{h\in[q_{\text{KEY}}]}$ are uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ since the variables $\{\widetilde{\zeta}_h\}_{h\in[q_{\text{KEY}}]}$ are sampled uniformly and independently (of the other variables) from $\mathbb{F}_q$. In order to observe that for all $h \in [q_{\text{KEY}}]$, the constants $(\{\widetilde{\sigma}'_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}'_{h,j'}, \widetilde{\gamma}'_{h,j'}\}_{j'\in[m_h]})$ are the outputs of $\mathsf{PGB}(f_h)$ with fresh uniform randomness, let $\vec{r}^{(h,1)}$ and $\vec{r}^{(h,2)}$ be the random vectors used by $\mathsf{PGB}(f_h)$ while generating the constants $(\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]})$ and $(\{\widetilde{\sigma}_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{h,j'}, \widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ respectively for $h \in [q_{\text{KEY}}]$. Then, by the linearity property of PGB, as described in Section 2.3, it follows that for all $h \in [q_{\text{KEY}}]$, the constants $(\{\widetilde{\sigma}'_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}'_{h,j'}, \widetilde{\gamma}'_{h,j'}\}_{j'\in[m_h]})$ are the output of $\mathsf{PGB}(f_h)$ with randomness $\vec{r}'^{(h)} = \vec{r}^{(h,2)} - \aleph\vec{r}^{(h,1)}$. Since for all $h \in [q_{\text{KEY}}]$, the vectors $\vec{r}^{(h,2)}$ are sampled uniformly and independently (of the other variables) from $\mathbb{F}_q^{m_h+n-1}$, the vectors $\{\vec{r}'^{(h)}\}_{h\in[q_{\text{KEY}}]}$ are also uniformly distributed in $\mathbb{F}_q^{m_h+n-1}$, and are independent of all the other variables.

Now, note that in the view of $\mathcal{H}$, both the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ are consistent with the public parameters MPK. Further, for all $h \in [q_{\text{KEY}}]$, the components of the hth decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their forms as in Eq. (3.8), which is their proper forms in Hyb_3, under the basis transformation. Finally, since the RHS of Eq. (B.20) and that of Eq. (3.4) have the same form except with probability $1/q$, it follows that the components of the ciphertext CT returned to $\mathcal{H}$ in Hyb_4 can be conceptually changed to those in Hyb_3 except with probability $1/q$. Hence the lemma follows. □

Lemma B.9: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_4$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(4)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(5)}_{\mathcal{H}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P2}}_{\mathcal{B}_4}(\lambda) + 5/q + \mathsf{negl}(\lambda)$, where negl is some negligible function.*

Proof: In order to prove Lemma B.9, we construct below a probabilistic algorithm $\mathcal{B}_4$ against $\mathsf{Problem\ 2}$ using a stateful probabilistic adversary $\mathcal{H}$ for distinguishing between Hyb_4 and Hyb_5 as a black-box sub-routine. Suppose $\mathcal{B}_4$ is given an instance of $\mathsf{Problem\ 2}$,

$$\varrho^{\mathsf{P2}}_{\widehat{\beta}} = (\mathsf{params}, \{\widetilde{\mathbb{B}}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}, \boldsymbol{y}^{(0,\widehat{\beta})}, \boldsymbol{f}^{(0)}, \{\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})}, \boldsymbol{f}^{(\imath,\ell)}\}_{\imath\in[n'+n],\ell\in[2]}),$$

where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathrm{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}));\\
&\widetilde{\delta}, \widetilde{\kappa}, \widetilde{\pi}, \widetilde{\omega}, \widetilde{\tau} \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,3)}, \ldots, \boldsymbol{b}^{(0,6)}\};\\
&\boldsymbol{y}^{(0,0)} = (\widetilde{\delta}, 0,0,0,\widetilde{\kappa},0)_{\mathbb{B}^*_0}, \boldsymbol{y}^{(0,1)} = (\widetilde{\delta}, \widetilde{\pi}, 0,0,\widetilde{\kappa},0)_{\mathbb{B}^*_0};\\
&\boldsymbol{f}^{(0)} = (\widetilde{\omega}, \widetilde{\tau}, 0,0,0,0)_{\mathbb{B}_0};\\
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,5)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0), \vec{e}^{(2)} = (0,1) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}
&\boldsymbol{y}^{(\imath,\ell,0)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
&\boldsymbol{y}^{(\imath,\ell,1)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}^*_\imath}\\
&\boldsymbol{f}^{(\imath,\ell)} = (\widetilde{\omega}\vec{e}^{(\ell)}, \widetilde{\tau}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, 0)_{\mathbb{B}_\imath}
\end{aligned}\right\} \text{ for } \imath \in [n'+n], \ell \in [2].
\end{aligned}$$

$\mathcal{B}_4$ interacts with $\mathcal{H}$ as follows:

1. First, $\mathcal{B}_4$ (implicitly) sets new dual orthogonal bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ using the bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ of the Problem 2 instance as

$$\left.\begin{aligned}
\mathbb{D}_\imath &= \{\boldsymbol{d}^{(\imath,1)}, \ldots, \boldsymbol{d}^{(\imath,9)}\} = \{\boldsymbol{b}^{(\imath,5)}, \boldsymbol{b}^{(\imath,6)}, \boldsymbol{b}^{(\imath,3)}, \boldsymbol{b}^{(\imath,4)}, \boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)},\\
&\qquad \boldsymbol{b}^{(\imath,7)}, \ldots, \boldsymbol{b}^{(\imath,9)}\}\\
\mathbb{D}^*_\imath &= \{\boldsymbol{d}^{*(\imath,1)}, \ldots, \boldsymbol{d}^{*(\imath,9)}\} = \{\boldsymbol{b}^{*(\imath,5)}, \boldsymbol{b}^{*(\imath,6)}, \boldsymbol{b}^{*(\imath,3)}, \boldsymbol{b}^{*(\imath,4)}, \boldsymbol{b}^{*(\imath,1)},\\
&\qquad \boldsymbol{b}^{*(\imath,2)}, \boldsymbol{b}^{*(\imath,7)}, \ldots, \boldsymbol{b}^{*(\imath,9)}\}
\end{aligned}\right\} \text{ for } \imath \in [n'+n]. \quad \text{(B.23)}$$

   Observe that $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ are indeed dual orthogonal, and are distributed the same as the bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ of the Problem 2 instance. $\mathcal{B}_4$ provides $\mathcal{H}$ the public parameters $\mathrm{MPK} = (\mathsf{params}, \{\widehat{\mathbb{D}}_\imath\}_{\imath\in[n'+n]})$, where params is taken from the given Problem 2 instance, and for all $\imath \in [n'+n]$, $\widehat{\mathbb{D}}_\imath = \{\boldsymbol{d}^{(\imath,1)}, \boldsymbol{d}^{(\imath,2)}, \boldsymbol{d}^{(\imath,9)}\} = \{\boldsymbol{b}^{(\imath,5)}, \boldsymbol{b}^{(\imath,6)}, \boldsymbol{b}^{(\imath,9)}\}$ is also constructed from the given Problem 2 instance.
2. For $h \in [q_{\mathrm{KEY}}]$, in response to the hth decryption key query of $\mathcal{H}$ for some function $f_h \in \mathcal{F}^{(q,n',n)}_{\mathrm{ABP\circ IP}}$, $\mathcal{B}_4$ proceeds as follows:
   (a) ($h \in [q_{\mathrm{KEY\text{-}PRE}}]$) $\mathcal{B}_4$ creates $\big((\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big), \big((\{\breve{\sigma}_{h,j}\}_{j\in[n]}, \{\breve{\alpha}_{h,j'}, \breve{\gamma}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big), \big((\{\bar{\sigma}_{h,j}\}_{j\in[n]}, \{\bar{\alpha}_{h,j'}, \bar{\gamma}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$, samples $\zeta_h, \breve{\zeta}_h, \bar{\zeta}_h$, $\{\widehat{\kappa}'_{h,j',1}, \widehat{\kappa}'_{h,j',2}\}_{j'\in[m_h]}, \{\widehat{\kappa}_{h,j,1}, \widehat{\kappa}_{h,j,2}\}_{j\in[n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes

$$\begin{aligned}
\boldsymbol{k}'^{(h,j')} &= \breve{\gamma}_{h,j'}\boldsymbol{y}^{(\rho_h(j'),1,\widehat{\beta})} + \breve{\alpha}_{h,j'}\boldsymbol{y}^{(\rho_h(j'),2,\widehat{\beta})} + \bar{\gamma}_{h,j'}\boldsymbol{b}^{*(\rho_h(j'),1)} + \bar{\alpha}_{h,j'}\boldsymbol{b}^{*(\rho_h(j'),2)} +\\
&\quad \gamma_{h,j'}\boldsymbol{b}^{*(\rho_h(j'),5)} + \alpha_{h,j'}\boldsymbol{b}^{*(\rho_h(j'),6)} + \widehat{\kappa}'_{h,j',1}\boldsymbol{b}^{*(\rho_h(j'),7)} + \widehat{\kappa}'_{h,j',2}\boldsymbol{b}^{(\rho_h(j'),8)} \text{ for } j' \in [m_h],\\
\boldsymbol{k}^{(h,j)} &= \breve{\sigma}_{h,j}\boldsymbol{y}^{(n'+j,1,\widehat{\beta})} + \breve{\zeta}_h\boldsymbol{y}^{(n'+j,2,\widehat{\beta})} + \bar{\sigma}_{h,j}\boldsymbol{b}^{*(n'+j,1)} + \bar{\zeta}_h\boldsymbol{b}^{*(n'+j,2)} +\\
&\quad \sigma_{h,j}\boldsymbol{b}^{*(n'+j,5)} + \zeta_h\boldsymbol{b}^{*(n'+j,6)} + \widehat{\kappa}_{h,j,1}\boldsymbol{b}^{*(n'+j,7)} + \widehat{\kappa}_{h,j,2}\boldsymbol{b}^{*(n'+j,8)} \text{ for } j \in [n],
\end{aligned}$$

   and returns the decryption key $\mathrm{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ to $\mathcal{H}$.
   (b) ($h \in [q_{\mathrm{KEY\text{-}PRE}}+1, q_{\mathrm{KEY}}]$) $\mathcal{B}_4$ provides $\mathcal{H}$ with a decryption key $\mathrm{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j\in[n]}$ of which are formed as in Eq. (3.8) using the bases $\{\mathbb{D}^*_\imath\}_{\imath\in[n'+n]}$ constructed in Step 1 by modifying the original bases $\{\mathbb{B}^*_\imath\}_{\imath\in[n'+n]}$ of the given Problem 2 instance.

3. When $\mathcal{B}_4$ receives the ciphertext query from $\mathcal{H}$ for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{B}_4$ samples $\vec{s}' \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n | R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{s})) = R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) \forall$ $h \in [q_{\text{KEY-PRE}}]\}$, samples $\widehat{\theta}, \{\varphi'_{\iota'}\}_{\iota' \in [n']}, \{\varphi_\iota\}_{\iota \in [n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes

$$\begin{aligned}\boldsymbol{c}'^{(\iota')} &= \widehat{\theta}\boldsymbol{f}^{(\iota',1)} + \widehat{\theta}x_{\iota'}\boldsymbol{f}^{(\iota',2)} + \varphi'_{\iota'}\boldsymbol{b}^{(\iota',9)} \text{ for } \iota' \in [n'],\\ \boldsymbol{c}^{(\iota)} &= \widehat{\theta}\boldsymbol{f}^{(n'+\iota,1)} + \widehat{\theta}z_\iota\boldsymbol{f}^{(n'+\iota,2)} + \widehat{\theta}s'_\iota\boldsymbol{b}^{(n'+\iota,2)} + \varphi_\iota\boldsymbol{b}^{(n'+\iota,9)} \text{ for } \iota \in [n],\end{aligned}$$

and hands $\mathcal{H}$ the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$.

4. $\mathcal{H}$ eventually outputs a bit $\beta \in \{0,1\}$. $\mathcal{B}_4$ outputs $\widehat{\beta}' = \beta$ as its guess bit in its $\mathsf{Problem\ 2}$ challenge.

Observe that when $\widehat{\beta} = 0$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,0)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n], \ell \in$ $[2]$, then for all $h \in [q_{\text{KEY-PRE}}]$, the components of the hth decryption key $\text{SK}(f_h)$ returned by $\mathcal{B}_4$ to $\mathcal{H}$ take the form

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}, \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}), \vec{0}^2, (\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{B}^*_{\rho_h(j')}}\\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}, \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}), (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{D}^*_{\rho_h(j')}}\\ &\qquad \text{for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}, \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h), \vec{0}^2, (\sigma_{h,j}, \zeta_h), (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{B}^*_{n'+j}}\\ &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}, \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h), (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],\end{aligned}$$

which coincides with those in Hyb_5 (Eq. (3.7)), where we have $\widehat{\alpha}_{h,j'} = \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}, \widehat{\gamma}_{h,j'} = \widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}$ for $j' \in [m_h]$, $\widehat{\sigma}_{h,j} = \widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}$ for $j \in [n]$, $\widehat{\zeta}_h = \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h$, $\vec{\kappa}'^{(h,j')} = (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2})$ for $j' \in [m_h]$, and $\vec{\kappa}^{(h,j)} = (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2})$ for $j \in [n]$. On the other hand, in case $\widehat{\beta} = 1$, i.e., $\boldsymbol{y}^{(\imath,\ell,\widehat{\beta})} = \boldsymbol{y}^{(\imath,\ell,1)} = (\widetilde{\delta}\vec{e}^{(\ell)}, \widetilde{\pi}\vec{e}^{(\ell)}, \vec{0}^2, \widetilde{\kappa}\vec{e}^{(\ell)}, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n], \ell \in [2]$, then for all $h \in [q_{\text{KEY-PRE}}]$, the components of the hth decryption key $\text{SK}(f_h)$ given by $\mathcal{B}_4$ to $\mathcal{H}$ take the form

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}, \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}), (\widetilde{\pi}\breve{\gamma}_{h,j'}, \widetilde{\pi}\breve{\alpha}_{h,j'}), (\gamma_{h,j'}, \alpha_{h,j'}),\\ &\qquad (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{B}^*_{\rho_h(j')}}\\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\pi}\breve{\gamma}_{h,j'}, \widetilde{\pi}\breve{\alpha}_{h,j'}), (\widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}, \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}),\\ &\qquad (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}, \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h), (\widetilde{\pi}\breve{\sigma}_{h,j}, \widetilde{\pi}\breve{\zeta}_h), (\sigma_{h,j}, \zeta_h), (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{B}^*_{n'+j}}\\ &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\pi}\breve{\sigma}_{h,j}, \widetilde{\pi}\breve{\zeta}_h), (\widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}, \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h),\\ &\qquad (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n],\end{aligned}$$

which coincides with that in Hyb_4 (Eq. (3.8)), where we have $\widehat{\alpha}_{h,j'} = \widetilde{\delta}\breve{\alpha}_{h,j'} + \bar{\alpha}_{h,j'}, \widehat{\gamma}_{h,j'} = \widetilde{\delta}\breve{\gamma}_{h,j'} + \bar{\gamma}_{h,j'}$, $\widetilde{\alpha}_{h,j'} = \widetilde{\pi}\breve{\alpha}_{h,j'}, \widetilde{\gamma}_{h,j'} = \widetilde{\pi}\breve{\gamma}_{h,j'}$ for $j' \in [m_h]$, $\widehat{\sigma}_{h,j} = \widetilde{\delta}\breve{\sigma}_{h,j} + \bar{\sigma}_{h,j}$, $\widetilde{\sigma}_{h,j} = \widetilde{\pi}\breve{\sigma}_{h,j}$ for $j \in [n]$, $\widehat{\zeta}_h = \widetilde{\delta}\breve{\zeta}_h + \bar{\zeta}_h$, $\widetilde{\zeta}_h = \widetilde{\pi}\breve{\zeta}_h$, $\vec{\kappa}'^{(h,j')} = (\widetilde{\kappa}\breve{\gamma}_{h,j'} + \widehat{\kappa}'_{h,j',1}, \widetilde{\kappa}\breve{\alpha}_{h,j'} + \widehat{\kappa}'_{h,j',2})$ for $j' \in [m_h]$, and $\vec{\kappa}^{(h,j)} = (\widetilde{\kappa}\breve{\sigma}_{h,j} + \widehat{\kappa}_{h,j,1}, \widetilde{\kappa}\breve{\zeta}_h + \widehat{\kappa}_{h,j,2})$ for $j \in [n]$.

Clearly, for all $h \in [q_{\text{KEY-PRE}}]$, $\widehat{\zeta}_h$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$, and so is $\widetilde{\zeta}_h$ except when $\widetilde{\pi} = 0$. This is because $\bar{\zeta}_h$ and $\breve{\zeta}_h$ are sampled uniformly and independently (of the other variables) from $\mathbb{F}_q$. For similar reasons, for all $h \in [q_{\text{KEY-PRE}}]$, each of $\{\vec{\kappa}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\vec{\kappa}^{(h,j)}\}_{j \in [n]}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q^2$. In order to see that for all $h \in [q_{\text{KEY-PRE}}]$, $(\{\widehat{\sigma}_{h,j}\}_{j \in [n]}, \{\widehat{\alpha}_{h,j'}, \widehat{\gamma}_{h,j'}\}_{j' \in [m_h]})$

and $(\{\widetilde{\sigma}_{h,j}\}_{j\in[n]},\{\widetilde{\alpha}_{h,j'},\widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ simulated by $\mathcal{B}_4$ have the desired distribution in Hyb_5 and Hyb$_4$, i.e., their distribution coincides with the output distribution of $\mathsf{PGB}(f_h)$ with uniform and independent (of the other variables) randomness, let for $h\in[q_{\text{KEY-PRE}}]$, $\vec{r}^{(h,1)}\xleftarrow{\mathsf{U}}\mathbb{F}_q^{m_h+n-1}$ and $\vec{r}^{(h,2)}\xleftarrow{\mathsf{U}}\mathbb{F}_q^{m_h+n-1}$ be the random vectors used while forming the constants $(\{\breve{\sigma}_{h,j}\}_{j\in[n]},\{\breve{\alpha}_{h,j'},\breve{\gamma}_{h,j'}\}_{j'\in[m_h]})$ and $(\{\bar{\sigma}_{h,j}\}_{j\in[n]},\{\bar{\alpha}_{h,j'},\bar{\gamma}_{h,j'}\}_{j'\in[m_h]})$ respectively. Then, by the linearity property of PGB (as described in Section 2.3), it follows that for all $h\in[q_{\text{KEY-PRE}}]$, the constants $(\{\widehat{\sigma}_{h,j}\}_{j\in[n]},\{\widehat{\alpha}_{h,j'},\widehat{\gamma}_{h,j'}\}_{j'\in[m_h]})$ and $(\{\widetilde{\sigma}_{h,j}\}_{j\in[n]},\{\widetilde{\alpha}_{h,j'},\widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ respectively are the outputs of $\mathsf{PGB}(f_h)$ with randomness $\vec{r}''^{(h)}=\widetilde{\delta}\vec{r}^{(h,1)}+\vec{r}^{(h,2)}$ and $\vec{r}'^{(h)}=\widetilde{\pi}\vec{r}^{(h,1)}$. Since, for all $h\in[q_{\text{KEY-PRE}}]$, the vectors $\vec{r}^{(h,1)}$ and $\vec{r}^{(h,2)}$ are uniformly and independently (of one another and of all the other variables) sampled from $\mathbb{F}_q^{m_h+n-1}$, it follows that $\vec{r}''^{(h)}$ is uniformly distributed in $\mathbb{F}_q^{m_h+n-1}$, and so is $\vec{r}'^{(h)}$ except when $\widetilde{\pi}=0$, as well as they are independent of one another and of all the other variables. Thus, for all $h\in[q_{\text{KEY-PRE}}]$, $(\{\widehat{\sigma}_{h,j}\}_{j\in[n]},\{\widehat{\alpha}_{h,j'},\widehat{\gamma}_{h,j'}\}_{j'\in[m_h]})$ and $(\{\widetilde{\sigma}_{h,j}\}_{j\in[n]},\{\widetilde{\alpha}_{h,j'},\widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ simulated by $\mathcal{B}_4$ have the desired distributions except when $\widetilde{\pi}=0$.

Also, the components of the ciphertext CT returned to $\mathcal{H}$ by $\mathcal{B}_4$ have the form

$$\begin{aligned}
\boldsymbol{c}'^{(\iota')}&=(\widehat{\theta}\widetilde{\omega}(1,x_{\iota'}),\widehat{\theta}\widetilde{\tau}(1,x_{\iota'}),\vec{0}^2,\vec{0}^2,\varphi'_{\iota'})_{\mathbb{B}_{\iota'}}\\
&=(\vec{0}^2,\widehat{\theta}\widetilde{\tau}(1,x_{\iota'}),\widehat{\theta}\widetilde{\omega}(1,x_{\iota'}),\vec{0}^2,\varphi'_{\iota'})_{\mathbb{D}_{\iota'}}\ \text{for}\ \iota'\in[n'],\\
\boldsymbol{c}^{(\iota)}&=(\widehat{\theta}\widetilde{\omega}(1,z_\iota)+\widehat{\theta}(0,s'_\iota),\widehat{\theta}\widetilde{\tau}(1,z_\iota),\vec{0}^2,\vec{0}^2,\varphi_\iota)_{\mathbb{B}_{n'+\iota}}\\
&=(\vec{0}^2,\widehat{\theta}\widetilde{\tau}(1,z_\iota),\widehat{\theta}\widetilde{\omega}(1,z_\iota)+\widehat{\theta}(0,s'_\iota),\vec{0}^2,\varphi_\iota)_{\mathbb{D}_{n'+\iota}}\ \text{for}\ \iota\in[n],
\end{aligned}$$

which coincides with those in Eq. (3.9) corresponding to both Hyb_4 and Hyb_5, where we have $\tau=\widehat{\theta}\widetilde{\tau}$, $\theta=\widehat{\theta}\widetilde{\omega}$, and $\vec{s}=\vec{z}+\widetilde{\omega}^{-1}\vec{s}'$. Clearly, $\tau=\widehat{\theta}\widetilde{\tau}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ except when $\widehat{\theta}=0$, since $\widetilde{\tau}$ is so. Similarly, $\theta=\widehat{\theta}\widetilde{\omega}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ except when $\widetilde{\omega}=0$, since $\widehat{\theta}$ is so. Further, $\vec{s}=\vec{z}+\widetilde{\omega}^{-1}\vec{s}'$ is uniformly and independently (of the other variables) distributed in S other than negligible probability, except when $\widetilde{\omega}=0$, since $\vec{s}'$ is sampled uniformly and independently (of the other variables) from S and $\vec{z}\in S$. Moreover, for all $h\in[q_{\text{KEY-PRE}}+1,q_{\text{KEY}}]$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]}$ and $\{\boldsymbol{k}_{h,j}\}_{j\in[n]}$ of the h^{th} decryption key $\text{SK}(f_h)$ given to $\mathcal{H}$ by $\mathcal{B}_4$ are generated the same as in Eq. (3.8) which are their proper forms both in Hyb_4 and in Hyb_5. Finally, the public parameters MPK provided to $\mathcal{H}$ by $\mathcal{B}_4$ are clearly distributed identically to those in both Hyb_4 and Hyb$_5$. Therefore, it follows that the view of $\mathcal{H}$ simulated by $\mathcal{B}_4$ given a $\mathsf{Problem\ 2}$ instance $\varrho^{\mathsf{P2}}_{\widehat{\beta}}$ for $\widehat{\beta}\in\{0,1\}$, coincides with that in Hyb_4 except with probability $3/q+\mathsf{negl}(\lambda)$, i.e., except when one of $\widetilde{\pi}$, $\widetilde{\omega}$, and $\widehat{\theta}$ is 0, if $\widehat{\beta}=1$, while that in Hyb_5 except with probability $2/q+\mathsf{negl}(\lambda)$, i.e., except when one of $\widetilde{\omega}$ and $\widehat{\theta}$ is 0, if $\widehat{\beta}=0$, where negl is some negligible function. Hence the lemma follows. □

Lemma B.10: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ,* $\mathsf{Adv}^{(5)}_{\mathcal{H}}(\lambda)=\mathsf{Adv}^{(6)}_{\mathcal{H}}(\lambda)$.

Proof: In order to prove Lemma B.10, we show that the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK},\{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]},\text{CT})$ in Hyb_5 and that in Hyb_6 are equivalent. Consider the distribution $(\text{MPK},\{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]},\text{CT})$ in Hyb_5. Let us define new set of dual orthonormal bases $\{\mathbb{D}_\iota,\mathbb{D}^*_\iota\}_{\iota\in[n'+n]}$ from the original set of bases $\{\mathbb{B}_\iota,\mathbb{B}^*_\iota\}_{\iota\in[n'+n]}$ used in Hyb_5 as follows: Generate matrices $\boldsymbol{Z}'^{(\iota')}\xleftarrow{\mathsf{U}}\{\boldsymbol{Z}\in\mathsf{GL}(2,\mathbb{F}_q)|(1,x_{\iota'})\boldsymbol{Z}=\vec{e}^{(2)}=(0,1)\}$ for $\iota'\in[n']$, $\boldsymbol{Z}_\iota\xleftarrow{\mathsf{U}}\{\boldsymbol{Z}\in\mathsf{GL}(2,\mathbb{F}_q|(1,z_\iota)\boldsymbol{Z}=\vec{e}^{(2)}=(0,1)\}$ for $\iota\in[n]$, define $\boldsymbol{U}'^{(\iota')}=((\boldsymbol{Z}'^{(\iota')})^{-1})^{\intercal}$ for $\iota'\in[n']$, $\boldsymbol{U}^{(\iota)}=((\boldsymbol{Z}^{(\iota)})^{-1})^{\intercal}$ for $\iota\in[n]$, where $(\vec{x},\vec{z})\in\mathbb{F}_q^{n'}\times\mathbb{F}_q^{n}$ is the pair of public-private attribute strings corresponding to the ciphertext query of $\mathcal{H}$, compute the vectors

$$\begin{pmatrix} \boldsymbol{d}^{(\iota',3)} \\ \boldsymbol{d}^{(\iota',4)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{-1} \begin{pmatrix} \boldsymbol{b}^{(\iota',3)} \\ \boldsymbol{b}^{(\iota',4)} \end{pmatrix}, \quad \begin{pmatrix} \boldsymbol{d}^{*(\iota',3)} \\ \boldsymbol{d}^{*(\iota',4)} \end{pmatrix} = (\boldsymbol{Z}'^{(\iota')})^{\intercal} \begin{pmatrix} \boldsymbol{b}^{*(\iota',3)} \\ \boldsymbol{b}^{*(\iota',4)} \end{pmatrix} \text{ for } \iota' \in [n'],$$

$$\begin{pmatrix} \boldsymbol{d}^{(n'+\iota,3)} \\ \boldsymbol{d}^{(n'+\iota,4)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{-1} \begin{pmatrix} \boldsymbol{b}^{(n'+\iota,3)} \\ \boldsymbol{b}^{(n'+\iota,4)} \end{pmatrix}, \begin{pmatrix} \boldsymbol{d}^{*(n'+\iota,3)} \\ \boldsymbol{d}^{*(n'+\iota,4)} \end{pmatrix} = (\boldsymbol{Z}^{(\iota)})^{\intercal} \begin{pmatrix} \boldsymbol{b}^{*(n'+\iota,3)} \\ \boldsymbol{b}^{*(n'+\iota,4)} \end{pmatrix} \text{ for } \iota \in [n],$$

and set (B.24)

$$\left.\begin{array}{l} \mathbb{D}_{\iota'} = \{\boldsymbol{b}^{(\iota',1)}, \boldsymbol{b}^{(\iota',2)}, \boldsymbol{d}^{(\iota',3)}, \boldsymbol{d}^{(\iota',4)}, \boldsymbol{b}^{(\iota',5)}, \ldots, \boldsymbol{b}^{(\iota',9)}\} \\ \mathbb{D}^*_{\iota'} = \{\boldsymbol{b}^{*(\iota',1)}, \boldsymbol{b}^{*(\iota',2)}, \boldsymbol{d}^{*(\iota',3)}, \boldsymbol{d}^{*(\iota',4)}, \boldsymbol{b}^{*(\iota',5)}, \ldots, \boldsymbol{b}^{*(\iota',9)}\} \end{array}\right\} \text{ for } \iota' \in [n'],$$

$$\left.\begin{array}{l} \mathbb{D}_{n'+\iota} = \{\boldsymbol{b}^{(n'+\iota,1)}, \boldsymbol{b}^{(n'+\iota,2)}, \boldsymbol{d}^{(n'+\iota,3)}, \boldsymbol{d}^{(n'+\iota,4)}, \boldsymbol{b}^{(n'+\iota,5)}, \ldots, \boldsymbol{b}^{(n'+\iota,9)}\} \\ \mathbb{D}^*_{n'+\iota} = \{\boldsymbol{b}^{*(n'+\iota,1)}, \boldsymbol{b}^{*(n'+\iota,2)}, \boldsymbol{d}^{*(n'+\iota,3)}, \boldsymbol{d}^{*(n'+\iota,4)}, \boldsymbol{b}^{*(n'+\iota,5)}, \ldots, \boldsymbol{b}^{*(n'+\iota,9)}\} \end{array}\right\} \text{ for } \iota \in [n].$$

It can be readily observed that the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ are indeed dual orthonormal, and are distributed the same as the original set of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [n'+n]}$.

Now, notice that the components of the ciphertext CT returned to $\mathcal{H}$ in Hyb_5 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ as

$$\begin{aligned} \boldsymbol{c}'^{(\iota')} &= (\vec{0}^2, \tau(1, x_{\iota'}), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \\ &= (\vec{0}^2, \tau\vec{e}^{(2)}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \\ &= (\vec{0}^2, (0, \tau), \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{D}_{\iota'}} \text{ for } \iota' \in [n'], \\ \boldsymbol{c}^{(\iota)} &= (\vec{0}^2, \tau(1, z_\iota), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \\ &= (\vec{0}^2, \tau\vec{e}^{(2)}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{D}_{n'+\iota}} \\ &= (\vec{0}^2, (0, \tau), \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{D}_{n'+\iota}} \text{ for } \iota \in [n], \end{aligned} \tag{B.25}$$

where all the variables are generated as in Hyb_5.

Also, for $h \in [q_{\text{KEY}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ provided to $\mathcal{H}$ in Hyb_5 can be expressed in terms of the new set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ as

(a) $(h \in [q_{\text{KEY-pre}}])$

$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ &= ((\sigma_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n]. \end{aligned} \tag{B.26}$$

(b) $(h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}])$

$$\begin{aligned} \boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'}), \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \\ &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})\boldsymbol{U}'^{(\rho_h(j'))}, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{D}^*_{\rho_h(j')}} \text{ for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h), \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \\ &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)\boldsymbol{U}^{(j)}, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{D}^*_{n'+j}} \text{ for } j \in [n], \end{aligned} \tag{B.27}$$

where the matrices $\{\boldsymbol{U}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{U}^{(\iota)}\}_{\iota \in [n]}$ are as computed above, and all the other variables are generated as in Hyb_5.

Now, note that in the view of $\mathcal{H}$, both the original sets of bases $\{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [n'+n]}$ and the transformed set of bases $\{\mathbb{D}_\imath, \mathbb{D}^*_\imath\}_{\imath \in [n'+n]}$ are consistent with the public parameters MPK. Further, for $h \in [q_{\text{KEY-PRE}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ returned to $\mathcal{H}$ preserve their form as in Eq. (3.7) which are their proper forms in Hyb_6, under the basis transformation. Finally, since the RHS of Eq. (B.25) (respectively Eq. (B.27)) and that of Eq. (3.10) (respectively Eq. (3.11)) have the same form, it follows that the components of the ciphertext CT and the components of the h^{th} decryption key $\text{SK}(f_h)$ for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, returned to $\mathcal{H}$ in Hyb_5 can be conceptually changed to those in Hyb_6. Hence the lemma follows. □

Lemma B.11: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_5$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(6)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(7)}_{\mathcal{H}}(\lambda)| \leq \mathsf{Adv}^{\mathsf{P4}}_{\mathcal{B}_5}(\lambda) + 1/q$.*

Proof: In order to prove Lemma B.11, we construct below a probabilistic algorithm $\mathcal{B}_5$ against Problem 4 using a stateful probabilistic adversary $\mathcal{H}$ for distinguishing between Hyb_6 and Hyb_7 as a black-box sub-routine. Suppose $\mathcal{B}_5$ is given an instance of Problem 4,

$$\varrho^{\mathsf{P4}}_{\widehat{\beta}} = (\mathsf{params}, \mathbb{B}_0, \mathbb{B}^*_0, \{\widetilde{\mathbb{B}}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [n'+n]}, \{\boldsymbol{y}^{(\imath,1,\widehat{\beta})}\}_{\imath \in [n'+n]}),$$

where

$$\begin{aligned}
&(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}^*_\imath\}_{\imath \in [0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9, \ldots, 9}^{n'+n}));\\
&\widetilde{\pi}, \widetilde{\kappa}_1, \widetilde{\kappa}_2 \xleftarrow{\mathsf{U}} \mathbb{F}_q;\\
&\widetilde{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,4)}, \ldots, \boldsymbol{b}^{(\imath,9)}\} \text{ for } \imath \in [n'+n];\\
&\vec{e}^{(1)} = (1,0) \in \mathbb{F}_q^2;\\
&\left.\begin{aligned}\boldsymbol{y}^{(\imath,1,0)} &= (\vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}^*_\imath}\\ \boldsymbol{y}^{(\imath,1,1)} &= (\vec{0}^2, \widetilde{\pi}\vec{e}^{(1)}, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}^*_\imath}\end{aligned}\right\} \text{ for } \imath \in [n'+n].
\end{aligned}$$

$\mathcal{B}_5$ interacts with $\mathcal{H}$ as follows:

1. First, $\mathcal{B}_5$ provides the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}\}_{\imath \in [n'+n]})$, all of which are taken from the given Problem 4 instance.
2. For $h \in [q_{\text{KEY}}]$, in response to the h^{th} decryption key query of $\mathcal{H}$ for some function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, $\mathcal{B}_5$ proceeds as follows:
   (a) ($h \in [q_{\text{KEY-PRE}}]$) $\mathcal{B}_5$ gives $\mathcal{H}$ a decryption key $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\boldsymbol{k}^{(h,j)}\}_{j \in [n]}$ of which are generated as in Eq. (3.7) using the bases $\{\mathbb{B}^*_\imath\}_{\imath \in [n'+n]}$ included within the given Problem 4 instance.
   (b) ($h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$) $\mathcal{B}_5$ generates $\big((\{\sigma_{h,j}\}_{j \in [n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']\big)$, $\big((\{\widetilde{\sigma}_{h,j}\}_{j \in [n]}, \{\widetilde{\alpha}_{h,j'}, \widetilde{\gamma}_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$, samples $\zeta_h, \widetilde{\zeta}_h, \{\widehat{\eta}'_{h,j'}\}_{j' \in [m_h]}$, $\{\widehat{\eta}_{h,j}\}_{j \in [n]}, \{\widehat{\kappa}'_{h,j',1}, \widehat{\kappa}'_{h,j',2}\}_{j' \in [m_h]}, \{\widehat{\kappa}_{h,j,1}, \widehat{\kappa}_{h,j,2}\}_{j \in [n]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, computes

$$\begin{aligned}
\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})\boldsymbol{U}'^{(\rho_h(j'))}, \vec{0}^2, (\widehat{\kappa}'_{h,j',1}, \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{B}^*_{\rho_h(j')}} +\\
&\quad \widehat{\eta}'_{h,j'}\boldsymbol{y}^{(\rho_h(j'),1,\widehat{\beta})} \text{ for } j' \in [m_h],\\
\boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)\boldsymbol{U}^{(j)}, \vec{0}^2, (\widehat{\kappa}_{h,j,1}, \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{B}^*_{n'+j}} + \widehat{\eta}_{h,j}\boldsymbol{y}^{(n'+j,1,\widehat{\beta})} \text{ for } j \in [n],
\end{aligned}$$

   where the matrices $\{\boldsymbol{U}'^{(\imath')}\}_{\imath' \in [n']}, \{\boldsymbol{U}^{(\imath)}\}_{\imath \in [n]}$ are formed in Step 3 below, and returns the decryption key $\text{SK}(f_h) = (f_h, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$ to $\mathcal{H}$. Note that for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the h^{th} decryption key query is made by $\mathcal{H}$ after making the ciphertext query. Hence, anything that $\mathcal{B}_5$ generates in Step 3 below can be used by $\mathcal{B}_5$ while answering these decryption key queries.

3. When $\mathcal{B}_5$ receives the ciphertext query from $\mathcal{H}$ for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{B}_5$ provides $\mathcal{H}$ with a ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$, the components $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}$ and $\{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$ of which are computed as in Eq. (3.10) using the partial bases $\{\widetilde{\mathbb{B}}_\imath\}_{\imath \in [n'+n]}$. $\mathcal{B}_5$ also samples $\boldsymbol{Z}'^{(\iota')} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, x_{\iota'}\boldsymbol{Z} = \vec{e}^{(2)} = (0, 1)\}$ for $\iota' \in [n']$, $\boldsymbol{Z}^{(\iota)} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, z_\iota)\boldsymbol{Z} = \vec{e}^{(2)} = (0, 1)\}$ for $\iota \in [n]$, as well as computes $\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^\intercal$ for $\iota' \in [n']$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^\intercal$ for $\iota \in [n]$. $\mathcal{B}_5$ uses $\{\boldsymbol{U}'^{(\iota')}\}_{\iota' \in [n']}$ and $\{\boldsymbol{U}^{(\iota)}\}_{\iota \in [n]}$ while simulating the decryption keys that $\mathcal{H}$ queries after the ciphertext query.
4. $\mathcal{H}$ eventually outputs a bit $\beta \in \{0, 1\}$. $\mathcal{B}_5$ outputs $\widehat{\beta}' = \beta$ as its guess bit in its $\mathsf{Problem\ 4}$ challenge.

First, note that by construction of the matrices $\{\boldsymbol{U}'^{(\iota')}\}_{\iota' \in [n']}$ and $\{\boldsymbol{U}^{(\iota)}\}_{\iota \in [n]}$, we have $\boldsymbol{U}'^{(\iota')}(\vec{e}^{(2)})^\intercal = (1, x_{\iota'})^\intercal$, i.e., the second column of $\boldsymbol{U}'^{(\iota')}$ is $(1, x_{\iota'})^\intercal$ for all $\iota' \in [n']$, and similarly, $\boldsymbol{U}^{(\iota)}(\vec{e}^{(2)})^\intercal = (1, z_\iota)^\intercal$, i.e., the second column of $\boldsymbol{U}^{(\iota)}$ is $(1, z_\iota)^\intercal$ for all $\iota \in [n]$. Consequently, observe that when $\widehat{\beta} = 0$, i.e., $\boldsymbol{y}^{(\imath, 1, \widehat{\beta})} = \boldsymbol{y}^{(\imath, 1, 0)} = (\vec{0}^2, \vec{0}^2, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n]$, then for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ returned by $\mathcal{B}_5$ to $\mathcal{H}$ take the form

$$\boldsymbol{k}'^{(h,j')} = ((\gamma_{h,j'}, \alpha_{h,j'}), (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'})\boldsymbol{U}'^{(\rho_h(j'))}, \vec{0}^2, (\widehat{\eta}'_{h,j'}\widetilde{\kappa}_1 + \widehat{\kappa}'_{h,j',1}, \widehat{\eta}'_{h,j'}\widetilde{\kappa}_2 + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{B}^*_{\rho_h(j')}} \quad \text{for } j' \in [m_h],$$
$$\boldsymbol{k}^{(h,j)} = ((\sigma_{h,j}, \zeta_h), (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h)\boldsymbol{U}^{(j)}, \vec{0}^2, (\widehat{\eta}_{h,j}\widetilde{\kappa}_1 + \widehat{\kappa}_{h,j,1}, \widehat{\eta}_{h,j}\widetilde{\kappa}_2 + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \quad \text{for } j \in [n],$$

which coincides with those in Hyb_6 (Eq. (3.11)), where we have $\vec{\kappa}'^{(h,j')} = (\widehat{\eta}'_{h,j'}\widetilde{\kappa}_1 + \widehat{\kappa}'_{h,j',1}, \widehat{\eta}'_{h,j'}\widetilde{\kappa}_2 + \widehat{\kappa}'_{h,j',2})$ for $j' \in [m_h]$, and $\vec{\kappa}^{(h,j)} = (\widehat{\eta}_{h,j}\widetilde{\kappa}_1 + \widehat{\kappa}_{h,j,1}, \widehat{\eta}_{h,j}\widetilde{\kappa}_2 + \widehat{\kappa}_{h,j,2})$ for $j \in [n]$. On the other hand, in case $\widehat{\beta} = 1$, i.e., $\boldsymbol{y}^{(\imath, 1, \widehat{\beta})} = \boldsymbol{y}^{(\imath, 1, 1)} = (\vec{0}^2, \widetilde{\pi}\vec{e}^{(1)}, \vec{0}^2, \widetilde{\kappa}_1, \widetilde{\kappa}_2, 0)_{\mathbb{B}_\imath^*}$ for $\imath \in [n'+n]$, then for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the components of the h^{th} decryption key $\text{SK}(f_h)$ given by $\mathcal{B}_5$ to $\mathcal{H}$ take the form

$$\begin{aligned}\boldsymbol{k}'^{(h,j')} &= ((\gamma_{h,j'}, \alpha_{h,j'}), ((\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'}) \cdot \vec{u}'^{(\rho_h(j'),1)} + \widehat{\eta}'_{h,j'}\widetilde{\pi}, (\widetilde{\alpha}_{h,j'}x_{\rho_h(j')} + \widetilde{\gamma}_{h,j'})), \vec{0}^2, \\ &\quad (\widehat{\eta}'_{h,j'}\widetilde{\kappa}_1 + \widehat{\kappa}'_{h,j',1}, \widehat{\eta}'_{h,j'}\widetilde{\kappa}_2 + \widehat{\kappa}'_{h,j',2}), 0)_{\mathbb{B}^*_{\rho_h(j')}} \quad \text{for } j' \in [m_h], \\ \boldsymbol{k}^{(h,j)} &= ((\sigma_{h,j}, \zeta_h), ((\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h) \cdot \vec{u}^{(j,1)} + \widehat{\eta}_{h,j}\widetilde{\pi}, (\widetilde{\zeta}_h z_\iota + \widetilde{\sigma}_{h,j})), \vec{0}^2, \\ &\quad (\widehat{\eta}_{h,j}\widetilde{\kappa}_1 + \widehat{\kappa}_{h,j,1}, \widehat{\eta}_{h,j}\widetilde{\kappa}_2 + \widehat{\kappa}_{h,j,2}), 0)_{\mathbb{B}^*_{n'+j}} \quad \text{for } j \in [n],\end{aligned}$$

which coincides with that in Hyb_7 (Eq. (3.12)), where we have $\eta_{h,j'} = (\widetilde{\gamma}_{h,j'}, \widetilde{\alpha}_{h,j'}) \cdot \vec{u}'^{(\rho_h(j'),1)} + \widehat{\eta}'_{h,j'}\widetilde{\pi}$ for $j' \in [m_h]$, $\eta_{h,j} = (\widetilde{\sigma}_{h,j}, \widetilde{\zeta}_h) \cdot \vec{u}^{(j,1)} + \widehat{\eta}_{h,j}\widetilde{\pi}$ for $j \in [n]$, $\vec{\kappa}'^{(h,j')} = (\widehat{\eta}'_{h,j'}\widetilde{\kappa}_1 + \widehat{\kappa}'_{h,j',1}, \widehat{\eta}'_{h,j'}\widetilde{\kappa}_2 + \widehat{\kappa}'_{h,j',2})$ for $j' \in [m_h]$, and $\vec{\kappa}^{(h,j)} = (\widehat{\eta}_{h,j}\widetilde{\kappa}_1 + \widehat{\kappa}_{h,j,1}, \widehat{\eta}_{h,j}\widetilde{\kappa}_2 + \widehat{\kappa}_{h,j,2})$ for $j \in [n]$. Here, $(\vec{u}'^{(\iota',1)})^\intercal$ stands for the first column of the matrix $\boldsymbol{U}'^{(\iota')}$ for all $\iota' \in [n']$, and $(\vec{u}^{(\iota,1)})^\intercal$ denotes the same for the matrix $\boldsymbol{U}^{(\iota)}$ for all $\iota \in [n]$.

Clearly for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, $\{\eta'_{h,j'}\}_{j' \in [m_h]}$ and $\{\eta_{h,j}\}_{j \in [n]}$ simulated by $\mathcal{B}_5$ are uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ except when $\widetilde{\pi} = 0$, since $\{\widehat{\eta}'_{h,j'}\}_{j' \in [m_h]}$ and $\{\widehat{\eta}_{h,j}\}_{j \in [n]}$ are so. For similar reasons, for all $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, each of $\{\vec{\kappa}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\vec{\kappa}^{(h,j)}\}_{j \in [n]}$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q^2$. Moreover, for all $h \in [q_{\text{KEY-PRE}}]$, the components $\{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}$ and $\{\boldsymbol{k}_{h,j}\}_{j \in [n]}$ of the h^{th} decryption key $\text{SK}(f_h)$ given to $\mathcal{H}$ by $\mathcal{B}_5$ are generated the same as in Eq. (3.7), which are their proper forms both in Hyb_6 and in Hyb_7. Further, the components $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}$ and $\{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$ of the ciphertext CT returned to $\mathcal{H}$ by $\mathcal{B}_5$ are generated as in Eq. (3.10), which are their proper forms both in Hyb_6 and in Hyb_7. Finally, the public parameters MPK provided to $\mathcal{H}$

by $\mathcal{B}_5$ are clearly distributed identically to those in both Hyb_6 and Hyb$_7$. Therefore, it follows that the view of $\mathcal{H}$ simulated by $\mathcal{B}_5$ given a $\mathsf{Problem\ 4}$ instance $\varrho^{\mathsf{P4}}_{\widehat{\beta}}$ for $\widehat{\beta} \in \{0,1\}$, coincides with that in Hyb_6, if $\widehat{\beta} = 0$, while that in Hyb_7 except with probability $1/q$, i.e., except when $\widetilde{\pi} = 0$, if $\widehat{\beta} = 1$. Hence the lemma follows. □

Lemma B.12: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ,* $\mathsf{Adv}^{(7)}_{\mathcal{H}}(\lambda) = \mathsf{Adv}^{(8)}_{\mathcal{H}}(\lambda)$.

Proof: In order to prove Lemma B.12, we note that the only difference between Hyb_7 and Hyb_8 is that for all $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, while generating the components $(\{\boldsymbol{k}'^{(h,j)}\}_{j\in[n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[m_h]})$ of the hth decryption key $\text{SK}(f_h)$ queried by $\mathcal{H}$ after making the ciphertext query, in place of using the shares $(\{\widetilde{\zeta}_h z_j + \widetilde{\sigma}_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{h,j'} x_{\rho_h(j')} + \widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ obtained by combining $(\{\widetilde{\sigma}_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{h,j'}, \widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$, outputted by $\mathsf{PGB}(f_h)$ using uniformly and independently (of the other variables) sampled randomness, with $(\vec{x}, \widetilde{\zeta}_h\vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$ the shares $(\{\nu_h\}_{h\in[n]}, \{\mu_{h,j'}\}_{j'\in[m_h]})$, outputted by $\mathsf{SIM}(f_h, \vec{x}, f_h(\vec{x}, \widetilde{\zeta}_h\vec{z}))$ with uniformly and independently (of the other variables) sampled randomness, are used as the coefficient of $(\{\boldsymbol{b}^{*(n'+j,4)}\}_{j\in[n]}, \{\boldsymbol{b}^{*(\rho_h(j'),4)}\}_{j'\in[m_h]})$ respectively, where $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$ is the pair of public-private attribute strings corresponding to the ciphertext query of $\mathcal{H}$ and $\widetilde{\zeta}_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$. Now, by the privacy property of PGB, as described in Section 2.3, it follows that the distribution of $(\{\widetilde{\zeta}_h z_j + \widetilde{\sigma}_{h,j}\}_{j\in[n]}, \{\widetilde{\alpha}_{h,j'} x_{\rho_h(j')} + \widetilde{\gamma}_{h,j'}\}_{j'\in[m_h]})$ and that of $(\{\nu_{h,j}\}_{j\in[n]}, \{\mu_{h,j'}\}_{j'\in[m_h]})$ are identical. Therefore, it follows that the distribution of the view of $\mathcal{H}$, i.e., the distribution $(\text{MPK}, \{\text{SK}(f_h)\}_{h\in[q_{\text{KEY}}]}, \text{CT})$ in Hyb_7 and that in Hyb_8 are identical. Hence the lemma follows. □

C Proof of Theorem 4.1

The proof of Theorem 4.1 is essentially an extension of that of Theorem 3.1. Here, we only provide a rough sketch of the proof highlighting the similarities and differences with that of Theorem 3.1. In the proof of Theorem 4.1, we consider the following two cases:

- **Case (1)**: $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 1$ for some $h \in [q_{\text{KEY-PRE}}]$, i.e., $\mathcal{H}$ makes at least one pre-ciphertext decryption key query corresponding to some function that satisfies the $R^{\text{ABP}\circ\text{IP}}$ relation with the public-private attribute pair associated with its ciphertext query.
- **Case (2)**: $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$ for all $h \in [q_{\text{KEY-PRE}}]$, i.e., none of the functions associated with the pre-ciphertext decryption key queries of $\mathcal{H}$ satisfies the $R^{\text{ABP}\circ\text{IP}}$ relation with the public-private attribute pair associated with its ciphertext query.

■ Description of the simulator

The simulator $\mathcal{S}$ is described below. The strategy of $\mathcal{S}$ differs in the two cases mentioned above. Although, $\mathcal{S}$'s strategy for simulating the pre-ciphertext decryption key queries is the same in both the cases since $\mathcal{S}$ realizes which case occurs after the ciphertext query is made by $\mathcal{H}$, and $\mathcal{S}$ obtains the oracle output on the whole pre-ciphertext decryption key queries of $\mathcal{H}$.

► **Case (1)**: In this case, the strategy of the simulator $\mathcal{S}$ is obtained by augmenting that of the simulator in the attribute-only case (Section 3.2) in essentially the same way the original KEM version of our PHPE scheme is obtained from the attribute-only version. The most crucial point to note here is that since in this case $\mathcal{H}$ makes at least one pre-ciphertext decryption key query whose corresponding function satisfies the $R^{\text{ABP}\circ\text{IP}}$ relation with the attribute pair $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^{n}$ associated with its ciphertext query, $\mathcal{S}$ is provided with a session key $\text{KEM} = g_T^{\xi} \in \mathbb{G}_T$ by the oracle $\mathcal{O}_{R^{\text{ABP}\circ\text{IP}}}$ at the time of simulating the queried ciphertext. Therefore, $\mathcal{S}$ simply uses $\xi \in \mathbb{F}_q$

while simulating the $\boldsymbol{c}'^{(0)}$ component of the queried ciphertext CT. Of course, for this $\mathcal{S}$ needs to perform a brute force discrete log computation to extract ξ from g_T^{ξ}, and it is the reason why $\mathcal{S}$ must run in super-polynomial time, since discrete log is computationally hard in $\mathbb{G}_T$. The simulator constructed by Wee for his PHPE construction [Wee17] also performs a similar discrete log computation to extract the exponent from the session key provided by the oracle, and consequently runs in super-polynomial time.

► **Case (2)**: In this case, the strategy of the simulator $\mathcal{S}$ is as follows:

- In order to generate the public parameters, $\mathcal{S}$ proceeds as follows:
  1. It first generates $(\mathsf{params}, \{\mathbb{B}_\imath, \mathbb{B}_\imath^*\}_{\imath\in[0,n'+n]}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\text{OB}}(n'+n, (6, \overbrace{9,\ldots,9}^{n'+n}))$.
  2. It sets $\widehat{\mathbb{B}}_0 = \{\boldsymbol{b}^{(0,1)}, \boldsymbol{b}^{(0,4)}, \boldsymbol{b}^{(0,6)}\}$.
  3. For $\imath \in [n'+n]$, it sets $\widehat{\mathbb{B}}_\imath = \{\boldsymbol{b}^{(\imath,1)}, \boldsymbol{b}^{(\imath,2)}, \boldsymbol{b}^{(\imath,9)}\}$.
  4. It outputs the public parameters $\text{MPK} = (\mathsf{params}, \{\widehat{\mathbb{B}}_\imath\}_{\imath\in[0,n'+n]})$.
- For $h \in [q_{\text{KEY-PRE}}]$, $\mathcal{S}$ simulates the h^{th} decryption key queried by $\mathcal{H}$ corresponding to some function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ as follows:
  1. At first, it generates $\big((\{\sigma_{h,j}\}_{j\in[n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big), \big((\{\widehat{\sigma}_{h,j}\}_{j\in[n]}, \{\widehat{\alpha}_{h,j'}, \widehat{\gamma}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$.
  2. Next, it samples $r_{h,0}, \widehat{r}_{h,0}, \widehat{r}'_{h,0}, \kappa_{h,0}, \zeta_h, \widehat{\zeta}_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
  $$\boldsymbol{k}'^{(h,0)} = (-r_{h,0}, 0, -\widehat{r}'_{h,0}, 1, \kappa_{h,0}, 0)_{\mathbb{B}_0^*}.$$
  3. Then, for $j' \in [m_h]$, it samples $b'_{h,j'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\kappa}'^{(h,j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\gamma_{h,j}^+ = \gamma_{h,j} + b'_{h,j'} r_{h,0}$, $\widehat{\gamma}_{h,j'}^+ = \widehat{\gamma}_{h,j'} + b'_{h,j'}\widehat{r}_{h,0}$, as well as
  $$\boldsymbol{k}'^{(h,j')} = ((\gamma_{h,j'}^+, \alpha_{h,j'}), \vec{0}^2, (\widehat{\gamma}_{h,j'}^+, \widehat{\alpha}_{h,j'}), \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}.$$
  4. Then, for $j \in [n]$, it samples $b_{h,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\vec{\kappa}^{(h,j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\sigma_{h,j}^+ = \sigma_{h,j} + b_{h,j} r_{h,0}$, $\widehat{\sigma}_{h,j}^+ = \widehat{\sigma}_{h,j} + b_{h,j}\widehat{r}_{h,0}$, as well as
  $$\boldsymbol{k}^{(h,j)} = ((\sigma_{h,j}^+, \zeta_h), \vec{0}^2, (\widehat{\sigma}_{h,j}^+, \widehat{\zeta}_h), \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$
  5. It outputs the decryption key $\text{SK}(f_h) = (f_h, \{b'_{h,j'}\}_{j'\in[m_h]}, \{b_{h,j}\}_{j\in[n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[0,m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$.
- When $\mathcal{H}$ queries a ciphertext for some pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$, $\mathcal{S}$ receives $\vec{x}$ and $(\overbrace{\perp,\ldots,\perp}^{q_{\text{KEY-PRE}}})$. Note that in this case, $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$ for all $h \in [q_{\text{KEY-PRE}}]$. It simulates the ciphertext as follows:
  1. At first, it samples $\vec{s} \xleftarrow{\mathsf{U}} S = \{\vec{s} \in \mathbb{F}_q^n | R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{s})) = 0\, \forall h \in [q_{\text{KEY-PRE}}]\}$ using the same technique as the simulator does in the attribute-only case.
  2. Then, it samples $\theta, \widetilde{\xi}, \varphi'_0 \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
  $$\boldsymbol{c}'^{(0)} = (\vec{0}^2, \theta, \widetilde{\xi}, 0, \varphi'_0)_{\mathbb{B}_0}.$$
  3. Next, for $\iota' \in [n']$, it samples $\varphi'_{\iota'} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
  $$\boldsymbol{c}'^{(\iota')} = (\vec{0}^4, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}}.$$

4. Then, for $\iota \in [n]$, it samples $\varphi_\iota \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{c}^{(\iota)} = (\vec{0}^4, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}}.$$
5. It outputs the ciphertext $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$.
6. It also samples $\boldsymbol{Z}'^{(\iota')} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, x_{\iota'})\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$ for $\iota' \in [n']$, $\boldsymbol{Z}^{(\iota)} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, s_\iota)\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$ for $\iota \in [n]$, and computes $\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^\intercal$ for $\iota' \in [n']$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^\intercal$ for $\iota \in [n]$. It uses these matrices in the next query phase.

- For $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, in response to the h^{th} decryption key query of $\mathcal{H}$ corresponding to some function $f_h \in \mathcal{F}^{(q,n',n)}_{\text{ABP}\circ\text{IP}}$, $\mathcal{S}$ executes the following steps:

1. It first generates $\big((\{\sigma_{h,j}\}_{j \in [n]}, \{\alpha_{h,j'}, \gamma_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{PGB}(f_h)$.
2. After that, it samples $\zeta_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$.
3. Next, it queries its oracle $\mathcal{O}_{R^{\text{ABP}\circ\text{IP}}}$ with the function f_h, and receives back either a session key $\text{KEM} = g_T^{\xi} \in \mathbb{G}_T$ for some $\xi \xleftarrow{\mathsf{U}} \mathbb{F}_q$, or $\perp$, depending on whether $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 1$ or 0. If it receives $\text{KEM} = g_T^{\xi} \in \mathbb{G}_T$, i.e., $R^{\text{ABP}\circ\text{IP}}(f_h(\vec{x}, \vec{z})) = 1$, or in other words, $f_h(\vec{x}, \vec{z}) = 0$, it forms $\big((\{\widehat{\nu}_{h,j}\}_{j \in [n]}, \{\widehat{\mu}_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, 0)$. It also determines $\xi \in \mathbb{F}_q$ from the obtained $\text{KEM} = g_T^{\xi}$ by solving a brute force discrete log. On the other hand, if it obtains $\perp$, i.e., $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$, or in other words, $f_h(\vec{x}, \vec{z}) \neq 0$, then it samples $\check{\zeta}_h \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and generates $\big((\{\widehat{\nu}_{h,j}\}_{j \in [n]}, \{\widehat{\mu}_{h,j'}\}_{j' \in [m_h]}), \rho_h : [m_h] \to [n']\big) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, \check{\zeta}_h)$.
4. After that, it samples $r_{h,0}, \widehat{r}_{h,0}, \widehat{r}'_{h,0}, \kappa'_{h,0} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, and computes
$$\boldsymbol{k}'^{(h,0)} = \begin{cases} (-r_{h,0}, 0, -\widehat{r}_{h,0} + \theta^{-1}(\xi - \widetilde{\xi}), 1, \kappa'_{h,0}, 0)_{\mathbb{B}_0^*} \text{ if } R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 1, \\ (-r_{h,0}, 0, -\widehat{r}'_{h,0}, 1, \kappa'_{h,0}, 0)_{\mathbb{B}_0^*} \text{ if } R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0, \end{cases}$$
where $\theta, \widetilde{\xi}$ are the same variables used while simulating the ciphertext CT. Note that $\mathcal{S}$ can determine the quantity $\theta^{-1}(\xi - \widetilde{\xi}) \in \mathbb{F}_q$ with all but negligible probability $1/q$, i.e., except when $\theta = 0$.
5. Next, for $j' \in [m_h]$, it samples $b'_{h,j'}, \widehat{\eta}'_{h,j'} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \vec{\kappa}'^{(h,j')} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\gamma^{+}_{h,j'} = \gamma_{h,j'} + b'_{h,j'} r_{h,0}$, $\widehat{\mu}^{+}_{h,j'} = \widehat{\mu}_{h,j'} + b'_{h,j'} \widehat{r}_{h,0}$, as well as
$$\boldsymbol{k}'^{(h,j')} = ((\gamma^{+}_{h,j'}, \alpha_{h,j'}), \vec{0}^2, (\widehat{\eta}'_{h,j'}, \widehat{\mu}^{+}_{h,j'})(\boldsymbol{U}'^{(\rho_h(j'))})^{-1}, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}}.$$
6. Then, for $j \in [n]$, it samples $b_{h,j}, \widehat{\eta}_{h,j} \xleftarrow{\mathsf{U}} \mathbb{F}_q, \vec{\kappa}^{(h,j)} \xleftarrow{\mathsf{U}} \mathbb{F}_q^2$, and computes $\sigma^{+}_{h,j} = \sigma_{h,j} + b_{h,j} r_{h,0}$, $\widehat{\nu}^{+}_{h,j} = \widehat{\nu}_{h,j} + b_{h,j} \widehat{r}_{h,0}$, as well as
$$\boldsymbol{k}^{(h,j)} = ((\sigma^{+}_{h,j}, \zeta_h), \vec{0}^2, (\widehat{\eta}_{h,j}, \widehat{\nu}_{h,j})(\boldsymbol{U}^{(j)})^{-1}, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}}.$$
7. It outputs the decryption key $\text{SK}(f_h) = (f_h, \{b'_{h,j'}\}_{j' \in [m_h]}, \{b_{h,j}\}_{j \in [n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [0, m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$.

■ **Sequence of Hybrid Experiments**

► **Case (1)**: In this case, we design a hybrid sequence Hyb_0 through Hyb_8, which is essentially the same as the one used in the attribute-only case (Section 3.2), except for the additional components of the queried ciphertext and decryption keys which are also altered in a similar fashion to the other components during the hybrid transitions. Since, the simulation strategy in this case is essentially the same as that in the attribute-only case, other than the additional components of the ciphertext and decryption keys that are simulated in a similar fashion to the other components, it is immediate that Hyb_8 coincides with the $\mathsf{Exp}^{\mathsf{PHPE,IDEAL}}_{\mathcal{H},\mathcal{S}}(\lambda)$ in this case.

► **Case (2)**: In this case also, we perform the hybrid transitions from Hyb_0 through Hyb_8 in the same manner to Case (1). However, in this case, we need to execute some additional hybrid transitions, which we describe below. As earlier, in the description of these hybrids also, a part framed by a box indicates coefficients that are altered in a transition from its previous hybrid.

Hyb_9: This experiment is the same as Hyb_8 except that in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x},\vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is created as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that

$$\begin{aligned}\boldsymbol{c}'^{(0)} &= (0, \tau, \theta, \xi, 0, \varphi'_0)_{\mathbb{B}_0},\\ \boldsymbol{c}'^{(\iota')} &= (\vec{0}^2, \boxed{\tau(1, x_{\iota'})}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],\\ \boldsymbol{c}^{(\iota)} &= (\vec{0}^2, \boxed{\tau(1, s_\iota)}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],\end{aligned} \tag{C.1}$$

while for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, the hth decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is generated as $\text{SK}(f_h) = (f_h, \{b'_{h,j'}\}_{j'\in[m_h]}, \{b_{h,j}\}_{j\in[n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[0,m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that

$$\begin{aligned}\boldsymbol{k}'^{(h,0)} &= (-r_{h,0}, -\widetilde{r}_{h,0}, 0, 1, \kappa'_{h,0}, 0)_{\mathbb{B}^*_0},\\ \boldsymbol{k}'^{(h,j')} &= ((\gamma^+_{h,j'}, \alpha_{h,j'}), \boxed{(\eta'_{h,j'}, \mu^+_{h,j'})(\boldsymbol{U}'^{(\rho_h(j'))})^{-1}}, \vec{0}^2, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\sigma^+_{h,j}, \zeta_h), \boxed{(\eta_{h,j}, \nu^+_{h,j})(\boldsymbol{U}^{(j)})^{-1}}, \vec{0}^2, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],\end{aligned} \tag{C.2}$$

where $\boldsymbol{Z}'^{(\iota')} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, x_{\iota'})\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$, $\boldsymbol{U}'^{(\iota')} = ((\boldsymbol{Z}'^{(\iota')})^{-1})^{\intercal}$, for $\iota' \in [n']$, $\boldsymbol{Z}^{(\iota)} \xleftarrow{\mathsf{U}} \{\boldsymbol{Z} \in \mathsf{GL}(2, \mathbb{F}_q) | (1, s_\iota)\boldsymbol{Z} = \vec{e}^{(2)} = (0,1)\}$, $\boldsymbol{U}^{(\iota)} = ((\boldsymbol{Z}^{(\iota)})^{-1})^{\intercal}$ for $\iota \in [n]$, and all the other variables are generated as in Hyb_8.

Hyb_{10}: This experiment is analogous to Hyb_9 except that in this experiment, for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, the hth decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}_{\text{ABP}\circ\text{IP}}^{(q,n',n)}$ is generated as $\text{SK}(f_h) = (f_h, \{b'_{h,j'}\}_{j'\in[m_h]}, \{b_{h,j}\}_{j\in[n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j'\in[0,m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j\in[n]})$ such that

$$\begin{aligned}\boldsymbol{k}'^{(h,0)} &= (-r_{h,0}, \boxed{0, -\widehat{r}_{h,0}}, 1, \kappa'_{h,0}, 0)_{\mathbb{B}^*_0},\\ \boldsymbol{k}'^{(h,j')} &= ((\gamma^+_{h,j'}, \alpha_{h,j'}), \boxed{\vec{0}^2, (\widehat{\eta}'_{h,j'}, \widehat{\mu}^+_{h,j'})(\boldsymbol{U}'^{(\rho_h(j'))})^{-1}}, \vec{\kappa}'^{(h,j')}, 0)_{\mathbb{B}^*_{\rho_h(j')}} \text{ for } j' \in [m_h],\\ \boldsymbol{k}^{(h,j)} &= ((\sigma^+_{h,j}, \zeta_h), \boxed{\vec{0}^2, (\widehat{\eta}_{h,j}, \widehat{\nu}^+_{h,j})(\boldsymbol{U}^{(j)})^{-1}}, \vec{\kappa}^{(h,j)}, 0)_{\mathbb{B}^*_{n'+j}} \text{ for } j \in [n],\end{aligned} \tag{C.3}$$

where $\{\widehat{r}_{h,0}\}_{h\in[q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]}, \{\widehat{\eta}'_{h,j'}\}_{h\in[q_{\text{KEY-PRE}}+1,q_{\text{KEY}}], j'\in[m_h]}, \{\widehat{\eta}_{h,j}\}_{h\in[q_{\text{KEY-PRE}}+1,q_{\text{KEY}}], j\in[n]}$, $\{\widehat{\zeta}_h\}_{h\in[q_{\text{KEY-PRE}}+1,q_{\text{KEY}}]} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $((\{\widehat{\nu}_{h,j}\}_{j\in[n]}, \{\widehat{\mu}_{h,j'}\}_{j'\in[m_h]}), \rho_h : [m_h] \to [n']) \xleftarrow{\mathsf{R}} \mathsf{SIM}(f_h, \vec{x}, f_h(\vec{x}, \widehat{\zeta}_h \vec{z}))$ for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, $\widehat{\mu}^+_{h,j'} = \widehat{\mu}_{h,j'} + b'_{h,j'}\widehat{r}_{h,0}$ for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], j' \in [m_h]$, $\widehat{\nu}^+_{h,j} = \widehat{\nu}_{h,j} + b_{h,j}\widehat{r}_{h,0}$ for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}], j \in [n]$, and all the other variables are generated as in Hyb_9.

Hyb_{11}: This experiment is the same as Hyb_{10} except that in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x},\vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is created as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota'\in[0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota\in[n]})$ such that

$$\begin{aligned}\boldsymbol{c}'^{(0)} &= (0, \boxed{0}, \theta, \xi, 0, \varphi'_0)_{\mathbb{B}_0},\\ \boldsymbol{c}'^{(\iota')} &= (\vec{0}^2, \boxed{\vec{0}^2}, \theta(1, x_{\iota'}), \vec{0}^2, \varphi'_{\iota'})_{\mathbb{B}_{\iota'}} \text{ for } \iota' \in [n'],\\ \boldsymbol{c}^{(\iota)} &= (\vec{0}^2, \boxed{\vec{0}^2}, \theta(1, s_\iota), \vec{0}^2, \varphi_\iota)_{\mathbb{B}_{n'+\iota}} \text{ for } \iota \in [n],\end{aligned} \tag{C.4}$$

where all the variables are generated as in Hyb_{10}.

Hyb_{12}: This experiment is identical to Hyb_{11} except that in this experiment, the ciphertext queried by $\mathcal{H}$ corresponding to the pair of public-private attribute strings $(\vec{x}, \vec{z}) \in \mathbb{F}_q^{n'} \times \mathbb{F}_q^n$ is created as $\text{CT} = (\vec{x}, \{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [0,n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]})$ such that $\{\boldsymbol{c}'^{(\iota')}\}_{\iota' \in [n']}, \{\boldsymbol{c}^{(\iota)}\}_{\iota \in [n]}$ are given by Eq. (C.4), and

$$\boldsymbol{c}'^{(0)} = (0, 0, \theta, \boxed{\widetilde{\xi}}, 0, \varphi'_0)_{\mathbb{B}_0}, \tag{C.5}$$

while for $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$, the h^{th} decryption key queried by $\mathcal{H}$ corresponding to the function $f_h \in \mathcal{F}_{\text{ABPoIP}}^{(q,n',n)}$ is generated as $\text{SK}(f_h) = (f_h, \{b'_{h,j'}\}_{j' \in [m_h]}, \{b_{h,j}\}_{j \in [n]}, \{\boldsymbol{k}'^{(h,j')}\}_{j' \in [0,m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]})$ such that $\{\boldsymbol{k}'^{(h,j')}\}_{j' \in [m_h]}, \{\boldsymbol{k}^{(h,j)}\}_{j \in [n]}$ are given by Eq. (C.3), and

$$\boldsymbol{k}'^{(h,0)} = \begin{cases} (-r_{h,0}, 0, \boxed{\widehat{r}'_{h,0}}, 1, \kappa'_{h,0}, 0)_{\mathbb{B}_0^*} \text{ if } R^{\text{ABPoIP}}(f_h, (\vec{x}, \vec{z})) = 0, \\ (-r_{h,0}, 0, \boxed{-\widehat{r}_{h,0} + \theta^{-1}(\xi - \widetilde{\xi})}, 1, \kappa'_{h,0}, 0)_{\mathbb{B}_0^*} \text{ if } R^{\text{ABPoIP}}(f_h, (\vec{x}, \vec{z})) = 1, \end{cases} \tag{C.6}$$

where $\widetilde{\xi} \xleftarrow{\mathsf{U}} \mathbb{F}_q$, $\widehat{r}'_{h,0} \xleftarrow{\mathsf{U}} \mathbb{F}_q$ for $h \in [q_{\text{KEY-PRE}} + 1, q_{\text{KEY}}]$ with $R^{\text{ABPoIP}}(f_h, (\vec{x}, \vec{z})) = 0$, and all the other variables are generated as in Hyb_{10}. Observe that this experiment coincides with the experiment $\mathsf{Exp}_{\mathcal{H},\mathcal{S}}^{\mathsf{PHPE,IDEAL}}(\lambda)$ with the simulator $\mathcal{S}$ described above.

■ Analysis

► **Case (1)**: Since, the hybrids are similar to that in the attribute-only case, essentially the same analysis apply. The most important difference is the analysis of the transition from $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ ($\chi \in [q_{\text{KEY-PRE}}]$), i.e., the central information-theoretic transition for the pre-ciphertext decryption keys (i.e., to prove a lemma similar to Lemma B.4). Now, we will consider an augmented matrix

$$\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z}) = \begin{pmatrix} -1 & 0 & \dots & 0 \\ b'_{\chi,1} & & & \\ \vdots & & & \\ b'_{\chi,m_\chi} & & \boldsymbol{L}^{(\chi)}(\vec{x}, \vec{z}) & \\ b_{\chi,1} & & & \\ \vdots & & & \\ b_{\chi,n} & & & \end{pmatrix} \in \mathbb{F}_q^{(m_\chi+n+1) \times (m_\chi+n+1)},$$

where $\boldsymbol{L}^{(\chi)} \in \mathbb{F}_q^{(m_\chi+n) \times (m_\chi+n)}$ is the matrix representation of the ABP Γ'_χ computing the function $f_\chi \in \mathcal{F}_{\text{ABPoIP}}^{(q,n',n)}$ queried by $\mathcal{H}$, as computed by PGB described in Section 2.3, and $(\{b'_{\chi,j'}\}_{j' \in [m_\chi]}, \{b_{\chi,j}\}_{j \in [n]}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_\chi+n}$ are the random coefficients selected during the generation of the decryption key $\text{SK}(f_\chi)$. We observe the following two facts:

(I) ($R^{\text{ABPoIP}}(f_\chi, (\vec{x}, \vec{z})) = 0$, i.e., $f_\chi(\vec{x}, \vec{z}) \neq 0$) As already noted in the proof of Claim B.5, from Lemma 2.1, we have $\det(\boldsymbol{L}^{(\chi)}(\vec{x}, \vec{z})) = f_\chi(\vec{x}, \vec{z})$. Hence, we have $\det(\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})) = -\det(\boldsymbol{L}^{(\chi)}(\vec{x}, \vec{z})) = -f_\chi(\vec{x}, \vec{z})$. Since $f_\chi(\vec{x}, \vec{z}) \neq 0$ in this case, it follows that $\det(\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})) \neq 0$, or in other words, the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})$ is invertible in this case. Therefore, the image of the linear transformation defined by the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})$ is $\mathbb{F}_q^{m_\chi+n+1}$.

(II) ($R^{\text{ABPoIP}}(f_\chi, (\vec{x}, \vec{z})) = 1$, i.e., $f_\chi(\vec{x}, \vec{z}) = 0$) In this case, the rank of the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})$ is $m_\chi + n$. In order to see this, first notice that the rank of the matrix $\boldsymbol{L}^{(\chi)}(\vec{x}, \vec{z})$ is $m_\chi + n - 1$ in this case, as argued in the proof of Claim B.5. Therefore, the sub-matrix of order $(m_\chi + n + 1) \times (m_\chi + n)$ of the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})$ obtained by removing its first column, i.e., the sub-matrix $\begin{pmatrix} 0 & \dots & 0 \\ & & \\ & \boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}) & \\ & & \end{pmatrix}$ clearly has rank $m_\chi + n - 1$. Moreover, the first column of the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})$, namely, the column $(-1, b'_{\chi,1}, \dots, b'_{\chi,m_\chi}, b_{\chi,1}, \dots, b_{\chi,n})^\intercal$ is linearly independent of all the other columns of the matrix, since the first entry of this column is -1, whereas those of the other columns of the matrix are all 0's. Therefore, it follows that the rank of the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})$ is $m_\chi + n$.

Also, in this case, it holds that $((\Omega'_{\chi,j'})_{j'\in[0,m_\chi]}, (\Omega_{\chi,j})_{j\in[n]})\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z}) = \vec{0}^{m_\chi+n+1}$, where $(\{\Omega_{\chi,j}\}_{j\in[n]}, \{\Omega'_{\chi,j'}\}_{j'\in[m_\chi]}) = \mathsf{REC}(f_\chi, \vec{x})$ and $\Omega'_{\chi,0} = \sum\limits_{j'\in[m_\chi]} \Omega'_{\chi,j'} b'_{\chi,j'} + \sum\limits_{j\in[n]} \Omega_{\chi,j} b_{\chi,j}$. To see this, first note that $((\Omega'_{\chi,j'})_{j'\in[m_\chi]}, (\Omega_{\chi,j})_{j\in[n]})\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}) = \vec{0}^{m_\chi+n}$ holds in this case, as already explained in the proof of Claim B.5. Hence, we clearly have

$$((\Omega'_{\chi,j'})_{j'\in[0,m_\chi]}, (\Omega_{\chi,j})_{j\in[n]}) \begin{pmatrix} 0 & \dots & 0 \\ & & \\ & \boldsymbol{L}^{(\chi)}(\vec{x},\vec{z}) & \\ & & \end{pmatrix} = \vec{0}^{m_\chi+n}.$$

Also, from the expression of $\Omega'_{\chi,0}$, it is clear that $((\Omega'_{\chi,j'})_{j'\in[0,m_\chi]}, (\Omega_{\chi,j})_{j\in[n]})(-1, (b'_{\chi,j'})_{j'\in[m_\chi]}, (b_{\chi,j})_{j\in[n]})^\intercal = 0$. Therefore, it follows that $((\Omega'_{\chi,j'})_{j'\in[0,m_\chi]}, (\Omega_{\chi,j})_{j\in[n]})\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z}) = \vec{0}^{m_\chi+n+1}$.

From the above two observations, it follows that the image of the linear transformation defined by the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})$ is given by the $(m_\chi+n)$-dimensional subspace $\mathsf{Im}(\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})) = \{\vec{v} = (v_1, \dots, v_{m_\chi+n+1}) \in \mathbb{F}_q^{m_\chi+n+1} \mid ((\Omega'_{\chi,j'})_{j'\in[0,m_\chi]}, (\Omega_{\chi,j})_{j\in[n]}) \cdot \vec{v} = \sum\limits_{j'\in[0,m_\chi]} \Omega'_{\chi,j'} v_{j'+1} + \sum\limits_{j\in[n]} \Omega_{\chi,j} v_{n'+j+1} = 0\} \subset \mathbb{F}_q^{m_\chi+n+1}$.

Thus, it is clear that the augmented matrix $\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})$ satisfies similar properties as the matrix $\boldsymbol{L}^{(\chi)}(\vec{x},\vec{z})$. Hence, we can argue the transition from $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ in the same manner as is done in the proof of Lemma B.4.

► **Case (2)**: As already mentioned, the hybrid transitions from Hyb_0 to Hyb_8 in this case is similar to that in Case (1). So, the analysis of those transitions would also be similar to Case (1). In particular, we will make use of the augmented matrix $\boldsymbol{L}^{+(\chi)}(\vec{x},\vec{z})$ for arguing the transition from $\mathsf{Hyb}_{2\text{-}\chi\text{-}2}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ in this case as well. We now turn to analysing the additional hybrid transitions in this case, namely, the hybrid transition from Hyb_8 to Hyb_{12}.

Lemma C.1: *For any probabilistic adversary $\mathcal{H}$, for any security parameter λ, $\mathsf{Adv}^{(8)}_{\mathcal{H}}(\lambda) = \mathsf{Adv}^{(9)}_{\mathcal{H}}(\lambda)$.*

Proof: The transition from Hyb_8 to Hyb_9 is essentially the reverse of the transition from Hyb_5 to Hyb_6. Therefore, Lemma C.1 can be proven by applying a similar information-theoretic basis transformation as used in the proof of Lemma B.10. □

Lemma C.2: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_6$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}^{(9)}_{\mathcal{H}}(\lambda) - \mathsf{Adv}^{(10)}_{\mathcal{H}}(\lambda)| \le \mathsf{Adv}^{\mathsf{P3}}_{\mathcal{B}_6}(\lambda) + 2/q$.*

Proof: Observe that the transition from Hyb_9 to Hyb_{10} is similar to that from $\mathsf{Hyb}_{2\text{-}\chi\text{-}3}$ to $\mathsf{Hyb}_{2\text{-}\chi\text{-}4}$, where we move some coefficients of the decryption key components from the second 2-dimensional block to the third 2-dimensional block. Hence, the proof of Lemma C.2 is analogous to that of Lemma B.6. □

Lemma C.3: *For any stateful probabilistic adversary $\mathcal{H}$, there exists a probabilistic algorithm $\mathcal{B}_7$, whose running time is essentially the same as that of $\mathcal{H}$, such that for any security parameter λ, $|\mathsf{Adv}_{\mathcal{H}}^{(10)}(\lambda) - \mathsf{Adv}_{\mathcal{H}}^{(11)}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_7}^{\mathsf{P1}}(\lambda) + 2/q$.*

Proof: Observe that the transition from Hyb_{10} to Hyb_{11} is essentially similar to the reverse of the transition from Hyb_0 to $\mathsf{Hyb}_{2\text{-}1\text{-}1}$. Hence, Lemma C.3 can be proven by an argument that amounts to a combination of the arguments used in the proofs of Lemma B.2 and Lemma B.1. □

Lemma C.4: *For any stateful probabilistic adversary $\mathcal{H}$, for any security parameter λ, $\mathsf{Adv}_{\mathcal{H}}^{(11)}(\lambda) = \mathsf{Adv}_{\mathcal{H}}^{(12)}(\lambda)$.*

Proof: Lemma C.4 can be proven by applying an information-theoretic linear transformation between the third and fourth vectors of the basis $(\mathbb{B}_0, \mathbb{B}_0^*)$, something similar to the one used in the proof of Lemma B.10. The fact that in Hyb_{12} for $h \in [q_{\text{KEY-PRE}}+1, q_{\text{KEY}}]$, the coefficient $\widehat{r}'_{h,0}$ of the component $\boldsymbol{k}'^{(h,0)}$ of the decryption key $\text{SK}(f_h)$ is uniformly and independently (of the other variables) distributed in $\mathbb{F}_q$ in case $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$, can be proven by a similar argument as used in the proof of Lemma B.4 by observing that the matrix

$$\widehat{\boldsymbol{L}}^{+(h)}(\vec{x}, \vec{z}) = \begin{pmatrix} -1 & 0 & \dots & 0 \\ b'_{h,1} & & & \\ \vdots & & & \\ b'_{h,m_h} & & \widehat{\boldsymbol{L}}(\vec{x}, \vec{z}) & \\ b_{h,1} & & & \\ \vdots & & & \\ b_{h,n} & & & \end{pmatrix},$$

where $\widehat{\boldsymbol{L}}^{(h)}(\vec{x}, \vec{z})$ is the matrix used by the PGB-simulator SIM to generate the constants $(\{\widehat{\nu}_{h,j}\}_{j\in[n]}, \{\widehat{\mu}_{h,j'}\}_{j\in[m_h]})$, and $(\{b'_{h,j'}\}_{j'\in[m_h]}, \{b_{h,j}\}_{j\in[n]}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{m_h+n}$ are the random coefficients sampled while generating the decryption key $\text{SK}(f_h)$, is invertible if $R^{\text{ABP}\circ\text{IP}}(f_h, (\vec{x}, \vec{z})) = 0$, by a similar logic as used for the matrix $\boldsymbol{L}^{+(\chi)}(\vec{x}, \vec{z})$ above. □