Splitting Bilinear Groups: New Translations from Composite- to Prime-Order with Applications to Batch Arguments for NP

David Balbás[1], Dario Fiore[2], and Duy Nguyen[3]

[1] ETH Zurich, Zurich, Switzerland
dbalbas@ethz.ch
[2] IMDEA Software Institute, Madrid, Spain
dario.fiore@imdea.org
[3] Telecom Paris, Institut Polytechnique de Paris, Palaiseau, France
dinh.nguyen@telecom-paris.fr

Abstract. Bilinear groups, also known as pairing groups, are a versatile tool that enables many efficient cryptographic constructions. Among bilinear groups, those with a composite order ($N = p \cdot q$ for two large, secret primes p, q) offer an additional algebraic structure which is advantageous in many applications. They are however dramatically less efficient than their prime-order counterparts, so multiple translation frameworks for constructions from composite- to prime-order groups have been introduced in the literature.

Motivated by the recent construction of Batch Arguments for NP (BARGs) with linear-size CRS by Chen, Elias and Wu [Asiacrypt '25], based on composite-order groups, we notice that these previous frameworks fail to transfer their scheme to the prime-order setting. In this work, we identify and close this gap by introducing a new translation framework based on a new abstraction called *dual encodings*. The crucial feature of these objects is a security property called indistinguishability that allows embedding hidden subgroups, emulating composite-order groups more faithfully and thus enabling more powerful translations. Then, we realize dual encodings using functional encryption for function-hiding inner products. Finally, we apply our framework to build the first BARG with linear-size CRS from prime-order groups, while preserving the (statistical) somewhere extractability of previous constructions. Our result represents a significant step toward practically efficient BARGs and showcases a surprising application of functional encryption which we find of independent interest.

1 Introduction

Bilinear groups $(\mathfrak{G}_1, \mathfrak{G}_2, \mathfrak{G}_T)$ are tuples of abelian groups equipped with a non-degenerate and efficiently computable bilinear map (also known as pairing) $e : \mathfrak{G}_1 \times \mathfrak{G}_2 \to \mathfrak{G}_T$. The power of these groups in cryptography stems from the ability to compute quadratic operations over encoded elements without revealing the underlying scalars. For instance, given generators $\mathfrak{g}_1 \in \mathfrak{G}_1, \mathfrak{g}_2 \in \mathfrak{G}_2$ and encodings $(A = \mathfrak{g}_1^a, B = \mathfrak{g}_2^b)$ of scalars a and b, one can compute $e(A, B) = e(\mathfrak{g}_1, \mathfrak{g}_2)^{ab}$. Beyond this homomorphic-like property, cryptographic constructions rely on various computational assumptions over these groups, typically rooted in the hardness of computing discrete logarithms.

Since their introduction to the landscape of cryptographic constructions [Jou04], bilinear groups have emerged as an extraordinarily versatile tool, providing solutions to countless cryptographic problems. Combined with the availability of very efficient constructions of these groups, pairing-based cryptographic solutions often excel in efficiency. While for several primitives pairing-based solutions are just a more performant option, for some cryptographic schemes the efficiency gap is huge as all known alternatives to pairings are extremely impractical.

A notable example of this efficiency gap – and the primary application of this work – are *Batch Arguments for NP* (BARGs) [KPY19, CJJ21]. In this domain, the pairing-based setting currently

A short version of this paper appears in the proceedings of CRYPTO 2026. This is the full version.

offers the only viable path toward practical efficiency. Alternative approaches build BARGs via the non-black-box combinations of cryptographic primitives, such as probabilistically checkable proofs and correlation-intractable hash functions [CGH98, CCH+19, PS19, JJ21], or indistinguishability obfuscation [BGI+01]. These approaches suffer from large efficiency overheads, i.e. the poly functions in proof size and prover/verifier complexity are often prohibitively large or not explicitly defined. In contrast, pairing-based BARGs [WW22, GLWW24, CEW25] offer a clean algebraic framework with measurable efficiency in terms of group elements and operations.

Pairing-based BARGs. BARGs are non-interactive proof systems that can prove a collection of NP statements $\mathtt{x}_1, \ldots, \mathtt{x}_m$, i.e., that $\mathtt{x}_i \in \mathcal{L}\ \forall i = 1, \ldots, m$, with a proof that scales *sublinearly in the number of statements*, e.g., of size $\mathsf{poly}(\lambda, \log m)$. While they share similarities with SNARGs, BARGs are distinguished by three properties. (1. Relaxed succinctness) Unlike SNARGs, proofs are allowed to grow linearly with the size of a single witness. (2. Somewhere extractability) A security notion tailored to the batch setting which demands for the black-box extraction of the witness $\mathtt{w}_{i^*}$ of the i^*-th statement of the batch. Here, the index i^* is chosen at the time of generating the common reference string (CRS) and is computationally hidden to the adversary. (3. Falsifiable assumptions) Due to the above relaxations of succinctness and security, BARGs evade the known impossibility result for SNARKs [GW11] and are realizable from standard falsifiable assumptions. All these properties make BARGs an appealing building block to realize many cryptographic primitives from standard assumptions; current applications include (but are not limited to) non-interactive delegation [BHK17, KPY19, KVZ21, CJJ22], incrementally verifiable computation [PP22, DGKV22, DJJ+25], aggregate signatures [WW22, BCJP24], and homomorphic signatures [ABF24, Goy24, ACG24].

Despite the advantages of pairings, current pairing-based BARGs are still far from being practical. Until recently, these schemes required a common reference string (CRS) whose size scales quadratically [WW22], or superlinearly [GLWW24] with the number of statements (i.e., having $O(m^2)$ or $O(m^{1+o(1)})$ group elements respectively). This barrier has been recently overcome by the work of Chen, Elias, and Wu [CEW25], who proposed the first pairing-based BARG with an $O(m)$-size CRS. To obtain this result, [CEW25] employs a structured CRS similar to the one of polynomial commitments [KZG10] with the novel twist of using elements from *composite-order bilinear groups*. Unfortunately, these groups are much more costly than their prime-order counterparts, as their order $N = p \cdot q$ must be large enough so that factoring is hard. This leads to group and pairing operations that are significantly, e.g., two orders of magnitude, slower [Gui13]. This situation leads to the first main question addressed in this work:

Can we construct BARGs for NP that have a linear-size CRS and work over prime-order bilinear groups?

At a first glance, this question does not appear difficult to settle, especially given the existing line of works on translating cryptographic constructions from composite- to prime-order groups [Fre10, Lew12, LM15, SC12, HHH+14]. For instance, Waters and Wu used this approach to design a prime-order version of their BARG with quadratic-size CRS [WW22]. Similar translations do not unfortunately work for the BARG of [CEW25]. While the nuances of this obstacle are rather technical and thus detailed in Section 2, here we give a high-level overview of the core difficulty, which lies in the structure of the CRS and in the underlying security assumption. The linear-size CRS in [CEW25] introduces terms in the source groups that are correlated.[4] These correlations

[4] More in detail, their proof of somewhere extractability relies on the m-subgroup decision exponent assumption, which is a generalization of the classical subgroup decision that gives similarly correlated terms to the adversary.

are publicly computable with the pairing and leak information about the secret scalars encoded in the CRS. In composite-order groups, this leakage is benign, and the assumption holds as shown in [CEW25].

Attempting to translate such a CRS to prime-order groups using existing frameworks [Fre10, Lew12, LM15, SC12, HHH+14] unfortunately fails. In these frameworks, each element from a composite-order group is emulated via a vector of prime-order group elements. Then, each pairing in the composite-order group is emulated via an inner-product pairing in the prime-order group. Although the inner product relations remain harmless for indistinguishability, the transition to a vector space allows the adversary to additionally test *outer product* relations. These outer products, which are not possible in the original composite-order setting, lead to an attack against the translated assumption. This state of affairs raises the second main question addressed by this paper:

Can we construct a translation framework that, using prime-order groups, ensures that the leakage induced by pairing relations is the same as for composite-order groups?

1.1 Our Contribution

In this paper, we answer the aforementioned questions in the affirmative. We present a new framework for translating cryptographic constructions from composite- to prime-order bilinear groups and apply it to realize the first pairing-based BARG with a linear-size CRS in the prime-order setting. We present our contributions in more detail below.

Splitting bilinear groups. We introduce *splitting bilinear groups*, a novel translation framework that emulates relevant properties of composite-order groups using prime-order groups. Splitting bilinear groups provide the same functionalities as prior translation frameworks, namely each (emulated) group $\mathfrak{G}_1, \mathfrak{G}_2, \mathfrak{G}_T$ is isomorphic to a product of prime-order groups, i.e., $\mathfrak{G}_1 \simeq \prod_i \mathfrak{G}_{1,i}$ and analogously for $\mathfrak{G}_2, \mathfrak{G}_T$. This means that one can encode vectors of scalars in these groups. As new feature, splitting bilinear groups provide an indistinguishability security property, which ensures that emulated group elements only leak inner product relations about the encoded vectors of scalars, and that no additional leakage (e.g. by outer product relations) is meaningful. Notably, this indistinguishability property goes beyond what is guaranteed by prior frameworks [Fre10, Lew12, LM15, SC12, HHH+14].

The primary innovation of our construction is to construct these emulated groups via the use of *linearly-homomorphic functional encryption for function-hiding inner products* (FH-IPFE). Intuitively, FH-IPFE uses a secret key to encode vectors of scalars $\boldsymbol{x}$ and $\boldsymbol{y}$ in ciphertexts and functional decryption keys respectively, from the two source groups, in such a way that the (functional) decryption process reveals only their inner product in the target group.

In our construction, we define group elements as FH-IPFE encodings of vectors where each coordinate corresponds to a different subgroup. Then, we use the linearly-homomorphic property of FH-IPFE to ensure that the group operation between encodings is well-defined. This layer of encryption offered by FH-IPFE, and its IND-CPA notion, allows us to precisely control the leakage of information about the encoded vectors that should be available to an adversary.[5]

[5] We remark that IND-CPA secure FH-IPFE can be realized from standard assumptions such as DLIN [TAO16], or SXDH [Lin17].

A BARG with linear-size CRS in prime-order groups. We abstract away the BARG of [CEW25] and describe it within the framework for splitting bilinear groups. In short, splitting enables us to generate a CRS in extractable mode with a trapdoor, such that the extractor can "project" a hidden component that we use to define the black-box extraction, as we explain in Section 2. Then, the indistinguishability property of our encoding enables us to argue that the CRS in extractable mode is computationally indistinguishable to the normal mode.

In this part, our main technical contribution is to translate the m-subgroup decision exponent assumption [CEW25] from composite-order groups to splitting bilinear groups. In particular, we use our framework to show that the translated assumption holds based on the IND-CPA security of the FE scheme and on a new m-type assumption (called Bilateral Diffie-Hellman Decision Exponent, or BDHDE in short) that we justify in the generic bilinear group model.

As shown in Table 1, where we compare existing algebraic BARGs, our construction is the first that achieves a linear-size CRS over prime-order groups, while preserving a proof size that is 1) constant in the number of statements k, and 2) linear in the size of the circuit $\mathcal{C}$ that decides the NP relation. Thus, our construction is a significant step forward towards efficient realizations of BARGs.

Scheme	$\lvert\mathsf{crs}\rvert$	$\lvert\pi\rvert$	**Prime-Order**	**Assumption**
WW22 [WW22]	m^2	$\ell_{\mathcal{C}}$	✓	d-Lin
WW22 [WW22]	m^2	$\ell_{\mathcal{C}}$	–	Subgroup Decision
GLWW24 [GLWW24]	$m^{1+o(1)}$	$\ell_{\mathcal{C}}$	✓	d-Lin
BFL25 [BFL25]	$m^5\ell_{\mathcal{C}}^5$	$\ell_{\mathtt{w}}$	✓	Bilateral d-Lin
CEW25 [CEW25]	m	$\ell_{\mathcal{C}}$	–	Subgroup Decision Exponent
This work	m	$\ell_{\mathcal{C}}$	✓	BDHDE† (def. 16) + SXDH*

Table 1: Comparison of existing algebraic BARGs. All schemes are pairing-based and support boolean circuits $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x}}} \times \{0,1\}^{\ell_{\mathtt{w}}} \to \{0,1\}$ of size bounded by $\ell_{\mathcal{C}}$, where $\ell_{\mathtt{x}}$ is the size of one NP statement and $\ell_{\mathtt{w}}$ of a witness. Constants and polynomial factors in the security parameter are omitted, e.g., m means $O(m) \cdot \mathsf{poly}(\lambda)$. †Both the BDHDE assumption in this work and the Subgroup Decision Exponent assumption in [CEW25] are proven secure in the generic bilinear group model. *The SXDH assumption comes from the choice of the FE scheme in our concrete instantiation [Lin17] and from the Groth-Sahai proof system [GS08].

To conclude, we remark that while in this work we only apply our splitting bilinear group framework to BARGs, we believe they can be of independent interest as they are strictly more powerful than prior translation techniques for bilinear groups. Hence, they can serve as a general-purpose tool for porting other composite-order primitives that rely on complex subgroup-decision assumptions.

2 Technical Overview

As we described, the core technique of our work is a novel method to translate constructions over composite-order bilinear groups to their prime-order analogues, which we later apply to the

BARG construction in [CEW25]. In this section, we take an inverse but more intuitive approach to illustrate and motivate our translation framework. We start by describing the m-subgroup decision exponent assumption from [CEW25] and how to use it in a BARG for achieving CRS indistinguishability and somewhere extractability. Second, we attempt a translation to prime-order groups using known techniques such as dual pairing vector spaces [OT08, OT09] and their extensions [SC12, Lew12, HHH+14, LM15]. More importantly, we illustrate why these techniques fail, and thus justify the need of a new framework. Finally, we present an overview of our functional encryption-based encodings and illustrate why their indistinguishability property enables us to overcome the previous challenges.

2.1 Starting Point: BARGs from the Subgroup Decision Exponent Assumption

We start by introducing notation for *composite-order bilinear groups*, following [CEW25]. We consider (symmetric) bilinear groups represented by a tuple $\mathcal{G} = (\mathfrak{G}, \mathfrak{G}_T, N, g, e)$ where $N = pq$ for $p, q > 2^\lambda$ distinct primes, $\mathfrak{G}, \mathfrak{G}_T$ are cyclic groups of order N, and $e : \mathfrak{G} \times \mathfrak{G} \to \mathfrak{G}_T$ is a non-degenerate bilinear map that is efficiently computable. We denote by $g_p = g^{qr}$ a generator of the subgroup of order p, and by $g_q = g^{pr}$ a generator of the subgroup of order q, where $r \leftarrow\!\!\$\, \mathbb{Z}_N$.

At a high level, and omitting several details, the BARG construction by Chen, Elias and Wu [CEW25] can be seen as follows. The setup algorithm generates a CRS that includes the terms $(g_p, g_p^\alpha, \ldots, g_p^{\alpha^{m-1}})$. Additionally, the CRS includes an element $Z \in \mathfrak{G}$ which encodes a KZG-style [KZG10] vanishing polynomial, To prove the validity of a batch of m binary statements $\mathtt{x}_1, \ldots, \mathtt{x}_m$, the prover proceeds in three steps. In what follows, we remark that a witness $\mathtt{w}_i$ is valid for a statement $\mathtt{x}_i$ if $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$ for a boolean circuit $\mathcal{C}$ that checks the relation.

1. Commit to the witnesses and to all circuit wires when evaluated as $\mathcal{C}(\mathtt{x}_1, \mathtt{w}_1), \ldots, \mathcal{C}(\mathtt{x}_m, \mathtt{w}_m)$ across the m instances. This generates commitments $U_j \in \mathfrak{G}$ for $j \in [\ell_\mathcal{C}]$, where $\ell_\mathcal{C}$ is the number of circuit wires. For example, to commit to the first witness wire across all instances, the prover computes $U_1 = \sum_{i=1}^m \mathtt{w}_{i,1} \cdot g_p^{\alpha^{i-1}}$.[6]
2. Prove that each of the commitments U_j corresponding to witness wires contains only binary values. To do this, the prover generates $\ell_\mathtt{w}$ proofs $V_j \in \mathfrak{G}$ that encode the cross-terms resulting from evaluating the equation $x(x-1) = 0$. Then, the verifier checks the proofs via simple pairing equations $e(U_j, U_j) = e(g_p, U_j) \cdot e(Z, V_j)$.
3. Prove that each of the circuit gates are satisfied by the committed wires (and by the statement wires, where applicable). For this, the strategy is very similar to the above.

A crucial observation is that *all operations take place in the subgroup of order p*. Namely, if we *split* the group $\mathfrak{G}$ into its components $\mathfrak{G} = \mathfrak{G}_p \oplus \mathfrak{G}_q$ such that $A \mapsto [\![a, b]\!]$ for $A = g_p^a \cdot g_q^b$, then all commitments U_j admit a representation $[\![a, 0]\!]$ for some $a \in \mathbb{Z}_p$. This is actually enforced by the pairing checks: as g_p and Z are honestly generated, all the $\mathfrak{G}_q$ terms cancel in the pairing.

Why, then, do we require a composite order group? The order q subgroup plays a critical role in the security proof, since it enables a straight-line extraction mechanism. To see how this works, consider a *trapdoor mode* of the CRS where $(g_p, g_p^\alpha, \ldots, g_p^{\alpha^{m-1}})$ is replaced by $(g_p, g_p^\alpha \cdot g_q^{i^*}, \ldots, g_p^{\alpha^{m-1}} \cdot g_q^{(i^*)^{m-1}})$ for some secret index $i^* \in [m]$. If the CRS is instead configured in this way, then one can *project*, with the help of a trapdoor $\mathsf{td} = g_q$, all commitments U_j into the subgroup of order q. For this, note

[6] In fact, the prover commits to these witness wires in the evaluation domain (also known as the FFT domain), but we prefer to skip such technicalities here.

that pairing U_j with g_q (which is $g_q = [\![0,1]\!]$) allows one to cancel all the $\mathfrak{G}_p$ subgroup terms. That is, if $U_j = [\![a,b]\!]$, then

$$e(U_j, g_q) = e([\![a,b]\!], [\![0,1]\!]) = [a \cdot 0 + b \cdot 1]_T = [b]_T,$$

where $[b]_T = g_T^b \in \mathfrak{G}_T$, effectively cancelling out the first component. If the CRS is carefully crafted, this enables *statistical* black-box extraction of the i^*-th coordinate committed in each U_j, which is sufficient to extract a full witness $\mathtt{w}_{i^*}$.

The final step is to argue that the CRS in trapdoor and normal mode are (computationally) indistinguishable. For this, Chen, Elias and Wu rely on an extension of the subgroup decision assumption, called *m-subgroup decision exponent assumption*, over composite-order groups. The assumption says that for random $r, \alpha \leftarrow\!\!\$\ \mathbb{Z}_N$, the following distributions are computationally indistinguishable:

$$\mathcal{D}_0 : (\mathcal{G}, (g_p^{\alpha}, \ldots, g_p^{\alpha^m}), g_p) \quad \text{and} \quad \mathcal{D}_1 : (\mathcal{G}, (g_p^{\alpha}, \ldots, g_p^{\alpha^m}), g_p \cdot g_q).$$

To summarize, any translation strategy from composite-order to prime-order groups should preserve (1) the "group splitting" property that allows one to project to a subgroup using a trapdoor, and (2) the indistinguishability between the resulting distributions.

2.2 Attempting a Translation to Prime-Order Groups

Among the many existing translations framework from composite- to prime-order groups, the most complete for our purposes is by Bishop and Meiklejohn [LM15]. This framework embeds vector spaces of prime-order group elements with orthogonality properties that enable one to (1) reason about different subspaces, and (2) project into a concrete one using a trapdoor, as we require. In this setting, we consider Type-3 prime-order bilinear groups $\mathcal{G} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e})$ where $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is a non-degenerate bilinear map. We use bracket notation $[a]_x := g_x^a \in \mathbb{G}_x$ for $x \in \{1, 2, T\}$ and g_x being a generator of $\mathbb{G}_x$ For vectors $\boldsymbol{a} \in \mathbb{Z}_q^k$, we write $[\boldsymbol{a}]_x = (g_x^{a_1}, \ldots, g_x^{a_k})$.

To generate a suitable commitment key for our BARG following [LM15], we let $\boldsymbol{v}_1, \boldsymbol{v}_2 \leftarrow\!\!\$\ \mathbb{Z}_q^k$ be random linearly independent vectors, and let $\boldsymbol{v}_1^*, \boldsymbol{v}_2^* \in \mathbb{Z}_q^k$ be dual vectors satisfying $\langle \boldsymbol{v}_i, \boldsymbol{v}_j^* \rangle = 0$ if $i \neq j$ and 1 if $i = j$. Essentially, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ define two independent basis vectors, and $\boldsymbol{v}_1^*, \boldsymbol{v}_2^*$ define their orthogonal spaces. Next, we sample a random scalar $\alpha \leftarrow \mathbb{Z}_q$ and attempt to translate the m-subgroup decision assumption as follows:

$$\mathcal{D}_{\mathsf{b}} = \begin{cases} [\mathbf{1}]_1,\ [\alpha \boldsymbol{v}_1]_1,\ [\alpha^2 \boldsymbol{v}_1]_1,\ \ldots,\ [\alpha^m \boldsymbol{v}_1]_1,\ T_{\mathsf{b}}^1 = [\boldsymbol{v}_1 + \mathsf{b}\boldsymbol{v}_1^*]_1 \\ [\mathbf{1}]_2,\ [\alpha \boldsymbol{v}_2]_2,\ [\alpha^2 \boldsymbol{v}_2]_2,\ \ldots,\ [\alpha^m \boldsymbol{v}_2]_2,\ T_{\mathsf{b}}^2 = [\boldsymbol{v}_2 + \mathsf{b}\boldsymbol{v}_2^*]_2 \end{cases}$$

Naturally, the bit b also corresponds to the normal ($\mathsf{b} = 0$) and trapdoor ($\mathsf{b} = 1$) modes of the CRS in the construction. Note that we need to generate copies of the CRS in both groups $\mathbb{G}_1$ and $\mathbb{G}_2$, including the challenge terms, as the BARG verification equations include a pairing check of the form $e(U_j, U_j)$, meaning that one needs to generate commitments to wires in both groups.[7] With such a translation technique, we recover the nice projection property we had before

[7] One may wonder why do we sample different $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. We do so for generality; in fact, the attack that we describe in the coming paragraphs also applies if $\boldsymbol{v}_1 = \boldsymbol{v}_2$ and $\boldsymbol{v}_1^* = \boldsymbol{v}_2^*$, as $\boldsymbol{v}_1 \boldsymbol{v}_1^{*\intercal} \neq \boldsymbol{v}_1^* \boldsymbol{v}_1^{\intercal}$.

via *inner products*. Indeed, we have that the inner product $[\boldsymbol{v}_1^\intercal]_1 \cdot [\boldsymbol{v}_2^*]_2 = [0]_T$. Therefore, for any vector $[\boldsymbol{a}]_1 = t_1[\boldsymbol{v}_1]_1 + t_1^*[\boldsymbol{v}_1^*]_1$, we have that

$$[\boldsymbol{a}^\intercal]_1 \cdot [\boldsymbol{v}_2^*]_2 = [t_1\langle \boldsymbol{v}_1^*, \boldsymbol{v}_2^*\rangle]_T$$

effectively cancelling out the $\boldsymbol{v}_1$ component. Similarly, one can cancel out the $\boldsymbol{v}_2$ component of a vector $[\boldsymbol{b}]_2 \in \mathbb{G}_2^k$ by pairing it with $[\boldsymbol{v}_1^{*\intercal}]_1$. Thus, $[\boldsymbol{v}_1^{*\intercal}]_1$ or $[\boldsymbol{v}_2^*]_2$ could be used as extraction trapdoors to prove somewhere extractability (and should never be given to the adversary). The remaining property is indistinguishability. To see whether this holds, consider the following identity:

$$[T_{\mathsf{b}}^1]_1 \cdot [\alpha \boldsymbol{v}_2^\intercal]_2 = [\alpha \boldsymbol{v}_1]_1 \cdot [T_{\mathsf{b}}^{2\intercal}]_2$$

where we are taking *outer products*, as opposed to the inner products that we used before. Clearly, any adversary can make such a check.

For $\mathsf{b} = 0$, we have that the identity above holds:

$$[\boldsymbol{v}_1]_1 \cdot [\alpha \boldsymbol{v}_2^\intercal]_2 = [\alpha \boldsymbol{v}_1 \boldsymbol{v}_2^\intercal]_T = [\alpha \boldsymbol{v}_1]_1 \cdot [\boldsymbol{v}_2^\intercal]_2$$

However, for $\mathsf{b} = 1$, we have:

$$[\boldsymbol{v}_1 + \boldsymbol{v}_1^*]_1 \cdot [\alpha \boldsymbol{v}_2^\intercal]_2 = [\alpha(\boldsymbol{v}_1 \boldsymbol{v}_2^\intercal + \boldsymbol{v}_1^* \boldsymbol{v}_2^\intercal)]_T \neq [\alpha(\boldsymbol{v}_1 \boldsymbol{v}_2^\intercal + \boldsymbol{v}_1 \boldsymbol{v}_2^{*\intercal})]_T = [\alpha \boldsymbol{v}_1]_1 \cdot [\boldsymbol{v}_2^\intercal + \boldsymbol{v}_2^{*\intercal}]_2$$

In other words, even though $\langle \boldsymbol{v}_1, \boldsymbol{v}_2^*\rangle = \langle \boldsymbol{v}_1^*, \boldsymbol{v}_2\rangle = 0$ as intended, such cancellations do not occur in outer products, breaking indistinguishability. In the composite-order setting, commitments and CRS elements were a single element of the base group $\mathfrak{G}$, and therefore inner and outer pairing products are the same. But this is no longer the case now, as these are vectors in $\mathbb{G}_1^k$ and $\mathbb{G}_2^k$, where inner and outer pairing products are different.

Thus, we are missing a mechanism to *prevent the leakage induced by the outer pairing products* while preserving the nice inner-product projection functionality. In other words, when we swap CRS terms with different subspace components to switch modes, the distributions must look indistinguishable to the adversary as long as the inner-product relations are preserved. To achieve this property, we leverage *inner-product functional encryption*, which provides us with the indistinguishability layer that we need.

2.3 Our Functional Encryption-Based Encoding

Following [BJK15], a functional encryption for function-hiding inner-product functions (FH-IPFE) over prime-order groups consists of four algorithms:

$\mathsf{SetUp}(1^\lambda, 1^\ell) \to (\mathsf{sk}, \mathsf{pp})$: The setup algorithm generates public parameters pp and a secret key sk.
$\mathsf{DKGen}(\mathsf{sk}, [\boldsymbol{y}]_1) \to \mathsf{dk}_{\boldsymbol{y}}$: On input a secret key sk and a function that is represented by a vector $[\boldsymbol{y}]_1 \in \mathbb{G}_1^\ell$, it outputs a functional (decryption) key $\mathsf{dk}_{\boldsymbol{y}}$.
$\mathsf{Enc}(\mathsf{sk}, [\boldsymbol{x}]_2) \to \mathsf{ct}_{\boldsymbol{x}}$: On input a secret key sk and a message $[\boldsymbol{x}]_2 \in \mathbb{G}_2^\ell$, it outputs a ciphertext $\mathsf{ct}_{\boldsymbol{x}}$.
$\mathsf{Dec}(\mathsf{dk}, \mathsf{ct}) \to \mathsf{res}$: On input a functional key dk and a ciphertext ct, it outputs res as either an element in $\mathbb{G}_T$ or an error message $\perp$.

At a high level, the security of FH-IPFE ensures that given functional keys dk_i that encapsulate $[\boldsymbol{y}_i]_1$ and ciphertexts ct_j that encapsulate $[\boldsymbol{x}_j]_2$, no information beyond the inner products $\{[\langle \boldsymbol{y}_i, \boldsymbol{x}_j\rangle]_T\}_{i,j}$ can be learned. This security can be formally captured by either a simulation-based

(SIM) notion, as in [AECL25, KLM+18] or a game-based (IND-CPA) notion, as in [BJK15, Lin17, TAO16]. For our purposes, we require the FH-IPFE scheme to be linearly homomorphic and secure against multiple challenge ciphertexts and multiple functional keys. Currently, only schemes achieving IND-CPA security are suitable for instantiating our framework.

Following the FE abstraction, let $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2 \leftarrow\!\!\$\ \mathbb{Z}_q^k$ be random vectors. We denote the dual encodings of these vectors by

$$[\![\boldsymbol{a}_1, \boldsymbol{a}_2]\!]_1 := \begin{pmatrix} \mathsf{DKGen}(\mathsf{sk}, [a_{1,1}, a_{2,1}]_1) \\ \cdots \\ \mathsf{DKGen}(\mathsf{sk}, [a_{1,k}, a_{2,k}]_1) \end{pmatrix}, \quad [\![\boldsymbol{b}_1, \boldsymbol{b}_2]\!]_2 := \begin{pmatrix} \mathsf{Enc}(\mathsf{sk}, [b_{1,1}, b_{2,1}]_2) \\ \cdots \\ \mathsf{Enc}(\mathsf{sk}, [b_{1,k}, b_{2,k}]_2) \end{pmatrix},$$

where the encryption and key generation are randomized algorithms. Given this encoding, we can use the linearly homomorphic properties of the FE to construct an additive operation: $[\![\boldsymbol{a}_1, \boldsymbol{a}_2]\!]_s + [\![\boldsymbol{a}_3, \boldsymbol{a}_4]\!]_s = [\![\boldsymbol{a}_1 + \boldsymbol{a}_3, \boldsymbol{a}_2 + \boldsymbol{a}_4]\!]_s$ for $s \in \{1, 2\}$, as well as a quadratic operation (where decryption is run pairwise between ciphertexts and functional keys): $[\![\boldsymbol{a}_1, \boldsymbol{a}_2]\!]_1 \bullet [\![\boldsymbol{b}_1, \boldsymbol{b}_2]\!]_2 = \mathsf{Dec}([\![\boldsymbol{a}_1, \boldsymbol{a}_2]\!]_1, [\![\boldsymbol{b}_1, \boldsymbol{b}_2]\!]_2) = [\boldsymbol{a}_1 \boldsymbol{b}_1^\intercal + \boldsymbol{a}_2 \boldsymbol{b}_2^\intercal]_T$.

The indistinguishability of dual encodings says that given $\mathbb{G}_1$ encodings $[\![\boldsymbol{a}_{1,i}, \boldsymbol{a}_{2,i}]\!]_1$ and $\mathbb{G}_2$ encodings $[\![\boldsymbol{b}_{1,j}, \boldsymbol{b}_{2,j}]\!]_2$, no information beyond the pairing results induced by $[\![\boldsymbol{a}_{1,i}, \boldsymbol{a}_{2,i}]\!]_1 \bullet [\![\boldsymbol{b}_{1,j}, \boldsymbol{b}_{2,j}]\!]_2$ can be learned. It will be shown that this indistinguishability follows by the IND-CPA security of FH-IPFE.

Moreover, our encodings effectively behave as a bilinear group which splits into the subgroups generated by $[\![\boldsymbol{a}_1, \boldsymbol{0}]\!]_1$ and $[\![\boldsymbol{0}, \boldsymbol{a}_2]\!]_1$ (and similarly for $[\![\cdot, \cdot]\!]_2$). With this, we state a Bilateral Decisional Subgroup Assumption (BDSE) over the encodings by stating that the following distributions $\mathcal{D}_0, \mathcal{D}_1$ are indistinguishable:

$$\mathcal{D}_\mathsf{b} = \begin{cases} [\![\alpha \boldsymbol{u}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_1,\ [\![\alpha^2 \boldsymbol{u}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_1,\ \ldots,\ [\![\alpha^m \boldsymbol{u}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_1,\ T_\mathsf{b}^1 = [\![\boldsymbol{u}_1, \mathsf{b}\boldsymbol{u}_2, \boldsymbol{O}]\!]_1 \\ [\![\alpha \boldsymbol{v}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_2,\ [\![\alpha^2 \boldsymbol{v}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_2,\ \ldots,\ [\![\alpha^m \boldsymbol{v}_1, \boldsymbol{0}, \boldsymbol{O}]\!]_2,\ T_\mathsf{b}^2 = [\![\boldsymbol{v}_1, \mathsf{b}\boldsymbol{v}_2, \boldsymbol{O}]\!]_2 \end{cases}$$

where $\boldsymbol{O}$ is a padding $k \times 3k$ zero matrix appended at the end of each dual encoding for security.

Our strategy to prove the hardness of BDSE is to transition the above challenge distribution into one over prime-order groups. The indistinguishability of dual encodings is crucial for this transition: it allows encoded components to be swapped within the distribution, provided that all pairing results remain unchanged. More concretely, we modify the distribution $\mathcal{D}_\mathsf{b}$ such that the challenge terms $\mathsf{b}\boldsymbol{u}_2$ and $\mathsf{b}\boldsymbol{v}_2$ no longer appear in separate splitting components, but are combined into the additively-masked product $\boldsymbol{A} = (\boldsymbol{u}_1 \boldsymbol{v}_1^\intercal + \mathsf{b}\boldsymbol{u}_2 \boldsymbol{v}_2^\intercal)$, which corresponds to the pairing between T_b^1 and T_b^2, as follows:

$$\mathcal{D}_\mathsf{b} = \begin{cases} \{[\![\alpha^j \boldsymbol{u}_1, \boldsymbol{0}, \alpha^j \boldsymbol{u}_1 \boldsymbol{v}_1^\intercal \boldsymbol{D}_1, \boldsymbol{O}, \boldsymbol{O}]\!]_1\}_{j \in [m]}, T_\mathsf{b}^1 = [\![\ \boldsymbol{0}\ , \ \boldsymbol{0}\ , \boldsymbol{O}, \boldsymbol{D}_2^{-1}, \boldsymbol{A}\boldsymbol{D}_3]\!]_1 \\ \{[\![\alpha^j \boldsymbol{v}_1, \boldsymbol{0}, \boldsymbol{O}, \alpha^j \boldsymbol{v}_1 \boldsymbol{u}_1^\intercal \boldsymbol{D}_2, \boldsymbol{O}]\!]_2\}_{j \in [m]}, T_\mathsf{b}^2 = [\![\ \boldsymbol{0}\ , \ \boldsymbol{0}\ , \boldsymbol{D}_1^{-1}, \boldsymbol{O}, \boldsymbol{D}_3^{-1}]\!]_2 \end{cases}$$

where $\boldsymbol{O}$ is now a $k \times k$ zero matrix and $\boldsymbol{D}_1, \boldsymbol{D}_2, \boldsymbol{D}_3$ are $k \times k$ random diagonal matrices. Note that all pairing results remain unchanged.

If we now remove the dual-encoding layer and study the underlying distribution over prime-order groups, we can argue hardness in the generic bilinear group model. For this, the matrices $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$ are crucial as they limit the leakage induced by pairing operations. We call the resulting assumption the Bilateral Diffie-Hellman Decision Exponent (BDHDE). For details, we refer to Section 5.

Last but not least, in the BARG the adversary should only be able to use valid encodings, i.e., CRS elements, to commit and to generate the proofs. Otherwise, it is not possible to argue that the extractor succeeds. We ensure this by adding a verifiable membership property to our dual encodings, whose soundness holds unconditionally, and which is based on Groth–Sahai proofs [GS08].

2.4 Comparison with Prior Uses of FH-IPFE

FH-IPFE has previously been used to ensure security for other cryptographic primitives, notably by Lin and Luo [LL20] to obtain adaptive security for attribute-based encryption (ABE). Although our construction also relies on FH-IPFE, the role it plays is fundamentally different.

Conceptually, our work introduces *dual encodings* with an indistinguishability property as an abstraction of group elements, thereby addressing a gap in emulating composite-order group features in prime-order bilinear groups. In contrast, the FH-IPFE layer in [LL20] serves as an abstract instantiation of dual-system techniques in the security analysis of ABE schemes.

Technically, in our framework, dual encodings are instantiated as structured collections of FH-IPFE ciphertexts (or functional keys), yielding a richer algebraic representation than in previous applications. Together with the homomorphic properties of FH-IPFE on both the ciphertext and functional-key sides, this representation enables our construction of splitting bilinear groups and realizes the projecting property. Moreover, we show that the cancelling property is enforced via pairings through the combination of this algebraic representation and the indistinguishability of dual encodings. This property is crucial for establishing the hardness of the translated BARG assumption in the prime-order setting.

3 Preliminaries

We denote by $\mathbb{N}$ the set of natural numbers > 0. We denote the security parameter by $\lambda \in \mathbb{N}$. We call a function ϵ *negligible*, denoted $\epsilon(\lambda) = \mathsf{negl}(\lambda)$, if $\epsilon(\lambda) = O(\lambda^{-c})$ for every constant $c > 0$, and call a function $p(\lambda)$ *polynomial*, denoted poly, if $p(\lambda) = O(\lambda^c)$ for some constant $c > 0$. We say that an algorithm is *probabilistic polynomial time* (PPT) or shortly *efficient* if it consumes randomness and its running time is bounded by some $p(\lambda) = \mathsf{poly}(\lambda)$. For a finite set S, $x \leftarrow\!\!\$\ S$ denotes sampling x uniformly at random in S. For an algorithm A, we write $y \leftarrow A(x)$ for the output of A on input x. For a positive $n \in \mathbb{N}$, $[n]$ is the set $\{1, \ldots, n\}$, and for $m \in \mathbb{N}$, $[m : n]$ is the set $\{m, \ldots, n\}$. We denote vectors and matrices using bold fonts. For a vector $\boldsymbol{x}$, x_i denotes its i-th component, and for a matrix $\mathbf{M}$, $M_{i,j}$ denotes its (i, j)-th entry. For a vector $\boldsymbol{M} = (M_{i,j})_{i\in[m],j\in[n]}$, we denote its transpose by $\boldsymbol{M}^\intercal = (M_{j,i})_{j\in[n],i\in[m]}$. Given an additive group $\mathbb{A}$, for any vectors $\boldsymbol{a} \in \mathbb{A}^m$ and $\boldsymbol{b} \in \mathbb{A}^n$, their inner product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ when $m = n$ is $\boldsymbol{a}^\intercal \boldsymbol{b} = \sum_{i\in[m]} a_i b_i$, and their outer product $\boldsymbol{a} \otimes \boldsymbol{b}$ is an $m \times n$ matrix with entries $(a_i b_j)_{i\in[m],j\in[n]}$. We write $\mathbf{0}_k$ for the vectors of k zeros. We also write $\mathbf{0}_{k\times k}$ for the zero matrices of dimension $k \times k$. For the ring $\mathbb{Z}_q$ of integers modulo an integer q, we denote by $\mathbb{Z}_q^*$ the set of invertible elements of $\mathbb{Z}_q$. For a prime q, let $\mathbb{Z}_q^*$ denote the set of invertible elements of $\mathbb{Z}_q$. Given two strings x, y, we denote their concatenation by $x||y$.

3.1 Circuit model

Throughout this work, we model boolean circuits $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x},\mathtt{w}}} \to \{0,1\}$ following the notation in [BFL25], where $\ell_{\mathtt{x},\mathtt{w}} = \ell_{\mathtt{x}} + \ell_{\mathtt{w}}$, to take as input a statement $\mathtt{x} = (\mathtt{x}_i)_{i\in[\ell_{\mathtt{x}}]} \in \{0,1\}^{\ell_{\mathtt{x}}}$ and a witness $\mathtt{w} = (\mathtt{w}_i)_{i\in[\ell_{\mathtt{w}}]} \in \{0,1\}^{\ell_{\mathtt{w}}}$. A boolean circuit typically supports arbitrary fan-in and fan-out gates over

$\{0,1\}$. In this work, without loss of generality, we assume that every non-input gate is a binary NAND gate, since any boolean circuit can be transformed into an equivalent binary NAND circuit with only linear size overhead. We say that an input $(\mathtt{x}, \mathtt{w})$ is accepted by $\mathcal{C}$ if $\mathcal{C}(\mathtt{x}, \mathtt{w}) = 1$.

We measure the size of a circuit by its total number of wires $\ell_{\mathcal{C}}$, which includes the statement and witness input wires, as well as the output wire. We associate each wire to the output of a gate $g \in [\ell_{\mathcal{C}}]$. For notational convenience, and such that the number of gates and that of wires are both $\ell_{\mathcal{C}}$, we consider that all inputs come from an "input gate" $g \in [\ell_{\mathtt{x},\mathtt{w}}]$ which simply outputs the hardwired g-th input. Given a circuit computation $\mathcal{C}(\mathtt{x}, \mathtt{w})$, the value of the g-th wire (equivalently, the output of the g-th gate) is denoted by w_g. For input wires, we set $w_g = \mathtt{x}_g$ for $g \in [\ell_{\mathtt{x}}]$, and $w_g = \mathtt{w}_g$ for $g \in [\ell_{\mathtt{x}} + 1, \ell_{\mathtt{x},\mathtt{w}}]$.

Since each non-input gate $g \in [\ell_{\mathcal{C}}]$ is a binary NAND gate, then the gate computes

$$\chi_g : \{0,1\}^2 \to \{0,1\}, \qquad \chi_g(x, y) = 1 - xy.$$

We also let T_g denote the set of input wires of gate g, and thus $|T_g| = 2$.

Finally, note that each gate has a single output wire, even if this can be input to multiple other gates.

3.2 Batch Arguments for NP

A Batch Argument for NP (BARG) is a proof system for a particular subclass of NP, which is the conjunction of m NP statements corresponding to the same NP language. More precisely, given an NP language $\mathcal{L}$, the prover attests that m statements $\mathtt{x}_1, \ldots, \mathtt{x}_m$ belong to $\mathcal{L}$. BARGs produce proofs π whose size is sublinear in the number of instances m, but possibly linear in the size of the circuit $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$ that decides the language given an input $\mathtt{x}_i$ and an NP witness $\mathtt{w}_i$. We adopt the notion of (somewhere extractable) BARGs for circuit satisfiability from [CJJ22, KLVW23], where we generically consider circuits $\mathcal{C} : \mathcal{M}^{\ell_{\mathtt{x},\mathtt{w}}} \to \mathcal{M}$.

Definition 1 (BARG for NP [CJJ22, KLVW23]). *Let statement length $\ell_{\mathtt{x}} = \mathsf{poly}(\lambda)$ and witness length $\ell_{\mathtt{w}} = \mathsf{poly}(\lambda)$ be fixed. A somewhere extractable batch argument (BARG) for NP is a tuple of algorithms $\mathsf{BARG} = (\mathsf{Setup}, \mathsf{Prove}, \mathsf{Ver}, \mathsf{SetupTd}, \mathsf{Ext})$:*

$\mathsf{Setup}(1^\lambda, 1^m) \to \mathsf{crs}$: *on input the security parameter λ, and a number of instances m, outputs a common reference string crs. Below, we implicitly assume that circuits $\mathcal{C} : \mathcal{M}^{\ell_{\mathtt{x},\mathtt{w}}} \to \mathcal{M}$ are of size $\ell_{\mathcal{C}}$, statements $\mathtt{x} \in \mathcal{M}^{\ell_{\mathtt{x}}}$, and witnesses $\mathtt{w} \in \mathcal{M}^{\ell_{\mathtt{w}}}$.*

$\mathsf{Prove}(\mathsf{crs}, \mathcal{C}, \{(\mathtt{x}_i, \mathtt{w}_i)\}_{i\in[m]}) \to \pi$: *on input the common reference string crs, a circuit $\mathcal{C}$, and a batch of input-witness pairs $\{(\mathtt{x}_i, \mathtt{w}_i)\}_{i\in[m]}$, outputs a proof π.*

$\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) \to b$: *on input crs, a circuit $\mathcal{C}$ of size $\ell_{\mathcal{C}}$, a batch of statements $\{\mathtt{x}_i\}_{i\in[m]}$, and a proof π, accepts ($b = 1$) or rejects ($b = 0$).*

$\mathsf{SetupTd}(1^\lambda, 1^m, i^*) \to (\mathsf{crs}, \mathsf{td})$ *works as Setup, and additionally outputs a trapdoor td associated to the index i^* that is given as input.*

$\mathsf{Ext}(\mathsf{td}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) \to \mathtt{w}^*$ *On input a trapdoor td, a circuit $\mathcal{C}$, a batch of statements $\{\mathtt{x}_i\}_{i\in[m]}$ and a proof π, outputs a witness $\mathtt{w}^*$ corresponding to the position specified by the trapdoor.*

A BARG must satisfy the following properties:

Completeness. *For any $\lambda, m \in \mathbb{N}$, any circuit $\mathcal{C}$, all statement-witness pairs $(\mathtt{x}_i, \mathtt{w}_i)_{i\in[m]}$ such that $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i) = 1$ for all $i \in [m]$, $\mathsf{crs} \in \mathsf{Setup}(1^\lambda, 1^m)$ and $\pi \in \mathsf{Prove}(\mathsf{crs}, \mathcal{C}, \{(\mathtt{x}_i, \mathtt{w}_i)\}_{i\in[m]})$, it holds that $\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) = 1$.*

Succinctness. *For any admissible parameters as before,* $|\pi| \leq \mathsf{poly}(\lambda, \log m, \ell_{\mathcal{C}})$.

Setup Indistinguishability. *For any PPT adversary* $\mathcal{A}$*, any* $m, \ell_{\mathcal{C}} = \mathsf{poly}(\lambda)$*, and index* $i^* \in [m]$*, the distributions of* crs *induced by* $(\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupTd}(1^\lambda, 1^m, i^*)$ *and* $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$ *are computationally indistinguishable.*

Somewhere Extractability. *For any PPT adversary* $\mathcal{A}$*, any* $m, \ell_{\mathcal{C}} = \mathsf{poly}(\lambda)$*, and index* $i^* \in [m]$*,*

$$\Pr\left[\begin{array}{l} \mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) = 1 \\ \wedge\ \mathcal{C}(\mathtt{x}_{i^*}, \mathtt{w}^*) \neq 1 \end{array} \middle| \begin{array}{l} (\mathsf{crs}, \mathsf{td}) \leftarrow \mathsf{SetupTd}(1^\lambda, 1^m, i^*) \\ (\mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \\ \mathtt{w}^* \leftarrow \mathsf{Ext}(\mathsf{td}, \{\mathtt{x}_i\}_{i\in[m]}, \mathcal{C}, \pi) \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Definition 2 (Efficient Verification for BARG). *A BARG for NP* BARG *has efficient verification with preprocessing if the algorithm* $\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi)$ *can be split into the following pair of algorithms:*

$\mathsf{PreVer}(\mathsf{crs}, \{\mathtt{x}_i\}_{i\in[m]}) \to \mathsf{vk}$: *On input the common reference string* crs*, and a batch of* m *statements* $\{\mathtt{x}_i\}_{i\in[m]}$ *decidable with a circuit of size at most* $\ell_{\mathcal{C}}$*, it creates a succinct verification key* vk *such that* $|\mathsf{vk}| \leq \mathsf{poly}(\lambda, \log m, \ell_{\mathcal{C}})$

$\mathsf{EffVer}(\mathsf{vk}, \mathcal{C}, \pi) \to b$ *On input the succinct verification key* vk*, a circuit* $\mathcal{C}$ *and a proof* π*, accepts* $(b = 1)$ *or rejects* $(b = 0)$ *in time bounded by* $\mathsf{poly}(\lambda, \log m, \ell_{\mathcal{C}})$.

3.3 Bilinear Groups, Properties, and Assumptions

In this work, we construct a new bilinear group and specify the properties required for BARG. Therefore, it is advantageous to first define a bilinear group generator in a generalized manner. We adopt the definition of a bilinear group generator with an asymmetric pairing from [Fre10], along the with projecting and cancelling properties, but explicitly allow the group generator to output a trapdoor and specify its role within the definition.

Definition 3 (Generalized Bilinear Group Generator). *A* bilinear group generator $\mathcal{BG}$ *is a PPT algorithm that on input a security parameter* λ *returns a description of five abelian multiplicative groups* $\mathsf{bgd} = (\mathfrak{G}_1, \tilde{\mathfrak{G}}_1, \mathfrak{G}_2, \tilde{\mathfrak{G}}_2, \mathfrak{G}_T, e)$ *with* $\tilde{\mathfrak{G}}_1 \subset \mathfrak{G}_1$ *and* $\tilde{\mathfrak{G}}_2 \subset \mathfrak{G}_2$*, together with a trapdoor* td*. In each group, the identity element is denoted by* 1*. The description* bgd *permits efficient group operations and random sampling, while the trapdoor* td *permits efficient sampling of any target element and membership testing. The algorithm* $\mathcal{BG}$ *also outputs an efficiently computable map* $e : \mathfrak{G}_1 \times \mathfrak{G}_2 \to \mathfrak{G}_T$ *that is*

Bilinear: $e(\mathfrak{g}_1\mathfrak{h}_1, \mathfrak{g}_2\mathfrak{h}_2) = e(\mathfrak{g}_1, \mathfrak{g}_2)e(\mathfrak{g}_1, \mathfrak{h}_2)e(\mathfrak{h}_1, \mathfrak{g}_2)e(\mathfrak{h}_1, \mathfrak{h}_2)$ *for all* $\mathfrak{g}_1, \mathfrak{h}_1 \in \mathfrak{G}_1$*,* $\mathfrak{g}_2, \mathfrak{h}_2 \in \mathfrak{G}_2$*;*
Non-degenerate: *Given any* $\mathfrak{g}_1 \in \mathfrak{G}_1$*, if* $e(\mathfrak{g}_1, \mathfrak{g}_2) = 1$ *for all* $\mathfrak{g}_2 \in \mathfrak{G}_2$*, then* $\mathfrak{g}_1 = 1$*. Similarly,* $\mathfrak{g}_2 = 1$ *if* $e(\mathfrak{g}_1, \mathfrak{g}_2) = 1$ *for all* $\mathfrak{g}_1 \in \mathfrak{G}_1$.

Definition 4 (Projecting Pairings). *Let* $\mathcal{BG}$ *be a bilinear group generator. We say that* $\mathcal{BG}$ *is projecting if it additionally generates a tuple* $(\tilde{\mathfrak{G}}_T, \mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_T)$ *where* $\tilde{\mathfrak{G}}_T$ *is a subgroup of* $\mathfrak{G}_T$*, and* $\mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_T$ *are three group homomorphisms that map* $\mathfrak{G}_1, \mathfrak{G}_2, \mathfrak{G}_T$ *to themselves, respectively, and satisfy the following properties*

Annihilating: $\tilde{\mathfrak{G}}_1, \tilde{\mathfrak{G}}_2, \tilde{\mathfrak{G}}_T$ *are contained in the kernels of* $\mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_T$*;*
Commuting: $e(\mathsf{proj}_1(\mathfrak{g}_1), \mathsf{proj}_2(\mathfrak{g}_2)) = \mathsf{proj}_T(e(\mathfrak{g}_1, \mathfrak{g}_2))$ *for all* $\mathfrak{g}_1 \in \mathfrak{G}_1$*,* $\mathfrak{g}_2 \in \mathfrak{G}_2$.

Definition 5 (Cancelling Pairings). *Let $\mathcal{BG}$ be a bilinear group generator. We say that $\mathcal{BG}$ is r-cancelling if it sets $\tilde{\mathfrak{G}}_1 := \tilde{\mathfrak{G}}_{1,1}$, $\tilde{\mathfrak{G}}_2 := \tilde{\mathfrak{G}}_{2,1}$ and outputs $\{\tilde{\mathfrak{G}}_{1,s}\}_{s\in[r]}$ as subgroups of $\mathfrak{G}_1$ and $\{\tilde{\mathfrak{G}}_{2,s}\}_{s\in[r]}$ as subgroups of $\mathfrak{G}_2$. The subgroups satisfy the following properties*

Decomposing: $\mathfrak{G}_1 \cong \tilde{\mathfrak{G}}_{1,1} \times \cdots \times \tilde{\mathfrak{G}}_{1,r}$ *and* $\mathfrak{G}_2 \cong \tilde{\mathfrak{G}}_{2,1} \times \cdots \times \tilde{\mathfrak{G}}_{2,r}$,
Cancelling: $e(\mathfrak{g}_{1,i}, \mathfrak{g}_{2,j}) = 1$ *whenever* $\mathfrak{g}_{1,i} \in \mathfrak{G}_1$, $\mathfrak{g}_{2,j} \in \mathfrak{G}_2$, *and* $i \neq j$.

Definition 6 (Verifiable Membership of Source Groups). *Let $\mathcal{BG}$ be a bilinear group generator. We say that $\mathcal{BG}$ supports publicly verifiable membership for its groups $\mathfrak{G}_1$ and $\mathfrak{G}_2$ if it additionally generates a pair of verification algorithms $\mathsf{Ver}_{\mathfrak{G}_1} : \{0,1\}^* \to \{0,1\}$ and $\mathsf{Ver}_{\mathfrak{G}_2} : \{0,1\}^* \to \{0,1\}$ that satisfy the following properties:*

Completeness Closure: *For any group $\mathfrak{G} \in \{\mathfrak{G}_1, \mathfrak{G}_2\}$, any elements $\mathfrak{g}_1, \ldots \mathfrak{g}_n \in \mathfrak{G}$ that are additionally given by $\mathcal{BG}$, it holds that $\mathsf{Ver}_{\mathfrak{G}}(\mathfrak{g}_i) = 1$ $\forall i$. Furthermore, for any scalars $\alpha_1, \ldots, \alpha_n \in \mathbb{Z}$, it holds that $\mathsf{Ver}_{\mathfrak{G}}(\prod_{i\in[n]} \mathfrak{g}_i^{\alpha_i}) = 1$.*
Perfect Soundness: *For any group $\mathfrak{G} \in \{\mathfrak{G}_1, \mathfrak{G}_2\}$, if any string $\mathfrak{g}$ satisfies $\mathsf{Ver}_{\mathfrak{G}}(\mathfrak{g}) = 1$, then it holds $\mathfrak{g} \in \mathfrak{G}$.*

Prime-Order Groups As a special case of Definition 3, we define a prime-order bilinear group generator $\mathcal{P}(1^\lambda)$ as an algorithm that returns $\mathsf{bgd} := (q, \mathfrak{G}_1 = \tilde{\mathfrak{G}}_1 = \mathbb{G}_1, \mathfrak{G}_2 = \tilde{\mathfrak{G}}_2 = \mathbb{G}_2, \tilde{\mathfrak{G}}_T = \mathbb{G}_T, g_1, g_2, \hat{e})$ and an empty string for td, where $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ are groups of prime order q, $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$ are generators. Furthermore, we assume that there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction (so-called Type-3 pairing groups). For prime-order groups, we use the bracket notation of [EHK+13] for group elements: for $s \in \{1, 2, T\}$ and $x \in \mathbb{Z}_q$, $[x]_s$ denotes $g_s^x \in \mathbb{G}_s$. We use additive notation for all $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$. We note that given an element $[x]_s \in \mathbb{G}_s$, for $s = 1, 2$, and a scalar a, one can efficiently compute $a[x] = [ax] = g_s^{ax} \in \mathbb{G}_s$; given group elements $[a]_1 \in \mathbb{G}_1$ and $[b]_2 \in \mathbb{G}_2$, one can efficiently compute $[ab]_T = [a]_1 \cdot [b]_2$. For a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$, we represent a matrix of group elements $g_s^{\mathbf{A}}$ as $[\mathbf{A}]_s \in \mathbb{G}_s^{m\times n}$. With appropriate dimensions, we also write $\boldsymbol{A}[\boldsymbol{B}]_s = [\boldsymbol{AB}]_s$ and $[\boldsymbol{A}]_1 \cdot [\boldsymbol{B}]_2 = [\boldsymbol{AB}]_T$.

3.4 Functional Encryption for Function-Hiding Inner Products

We adopt the definition of functional encryption within bilinear groups, which was introduced in [SW05, O'N10, BSW11] and specified for function-hiding inner products in [BJK15], along with homomorphic properties as follows.

Definition 7 (Pairing-based FE for Function-Hiding Inner Products). *Given an asymmetric pairing group $\mathsf{bgd} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$, a functional encryption for hiding inner-product functions (FH-IPFE) over bgd consists of four algorithms $\mathsf{IPE} = (\mathsf{SetUp}, \mathsf{Enc}, \mathsf{DKGen}, \mathsf{Dec})$:*

$\mathsf{SetUp}(1^\lambda, 1^{\ell_{\mathsf{ip}}}) \to (\mathsf{sk}, \mathsf{pp})$: *on input the security parameter λ and the length of vectors for the functional keys and ciphertexts ℓ_{ip}, it outputs public parameters pp and a secret key sk. The parameters pp are implicitly assumed to be an argument to other algorithms.*
$\mathsf{DKGen}(\mathsf{sk}, [\boldsymbol{y}]_1) \to \mathsf{dk}_{\boldsymbol{y}}$: *on input a secret key sk and a function that is represented by a vector $[\boldsymbol{y}]_1 \in \mathbb{G}_1^{\ell_{\mathsf{ip}}}$, it outputs a functional (decryption) key $\mathsf{dk}_{\boldsymbol{y}}$.*
$\mathsf{Enc}(\mathsf{sk}, [\boldsymbol{x}]_2) \to \mathsf{ct}_{\boldsymbol{x}}$: *on input a secret key sk and a message $[\boldsymbol{x}]_2 \in \mathbb{G}_2^{\ell_{\mathsf{ip}}}$, it outputs a ciphertext $\mathsf{ct}_{\boldsymbol{x}}$.*
$\mathsf{Dec}(\mathsf{dk}, \mathsf{ct}) \to \mathsf{res}$: *on input a functional key dk and a ciphertext ct, it outputs res as either an element in $\mathbb{G}_T$ or an error message $\perp$.*

A homomorphic FH-IPFE scheme must satisfy the following properties:

Correctness. *For all parameters* $\lambda, \ell_{\mathsf{ip}} \in \mathbb{N}$*, all secret keys* sk *generated by* $\mathsf{SetUp}(1^\lambda, 1^{\ell_{\mathsf{ip}}})$*, all key vectors* $\boldsymbol{y} \in \mathbb{Z}_q^{\ell_{\mathsf{ip}}}$ *and all message vectors* $\boldsymbol{x} \in \mathbb{Z}_q^{\ell_{\mathsf{ip}}}$*, given* $\mathsf{dk} = \mathsf{DKGen}(\mathsf{sk}, [\boldsymbol{y}]_1)$ *and* $\mathsf{ct} = \mathsf{Enc}(\mathsf{sk}, [\boldsymbol{x}]_2)$*, it holds that* $\mathsf{Dec}(\mathsf{dk}_{\boldsymbol{y}}, \mathsf{ct}_{\boldsymbol{x}}) = [\boldsymbol{x}^\mathsf{T}\boldsymbol{y}]_T$*.*

Linearly Homomorphic Functional Keys. *An FH-IPFE scheme adopts linearly homomorphic functional keys if the algorithm* DKGen *adopts a deterministic algorithm* DKGen' *such that*

$$\begin{aligned}\mathsf{DKGen}(\mathsf{sk}, [\boldsymbol{y}]_1) &= \mathsf{DKGen}'(\mathsf{sk}, [\boldsymbol{y}]_1; \boldsymbol{r}) \\ &:= [\sum_{i \in [\ell_{\mathsf{ip}}]} y_i \boldsymbol{c}_{\mathsf{sk},i} + \sum_{i \in [p(\lambda)]} r_i \boldsymbol{c}_{\mathsf{sk},\ell_{\mathsf{ip}}+i}]_1 \in \mathbb{G}_1^{q(\ell_{\mathsf{ip}},\lambda)}\end{aligned}$$

for some uniform randomness $\boldsymbol{r} \in \mathbb{Z}_q^{p(\lambda)}$ *and some secret-key vectors* $\boldsymbol{c}_{\mathsf{sk},1}, \ldots, \boldsymbol{c}_{\mathsf{sk},\ell_{\mathsf{ip}}+p(\lambda)} \in \mathbb{Z}_q^{q(\ell_{\mathsf{ip}},\lambda)}$*. Here,* $q(\ell_{\mathsf{ip}}, \lambda)$ *and* $p(\lambda)$ *are polynomials mapping into* $\mathbb{N}$ *that represent sizes of the functional keys and their randomness, respectively.*

Linearly Homomorphic Ciphertexts. *An FH-IPFE scheme adopts linearly homomorphic ciphertexts if the algorithm* Enc *adopts a deterministic algorithm* Enc' *such that*

$$\begin{aligned}\mathsf{Enc}(\mathsf{sk}, [\boldsymbol{x}]_2) &= \mathsf{Enc}'(\mathsf{sk}, [\boldsymbol{x}]_2; \boldsymbol{r}^*) \\ &:= [\sum_{i \in [\ell_{\mathsf{ip}}]} x_i \boldsymbol{c}^*_{\mathsf{sk},i} + \sum_{i \in [p^*(\lambda)]} r^*_i \boldsymbol{c}^*_{\mathsf{sk},\ell_{\mathsf{ip}}+i}]_2 \in \mathbb{G}_2^{q^*(\ell_{\mathsf{ip}},\lambda)}\end{aligned}$$

for some uniform randomness $\boldsymbol{r}^* \in \mathbb{Z}_q^{p^*(\lambda)}$ *and some secret-key vectors* $\boldsymbol{c}^*_{\mathsf{sk},1}, \ldots, \boldsymbol{c}^*_{\mathsf{sk},\ell_{\mathsf{ip}}+p(\lambda)} \in \mathbb{Z}_q^{q^*(\ell_{\mathsf{ip}},\lambda)}$*. Here,* $q^*(\ell_{\mathsf{ip}}, \lambda)$ *and* $p^*(\lambda)$ *are polynomials mapping into* $\mathbb{N}$ *that represent sizes of the ciphertexts and their randomness, respectively.*

From the linear homomorphic properties of functional keys and ciphertexts, we have the following remarks.

Remark 1. Given any two functional keys

$$\mathsf{dk}_1 = \mathsf{DKGen}'(\mathsf{sk}, [\boldsymbol{y}_1]_1; \boldsymbol{r}_1) \text{ and } \mathsf{dk}_2 = \mathsf{DKGen}'(\mathsf{sk}, [\boldsymbol{y}_2]_1; \boldsymbol{r}_2),$$

for any linear combination defined by $\alpha_1, \alpha_2 \in \mathbb{Z}_q$, it holds that

$$\alpha_1 \mathsf{dk}_1 + \alpha_2 \mathsf{dk}_2 = \mathsf{DKGen}'(\mathsf{sk}, [\alpha_1 \boldsymbol{y}_1 + \alpha_2 \boldsymbol{y}_2]_1; \alpha_1 \boldsymbol{r}_1 + \alpha_2 \boldsymbol{r}_2).$$

Remark 2. Given any two ciphertexts

$$\mathsf{ct}_1 = \mathsf{Enc}'(\mathsf{sk}, [\boldsymbol{x}_1]_2; \boldsymbol{r}_1) \text{ and } \mathsf{ct}_2 = \mathsf{Enc}'(\mathsf{sk}, [\boldsymbol{x}_2]_2; \boldsymbol{r}_2),$$

for any linear combination defined by $\alpha_1, \alpha_2 \in \mathbb{Z}_q$, it holds that

$$\alpha_1 \mathsf{ct}_1 + \alpha_2 \mathsf{ct}_2 = \mathsf{Enc}'(\mathsf{sk}, [\alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2]_2; \alpha_1 \boldsymbol{r}_1 + \alpha_2 \boldsymbol{r}_2).$$

The game-based definition is provided as follows.

Definition 8 (Indistinguishability-based Security). *Let* IPE *be a functional encryption scheme for hiding inner products of length* ℓ_{ip} *over a pairing group* bgd*. We define the experiments* $\mathrm{IND}_{\mathsf{IPE},\mathsf{b}}$ *in Figure 1, where* $\mathsf{b} \in \{0,1\}$ *and the oracles and condition are defined as:*

- ***Functional-key oracle*** $\mathsf{QDKGen}_{\mathsf{b}}([\boldsymbol{y}^0]_1, [\boldsymbol{y}^1]_1)$*: on input two challenge function vectors* $([\boldsymbol{y}^0]_1, [\boldsymbol{y}^1]_1)$*, it returns the decryption key* $\mathsf{dk} \leftarrow \mathsf{DKGen}(\mathsf{sk}, [\boldsymbol{y}^{\mathsf{b}}]_1)$*.*
- ***Encryption oracle*** $\mathsf{QEnc}_{\mathsf{b}}([\boldsymbol{x}^0]_2, [\boldsymbol{x}^1]_2)$*: on input two challenge plaintext vectors* $([\boldsymbol{x}^0]_2, [\boldsymbol{x}^1]_2)$*, it returns the ciphertext* $\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{sk}, [\boldsymbol{x}^{\mathsf{b}}]_2)$*.*
- ***Condition*** $(*)$***:*** *Every functional-key query on* $([\boldsymbol{y}^0]_1, [\boldsymbol{y}^1]_1)$ *and every encryption query on* $([\boldsymbol{x}^0]_2, [\boldsymbol{x}^1]_2)$ *satisfy* $[\langle \boldsymbol{y}^0, \boldsymbol{x}^0 \rangle]_T = [\langle \boldsymbol{y}^1, \boldsymbol{x}^1 \rangle]_T$*.*

The IPE *scheme is said to be IND-CPA secure if given any parameter* ℓ_{ip}*, for every PPT adversary* $\mathcal{A}$*, the advantage defined as follows is bounded by* $\mathsf{negl}(\lambda)$*:*

$$\mathsf{Adv}_{\mathsf{IPE},\mathcal{A}}(\lambda, \ell_{\mathsf{ip}}) := \Big|\Pr[\mathrm{IND}_{\mathsf{IPE},0}(1^\lambda, 1^{\ell_{\mathsf{ip}}}, \mathcal{A}) = 1] - \Pr[\mathrm{IND}_{\mathsf{IPE},1}(1^\lambda, 1^{\ell_{\mathsf{ip}}}, \mathcal{A}) = 1]\Big|.$$

$\underline{\mathrm{IND}_{\mathsf{IPE},\mathsf{b}}(1^\lambda, 1^{\ell_{\mathsf{ip}}}, \mathcal{A})}$:
$(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{SetUp}(1^\lambda, 1^{\ell_{\mathsf{ip}}})$; $\quad \alpha \leftarrow \mathcal{A}^{\mathsf{QDKGen}_{\mathsf{b}}(\cdot,\cdot), \mathsf{QEnc}_{\mathsf{b}}(\cdot,\cdot)}(\mathsf{pp})$
Output: $\beta = \alpha$ if Condition $(*)$ is satisfied, $\beta \leftarrow\!\!\$\ \{0,1\}$ otherwise.

Fig. 1: Security experiment for IPE.

Schemes realizing the above definitions are given in [TAO16] and [Lin17], and are proven secure under the Decisional Linear (DLIN) assumption and the Symmetric External Diffie–Hellman (SXDH) assumption, respectively. Both assumptions can be viewed as particular cases of the standard parameterized d-Lin assumption in prime-order bilinear groups.

4 Functional Encryption-Based Bilinear Maps

4.1 Dual Encoding from Homomorphic FH-IPFE

We present our dual encoding for vectors of elements from the source groups of prime-order bilinear groups. We base our constructions on the homomorphic FH-IPFE from the previous section. We start by introducing the dual-encoding notation.

Definition 9 (Dual-Encoding Notation). *Let* $\mathsf{IPE} = (\mathsf{SetUp}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{Dec})$ *be an FH-IPFE scheme over a prime-order bilinear group* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$*,* r *be an integer, and* $\mathsf{sk} \leftarrow \mathsf{SetUp}(1^\lambda, 1^r)$ *be an* IPE *secret key.*

- *For any* m*-dimensional vectors* $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r \in \mathbb{Z}_q^m$*, we define the* $\mathbb{G}_1$*-encoding* $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1$ *as the equivalence class of* m*-tuples of functional keys* $(\mathsf{dk}_i)_{i\in[m]}$ *where* $\mathsf{dk}_i \leftarrow \mathsf{DKGen}(\mathsf{sk}, [a_{1,i}, \ldots, a_{r,i}]_1)$*. Different choices of the internal randomness in* DKGen *yield different representatives of the same class.*
- *For any* n*-dimensional vectors* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r \in \mathbb{Z}_q^n$*, we define the* $\mathbb{G}_2$*-encoding* $[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2$ *as the equivalence class of* n*-tuples of ciphertexts* $(\mathsf{ct}_i)_{i\in[n]}$ *where* $\mathsf{ct}_i \leftarrow \mathsf{Enc}(\mathsf{sk}, [b_{1,i}, \ldots, b_{r,i}]_2)$*. Different choices of the internal randomness in* Enc *yield different representatives of the same class.*

Note that r is a fixed parameter that corresponds to the length of functional keys and ciphertexts ℓ_{ip}. For notational convenience, we treat every $\mathbb{G}_1$-encoding $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1$ as a tuple of functional keys $(\mathsf{DKGen}'(\mathsf{sk}, [a_{1,i}, \ldots, a_{r,i}]_1; \boldsymbol{r}_i))_{i\in[m]}$ where $\{\boldsymbol{r}_i\}_{i\in[m]}$ is sampled at random, i.e., we refer to a random representative of the equivalence class. Similarly, every $\mathbb{G}_2$-encoding $[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2$ is treated as a tuple of ciphertexts $(\mathsf{Enc}'(\mathsf{sk}, [b_{1,i}, \ldots, b_{r,i}]_2; \boldsymbol{r}_i^*))_{i\in[n]}$.

Next, we define pairwise operations between encodings.

Definition 10 (Linear Operation). *Given any $\mathbb{G}_1$-encodings $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1$, $[\![\boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_r]\!]_1$ and any scalars $\alpha, \alpha' \in \mathbb{Z}_q$, we define the linear operation between these two encodings as*

$$\begin{aligned}&\alpha[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1 + \alpha'[\![\boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_r]\!]_1\\ :=\ &(\alpha\mathsf{DKGen}'(\mathsf{sk}, [a_{1,i}, \ldots, a_{r,i}]_1; \boldsymbol{r}_i) + \alpha'\mathsf{DKGen}'(\mathsf{sk}, [a'_{1,i}, \ldots, a'_{r,i}]_1; \boldsymbol{r}'_i))_{i\in[m]}.\end{aligned}$$

Analogously, given any $\mathbb{G}_2$-encodings $[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2$, $[\![\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_r]\!]_2$ and any scalars $\beta, \beta' \in \mathbb{Z}_q$, we define the linear operation between these two encodings as

$$\begin{aligned}&\beta[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2 + \beta'[\![\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_r]\!]_2\\ :=\ &(\beta\mathsf{Enc}'(\mathsf{sk}, [b_{1,i}, \ldots, b_{r,i}]_2; \boldsymbol{r}_i) + \beta'\mathsf{Enc}'(\mathsf{sk}, [b'_{1,i}, \ldots, b'_{r,i}]_2; \boldsymbol{r}'_i))_{i\in[n]}.\end{aligned}$$

Remark 3. By Remark 1, both the $\mathbb{G}_1$- and the $\mathbb{G}_2$-encodings are homomorphic with respect to the linear operation:

$$\begin{aligned}\alpha[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1 + \alpha'[\![\boldsymbol{a}'_1, \ldots, \boldsymbol{a}'_r]\!]_1 &= [\![\alpha\boldsymbol{a}_1 + \alpha'\boldsymbol{a}'_1, \ldots, \alpha\boldsymbol{a}_r + \alpha'\boldsymbol{a}'_r]\!]_1.\\ \beta[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2 + \beta'[\![\boldsymbol{b}'_1, \ldots, \boldsymbol{b}'_r]\!]_2 &= [\![\beta\boldsymbol{b}_1 + \beta'\boldsymbol{b}'_1, \ldots, \beta\boldsymbol{b}_r + \beta'\boldsymbol{b}'_r]\!]_2.\end{aligned}$$

Definition 11 (Quadratic Operation). *Given any $\mathbb{G}_1$-encoding $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1$ and $\mathbb{G}_2$-encoding $[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2$, their inner product $\bullet$ is an $m \times n$ matrix defined as $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1 \bullet [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2 := (\mathsf{Dec}(\mathsf{dk}_i, \mathsf{ct}_j))_{i\in[m],j\in[n]} \in \mathbb{G}_T^{m\times n}$.*

Remark 4. By the IPE inner-product decryption, this product implies the following equalities:

$$\begin{aligned}[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1 \bullet [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2 &= ([a_{1,i}b_{1,j} + \cdots + a_{r,i}b_{r,j}]_T)_{i\in[m],j\in[n]}\\ &= [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \cdots + \boldsymbol{a}_r \otimes \boldsymbol{b}_r]_T.\end{aligned}$$

Remark 5. If we have matrices $\mathbf{A} \in \mathbb{Z}_q^{m\times(r-k)}$ and $\mathbf{B} \in \mathbb{Z}_q^{n\times(r-k)}$ formed by the column vectors $\{\boldsymbol{a}_i\}_{i\in[k+1,r]}$ and $\{\boldsymbol{b}_i\}_{i\in[k+1,r]}$ respectively, we can write $[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k, \mathbf{A}]\!]_1 := [\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_r]\!]_1$ and $[\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k, \mathbf{B}]\!]_2 := [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_r]\!]_2$, consistently with the notation above. In this case, the following equalities hold

$$\begin{aligned}&[\![\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k, \mathbf{A}]\!]_1 \bullet [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k, \mathbf{B}]\!]_2\\ =\ &([a_{1,i}b_{1,j} + \cdots + a_{k,i}b_{k,j} + a_{k+1,i}b_{k+1,j} + \cdots + a_{r,i}b_{r,j}]_T)_{i\in[m],j\in[n]}\\ =\ &[\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \cdots + \boldsymbol{a}_k \otimes \boldsymbol{b}_k]_T + ([A_{i,1}B_{j,1} + \cdots + A_{i,r-k}B_{j,r-k}]_T)_{i\in[m],j\in[n]}\\ =\ &[\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 + \cdots + \boldsymbol{a}_k \otimes \boldsymbol{b}_k + \mathbf{A}\mathbf{B}^\intercal]_T.\end{aligned}$$

In the following, we show that the IND-CPA security of the underlying FE scheme for inner products implies the IND-CPA security of the dual encoding. For this, we first define the security notion for dual encodings, which is naturally inspired by FE security.

Definition 12 (Indistinguishability of Dual Encoding). *Given a dual encoding system* $\mathsf{DE} = (\llbracket\cdot\rrbracket_1, \llbracket\cdot\rrbracket_2)$ *from an FH-IPFE scheme* $\mathsf{IPE} = (\mathsf{SetUp}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{Dec})$ *over a prime-order bilinear group* $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$ *as in Definition 9, we define the experiments* $\mathrm{IND}_{\mathsf{DE},\mathsf{b}}$ *in Figure 2, where* $\mathsf{b} \in \{0,1\}$ *and the oracles are defined as:*

- $\mathbb{G}_1$***-encoding oracle*** $\mathsf{LEnc}^m_{\mathsf{b}}([\boldsymbol{a}^0_1, \ldots, \boldsymbol{a}^0_r]_1, [\boldsymbol{a}^1_1, \ldots, \boldsymbol{a}^1_r]_1)$*: on input* $\mathbb{G}_1$*-element vectors* $([\boldsymbol{a}^0_1, \ldots, \boldsymbol{a}^0_r]_1, [\boldsymbol{a}^1_1, \ldots, \boldsymbol{a}^1_r]_1)$*, it returns*

$$\llbracket \boldsymbol{a}^{\mathsf{b}}_1, \ldots, \boldsymbol{a}^{\mathsf{b}}_r \rrbracket_1 \leftarrow (\mathsf{DKGen}(\mathsf{sk}, [a^{\mathsf{b}}_{1,i}, \ldots, a^{\mathsf{b}}_{r,i}]_1))_{i\in[m]}.$$

- $\mathbb{G}_2$***-encoding oracle*** $\mathsf{REnc}^n_{\mathsf{b}}([\boldsymbol{b}^0_1, \ldots, \boldsymbol{b}^0_r]_2, [\boldsymbol{b}^1_1, \ldots, \boldsymbol{b}^1_r]_2)$*: on input* $\mathbb{G}_2$*-element vectors* $([\boldsymbol{b}^0_1, \ldots, \boldsymbol{b}^0_r]_2, [\boldsymbol{b}^1_1, \ldots, \boldsymbol{b}^1_r]_2)$*, it returns*

$$\llbracket \boldsymbol{b}^{\mathsf{b}}_1, \ldots, \boldsymbol{b}^{\mathsf{b}}_r \rrbracket_2 \leftarrow (\mathsf{Enc}(\mathsf{sk}, [b^{\mathsf{b}}_{1,i}, \ldots, b^{\mathsf{b}}_{r,i}]_2))_{i\in[n]}.$$

- ***Condition*** $(*)$***:*** *Every* $\mathbb{G}_1$*-encoding query on* $([\boldsymbol{a}^0_1, \ldots, \boldsymbol{a}^0_r]_1, [\boldsymbol{a}^1_1, \ldots, \boldsymbol{a}^1_r]_1)$ *and every* $\mathbb{G}_2$*-encoding query on* $([\boldsymbol{b}^0_1, \ldots, \boldsymbol{b}^0_r]_2, [\boldsymbol{b}^1_1, \ldots, \boldsymbol{b}^1_r]_2)$ *satisfy*

$$[\boldsymbol{a}^0_1 \otimes \boldsymbol{b}^0_1 + \cdots + \boldsymbol{a}^0_r \otimes \boldsymbol{b}^0_r]_T = [\boldsymbol{a}^1_1 \otimes \boldsymbol{b}^1_1 + \cdots + \boldsymbol{a}^1_r \otimes \boldsymbol{b}^1_r]_T.$$

The dual encoding system $(\llbracket\cdot\rrbracket_1, \llbracket\cdot\rrbracket_2)$ *is said to be IND-CPA secure if given any parameters* r*,* m*, and* n*, for every PPT adversary* $\mathcal{A}$*, the advantage defined as follows is bounded by* $\mathsf{negl}(\lambda)$*:*

$$\begin{aligned}&\mathsf{Adv}_{\mathsf{DE},\mathcal{A}}(\lambda, r, m, n)\\ &:= \Big|\Pr[\mathrm{IND}_{\mathsf{DE},0}(1^\lambda, 1^r, 1^m, 1^n, \mathcal{A}) = 1] - \Pr[\mathrm{IND}_{\mathsf{DE},1}(1^\lambda, 1^r, 1^m, 1^n, \mathcal{A}) = 1]\Big|.\end{aligned}$$

$\underline{\mathrm{IND}_{\mathsf{DE},\mathsf{b}}(1^\lambda, 1^r, 1^m, 1^n, \mathcal{A})}$:
$(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{SetUp}(1^\lambda, 1^r)$; $\quad \alpha \leftarrow \mathcal{A}^{\mathsf{LEnc}^m_{\mathsf{b}}(\cdot,\cdot), \mathsf{REnc}^n_{\mathsf{b}}(\cdot,\cdot)}(\mathsf{pp})$
Output: $\beta = \alpha$ if Condition $(*)$ is satisfied, $\beta \leftarrow\!\!\$\ \{0,1\}$ otherwise.

Fig. 2: Security experiment for dual encoding.

Theorem 1 (Indistinguishability of Dual Encoding). *If the* IPE *scheme is IND-CPA secure (as per Definition 8), then the dual encoding system* $\mathsf{DE} = (\llbracket\cdot\rrbracket_1, \llbracket\cdot\rrbracket_2)$ *is also IND-CPA secure (as per Definition 12).*

More precisely, for any adversary $\mathcal{A}_{q_{\mathsf{LEnc}}, q_{\mathsf{REnc}}}$ *that breaks the security of* DE *by making at most* q_{LEnc} *and* q_{REnc} *queries to* $\mathsf{LEnc}^m_{\mathsf{b}}$ *and* $\mathsf{REnc}^n_{\mathsf{b}}$ *oracles, respectively, there exists an adversary* $\mathcal{B}_{mq_{\mathsf{LEnc}}, nq_{\mathsf{REnc}}}$ *that breaks the security of* IPE *by making at most* mq_{LEnc} *and* nq_{REnc} *queries to* $\mathsf{QDKGen}_{\mathsf{b}}$ *and* $\mathsf{QEnc}_{\mathsf{b}}$ *oracles, respectively, such that*

$$\mathsf{Adv}_{\mathsf{DE},\mathcal{A}_{q_{\mathsf{LEnc}}, q_{\mathsf{REnc}}}}(\lambda, r, m, n) \le \mathsf{Adv}_{\mathcal{B}_{\mathsf{IPE}, mq_{\mathsf{LEnc}}, nq_{\mathsf{REnc}}}}(\lambda, r).$$

Proof. Given an adversary $\mathcal{A}$ for the dual encoding system DE, we construct an adversary $\mathcal{B}$ for the functional encryption scheme IPE as follows:

– At the beginning, $\mathcal{B}$ receives the IPE parameters pp and sends them to $\mathcal{A}$.
– On receiving any query $\mathsf{LEnc}_{\mathsf{b}}^{m}([\boldsymbol{a}_1^0,\ldots,\boldsymbol{a}_r^0]_1,[\boldsymbol{a}_1^1,\ldots,\boldsymbol{a}_r^1]_1)$, $\mathcal{B}$ obtains

$$\mathsf{dk}_i \leftarrow \mathsf{QDKGen}_{\mathsf{b}}([a_{1,i}^0,\ldots,a_{r,i}^0]_1,[a_{1,i}^1,\ldots,a_{r,i}^1]_1)$$

for $i\in[m]$ and returns $(\mathsf{dk}_i)_{i\in[m]}$ to $\mathcal{A}$.
– On receiving any query $\mathsf{REnc}_{\mathsf{b}}^{n}([\boldsymbol{b}_1^0,\ldots,\boldsymbol{b}_r^0]_2,[\boldsymbol{b}_1^1,\ldots,\boldsymbol{b}_r^1]_2)$, $\mathcal{B}$ obtains

$$\mathsf{ct}_i \leftarrow \mathsf{QEnc}_{\mathsf{b}}([b_{1,i}^0,\ldots,b_{r,i}^0]_2,[b_{1,i}^1,\ldots,b_{r,i}^1]_2)$$

for $i\in[n]$ and returns $(\mathsf{ct}_i)_{i\in[n]}$ to $\mathcal{A}$.
– At the end, $\mathcal{B}$ outputs what $\mathcal{A}$ outputs.

Assume that $\mathcal{A}$ satisfies the admissibility condition. Note that every functional-key query on $([a_{1,i}^0,\ldots,a_{r,i}^0]_1,[a_{1,i}^1,\ldots,a_{r,i}^1]_1)$ and every encryption query on $([b_{1,j}^0,\ldots,b_{r,j}^0]_2,[b_{1,j}^1,\ldots,b_{r,j}^1]_2)$ that $\mathcal{B}$ makes are part of the encoding queries made by $\mathcal{A}$ on respectively $([\boldsymbol{a}_1^0,\ldots,\boldsymbol{a}_r^0]_1,[\boldsymbol{a}_1^1,\ldots,\boldsymbol{a}_r^1]_1)$ and $([\boldsymbol{b}_1^0,\ldots,\boldsymbol{b}_r^0]_2,[\boldsymbol{b}_1^1,\ldots,\boldsymbol{b}_r^1]_2)$. Since $\mathcal{A}$ is an admissible adversary, then

$$[\boldsymbol{a}_1^0\otimes\boldsymbol{b}_1^0+\cdots+\boldsymbol{a}_r^0\otimes\boldsymbol{b}_r^0]_T=[\boldsymbol{a}_1^1\otimes\boldsymbol{b}_1^1+\cdots+\boldsymbol{a}_r^1\otimes\boldsymbol{b}_r^1]_T.$$

As the identity holds as a matrix, it holds for every element, and so $[a_{1,i}^0b_{1,j}^0+\cdots+a_{r,i}^0b_{r,j}^0]_T=[a_{1,i}^1b_{1,j}^1+\cdots+a_{r,i}^1b_{r,j}^1]_T$, which means that $\mathcal{B}$ also satisfies the admissibility condition. Therefore, if $\mathcal{A}$ wins then $\mathcal{B}$ also wins. □

4.2 Splitting Bilinear Group Generator

After introducing our dual encoding over prime-order groups, our next step is to construct a bilinear group generator with the projecting and cancelling properties that we seek. We call such a generator *splitting*. Later on, we will show how to use our splitting bilinear group generator, together with the indistinguishability property of the dual encodings, to translate the subgroup decision exponent assumption into an assumption over prime-order groups.

To construct a splitting bilinear group generator, it suffices to consider a specific case of dual-encoding parameters where $m=n=k$. We further introduce a parameter $\rho:=r-k$ to define the length of the zero padding $\mathbf{0}_{k\times\rho}$ appended at the end of each dual encoding. This zero padding will play a crucial role in the hardness proofs of the subgroup decision exponent assumption defined over splitting bilinear groups. Additionally, we assume that $\{\boldsymbol{a}_i\}_{i\in[k]}$ and $\{\boldsymbol{b}_i\}_{i\in[k]}$ are two bases of the $\mathbb{Z}_q$-vector space $\mathbb{Z}_q^k$. Consequently, the set $\{\boldsymbol{a}_i\otimes\boldsymbol{b}_i\}_{i\in[k]}$ consists of linearly independent vectors in $\mathbb{Z}_q^{k\times k}$.

Definition 13 (Splitting Groups). *We define the set of (equivalence) classes*

$$\mathfrak{G}_1:=\{[\![x_1\boldsymbol{a}_1,(x_l\boldsymbol{a}_l)_{l\in[2:k]},\mathbf{0}_{k\times\rho}]\!]_1:x_i\in\mathbb{Z}_q\ \forall i\in[k]\}.$$

This set is closed and associative under the linear operation, with identity element $1_{\mathfrak{G}_1}=[\![0\boldsymbol{a}_1,(\mathbf{0})_{l\in[2:k]},\mathbf{0}_{k\times\rho}]\!]_1$. *Moreover, for every* $\mathfrak{g}=[\![x_1\boldsymbol{a}_1,(x_l\boldsymbol{a}_l)_{l\in[2:k]},\mathbf{0}_{k\times\rho}]\!]_1$, *its inverse is* $\mathfrak{g}^{-1}=[\![-x_1\boldsymbol{a}_1,(-x_l\boldsymbol{a}_l)_{l\in[2:k]},\mathbf{0}_{k\times\rho}]\!]_1$. *Hence,* $\mathfrak{G}_1$ *is an abelian group isomorphic to* $\mathbb{Z}_q^k$. *Additionally,* $\mathfrak{G}_1$ *adopts the subset*

$$\tilde{\mathfrak{G}}_1:=\{[\![x_1\boldsymbol{a}_1,(\mathbf{0})_{l\in[2:k]},\mathbf{0}_{k\times\rho}]\!]_1:x_1\in\mathbb{Z}_q\}$$

as a subgroup. Similarly, we define the set of classes

$$\mathfrak{G}_2 := \{[\![y_1\boldsymbol{b}_1, (y_l\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2 : y_i \in \mathbb{Z}_q \ \forall i \in [k]\}.$$

Then $\mathfrak{G}_2$ is also an abelian group isomorphic to $\mathbb{Z}_q^k$. The set

$$\tilde{\mathfrak{G}}_2 := \{[\![y_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2 : y_1 \in \mathbb{Z}_q\}$$

is adopted by $\mathfrak{G}_2$ as a subgroup. We define the set

$$\mathfrak{G}_T := \{[z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \sum_{i\in[2:k]} z_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T : z_i \in \mathbb{Z}_q \ \forall i \in [k]\}.$$

This set is closed and associative under the group operation of $\mathbb{G}_T^{k\times k}$, with identity element $1_{\mathfrak{G}_T} = [\mathbf{0}]_T$. Every element $\mathfrak{h} = [z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \sum_{l\in[2:k]} z_l(\boldsymbol{a}_l \otimes \boldsymbol{b}_l)]_T$ has an inverse given by $\mathfrak{h}^{-1} = [-z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \sum_{l\in[2:k]} -z_l(\boldsymbol{a}_l \otimes \boldsymbol{b}_l)]_T$. Hence, $\mathfrak{G}_T$ is an abelian group isomorphic to $\mathbb{Z}_q^k$.

With the groups $\mathfrak{G}_1, \mathfrak{G}_2, \mathfrak{G}_T$ and the map from $\mathfrak{G}_1 \times \mathfrak{G}_2$ to $\mathfrak{G}_T$ induced by the inner product $\bullet$ of the dual encoding system, we introduce our splitting bilinear group generator as follows.

Definition 14 (Splitting Bilinear Group Generator). *Let $\mathcal{P}$ be a prime-order bilinear group generator and $\mathsf{IPE} = (\mathsf{SetUp}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{Dec})$ be an FH-IPFE scheme over groups generated by $\mathcal{P}$. For integers $k \geq 2$ and ρ, we define our splitting bilinear group generator $\mathcal{BG}_{\mathcal{P},\mathsf{IPE}}^{k,\rho}$ as an algorithm that on input λ does the following:*

1. *Let $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{SetUp}(1^\lambda, 1^{k+\rho})$. Include $\mathsf{bgd}_{\mathcal{P}}$ into pp.*
2. *Choose random $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{Z}_q^k$ for $i \in [k]$, such that the sets $\{\boldsymbol{a}_i\}_{i\in[k]}$ and $\{\boldsymbol{b}_i\}_{i\in[k]}$ are each a basis of $\mathbb{Z}_q^k$. Include $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i\in[k]}$ into sk.*
3. *For any tuple of coefficients $(x_1, \ldots, x_k) \in \mathbb{Z}_q^k$, denote by*
   - *$\mathcal{D}_{\mathbb{G}_1}^{k,\rho}(\mathsf{sk}, x_1\boldsymbol{a}_1, (x_l\boldsymbol{a}_l)_{l\in[2:k]})$ the distribution induced by sampling a random representative of the equivalence class $[\![x_1\boldsymbol{a}_1, (x_l\boldsymbol{a}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1$, i.e.*

$$\mathfrak{g} \leftarrow (\mathsf{DKGen}(\mathsf{sk}, [x_1a_{1,s}, \ldots, x_ka_{k,s}, \mathbf{0}_{1\times\rho}]_1))_{s\in[k]},$$

   - *$\mathcal{D}_{\mathbb{G}_2}^{k,\rho}(\mathsf{sk}, x_1\boldsymbol{b}_1, (x_l\boldsymbol{b}_l)_{l\in[2:k]})$ the distribution induced by sampling a random representative of the equivalence class $[\![y_1\boldsymbol{b}_1, (y_l\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2$, i.e.*

$$\mathfrak{h} \leftarrow (\mathsf{Enc}(\mathsf{sk}, [x_1b_{1,s}, \ldots, x_kb_{k,s}, \mathbf{0}_{1\times\rho}]_2))_{s\in[k]}.$$

4. *Parse*
   - *$\mathfrak{G}_1 := \{[\![x_1\boldsymbol{a}_1, (x_l\boldsymbol{a}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1 : x_i \in \mathbb{Z}_q \ \forall i \in [k]\}$,*
   - *$\mathfrak{G}_2 := \{[\![y_1\boldsymbol{b}_1, (y_l\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2 : y_i \in \mathbb{Z}_q \ \forall i \in [k]\}$,*
   - *$\mathfrak{G}_T := \{[z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \sum_{l\in[2:k]} z_l(\boldsymbol{a}_l \otimes \boldsymbol{b}_l)]_T : z_i \in \mathbb{Z}_q \ \forall i \in [k]\}$*

   as groups.
5. *Choose random full-rank matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{k\times k}$.*
6. *Generate sets of group generators for $\mathfrak{G}_1$ and $\mathfrak{G}_2$: For $i \in [k]$,*

– $\mathfrak{g}_{1,i} := [\![A_{i,1}\boldsymbol{a}_1, (A_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1$ *where*

$$\mathfrak{g}_{1,i} \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, A_{i,1}\boldsymbol{a}_1, (A_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}),$$

– $\mathfrak{g}_{2,i} := [\![B_{i,1}\boldsymbol{b}_1, (B_{i,l}\boldsymbol{b}_l), \mathbf{0}_{k\times\rho}]\!]_2$ *where*

$$\mathfrak{g}_{2,i} \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, B_{i,1}\boldsymbol{b}_1, (B_{i,l}\boldsymbol{b}_l)_{l\in[2:k]}).$$

7. *Parse*
   – $\tilde{\mathfrak{G}}_1 := \{[\![x_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1 : x_1 \in \mathbb{Z}_q\}$,
   – $\tilde{\mathfrak{G}}_2 := \{[\![y_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2 : y_1 \in \mathbb{Z}_q\}$

   as the subgroups of $\mathfrak{G}_1$ *and* $\mathfrak{G}_2$*, respectively.*
8. *Generate subgroup generators for* $\tilde{\mathfrak{G}}_1$ *and* $\tilde{\mathfrak{G}}_2$*: Let* $\tilde{x}, \tilde{y} \leftarrow\!\!\$\, \mathbb{Z}_q$*,*
   – $\tilde{\mathfrak{g}}_1 := [\![\tilde{x}\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1$ *where* $\tilde{\mathfrak{g}}_1 \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, \tilde{x}\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]})$,
   – $\tilde{\mathfrak{g}}_2 := [\![\tilde{y}\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2$ *where* $\tilde{\mathfrak{g}}_2 \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, \tilde{y}\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]})$.
9. *Generate randomizers for encodings: Let* $p(\lambda)$ *be the size of randomness used in functional keys and ciphertexts of* IPE[8]*, for each* $i \in [p(\lambda)]$*, compute*
   – *a functional-key randomizer* $\mathfrak{r}_{1,i} := [\![0\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1$ *where* $\mathfrak{r}_{1,i} \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, 0\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]})$,
   – *a ciphertext randomizer* $\mathfrak{r}_{2,i} := [\![0\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2$ *where* $\mathfrak{r}_{2,i} \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk},\ 0\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]})$.
10. *Define the map* $e : \mathfrak{G}_1 \times \mathfrak{G}_2 \to \mathfrak{G}_T$ *as follows: Given any two elements* $\mathfrak{g} = [\![x_1\boldsymbol{a}_1, (x_l\boldsymbol{a}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1 \in \mathfrak{G}_1$ *and* $\mathfrak{h} = [\![y_1\boldsymbol{b}_1, (y_l\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2 \in \mathfrak{G}_2$*, set*

$$e(\mathfrak{g}, \mathfrak{h}) := [\![x_1\boldsymbol{a}_1, (x_l\boldsymbol{a}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, (y_l\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times\rho}]\!]_2.$$

11. *Output* $\mathsf{bgd} = (\mathsf{pp}, (\mathfrak{G}_1, \{\mathfrak{g}_{1,i}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_1, \tilde{\mathfrak{g}}_1), (\mathfrak{G}_2, \{\mathfrak{g}_{2,i}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_2, \tilde{\mathfrak{g}}_2), \mathfrak{G}_T, q, e,\ \{\mathfrak{r}_{1,i}, \mathfrak{r}_{2,i}\}_{i\in[p(\lambda)]})$ *and* $\mathsf{td} = \mathsf{sk}$.

Next, we will show that the bilinear map generated by the generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ in Definition 14 is bilinear, non-degenerate, projecting and cancelling.

Theorem 2 (Bilinearity and Non-Degeneracy). *The map* $e : \mathfrak{G}_1 \times \mathfrak{G}_2 \to \mathfrak{G}_T$ *that is output by* $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$ *as given in Definition 14, is bilinear and non-degenerate.*

Proof. We proceed as follows.

[8] W.L.O.G., we assume that randomness used in functional keys and ciphertexts have the same length.

Bilinearity: For any $\mathfrak{g}_1 = [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$, $\mathfrak{h}_1 = [\![x'_1\boldsymbol{a}_1, \ldots, x'_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \in \mathfrak{G}_1$ and $\mathfrak{g}_2 = [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$, $\mathfrak{h}_2 = [\![y'_1\boldsymbol{b}_1, \ldots, y'_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \in \mathfrak{G}_2$, we have

$$
\begin{aligned}
e(\mathfrak{g}_1\mathfrak{h}_1, \mathfrak{g}_2\mathfrak{h}_2) &= ([\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 + [\![x'_1\boldsymbol{a}_1, \ldots, x'_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1) \\
&\quad \bullet ([\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 + [\![y'_1\boldsymbol{b}_1, \ldots, y'_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2) \\
&= ([\![(x_1 + x'_1)\boldsymbol{a}_1, \ldots, (x_k + x'_k)\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1) \\
&\quad \bullet ([\![(y_1 + y'_1)\boldsymbol{b}_1, \ldots, (y_k + y'_k)\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2) \\
&= [(x_1 + x'_1)(y_1 + y'_1)(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + (x_k + x'_k)(y_k + y'_k)(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= [x_1y_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x_ky_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&\quad + [x_1y'_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x_ky'_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&\quad + [x'_1y_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x'_ky_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&\quad + [x'_1y'_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x'_ky'_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&\quad + [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y'_1\boldsymbol{b}_1, \ldots, y'_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&\quad + [\![x'_1\boldsymbol{a}_1, \ldots, x'_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&\quad + [\![x'_1\boldsymbol{a}_1, \ldots, x'_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y'_1\boldsymbol{b}_1, \ldots, y'_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&= e(\mathfrak{g}_1, \mathfrak{g}_2)e(\mathfrak{g}_1, \mathfrak{h}_2)e(\mathfrak{h}_1, \mathfrak{g}_2)e(\mathfrak{h}_1, \mathfrak{h}_2).
\end{aligned}
$$

Non-Degeneracy: Given $\mathfrak{g}_1 = [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \in \mathfrak{G}_1$ and assume that $e(\mathfrak{g}_1, \mathfrak{g}_2) = 1$ for all $\mathfrak{g}_2 \in \mathfrak{G}_2$. We can choose $\mathfrak{g}_2 = [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$, which implies that

$$
\begin{aligned}
[\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 &= [x_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= [0(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + 0(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= 1_{\mathfrak{G}_T}.
\end{aligned}
$$

Since $\{\boldsymbol{a}_i \otimes \boldsymbol{b}_i\}_{i\in[k]}$ are linearly independent in $\mathbb{Z}_q^{k\times k}$, this implies $x_1 = \cdots = x_k = 0$. Hence, $\mathfrak{g}_1$ is the identity element $1_{\mathfrak{G}_1}$. □

Theorem 3 (Projecting). *We define* $\mathsf{proj}_1, \mathsf{proj}_2$ *and* proj_T *be mappings from* $\mathfrak{G}_1, \mathfrak{G}_2$ *and* $\mathfrak{G}_T$ *into themselves, respectively, as follows:*

– *For any* $[\![x_1\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \in \mathfrak{G}_1$,

$$
\mathsf{proj}_1([\![x_1\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1) := [\![0\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1.
$$

– *For any* $[\![y_1\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \in \mathfrak{G}_2$,

$$
\mathsf{proj}_2([\![y_1\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2) := [\![0\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2.
$$

– *For any* $[z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + z_2(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + z_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \in \mathfrak{G}_T$,

$$
\begin{aligned}
&\mathsf{proj}_T([z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + z_2(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + z_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T) \\
:= &[0(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + z_2(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + z_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T.
\end{aligned}
$$

Additionally, let $\tilde{\mathfrak{G}}_T := \{[z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + 0(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + 0(\boldsymbol{a}_k \otimes \boldsymbol{b}_k))]_T\}$ *be a subgroup of* $\mathfrak{G}_T$*. Then the generator* $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ *in Definition 14 is projecting with respect to* $(\mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_T, \tilde{\mathfrak{G}}_T)$.

Proof. By inspection, the mappings $\mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_3$ are homomorphisms. We prove the properties of annihilating and commuting as follows.

Annihilating: As $\mathsf{proj}_1, \mathsf{proj}_2, \mathsf{proj}_3$ project the coefficients of all dimensions except the first one, then $\tilde{\mathfrak{G}}_1$, $\tilde{\mathfrak{G}}_2$ and $\tilde{\mathfrak{G}}_T$ are contained in the kernels of these homomorphisms, respectively.

Commuting: For all $\mathfrak{g}_1 = [\![x_1\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$ and $\mathfrak{g}_2 = [\![y_1\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$, we have

$$\begin{aligned}
e(\mathsf{proj}_1(\mathfrak{g}_1), \mathsf{proj}_2(\mathfrak{g}_2)) &= [\![0\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![0\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&= [0(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + x_2y_2(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + x_ky_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= \mathsf{proj}_T([x_1y_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + x_2y_2(\boldsymbol{a}_2 \otimes \boldsymbol{b}_2) + \cdots + x_ky_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T) \\
&= \mathsf{proj}_T([\![x_1\boldsymbol{a}_1, x_2\boldsymbol{a}_2, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, y_2\boldsymbol{b}_2, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2) \\
&= \mathsf{proj}_T(e(\mathfrak{g}_1, \mathfrak{g}_2)).
\end{aligned}$$

□

Theorem 4 (Cancelling). *We define the following sets: For each $s \in [k]$,*

$$\begin{aligned}
\tilde{\mathfrak{G}}_{1,s} &:= \{[\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 : x_s \in \mathbb{Z}_q, x_j = 0\ \forall j \in [k] \setminus \{s\}\} \text{ and} \\
\tilde{\mathfrak{G}}_{2,s} &:= \{[\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 : y_s \in \mathbb{Z}_q, y_j = 0\ \forall j \in [k] \setminus \{s\}\}.
\end{aligned}$$

Then $\{\tilde{\mathfrak{G}}_{1,s}\}_{s\in[k]}$ are subgroups of $\mathfrak{G}_1$ and $\{\tilde{\mathfrak{G}}_{2,s}\}_{s\in[k]}$ are subgroups of $\mathfrak{G}_2$. By additionally outputting these subgroups, the generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ in Definition 14 satisfies the properties of group decomposing and cross cancelling.

Proof. We proceed as follows.

Group Decomposing: Each group $\tilde{\mathfrak{G}}_{1,s}$ represents uniquely the dimension spanned by $\boldsymbol{a}_s$ of $\mathfrak{G}_1$, therefore $\mathfrak{G}_1$ is decomposed into $\tilde{\mathfrak{G}}_{1,1} \times \cdots \times \tilde{\mathfrak{G}}_{1,k}$. By a similar argument, $\mathfrak{G}_2$ is decomposed into $\tilde{\mathfrak{G}}_{2,1} \times \cdots \times \tilde{\mathfrak{G}}_{2,k}$.

Cross Cancelling: Given any $i \neq j$, for all $\tilde{\mathfrak{g}}_{1,i} = [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \in \tilde{\mathfrak{G}}_{1,i}$ and $\tilde{\mathfrak{g}}_{2,j} = [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \in \tilde{\mathfrak{G}}_{2,j}$, since $x_j = y_i = 0$ and $x_s = y_s = 0$ for all $s \in [k] \setminus \{i, j\}$, then $x_sy_s = 0$ for all $s \in [k]$. We thus have

$$\begin{aligned}
e(\tilde{\mathfrak{g}}_{1,i}, \tilde{\mathfrak{g}}_{2,j}) &= [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&= [x_1y_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + x_ky_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= [0(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + 0(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T \\
&= 1_{\mathfrak{G}_T}
\end{aligned}$$

□

4.3 Verifiability for Membership of Source Groups

Our next step is to extend the splitting bilinear group generator from Definition 14 to support verifiability for membership in the source groups. This property is needed in applications such as our BARG construction in Section 6, as it prevents adversaries from crafting proof elements which do not belong to the source groups. To construct verifiable membership, we first define the languages

that capture membership of the source groups. Then, we show how to prove membership in those languages using an adaptation of the Groth-Sahai proof system [GS08] in soundness mode. Our verifiability mechanism is perfectly sound and critically relies on the additive homomorphism of Groth-Sahai proofs. Eventually, we obtain an extended splitting bilinear group generator. We defer all the technical material to Appendix A and state the main result below.

Theorem 5 (Verifiable Membership). *The bilinear group generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ given in Definition 19 supports publicly verifiable membership of source groups $\mathfrak{G}_1$ and $\mathfrak{G}_2$.*

4.4 Efficiency of the Bilinear Group Generators

We give concrete efficiency estimates for the bilinear group generators $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ from Definition 14 and Definition 19. All costs are measured in terms of the number of base group elements and the number of prime-order pairing operations. We recall the parameters of $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$:

Message space dimensions	$r = k + \rho$
FE randomness dimensions	$p(\lambda)$
Size of FE ciphertexts and keys	$q(r, \lambda)$
Dual-encoding size (elements in $\mathbb{G}_1$ or $\mathbb{G}_2$)	$k \cdot q(r, \lambda)$
Pairing cost (pairings $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$)	$k^2 \cdot q(r, \lambda)$

For the verifiable generator in Definition 19, the dual encoding is equipped with a Groth–Sahai proof system with the following parameters. Overall, the size of a splitting group element in the verifiable generator consists of the dual-encoding size plus the corresponding Groth–Sahai proof.

- Commitment groups: $\mathbb{V}_1 = \mathbb{G}_1^2$ and $\mathbb{V}_2 = \mathbb{G}_2^2$.
- For $\mathfrak{G}_1$ elements:
  - Message and randomness witness dimension: $m^\star = k(p(\lambda) + 1)$.
  - Secret-key witness dimension: $n^\star = (k^2 + p(\lambda))q(r, \lambda)$.
- For $\mathfrak{G}_2$ elements:
  - Message and randomness witness dimension: $n^\dagger = k(p(\lambda) + 1)$.
  - Secret-key witness dimension: $m^\dagger = (k^2 + p(\lambda))q(r, \lambda)$.
- Number of multi-scalar multiplication equations: $l^\star = l^\dagger = kq(r, \lambda)$, which coincides with the dual-encoding size.
- The proof for each $\mathfrak{G}_1$ element lives in $(\mathbb{V}_1^{m^\star}, \{\mathbb{V}_2^2, \mathbb{V}_1\}_{l^\star})$, corresponding to $2\big(k(p(\lambda)+1)+kq(r,\lambda)\big)$ elements in $\mathbb{G}_1$ and $4kq(r, \lambda)$ elements in $\mathbb{G}_2$.
- The proof for each $\mathfrak{G}_2$ element lives in $(\mathbb{V}_2^{n^\dagger}, \{\mathbb{V}_2, \mathbb{V}_1^2\}_{l^\dagger})$, corresponding to $2\big(k(p(\lambda)+1)+kq(r,\lambda)\big)$ elements in $\mathbb{G}_2$ and $4kq(r, \lambda)$ elements in $\mathbb{G}_1$.

Instantiation for BARG with $k = 2$ and $\rho = 6$. We consider the linearly homomorphic FH-IPFE scheme with the best known efficiency, namely Lin's construction [Lin17].

- $p(\lambda) = 1$.
- $q(r, \lambda) = 2(k + \rho) + 2 = 18$.
- Dual-encoding size: 36 $\mathbb{G}_1$ or $\mathbb{G}_2$ group elements.
- Pairing cost: 72 prime-order pairings $\mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- G-S proof for each $\mathfrak{G}_1$ element: 80 $\mathbb{G}_1$ elements, 144 $\mathbb{G}_2$ elements.
- G-S proof for each $\mathfrak{G}_2$ element: 80 $\mathbb{G}_2$ elements, 144 $\mathbb{G}_1$ elements.

The main bottleneck towards concrete efficiency is the overhead of the Groth-Sahai proofs. We leave it as an open problem to obtain more efficient verifiable membership mechanisms. We also note that if one is willing to accept *computational* soundness, there may be significantly more efficient solutions. Nevertheless, this has an impact on constructions. For instance, our BARG construction would only achieve computational somewhere extractability (whereas currently this holds statistically).

5 Hardness Assumptions over Splitting Bilinear Groups

In this section, we introduce an assumption over our dual-encoding-based bilinear groups that we call *bilateral subgroup decision exponent* (BSDE) assumption. This assumption generalizes the subgroup decision exponent assumption from [CEW25] by extending it from the two-subgroup setting to a parameterized setting with an arbitrary number k of subgroups. In Section 6, we show that a specific instantiation of the BSDE assumption enables our BARG construction. Then, by relying on the IND-CPA security of dual encodings, we show a reduction from the BSDE assumption to an assumption over prime-order groups that is secure in the generic bilinear group model. Below, we state the BSDE assumption, for which we instantiate our bilinear group generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ with $\rho = 3k$.

Definition 15 ((m,k)-Bilateral Subgroup Decision Exponent Problem). *Let $\mathcal{BG}^{k,3k}_{\mathcal{P},\mathsf{IPE}}$ be a bilinear group generator, $m \geq 0$ and $k \geq 2$ be integer parameters, we define the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ as follows:*

$$\mathsf{sk}, \mathsf{bgd} = \begin{cases} \mathsf{pp}, \mathsf{crs}, (\mathfrak{G}_1, \{\mathfrak{g}_{1,i}, \pi_{\mathfrak{g}_{1,i}}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_1, \tilde{\mathfrak{g}}_1, \pi_{\tilde{\mathfrak{g}}_1}), \\ (\mathfrak{G}_2, \{\mathfrak{g}_{2,i}, \pi_{\mathfrak{g}_{2,i}}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_2, \tilde{\mathfrak{g}}_2, \pi_{\tilde{\mathfrak{g}}_2}), \mathfrak{G}_T, q, e, \\ \{\mathfrak{r}_{1,i}, \pi_{\mathfrak{r}_{1,i}}, \mathfrak{r}_{2,i}, \pi_{\mathfrak{r}_{2,i}}\}_{i\in[p(\lambda)]} \end{cases} \leftarrow \mathcal{BG}^{k,3k}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$$

where sk is assumed to include the bases $\{\boldsymbol{a}_l, \boldsymbol{b}_l\}_{l\in[k]}$ generated in Item 2 and the distributions $\{\mathcal{D}^{k,3k}_{\mathbb{G}_1}, \mathcal{D}^{k,3k}_{\mathbb{G}_2}\}$ generated in Item 5 of Definition 19.

$\alpha \leftarrow\!\!\$\, \mathbb{Z}_q$
$r_l \leftarrow\!\!\$\, \mathbb{Z}_q, \boldsymbol{u}_l \leftarrow r_l \boldsymbol{a}_l$ $\quad \forall l \in [k]$
$t_l \leftarrow\!\!\$\, \mathbb{Z}_q, \boldsymbol{v}_l \leftarrow t_l \boldsymbol{b}_l$ $\quad \forall l \in [k]$
$[\![\alpha^j \boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi_{1,j} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_1}(\mathsf{sk}, \alpha^j \boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]})$ $\quad \forall j \in [m]$
$[\![\alpha^j \boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi_{2,j} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_2}(\mathsf{sk}, \alpha^j \boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]})$ $\quad \forall j \in [m]$
$[\![\boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi^{(0)}_{1,0} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_1}(\mathsf{sk}, \boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]})$
$[\![\boldsymbol{u}_1, (\boldsymbol{u}_l)_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi^{(1)}_{1,0} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_1}(\mathsf{sk}, \boldsymbol{u}_1, (\boldsymbol{u}_l)_{l\in[2:k]})$
$[\![\boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi^{(0)}_{2,0} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_2}(\mathsf{sk}, \boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]})$
$[\![\boldsymbol{v}_1, (\boldsymbol{v}_l)_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi^{(1)}_{2,0} \leftarrow \mathcal{D}^{k,3k}_{\mathbb{G}_2}(\mathsf{sk}, \boldsymbol{v}_1, (\boldsymbol{v}_l)_{l\in[2:k]})$
$\mathcal{D}_0$ *outputs*

$$\mathsf{D} := \mathsf{bgd}, \begin{cases} \{[\![\alpha^j \boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi_{1,j}\}_{j\in[m]}, [\![\boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi^{(0)}_{1,0} \\ \{[\![\alpha^j \boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi_{2,j}\}_{j\in[m]}, [\![\boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi^{(0)}_{2,0} \end{cases};$$

$\mathcal{D}_1$ *outputs*

$$\mathsf{D} := \mathsf{bgd}, \begin{cases} \{[\![\alpha^j \boldsymbol{u}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi_{1,j}\}_{j\in[m]}, [\![\boldsymbol{u}_1, (\boldsymbol{u}_l)_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_1, \pi^{(1)}_{1,0} \\ \{[\![\alpha^j \boldsymbol{v}_1, (\boldsymbol{0})_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi_{2,j}\}_{j\in[m]}, [\![\boldsymbol{v}_1, (\boldsymbol{v}_l)_{l\in[2:k]}, \boldsymbol{0}_{k\times 3k}]\!]_2, \pi^{(1)}_{2,0} \end{cases}.$$

The bilinear group generator $\mathcal{BG}^{k,3k}_{\mathcal{P},\mathsf{IPE}}$ satisfies the (m,k)-Bilateral Subgroup Decision Exponent (BSDE) assumption if given any parameters $m \geq 0$, $k \geq 2$, for every PPT adversary $\mathcal{A}$, the advantage defined as follows is bounded by $\mathsf{negl}(\lambda)$:

$$\mathsf{Adv}_{\mathsf{BSDE},\mathcal{A}}(\lambda,m,k) := |\Pr[\mathcal{A}(\mathsf{D})=1|\mathsf{D}\leftarrow\mathcal{D}_0] - \Pr[\mathcal{A}(\mathsf{D})=1|\mathsf{D}\leftarrow\mathcal{D}_1]|.$$

As highlighted in blue, the difference between two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ is that in $\mathcal{D}_0$ the challenge elements $[\![\boldsymbol{u}_1,(\boldsymbol{0})_{l\in[2:k]},\boldsymbol{0}_{k\times 3k}]\!]_1$ and $[\![\boldsymbol{v}_1,(\boldsymbol{0})_{l\in[2:k]},\boldsymbol{0}_{k\times 3k}]\!]_2$ are sampled from the subgroups $\tilde{\mathfrak{G}}_1$ and $\tilde{\mathfrak{G}}_2$, respectively. Meanwhile, in $\mathcal{D}_1$, the challenge elements $[\![\boldsymbol{u}_1,(\boldsymbol{u}_l)_{l\in[2:k]},\boldsymbol{0}_{k\times 3k}]\!]_1$ and $[\![\boldsymbol{v}_1,(\boldsymbol{v}_l)_{l\in[2:k]},\boldsymbol{0}_{k\times 3k}]\!]_2$ are sampled from the full groups $\mathfrak{G}_1$ and $\mathfrak{G}_2$, respectively.

Remark 6. When $m=0$, the BSDE assumption contains no powers of α. In this case, the BSDE assumption can be viewed as the standard Subgroup Decision assumption in splitting bilinear groups generated by $\mathcal{BG}^{k,3k}_{\mathcal{P},\mathsf{IPE}}$.

Next, we define an auxiliary assumption[9] in prime-order groups and prove its hardness in the generic bilinear group model.

Definition 16 ((m,k)-Bilateral Diffie-Hellman Decision Exponent Problem). *Let $\mathcal{P}$ be a prime-order bilinear group generator, $m \geq 0$ and $k \geq 1$ be integer parameters, we define the distribution $\mathcal{D}_0$ and $\mathcal{D}_1$ as follows:*

$\mathsf{bgd}_{\mathcal{P}} = (q,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,\bar{g}_1,\bar{g}_2,\hat{e}) \leftarrow \mathcal{P}(1^\lambda)$

$\alpha,r,t \leftarrow_{\$} \mathbb{Z}_q^*$	$\boldsymbol{a},\boldsymbol{b} \leftarrow_{\$} \mathbb{Z}_q^{*k}$	
$\boldsymbol{z}_w \leftarrow_{\$} \mathbb{Z}_q^{*k}$	$\boldsymbol{D}_w \leftarrow \mathsf{diag}(\boldsymbol{z}_w)$	$\forall w \in [3]$
$\boldsymbol{K} \leftarrow rt(\boldsymbol{a}\otimes\boldsymbol{b})\boldsymbol{D}_3$	$\boldsymbol{Z} \leftarrow_{\$} \mathbb{Z}_q^{k\times k}$	
$d,e \leftarrow_{\$} \mathbb{Z}_q^*$	$\boldsymbol{X},\boldsymbol{Y} \leftarrow_{\$} \mathbb{Z}_q^{k\times k}$	
$\boldsymbol{u}_j \leftarrow \alpha^j r\boldsymbol{a}$	$\boldsymbol{v}_j \leftarrow \alpha^j t\boldsymbol{b}$	$\forall j \in [m]$
$\boldsymbol{U}_j \leftarrow \alpha^j rt(\boldsymbol{a}\otimes\boldsymbol{b})\boldsymbol{D}_1$	$\boldsymbol{V}_j \leftarrow \alpha^j rt\boldsymbol{D}_2(\boldsymbol{a}\otimes\boldsymbol{b})$	$\forall j \in [m]$
$\boldsymbol{g} \leftarrow d\boldsymbol{a}$	$\boldsymbol{h} \leftarrow e\boldsymbol{b}$	
$\boldsymbol{G} \leftarrow dt(\boldsymbol{a}\otimes\boldsymbol{b})\boldsymbol{D}_1$	$\boldsymbol{H} \leftarrow er\boldsymbol{D}_2(\boldsymbol{a}\otimes\boldsymbol{b})$	

$$\mathcal{D}_0 \text{ outputs } \mathsf{D} := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1,[\boldsymbol{D}_2^{-1}]_1,[\boldsymbol{D}_3]_1,\{[\boldsymbol{u}_j]_1,[\boldsymbol{U}_j]_1\}_{j\in[m]},[\boldsymbol{g}]_1,[\boldsymbol{G}]_1,[\boldsymbol{K}]_1 \\ [\boldsymbol{D}_1^{-1}]_2,[\boldsymbol{D}_2]_2,[\boldsymbol{D}_3^{-1}]_2,\{[\boldsymbol{v}_j]_2,[\boldsymbol{V}_j]_2\}_{j\in[m]},[\boldsymbol{h}]_2,[\boldsymbol{H}]_2 \end{cases}$$

$$\mathcal{D}_1 \text{ outputs } \mathsf{D} := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1,[\boldsymbol{D}_2^{-1}]_1,[\boldsymbol{D}_3]_1,\{[\boldsymbol{u}_j]_1,[\boldsymbol{U}_j]_1\}_{j\in[m]},[\boldsymbol{g}]_1,[\boldsymbol{X}]_1,[\boldsymbol{Z}]_1 \\ [\boldsymbol{D}_1^{-1}]_2,[\boldsymbol{D}_2]_2,[\boldsymbol{D}_3^{-1}]_2,\{[\boldsymbol{v}_j]_2,[\boldsymbol{V}_j]_2\}_{j\in[m]},[\boldsymbol{h}]_2,[\boldsymbol{Y}]_2 \end{cases}$$

The bilinear group generator $\mathcal{P}$ is said to satisfy the (m,k)-Bilateral Diffie-Hellman Decision Exponent assumption if given any parameters $m \geq 0$ and $k \geq 1$, for every PPT adversary $\mathcal{A}$, the advantage defined as follows is bounded by $\mathsf{negl}(\lambda)$:

$$\mathsf{Adv}_{\mathsf{BDHDE},\mathcal{A}}(\lambda,m,k) := |\Pr[\mathcal{A}(\mathsf{D})=1|\mathsf{D}\leftarrow\mathcal{D}_0] - \Pr[\mathcal{A}(\mathsf{D})=1|\mathsf{D}\leftarrow\mathcal{D}_1]|.$$

As highlighted in blue, the difference between two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ is that in $\mathcal{D}_0$ the challenge elements $[\boldsymbol{G}]_1$, $[\boldsymbol{K}]_1$, and $[\boldsymbol{H}]_2$ are sampled in correlation with other elements. In contrast,

[9] For the motivation behind the formulation of this assumption, see the hardness proof of the BDSE assumption in Theorem 7.

in $\mathcal{D}_1$, the challenge elements $[\boldsymbol{X}]_1$, $[\boldsymbol{Z}]_1$, and $[\boldsymbol{Y}]_2$ are sampled independently and uniformly at random.

Theorem 6 (Hardness of Bilateral Diffie-Hellman Decision Exponent Assumption). *For any integers $m \geq 0$ and $k \geq 1$, $\mathcal{P}$ satisfies the (m,k)-Bilateral Diffie-Hellman Decision Exponent assumption in the generic bilinear group model.*

We defer the proof of the theorem to Appendix B.1. The hardness of our main assumption also critically relies on the indistinguishability of the dual encoding system, as stated in the following theorem.

Theorem 7 (Hardness of Bilateral Subgroup Decision Exponent Assumption). *For any integers $m \geq 0$ and $k \geq 2$, $\mathcal{BG}^{k,3k}_{\mathcal{P},\mathsf{IPE}}$ satisfies the (m,k)-Bilateral Subgroup Decision Exponent assumption.*

More precisely, for any adversary $\mathcal{A}_{\mathsf{BSDE}}$ that breaks the (m,k)-BSDE assumption, there exist adversaries

- *$\mathcal{B}_{\mathsf{GS,crs}}$ that breaks the CRS indistinguishability of the Groth-Sahai proof system,*
- *$\mathcal{B}_{\mathsf{DE}}$ that breaks the IND-CPA security of the dual encoding system,*
- *$\mathcal{B}_{\mathsf{BDHDE}}$ that breaks the (m,k)-Bilateral Diffie-Hellman Decision Exponent assumption*

such that

$$\mathsf{Adv}_{\mathcal{A}_{\mathsf{BSDE}}}(\lambda,m,k) \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{GS,crs}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{DE}}}(\lambda) + (k+1)\mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda,m,k) + \mathsf{negl}(\lambda).$$

Proof. As a first step, we explicitly write out the exponents of the generators $\{(\mathfrak{g}_{1,i},\mathfrak{g}_{2,i})_{i\in[k]},\tilde{\mathfrak{g}}_1,\tilde{\mathfrak{g}}_2\}$ and the randomizers $\{\mathfrak{r}_{1,i},\mathfrak{r}_{2,i}\}_{i\in[p(\lambda)]}$ in the group description bgd, and list all encodings that are sampled from the distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, along with their respective zero-knowledge proofs, as follows:

$$\mathsf{D} = \begin{cases}
\mathfrak{g}_{1,i} := [\![\ x_{i,1}\boldsymbol{a}_1,\ (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1 \ \forall i\in[k] \\
\tilde{\mathfrak{g}}_1 := [\![\ \tilde{x}_1\boldsymbol{a}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{r}_{1,h} := [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1 \ \forall h\in[p(\lambda)] \\
\mathfrak{u}_j := [\![\ \alpha^j r_1\boldsymbol{a}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1 \ \forall j\in[m] \\
\mathfrak{t}_1 := [\![\ r_1\boldsymbol{a}_1,\ (\mathsf{b}r_l\boldsymbol{a}_l)_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1 \\
\\
\mathfrak{g}_{2,i} := [\![\ y_{i,1}\boldsymbol{b}_1,\ (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2 \ \forall i\in[k] \\
\tilde{\mathfrak{g}}_2 := [\![\ \tilde{y}_1\boldsymbol{b}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{r}_{2,h} := [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2 \ \forall h\in[p(\lambda)] \\
\mathfrak{v}_j := [\![\ \alpha^j t_1\boldsymbol{b}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2 \ \forall j\in[m] \\
\mathfrak{t}_2 := [\![\ t_1\boldsymbol{b}_1,\ (\mathsf{b}t_l\boldsymbol{b}_l)_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2 \\
\\
\{\pi_{\mathfrak{g}_{1,i}},\pi_{\mathfrak{g}_{2,i}}\}_{i\in[k]},\pi_{\tilde{\mathfrak{g}}_1},\pi_{\tilde{\mathfrak{g}}_2},\{\pi_{\mathfrak{r}_{1,h}},\pi_{\mathfrak{r}_{2,h}}\}_{h\in[p(\lambda)]},\{\pi_{\mathfrak{u}_j},\pi_{\mathfrak{v}_j}\}_{j\in[m]},\pi_{\mathfrak{t}_1},\pi_{\mathfrak{t}_2}
\end{cases}$$

where $\{x_{i,l},y_{i,l}\}_{i,l\in[k]},\tilde{x}_1,\tilde{y}_1,\alpha,\{r_l,t_l\}_{l\in[k]}$ are independently and uniformly distributed over $\mathbb{Z}_q$, and the bit $\mathsf{b}\in\{0,1\}$ determines whether dimensions other than the first are present in $\mathfrak{t}_1$ and $\mathfrak{t}_2$. This means when $\mathsf{b}=0$, D is output by $\mathcal{D}_0$. When $\mathsf{b}=1$, D is output by $\mathcal{D}_1$.

Now that all encodings of the (m,k)-BSDE problem are explicitly written, given any adversary $\mathcal{A}$, we proceed with the second step of the proof via a hybrid argument on D, for which we define five security games $\mathsf{G}_0,\ldots,\mathsf{G}_4$, along with modifications between each two games highlighted in gray. Furthermore, the advantage of $\mathcal{A}$ in each game G_i is denoted by $\mathsf{Adv}_{\mathsf{G}_i,\mathcal{A}}(\lambda)$. We describe the game hops and the transitions but defer the proofs of some technical lemmas to Appendix B.2.

Game G_0: This game corresponds to the (m,k)-BSDE problem, where $\mathcal{A}$ guesses the challenge bit b from the output D of the distribution $\mathcal{D}_{\mathsf{b}}$.

Game G_1: The output D is replaced by the output D_1, in which the common reference string crs is generated in the zero-knowledge mode of the membership proof system, i.e. $\mathsf{crs} \leftarrow \mathsf{SetupTd}(1^\lambda, \mathsf{param}_{\mathfrak{G}_1}, \mathsf{param}_{\mathfrak{G}_2})$, where $\mathsf{param}_{\mathfrak{G}_1}$ and $\mathsf{param}_{\mathfrak{G}_2}$ are parameters that define the languages of source-group memberships. This setup algorithm and the parameters are included in the recast Groth-Sahai proof, which is specified in Appendix A.
Moreover, all proofs, i.e. $\{\pi_{\mathfrak{g}_{1,i}},\pi_{\mathfrak{g}_{2,i}}\}_{i\in[k]}$, $\pi_{\tilde{\mathfrak{g}}_1}$, $\pi_{\tilde{\mathfrak{g}}_2}$, $\{\pi_{\mathfrak{r}_{1,h}},\pi_{\mathfrak{r}_{2,h}}\}_{h\in[p(\lambda)]}$ in bgd and $\{\pi_{\mathfrak{u}_j},\pi_{\mathfrak{v}_j}\}_{j\in[m]}$, $\pi_{\mathfrak{t}_1}$, $\pi_{\mathfrak{t}_2}$ generated from sk and b are simulated by the zero-knowledge simulator. The indistinguishability between G_0 and G_1 follows directly by the CRS indistinguishability under the Decisional Diffie-Hellman assumption of the Groth-Sahai proof system in [GS08]. Therefore, it holds that $|\mathsf{Adv}_{\mathsf{G}_0,\mathcal{A}}(\lambda)-\mathsf{Adv}_{\mathsf{G}_1,\mathcal{A}}(\lambda)|\le\mathsf{Adv}_{\mathcal{B}_{\mathsf{GS},\mathsf{crs}}}(\lambda)$.

Game G_2: The output D_1 is replaced by the output D_2, obtained by the following changes to the encodings.

$$
\begin{aligned}
\mathfrak{g}_{1,i} &= [\![\; x_{i,1}\boldsymbol{a}_1,\ (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]},\ (x_{i,1}t_1\boldsymbol{T}_1+\mathsf{b}\boldsymbol{A}_i)\boldsymbol{D}_1,\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k}\;]\!]_1\\
\tilde{\mathfrak{g}}_1 &= [\![\; \tilde{x}_1\boldsymbol{a}_1,\ (\boldsymbol{0})_{l\in[2:k]},\ \tilde{x}_1t_1\boldsymbol{T}_1\boldsymbol{D}_1,\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k}\;]\!]_1\\
\mathfrak{r}_{1,h} &= [\![\; \boldsymbol{0},\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k}\;]\!]_1\\
\mathfrak{u}_j &= [\![\; \alpha^j r_1\boldsymbol{a}_1,\ (\boldsymbol{0})_{l\in[2:k]},\ \alpha^j r_1t_1\boldsymbol{T}_1\boldsymbol{D}_1,\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k}\;]\!]_1\\
\mathfrak{t}_1 &= [\![\; \boldsymbol{0},\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{D}_2^{-1},\ (r_1t_1\boldsymbol{T}_1+\mathsf{b}\boldsymbol{C})\boldsymbol{D}_3\;]\!]_1
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{g}_{2,i} &= [\![\; y_{i,1}\boldsymbol{b}_1,\ (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ (y_{i,1}r_1\boldsymbol{T}_1+\mathsf{b}\boldsymbol{B}_i)^\intercal\boldsymbol{D}_2,\ \boldsymbol{0}_{k\times k}\;]\!]_2\\
\tilde{\mathfrak{g}}_2 &= [\![\; \tilde{y}_1\boldsymbol{b}_1,\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ \tilde{y}_1r_1\boldsymbol{T}_1^\intercal\boldsymbol{D}_2,\ \boldsymbol{0}_{k\times k}\;]\!]_2\\
\mathfrak{r}_{2,h} &= [\![\; \boldsymbol{0},\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{0}_{k\times k}\;]\!]_2\\
\mathfrak{v}_j &= [\![\; \alpha^j t_1\boldsymbol{b}_1,\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{0}_{k\times k},\ \alpha^j r_1t_1\boldsymbol{T}_1^\intercal\boldsymbol{D}_2,\ \boldsymbol{0}_{k\times k}\;]\!]_2\\
\mathfrak{t}_2 &= [\![\; \boldsymbol{0},\ (\boldsymbol{0})_{l\in[2:k]},\ \boldsymbol{D}_1^{-1},\ \boldsymbol{0}_{k\times k},\ \boldsymbol{D}_3^{-1}\;]\!]_2
\end{aligned}
$$

where $\boldsymbol{T}_l=\boldsymbol{a}_l\otimes\boldsymbol{b}_l$ for all $l\in[k]$, $\boldsymbol{A}_i=\sum_{l\in[2:k]}x_{i,l}t_l\boldsymbol{T}_l$, $\boldsymbol{B}_i=\sum_{l\in[2:k]}y_{i,l}r_l\boldsymbol{T}_l$ for all $i\in[k]$ and $\boldsymbol{C}=\sum_{l\in[2:k]}r_lt_l\boldsymbol{T}_l$. Furthermore, $\boldsymbol{D}_w=\mathsf{diag}(\boldsymbol{z}_w)$ where $\boldsymbol{z}_w\leftarrow\!\!\$\ \mathbb{Z}_q^{*k}$ for all $w\in[3]$.
The goal of this hybrid is to additively mask all b-dependent terms inside the dual encodings without modifying any pairing results between $\mathfrak{G}_1$ elements in $\{\{\mathfrak{g}_{1,i}\}_{i\in[k]},\tilde{\mathfrak{g}}_1,\{\mathfrak{r}_{1,h}\}_{h\in[p(\lambda)]},\{\mathfrak{u}_j\}_{j\in[m]},\mathfrak{t}_1\}$ and $\mathfrak{G}_2$ elements in $\{\{\mathfrak{g}_{2,i}\}_{i\in[k]},\tilde{\mathfrak{g}}_2,\{\mathfrak{r}_{2,h}\}_{h\in[p(\lambda)]},\{\mathfrak{v}_j\}_{j\in[m]},\mathfrak{t}_2\}$.
Notably, as part of the proof strategy, we additionally introduce the matrices $\boldsymbol{D}_1$, $\boldsymbol{D}_2$, $\boldsymbol{D}_3$, together with their inverses, into the distribution output D_2. This is because, in order to ultimately reduce the assumption to hardness in the generic group model, we need to consider these terms outside the dual-encoding layer, e.g., as prime-order group elements. Since the adversary then gains access

to additional pairings involving these elements (beyond inner-product ones), the matrices serve as pairing predicates that limit the leakage from such pairings, which could otherwise help the adversary distinguish between the distributions.

By Lemma 8, the indistinguishability between G_1 and G_2 follows by the IND-CPA security of the dual encoding system. Furthermore, it holds that $|\mathsf{Adv}_{\mathsf{G}_1,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_2,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{DE}}}(\lambda)$.

Game G_3: The output D_2 is replaced by the output D_3, which is obtained by modifying the sampling of the underlying scalars as follows:

$$\boldsymbol{a}_l, \boldsymbol{b}_l \leftarrow\!\$\ \mathbb{Z}_q^k \ \forall l \in [2:k], \qquad \boldsymbol{a}_1, \boldsymbol{b}_1 \leftarrow\!\$\ \mathbb{Z}_q^{*k}, \qquad \alpha \leftarrow\!\$\ \mathbb{Z}_q^*$$
$$x_{i,1}, y_{i,1} \leftarrow\!\$\ \mathbb{Z}_q^* \ \forall i \in [k], \qquad \tilde{x}_1, \tilde{y}_1 \leftarrow\!\$\ \mathbb{Z}_q^*, \qquad r_1, t_1 \leftarrow\!\$\ \mathbb{Z}_q^*$$

The goal of this hybrid is to enable a reduction to the $(m, k+1, k)$-BDHDE assumption, which requires the relevant scalars to be sampled uniformly from $\mathbb{Z}_q^*$ instead of $\mathbb{Z}_q$. The indistinguishability between G_2 and G_3 follows directly by these statistical changes, hence $|\mathsf{Adv}_{\mathsf{G}_2,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_3,\mathcal{A}}(\lambda)| \leq \mathsf{negl}(\lambda)$.

Game G_4: The output D_3 is replaced by the output D_4, obtained by the following changes to the encodings.

$$\begin{aligned}
\mathfrak{g}_{1,i} &= [\![\ x_{i,1}\boldsymbol{a}_1, (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}, \boldsymbol{X}_i\boldsymbol{D}_1, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]\!]_1 \\
\tilde{\mathfrak{g}}_1 &= [\![\ \tilde{x}_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{X}_{k+1}\boldsymbol{D}_1, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{r}_{1,h} &= [\![\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{u}_j &= [\![\ \alpha^j r_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \alpha^j r_1 t_1 \boldsymbol{T}_1\boldsymbol{D}_1, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{t}_1 &= [\![\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{D}_2^{-1}, \boldsymbol{Z}\boldsymbol{D}_3]\!]_1
\end{aligned}$$

$$\begin{aligned}
\mathfrak{g}_{2,i} &= [\![\ y_{i,1}\boldsymbol{b}_1, (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{Y}_i^\intercal\boldsymbol{D}_2, \mathbf{0}_{k\times k}]\!]_2 \\
\tilde{\mathfrak{g}}_2 &= [\![\ \tilde{y}_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{Y}_{k+1}^\intercal\boldsymbol{D}_2, \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{r}_{2,h} &= [\![\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{v}_j &= [\![\ \alpha^j t_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \alpha^j t_1 r_1 \boldsymbol{T}_1^\intercal\boldsymbol{D}_2, \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{t}_2 &= [\![\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{D}_1^{-1}, \mathbf{0}_{k\times k}, \boldsymbol{D}_3^{-1}]\!]_2
\end{aligned}$$

where $\boldsymbol{T}_1 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1$, $\boldsymbol{X}_i \leftarrow\!\$\ \mathbb{Z}_q^{k\times k}$, $\boldsymbol{Y}_i \leftarrow\!\$\ \mathbb{Z}_q^{k\times k}$ for all $i \in [k+1]$ and $\boldsymbol{Z} \leftarrow\!\$\ \mathbb{Z}_q^{k\times k}$. Furthermore, $\boldsymbol{D}_w = \mathsf{diag}(\boldsymbol{z}_w)$ where $\boldsymbol{z}_w \leftarrow\!\$\ \mathbb{Z}_q^{*k}$ for all $w \in [3]$.

To switch the highlighted elements to uniform randomness, we introduce a parameterized extension of the (m,k)-BDHDE assumption, called the (m,τ,k)-BDHDE assumption. The parameter $\tau \geq 0$ allows additional matrices $[\boldsymbol{G}]$ and $[\boldsymbol{H}]$ to be switched to uniform randomness. This assumption is formally defined in Definition 20. Moreover, hardness for $\tau = 1$, which coincides the original (m,k)-BDHDE assumption, implies hardness for arbitrary τ via the reduction in Theorem 11.

By the modification, no information about the challenge bit b is revealed in this game, hence $\mathsf{Adv}_{\mathsf{G}_4} = 0$.

By Lemma 9, the indistinguishability between G_3 and G_4 follows by the (m,τ,k)-BDHDE assumption with the parameter $\tau = k+1$. Furthermore, it holds that $|\mathsf{Adv}_{\mathsf{G}_3,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_4,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda, m, k+1, k)$. By Theorem 11, $|\mathsf{Adv}_{\mathsf{G}_3,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_4,\mathcal{A}}(\lambda)| \leq (k+1)\mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda, m, k)$.

Taking all transitions from G_0 to G_4,

$$\begin{aligned}\mathsf{Adv}_{\mathcal{A}_{\mathsf{BSDE}}}(\lambda, m, k) &\leq \sum_{i\in[4]} \left|\mathsf{Adv}_{\mathsf{G}_{i-1},\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_i,\mathcal{A}}(\lambda)\right| \\ &\leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{GS},\mathsf{crs}}}(\lambda) + \mathsf{Adv}_{\mathcal{B}_{\mathsf{DE}}}(\lambda) + (k+1)\mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda, m, k) + \mathsf{negl}(\lambda),\end{aligned}$$

completes the proof. □

6 BARG with Linear-Size CRS from Prime-Order Groups

In this section, we build a Batch Argument for NP with a linear-size CRS from prime-order bilinear groups. Our strategy is to replicate the subgroup structure in the BARG by Chen, Elias and Wu [CEW25] using our splitting bilinear group generator with verifiable membership (Definition 19)—note that membership verifiability is required for preventing proof forgery attempts that rely on outputting elements outside the groups. We prove that our BARG achieves CRS indistinguishability if the BSDE assumption (Definition 15) holds, as well as statistical somewhere extractability. We present our construction next.

$\underline{\mathsf{Setup}(1^\lambda, 1^m)}$: on input the security parameter λ and a number of instances m, the setup algorithm proceeds as follows:

1. Run $(\mathsf{bgd}, \mathsf{sk}) \leftarrow \mathcal{BG}^{2,6}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$;
2. Take the vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ that are sampled and included in sk as in Item 2 of Definition 19;
3. Sample $r_1, t_1, r_2, t_2 \leftarrow\!\!\$\, \mathbb{Z}_q$ and assign $\boldsymbol{u}_1 = r_1\boldsymbol{a}_1, \boldsymbol{v}_1 = t_1\boldsymbol{b}_1, \boldsymbol{u}_2 = r_2\boldsymbol{a}_2, \boldsymbol{v}_2 = t_2\boldsymbol{b}_2$;
4. Sample $\alpha \leftarrow\!\!\$\, \mathbb{Z}_q$.
5. For each $i \in [0, m-1]$, verifiably sample[10] and assign $\mathfrak{a}_i = [\![\alpha^i\boldsymbol{u}_1, \boldsymbol{0}]\!]_1$, $\mathfrak{b}_i = [\![\alpha^i\boldsymbol{v}_1, \boldsymbol{0}]\!]_2$[11].
6. Define the polynomial $\zeta(x) := \prod_{i\in[m]}(x - i) \in \mathbb{Z}_q[x]$, verifiably sample and assign $\mathfrak{z} = [\![\zeta(\alpha) \cdot \boldsymbol{u}_1, \boldsymbol{0}]\!]_1$.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}, (\mathfrak{a}_0, \mathfrak{b}_0), \ldots, (\mathfrak{a}_{m-1}, \mathfrak{b}_{m-1}), \mathfrak{z})$.

$\underline{\mathsf{Prove}(\mathsf{crs}, \mathcal{C}, \{(\mathtt{x}_i, \mathtt{w}_i)\}_{i\in[m]})}$: on input the crs, a Boolean circuit $\mathcal{C} : \{0,1\}^{\ell_\mathtt{x}} \times \{0,1\}^{\ell_\mathtt{w}} \to \{0,1\}$, instances $\mathtt{x}_1, \ldots, \mathtt{x}_m \in \{0,1\}^{\ell_\mathtt{x}}$, and witnesses $\mathtt{w}_1, \ldots, \mathtt{w}_m \in \{0,1\}^{\ell_\mathtt{w}}$, let $\ell_\mathcal{C}$ be the number of wires (and also gates) in $\mathcal{C}$. For each $i \in [m]$ and $j \in [\ell_\mathcal{C}]$, let $w_{i,j}$ denote the value of the jth wire in $C(\mathtt{x}_i, \mathtt{w}_i)$. The prover constructs the proof as follows:

1. **Interpolate wire values**: For each $j \in [\ell_\mathcal{C}]$, let $\Phi_j \in \mathbb{Z}_q[x]$ be the unique polynomial of degree at most $m-1$ where $\Phi_j(i) = w_{i,j}$ for all $i \in [m]$. Compute Φ_j using the Lagrange interpolation.
2. **Commit wire values**: For each $j \in [\ell_\mathtt{x} + 1, \ell_\mathcal{C}]$, let $\Phi_j(x) = \sum_{i\in[0,m-1]} \phi_{j,i}x^i$. Compute $\mathfrak{u}_j = \prod_{i\in[0,m-1]} \mathfrak{a}_i^{\phi_{j,i}}$ and $\mathfrak{v}_j = \prod_{i\in[0,m-1]} \mathfrak{b}_i^{\phi_{j,i}}$.
3. **Validity of witness-wire assignments**: For each $j \in [\ell_\mathtt{x} + 1, \ell_{\mathtt{x},\mathtt{w}}]$, define the quotient polynomial $Q_j(x) := \frac{\Phi_j^2(x) - \Phi_j(x)}{\zeta(x)}$. Write $Q_j(x) = \sum_{i\in[0,m-2]} q_{j,i}x^i$. If the polynomial Q_j does not exist or cannot be written in this form, then abort with output $\perp$. Compute $\mathfrak{p}_j = \prod_{i\in[0,m-2]} \mathfrak{b}_i^{q_{j,i}}$.

[10] In this construction, we assume the respective distributions $\mathcal{D}^{k,\rho}_{\mathbb{G}_1}$ and $\mathcal{D}^{k,\rho}_{\mathbb{G}_2}$ can be generated from sk and output membership proofs as in Item 5 of Definition 19.

[11] For notational convenience, we omit in this construction the padded zero blocks $\boldsymbol{0}_{2\times 6}$ in the dual encodings that ensure the BSDE assumption holds.

4. **Validity of gate computation**: For each internal NAND gate $G_j = (j_1, j_2, j)$ for $j \in [\ell_{\mathtt{x},\mathtt{w}} + 1, \ell_{\mathcal{C}}]$, where $j_1, j_2 \le j$ are the input wires and j is the output one, compute the polynomial $R_j(x) := \frac{1-\Phi_j(x)-\Phi_{j_1}(x)\Phi_{j_2}(x)}{\zeta(x)}$. Write $R_j(x) = \sum_{i\in[0,m-2]} r_{j,i}x^i$. If the polynomial R_j does not exist or cannot be written in this form, then abort with output $\perp$. Compute $\mathfrak{q}_j = \prod_{i\in[0,m-2]} \mathfrak{b}_i^{r_{j,i}}$.

Finally, output $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$.

$\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi)$: on input crs, a circuit $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x}}} \times \{0,1\}^{\ell_{\mathtt{w}}} \to \{0,1\}$, a batch of statements $\{\mathtt{x}_i\}_{i\in[m]}$, and $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$, the verification algorithm proceeds as follows:

1. **Commit to statements**: For each $j \in [\ell_{\mathtt{x}}]$, let $\Phi_j \in \mathbb{Z}_q[x]$ be the unique polynomial of degree at most $m-1$ where $\Phi_j(i) = \mathtt{x}_{i,j}$ for all $i \in [m]$. Compute Φ_j using the Lagrange interpolation. Write $\Phi_j(x) = \sum_{i\in[0,m-1]} \phi_{j,i}x^i$ and compute $\mathfrak{u}_j = \prod_{i\in[0,m-1]} \mathfrak{a}_i^{\phi_{j,i}}$ and $\mathfrak{v}_j = \prod_{i\in[0,m-1]} \mathfrak{b}_i^{\phi_{j,i}}$.
2. **Validity of source-group memberships**: Given the verification algorithms $\mathsf{Ver}_{\mathfrak{G}_1}$ and $\mathsf{Ver}_{\mathfrak{G}_2}$ included in bgd through crs, check that
   - for each $j \in [\ell_{\mathtt{x}} + 1, \ell_{\mathcal{C}}]$, $\mathsf{Ver}_{\mathfrak{G}_1}(\mathfrak{u}_j) = \mathsf{Ver}_{\mathfrak{G}_2}(\mathfrak{v}_j) = 1$;
   - for each $j \in [\ell_{\mathtt{x}} + 1, \ell_{\mathtt{x},\mathtt{w}}]$, $\mathsf{Ver}_{\mathfrak{G}_2}(\mathfrak{p}_j) = 1$;
   - for each $j \in [\ell_{\mathtt{x},\mathtt{w}} + 1, \ell_{\mathcal{C}}]$, $\mathsf{Ver}_{\mathfrak{G}_2}(\mathfrak{q}_j) = 1$.
3. **Check validity of witness-wire assignments**: For each $j \in [\ell_{\mathtt{x}} + 1, \ell_{\mathtt{x},\mathtt{w}}]$, check that $e(\mathfrak{u}_j, \mathfrak{v}_j) = e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{p}_j) \wedge e(\mathfrak{u}_j, \mathfrak{v}_j) = e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{p}_j)$.
4. **Check validity of gate computations**: For each NAND gate $G_j = (j_1, j_2, j)$ for $j \in [\ell_{\mathtt{x},\mathtt{w}} + 1, \ell_{\mathcal{C}}]$, check that

$$e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{q}_j) = e(\mathfrak{a}_0, \mathfrak{b}_0) \ \wedge \ e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{q}_j) = e(\mathfrak{a}_0, \mathfrak{b}_0).$$

5. **Check output satisfiability**: Check that $\mathfrak{u}_{\ell_{\mathcal{C}}} = \mathfrak{a}_0$ and $\mathfrak{v}_{\ell_{\mathcal{C}}} = \mathfrak{b}_0$.

Output 1 if all checks pass and 0 otherwise.

$\mathsf{SetupTd}(1^\lambda, 1^m, i^*)$: The trapdoor setup proceeds as follows:

1. Run $(\mathsf{bgd}, \mathsf{sk}) \leftarrow \mathcal{BG}^{2,6}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$;
2. Take the vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2$ that are sampled and included in sk as in Item 2 of Definition 19;
3. Sample $r_1, t_1, r_2, t_2 \leftarrow\!\!\$\ \mathbb{Z}_q$ and assign $\boldsymbol{u}_1 = r_1\boldsymbol{a}_1, \boldsymbol{v}_1 = t_1\boldsymbol{b}_1, \boldsymbol{u}_2 = r_2\boldsymbol{a}_2, \boldsymbol{v}_2 = t_2\boldsymbol{b}_2$;
4. Sample $\alpha \leftarrow\!\!\$\ \mathbb{Z}_q$.
5. For each $i \in [0, m-1]$, verifiably sample and assign $\mathfrak{a}_i = [\![\alpha^i\boldsymbol{u}_1, (i^*)^i\boldsymbol{u}_2]\!]_1$, $\mathfrak{b}_i = [\![\alpha^i\boldsymbol{v}_1, (i^*)^i\boldsymbol{v}_2]\!]_2$.
6. Define the polynomial $\zeta(x) := \prod_{i\in[m]}(x - i) \in \mathbb{Z}_q[x]$, verifiably sample and assign $\mathfrak{z} = [\![\zeta(\alpha) \cdot \boldsymbol{u}_1, \boldsymbol{0}]\!]_1$.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}, (\mathfrak{a}_0, \mathfrak{b}_0), \ldots, (\mathfrak{a}_{m-1}, \mathfrak{b}_{m-1}), \mathfrak{z})$ and the extraction trapdoor $\mathsf{td} = [\![\boldsymbol{0}, \boldsymbol{v}_2]\!]_2$. The parts highlighted in blue indicate the differences between Setup and $\mathsf{SetupTd}$.

$\mathsf{Ext}(\mathsf{td}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi)$: On input the trapdoor $\mathsf{td} = [\![\boldsymbol{0}, \boldsymbol{v}_2]\!]_2$, the circuit $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x}}} \times \{0,1\}^{\ell_{\mathtt{w}}} \to \{0,1\}$, a batch of statements $\{\mathtt{x}_i\}_{i\in[m]}$, and a proof $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$, the extraction algorithm sets wire value $w^*_j = 0$ if $e(\mathfrak{u}_j, [\![\boldsymbol{0}, \boldsymbol{v}_2]\!]_2) = [\boldsymbol{0}_{2\times 2}]_T$ and $w^*_j = 1$ otherwise for all $j \in [\ell_{\mathtt{x}} + 1, \ell_{\mathtt{x},\mathtt{w}}]$. Then, it outputs the extracted witness $\boldsymbol{w}^* = (w^*_{\ell_{\mathtt{x}}+1}, \ldots, w^*_{\ell_{\mathtt{x},\mathtt{w}}})$.

Theorem 8 (Completeness). *The BARG construction in Section 6 is complete.*

Proof. Given any circuit $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x}}} \times \{0,1\}^{\ell_{\mathtt{w}}} \to \{0,1\}$, instances $\mathtt{x}_1, \ldots, \mathtt{x}_m \in \{0,1\}^{\ell_{\mathtt{x}}}$, and witnesses $\mathtt{w}_1, \ldots, \mathtt{w}_m \in \{0,1\}^{\ell_{\mathtt{w}}}$ such that $C(\mathtt{x}_i, \mathtt{w}_i) = 1$ for all $i \in [m]$. Let $\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda, 1^m)$, $\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathcal{C}, \{(\mathtt{x}_i, \mathtt{w}_i)\}_{i\in[m]})$ and $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$ We show that $\mathsf{Ver}(\mathsf{crs}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi) = 1$. From the construction, we have the following equalities:

– For each $j \in [\ell_{\mathcal{C}}]$, we have $\Phi_j(x) = \sum_{i\in[0,m-1]} \phi_{j,i} x^i$ such that $\Phi_j(i) = \mathtt{w}_{i,j}$ for all $i \in [m]$. With $\mathfrak{a}_i = [\![\alpha^i \boldsymbol{u}_1, \boldsymbol{0}]\!]_1$ and $\mathfrak{b}_i = [\![\alpha^i \boldsymbol{v}_1, \boldsymbol{0}]\!]_2$ for all $i \in [0, m-1]$, the following holds

$$\mathfrak{u}_j = \prod_{i\in[0,m-1]} \mathfrak{a}_i^{\phi_{j,i}} = \sum_{i\in[0,m-1]} \phi_{j,i} [\![\alpha^i \boldsymbol{u}_1, \boldsymbol{0}]\!]_1 = [\![\Phi_j(\alpha)\boldsymbol{u}_1, \boldsymbol{0}]\!]_1; \tag{6.1}$$

$$\mathfrak{v}_j = \prod_{i\in[0,m-1]} \mathfrak{b}_i^{\phi_{j,i}} = \sum_{i\in[0,m-1]} \phi_{j,i} [\![\alpha^i \boldsymbol{v}_1, \boldsymbol{0}]\!]_2 = [\![\Phi_j(\alpha)\boldsymbol{v}_1, \boldsymbol{0}]\!]_2. \tag{6.2}$$

– For all $i \in [m]$ and $j \in [\ell_{\mathcal{C}}]$, we have $\mathtt{w}_{i,j} \in \{0,1\}$ since $\mathcal{C}$ is a Boolean circuit. This implies that $\mathtt{w}_{i,j}^2 = \mathtt{w}_{i,j}$. Since $\Phi_j(i) = \mathtt{w}_{i,j}$, the following holds for all $j \in [\ell_{\mathcal{C}}]$.

$$\forall i \in [m] : \Phi_j^2(i) = \mathtt{w}_{i,j}^2 = \mathtt{w}_{i,j} = \Phi_j(i),$$

which implies that $\Phi_j^2(i) - \Phi_j(i) = 0$. This means the polynomial $\Phi_j^2(x) - \Phi_j(x)$ has roots in $x \in \{1, \ldots, m\}$. Therefore, given $\zeta(x) = \prod_{i\in[m]}(x-i)$, there exists a polynomial $Q_j \in \mathbb{Z}_q[x]$ such that

$$\Phi_j^2(x) - \Phi_j(x) = Q_j(x) \cdot \prod_{i\in[m]} (x-i) = Q_j(x) \cdot \zeta(x). \tag{6.3}$$

The polynomial $Q_j(x)$ has degree at most $m-2$. Write $Q_j(x) = \sum_{i\in[0,m-2]} q_{j,i} x^i$, the following holds

$$\mathfrak{p}_j = \prod_{i\in[0,m-2]} \mathfrak{b}_i^{q_{j,i}} = \sum_{i\in[0,m-2]} q_{j,i} [\![\alpha^i \boldsymbol{v}_1, \boldsymbol{0}]\!]_2 = [\![Q_j(\alpha)\boldsymbol{v}_1, \boldsymbol{0}]\!]_2. \tag{6.4}$$

– Finally, for each NAND gate $G_j = (j_1, j_2, j)$ in $\mathcal{C}$ where $j \in [\ell_{\mathtt{x},\mathtt{w}} + 1, \ell_{\mathcal{C}}]$ and $j_1, j_2 \le j$, we have for all $i \in [m]$ that $\mathtt{w}_{i,j} = \mathrm{NAND}(\mathtt{w}_{i,j_1}, \mathtt{w}_{i,j_2}) = 1 - \mathtt{w}_{i,j_1}\mathtt{w}_{i,j_2}$. Since $\Phi_j(i) = \mathtt{w}_{i,j}$, we have that for all $i \in [m]$,

$$0 = 1 - \mathtt{w}_{i,j_1}\mathtt{w}_{i,j_2} - \mathtt{w}_{i,j} = 1 - \Phi_{j_1}(i)\Phi_{j_2}(i) - \Phi_j(i).$$

This means the polynomial $1 - \Phi_{j_1}(x)\Phi_{j_2}(x) - \Phi_j(x)$ has roots at $x \in \{1, \ldots, m\}$. Therefore, given $\zeta(x) = \prod_{i\in[m]}(x-i)$, there exists a polynomial $R_j \in \mathbb{Z}_q[x]$ such that

$$1 - \Phi_{j_1}(x)\Phi_{j_2}(x) - \Phi_j(x) = R_j(x) \cdot \prod_{i\in[m]} (x-i) = R_j(x) \cdot \zeta(x). \tag{6.5}$$

The polynomial $R_j(x)$ has degree at most $m-2$. Write $R_j(x) = \sum_{i\in[0,m-2]} r_{j,i} x^i$, the following holds

$$\mathfrak{q}_j = \prod_{i\in[0,m-2]} \mathfrak{b}_i^{r_{j,i}} = \sum_{i\in[0,m-2]} r_{j,i} [\![\alpha^i \boldsymbol{v}_1, \boldsymbol{0}]\!]_2 = [\![R_j(\alpha)\boldsymbol{v}_1, \boldsymbol{0}]\!]_2. \tag{6.6}$$

We now consider the checks that Ver performs:

1. **Construct statement commitments**: For each $j \in [\ell_{\mathtt{x}}]$, the verification algorithm interpolates the polynomial $\Phi_j(x)$ as described in the above analysis, thus correspondingly achieving $\{\mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}]}$ as described in Equation (6.1) and Equation (6.2) respectively.
2. **Validity of source-group memberships**: By construction,

– for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathcal{C}}]$, we have $\mathfrak{u}_j \in \mathfrak{G}_1$ and $\mathfrak{v}_j \in \mathfrak{G}_2$;
– for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$, we have $\mathfrak{p}_j \in \mathfrak{G}_2$;
– for each $j \in [\ell_{\mathtt{x},\mathtt{w}}+1, \ell_{\mathcal{C}}]$, we have $\mathfrak{q}_j \in \mathfrak{G}_2$.

Then by the completeness of the proof system for source-group memberships, this check passes.

3. **Validity of witness-wire assignments**: From Equation (6.1), Equation (6.2), Equation (6.3) and Equation (6.4), for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$ we have

$$\begin{aligned}
e(\mathfrak{u}_j, \mathfrak{v}_j) &= [\![\Phi_j(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\Phi_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= \Phi_j^2(\alpha)[\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= (\Phi_j(\alpha) + Q_j(\alpha)\cdot\zeta(\alpha))[\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= [\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\Phi_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\zeta(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![Q_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= [\![\Phi_j(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\zeta(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![Q_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{p}_j) \\
&= e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{p}_j).
\end{aligned}$$

   Thus, this check passes.

4. **Validity of gate computation**: From Equation (6.1), Equation (6.2), Equation (6.5) and Equation (6.6), for each $j \in [\ell_{\mathtt{x},\mathtt{w}}+1, \ell_{\mathcal{C}}]$ we have

$$\begin{aligned}
e(\mathfrak{a}_0, \mathfrak{b}_0) &= [\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= (\Phi_{j_1}(\alpha)\Phi_{j_2}(\alpha) + \Phi_j(\alpha) + R_j(\alpha)\zeta(\alpha))[\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= [\![\Phi_{j_1}(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\Phi_{j_2}(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\Phi_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\zeta(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![R_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= [\![\Phi_{j_1}(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\Phi_{j_2}(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\Phi_j(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![\boldsymbol{v}_1, \mathbf{0}]\!]_2 + [\![\zeta(\alpha)\boldsymbol{u}_1, \mathbf{0}]\!]_1 \bullet [\![R_j(\alpha)\boldsymbol{v}_1, \mathbf{0}]\!]_2 \\
&= e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{q}_j) \\
&= e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{q}_j).
\end{aligned}$$

   Thus, this check passes.

5. **Output satisfiability**: For all $i \in [m]$, we have that $\mathtt{w}_{i,\ell_{\mathcal{C}}}$ points to the output wire of each $\mathcal{C}(\mathtt{x}_i, \mathtt{w}_i)$. On the other hand, $\Phi_{\ell_{\mathcal{C}}}(x)$ is the unique polynomial of degree at most $m-1$ such that $\Phi_{\ell_{\mathcal{C}}}(i) = \mathtt{w}_{i,\ell_{\mathcal{C}}} = 1$. This means $\Phi_{\ell_{\mathcal{C}}}(x) = 1$, implying $\mathfrak{u}_{\ell_{\mathcal{C}}} = \mathfrak{a}_0$ and $\mathfrak{v}_{\ell_{\mathcal{C}}} = \mathfrak{b}_0$. Thus, this check passes.

Since all the checks pass, the Ver algorithm always outputs 1. □

Theorem 9 (Succinctness). *The BARG construction in Section 6 is succinct and has a CRS of size* $|\mathsf{crs}| = m \cdot \mathcal{O}(\lambda)$.

Proof. Consider any parameters $\lambda, m, \ell_{\mathcal{C}} \in \mathbb{N}$. A proof $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$ contains a total of $\ell_{\mathcal{C}} - \ell_{\mathtt{x}}$ elements in $\mathfrak{G}_1$ and $2(\ell_{\mathcal{C}} - \ell_{\mathtt{x}})$ elements in $\mathfrak{G}_2$. The crs has size linear in m, consisting of $m+1$ elements in $\mathfrak{G}_1$ and m elements in $\mathfrak{G}_2$, plus the constant-size description bgd. □

Remark 7 (Faster Verification with Preprocessing). As in [CEW25] and other previous BARG constructions, the verifier can precompute the encodings of the statements if these are known in advance. The verification key output by $\mathsf{PreVer}(\mathsf{crs}, \{\mathtt{x}_i\}_{i\in[m]})$ becomes $\mathsf{vk} = \{\mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}]}$ and contains exactly $\ell_{\mathtt{x}}$ elements in $\mathfrak{G}_1$ and $\ell_{\mathtt{x}}$ elements in $\mathfrak{G}_2$. The online efficient verification algorithm $\mathsf{EffVer}(\mathsf{vk}, \mathcal{C}, \pi)$ can simply run all the checks in time $\mathsf{poly}(\ell_{\mathcal{C}}, \lambda)$.

Theorem 10 (Somewhere Extractability). *The BARG construction in Section 6 satisfies somewhere extractability under the m-bilateral subgroup decision exponent assumption (Definition 15) and the perfect soundness of the verifiable source-group membership (Definition 6) that are satisfied by $\mathcal{BG}^{2,6}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$.*

Proof. We first prove that the common reference strings generated by the Setup and $\mathsf{SetupTd}$ algorithms are indistinguishable in Lemma 1. Then, we prove somewhere extractability in Lemma 2.

Lemma 1 (CRS Indistinguishability). *The BARG construction in Section 6 satisfies CRS indistinguishability under the m-bilateral subgroup decision exponent assumption.*

Proof. Given any adversary $\mathcal{A}$ that breaks CRS indistinguishability with an advantage ϵ, we construct an adversary $\mathcal{B}$ that breaks the m-BSDE assumption:

- At the beginning, $\mathcal{B}$ receives an instance $\mathsf{D} := \mathsf{bgd}, \begin{cases} \{[\![\alpha^j \boldsymbol{u}_1, \boldsymbol{0}]\!]_1\}_{j\in[m]}, [\![\boldsymbol{u}_1, \mathsf{b}\boldsymbol{u}_2]\!]_1 \\ \{[\![\alpha^j \boldsymbol{v}_1, \boldsymbol{0}]\!]_2\}_{j\in[m]}, [\![\boldsymbol{v}_1, \mathsf{b}\boldsymbol{v}_2]\!]_2 \end{cases}$ sampled from the challenge distribution $\mathcal{D}_\mathsf{b}$ where $\mathsf{b} \in \{0,1\}$ is the hidden bit.
- For any $i^* \in [m]$, $\mathcal{B}$ plays as the BARG setup and generates its crs to respond to $\mathcal{A}$ as follows:
  1. For all $i \in [0,m]$, compute

$$\mathfrak{a}_i = (i^*)^i [\![\boldsymbol{u}_1, \mathsf{b}\boldsymbol{u}_2]\!]_1 + \sum_{j\in[i]} \binom{i}{j} (i^*)^{i-j} [\![\alpha^j \boldsymbol{u}_1, \boldsymbol{0}]\!]_1 = [\![(\alpha+i^*)^i \boldsymbol{u}_1, (i^*)^i \mathsf{b}\boldsymbol{u}_2]\!]_1; \tag{6.7}$$

$$\mathfrak{b}_i = (i^*)^i [\![\boldsymbol{v}_1, \mathsf{b}\boldsymbol{v}_2]\!]_2 + \sum_{j\in[i]} \binom{i}{j} (i^*)^{i-j} [\![\alpha^j \boldsymbol{u}_1, \boldsymbol{0}]\!]_2 = [\![(\alpha+i^*)^i \boldsymbol{v}_1, (i^*)^i \mathsf{b}\boldsymbol{v}_2]\!]_2. \tag{6.8}$$

  2. Given $\zeta(x) = \prod_{i\in[m]}(x-i) := \sum_{i\in[0,m]} \zeta_i x^i$, compute

$$\mathfrak{z} = \prod_{i\in[0,m]} \mathfrak{a}_i^{\zeta_i} = \sum_{i\in[0,m]} \zeta_i [\![(\alpha+i^*)^i \boldsymbol{u}_1, (i^*)^i \mathsf{b}\boldsymbol{u}_2]\!]_1 = [\![\zeta(\alpha+i^*)\boldsymbol{u}_1, \zeta(i^*)\mathsf{b}\boldsymbol{u}_2]\!]_1$$
$$= [\![\zeta(\alpha+i^*)\boldsymbol{u}_1, \boldsymbol{0}]\!]_1.$$

  3. $\mathcal{B}$ uses randomizers $\{\mathfrak{r}_h\}_{h\in[p(\lambda)]}$ given in bgd to randomize encodings in steps 1 and 2.
  4. Give $\mathsf{crs} = (\mathsf{bgd}, (\mathfrak{a}_0, \mathfrak{b}_0), \ldots, (\mathfrak{a}_{m-1}, \mathfrak{b}_{m-1}), \mathfrak{z})$ to $\mathcal{A}$ and outputs what $\mathcal{A}$ outputs.

With Equation (6.7) and Equation (6.8), we consider the distribution of crs depending on the value of b:

- If $\mathsf{b} = 0$, we have $\mathfrak{a}_i = [\![(\alpha+i^*)^i \boldsymbol{u}_1, \boldsymbol{0}]\!]_1$, $\mathfrak{b}_i = [\![(\alpha+i^*)^i \boldsymbol{v}_1, \boldsymbol{0}]\!]_2$ for all $i \in [m]$ and $\mathfrak{z} = [\![\zeta(\alpha+i^*)\boldsymbol{u}_1, \boldsymbol{0}]\!]_1$. Since α is uniformly random over $\mathbb{Z}_q$, so is $(\alpha+i^*)$. Therefore, the crs output by $\mathcal{B}$ is distributed as by $\mathsf{Setup}(1^\lambda, 1^m)$.
- If $\mathsf{b} = 1$, we have $\mathfrak{a}_i = [\![(\alpha+i^*)^i \boldsymbol{u}_1, (i^*)^i \boldsymbol{u}_2]\!]_1$, $\mathfrak{b}_i = [\![(\alpha+i^*)^i \boldsymbol{v}_1, (i^*)^i \boldsymbol{v}_2]\!]_2$ for all $i \in [m]$ and $\mathfrak{z} = [\![\zeta(\alpha+i^*)\boldsymbol{u}_1, \boldsymbol{0}]\!]_1$. Since α is uniformly random over $\mathbb{Z}_q$, so is $(\alpha+i^*)$. Therefore, the crs output by $\mathcal{B}$ is distributed as by $\mathsf{SetupTd}(1^\lambda, 1^m, i^*)$.

We conclude that $\mathcal{B}$ breaks the m-BDSE assumption with the same advantage ϵ. □

Lemma 2 (Somewhere Extractable in Trapdoor Mode). *The BARG construction in Section 6 is somewhere extractable in trapdoor mode under the perfect soundness of the verifiable source-group membership.*

Proof. Given any index $i^* \in [m]$, let $(\mathsf{crs}^*, \mathsf{td}) \leftarrow \mathsf{SetupTd}(1^\lambda, 1^m, i^*)$. We can write

$$\mathsf{crs}^* = (\mathsf{bgd}, (\mathfrak{a}_0, \mathfrak{b}_0), \ldots, (\mathfrak{a}_{m-1}, \mathfrak{b}_{m-1}), \mathfrak{z})$$

where the distributions of relevant components are recalled as follows

- $(\mathsf{bgd}, \mathsf{sk}, \boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{b}_1, \boldsymbol{b}_2) \leftarrow \mathcal{BG}^{2,6}_{\mathcal{P},\mathsf{IPE}}(1^\lambda)$;
- $(\mathsf{bgd}_{\mathcal{P}}, \mathfrak{G}_1 = \tilde{\mathfrak{G}}_{1,1} \times \tilde{\mathfrak{G}}_{1,2}, \mathfrak{G}_2 = \tilde{\mathfrak{G}}_{2,1} \times \tilde{\mathfrak{G}}_{2,2}, \mathfrak{G}_T, q, e) \in \mathsf{bgd}$;
- $r_1, t_1, r_2, t_2, \alpha \leftarrow\!\!\$\ \mathbb{Z}_q$;
- $\boldsymbol{u}_1 = r_1\boldsymbol{a}_1, \boldsymbol{v}_1 = t_1\boldsymbol{b}_1, \boldsymbol{u}_2 = r_2\boldsymbol{a}_2, \boldsymbol{v}_2 = t_2\boldsymbol{b}_2$;
- $\mathfrak{a}_i = [\![\alpha^i\boldsymbol{u}_1, (i^*)^i\boldsymbol{u}_2]\!]_1\ \forall i \in [0, m-1]$;
- $\mathfrak{b}_i = [\![\alpha^i\boldsymbol{v}_1, (i^*)^i\boldsymbol{v}_2]\!]_2\ \forall i \in [0, m-1]$;
- $\mathfrak{z} = [\![\zeta(\alpha)\boldsymbol{u}_1, \boldsymbol{0}]\!]_1$.

As specified in Definition 19, $\mathsf{bgd}_{\mathcal{P}}$ is the parameter of the underlying q-order group. The sets $\{\boldsymbol{a}_1, \boldsymbol{a}_2\}$ and $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$ generate $\mathfrak{G}_1$ and $\mathfrak{G}_2$ respectively and are each linearly independent in $\mathbb{Z}_q^2$. The subgroups $\tilde{\mathfrak{G}}_{1,1}$ and $\tilde{\mathfrak{G}}_{1,2}$ of $\mathfrak{G}_1$ are generated by $\boldsymbol{a}_1$ and $\boldsymbol{a}_2$ respectively. Similarly, the subgroups $\tilde{\mathfrak{G}}_{2,1}$ and $\tilde{\mathfrak{G}}_{2,2}$ of $\mathfrak{G}_2$ are generated by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ respectively. We have the following remark.

Remark 8 (Basis Changes). Except a negligible probability bounded by $\frac{4}{q}$, the randomness r_1, t_1, r_2, t_2 are all non-zero. Then, alternative to $\{\boldsymbol{a}_1, \boldsymbol{a}_2\}$ and $\{\boldsymbol{b}_1, \boldsymbol{b}_2\}$, the sets $\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$ and $\{\boldsymbol{v}_1, \boldsymbol{v}_2\}$ generate $\mathfrak{G}_1$ and $\mathfrak{G}_2$ respectively and are each linearly independent in $\mathbb{Z}_q^2$. The subgroups $\tilde{\mathfrak{G}}_{1,1}$ and $\tilde{\mathfrak{G}}_{1,2}$ of $\mathfrak{G}_1$ are generated by $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ respectively. Similarly, the subgroups $\tilde{\mathfrak{G}}_{2,1}$ and $\tilde{\mathfrak{G}}_{2,2}$ of $\mathfrak{G}_2$ are generated by $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ respectively.

For any element $\mathfrak{g} \in \mathfrak{G}_1$, we denote by $(\gamma_{1,\mathfrak{g}}, \gamma_{2,\mathfrak{g}})$ its coefficients when decomposed as $\mathfrak{g} = [\![\gamma_{1,\mathfrak{g}}\boldsymbol{u}_1, \gamma_{2,\mathfrak{g}}\boldsymbol{u}_2]\!]_1$. Similarly, for any element $\mathfrak{h} \in \mathfrak{G}_2$, we denote by $(\theta_{1,\mathfrak{h}}, \theta_{2,\mathfrak{h}})$ its coefficients when decomposed as $\mathfrak{h} = [\![\theta_{1,\mathfrak{h}}\boldsymbol{v}_1, \theta_{2,\mathfrak{h}}\boldsymbol{v}_2]\!]_2$.

Take any circuit $\mathcal{C} : \{0,1\}^{\ell_{\mathtt{x}}} \times \{0,1\}^{\ell_{\mathtt{w}}} \to \{0,1\}$, any instances $\mathtt{x}_1, \ldots, \mathtt{x}_m \in \{0,1\}^{\ell_{\mathtt{x}}}$ and any proof $\pi = (\{j, \mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathcal{C}}]}, \{j, \mathfrak{p}_j\}_{j\in[\ell_{\mathtt{x}}+1,\ell_{\mathtt{x},\mathtt{w}}]}, \{j, \mathfrak{q}_j\}_{j\in[\ell_{\mathtt{x},\mathtt{w}}+1,\ell_{\mathcal{C}}]})$ such that $\mathsf{Ver}(\mathsf{crs}^*, \mathcal{C}, (\mathtt{x}_1, \ldots, \mathtt{x}_m), \pi) = 1$, we argue extractability by considering steps in the verification algorithm:

1. **Construct statement commitments**: For each $j \in [\ell_{\mathtt{x}}]$, let $\Phi_j \in \mathbb{Z}_q[x]$ be the unique polynomial of degree at most $m-1$ where $\Phi_j(i) = \mathtt{x}_{i,j}$ for all $i \in [m]$. Write $\Phi_j(x) = \sum_{i\in[0,m-1]} \phi_{j,i}x^i$ and we have

$$\mathfrak{u}_j = \prod_{i\in[0,m-1]} \mathfrak{a}_i^{\phi_{j,i}} = \sum_{i\in[0,m-1]} \phi_{j,i}[\![\alpha^i\boldsymbol{u}_1, (i^*)^i\boldsymbol{u}_2]\!]_1 = [\![\Phi_j(\alpha)\boldsymbol{u}_1, \Phi_j(i^*)\boldsymbol{u}_2]\!]_1 = [\![\Phi_j(\alpha)\boldsymbol{u}_1, \mathtt{x}_{i^*,j}\boldsymbol{u}_2]\!]_1;$$

$$\mathfrak{v}_j = \prod_{i\in[0,m-1]} \mathfrak{b}_i^{\phi_{j,i}} = \sum_{i\in[0,m-1]} \phi_{j,i}[\![\alpha^i\boldsymbol{v}_1, (i^*)^i\boldsymbol{v}_2]\!]_2 = [\![\Phi_j(\alpha)\boldsymbol{v}_1, \Phi_j(i^*)\boldsymbol{v}_2]\!]_2 = [\![\Phi_j(\alpha)\boldsymbol{v}_1, \mathtt{x}_{i^*,j}\boldsymbol{v}_2]\!]_2.$$

   This means $\gamma_{2,\mathfrak{u}_j} = \theta_{2,\mathfrak{v}_j} = x_{i^*,j}$ and $\gamma_{2,\mathfrak{u}_j}, \theta_{2,\mathfrak{v}_j} \in \{0,1\}$ for all $j \in [\ell_{\mathtt{x}}]$.
2. **Validity of source-group memberships**: Under the perfect soundness of the verifiable source-group membership, we have
   - for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathcal{C}}]$, $\mathfrak{u}_j \in \mathfrak{G}_1$ and $\mathfrak{v}_j \in \mathfrak{G}_2$;
   - for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$, $\mathfrak{p}_j \in \mathfrak{G}_2$;
   - for each $j \in [\ell_{\mathtt{x},\mathtt{w}}+1, \ell_{\mathcal{C}}]$, $\mathfrak{q}_j \in \mathfrak{G}_2$.

3. **Validity of witness-wire assignments**: For each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$, we have a valid check such that
$$e(\mathfrak{u}_j, \mathfrak{v}_j) = e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{p}_j) \text{ and } e(\mathfrak{u}_j, \mathfrak{v}_j) = e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{p}_j).$$
By projecting $\mathfrak{G}_1$ elements onto $\tilde{\mathfrak{G}}_{1,2}$ and $\mathfrak{G}_2$ elements onto $\tilde{\mathfrak{G}}_{2,2}$, this check implies
$$\gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{v}_j} = \gamma_{2,\mathfrak{a}_0}\theta_{2,\mathfrak{v}_j} + \gamma_{2,\mathfrak{z}}\theta_{2,\mathfrak{p}_j} \text{ and } \gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{v}_j} = \gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{b}_0} + \gamma_{2,\mathfrak{z}}\theta_{2,\mathfrak{p}_j}$$
in $\mathbb{Z}_q$. On the other hand, since $\mathfrak{z} \in \tilde{\mathfrak{G}}_1$, then $\gamma_{2,\mathfrak{z}} = 0$. Additionally, by construction of crs^*, we have $\gamma_{2,\mathfrak{a}_0} = \theta_{2,\mathfrak{b}_0} = 1$. Therefore, from the check, we have
$$\gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{v}_j} = \theta_{2,\mathfrak{v}_j} \text{ and } \gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{v}_j} = \gamma_{2,\mathfrak{u}_j}.$$
This means $\gamma_{2,\mathfrak{u}_j} = \theta_{2,\mathfrak{v}_j}$ and $\gamma_{2,\mathfrak{u}_j}, \theta_{2,\mathfrak{v}_j} \in \{0,1\}$ for all $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$.
4. **Validity of gate computation**: For each NAND gate $G_j = (j_1, j_2, j)$ for $j \in [\ell_{\mathtt{x},\mathtt{w}}+1, \ell_{\mathcal{C}}]$, we have a valid check such that
$$e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{a}_0, \mathfrak{v}_j)e(\mathfrak{z}, \mathfrak{q}_j) = e(\mathfrak{a}_0, \mathfrak{b}_0) \text{ and } e(\mathfrak{u}_{j_1}, \mathfrak{v}_{j_2})e(\mathfrak{u}_j, \mathfrak{b}_0)e(\mathfrak{z}, \mathfrak{q}_j) = e(\mathfrak{a}_0, \mathfrak{b}_0).$$
By projecting $\mathfrak{G}_1$ elements onto $\tilde{\mathfrak{G}}_{1,2}$ and $\mathfrak{G}_2$ elements onto $\tilde{\mathfrak{G}}_{2,2}$, this check implies
$$\gamma_{2,\mathfrak{u}_{j_1}}\theta_{2,\mathfrak{v}_{j_2}} + \gamma_{2,\mathfrak{a}_0}\theta_{2,\mathfrak{v}_j} + \gamma_{2,\mathfrak{z}}\theta_{2,\mathfrak{q}_j} = \gamma_{2,\mathfrak{a}_0}\theta_{2,\mathfrak{b}_0} \text{ and } \gamma_{2,\mathfrak{u}_{j_1}}\theta_{2,\mathfrak{v}_{j_2}} + \gamma_{2,\mathfrak{u}_j}\theta_{2,\mathfrak{b}_0} + \gamma_{2,\mathfrak{z}}\theta_{2,\mathfrak{q}_j} = \gamma_{2,\mathfrak{a}_0}\theta_{2,\mathfrak{b}_0}$$
in $\mathbb{Z}_q$. By construction of crs^*, we have $\gamma_{2,\mathfrak{z}} = 0$ and $\gamma_{2,\mathfrak{a}_0} = \theta_{2,\mathfrak{b}_0} = 1$. Therefore, from the check, we have
$$\gamma_{2,\mathfrak{u}_{j_1}}\theta_{2,\mathfrak{v}_{j_2}} + \theta_{2,\mathfrak{v}_j} = 1 \text{ and } \gamma_{2,\mathfrak{u}_{j_1}}\theta_{2,\mathfrak{v}_{j_2}} + \gamma_{2,\mathfrak{u}_j} = 1.$$
We prove that $\gamma_{2,\mathfrak{u}_{j_1}}, \theta_{2,\mathfrak{v}_{j_2}} \in \{0,1\}$ and $\theta_{2,\mathfrak{v}_j} = \gamma_{2,\mathfrak{u}_j} = \mathrm{NAND}(\gamma_{2,\mathfrak{u}_{j_1}}, \theta_{2,\mathfrak{v}_{j_2}})$ for all $j \in [\ell_{\mathtt{x},\mathtt{w}}+1, \ell_{\mathcal{C}}]$ by induction. We start with NAND gates indexed by j' which have input wires $j_1, j_2 \in [\ell_{\mathtt{x},\mathtt{w}}]$. As proven in Items 3 and 4, we have $\gamma_{2,\mathfrak{u}_j}, \theta_{2,\mathfrak{v}_j} \in \{0,1\}$ for all $j \in [\ell_{\mathtt{x},\mathtt{w}}]$. This implies that $\theta_{2,\mathfrak{v}_{j'}} = \gamma_{2,\mathfrak{u}_{j'}} = \mathrm{NAND}(\gamma_{2,\mathfrak{u}_{j_1}}, \theta_{2,\mathfrak{v}_{j_2}})$ and $\theta_{2,\mathfrak{v}_{j'}}, \gamma_{2,\mathfrak{u}_{j'}} \in \{0,1\}$.
Since the output wires indexed by j' serve as input wires of other gates indexed by $j \geq j'$, the validity of gate computation follows.
5. **Output satisfiability**: By projecting the equalities $\mathfrak{u}_{\ell_{\mathcal{C}}} = \mathfrak{a}_0$ onto $\tilde{\mathfrak{G}}_{1,2}$ and $\mathfrak{v}_{\ell_{\mathcal{C}}} = \mathfrak{b}_0$ onto $\tilde{\mathfrak{G}}_{2,2}$, we have $\gamma_{2,\mathfrak{u}_{\ell_{\mathcal{C}}}} = \theta_{2,\mathfrak{v}_{\ell_{\mathcal{C}}}} = 1$.

Taken together, the above properties show that $\{\gamma_{2,\mathfrak{u}_j}\}_{j\in[\ell_{\mathcal{C}}]}$ are a valid labeling of the wires of $\mathcal{C}((\gamma_{2,\mathfrak{u}_1}, \ldots, \gamma_{2,\mathfrak{u}_{\ell_{\mathtt{x}}}}), (\gamma_{2,\mathfrak{u}_{\ell_{\mathtt{x}}+1}}, \ldots, \gamma_{2,\mathfrak{u}_{\ell_{\mathtt{x},\mathtt{w}}}})) = 1$ where the first $\ell_{\mathtt{x}}$ inputs coincides with $\mathtt{x}_{i^*}$ and the output bit is $\gamma_{2,\mathfrak{u}_{\ell_{\mathcal{C}}}} = 1$. To complete the proof, it suffices to show that $\boldsymbol{w}^* \leftarrow \mathsf{Ext}(\mathsf{td}, \mathcal{C}, \{\mathtt{x}_i\}_{i\in[m]}, \pi)$ coincides with $(\gamma_{2,\mathfrak{u}_{\ell_{\mathtt{x}}+1}}, \ldots, \gamma_{2,\mathfrak{u}_{\ell_{\mathtt{x},\mathtt{w}}}})$.

In the extraction algorithm, for each $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$, the witness-wire value w_j^* is set to be 0 when the equality $e(\mathfrak{u}_j, [\![\mathbf{0}, \boldsymbol{v}_2]\!]_2) = [\mathbf{0}_{2\times 2}]_T$ holds. This happens if and only if $\mathfrak{u}_j = [\![\gamma_{1,\mathfrak{u}_j}\boldsymbol{u}_1, \mathbf{0}]\!]_1$, which means $\gamma_{2,\mathfrak{u}_j} = 0$. Otherwise, w_j^* is set to be 1. Therefore, $w_j^* = \gamma_{2,\mathfrak{u}_j}$ for all $j \in [\ell_{\mathtt{x}}+1, \ell_{\mathtt{x},\mathtt{w}}]$. □

Acknowledgements This work was done while David Balbás and Duy Nguyen were affiliated with IMDEA Software Institute. We thank the anonymous reviewers for their helpful comments and suggestions. This work is supported by the PICOCRYPT project that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 101001283), and partially funded by the grants PID2022-142290OB-I00 and CEX2024-001471-M, funded by MCIN/AEI/10.13039/501100011033/ FEDER, UE. Duy Nguyen acknowledges the support of the France 2030 ANR Project ANR-22-PECY-003 SecureCompute.

References

ABF24. Gaspard Anthoine, David Balbás, and Dario Fiore. Fully-succinct multi-key homomorphic signatures from standard assumptions. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part III*, volume 14922 of *Lecture Notes in Computer Science*, pages 317–351. Springer, Cham, August 2024. doi:10.1007/978-3-031-68382-4_10.

ACG24. Abtin Afshar, Jiaqi Cheng, and Rishab Goyal. Leveled fully-homomorphic signatures from batch arguments. Cryptology ePrint Archive, Report 2024/931, 2024. URL: https://eprint.iacr.org/2024/931.

AECL25. Ferran Alborch Escobar, Sébastien Canard, and Fabien Laguillaumie. Simulation secure multi-input quadratic functional encryption: applications to differential privacy. *Designs, Codes and Cryptography*, 94(1):1, Dec 2025. doi:10.1007/s10623-025-01754-1.

BCJP24. Maya Farber Brodsky, Arka Rai Choudhuri, Abhishek Jain, and Omer Paneth. Monotone-policy aggregate signatures. In Marc Joye and Gregor Leander, editors, *Advances in Cryptology – EUROCRYPT 2024, Part IV*, volume 14654 of *Lecture Notes in Computer Science*, pages 168–195. Springer, Cham, May 2024. doi:10.1007/978-3-031-58737-5_7.

BFL25. David Balbás, Dario Fiore, and Russell W. F. Lai. Circuit-succinct algebraic batch arguments from projective functional commitments. Cryptology ePrint Archive, Report 2025/1943, 2025. URL: https://eprint.iacr.org/2025/1943.

BGI+01. Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil P. Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 1–18. Springer, Berlin, Heidelberg, August 2001. doi:10.1007/3-540-44647-8_1.

BHK17. Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th Annual ACM Symposium on Theory of Computing*, pages 474–482. ACM Press, June 2017. doi:10.1145/3055399.3055497.

BJK15. Allison Bishop, Abhishek Jain, and Lucas Kowalczyk. Function-hiding inner product encryption. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology – ASIACRYPT 2015, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 470–491. Springer, Berlin, Heidelberg, November / December 2015. doi:10.1007/978-3-662-48797-6_20.

Boy08. Xavier Boyen. The uber-assumption family (invited talk). In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 39–56. Springer, Berlin, Heidelberg, September 2008. doi:10.1007/978-3-540-85538-5_3.

BSW11. Dan Boneh, Amit Sahai, and Brent Waters. Functional encryption: Definitions and challenges. In Yuval Ishai, editor, *TCC 2011: 8th Theory of Cryptography Conference*, volume 6597 of *Lecture Notes in Computer Science*, pages 253–273. Springer, Berlin, Heidelberg, March 2011. doi:10.1007/978-3-642-19571-6_16.

CCH+19. Ran Canetti, Yilei Chen, Justin Holmgren, Alex Lombardi, Guy N. Rothblum, Ron D. Rothblum, and Daniel Wichs. Fiat-Shamir: from practice to theory. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 1082–1090. ACM Press, June 2019. doi:10.1145/3313276.3316380.

CEW25. Binyi Chen, Noel Elias, and David J. Wu. Pairing-based batch arguments for NP with a linear-size CRS. In Goichiro Hanaoka and Bo-Yin Yang, editors, *Advances in Cryptology – ASIACRYPT 2025, Part V*, volume 16249 of *Lecture Notes in Computer Science*, pages 554–577. Springer, Singapore, December 2025. doi:10.1007/978-981-95-5116-3_18.

CGH98. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *30th Annual ACM Symposium on Theory of Computing*, pages 209–218. ACM Press, May 1998. doi:10.1145/276698.276741.

CJJ21. Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. Non-interactive batch arguments for NP from standard assumptions. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology – CRYPTO 2021, Part IV*, volume 12828 of *Lecture Notes in Computer Science*, pages 394–423, Virtual Event, August 2021. Springer, Cham. doi:10.1007/978-3-030-84259-8_14.

CJJ22. Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for $\mathcal{P}$ from LWE. In *62nd Annual Symposium on Foundations of Computer Science*, pages 68–79. IEEE Computer Society Press, February 2022. doi:10.1109/FOCS52979.2021.00016.

DGKV22. Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. In *63rd Annual Symposium on Foundations of Computer Science*, pages 1057–1068. IEEE Computer Society Press, October / November 2022. doi:10.1109/FOCS54457.2022.00103.

DJJ+25. Pratish Datta, Abhishek Jain, Zhengzhong Jin, Alexis Korb, Surya Mathialagan, and Amit Sahai. Incrementally verifiable computation for NP from standard assumptions. In Yael Tauman Kalai and Seny F. Kamara, editors, *Advances in Cryptology – CRYPTO 2025, Part VII*, volume 16006 of *Lecture Notes in Computer Science*, pages 611–642. Springer, Cham, August 2025. doi:10.1007/978-3-032-01907-3_20.

EHK+13. Alex Escala, Gottfried Herold, Eike Kiltz, Carla Ràfols, and Jorge Villar. An algebraic framework for Diffie-Hellman assumptions. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part II*, volume 8043 of *Lecture Notes in Computer Science*, pages 129–147. Springer, Berlin, Heidelberg, August 2013. doi:10.1007/978-3-642-40084-1_8.

Fre10. David Mandell Freeman. Converting pairing-based cryptosystems from composite-order groups to prime-order groups. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, Berlin, Heidelberg, May / June 2010. doi:10.1007/978-3-642-13190-5_3.

GLWW24. Rachit Garg, George Lu, Brent Waters, and David J. Wu. Reducing the CRS size in registered ABE systems. In Leonid Reyzin and Douglas Stebila, editors, *Advances in Cryptology – CRYPTO 2024, Part III*, volume 14922 of *Lecture Notes in Computer Science*, pages 143–177. Springer, Cham, August 2024. doi:10.1007/978-3-031-68382-4_5.

Goy24. Rishab Goyal. Mutable batch arguments and applications. Cryptology ePrint Archive, Report 2024/737, 2024. URL: https://eprint.iacr.org/2024/737.

GS08. Jens Groth and Amit Sahai. Efficient non-interactive proof systems for bilinear groups. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 415–432. Springer, Berlin, Heidelberg, April 2008. doi:10.1007/978-3-540-78967-3_24.

Gui13. Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In Michael J. Jacobson, Jr., Michael E. Locasto, Payman Mohassel, and Reihaneh Safavi-Naini, editors, *ACNS 2013: 11th International Conference on Applied Cryptography and Network Security*, volume 7954 of *Lecture Notes in Computer Science*, pages 357–372. Springer, Berlin, Heidelberg, June 2013. doi:10.1007/978-3-642-38980-1_22.

GW11. Craig Gentry and Daniel Wichs. Separating succinct non-interactive arguments from all falsifiable assumptions. In Lance Fortnow and Salil P. Vadhan, editors, *43rd Annual ACM Symposium on Theory of Computing*, pages 99–108. ACM Press, June 2011. doi:10.1145/1993636.1993651.

HHH+14. Gottfried Herold, Julia Hesse, Dennis Hofheinz, Carla Ràfols, and Andy Rupp. Polynomial spaces: A new framework for composite-to-prime-order transformations. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 261–279. Springer, Berlin, Heidelberg, August 2014. doi:10.1007/978-3-662-44371-2_15.

JJ21. Abhishek Jain and Zhengzhong Jin. Non-interactive zero knowledge from sub-exponential DDH. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021, Part I*, volume 12696 of *Lecture Notes in Computer Science*, pages 3–32. Springer, Cham, October 2021. doi:10.1007/978-3-030-77870-5_1.

Jou04. Antoine Joux. A one round protocol for tripartite Diffie–Hellman. *Journal of Cryptology*, 17(4):263–276, September 2004. doi:10.1007/s00145-004-0312-y.

KLM+18. Sam Kim, Kevin Lewi, Avradip Mandal, Hart Montgomery, Arnab Roy, and David J. Wu. Function-hiding inner product encryption is practical. In Dario Catalano and Roberto De Prisco, editors, *SCN 18: 11th International Conference on Security in Communication Networks*, volume 11035 of *Lecture Notes in Computer Science*, pages 544–562. Springer, Cham, September 2018. doi:10.1007/978-3-319-98113-0_29.

KLVW23. Yael Kalai, Alex Lombardi, Vinod Vaikuntanathan, and Daniel Wichs. Boosting batch arguments and RAM delegation. In Barna Saha and Rocco A. Servedio, editors, *55th Annual ACM Symposium on Theory of Computing*, pages 1545–1552. ACM Press, June 2023. doi:10.1145/3564246.3585200.

KPY19. Yael Tauman Kalai, Omer Paneth, and Lisa Yang. How to delegate computations publicly. In Moses Charikar and Edith Cohen, editors, *51st Annual ACM Symposium on Theory of Computing*, pages 1115–1124. ACM Press, June 2019. doi:10.1145/3313276.3316411.

KVZ21. Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and SNARGs. In Kobbi Nissim and Brent Waters, editors, *TCC 2021: 19th Theory of Cryptography Conference, Part I*, volume 13042 of *Lecture Notes in Computer Science*, pages 330–368. Springer, Cham, November 2021. doi:10.1007/978-3-030-90459-3_12.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology – ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, Berlin, Heidelberg, December 2010. doi:10.1007/978-3-642-17373-8_11.

Lew12. Allison B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 318–335. Springer, Berlin, Heidelberg, April 2012. doi:10.1007/978-3-642-29011-4_20.

Lin17. Huijia Lin. Indistinguishability obfuscation from SXDH on 5-linear maps and locality-5 PRGs. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017, Part I*, volume 10401 of *Lecture Notes in Computer Science*, pages 599–629. Springer, Cham, August 2017. doi:10.1007/978-3-319-63688-7_20.

LL20. Huijia Lin and Ji Luo. Compact adaptively secure ABE from k-Lin: Beyond NC^1 and towards NL. In Anne Canteaut and Yuval Ishai, editors, *Advances in Cryptology – EUROCRYPT 2020, Part III*, volume 12107 of *Lecture Notes in Computer Science*, pages 247–277. Springer, Cham, May 2020. doi:10.1007/978-3-030-45727-3_9.

LM15. Allison B. Lewko and Sarah Meiklejohn. A profitable sub-prime loan: Obtaining the advantages of composite order in prime-order bilinear groups. In Jonathan Katz, editor, *PKC 2015: 18th International Conference on Theory and Practice of Public Key Cryptography*, volume 9020 of *Lecture Notes in Computer Science*, pages 377–398. Springer, Berlin, Heidelberg, March / April 2015. doi:10.1007/978-3-662-46447-2_17.

O'N10. Adam O'Neill. Definitional issues in functional encryption. Cryptology ePrint Archive, Report 2010/556, 2010. URL: https://eprint.iacr.org/2010/556.

OT08. Tatsuaki Okamoto and Katsuyuki Takashima. Homomorphic encryption and signatures from vector decomposition. In Steven D. Galbraith and Kenneth G. Paterson, editors, *PAIRING 2008: 2nd International Conference on Pairing-based Cryptography*, volume 5209 of *Lecture Notes in Computer Science*, pages 57–74. Springer, Berlin, Heidelberg, September 2008. doi:10.1007/978-3-540-85538-5_4.

OT09. Tatsuaki Okamoto and Katsuyuki Takashima. Hierarchical predicate encryption for inner-products. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 214–231. Springer, Berlin, Heidelberg, December 2009. doi:10.1007/978-3-642-10366-7_13.

PP22. Omer Paneth and Rafael Pass. Incrementally verifiable computation via rate-1 batch arguments. In *63rd Annual Symposium on Foundations of Computer Science*, pages 1045–1056. IEEE Computer Society Press, October / November 2022. doi:10.1109/FOCS54457.2022.00102.

PS19. Chris Peikert and Sina Shiehian. Noninteractive zero knowledge for NP from (plain) learning with errors. In Alexandra Boldyreva and Daniele Micciancio, editors, *Advances in Cryptology – CRYPTO 2019, Part I*, volume 11692 of *Lecture Notes in Computer Science*, pages 89–114. Springer, Cham, August 2019. doi:10.1007/978-3-030-26948-7_4.

SC12. Jae Hong Seo and Jung Hee Cheon. Beyond the limitation of prime-order bilinear groups, and round optimal blind signatures. In Ronald Cramer, editor, *TCC 2012: 9th Theory of Cryptography Conference*, volume 7194 of *Lecture Notes in Computer Science*, pages 133–150. Springer, Berlin, Heidelberg, March 2012. doi:10.1007/978-3-642-28914-9_8.

SW05. Amit Sahai and Brent R. Waters. Fuzzy identity-based encryption. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 457–473. Springer, Berlin, Heidelberg, May 2005. doi:10.1007/11426639_27.

TAO16. Junichi Tomida, Masayuki Abe, and Tatsuaki Okamoto. Efficient functional encryption for inner-product values with full-hiding security. In Matt Bishop and Anderson C. A. Nascimento, editors, *ISC 2016: 19th International Conference on Information Security*, volume 9866 of *Lecture Notes in Computer Science*, pages 408–425. Springer, Cham, September 2016. doi:10.1007/978-3-319-45871-7_24.

WW22. Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 433–463. Springer, Cham, August 2022. doi:10.1007/978-3-031-15979-4_15.

A Verifiability for Membership of Source Groups

Languages for dual-encoding groups. Let $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ be a bilinear group generator. The underlying IPE scheme is instantiated with the parameters $r = k + \rho$ be the inner-product length, $p(\lambda)$ be the number of randomness dimensions, $q(r,\lambda)$ be the size of ciphertexts and functional keys. As in Definition 14, on input λ, $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ outputs $\mathsf{bgd} = (\mathsf{pp}, (\mathfrak{G}_1, \{\mathfrak{g}_{1,i}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_1, \tilde{\mathfrak{g}}_1), (\mathfrak{G}_2, \{\mathfrak{g}_{2,i}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_2, \tilde{\mathfrak{g}}_2), \mathfrak{G}_T, q, e, \{\mathfrak{r}_{1,i}, \mathfrak{r}_{2,i}\}_{i\in[p(\lambda)]})$ and $\mathsf{td} = \mathsf{sk}$.

Each $\mathfrak{G}_1$ element $[\boldsymbol{t}]_1 := [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$ can be decomposed as follows, following the structure presented in Definition 7, and in particular the linear homomorphism properties of functional keys and ciphertexts:

$$\begin{aligned}
[\boldsymbol{t}]_1 &= [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \\
&= (\mathsf{Enc}'(\mathsf{sk}, [x_1 a_{1,i}, \ldots, x_k a_{k,i}, \mathbf{0}_{1\times\rho}]_1; \boldsymbol{r}_i))_{i\in[k]} \\
&= ([x_1 a_{1,i}\boldsymbol{c}_{\mathsf{sk},1} + \cdots + x_k a_{k,i}\boldsymbol{c}_{\mathsf{sk},k} + r_{i,1}\boldsymbol{c}_{\mathsf{sk},k+1} \ldots r_{i,p(\lambda)}\boldsymbol{c}_{\mathsf{sk},k+p(\lambda)}]_1)_{i\in[k]} \\
&= (\sum_{j\in[k]} [x_j]_1 a_{j,i}\boldsymbol{c}_{\mathsf{sk},j} + \sum_{j\in[p(\lambda)]} [r_{i,j}]_1 \boldsymbol{c}_{\mathsf{sk},k+j})_{i\in[k]} \\
&= (\sum_{j\in[k]} [x_j]_1 a_{j,i} c_{\mathsf{sk},j,l} + \sum_{j\in[p(\lambda)]} [r_{i,j}]_1 c_{\mathsf{sk},k+j,l})_{i\in[k],l\in[q(r,\lambda)]}.
\end{aligned}$$

For each element $[t_l]_1$ of $[\boldsymbol{t}]_1$, we want to express the above relation as a multi-scalar multiplication. To this end, we concatenate the summands of the expression as vectors,

$$\begin{aligned}
\boldsymbol{x}^\star &:= (x_1 || \ldots || x_k) || (r_{i,1} || \ldots || r_{i,p(\lambda)})_{i\in[k]} \in \mathbb{Z}_q^{m^\star} \\
\boldsymbol{y}^\star &:= ((a_{1,i} c_{\mathsf{sk},1,l} || \ldots || a_{k,i} c_{\mathsf{sk},k,l})_{i\in[k]} || (c_{\mathsf{sk},k+1,l} || \ldots || c_{\mathsf{sk},k+p(\lambda),l}))_{l\in[q(r,\lambda)]} \in \mathbb{Z}_q^{n^\star}
\end{aligned}$$

where $m^\star = k(p(\lambda)+1)$ and $n^\star = (k^2 + p(\lambda))q(r,\lambda)$. Then, we define $\boldsymbol{\Gamma}^\star_{l^\star} \in \mathbb{Z}_q^{m^\star \times n^\star}$ to be the coefficient matrices corresponding to the expression above (containing only values in $\{0,1\}$):

$$[\boldsymbol{x}^{\star\intercal}]_1 \cdot \boldsymbol{\Gamma}^\star_{l^\star} \boldsymbol{y}^\star = [t_{l^\star}]_1 \ \forall l^\star \in [kq(r,\lambda)].$$

Since sk and $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\}$ are generated by the generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$, the vector $\boldsymbol{y}^\star$ is then generated by this algorithm and defines the group $\mathfrak{G}_1$. The membership language of $\mathfrak{G}_1$ can now be defined in the following.

Definition 17 (Membership language of $\mathfrak{G}_1$). *The language for membership of the group $\mathfrak{G}_1$, denoted by $L_{\mathfrak{G}_1}$, is parameterized by $\mathsf{param}_{\mathfrak{G}_1} := (\boldsymbol{y}^\star, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star})$ and defined such that for any statement $\mathtt{x}$:*

$$\mathtt{x} \in L_{\mathfrak{G}_1} \iff \begin{cases} \mathtt{x} = [\boldsymbol{t}]_1 \in \mathbb{G}_1^{kq(r,\lambda)} \\ \exists \mathtt{w} := [\boldsymbol{x}^\star]_1 \in \mathbb{G}_1^{m^\star} \ \ s.t.\ [\boldsymbol{x}^{\star\intercal}]_1 \cdot \boldsymbol{\Gamma}^\star_{l^\star} \boldsymbol{y}^\star = [t_{l^\star}]_1 \ \forall l^\star \in [kq(r,\lambda)]. \end{cases}$$

The membership language of $\mathfrak{G}_2$ is formalized in a similar way. Each $\mathfrak{G}_2$ element $[\boldsymbol{t}]_2 := [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$ can be decomposed as follows

$$\begin{aligned}
&[\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \\
&= (\mathsf{Enc}'(\mathsf{sk}, [y_1 b_{1,i}, \ldots, y_k b_{k,i}, \mathbf{0}_{1\times\rho}]_2; \boldsymbol{r}^*_i))_{i\in[k]} \\
&= ([y_1 b_{1,i}\boldsymbol{c}^*_{\mathsf{sk},1} + \cdots + y_k b_{k,i}\boldsymbol{c}^*_{\mathsf{sk},k} + r^*_{i,1}\boldsymbol{c}^*_{\mathsf{sk},k+1} \ldots r^*_{i,p(\lambda)}\boldsymbol{c}^*_{\mathsf{sk},k+p(\lambda)}]_2)_{i\in[k]} \\
&= (\sum_{j\in[k]} b_{j,i}\boldsymbol{c}^*_{\mathsf{sk},j}[y_j]_2 + \sum_{j\in[p(\lambda)]} \boldsymbol{c}^*_{\mathsf{sk},k+j}[r^*_{i,j}]_2)_{i\in[k]} \\
&= (\sum_{j\in[k]} b_{j,i}c^*_{\mathsf{sk},j,l}[y_j]_2 + \sum_{j\in[p(\lambda)]} c^*_{\mathsf{sk},k+j,l}[r^*_{i,j}]_2)_{i\in[k],l\in[q(r,\lambda)]}.
\end{aligned}$$

By concatenating

$$\begin{aligned}
\boldsymbol{x}^\dagger &:= ((b_{1,i}c^*_{\mathsf{sk},1,l}||\ldots||b_{k,i}c^*_{\mathsf{sk},k,l})_{i\in[k]}||(c^*_{\mathsf{sk},k+1,l}||\ldots||c^*_{\mathsf{sk},k+p(\lambda),l}))_{l\in[q(r,\lambda)]} \in \mathbb{Z}_q^{m^\dagger} \\
\boldsymbol{y}^\dagger &:= (y_1||\ldots||y_k)||(r^*_{i,1}||\ldots||r^*_{i,p(\lambda)})_{i\in[k]} \in \mathbb{Z}_q^{n^\dagger}
\end{aligned}$$

where $m^\dagger = (k^2 + p(\lambda))q(r,\lambda)$ and $n^\dagger = k(p(\lambda)+1)$. Then, we define $\boldsymbol{\Gamma}^\dagger_{l^\dagger} \in \mathbb{Z}_q^{m^\dagger\times n^\dagger}$ to be the coefficient matrices corresponding to the expression above (containing only values in $\{0,1\}$):

$$\boldsymbol{x}^{\dagger\intercal} \cdot \boldsymbol{\Gamma}^\dagger_{l^\dagger}[\boldsymbol{y}^\dagger]_2 = [t_{l^\dagger}]_2 \ \forall l^\dagger \in [kq(r,\lambda)].$$

Since sk and $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_k\}$ are generated by the generator $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$, the vector $\boldsymbol{x}^\dagger$ is also generated by this algorithm and defines the group $\mathfrak{G}_2$. The membership language of $\mathfrak{G}_2$ can now be defined in the following.

Definition 18 (Membership language of $\mathfrak{G}_2$). *The language for membership of the group $\mathfrak{G}_2$, denoted by $L_{\mathfrak{G}_2}$, is parameterized by $\mathsf{param}_{\mathfrak{G}_2} := (\boldsymbol{x}^\dagger, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$ and defined such that for any statement $\mathtt{x}$:*

$$\mathtt{x} \in L_{\mathfrak{G}_2} \iff \begin{cases} \mathtt{x} = [\boldsymbol{t}]_2 \in \mathbb{G}_2^{kq(r,\lambda)} \\ \exists \mathtt{w} := [\boldsymbol{y}^\dagger]_2 \in \mathbb{G}_2^{n^\dagger} \ \textit{s.t.}\ \boldsymbol{x}^{\dagger\intercal} \cdot \boldsymbol{\Gamma}^\dagger_{l^\dagger}[\boldsymbol{y}^\dagger]_2 = [t_{l^\dagger}]_2 \ \forall l^\dagger \in [kq(r,\lambda)]. \end{cases}$$

Next, we show that the languages $L_{\mathfrak{G}_1}$ and $L_{\mathfrak{G}_2}$ are supported by the Groth-Sahai proof system. Moreover, the system allows the secret-key vectors $\boldsymbol{y}^\star$ and $\boldsymbol{x}^\dagger$ to be committed once and reused as part of the common reference string, and its proofs are homomorphic with respect to the statements $[\boldsymbol{t}]_1$ and $[\boldsymbol{t}]_2$ and the message-randomness vectors $[\boldsymbol{x}^\star]_1$ and $[\boldsymbol{y}^\dagger]_2$, respectively.

Recap of the Groth-Sahai proof system [GS08]. Let $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$. The proof system additionally works over a bilinear additive group $(q, \mathbb{V}_1 := \mathbb{G}_1^2, \mathbb{V}_2 := \mathbb{G}_2^2, \mathbb{V}_T := \mathbb{G}_T^4, \tilde{e})$ that is induced from $\mathcal{P}$. For all elements $\mathsf{g} = [\boldsymbol{u}]_1 \in \mathbb{V}_1$ and $\mathsf{h} = [\boldsymbol{v}]_2 \in \mathbb{V}_2$, we have $\tilde{e}(\mathsf{g},\mathsf{h}) = [\boldsymbol{u}\otimes\boldsymbol{v}]_T$. We write $\mathsf{g} \bullet\!\!\!\sim \mathsf{h}$ as a shorthand for $\tilde{e}(\mathsf{g},\mathsf{h})$. Furthermore, for all vectors $\mathbf{g} \in \mathbb{V}_1^m$ and $\mathbf{h} \in \mathbb{V}_2^m$, we denote $\mathbf{g}^\intercal \bullet\!\!\!\sim \mathbf{h} := \sum_{i\in[m]} \tilde{e}(\mathsf{g}_i, \mathsf{h}_i)$.

The Groth-Sahai proof system supports the languages of multi-scalar multiplication equations either in $\mathbb{G}_1$, given by

$$\mathtt{x} := \{[\boldsymbol{a}_l]_1, \boldsymbol{b}_l, \boldsymbol{\Gamma}_l, [t_l]_1\}_l \in L \iff \begin{array}{l} \exists \mathtt{w} := ([\boldsymbol{x}]_1, \boldsymbol{y}) \text{ s.t.} \\ [\boldsymbol{a}_l^\intercal]_1 \cdot \boldsymbol{y} + [\boldsymbol{x}^\intercal]_1 \cdot \boldsymbol{b}_l + [\boldsymbol{x}^\intercal]_1 \cdot \boldsymbol{\Gamma}_l\boldsymbol{y} = [t_l]_1 \ \forall l, \end{array}$$

or in $\mathbb{G}_2$, given by

$$\mathtt{x} := \{\boldsymbol{a}_l, [\boldsymbol{b}_l]_2, \boldsymbol{\Gamma}_l, [t_l]_2\}_l \in L \iff \begin{array}{l} \exists \mathtt{w} := (\boldsymbol{x}, [\boldsymbol{y}]_2) \text{ s.t.} \\ \boldsymbol{a}_l^\intercal \cdot [\boldsymbol{y}]_2 + \boldsymbol{x}^\intercal \cdot [\boldsymbol{b}_l]_2 + \boldsymbol{x}^\intercal \cdot \boldsymbol{\Gamma}_l [\boldsymbol{y}]_2 = [t_l]_2 \ \forall l. \end{array}$$

The proof system operates over commitments to inputs in each domain of $\{\mathbb{Z}_q, \mathbb{V}_1, \mathbb{V}_2\}$. For inputs in $\mathbb{G}_1$ and $\mathbb{G}_2$, the commitment keys are denoted by $\mathsf{ck}_1 := \mathbf{u} \in \mathbb{V}_1^2$ and $\mathsf{ck}_2 := \mathbf{v} \in \mathbb{V}_2^2$, respectively. For inputs in $\mathbb{Z}_q$, a commitment key is denoted by either $\mathsf{ck}'_1 := \mathsf{u} \in \mathbb{V}_1$ or $\mathsf{ck}'_2 := \mathsf{v} \in \mathbb{V}_2$, depending on whether the commitment maps the input to $\mathbb{V}_1$ or $\mathbb{V}_2$, respectively. In the soundness mode, commitment keys are generated to be perfectly binding, whereas in the zero-knowledge mode they are generated to be perfectly hiding. Moreover, commitment keys generated in these two modes are computationally indistinguishable under the Symmetric External Diffie-Hellman (SXDH) assumption in $\mathbb{G}_1$ and $\mathbb{G}_2$.

We will also use the inclusion maps $(\iota_1 : \mathbb{G}_1 \to \mathbb{V}_1, \iota'_1 : \mathbb{Z}_q \to \mathbb{V}_1)$ and $(\iota_2 : \mathbb{G}_2 \to \mathbb{V}_2, \iota'_2 : \mathbb{Z}_q \to \mathbb{V}_2)$, along with the maps $\tilde{\iota}_T : \mathbb{G}_1 \to \mathbb{V}_T$ and $\hat{\iota}_T : \mathbb{G}_2 \to \mathbb{V}_T$. These maps are defined under the same notation in [GS08] and are applied to vectors element-wise.

Taken together, we present the Groth-Sahai proof system for special cases when $\boldsymbol{a}_l = \boldsymbol{b}_l = \mathbf{0}$ in asymmetric prime-order bilinear groups.

$\underline{\mathsf{Setup}(1^\lambda)}$: On input the security parameter λ, the setup algorithm proceeds as follows:

1. Generate $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$;
2. Generate commitment keys $\mathsf{ck}_1 = \mathbf{u} \in \mathbb{V}_1^2$ and $\mathsf{ck}_2 = \mathbf{v} \in \mathbb{V}_2^2$ for the soundness mode.
3. The commitment keys ck'_1 and ck'_2 are implicitly set as $\mathsf{ck}'_1 = \mathsf{u}_1$ and $\mathsf{ck}'_2 = \mathsf{v}_1$, respectively.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2)$.

$\underline{\mathsf{SetupTd}(1^\lambda)}$: On input the security parameter λ, the setup algorithm proceeds as follows:

1. Generate $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$;
2. Generate commitment keys $\mathsf{ck}_1 = \mathbf{u} \in \mathbb{V}_1^2$ and $\mathsf{ck}_2 = \mathbf{v} \in \mathbb{V}_2^2$ for the zero-knowledge mode, along with a trapdoor td.
3. The commitment keys ck'_1 and ck'_2 are implicitly set as $\mathsf{ck}'_1 = \mathsf{u}_1$ and $\mathsf{ck}'_2 = \mathsf{v}_1$, respectively.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2)$.

$\underline{\mathsf{Prove}(\mathsf{crs}, \mathtt{x}, \mathtt{w})}$: On input the $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2)$,

– if the input statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_1\}_l$ where $\boldsymbol{\Gamma}_l \in \mathbb{Z}_q^{m \times n}$ and the input witness $\mathtt{w} = ([\boldsymbol{x}]_1, \boldsymbol{y}) \in \mathbb{G}_1^m \times \mathbb{Z}_q^n$, the prover proceeds as follows:

1. Commit to group-element witness $[\boldsymbol{x}]_1$:

$$\mathsf{com}(\mathsf{ck}_1, [\boldsymbol{x}]_1; \boldsymbol{R}) := \mathbf{c} = \iota_1([\boldsymbol{x}]_1) + \boldsymbol{R}\mathbf{u} \in \mathbb{V}_1^m$$

where $\boldsymbol{R} \leftarrow\!\!\$\ \mathbb{Z}_q^{m \times 2}$.

2. Commit to scalar witness $\boldsymbol{y}$:

$$\mathsf{com}'(\mathsf{ck}'_2, \boldsymbol{y}; \boldsymbol{s}) := \mathbf{d}' = \iota'_2(\boldsymbol{y}) + \boldsymbol{s}\mathsf{v}_1 \in \mathbb{V}_2^n$$

where $\boldsymbol{s} \leftarrow\!\!\$\ \mathbb{Z}_q^n$.

3. Validity of each equation l:

$$\begin{aligned} \boldsymbol{\delta}_l &:= \boldsymbol{R}^\intercal \boldsymbol{\Gamma}_l \iota'_2(\boldsymbol{y}) + (\boldsymbol{R}^\intercal \boldsymbol{\Gamma}_l \boldsymbol{s} - \boldsymbol{T}^\intercal)\mathsf{v}_1 \in \mathbb{V}_2^2 \\ \theta_l &:= \boldsymbol{s}^\intercal \boldsymbol{\Gamma}_l^\intercal \iota_1([\boldsymbol{x}]_1) + \boldsymbol{T}\mathbf{u} \in \mathbb{V}_1 \end{aligned}$$

where $\boldsymbol{T} \leftarrow\!\!\$\ \mathbb{Z}_q^{1 \times 2}$.

Output the proof $\pi = (\mathbf{c}, \mathbf{d'}, \{\boldsymbol{\delta}_l, \theta_l\}_l)$.

– if the input statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_2\}_l$ where $\boldsymbol{\Gamma}_l \in \mathbb{Z}_q^{m\times n}$ and the input witness $\mathtt{w} = (\boldsymbol{x}, [\boldsymbol{y}]_2) \in \mathbb{Z}_q^m \times \mathbb{G}_2^n$, the prover proceeds as follows:

1. Commit to scalar witness $\boldsymbol{x}$:

$$\mathsf{com}'(\mathsf{ck}'_1, \boldsymbol{x}; \boldsymbol{r}) := \mathbf{c'} = \iota'_1(\boldsymbol{x}) + \boldsymbol{r}\mathsf{u}_1 \in \mathbb{V}_1^m$$

where $\boldsymbol{r} \leftarrow\!\!\$\ \mathbb{Z}_q^m$.

2. Commit to group-element witness $[\boldsymbol{y}]_2$:

$$\mathsf{com}(\mathsf{ck}_2, [\boldsymbol{y}]_2; \boldsymbol{S}) := \mathbf{d} = \iota_2([\boldsymbol{y}]_2) + \boldsymbol{S}\mathbf{v} \in \mathbb{V}_2^n$$

where $\boldsymbol{S} \leftarrow\!\!\$\ \mathbb{Z}_q^{n\times 2}$.

3. Validity of each equation l:

$$\delta_l := \boldsymbol{r}^\mathsf{T}\boldsymbol{\Gamma}_l\iota_2([\boldsymbol{y}]_2) + (\boldsymbol{r}^\mathsf{T}\boldsymbol{\Gamma}_l\boldsymbol{S} - \boldsymbol{T}^\mathsf{T})\mathbf{v} \in \mathbb{V}_2$$
$$\boldsymbol{\theta}_l := \boldsymbol{S}^\mathsf{T}\boldsymbol{\Gamma}_l^\mathsf{T}\iota'_1(\boldsymbol{x}) + \boldsymbol{T}\mathsf{u}_1 \in \mathbb{V}_1^2$$

where $\boldsymbol{T} \leftarrow\!\!\$\ \mathbb{Z}_q^{2\times 1}$.

Output the proof $\pi = (\mathbf{c'}, \mathbf{d}, \{\delta_l, \boldsymbol{\theta}_l\}_l)$.

$\underline{\mathsf{Ver}(\mathsf{crs}, \mathtt{x}, \pi):}$ On input the $\mathsf{crs} = (\mathsf{bgd}_\mathcal{P}, \mathsf{ck}_1, \mathsf{ck}_2)$,

– if the input statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_1\}_l$ and the input proof $\pi = (\mathbf{c}, \mathbf{d'}, \{\boldsymbol{\delta}_l, \theta_l\}_l)$, the verifier outputs 1 as valid if it holds that

$$\mathbf{c}^\mathsf{T} \bullet \boldsymbol{\Gamma}_l\mathbf{d'} = \tilde{\iota}_T([t_l]_1) + \mathbf{u}^\mathsf{T} \bullet \boldsymbol{\delta}_l + \tilde{e}(\theta_l, \mathsf{v}_1)$$

and outputs 0 otherwise.

– if the input statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_2\}_l$ and the input proof $\pi = (\mathbf{c'}, \mathbf{d}, \{\delta_l, \boldsymbol{\theta}_l\}_l)$, the verifier outputs 1 as valid if it holds that

$$\mathbf{c'}^\mathsf{T} \bullet \boldsymbol{\Gamma}_l\mathbf{d} = \hat{\iota}_T([t_l]_2) + \tilde{e}(\mathsf{u}_1, \delta_l) + \boldsymbol{\theta}_l^\mathsf{T} \bullet \mathbf{v}$$

and outputs 0 otherwise.

From the scheme description, we remark the following facts.

Remark 9 (Linearly homomorphic proofs). Let $\hat{\pi} = (\hat{\mathbf{c}}, \mathbf{d'}, \{\hat{\boldsymbol{\delta}}_l, \hat{\theta}_l\}_l)$ and $\bar{\pi} = (\bar{\mathbf{c}}, \mathbf{d'}, \{\bar{\boldsymbol{\delta}}_l, \bar{\theta}_l\}_l)$ be valid proofs with respect to statements $\hat{\mathtt{x}} = \{\boldsymbol{\Gamma}_l, [\hat{t}_l]_1\}_l$ and $\bar{\mathtt{x}} = \{\boldsymbol{\Gamma}_l, [\bar{t}_l]_1\}_l$, respectively. For any $\hat{\alpha}$ and $\bar{\alpha}$, the proof

$$\pi := (\hat{\alpha}\hat{\mathbf{c}} + \bar{\alpha}\bar{\mathbf{c}}, \mathbf{d'}, \{\hat{\alpha}\hat{\boldsymbol{\delta}}_l + \bar{\alpha}\bar{\boldsymbol{\delta}}_l, \hat{\alpha}\hat{\theta}_l + \bar{\alpha}\bar{\theta}_l\}_l)$$

is valid with respect to the statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [\hat{\alpha}\hat{t}_l + \bar{\alpha}\bar{t}_l]_1\}_l$.

Similarly, let $\hat{\pi} = (\mathbf{c'}, \hat{\mathbf{d}}, \{\hat{\delta}_l, \hat{\boldsymbol{\theta}}_l\}_l)$ and $\bar{\pi} = (\mathbf{c'}, \bar{\mathbf{d}}, \{\bar{\delta}_l, \bar{\boldsymbol{\theta}}_l\}_l)$ be valid proofs with respect to statements $\hat{\mathtt{x}} = \{\boldsymbol{\Gamma}_l, [\hat{t}_l]_2\}_l$ and $\bar{\mathtt{x}} = \{\boldsymbol{\Gamma}_l, [\bar{t}_l]_2\}_l$, respectively. For any $\hat{\alpha}$ and $\bar{\alpha}$, the proof

$$\pi := (\mathbf{c'}, \hat{\alpha}\hat{\mathbf{d}} + \bar{\alpha}\bar{\mathbf{d}}, \{\hat{\alpha}\hat{\delta}_l + \bar{\alpha}\bar{\delta}_l, \hat{\alpha}\hat{\boldsymbol{\theta}}_l + \bar{\alpha}\bar{\boldsymbol{\theta}}_l\}_l)$$

is valid with respect to the statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [\hat{\alpha}\hat{t}_l + \bar{\alpha}\bar{t}_l]_2\}_l$.

Remark 10 (Commit-and-prove mode). Let crs be generated for the soundness mode from $\mathsf{Setup}(1^\lambda)$, $\{\boldsymbol{\Gamma}_l\}_l$ be a coefficient set of equations, $\mathbf{d'} = \mathsf{com}'(\mathsf{ck}'_2, \boldsymbol{y}; \boldsymbol{s})$ and $\mathbf{c'} = \mathsf{com}'(\mathsf{ck}'_1, \boldsymbol{x}; \boldsymbol{r})$ be commitments to $\boldsymbol{y}$ and $\boldsymbol{x}$, respectively.

Given all valid proofs $\pi = (\mathbf{c}, \mathbf{d}', \{\boldsymbol{\delta}_l, \theta_l\}_l)$ with respect to crs and the statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_1\}_l$, it additionally holds that

$$\exists [\boldsymbol{x}']_1 \in \mathbb{G}_1^m \text{ s.t. } [\boldsymbol{x}'^{\mathsf{T}}]_1 \cdot \boldsymbol{\Gamma}_l \boldsymbol{y} = [t_l]_1 \ \forall l.$$

Similarly, given all valid proofs $\pi = (\mathbf{c}', \mathbf{d}, \{\delta_l, \boldsymbol{\theta}_l\}_l)$ with respect to crs and the statement $\mathtt{x} = \{\boldsymbol{\Gamma}_l, [t_l]_2\}_l$, it additionally holds that

$$\exists [\boldsymbol{y}']_2 \in \mathbb{G}_2^n \text{ s.t. } \boldsymbol{x}^{\mathsf{T}} \cdot \boldsymbol{\Gamma}_l [\boldsymbol{y}']_2 = [t_l]_2 \ \forall l.$$

Membership Verifiability from Groth-Sahai Proofs. From Remark 9 and Remark 10, we recast the Groth-Sahai proof system for membership languages $L_{\mathfrak{G}_1}$ and $L_{\mathfrak{G}_2}$, as given in Definition 17 and Definition 18, respectively.

$\underline{\mathsf{Setup}(1^\lambda, \mathsf{param}_{\mathfrak{G}_1}, \mathsf{param}_{\mathfrak{G}_2})}$: On input the security parameter λ and the parameters $\mathsf{param}_{\mathfrak{G}_1} := (\boldsymbol{y}^\star, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star})$ for $L_{\mathfrak{G}_1}$ and $\mathsf{param}_{\mathfrak{G}_2} := (\boldsymbol{x}^\dagger, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$ for $L_{\mathfrak{G}_2}$, the setup algorithm proceeds as follows:

1. Generate $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$;
2. Generate commitment keys $\mathsf{ck}_1 = \mathbf{u}$ and $\mathsf{ck}_2 = \mathbf{v}$ for the soundness mode.
3. The commitment keys ck'_1 and ck'_2 are implicitly set as $\mathsf{ck}'_1 = \mathsf{u}_1$ and $\mathsf{ck}'_2 = \mathsf{v}_1$, respectively.
4. Commit to $\boldsymbol{y}^\star$:
$$\mathbf{d'} = \mathsf{com}'(\mathsf{ck}'_2, \boldsymbol{y}^\star; \boldsymbol{s})$$
where $\boldsymbol{s} \leftarrow\!\$\ \mathbb{Z}_q^{n^\star}$.
5. Commit to $\boldsymbol{x}^\dagger$:
$$\mathbf{c'} = \mathsf{com}'(\mathsf{ck}'_1, \boldsymbol{x}^\dagger; \boldsymbol{r})$$
where $\boldsymbol{r} \leftarrow\!\$\ \mathbb{Z}_q^{m^\dagger}$.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2, \mathbf{d'}, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star}, \mathbf{c'}, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$.

$\underline{\mathsf{SetupTd}(1^\lambda, \mathsf{param}_{\mathfrak{G}_1}, \mathsf{param}_{\mathfrak{G}_2})}$: On input the security parameter λ and the parameters $\mathsf{param}_{\mathfrak{G}_1} := (\boldsymbol{y}^\star, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star})$ for $L_{\mathfrak{G}_1}$ and $\mathsf{param}_{\mathfrak{G}_2} := (\boldsymbol{x}^\dagger, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$ for $L_{\mathfrak{G}_2}$, the setup algorithm proceeds as follows:

1. Generate $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$;
2. Generate commitment keys $\mathsf{ck}_1 = \mathbf{u}$ and $\mathsf{ck}_2 = \mathbf{v}$ for the zero-knowledge mode, along with a trapdoor td.
3. The commitment keys ck'_1 and ck'_2 are implicitly set as $\mathsf{ck}'_1 = \mathsf{u}_1$ and $\mathsf{ck}'_2 = \mathsf{v}_1$, respectively.
4. Commit to $\boldsymbol{y}^\star$:
$$\mathbf{d'} = \mathsf{com}'(\mathsf{ck}'_2, \boldsymbol{y}^\star; \boldsymbol{s})$$
where $\boldsymbol{s} \leftarrow\!\$\ \mathbb{Z}_q^{n^\star}$.
5. Commit to $\boldsymbol{x}^\dagger$:
$$\mathbf{c'} = \mathsf{com}'(\mathsf{ck}'_1, \boldsymbol{x}^\dagger; \boldsymbol{r})$$
where $\boldsymbol{r} \leftarrow\!\$\ \mathbb{Z}_q^{m^\dagger}$.

Output the common reference string $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2, \mathbf{d'}, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star}, \mathbf{c'}, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$ and the trapdoor td.

$\underline{\mathsf{Prove}(\mathsf{crs}, \mathtt{x}, \mathtt{w})}$: On input $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2, \mathbf{d'}, \{\boldsymbol{\Gamma}^\star_{l^\star}\}_{l^\star}, \mathbf{c'}, \{\boldsymbol{\Gamma}^\dagger_{l^\dagger}\}_{l^\dagger})$,

– Parsing the input statement $\mathtt{x} = [\boldsymbol{t}]_1$ and the input witness $\mathtt{w} = ([\boldsymbol{x}]_1, \boldsymbol{y}^\star, \boldsymbol{s})$, the prover proves $\mathtt{x} \in L_{\mathfrak{G}_1}$ as follows:
  1. Commit to group-element witness $[\boldsymbol{x}]_1$:
  $$\mathsf{c} = \iota_1([\boldsymbol{x}]_1) + \boldsymbol{R}\mathsf{u}$$
  where $\boldsymbol{R} \leftarrow\!\!\$\ \mathbb{Z}_q^{m\times 2}$.
  2. Validity of each equation $l^\star$:
  $$\begin{aligned}\boldsymbol{\delta}_{l^\star} &:= \boldsymbol{R}^\intercal \boldsymbol{\Gamma}_{l^\star} \iota_2'(\boldsymbol{y}^\star) + (\boldsymbol{R}^\intercal \boldsymbol{\Gamma}_{l^\star}\boldsymbol{s} - \boldsymbol{T}^\intercal)\mathsf{v}_1\\ \theta_{l^\star} &:= \boldsymbol{s}^\intercal \boldsymbol{\Gamma}_{l^\star}^\intercal \iota_1([\boldsymbol{x}]_1) + \boldsymbol{T}\mathsf{u}\end{aligned}$$
  where $\boldsymbol{T} \leftarrow\!\!\$\ \mathbb{Z}_q^{1\times 2}$.

  Output the proof $\pi = (\mathsf{c}, \{\boldsymbol{\delta}_{l^\star}, \theta_l\}_{l^\star})$.
– Parsing the input statement $\mathtt{x} = [\boldsymbol{t}]_2$ and the input witness $\mathtt{w} = (\boldsymbol{x}^\dagger, \boldsymbol{r}, [\boldsymbol{y}]_2)$, the prover proves $\mathtt{x} \in L_{\mathfrak{G}_2}$ as follows:
  1. Commit to group-element witness $[\boldsymbol{y}]_2$:
  $$\mathsf{d} = \iota_2([\boldsymbol{y}]_2) + \boldsymbol{S}\mathsf{v}$$
  where $\boldsymbol{S} \leftarrow\!\!\$\ \mathbb{Z}_q^{n^\dagger\times 2}$.
  2. Validity of each equation $l^\dagger$:
  $$\begin{aligned}\delta_{l^\dagger} &:= \boldsymbol{r}^\intercal \boldsymbol{\Gamma}_{l^\dagger} \iota_2([\boldsymbol{y}]_2) + (\boldsymbol{r}^\intercal \boldsymbol{\Gamma}_{l^\dagger}\boldsymbol{S} - \boldsymbol{T}^\intercal)\mathsf{v}\\ \boldsymbol{\theta}_l &:= \boldsymbol{S}^\intercal \boldsymbol{\Gamma}_{l^\dagger}^\intercal \iota_1'(\boldsymbol{x}^\dagger) + \boldsymbol{T}\mathsf{u}_1\end{aligned}$$
  where $\boldsymbol{T} \leftarrow\!\!\$\ \mathbb{Z}_q^{2\times 1}$.

  Output the proof $\pi = (\mathsf{d}, \{\delta_{l^\dagger}, \boldsymbol{\theta}_{l^\dagger}\}_{l^\dagger})$.

$\underline{\mathsf{Deriv}(\mathsf{crs}, \{\alpha_i, \mathtt{x}_i, \pi_i\}_i):}$ On input crs and linear coefficients $\alpha_i \in \mathbb{Z}_q$,

– Parsing the statements $\mathtt{x}_i = [\boldsymbol{t}^{(i)}]_1$ and the proofs $\pi_i = (\mathsf{c}^{(i)}, \{\boldsymbol{\delta}_{l^\star}^{(i)}, \theta_{l^\star}^{(i)}\}_{l^\star})$, a proof for the statement $\mathtt{x} := \sum_i \alpha_i [\boldsymbol{t}^{(i)}]_1$ is derived as follows:
$$\mathsf{c} := \sum_i \alpha_i \mathsf{c}^{(i)} \qquad \boldsymbol{\delta}_{l^\star} := \sum_i \alpha_i \boldsymbol{\delta}_{l^\star}^{(i)}\ \forall l^\star \qquad \theta_{l^\star} := \sum_i \alpha_i \theta_{l^\star}^{(i)}\ \forall l^\star$$
Output $\mathtt{x}$ and $\pi := (\mathsf{c}, \{\boldsymbol{\delta}_{l^\star}, \theta_{l^\star}\}_{l^\star})$.

– Parsing the statements $\mathtt{x}_i = [\boldsymbol{t}^{(i)}]_2$ and the proofs $\pi_i = (\mathsf{d}^{(i)}, \{\delta_l^{(i)}, \boldsymbol{\theta}_{l^\dagger}^{(i)}\}_{l^\dagger})$, a proof for the statement $\mathtt{x} := \sum_i \alpha_i [\boldsymbol{t}^{(i)}]_2$ is derived as follows:
$$\mathsf{d} := \sum_i \alpha_i \mathsf{d}^{(i)} \qquad \delta_{l^\dagger} := \sum_i \alpha_i \delta_{l^\dagger}^{(i)}\ \forall l^\dagger \qquad \boldsymbol{\theta}_{l^\dagger} := \sum_i \alpha_i \boldsymbol{\theta}_{l^\dagger}^{(i)}\ \forall l^\dagger$$
Output $\mathtt{x}$ and the proof $\pi = (\mathsf{d}, \{\delta_l, \boldsymbol{\theta}_l\}_l)$.

$\underline{\mathsf{Ver}(\mathsf{crs}, \mathtt{x}, \pi):}$ On input the $\mathsf{crs} = (\mathsf{bgd}_{\mathcal{P}}, \mathsf{ck}_1, \mathsf{ck}_2, \mathsf{d}', \{\boldsymbol{\Gamma}_{l^\star}^\star\}_{l^\star}, \mathsf{c}', \{\boldsymbol{\Gamma}_{l^\dagger}^\dagger\}_{l^\dagger})$,

– Parsing the statement $\mathtt{x} = [\boldsymbol{t}]_1$ and the proof $\pi = (\mathsf{c}, \{\boldsymbol{\delta}_{l^\star}, \theta_{l^\star}\}_{l^\star})$, the verifier verifies $\mathtt{x} \in L_{\mathfrak{G}_1}$ and outputs 1 as valid if it holds that
$$\mathsf{c}^\intercal \tilde{\bullet}\ \boldsymbol{\Gamma}_{l^\star}\mathsf{d}' = \tilde{\iota}_T([t_{l^\star}]_1) + \mathsf{u}^\intercal \tilde{\bullet}\ \boldsymbol{\delta}_{l^\star} + \tilde{e}(\theta_{l^\star}, \mathsf{v}_1)$$
and 0 otherwise.

– parsing the statement $\mathtt{x} = [\boldsymbol{t}]_2$ and the proof $\pi = (\mathbf{d}, \{\delta_{l\dagger}, \boldsymbol{\theta}_{l\dagger}\}_{l\dagger})$, the verifier verifies $\mathtt{x} \in L_{\mathfrak{G}_2}$ and outputs 1 as valid if it holds that

$$\mathbf{c'}^{\mathsf{T}} \bullet \boldsymbol{\Gamma}_{l\dagger}\mathbf{d} = \hat{\imath}_T([t_{l\dagger}]_2) + \tilde{e}(\mathsf{u}_1, \delta_{l\dagger}) + \boldsymbol{\theta}_{l\dagger}^{\mathsf{T}} \tilde{\bullet} \mathbf{v}$$

and 0 otherwise.

For publicly verifiable membership of source groups, we highlight in blue the components that are added to the bilinear group generator in Definition 14.

Definition 19 ($\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ with Verifiable Membership). *Let $\mathcal{P}$ be a prime-order bilinear group generator, $\mathsf{IPE} = (\mathsf{IPE.SetUp}, \mathsf{DKGen}, \mathsf{Enc}, \mathsf{Dec})$ be an FH-IPFE scheme over groups generated by $\mathcal{P}$ and $\mathsf{GS} = (\mathsf{GS.Setup}, \mathsf{GS.SetupTd}, \mathsf{Prove}, \mathsf{Deriv}, \mathsf{Ver})$ be a Groth-Sahai-based membership proof system also over groups generated by $\mathcal{P}$. For integers $k \geq 2$ and ρ, we define $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ with verifiable membership of source groups to be a bilinear group generator that on input λ does the following:*

1. *Let $\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(\lambda)$ and $(\mathsf{pp}, \mathsf{sk}) \leftarrow \mathsf{IPE.SetUp}(1^\lambda, 1^{k+\rho})$. Include $\mathsf{bgd}_{\mathcal{P}}$ in pp.*
2. *Choose random $\boldsymbol{a}_i, \boldsymbol{b}_i \in \mathbb{Z}_q^k$ for $i \in [k]$, such that the sets $\{\boldsymbol{a}_i\}_{i\in[k]}$ and $\{\boldsymbol{b}_i\}_{i\in[k]}$ are each linearly independent in the $\mathbb{Z}_q$-vector space $\mathbb{Z}_q^k$. Include $\{\boldsymbol{a}_i, \boldsymbol{b}_i\}_{i\in[k]}$ into sk.*
3. *Set up $\mathsf{param}_{\mathfrak{G}_1}$ for $L_{\mathfrak{G}_1}$ and $\mathsf{param}_{\mathfrak{G}_2}$ for $L_{\mathfrak{G}_2}$ from the appropriate information in sk. Let $\mathsf{crs} \leftarrow \mathsf{GS.Setup}(1^\lambda, \mathsf{param}_{\mathfrak{G}_1}, \mathsf{param}_{\mathfrak{G}_2})$.*
4. *Parse*
   – $\mathfrak{G}_1 := \{[\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 : x_i \in \mathbb{Z}_q \ \forall i \in [k]\}$,
   – $\mathfrak{G}_2 := \{[\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 : y_i \in \mathbb{Z}_q \ \forall i \in [k]\}$,
   – $\mathfrak{G}_T := \{[z_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \cdots + z_k(\boldsymbol{a}_k \otimes \boldsymbol{b}_k)]_T : z_i \in \mathbb{Z}_q \ \forall i \in [k]\}$

   as groups.
5. *For any tuple of coefficients $(x_1, \ldots, x_k) \in \mathbb{Z}_q^k$, denote by*
   – $\mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k)$ *the distribution induced by*

$$\mathfrak{g} \leftarrow (\mathsf{DKGen}(\mathsf{sk}, [x_1 a_{1,s}, \ldots, x_k a_{k,s}, \mathbf{0}_{1\times\rho}]_1))_{s\in[k]} \text{ and}$$
$$\pi_{\mathfrak{g}} \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathfrak{g}, \mathtt{w})$$

   where $\mathtt{w}$ is the appropriate witness from the state.
   – $\mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, x_1\boldsymbol{b}_1, \ldots, x_k\boldsymbol{b}_k)$ *the distribution induced by*

$$\mathfrak{h} \leftarrow (\mathsf{Enc}(\mathsf{sk}, [x_1 b_{1,s}, \ldots, x_k b_{k,s}, \mathbf{0}_{1\times\rho}]_2))_{s\in[k]} \text{ and}$$
$$\pi_{\mathfrak{h}} \leftarrow \mathsf{Prove}(\mathsf{crs}, \mathfrak{h}, \mathtt{w})$$

   where $\mathtt{w}$ is the appropriate witness from the state.
6. *Choose random full-rank matrices $\mathbf{A}, \mathbf{B} \in \mathbb{Z}_q^{k\times k}$.*
7. *Generate sets of group generators for $\mathfrak{G}_1$ and $\mathfrak{G}_2$: For $i \in [k]$,*
   – $\mathfrak{g}_{1,i} := [\![A_{i,1}\boldsymbol{a}_1, \ldots, A_{i,k}\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$ *and $\pi_{\mathfrak{g}_{1,i}}$ where*

$$(\mathfrak{g}_{1,i}, \pi_{\mathfrak{g}_{1,i}}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, A_{i,1}\boldsymbol{a}_1, \ldots, A_{i,k}\boldsymbol{a}_k),$$

   – $\mathfrak{g}_{2,i} := [\![B_{i,1}\boldsymbol{b}_1, \ldots, B_{i,k}\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$ *and $\pi_{\mathfrak{g}_{2,i}}$ where*

$$(\mathfrak{g}_{2,i}, \pi_{\mathfrak{g}_{2,i}}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, B_{i,1}\boldsymbol{b}_1, \ldots, B_{i,k}\boldsymbol{b}_k).$$

8. *Parse*
   - $\tilde{\mathfrak{G}}_1 := \{[\![x_1\boldsymbol{a}_1, 0\boldsymbol{a}_2, \ldots, 0\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 : x_1 \in \mathbb{Z}_q\}$,
   - $\tilde{\mathfrak{G}}_2 := \{[\![y_1\boldsymbol{b}_1, 0\boldsymbol{b}_2, \ldots, 0\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 : y_1 \in \mathbb{Z}_q\}$

   as the subgroups of $\mathfrak{G}_1$ *and* $\mathfrak{G}_2$*, respectively.*
9. *Generate subgroup generators for* $\tilde{\mathfrak{G}}_1$ *and* $\tilde{\mathfrak{G}}_2$*: Let* $\tilde{x}, \tilde{y} \leftarrow\!\!\$\ \mathbb{Z}_q$,
   - $\tilde{\mathfrak{g}}_1 := [\![\tilde{x}\boldsymbol{a}_1, 0\boldsymbol{a}_2, \ldots, 0\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$ *and* $\pi_{\tilde{\mathfrak{g}}_1}$ *where*

$$(\tilde{\mathfrak{g}}_1, \pi_{\tilde{\mathfrak{g}}_1}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, \tilde{x}\boldsymbol{a}_1, 0\boldsymbol{a}_2, \ldots, 0\boldsymbol{a}_k),$$

   - $\tilde{\mathfrak{g}}_2 := [\![\tilde{y}\boldsymbol{b}_1, 0\boldsymbol{b}_2, \ldots, 0\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$ *and* $\pi_{\tilde{\mathfrak{g}}_2}$ *where*

$$(\tilde{\mathfrak{g}}_2, \pi_{\tilde{\mathfrak{g}}_2}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, \tilde{y}\boldsymbol{b}_1, 0\boldsymbol{b}_2, \ldots, 0\boldsymbol{b}_k).$$

10. *Generate randomizers for encodings: Let* $p(\lambda)$ *be the size of randomness used in functional keys and ciphertexts of* IPE*, for each* $i \in [p(\lambda)]$*, compute*
   - *a functional-key randomizer* $\mathfrak{r}_{1,i} := [\![0\boldsymbol{a}_1, \ldots, 0\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1$ *and* $\pi_{\mathfrak{r}_{1,i}}$ *where*

$$(\mathfrak{r}_{1,i}, \pi_{\mathfrak{r}_{1,i}}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_1}(\mathsf{sk}, 0\boldsymbol{a}_1, \ldots, 0\boldsymbol{a}_k),$$

   - *a ciphertext randomizer* $\mathfrak{r}_{2,i} := [\![0\boldsymbol{b}_1, \ldots, 0\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2$ *and* $\pi_{\mathfrak{r}_{2,i}}$ *where*

$$(\mathfrak{r}_{2,i}, \pi_{\mathfrak{r}_{2,i}}) \leftarrow \mathcal{D}^{k,\rho}_{\mathbb{G}_2}(\mathsf{sk}, 0\boldsymbol{b}_1, \ldots, 0\boldsymbol{b}_k).$$

11. *Define the map* $e : \mathfrak{G}_1 \times \mathfrak{G}_2 \to \mathfrak{G}_T$ *as follows: Given any two elements* $\mathfrak{g} = [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \in \mathfrak{G}_1$ *and* $\mathfrak{h} = [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2 \in \mathfrak{G}_2$*, set*

$$e(\mathfrak{g}, \mathfrak{h}) := [\![x_1\boldsymbol{a}_1, \ldots, x_k\boldsymbol{a}_k, \mathbf{0}_{k\times\rho}]\!]_1 \bullet [\![y_1\boldsymbol{b}_1, \ldots, y_k\boldsymbol{b}_k, \mathbf{0}_{k\times\rho}]\!]_2.$$

12. *Define the verification algorithms* $\mathsf{Ver}_{\mathfrak{G}_1}$ *and* $\mathsf{Ver}_{\mathfrak{G}_2}$ *as follows:*
   - *For any pair of statement and proof* $(\mathfrak{g}, \pi_{\mathfrak{g}})$ *for* $L_{\mathfrak{G}_1}$*,*

$$\mathsf{Ver}_{\mathfrak{G}_1}(\mathfrak{g}) = 1 \iff \mathsf{Ver}(\mathsf{crs}, \mathfrak{g}, \pi_{\mathfrak{g}}) = 1.$$

   - *For any pair of statement and proof* $(\mathfrak{h}, \pi_{\mathfrak{h}})$ *for* $L_{\mathfrak{G}_2}$*,*

$$\mathsf{Ver}_{\mathfrak{G}_2}(\mathfrak{h}) = 1 \iff \mathsf{Ver}(\mathsf{crs}, \mathfrak{h}, \pi_{\mathfrak{h}}) = 1.$$

13. *Output* $\mathsf{bgd} = \begin{cases} \mathsf{pp}, \mathsf{crs}, (\mathfrak{G}_1, \{\mathfrak{g}_{1,i}, \pi_{\mathfrak{g}_{1,i}}\}_{i\in[k]}), (\tilde{\mathfrak{G}}_1, \tilde{\mathfrak{g}}_1, \pi_{\tilde{\mathfrak{g}}_1}), (\mathfrak{G}_2, \{\mathfrak{g}_{2,i}, \pi_{\mathfrak{g}_{2,i}}\}_{i\in[k]}), \\ (\tilde{\mathfrak{G}}_2, \tilde{\mathfrak{g}}_2, \pi_{\tilde{\mathfrak{g}}_2}), \mathfrak{G}_T, q, e, \{\mathfrak{r}_{1,i}, \pi_{\mathfrak{r}_{1,i}}, \mathfrak{r}_{2,i}, \pi_{\mathfrak{r}_{2,i}}\}_{i\in[p(\lambda)]} \end{cases}$ *and* $\mathsf{td} = \mathsf{sk}$.

A.1 Proof of Theorem 5

Proof. As the verification algorithms $\mathsf{Ver}_{\mathfrak{G}_1}$ and $\mathsf{Ver}_{\mathfrak{G}_2}$ of $\mathcal{BG}^{k,\rho}_{\mathcal{P},\mathsf{IPE}}$ are induced by the Groth-Sahai-based proof system for membership in the languages $L_{\mathfrak{G}_1}$ and $L_{\mathfrak{G}_1}$, respectively, we proceed with the following arguments.

Completeness Closure: For any group $\mathfrak{G}$ in $\{\mathfrak{G}_1, \mathfrak{G}_2\}$, any elements $\mathfrak{g}_1, \ldots \mathfrak{g}_n \in \mathfrak{G}$ that are additionally given by $\mathcal{BG}$ along with respective proofs $\pi_{\mathfrak{g}_1}, \ldots, \pi_{\mathfrak{g}_n}$:

- By construction, we have $\pi_{\mathfrak{g}_i} = \mathsf{Prove}(\mathsf{crs}, \mathfrak{g}_i, \mathfrak{w})$. By the perfect completeness of the Groth-Sahai proof system, if $\mathsf{Ver}(\mathsf{crs}, \mathfrak{g}_i, \pi_{\mathfrak{g}_i}) = 1$ then it holds that $\mathsf{Ver}_{\mathfrak{G}}(\mathfrak{g}_i) = 1$ for all i.
- For any scalars $\alpha_1, \ldots, \alpha_n$, the element $\mathfrak{g} = \prod_{i\in[n]} \mathfrak{g}_i^{\alpha_i}$ has a proof $\pi_{\mathfrak{g}}$ such that they are induced by $(\mathfrak{g}, \pi_{\mathfrak{g}}) \leftarrow \mathsf{Deriv}(\mathsf{crs}, \{\alpha_i, \mathfrak{g}_i, \pi_{\mathfrak{g}_i}\}_{i\in[n]})$. By Remark 9, it holds that $\mathsf{Ver}(\mathsf{crs}, \mathfrak{g}, \pi_{\mathfrak{g}}) = 1$. Therefore, $\mathsf{Ver}_{\mathfrak{G}}(\mathfrak{g}) = 1$.

Perfect Soundness: The perfect soundness is implied by that of the Groth-Sahai proof system in its commit-and-prove mode (Remark 10).

□

B Deferred Proofs for Hardness of Assumptions

B.1 Proof of Theorem 6

Proof. We proceed with the proof by starting from the distribution $\mathcal{D}_0$. In this distribution, each group element can be written as a power of the corresponding group generator, where the exponent is obtained by evaluating a function of a collection of randomly sampled scalars $\boldsymbol{x} = \{\{z_{1,h}\}_{h\in[k]}, \{z_{2,l}\}_{l\in[k]}, \{z_{3,h}\}_{h\in[k]}, \alpha, r, t, d, e, \{a_l\}_{l\in[k]}, \{b_h\}_{h\in[k]}\}$.

As all scalars in $\boldsymbol{x}$ are independently and uniformly random over $\mathbb{Z}_q^*$, we associate each scalar in $\boldsymbol{x}$ with a formal variable and denote the resulting set of variables by $\boldsymbol{\mathcal{X}} = \{\{\mathcal{X}_{z_{1,h}}\}_{h\in[k]}, \{\mathcal{X}_{z_{2,l}}\}_{l\in[k]}, \{\mathcal{X}_{z_{3,h}}\}_{h\in[k]}, \mathcal{X}_\alpha, \mathcal{X}_r, \mathcal{X}_t, \mathcal{X}_d, \mathcal{X}_e, \{\mathcal{X}_{a_l}\}_{l\in[k]}, \{\mathcal{X}_{b_h}\}_{h\in[k]}\}$.

Parsing $\boldsymbol{\mathcal{X}} = \{\mathcal{X}_1, \ldots, \mathcal{X}_t\}$, we now work over the Laurent polynomial ring

$$\mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}] := \mathbb{Z}_q[\mathcal{X}_1^{\pm 1}, \ldots, \mathcal{X}_t^{\pm 1}] = \left\{ \sum_{j=1}^{n} c_j \prod_{i=1}^{t} \mathcal{X}_i^{e_{i,j}} \;\middle|\; c_j \in \mathbb{Z}_q,\ e_{i,j} \in \mathbb{Z},\ n \in \mathbb{N} \right\}.$$

In this ring, two non-zero monomials $M(\boldsymbol{\mathcal{X}})$ and $N(\boldsymbol{\mathcal{X}})$ are equivalent and denoted by $M(\boldsymbol{\mathcal{X}}) \equiv N(\boldsymbol{\mathcal{X}})$ if there exists $a \in \mathbb{Z}_q$ such that $\alpha M(\boldsymbol{\mathcal{X}}) = N(\boldsymbol{\mathcal{X}})$ in $\mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$; otherwise, $M(\boldsymbol{\mathcal{X}})$ and $N(\boldsymbol{\mathcal{X}})$ are non-equivalent and denoted by $M(\boldsymbol{\mathcal{X}}) \not\equiv N(\boldsymbol{\mathcal{X}})$. Furthermore, for a formal monomial $M = \prod_{j=1}^{t} \mathcal{X}_j^{e_j} \in \mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$, and a variable $\mathcal{X}_i \in \boldsymbol{\mathcal{X}}$, we define $\deg_{\mathcal{X}_i}(M) := e_i$.

Each non-identity group element sampled from $\mathcal{D}_0$ is represented by a unique monomial in $\mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$ corresponding to its exponent. We explicitly list them as follows:

- The generators $\bar{g}_1$ and $\bar{g}_2$ give $[1]_1$ and $[1]_2$, which are represented by $P_{\bar{g}_1}(\boldsymbol{\mathcal{X}}) := 1$ and $Q_{\bar{g}_2}(\boldsymbol{\mathcal{X}}) := 1$, respectively.
- Since $\boldsymbol{D}_1$ is the diagonal matrix with diagonal $\boldsymbol{z}_1$, $[\boldsymbol{D}_1]_1$ gives $[z_{1,h}]_1$ for $h \in [k]$, which are represented by $P_{D_{1,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{1,h}}$.
- Since $\boldsymbol{D}_2$ is the diagonal matrix with diagonal $\boldsymbol{z}_2$, $[\boldsymbol{D}_2]_2$ gives $[z_{2,l}]_2$ for $l \in [k]$, which are represented by $Q_{D_{2,l}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{2,l}}$.
- Since $\boldsymbol{D}_3$ is the diagonal matrix with diagonal $\boldsymbol{z}_3$, $[\boldsymbol{D}_3]_1$ gives $[z_{3,h}]_1$ for $h \in [k]$, which are represented by $P_{D_{3,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{3,h}}$.
- Since $\boldsymbol{D}_1^{-1}$ is the inverse of $\boldsymbol{D}_1$, $[\boldsymbol{D}_1^{-1}]_2$ gives $[z_{1,h}^{-1}]_2$ for $h \in [k]$, which are represented by $Q_{D_{1,h}^{-1}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{1,h}}^{-1}$.
- Since $\boldsymbol{D}_2^{-1}$ is the inverse of $\boldsymbol{D}_2$, $[\boldsymbol{D}_2^{-1}]_1$ gives $[z_{2,l}^{-1}]_1$ for $l \in [k]$, which are represented by $P_{D_{2,l}^{-1}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{2,l}}^{-1}$

– Since $\boldsymbol{D}_3^{-1}$ is the inverse of $\boldsymbol{D}_3$, $[\boldsymbol{D}_3^{-1}]_2$ gives $[z_{3,h}^{-1}]_2$ for $h \in [k]$, which are represented by $Q_{D_{3,h}^{-1}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_{z_{3,h}^{-1}}$.
– $[\boldsymbol{u}_j]_1$ gives $[\alpha^j r a_l]_1$ for $l \in [k]$, which are represented by $P_{u_{j,l}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_\alpha^j \mathcal{X}_r \mathcal{X}_{a_l}$.
– $[\boldsymbol{v}_j]_2$ gives $[\alpha^j t b_h]_2$ for $h \in [k]$, which are represented by $Q_{v_{j,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_\alpha^j \mathcal{X}_t \mathcal{X}_{b_h}$.
– $[\boldsymbol{U}_j]_1$ gives $[\alpha^j r t a_l b_h z_{1,h}]_1$ for $l, h \in [k]$, which are represented by $P_{U_{j,l,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_\alpha^j \mathcal{X}_r \mathcal{X}_t \mathcal{X}_{a_l} \mathcal{X}_{b_h} \mathcal{X}_{z_{1,h}}$.
– $[\boldsymbol{V}_j]_2$ gives $[\alpha^j r t z_{2,l} a_l b_h]_2$ for $l, h \in [k]$, which are represented by $Q_{V_{j,l,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_\alpha^j \mathcal{X}_r \mathcal{X}_t \mathcal{X}_{z_{2,l}} \mathcal{X}_{a_l} \mathcal{X}_{b_h}$.
– $[\boldsymbol{g}]_1$ gives $[d a_l]_1$ for $l \in [k]$, which are represented by $P_{g_l}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_d \mathcal{X}_{a_l}$.
– $[\boldsymbol{h}]_2$ gives $[e b_h]_2$ for $h \in [k]$, which are represented by $Q_{h_h}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_e \mathcal{X}_{b_h}$.
– $[\boldsymbol{G}]_1$ gives $[dt a_l b_h z_{1,h}]_1$ for $l, h \in [k]$, which are represented by $P_{G_{l,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_d \mathcal{X}_t \mathcal{X}_{a_l} \mathcal{X}_{b_h} \mathcal{X}_{z_{1,h}}$.
– $[\boldsymbol{H}]_2$ gives $[e r z_{2,l} a_l b_h]_2$ for $l, h \in [k]$, which are represented by $Q_{H_{l,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_e \mathcal{X}_r \mathcal{X}_{z_{2,l}} \mathcal{X}_{a_l} \mathcal{X}_{b_h}$.
– $[\boldsymbol{K}]_1$ gives $[r t a_l b_h z_{3,h}]_1$ for $l, h \in [k]$, which are represented by $P_{K_{l,h}}(\boldsymbol{\mathcal{X}}) := \mathcal{X}_r \mathcal{X}_t \mathcal{X}_{a_l} \mathcal{X}_{b_h} \mathcal{X}_{z_{3,h}}$.

Then the output D of $\mathcal{D}_0$ are represented by the following monomial sets

$$\begin{aligned}\mathcal{M}_{\mathbb{G}_1} :=& \{P_{\bar{g}_1}, \{P_{D_{1,h}}, P_{D_{3,h}}\}_{h\in[k]}, \{P_{D_{2,l}^{-1}}\}_{l\in[k]}, \{P_{g_l}, \{P_{G_{l,h}}\}_{h\in[k]}\}_{l\in[k]}, \{P_{u_{j,l}}, \{P_{U_{j,l,h}}\}_{h\in[k]}\}_{j\in[m],l\in[k]}, \\ & \{P_{K_{l,h}}\}_{l,h\in[k]}\} \\ \mathcal{M}_{\mathbb{G}_2} :=& \{Q_{\bar{g}_2}, \{Q_{D_{1,h}^{-1}}, Q_{D_{3,h}^{-1}}\}_{h\in[k]}, \{P_{D_{2,l}}\}_{l\in[k]}, \{Q_{h_l}, \{Q_{H_{l,h}}\}_{h\in[k]}\}_{l\in[k]}, \{Q_{v_{j,l}}, \{Q_{V_{j,l,h}}\}_{h\in[k]}\}_{j\in[m],l\in[k]}\}\end{aligned}$$

We also define $\deg_{\mathcal{X}}(\mathbb{G}_1) := \{\deg_{\mathcal{X}}(M) \mid M \in \mathcal{M}_{\mathbb{G}_1}\}$ as the set of all possible degrees of $\mathcal{X}$ that can occur among $\mathcal{M}_{\mathbb{G}_1}$ monomials. Similarly, we define $\deg_{\mathcal{X}}(\mathbb{G}_2) := \{\deg_{\mathcal{X}}(M) \mid M \in \mathcal{M}_{\mathbb{G}_2}\}$.

To show that $[\boldsymbol{G}]_1$, $[\boldsymbol{H}]_2$ and $[\boldsymbol{K}]_1$ in $\mathcal{D}_0$ are indistinguishable from random $[\boldsymbol{X}]_1$, $[\boldsymbol{Y}]_2$ and $[\boldsymbol{Z}]$ in $\mathcal{D}_1$, respectively, we first show that the monomials corresponding to $[\boldsymbol{G}]_1$, $[\boldsymbol{H}]_2$ and $[\boldsymbol{K}]_1$ do not satisfy any non-trivial pairing equation with other monomials via the following Lemma 3, Lemma 4 and Lemma 5. By later invoking Lemma 6 as a master lemma on the indistinguishability of independent monomials from random elements, we then conclude that $[\boldsymbol{G}]_1$, $[\boldsymbol{H}]_2$, and $[\boldsymbol{K}]_1$ are indeed indistinguishable from random.

Lemma 3 (Independence of monomials $P_{G_{l,h}}$). *There do not exist any monomials $\hat{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ and $\tilde{Q}, \bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ such that*

$$P_{G_{l,h}}(\boldsymbol{\mathcal{X}})\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d \mathcal{X}_t \mathcal{X}_{a_l} \mathcal{X}_{b_h} \mathcal{X}_{z_{1,h}} \tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \hat{P}(\boldsymbol{\mathcal{X}})\bar{Q}(\boldsymbol{\mathcal{X}}) \text{ and } \hat{P} \not\equiv P_{G_{l,h}}. \tag{B.1}$$

Proof. We suppose for a contradiction that such monomials exist.
Since $\deg_{\mathcal{X}_{z_{1,h}}}(P_{G_{l,h}}) = 1$ and $\deg_{\mathcal{X}_{z_{1,h}}}(\mathbb{G}_2) = \{-1, 0\}$, then $\deg_{\mathcal{X}_{z_{1,h}}}(P_{G_{l,h}}\tilde{Q}) \in \{0, 1\}$. We consider the following cases:

– $\deg_{\mathcal{X}_{z_{1,h}}}(P_{G_{l,h}}\tilde{Q}) = 1$: Since $\deg_{\mathcal{X}_{z_{1,h}}}(\mathbb{G}_1) = \{0, 1\}$, this case implies that $\deg_{\mathcal{X}_{z_{1,h}}}(\hat{P}) = 1$ and $\deg_{\mathcal{X}_{z_{1,h}}}(\bar{Q}) = 0$. Then $\hat{P} \equiv P_{U_{\hat{j},\hat{l},h}}$ or $\hat{P} \equiv P_{G_{\hat{l},h}}$.
  • If $\hat{P} \equiv P_{U_{\hat{j},\hat{l},h}}$: Equation (B.1) becomes

$$\mathcal{X}_d \mathcal{X}_t \mathcal{X}_{a_l} \mathcal{X}_{b_h} \mathcal{X}_{z_{1,h}} \tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_\alpha^{\hat{j}} \mathcal{X}_r \mathcal{X}_t \mathcal{X}_{a_{\hat{l}}} \mathcal{X}_{b_h} \mathcal{X}_{z_{1,h}} \bar{Q}(\boldsymbol{\mathcal{X}}) \implies \mathcal{X}_d \mathcal{X}_{a_l} \tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_\alpha^{\hat{j}} \mathcal{X}_r \mathcal{X}_{a_{\hat{l}}} \bar{Q}(\boldsymbol{\mathcal{X}})$$

Since $\deg_{\mathcal{X}_d}(\mathbb{G}_2) = \{0\}$, then the degree of $\mathcal{X}_d$ in RHS is always 0, hence contradiction.

- If $\hat{P} = P_{G_{\hat{l},h}}$: Equation (B.1) becomes

$$\mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h}\mathcal{X}_{z_{1,h}}\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_h}\mathcal{X}_{z_{1,h}}\bar{Q}(\boldsymbol{\mathcal{X}}) \implies \mathcal{X}_{a_l}\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \tag{B.2}$$

  Since $P_{G_{\hat{l},h}} \not\equiv P_{G_{l,h}}$, we must have $\hat{l} \neq l$. This means $\deg_{\mathcal{X}_{a_{\hat{l}}}}(\tilde{Q}) = 1$, so $\tilde{Q} \equiv Q_{V_{\tilde{j},\hat{l},\tilde{h}}}$ or $\tilde{Q} \equiv Q_{H_{\hat{l},\tilde{h}}}$.
  - If $\tilde{Q} \equiv Q_{V_{\tilde{j},\hat{l},\tilde{h}}}$: Equation (B.2) becomes

$$\mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_l}\mathcal{X}_{\alpha}^{\tilde{j}}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{\alpha}^{\tilde{j}}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}}$$

    There is no monomial $\bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ that contains the term $\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}$ with $\hat{l} \neq l$, hence contradiction.
  - If $\tilde{Q} \equiv Q_{H_{\hat{l},\tilde{h}}}$: Equation (B.2) becomes

$$\mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_l}\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}}$$

    There is no monomial $\bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ that contains the term $\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}$ with $\hat{l} \neq l$, hence contradiction.

– $\deg_{\mathcal{X}_{z_{1,h}}}(P_{G_{l,h}}\tilde{Q}) = 0$: From Equation (B.1), this case implies $\deg_{\mathcal{X}_{z_{1,h}}}(\tilde{Q}) = -1$. This means $\tilde{Q} \equiv Q_{D_{1,h}^{-1}}$ and

$$\hat{P}(\boldsymbol{\mathcal{X}})\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h} \tag{B.3}$$

  Since $\deg_{\mathcal{X}_d}(\mathbb{G}_2) = \{0\}$, then $\deg_{\mathcal{X}_d}(\hat{P}) = 1$. This means $\hat{P} \equiv P_{g_{\hat{l}}}$ or $\hat{P} \equiv P_{G_{\hat{l},\hat{h}}}$.
  - If $\hat{P} \equiv P_{g_{\hat{l}}}$: Equation (B.3) becomes

$$\mathcal{X}_d\mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}^{-1}\mathcal{X}_{a_l}\mathcal{X}_{b_h} \tag{B.4}$$

    Since $\deg_{\mathcal{X}_{a_{\hat{l}}}}(\mathbb{G}_2) = \{0,1\}$, we must have $\hat{l} = l$. This means $\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_t\mathcal{X}_{b_h}$, which is not equal to any monomial in $\mathcal{M}_{\mathbb{G}_2}$, hence contradiction.
  - If $\hat{P} \equiv P_{G_{\hat{l},\hat{h}}}$: Equation (B.3) becomes

$$\mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\hat{h}}}\mathcal{X}_{z_{1,\hat{h}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\hat{h}}}\mathcal{X}_{z_{1,\hat{h}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h}$$

    Since $\deg_{\mathcal{X}_{z_{1,\hat{h}}}}(\mathbb{G}_2) = \{0,-1\}$, then $\bar{Q} \equiv Q_{D_{1,\hat{h}}^{-1}}$. The last equation becomes

$$\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\hat{h}}}\mathcal{X}_{z_{1,\hat{h}}}\mathcal{X}_{z_{1,\hat{h}}}^{-1} \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\hat{h}}} \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h}.$$

    This means $\hat{l} = l$ and $\hat{h} = h$, which contradicts the assumption $\hat{P} \not\equiv P_{G_{l,h}}$ in Equation (B.1).

Since all possible cases lead to a contradiction, the proof is complete. □

Lemma 4 (Independence of monomials $Q_{H_{l,h}}$). *There do not exist any monomials $\hat{P}, \bar{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ and $\tilde{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ such that*

$$\hat{P}(\boldsymbol{\mathcal{X}})Q_{H_{l,h}}(\boldsymbol{\mathcal{X}}) \equiv \hat{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,l}}\mathcal{X}_{a_l}\mathcal{X}_{b_h} \equiv \bar{P}(\boldsymbol{\mathcal{X}})\tilde{Q}(\boldsymbol{\mathcal{X}}) \text{ and } \tilde{Q} \not\equiv Q_{H_{l,h}}. \tag{B.5}$$

Proof. We suppose for a contradiction that such monomials exist.
Since $\deg_{\mathcal{X}_{z_{2,l}}}(Q_{H_{l,h}}) = 1$ and $\deg_{\mathcal{X}_{z_{2,l}}}(\mathbb{G}_1) = \{-1, 0\}$, then $\deg_{\mathcal{X}_{z_{2,l}}}(\hat{P}Q_{H_{l,h}}) \in \{0, 1\}$. We consider the following cases:

– $\deg_{\mathcal{X}_{z_{2,l}}}(\hat{P}Q_{H_{l,h}}) = 1$: Since $\deg_{\mathcal{X}_{z_{2,l}}}(\mathbb{G}_2) = \{0, 1\}$, this case implies that $\deg_{\mathcal{X}_{z_{2,l}}}(\tilde{Q}) = 1$ and $\deg_{\mathcal{X}_{z_{2,l}}}(\bar{P}) = 0$. Then $\tilde{Q} \equiv Q_{V_{\tilde{j},l,\tilde{h}}}$ or $\tilde{Q} \equiv Q_{H_{l,\tilde{h}}}$.
- If $\tilde{Q} \equiv Q_{V_{\tilde{j},l,\tilde{h}}}$: Equation (B.5) becomes

$$\hat{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,l}}\mathcal{X}_{a_l}\mathcal{X}_{b_h} \equiv \bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_\alpha^{\tilde{j}}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{z_{2,l}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}} \implies \hat{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_{b_h} \equiv \bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_\alpha^{\tilde{j}}\mathcal{X}_t\mathcal{X}_{b_{\tilde{h}}}$$

  Since $\deg_{\mathcal{X}_e}(\mathbb{G}_1) = \{0\}$, then the degree of $\mathcal{X}_e$ in RHS is always 0, hence contradiction.
- If $\tilde{Q} \equiv Q_{H_{l,\tilde{h}}}$: Equation (B.5) becomes

$$\hat{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,l}}\mathcal{X}_{a_l}\mathcal{X}_{b_h} \equiv \bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,l}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}} \implies \hat{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{b_h} \equiv \bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{b_{\tilde{h}}} \tag{B.6}$$

  Since $Q_{H_{l,\tilde{h}}} \not\equiv Q_{H_{l,h}}$, then we must have $\tilde{h} \neq h$. This means $\deg_{\mathcal{X}_{b_{\tilde{h}}}}(\hat{P}) = 1$, so $\hat{P} \equiv P_{U_{\hat{j},\hat{l},\tilde{h}}}$, $\hat{P} \equiv P_{G_{\hat{l},\tilde{h}}}$, or $\hat{P} \equiv P_{K_{\hat{l},\tilde{h}}}$.
  * If $\hat{P} \equiv P_{U_{\hat{j},\hat{l},\tilde{h}}}$: Equation (B.6) becomes

$$\bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_\alpha^{\hat{j}}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}}\mathcal{X}_{z_{1,\tilde{h}}}\mathcal{X}_{b_h} \implies \bar{P}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_\alpha^{\hat{j}}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_h}\mathcal{X}_{z_{1,\tilde{h}}}$$

    There is no monomial $\bar{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ that contains the term $\mathcal{X}_{b_h}\mathcal{X}_{z_{1,\tilde{h}}}$ with $\tilde{h} \neq h$, hence contradiction.
  * If $\hat{P} \equiv P_{G_{\hat{l},\tilde{h}}}$: Equation (B.6) becomes

$$\bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}}\mathcal{X}_{z_{1,\tilde{h}}}\mathcal{X}_{b_h} \implies \bar{P}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_d\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_h}\mathcal{X}_{z_{1,\tilde{h}}}$$

    There is no monomial $\bar{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ that contains the term $\mathcal{X}_{b_h}\mathcal{X}_{z_{1,\tilde{h}}}$ with $\tilde{h} \neq h$, hence contradiction.
  * If $\hat{P} \equiv P_{K_{\hat{l},\tilde{h}}}$: Equation (B.6) becomes

$$\bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}}\mathcal{X}_{z_{3,\tilde{h}}}\mathcal{X}_{b_h} \implies \bar{P}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_h}\mathcal{X}_{z_{3,\tilde{h}}}$$

    There is no monomial $\bar{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ that contains the term $\mathcal{X}_{b_h}\mathcal{X}_{z_{3,\tilde{h}}}$ with $\tilde{h} \neq h$, hence contradiction.

– $\deg_{\mathcal{X}_{z_{2,l}}}(\hat{P}Q_{H_{l,h}}) = 0$: From Equation (B.5), this case implies $\deg_{\mathcal{X}_{z_{2,l}}}(\hat{P}) = -1$. This means $\hat{P} \equiv P_{D_{2,l}^{-1}}$ and

$$\bar{P}(\boldsymbol{\mathcal{X}})\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_e\mathcal{X}_r\mathcal{X}_{a_l}\mathcal{X}_{b_h} \tag{B.7}$$

  Since $\deg_{\mathcal{X}_e}(\mathbb{G}_1) = \{0\}$, then $\deg_{\mathcal{X}_e}(\tilde{Q}) = 1$. This means $\tilde{Q} \equiv \tilde{Q}_{h_{\tilde{h}}}$ or $\tilde{Q} \equiv \tilde{Q}_{H_{\tilde{l},\tilde{h}}}$.
- If $\tilde{Q} \equiv \tilde{Q}_{h_{\tilde{h}}}$: Equation (B.7) becomes

$$\bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_e\mathcal{X}_r\mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \bar{P}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_{a_l}\mathcal{X}_{b_h}\mathcal{X}_{b_{\tilde{h}}}^{-1}$$

  Since $\deg_{\mathcal{X}_{b_{\tilde{h}}}}(\mathbb{G}_1) = \{0, 1\}$, we must have $\tilde{h} = h$. This means $\bar{P}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_{a_l}$, which is not equal to any monomial in $\mathcal{M}_{\mathbb{G}_1}$, hence contradiction.

- If $\tilde{Q} \equiv \tilde{Q}_{H_{\tilde{l},\tilde{h}}}$: Equation (B.7) becomes

$$\bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,\tilde{l}}}\mathcal{X}_{a_{\tilde{l}}}\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_e\mathcal{X}_r\mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \bar{P}(\boldsymbol{\mathcal{X}})\mathcal{X}_{z_{2,\tilde{l}}}\mathcal{X}_{a_{\tilde{l}}}\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h}$$

  Since $\deg_{\mathcal{X}_{z_{2,\tilde{l}}}}(\mathbb{G}_1) = \{0, -1\}$, then $\bar{P} \equiv P_{D^{-1}_{2,\tilde{l}}}$. The last equation becomes

$$\mathcal{X}^{-1}_{z_{2,\tilde{l}}}\mathcal{X}_{z_{2,\tilde{l}}}\mathcal{X}_{a_{\tilde{l}}}\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \mathcal{X}_{a_{\tilde{l}}}\mathcal{X}_{b_{\tilde{h}}} \equiv \mathcal{X}_{a_l}\mathcal{X}_{b_h}$$

  This means $\tilde{l} = l$ and $\tilde{h} = h$, which contradicts the assumption $\tilde{Q} \not\equiv Q_{H_{l,h}}$ in Equation (B.5).

Since all possible cases lead to a contradiction, the proof is complete. □

Lemma 5 (Independence of monomials $P_{K_{l,h}}$). *There do not exist monomials $\hat{P}$ in $\mathcal{M}_{\mathbb{G}_1}$ and $\tilde{Q}, \bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ such that*

$$P_{K_{l,h}}(\boldsymbol{\mathcal{X}})\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h}\mathcal{X}_{z_{3,h}}\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \hat{P}(\boldsymbol{\mathcal{X}})\bar{Q}(\boldsymbol{\mathcal{X}}) \text{ and } \hat{P} \not\equiv P_{K_{l,h}}. \tag{B.8}$$

Proof. We suppose for a contradiction that such monomials exist.
Since $\deg_{\mathcal{X}_{z_{3,h}}}(P_{K_{l,h}}) = 1$ and $\deg_{\mathcal{X}_{z_{3,h}}}(\mathbb{G}_2) = \{-1, 0\}$, then $\deg_{\mathcal{X}_{z_{3,h}}}(P_{K_{l,h}}\tilde{Q}) \in \{0, 1\}$. We consider the following cases:

– If $\deg_{\mathcal{X}_{z_{3,h}}}(P_{K_{l,h}}\tilde{Q}) = 1$: Since $\deg_{\mathcal{X}_{z_{3,h}}}(\mathbb{G}_1) = \{0, 1\}$, this case implies that $\deg_{\mathcal{X}_{z_{3,h}}}(\hat{P}) = 1$ and $\deg_{\mathcal{X}_{z_{3,h}}}(\bar{Q}) = 0$. Then $\hat{P} = P_{K_{\hat{l},h}}$ with $\hat{l} \neq l$. Consequently, Equation (B.8) becomes

$$\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h}\mathcal{X}_{z_{3,h}}\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_h}\mathcal{X}_{z_{3,h}}\bar{Q}(\boldsymbol{\mathcal{X}}) \implies \mathcal{X}_{a_l}\tilde{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \tag{B.9}$$

  This means $\deg_{\mathcal{X}_{a_{\hat{l}}}}(\tilde{Q}) = 1$, so $\tilde{Q} \equiv Q_{V_{\tilde{j},\hat{l},\tilde{h}}}$ or $\tilde{Q} \equiv Q_{H_{\hat{l},\tilde{h}}}$.
  - If $\tilde{Q} \equiv Q_{V_{\tilde{j},\hat{l},\tilde{h}}}$: Equation (B.9) becomes

$$\mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_l}\mathcal{X}^{\tilde{j}}_{\alpha}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}^{\tilde{j}}_{\alpha}\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}}$$

    There is no monomial $\bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ that contains the term $\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}$ with $\hat{l} \neq l$, hence contradiction.
  - If $\tilde{Q} \equiv Q_{H_{\tilde{l},\tilde{h}}}$: Equation (B.9) becomes

$$\mathcal{X}_{a_{\hat{l}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{a_l}\mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_{\hat{l}}}\mathcal{X}_{b_{\tilde{h}}} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_e\mathcal{X}_r\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}\mathcal{X}_{b_{\tilde{h}}}$$

    There is no monomial $\bar{Q}$ in $\mathcal{M}_{\mathbb{G}_2}$ that contains the term $\mathcal{X}_{z_{2,\hat{l}}}\mathcal{X}_{a_l}$ with $\hat{l} \neq l$, hence contradiction.

– If $\deg_{\mathcal{X}_{z_{3,h}}}(P_{K_{l,h}}\tilde{Q}) = 0$: From Equation (B.8), this case implies $\deg_{\mathcal{X}_{z_{3,h}}}(\tilde{Q}) = -1$. This means $\tilde{Q} \equiv Q_{D^{-1}_{3,h}}$ and

$$\hat{P}(\boldsymbol{\mathcal{X}})\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h} \tag{B.10}$$

  This means

$$\deg_{\mathcal{X}_\alpha}(\hat{P}\bar{Q}) = \deg_{\mathcal{X}_d}(\hat{P}\bar{Q}) = \deg_{\mathcal{X}_e}(\hat{P}\bar{Q}) = 0.$$

  On the other hand, the degrees of variables $\mathcal{X}_\alpha$, $\mathcal{X}_d$ and $\mathcal{X}_e$ in any monomial are all non-negative. Therefore, $\deg_{\mathcal{X}_\alpha}(\hat{P}) = \deg_{\mathcal{X}_\alpha}(\bar{Q}) = 0$ and $\deg_{\mathcal{X}_d}(\hat{P}) = \deg_{\mathcal{X}_d}(\bar{Q}) = \deg_{\mathcal{X}_e}(\hat{P}) = \deg_{\mathcal{X}_e}(\bar{Q}) = 0$.

Furthermore, since $\deg_{\mathcal{X}_{a_l}}(\hat{P}\bar{Q}) = 1$, the only possible case is that $\hat{P} \equiv P_{K_{l,\hat{h}}}$. Equation (B.10) becomes

$$\mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_{\hat{h}}}\bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_r\mathcal{X}_t\mathcal{X}_{a_l}\mathcal{X}_{b_h} \implies \bar{Q}(\boldsymbol{\mathcal{X}}) \equiv \mathcal{X}_{b_{\hat{h}}}^{-1}\mathcal{X}_{b_h}$$

This means $\hat{h} = h$, which contradicts the assumption $\hat{P} \not\equiv P_{K_{l,h}}$ in Equation (B.8).

Since all possible cases lead to a contradiction, the proof is complete. □

Following the framework of Boyen [Boy08], which formalizes the indistinguishability of elements in bilinear groups via generic group arguments, we invoke the following lemma on monomial exponents. The statement captures the intuition that, in a prime-order group, elements whose exponents correspond to monomials that do not satisfy any non-trivial pairing equation with other monomials are computationally indistinguishable from random.

Lemma 6 (Indistinguishability from independent-monomial exponents). *Let $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \bar{g}_1, \bar{g}_2, \hat{e})$ be a prime-order group. Let $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \ldots, \mathcal{X}_t)$ be a tuple of independent variables and $\mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$ be the ring of Laurent polynomials in $\boldsymbol{\mathcal{X}}$ with coefficients in $\mathbb{Z}_q$. Let $m, n \in \mathbb{N}$ and sample $\boldsymbol{x} = (x_1, \ldots x_t) \leftarrow\!\!\$\, \mathbb{Z}_q^{*t}$. We construct the following group elements:*

- *For each $i \in [m]$, let $p_i := [P_i(\boldsymbol{x})]_1 \in \mathbb{G}_1$, where $P_i \in \mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$ are pairwise non-equivalent and non-zero monomials.*
- *For each $j \in [n]$, let $q_j := [Q_j(\boldsymbol{x})]_2 \in \mathbb{G}_2$, where $Q_j \in \mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$ are pairwise non-equivalent and non-zero monomials.*

Then for any PPT adversary $\mathcal{A}$ with generic oracle access to the group and given the elements $\{(F_i, f_i)\}_{i\in[m]}$ and $\{(H_j, h_j)\}_{j\in[n]}$, the following holds except with negligible probability:

- *For any $i^* \in [m]$, the element f_{i^*} is indistinguishable from a uniformly random element of $\mathbb{G}_1$ to $\mathcal{A}$ if and only if $F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}}) \not\equiv F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})$ for all $k \neq i^*$ and all j, l.*
- *For any $j^* \in [n]$, the element h_{j^*} is indistinguishable from a uniformly random element of $\mathbb{G}_2$ to $\mathcal{A}$ if and only if $F_i(\boldsymbol{\mathcal{X}}) \cdot H_{j^*}(\boldsymbol{\mathcal{X}}) \not\equiv F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})$ for all $l \neq j^*$ and all i, k.*

Proof. We prove the first statement. The second statement follows by symmetric arguments with the roles of $\mathbb{G}_1$ and $\mathbb{G}_2$ exchanged.

We prove the necessary condition by assuming that f_{i^*} is indistinguishable from a random element of $\mathbb{G}_1$. Suppose that there exist $i \in [m]$, $j, l \in [n]$ and $\alpha \in \mathbb{Z}_q$ such that $k \neq i^*$ and $\alpha F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}}) = F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})$. Then it holds true that

$$\alpha[F_{i^*}(\boldsymbol{x})]_1 \cdot [H_j(\boldsymbol{x})]_2 = [F_k(\boldsymbol{x})]_1 \cdot [H_l(\boldsymbol{x})]_2$$

for any $\boldsymbol{x} \in \mathbb{Z}_q^{*t}$. This means the adversary $\mathcal{A}$ can simply test if $\hat{e}(f_{i^*}, h_j)^{\alpha} \equiv \hat{e}(f_k, h_l)$. If yes, then f_{i^*} is not a random element, hence contradiction.

We prove the sufficient condition by assuming that $F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}}) \not\equiv F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})$ for all $k \in [m] \setminus i^*$ and $j, l \in [n]$. Since $F_{i^*}(\boldsymbol{\mathcal{X}})$ is non-zero and $\{H_j(\boldsymbol{\mathcal{X}})\}_{j\in[n]}$ are pairwise non-equivalent and non-zero monomials, we have $\{F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}})\}_{j\in[n]}$ are also pairwise non-equivalent and non-zero monomials.

Let $\mathcal{S}_{i^*}$ be a maximal subset taken from $\{F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})\}_{k\in[m]\setminus i^*, l\in[n]}$ such that all monomials in $\mathcal{S}_{i^*}$ are pairwise non-equivalent. This also means that $\mathcal{S}_{i^*}$ contains all monomials in $\{F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})\}_{k\in[m]\setminus i^*, l\in[n]}$ up to scalar equivalence. Consequently, we have the set

$$\mathcal{T}_{i^*} := \{F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}})\}_{j\in[n]} \cup \mathcal{S}_{i^*}$$

contains all monomials in $\{F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}})\}_{j\in[n]} \cup \{F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}})\}_{k\in[m]\setminus i^*, l\in[n]}$ up to scalar equivalence and all monomials in $\mathcal{T}_{i^*}$ are pairwise non-equivalent and non-zero. Since non-equivalent monomials form a basis of $\mathbb{Z}_q[\boldsymbol{\mathcal{X}}^{\pm}]$ as a $\mathbb{Z}_q$-module, then elements in $\mathcal{T}_{i^*}$ are also linearly independent.

On the other hand, in the generic group model, the only way that $\mathcal{A}$ can distinguish f_{i^*} from random group elements is to find a pairing-induced relation of the following form:

$$\prod_j \hat{e}(f_{i^*}, h_j)^{a_j} = \sum_j a_j [F_{i^*}(\boldsymbol{x}) \cdot H_j(\boldsymbol{x})]_T = \sum_{k\neq i^*} \sum_l b_{k,l} [F_k(\boldsymbol{x}) \cdot H_l(\boldsymbol{x})]_T \tag{B.11}$$

$$= \prod_{k\neq i^*} \prod_l \hat{e}(f_k, h_l)^{b_{k,l}} \tag{B.12}$$

where $\{a_j\}_{j\in[n]}$ are not simultaneously zero. But we already know that

$$\begin{aligned} &\sum_j a_j F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}}) - \sum_{k\neq i^*} \sum_l b_{k,l} F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}}) \\ &= \sum_j a_j F_{i^*}(\boldsymbol{\mathcal{X}}) \cdot H_j(\boldsymbol{\mathcal{X}}) - \sum_{F_k \cdot H_l \in \mathcal{S}_{i^*}} b'_{k,l} F_k(\boldsymbol{\mathcal{X}}) \cdot H_l(\boldsymbol{\mathcal{X}}) \\ &\neq 0 \end{aligned}$$

by the linear independence of $\mathcal{T}_{i^*}$. This means Equation (B.11) does not hold unless

$$\sum_j a_j F_{i^*}(\boldsymbol{x}) \cdot H_j(\boldsymbol{x}) - \sum_{F_k \cdot H_l \in \mathcal{S}_{i^*}} b'_{k,l} F_k(\boldsymbol{x}) \cdot H_l(\boldsymbol{x}) = 0,$$

which happens with only negligible probability over the random choice of $\boldsymbol{x}$. Therefore, the probability that $\mathcal{A}$ can distinguish between f_{i^*} and a random element is negligible.
The lemma proof is complete as both conditions are satisfied. □

Since all monomials in the sets $\mathcal{M}_{\mathbb{G}_1}$ and $\mathcal{M}_{\mathbb{G}_2}$ are pairwise non-equivalent and non-zero, Lemma 6 implies joint indistinguishability from random for any set of group elements represented by independent monomials. Moreover, Lemma 3, Lemma 4, and Lemma 5 ensure that the entries of $[\boldsymbol{G}]_1$, $[\boldsymbol{H}]_2$, and $[\boldsymbol{K}]_1$ are represented by such independent monomials.

Therefore, the tuple $([\boldsymbol{G}]_1, [\boldsymbol{H}]_2, [\boldsymbol{K}]_1)$ is jointly indistinguishable from the random tuple $([\boldsymbol{X}]_1, [\boldsymbol{Y}]_2, [\boldsymbol{Z}]_1)$, and thus $\mathcal{D}_0$ is indistinguishable from $\mathcal{D}_1$, which concludes the proof. □

B.2 Proof of Theorem 7

We first define a parameterized extension of the BDHDE assumption, with the additional parameter $\tau \geq 0$. We prove that hardness for $\tau = 1$, which corresponds to the original (m, k)-BDHDE assumption, implies hardness for arbitrary τ via a reduction, and thus the security established in the generic bilinear group model extends to the full parameterized assumption for arbitrary polynomial-size τ.

Definition 20 ((m,τ,k)-Bilateral Diffie-Hellman Decision Exponent Problem). *Let $\mathcal{P}$ be a prime-order bilinear group generator, $m,\tau \geq 0$ and $k \geq 1$ be integer parameters, we define the distribution $\mathcal{D}_0$ and $\mathcal{D}_1$ as follows:*

$\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(1^\lambda)$

$\alpha, r, t \leftarrow\!\!\$\ \mathbb{Z}_q^*$	$\boldsymbol{a}, \boldsymbol{b} \leftarrow\!\!\$\ \mathbb{Z}_q^{*k}$	
$\boldsymbol{z}_w \leftarrow\!\!\$\ \mathbb{Z}_q^{*k}$	$\boldsymbol{D}_w \leftarrow \mathsf{diag}(\boldsymbol{z}_w)$	$\forall w \in [3]$
$\boldsymbol{K} \leftarrow rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_3$	$\boldsymbol{Z} \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times k}$	
$d_i, e_i \leftarrow\!\!\$\ \mathbb{Z}_q^*$	$\boldsymbol{X}_i, \boldsymbol{Y}_i \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times k}$	$\forall i \in [\tau]$
$\boldsymbol{u}_j \leftarrow \alpha^j r\boldsymbol{a}$	$\boldsymbol{v}_j \leftarrow \alpha^j t\boldsymbol{b}$	$\forall j \in [m]$
$\boldsymbol{U}_j \leftarrow \alpha^j rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1$	$\boldsymbol{V}_j \leftarrow \alpha^j rt\boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b})$	$\forall j \in [m]$
$\boldsymbol{g}_i \leftarrow d_i\boldsymbol{a}$	$\boldsymbol{h}_i \leftarrow e_i\boldsymbol{b}$	$\forall i \in [\tau]$
$\boldsymbol{G}_i \leftarrow d_i t(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1$	$\boldsymbol{H}_i \leftarrow e_i r\boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b})$	$\forall i \in [\tau]$

$\mathcal{D}_0$ *outputs*

$$\mathsf{D} := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{G}_i]_1\}_{i\in[\tau]}, [\boldsymbol{K}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{H}_i]_2\}_{i\in[\tau]} \end{cases}$$

$\mathcal{D}_1$ *outputs*

$$\mathsf{D} := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{X}_i]_1\}_{i\in[\tau]}, [\boldsymbol{Z}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{Y}_i]_2\}_{i\in[\tau]} \end{cases}$$

The bilinear group generator $\mathcal{P}$ is said to satisfy the (m,τ,k)-Bilateral Diffie-Hellman Decision Exponent assumption if given any parameters $m,\tau \geq 0$ and $k \geq 1$, for every PPT adversary $\mathcal{A}$, the advantage defined as follows is bounded by $\mathsf{negl}(\lambda)$:

$$\mathsf{Adv}_{\mathsf{BDHDE},\mathcal{A}}(\lambda, m, \tau, k) := |\Pr[\mathcal{A}(\mathsf{D}) = 1 | \mathsf{D} \leftarrow \mathcal{D}_0] - \Pr[\mathcal{A}(\mathsf{D}) = 1 | \mathsf{D} \leftarrow \mathcal{D}_1]|.$$

As highlighted in blue, the difference between two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ is that in $\mathcal{D}_0$ the challenge elements $\{[\boldsymbol{G}_i]_1\}_{i\in[\tau]}$, $[\boldsymbol{K}]_1$, and $\{[\boldsymbol{H}_i]_2\}_{i\in[\tau]}$ are sampled in correlation with other elements. In contrast, in $\mathcal{D}_1$, the challenge elements $\{[\boldsymbol{X}_i]_1\}_{i\in[\tau]}$, $[\boldsymbol{Z}]_1$, and $\{[\boldsymbol{Y}_i]_2\}_{i\in[\tau]}$ are sampled independently and uniformly random.

Theorem 11 (Hardness of Multi-τ-Bilateral Diffie-Hellman Decision Exponent Assumption). *For any integers $m,\tau \geq 0$ and $k \geq 1$, $\mathcal{P}$ satisfies the (m,τ,k)-Bilateral Diffie-Hellman Decision Exponent assumption in the generic bilinear group model.*

More precisely, for any adversary $\mathcal{A}_{\mathsf{BDHDE}}$ that breaks the (m,τ,k)-Bilateral Diffie-Hellman Decision Exponent assumption, there exists an adversary $\mathcal{B}_{\mathsf{BDHDE}}$ that breaks the (m,k)-Bilateral Diffie-Hellman Decision Exponent assumption (where $\tau = 1$), such that

$$\mathsf{Adv}_{\mathcal{A}_{\mathsf{BDHDE}}}(\lambda, m, \tau, k) \leq \tau \mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda, m, k).$$

Proof. Let $\mathcal{A}$ be an adversary that breaks the (m,τ,k)-Bilateral Diffie-Hellman Decision Exponent assumption and b be the challenge bit of from an instance of this assumption. We proceed with the proof via a hybrid argument, for which we define $\tau+1$ security games $\mathsf{G}_0, \ldots, \mathsf{G}_\tau$, along with modifications between each two games highlighted in gray as follows.

Game G_0: This game corresponds to the (m,τ,k)-BDHDE problem, where $\mathcal{A}$ guesses the challenge bit b from the output D of the distribution $\mathcal{D}_{\mathsf{b}}$. Keeping the notation for group elements as in Definition 20, we rewrite the output D as follows.

$$\mathsf{D} := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{G}_i^{(\mathsf{b})}]_1\}_{i\in[\tau]}, [\boldsymbol{K}^{(\mathsf{b})}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{H}_i^{(\mathsf{b})}]_2\}_{i\in[\tau]} \end{cases}$$

$$\text{where} \begin{cases} \boldsymbol{G}_i^{(0)} = \boldsymbol{G}_i, \boldsymbol{H}_i^{(0)} = \boldsymbol{H}_i, \boldsymbol{K}^{(0)} = \boldsymbol{K}, \\ \boldsymbol{G}_i^{(1)} = \boldsymbol{X}_i, \boldsymbol{H}_i^{(1)} = \boldsymbol{Y}_i, \boldsymbol{K}^{(1)} = \boldsymbol{Z}. \end{cases}$$

Game G_1: This game is similar to G_0, except the output D is modified as follows.

$$\boldsymbol{G}_1^{(\mathsf{b})} = \boldsymbol{X}_1, \quad \boldsymbol{H}_1^{(\mathsf{b})} = \boldsymbol{Y}_1, \quad \boldsymbol{K}^{(\mathsf{b})} = \boldsymbol{Z}.$$

From Lemma 7, the indistinguishability between G_0 and G_1 follows by the (m,k)-BDHDE assumption.

Games $\mathsf{G}_{i^\star}$: For $i^\star \in [2:\tau]$, the game $\mathsf{G}_{i^\star}$ is similar to the game $\mathsf{G}_{i^\star-1}$, except the output D is modified as follows.

$$\boldsymbol{G}_{i^\star}^{(\mathsf{b})} = \boldsymbol{X}_{i^\star}, \quad \boldsymbol{H}_{i^\star}^{(\mathsf{b})} = \boldsymbol{Y}_{i^\star}$$

We note that when compared with G_0, the modification is that

$$\boldsymbol{G}_i^{(\mathsf{b})} = \boldsymbol{X}_i, \boldsymbol{H}_i^{(\mathsf{b})} = \boldsymbol{Y}_i \; \forall i \in [i^\star] \quad \text{and} \quad \boldsymbol{K}^{(\mathsf{b})} = \boldsymbol{Z}.$$

Moreover, in the final game G_τ, the output D is modified as follows

$$\mathsf{D} = \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{X}_i]_1\}_{i\in[\tau]}, [\boldsymbol{Z}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{Y}_i]_2\}_{i\in[\tau]} \end{cases}$$

which contains no information about the challenge bit b. As a consequence, the advantage of $\mathcal{A}$ in G_τ, denoted by $\mathsf{Adv}_{\mathsf{G}_\tau,\mathcal{A}}(\lambda)$, equals 0.
From Lemma 7, the indistinguishability between each $\mathsf{G}_{i^\star-1}$ and $\mathsf{G}_{i^\star}$ follows by the (m,k)-BDHDE assumption.

Taking all transitions from G_0 to G_τ, along with the advantage of $\mathcal{A}$ in each game as $\mathsf{Adv}_{\mathsf{G}_i,\mathcal{A}}(\lambda)$, we obtain

$$\mathsf{Adv}_{\mathcal{A}_{\mathsf{BDHDE}}}(\lambda,m,\tau,k) \leq \sum_{i\in[\tau]} \left|\mathsf{Adv}_{\mathsf{G}_{i-1},\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_i,\mathcal{A}}(\lambda)\right| \leq \tau \mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda,m,k).$$

Therefore, the proof of the theorem is complete. □

Lemma 7 (Indistinguishability between $\mathsf{G}_{i^\star-1}$ and $\mathsf{G}_{i^\star}$ in Theorem 11). *For any adversary $\mathcal{A}$ and any $i^\star \in \{1,\ldots,\tau\}$, let $\mathsf{Adv}_{\mathsf{G}_{i^\star-1},\mathcal{A}}(\lambda)$ and $\mathsf{Adv}_{\mathsf{G}_{i^\star},\mathcal{A}}(\lambda)$ be its advantages in games $\mathsf{G}_{i^\star-1}$ and $\mathsf{G}_{i^\star}$ of Theorem 11 respectively. Then there exists an adversary $\mathcal{B}_{\mathsf{BDHDE}}$ that breaks the (m,k)-Bilateral Diffie-Hellman Decision Exponent assumption (where $\tau = 1$ as in Definition 16), such that*

$$\left|\mathsf{Adv}_{\mathsf{G}_{i^\star-1},\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_{i^\star},\mathcal{A}}(\lambda)\right| \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda,m,k).$$

Proof. From any adversary $\mathcal{A}$ that distinguishes between games $\mathsf{G}_{i^\star-1}$ and $\mathsf{G}_{i^\star}$, we construct an adversary $\mathcal{B}$ that breaks the (m,k)-BDHDE assumption.

By recalling the distributions of the BDHDE problem for the parameter tuple (m,k):

$$
\begin{array}{lll}
\multicolumn{3}{l}{\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(1^\lambda)} \\
\alpha, r, t \leftarrow\!\$\, \mathbb{Z}_q^* & \boldsymbol{a}, \boldsymbol{b} \leftarrow\!\$\, \mathbb{Z}_q^{*k} & \\
\boldsymbol{z}_w \leftarrow\!\$\, \mathbb{Z}_q^{*k} & \boldsymbol{D}_w \leftarrow \mathsf{diag}(\boldsymbol{z}_w) & \forall w \in [3] \\
\boldsymbol{K}' \leftarrow rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_3 & \boldsymbol{Z}' \leftarrow\!\$\, \mathbb{Z}_q^{k\times k} & \\
d, e \leftarrow\!\$\, \mathbb{Z}_q^* & \boldsymbol{X}, \boldsymbol{Y} \leftarrow\!\$\, \mathbb{Z}_q^{k\times k} & \\
\boldsymbol{u}_j \leftarrow \alpha^j r\boldsymbol{a} & \boldsymbol{v}_j \leftarrow \alpha^j t\boldsymbol{b} & \forall j \in [m] \\
\boldsymbol{U}_j \leftarrow \alpha^j rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1 & \boldsymbol{V}_j \leftarrow \alpha^j rt\boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b}) & \forall j \in [m] \\
\boldsymbol{g} \leftarrow d\boldsymbol{a} & \boldsymbol{h} \leftarrow e\boldsymbol{b} & \\
\boldsymbol{G} \leftarrow dt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1 & \boldsymbol{H} \leftarrow er\boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b}) &
\end{array}
$$

$\mathcal{D}'_{\mathsf{b}'}$ outputs

$$
\mathsf{D}' := \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, [\boldsymbol{g}]_1, [\boldsymbol{G}^{(\mathsf{b}')}]_1, [\boldsymbol{K}'^{(\mathsf{b}')}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, [\boldsymbol{h}]_2, [\boldsymbol{H}^{(\mathsf{b}')}]_2 \end{cases}
$$

$$
\text{where } \begin{cases} \boldsymbol{G}^{(0)} = \boldsymbol{G}, \boldsymbol{H}^{(0)} = \boldsymbol{H}, \boldsymbol{K}^{(0)} = \boldsymbol{K}' \\ \boldsymbol{G}^{(1)} = \boldsymbol{X}, \boldsymbol{H}^{(1)} = \boldsymbol{Y}, \boldsymbol{K}^{(1)} = \boldsymbol{Z}' \end{cases},
$$

we proceeds with the reduction as follows:

- At the beginning of the game, $\mathcal{B}$ samples a challenge bit $\mathsf{b} \leftarrow\!\$\, \{0,1\}$.
- $\mathcal{B}$ receives the output D' from its challenge BDHDE instance and proceeds as follows:
  - If $\mathsf{b} = 1$, there is no difference between $\mathsf{G}_{i^\star-1}$ and $\mathsf{G}_{i^\star}$, $\mathcal{B}$ only needs to sample the common output D as specified in two games on its own.
  - If $\mathsf{b} = 0$, $\mathcal{B}$ embeds the output D' into the output D of the game via the following assignments:

$$
\begin{array}{rl}
[\boldsymbol{g}_{i^\star}]_1 \leftarrow [\boldsymbol{g}]_1, & [\boldsymbol{g}_i]_1 \leftarrow d'_i[\boldsymbol{u}_1]_1 \ \forall i \in [\tau] \setminus \{i^\star\} \\
[\boldsymbol{h}_{i^\star}]_2 \leftarrow [\boldsymbol{h}]_2, & [\boldsymbol{h}_i]_2 \leftarrow e'_i[\boldsymbol{v}_1]_2 \ \forall i \in [\tau] \setminus \{i^\star\} \\
[\boldsymbol{G}_{i^\star}]_1 \leftarrow [\boldsymbol{G}^{(\mathsf{b}')}]_1, & [\boldsymbol{G}_i^{(b)}]_1 \leftarrow\!\$\, \mathbb{G}_1^{k\times k} \ \forall i \geq (i^\star+1), \\
& [\boldsymbol{G}_i^{(b)}]_1 \leftarrow d'_i[\boldsymbol{U}_1]_1 \ \forall i \geq (i^\star+1) \\
[\boldsymbol{H}_{i^\star}]_2 \leftarrow [\boldsymbol{H}^{(\mathsf{b}')}]_2, & [\boldsymbol{H}_i^{(b)}]_2 \leftarrow\!\$\, \mathbb{G}_2^{k\times k} \ \forall i \geq (i^\star+1), \\
& [\boldsymbol{H}_i^{(b)}]_2 \leftarrow e'_i[\boldsymbol{V}_1]_2 \ \forall i \geq (i^\star+1)
\end{array}
$$

    where $d'_i, e'_i \leftarrow\!\$\, \mathbb{Z}_q^*$ for all $i \in [\tau] \setminus \{i^*\}$. Furthermore, within the case $\mathsf{b} = 0$:
    - If $i^\star = 1$, $\mathcal{B}$ is constructed to assign $[\boldsymbol{K}^{(b)}]_1 \leftarrow [\boldsymbol{K}'^{(b')}]_1$.
    - If $i^\star > 1$, $\mathcal{B}$ is constructed to assign $[\boldsymbol{K}^{(b)}]_1 \leftarrow\!\$\, \mathbb{G}_1^{k\times k}$, which means in this case the term $[\boldsymbol{K}'^{(b')}]_1$ from D' can be omitted.
- $\mathcal{B}$ sends the output D that it generates to $\mathcal{A}$. In particular, when $\mathsf{b} = 0$, the output is written as follows

$$
\mathsf{D} = \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j\in[m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{G}_i^{(\mathsf{b})}]_1\}_{i\in[\tau]}, [\boldsymbol{K}^{(\mathsf{b})}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j\in[m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{H}_i^{(\mathsf{b})}]_2\}_{i\in[\tau]} \end{cases}
$$

– At the end of the game, $\mathcal{B}$ outputs what $\mathcal{A}$ outputs.

It remains to show that the distribution $\mathcal{D}'_{\mathsf{b}'}$ of the (m,k)-BDHDE instance defines either the game $\mathsf{G}_{i^\star-1}$ or the game $\mathsf{G}_{i^\star}$ for the case $\mathsf{b}=0$. To this end, it suffices to consider the distribution of the assigned elements during reduction by elaborating them follows

$$\begin{aligned} \boldsymbol{g}_{i^\star} &= [d\boldsymbol{a}]_1, \quad \boldsymbol{g}_i = (d'_i\alpha r)\boldsymbol{a}\ \forall i \in [\tau]\setminus\{i^\star\} \\ \boldsymbol{h}_{i^\star} &= [e\boldsymbol{b}]_2, \quad \boldsymbol{h}_i = (e'_i\alpha t)\boldsymbol{b}\ \forall i \in [\tau]\setminus\{i^\star\} \end{aligned}$$

$$\boldsymbol{G}_{i^\star} = \begin{cases} \boldsymbol{G} = dt(\boldsymbol{a}\otimes\boldsymbol{b})\boldsymbol{D}_1 & \text{if } \mathsf{b}'=0 \\ \boldsymbol{X} \leftarrow \mathbb{Z}_q^{k\times k} & \text{if } \mathsf{b}'=1 \end{cases}, \begin{cases} \boldsymbol{G}_i^{(b)} \leftarrow\!\$\ \mathbb{Z}_q^{k\times k} & \forall i \geq (i^\star+1), \\ \boldsymbol{G}_i^{(b)} \leftarrow (d'_i\alpha r)t(\boldsymbol{a}\otimes\boldsymbol{b})\boldsymbol{D}_1 & \forall i \geq (i^\star+1) \end{cases}$$

$$\boldsymbol{H}_{i^\star} = \begin{cases} \boldsymbol{H} = er\boldsymbol{D}_2(\boldsymbol{a}\otimes\boldsymbol{b}) & \text{if } \mathsf{b}'=0 \\ \boldsymbol{Y} \leftarrow\!\$\ \mathbb{Z}_q^{k\times k} & \text{if } \mathsf{b}'=1 \end{cases}, \begin{cases} \boldsymbol{H}_i^{(b)} \leftarrow\!\$\ \mathbb{G}_2^{k\times k} & \forall i \geq (i^\star+1), \\ \boldsymbol{H}_i^{(b)} \leftarrow (e'_i\alpha t)r\boldsymbol{D}_2(\boldsymbol{a}\otimes\boldsymbol{b}) & \forall i \geq (i^\star+1) \end{cases}$$

By parsing $d_{i^\star} := d$, $e_{i^\star} := e$, $d_i := d'_i\alpha r$, $e_i := e'_i\alpha t$ for all $i \in [\tau]\setminus\{i^\star\}$, the scalars in $\{d_i, e_i\}_{i\in[\tau]}$ are thus uniformly random over $\mathbb{Z}_q^*$. The resulting distribution of D is then identical to that of $\mathsf{G}_{i^\star-1}$ when $\mathsf{b}'=0$, and identical to that of $\mathsf{G}_{i^\star}$ when $\mathsf{b}'=1$. Therefore, if $\mathcal{A}$ can distinguish between these two games, then $\mathcal{B}$ can guess the bit b' with the same advantage. This completes the proof. □

We now prove the transitions between security games that are specified in the proof of Theorem 7 for the main BDSE assumption.

Lemma 8 (Indistinguishability between G_1 and G_2 in Theorem 7). *For any adversary $\mathcal{A}$, let $\mathsf{Adv}_{\mathsf{G}_1,\mathcal{A}}(\lambda)$ and $\mathsf{Adv}_{\mathsf{G}_2,\mathcal{A}}(\lambda)$ be its advantages in games G_1 and G_2 of Theorem 7 respectively. Then there exists an adversary $\mathcal{B}_{\mathsf{DE}}$ that breaks the IND-CPA game of the dual encoding system with parameters $r=4k$, $m=n=k$, $q_{\mathsf{LEnc}} = q_{\mathsf{REnc}} = p(\lambda)+m+k+2$, such that*

$$|\mathsf{Adv}_{\mathsf{G}_1,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_2,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{DE}}}(\lambda).$$

Proof. From any adversary $\mathcal{A}$ that distinguishes between games G_1 and G_2, we construct an adversary $\mathcal{B}$ of the dual encoding system as follows:

– At the beginning, $\mathcal{B}$ samples a challenge bit $\mathsf{b} \leftarrow\!\$\ \{0,1\}$ and receives public parameters pp that contain a prime-order group $(q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e})$ from the dual encoding setup. $\mathcal{B}$ sends pp to $\mathcal{A}$.
– To simulate each element $\mathfrak{c}$ of the set $\mathcal{E} := \{\{\mathfrak{g}_{1,i}, \mathfrak{g}_{2,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_1, \tilde{\mathfrak{g}}_2, \{\mathfrak{r}_{1,h}, \mathfrak{r}_{1,}\}_{h\in[p(\lambda)]}, \{\mathfrak{u}_j, \mathfrak{v}_j\}_{j\in[m]}, \mathfrak{t}_1, \mathfrak{t}_2\}$, $\mathcal{B}$ generates the challenge messages as $(\boldsymbol{m}_0, \boldsymbol{m}_1) \in \mathbb{Z}_q^{k\times 4k} \times \mathbb{Z}_q^{k\times 4k}$ where $\boldsymbol{m}_0$ is what $\mathfrak{c}$ encodes in D_1 and $\boldsymbol{m}_1$ is what $\mathfrak{c}$ encodes in D_2.
  - If $\mathfrak{c} \in \mathfrak{G}_1$, $\mathcal{A}$ sends $([\boldsymbol{m}_0]_1, [\boldsymbol{m}_1]_1)$ to the $\mathbb{G}_1$-encoding oracle $\mathsf{LEnc}^2_{\mathsf{b}'}$.
  - If $\mathfrak{c} \in \mathfrak{G}_2$, $\mathcal{A}$ sends $([\boldsymbol{m}_0]_2, [\boldsymbol{m}_1]_2)$ to the $\mathbb{G}_2$-encoding oracle $\mathsf{REnc}^2_{\mathsf{b}'}$.
– At the end, $\mathcal{B}$ outputs what $\mathcal{A}$ outputs.

From above, if the hidden bit in the dual encoding game $\mathsf{b}'=0$, then $\mathcal{B}$ is playing as the challenger in G_1. If $\mathsf{b}'=1$, then $\mathcal{B}$ is playing as the challenger in G_2. We note that the group description bgd are completely defined by the parameters pp, the common reference string crs, the elements $\{\mathfrak{g}_{1,i}, \mathfrak{g}_{2,i}\}_{i\in[k]}$, $\tilde{\mathfrak{g}}_1$, $\tilde{\mathfrak{g}}_2$, $\{\mathfrak{r}_{1,i}, \mathfrak{r}_{1,i}\}_{i\in[p(\lambda)]}$ and its respective simulated proofs $\{\pi_{\mathfrak{g}_{1,i}}, \pi_{\mathfrak{g}_{2,i}}\}_{i\in[k]}$, $\pi_{\tilde{\mathfrak{g}}_1}$, $\pi_{\tilde{\mathfrak{g}}_2}$, $\{\pi_{\mathfrak{r}_{1,i}}, \pi_{\mathfrak{r}_{1,i}}\}_{i\in[p(\lambda)]}$.

We now show that all queries made by $\mathcal{A}$ satisfy the admissibility condition of the dual encoding game. To this end, in the output D_2 of G_2 (see Theorem 7), we re-highlight the game modifications as follows:

$$
\begin{aligned}
\mathfrak{g}_{1,i} &= [\![\ x_{i,1}\boldsymbol{a}_1,\ (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]},\ (x_{i,1}t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{A}_i)\boldsymbol{D}_1,\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1\\
\tilde{\mathfrak{g}}_1 &= [\![\ \tilde{x}_1\boldsymbol{a}_1,\ (\mathbf{0})_{l\in[2:k]},\ \tilde{x}_1t_1\boldsymbol{T}_1\boldsymbol{D}_1,\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1\\
\mathfrak{r}_{1,h} &= [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1\\
\mathfrak{u}_j &= [\![\ \alpha^j r_1\boldsymbol{a}_1,\ (\mathbf{0})_{l\in[2:k]},\ \alpha^j r_1t_1\boldsymbol{T}_1\boldsymbol{D}_1,\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_1\\
\mathfrak{t}_1 &= [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \boldsymbol{D}_2^{-1},\ (r_1t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{C})\boldsymbol{D}_3]\!]_1
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{g}_{2,i} &= [\![\ y_{i,1}\boldsymbol{b}_1,\ (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ (y_{i,1}r_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{B}_i)^\intercal\boldsymbol{D}_2,\ \mathbf{0}_{k\times k}]\!]_2\\
\tilde{\mathfrak{g}}_2 &= [\![\ \tilde{y}_1\boldsymbol{b}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \tilde{y}_1r_1\boldsymbol{T}_1{}^\intercal\boldsymbol{D}_2,\ \mathbf{0}_{k\times k}]\!]_2\\
\mathfrak{r}_{2,h} &= [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k},\ \mathbf{0}_{k\times k}]\!]_2\\
\mathfrak{v}_j &= [\![\ \alpha^j t_1\boldsymbol{b}_1,\ (\mathbf{0})_{l\in[2:k]},\ \mathbf{0}_{k\times k},\ \alpha^j r_1t_1\boldsymbol{T}_1{}^\intercal\boldsymbol{D}_2,\ \mathbf{0}_{k\times k}]\!]_2\\
\mathfrak{t}_2 &= [\![\ \mathbf{0},\ (\mathbf{0})_{l\in[2:k]},\ \boldsymbol{D}_1^{-1},\ \mathbf{0}_{k\times k},\ \boldsymbol{D}_3^{-1}]\!]_2
\end{aligned}
$$

– The gray slots correspond to modifications within $\mathfrak{t}_1$ and $\mathfrak{t}_2$ themselves. Instead of containing $(r_1\boldsymbol{a}_1, (\mathsf{b}r_l\boldsymbol{a}_l)_{l\in[2:k]})$ in $\mathfrak{t}_1$ and $(t_1\boldsymbol{b}_1, (\mathsf{b}t_l\boldsymbol{b}_l)_{l\in[2:k]})$ in $\mathfrak{t}_2$, these slots contain $(\mathbf{0}, (\mathbf{0})_{l\in[2:k]})$ in both elements.

– The blue slots correspond to modifications that preserve pairing results between $\mathfrak{t}_2$ and each element of $\{\{\mathfrak{g}_{1,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_1, \{\mathfrak{u}_j\}_{j\in[m]}\}$. For each element in this set, the blue slot contains its pairing with $\mathfrak{t}_2$, while the blue slot in $\mathfrak{t}_2$ contains the matrix $\boldsymbol{D}_1^{-1}$, which is cancelled out upon pairing. More precisely, we have:

$$
\begin{aligned}
e(\mathfrak{g}_{1,i}, \mathfrak{t}_2) &= [(x_{i,1}t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{A}_i)\boldsymbol{D}_1\boldsymbol{D}_1^{-1}]_T\\
&= [x_{i,1}t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{A}_i]_T\\
&= [x_{i,1}t_1(\boldsymbol{a}_1\otimes\boldsymbol{b}_1) + \mathsf{b}\sum_{l\in[2:k]} x_{i,l}t_l(\boldsymbol{a}_l\otimes\boldsymbol{b}_l)]_T
\end{aligned}
$$

$$
\begin{aligned}
e(\tilde{\mathfrak{g}}_1, \mathfrak{t}_2) &= [(\tilde{x}_1t_1\boldsymbol{T}_1)\boldsymbol{D}_1\boldsymbol{D}_1^{-1}]_T\\
&= [\tilde{x}_1t_1\boldsymbol{T}_1]_T\\
&= [\tilde{x}_1t_1(\boldsymbol{a}_1\otimes\boldsymbol{b}_1)]_T
\end{aligned}
$$

$$
\begin{aligned}
e(\mathfrak{u}_j, \mathfrak{t}_2) &= [(\alpha^j r_1t_1\boldsymbol{T}_1)\boldsymbol{D}_1\boldsymbol{D}_1^{-1}]_T\\
&= [\alpha^j r_1t_1\boldsymbol{T}_1]_T\\
&= [\alpha^j r_1t_1(\boldsymbol{a}_1\otimes\boldsymbol{b}_1)]_T
\end{aligned}
$$

– The magenta slots correspond to modifications that preserve the pairing results between $\mathfrak{t}_1$ and each element of $\{\{\mathfrak{g}_{2,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_2, \{\mathfrak{v}_j\}_{j\in[m]}\}$. For each element in this set, the magenta slot contains its pairing with $\mathfrak{t}_1$, while the magenta slot in $\mathfrak{t}_1$ contains the matrix $\boldsymbol{D}_2^{-1}$, which is cancelled out

upon pairing. More precisely, we have:

$$\begin{aligned} e(\mathfrak{t}_1, \mathfrak{g}_{2,i}) &= [\boldsymbol{D}_2^{-1}\boldsymbol{D}_2(y_{i,1}r_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{B}_i)]_T \\ &= [y_{i,1}r_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{B}_i]_T \\ &= [y_{i,1}r_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \mathsf{b}\sum_{l\in[2:k]} y_{i,l}r_l(\boldsymbol{a}_l \otimes \boldsymbol{b}_l)]_T \end{aligned}$$

$$\begin{aligned} e(\mathfrak{t}_1, \tilde{\mathfrak{g}}_2) &= [\boldsymbol{D}_2^{-1}\boldsymbol{D}_2(\tilde{y}_1r_1\boldsymbol{T}_1)]_T \\ &= [\tilde{y}_1r_1\boldsymbol{T}_1]_T \\ &= [\tilde{y}_1r_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1)]_T \end{aligned}$$

$$\begin{aligned} e(\mathfrak{t}_1, \mathfrak{v}_j) &= [\boldsymbol{D}_2^{-1}\boldsymbol{D}_2(\alpha^j r_1t_1\boldsymbol{T}_1)]_T \\ &= [\alpha^j r_1t_1\boldsymbol{T}_1]_T \\ &= [\alpha^j r_1t_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1)]_T \end{aligned}$$

– The teal slots correspond to the modification that preserve the pairing result between $\mathfrak{t}_1$ and $\mathfrak{t}_2$. In $\mathfrak{t}_1$, the highlighted slot contains its pairing with $\mathfrak{t}_2$, while the highlighted slot in $\mathfrak{t}_2$ contains the matrix $\boldsymbol{D}_3^{-1}$, which is cancelled out upon pairing. More precisely, we have:

$$\begin{aligned} e(\mathfrak{t}_1, \mathfrak{t}_2) &= [(r_1t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{C})\boldsymbol{D}_3\boldsymbol{D}_3^{-1}]_T \\ &= [r_1t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{C}]_T \\ &= [r_1t_1(\boldsymbol{a}_1 \otimes \boldsymbol{b}_1) + \mathsf{b}\sum_{l\in[2:k]} r_lt_l(\boldsymbol{a}_l \otimes \boldsymbol{b}_l)]_T \end{aligned}$$

As shown, the pairing results between the sets $\{\mathfrak{t}_2\}$ and $\{\{\mathfrak{g}_{1,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_1, \{\mathfrak{u}_j\}_{j\in[m]}\}$, between the sets $\{\mathfrak{t}_1\}$ and $\{\{\mathfrak{g}_{2,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_2, \{\mathfrak{v}_j\}_{j\in[m]}\}$, and between $\mathfrak{t}_1$ and $\mathfrak{t}_2$, remain unchanged when transitioning from G_0 to G_1.

Moreover, the modified slots in the elements of $\{\{\mathfrak{g}_{1,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_1, \{\mathfrak{u}_j\}_{j\in[m]}\}$ are disjoint from those in $\{\{\mathfrak{g}_{2,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_2, \{\mathfrak{v}_j\}_{j\in[m]}\}$. Consequently, the pairing results between these two sets are unaffected by the modifications.

Finally, the randomizers $\{\mathfrak{r}_{1,h}\}_{h\in[p(\lambda)]}$ and $\{\mathfrak{r}_{2,h}\}_{h\in[p(\lambda)]}$ are the identity elements $1_{\mathfrak{G}_1}$ and $1_{\mathfrak{G}_2}$, respectively. Hence, any pairing involving a randomizer always evaluates to $1_{\mathfrak{G}_T}$.

It now follows that all pairing results between the sets $\{\{\mathfrak{g}_{1,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_1, \{\mathfrak{r}_{1,h}\}_{h\in[p(\lambda)]}, \{\mathfrak{u}_j\}_{j\in[m]}, \mathfrak{t}_1\}$ and $\{\{\mathfrak{g}_{2,i}\}_{i\in[k]}, \tilde{\mathfrak{g}}_2, \{\mathfrak{r}_{2,h}\}_{h\in[p(\lambda)]}, \{\mathfrak{v}_j\}_{j\in[m]}, \mathfrak{t}_2\}$ remain unchanged. Therefore, all encoding queries made by $\mathcal{B}$ satisfy the admissibility condition of the dual encoding game. This means if $\mathcal{B}$ wins then $\mathcal{A}$ also wins.

Finally, by the parameters of the (m, k)-BSDE problem, we have $r = 4k$ and $m = n = k$. The numbers of $\mathbb{G}_1$-encoding and $\mathbb{G}_2$-encoding queries made by $\mathcal{A}$ equal the numbers of $\mathfrak{G}_1$ and $\mathfrak{G}_2$ elements, respectively, in the output set $\mathcal{E}$. This means $q_{\mathsf{LEnc}} = q_{\mathsf{REnc}} = p(\lambda) + m + k + 2$. The proof of the lemma is therefore complete. □

Lemma 9 (Indistinguishability between G_3 and G_4 in Theorem 7). *For any adversary $\mathcal{A}$, let $\mathsf{Adv}_{\mathsf{G}_3,\mathcal{A}}(\lambda)$ and $\mathsf{Adv}_{\mathsf{G}_4,\mathcal{A}}(\lambda)$ be its advantages in games G_3 and G_4 of Theorem 7 respectively. Then*

there exists an adversary $\mathcal{B}_{\mathsf{BDHDE}}$ that breaks the $(m, k+1, k)$-Bilateral Diffie-Hellman Decision Exponent assumption, such that

$$|\mathsf{Adv}_{\mathsf{G}_3,\mathcal{A}}(\lambda) - \mathsf{Adv}_{\mathsf{G}_4,\mathcal{A}}(\lambda)| \leq \mathsf{Adv}_{\mathcal{B}_{\mathsf{BDHDE}}}(\lambda, m, k+1, k).$$

Proof. From any adversary $\mathcal{A}$ that distinguishes between games G_3 and G_4, we construct an adversary $\mathcal{B}$ of the $(m, k+1, k)$-BDHDE assumption.

The proof will no longer rely on the security of the dual encoding system, so the reduction algorithm can generate the FE secret key on its own. As a consequence, we implicitly assume that the group description bgd of the splitting bilinear group can also be generated by this algorithm, and that elements of the prime-order bilinear group can be encoded into the splitting bilinear group in a straightforward manner.

By recalling the distributions of the BDHDE assumption for the parameter tuple $(m, k+1, k)$:

$$\begin{array}{lll}
\mathsf{bgd}_{\mathcal{P}} = (q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, \hat{e}) \leftarrow \mathcal{P}(1^\lambda) & & \\
\alpha, r, t \leftarrow\!\!\$\ \mathbb{Z}_q^* & \boldsymbol{a}, \boldsymbol{b} \leftarrow\!\!\$\ \mathbb{Z}_q^{*k} & \\
\boldsymbol{z}_w \leftarrow\!\!\$\ \mathbb{Z}_q^{*k} & \boldsymbol{D}_w \leftarrow \mathsf{diag}(\boldsymbol{z}_w) & \forall w \in [3] \\
\boldsymbol{K}^{(0)} \leftarrow rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_3 & \boldsymbol{K}^{(1)} \leftarrow\!\!\$\ \mathbb{Z}_q^{k \times k} & \\
d_i, e_i \leftarrow\!\!\$\ \mathbb{Z}_q^* & \boldsymbol{G}_i^{(1)}, \boldsymbol{H}_i^{(1)} \leftarrow\!\!\$\ \mathbb{Z}_q^{k \times k} & \forall i \in [k+1] \\
\boldsymbol{u}_j \leftarrow \alpha^j r \boldsymbol{a} & \boldsymbol{v}_j \leftarrow \alpha^j t \boldsymbol{b} & \forall j \in [m] \\
\boldsymbol{U}_j \leftarrow \alpha^j rt(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1 & \boldsymbol{V}_j \leftarrow \alpha^j rt \boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b}) & \forall j \in [m] \\
\boldsymbol{g}_i \leftarrow d_i \boldsymbol{a} & \boldsymbol{h}_i \leftarrow e_i \boldsymbol{b} & \forall i \in [k+1] \\
\boldsymbol{G}_i^{(0)} \leftarrow d_i t(\boldsymbol{a} \otimes \boldsymbol{b})\boldsymbol{D}_1 & \boldsymbol{H}_i^{(0)} \leftarrow e_i r \boldsymbol{D}_2(\boldsymbol{a} \otimes \boldsymbol{b}) & \forall i \in [k+1]
\end{array}$$

$\forall \mathsf{b}' \in \{0,1\}$, $\mathcal{D}^\star_{\mathsf{b}'}$ outputs

$$\mathsf{D}^\star = \mathsf{bgd}_{\mathcal{P}}, \begin{cases} [\boldsymbol{D}_1]_1, [\boldsymbol{D}_2^{-1}]_1, [\boldsymbol{D}_3]_1, \{[\boldsymbol{u}_j]_1, [\boldsymbol{U}_j]_1\}_{j \in [m]}, \{[\boldsymbol{g}_i]_1, [\boldsymbol{G}_i^{(\mathsf{b}')}]_1\}_{i \in [k+1]}, [\boldsymbol{K}^{(\mathsf{b}')}]_1 \\ [\boldsymbol{D}_1^{-1}]_2, [\boldsymbol{D}_2]_2, [\boldsymbol{D}_3^{-1}]_2, \{[\boldsymbol{v}_j]_2, [\boldsymbol{V}_j]_2\}_{j \in [m]}, \{[\boldsymbol{h}_i]_2, [\boldsymbol{H}_i^{(\mathsf{b}')}]_2\}_{i \in [k+1]} \end{cases}$$

we proceeds with the reduction as follows:

- At the beginning of the game, $\mathcal{B}$ samples $\mathsf{b} \leftarrow\!\!\$\ \{0,1\}$ as the challenge bit of the games G_3 and G_4.
- $\mathcal{B}$ receives $\mathsf{D}^\star$ as a BDHDE instance. By using $\mathsf{bgd}_{\mathcal{P}}$, $\mathcal{B}$ constructs the splitting bilinear group and sends its description bgd to $\mathcal{A}$.
- $\mathcal{B}$ samples the following scalars
  - $\boldsymbol{a}_l, \boldsymbol{b}_l \leftarrow\!\!\$\ \mathbb{Z}_q^k$ for all $l \in [2:k]$,
  - $x_{i,l}, y_{i,l} \leftarrow\!\!\$\ \mathbb{Z}_q$ for all $i \in [k], l \in [2:k]$,
  - $r_l, t_l \leftarrow\!\!\$\ \mathbb{Z}_q$ for all $l \in [2:k]$,
  - $\boldsymbol{T}_l \leftarrow \boldsymbol{a}_l \otimes \boldsymbol{b}_l$ for all $l \in [2:k]$,
  - $\boldsymbol{A}_i \leftarrow \sum_{l \in [2:k]} x_{i,l} t_l \boldsymbol{T}_l$, $\boldsymbol{B}_i \leftarrow \sum_{l \in [2:k]} y_{i,l} r_l \boldsymbol{T}_l$ for all $i \in [k]$,
  - $\boldsymbol{C} \leftarrow \sum_{l \in [2:k]} r_l t_l \boldsymbol{T}_l$.
- To simulate the dual encodings in D_3 and D_4 of the games G_3 and G_4, respectively, $\mathcal{B}$ generates the following encodings

$$
\begin{aligned}
\mathfrak{g}_{1,i} &= [\![\quad \boldsymbol{g}_i, (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}, \boldsymbol{G}_i^{(\mathsf{b}')} + \mathsf{b}\boldsymbol{A}_i \boldsymbol{D}_1, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]\!]_1 \\
\tilde{\mathfrak{g}}_1 &= [\![\boldsymbol{g}_{k+1}, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{G}_{k+1}^{(\mathsf{b}')}, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{r}_{1,h} &= [\![\mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{u}_j &= [\![\boldsymbol{u}_j, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{U}_j, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]\!]_1 \\
\mathfrak{t}_1 &= [\![\mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, && \boldsymbol{D}_2^{-1}, \boldsymbol{K}^{(\mathsf{b}')} + \mathsf{b}\boldsymbol{C}\boldsymbol{D}_3]\!]_1
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{g}_{2,i} &= [\![\quad \boldsymbol{h}_i, (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \boldsymbol{H}_i^{(\mathsf{b}')\mathsf{T}} + \mathsf{b}\boldsymbol{B}_i^{\mathsf{T}} \boldsymbol{D}_2, && \mathbf{0}_{k\times k}]\!]_2 \\
\tilde{\mathfrak{g}}_2 &= [\![\boldsymbol{h}_{k+1}, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \boldsymbol{H}_{k+1}^{(\mathsf{b}')\,\mathsf{T}}, && \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{r}_{2,h} &= [\![\mathbf{0}, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{v}_j &= [\![\boldsymbol{v}_j, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \boldsymbol{V}_j^{\mathsf{T}}, && \mathbf{0}_{k\times k}]\!]_2 \\
\mathfrak{t}_2 &= [\![\mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{D}_1^{-1}, && \mathbf{0}_{k\times k}, && \boldsymbol{D}_3^{-1}]\!]_2
\end{aligned}
$$

Here, the prime-order group elements provided by the BDHDE instance are encoded into splitting-group elements by $\mathcal{B}$. These elements are highlighted in gray.

– At the end of the game, $\mathcal{B}$ outputs what $\mathcal{A}$ outputs.

It remains to show that the distribution $\mathcal{D}^{\star}_{\mathsf{b}'}$ of the BDHDE instance defines either the game G_3 or the game G_4. To do that, we substitute

$$
\begin{aligned}
&r_1 := r,\ t_1 := t,\ \boldsymbol{a}_1 := \boldsymbol{a},\ \boldsymbol{b}_1 := \boldsymbol{b}, \\
&x_{i,1} := d_i,\ y_{i,1} := e_i\ \forall i \in [k],\ \tilde{x}_1 := e_{k+1},\ \tilde{y}_1 := d_{k+1}
\end{aligned}
$$

The analysis on the value of b' is as follows.

– When $\mathsf{b}' = 0$, we have

$$
\begin{aligned}
\mathfrak{g}_{1,i} &= [\ x_{i,1}\boldsymbol{a}_1, (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}, (x_{i,1}t_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{A}_i)\boldsymbol{D}_1, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]_1 \\
\tilde{\mathfrak{g}}_1 &= [\ \tilde{x}_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \tilde{x}_1 t_1 \boldsymbol{T}_1\boldsymbol{D}_1, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]_1 \\
\mathfrak{r}_{1,h} &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]_1 \\
\mathfrak{u}_j &= [\ \alpha^j r_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \alpha^j r_1 t_1 \boldsymbol{T}_1\boldsymbol{D}_1, && \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]_1 \\
\mathfrak{t}_1 &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, && \boldsymbol{D}_2^{-1}, (r_1 t_1 \boldsymbol{T}_1 + \mathsf{b}\boldsymbol{C})\boldsymbol{D}_3]_1
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{g}_{2,i} &= [\ y_{i,1}\boldsymbol{b}_1, (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]}, && \mathbf{0}_{k\times k}, (y_{i,1}r_1\boldsymbol{T}_1 + \mathsf{b}\boldsymbol{B}_i)^{\mathsf{T}}\boldsymbol{D}_2, && \mathbf{0}_{k\times k}]_2 \\
\tilde{\mathfrak{g}}_2 &= [\ \tilde{y}_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \tilde{y}_1 r_1 \boldsymbol{T}_1{}^{\mathsf{T}}\boldsymbol{D}_2, && \mathbf{0}_{k\times k}]_2 \\
\mathfrak{r}_{2,h} &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, && \mathbf{0}_{k\times k}]_2 \\
\mathfrak{v}_j &= [\ \alpha^j t_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, && \mathbf{0}_{k\times k}, \alpha^j t_1 r_1 \boldsymbol{T}_1{}^{\mathsf{T}}\boldsymbol{D}_2, && \mathbf{0}_{k\times k}]_2 \\
\mathfrak{t}_2 &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{D}_1^{-1}, && \mathbf{0}_{k\times k}, && \boldsymbol{D}_3^{-1}]_2
\end{aligned}
$$

where $\boldsymbol{T}_1 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1$. The distribution of dual encodings is then identical to that in G_3. Hence $\mathcal{D}^{\star}_0$ corresponds to G_3.

– When $\mathsf{b}' = 1$, we have

$$
\begin{aligned}
\mathfrak{g}_{1,i} &= [\ x_{i,1}\boldsymbol{a}_1, (x_{i,l}\boldsymbol{a}_l)_{l\in[2:k]}, \boldsymbol{X}_i, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]_1\\
\tilde{\mathfrak{g}}_1 &= [\ \tilde{x}_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{X}_{k+1}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]_1\\
\mathfrak{r}_{1,h} &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]_1\\
\mathfrak{u}_j &= [\ \alpha^j r_1\boldsymbol{a}_1, (\mathbf{0})_{l\in[2:k]}, \alpha^j r_1 t_1 \boldsymbol{T}_1\boldsymbol{D}_1, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]_1\\
\mathfrak{t}_1 &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{D}_2^{-1}, \boldsymbol{Z}]_1
\end{aligned}
$$

$$
\begin{aligned}
\mathfrak{g}_{2,i} &= [\ y_{i,1}\boldsymbol{b}_1, (y_{i,l}\boldsymbol{b}_l)_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{Y}_i^\intercal, \mathbf{0}_{k\times k}]_2\\
\tilde{\mathfrak{g}}_2 &= [\ \tilde{y}_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \boldsymbol{Y}_{k+1}^\intercal, \mathbf{0}_{k\times k}]_2\\
\mathfrak{r}_{2,h} &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}, \mathbf{0}_{k\times k}]_2\\
\mathfrak{v}_j &= [\ \alpha^j t_1\boldsymbol{b}_1, (\mathbf{0})_{l\in[2:k]}, \mathbf{0}_{k\times k}, \alpha^j t_1 r_1 \boldsymbol{T}_1{}^\intercal\boldsymbol{D}_2, \mathbf{0}_{k\times k}]_2\\
\mathfrak{t}_2 &= [\ \mathbf{0}, (\mathbf{0})_{l\in[2:k]}, \boldsymbol{D}_1^{-1}, \mathbf{0}_{k\times k}, \boldsymbol{D}_3^{-1}]_2
\end{aligned}
$$

where $\boldsymbol{T}_1 = \boldsymbol{a}_1 \otimes \boldsymbol{b}_1$, $\boldsymbol{X}_i \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times k}$, $\boldsymbol{Y}_i \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times k}$ for all $i \in [k+1]$ and $\boldsymbol{Z} \leftarrow\!\!\$\ \mathbb{Z}_q^{k\times k}$. The distribution of dual encodings is then identical to that in G_4. Hence $\mathcal{D}_1^\star$ corresponds to G_4.

Finally, if $\mathcal{A}$ wins, then $\mathcal{B}$ also wins. This completes the proof of the lemma. □